

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE  
DEPARTEMENT D'INFORMATIQUE



# **Mémoire**

***En vue d'obtention du diplôme***  
***Master en Informatique***  
***Spécialité CPI***

## **Thème**

***Conception et réalisation***  
***d'une application Client/Serveur(J2EE)***

Direction Régionale  
Tizi Ouzou

***pour la gestion des activités de***  
***la DGSI Sonelgaz***



Cas: Direction de distribution Tizi ouzou

Réalisé par :

***Melle Galleze Hadjila***

Encadreur :

***Mr OUAMRANE***

***Année: 2013/2014***

## **Résumé**

Notre travail aborde la problématique de coopération et collaboration entre les utilisateurs distants dans les applications Client/serveur multi niveaux orientées services basées sur la technologie des services web et réseaux.

Alors le projet porte sur la conception et la réalisation d'un système 3-tiers avec une centralisation de données, en utilisant le patron de conception logiciel J2EE, qui est un standard de développement des applications d'entreprises distribuées et multi-niveaux à base des composants.

L'objectif est alors le suivi des activités de la division de gestion des systèmes informatiques (DGSI) de sonelgaz et assurer un environnement interactif entre toutes les personnes concernées par ces activités, et ce via un réseau.

# Remerciements

*Le plaisir avec lequel je réserve ces remerciements est en signe de gratitude et de profonde reconnaissance à tous ceux qui ont voulu apporter l'assistance nécessaire au déroulement de mon travail.*

*Tout d'abord merci Dieu qui m'a avoir donné le courage d'accomplir ce projet.*

*Je veux encore remercier mon encadreur Mr OUAMRANE qui a accepté de me diriger pendant la période de mon travail ainsi que la confiance qu'il m'a témoigné que j'ai pu capable de réussir dans mon projet.*

*Je tiens à exprimer mon profond respect et mes vifs remerciements envers les membres de jury pour l'honneur qu'ils m'ont fait en acceptant le jugement de ce travail.*

*Que la direction de distribution Sonelgaz de Tizi Ouzou trouve ici mes plus vifs remerciements pour sa bienveillance.*

*J'adresse encore mes remerciements aux membres de la DGSI parmi eux Mr Becha qui trouve dans mon travail l'expression de mon profond respect et mon infinie reconnaissance, grâce à son soutien constant qu'il m'a donné le courage et la volonté pour l'élaboration de ce travail.*

*Un remerciement ensoleillé à tous mes enseignants de la faculté pour leurs efforts qui ont guidé mes pas à enrichir mes connaissances.*

*Il est indispensable aussi de ne pas rater cette occasion pour exprimer mes profonds respects et mes meilleurs remerciements à toute personne qui a contribué de près ou de loin à ce que ce travail voit le jour.*

# ❧ *Dédicaces* ❧

*Je dédie ce modeste travail à :*

*Mes parents (Mère et père que je souhaite que Dieu les protège)*

*Mon frère Ahmed, sa femme et ses enfants*

*Mon frère Belkacem*

*Mon frère Hamid et sa fiancée*

*Mon frère Omar et sa femme*

*Mes sœurs pour lesquelles je souhaite une belle vie, une bonne chance, et une  
meilleure réussite*

*Mes oncles, mes tentes, et les cousines maternelle*

*Tous ceux qui comptent pour moi*

# Sommaire

Introduction générale .....	1
-----------------------------	---

## Partie1 : Les réseaux

### Chapitre I : Réseaux et architecture Client/serveur

I. 1.1. Introduction : .....	3
I. 1.2. Définition des réseaux informatiques.....	3
I. 1.3. Les domaines d'application des réseaux .....	4
I. 1.4. Les objectifs des réseaux.....	4
I. 1.5. Typage et classification des réseaux informatiques.....	6
I. 1.5.1. réseaux de type filaire.....	6
A. Les réseaux locaux.....	6
➤ Topologie des réseaux locaux.....	6
• Topologie en étoile.....	8
• Topologie e anneau.....	9
• Topologie en bus.....	10
B. Les réseaux métropolitains (MAN).....	11
C. Les réseaux étendus (WAN).....	11
I. 1.5.2. Réseaux sans fils.....	12
I. 1.5.2.1. Les réseaux personnels ou Bluetooth.....	12
I. 1.5.2.2. Les réseaux locaux sans fils (WLAN).....	12
I. 1.5.2.3. Les réseaux métropolitains sans fils (WMAN).....	12
I. 1.5.2.4. Les réseaux longs distance.....	12
I. 1.6. Architecture des réseaux et stratégies de conception.....	13

## Partie2 : L'architecture Client/serveur

I. 2.1. L'apparition du modèle Client/serveur.....	17
I. 2.2. Environnement Client/serveur.....	17
I. 2.2. 1. Présentation du modèle Client/serveur.....	17
I. 2.2.2. Structure du mode Client/serveur.....	18
I. 2.2.3. Avantages du modèle Client/serveur.....	18
I. 2.2.4. Inconvénients du modèle Client/serveur.....	19
I. 2.2.5. Le processus de fonctionnement d'une requête SQL en Client/serveur.....	20
I. 2.2.6. le paradigme Client/serveur.....	20
I. 2.3. Classification des architectures Client/serveur.....	21
I. 2.3.1. L'architecture centralisée.....	21
I. 2.3.2. L'architecture à serveur de fichiers.....	22
I. 2.3.3. L'architecture à serveur de bases de données.....	23

I. 2.4. Types d'architectures Client/serveur.....	24
I. 2.4.1. L'architecture à deux niveaux.....	24
I. 2.4.2. L'architecture à trois niveaux.....	25
I. 2.4.3. Comparaison entre les architectures à deux et à trois niveaux.....	26
I. 2.4.4. L'architecture à n niveaux.....	28
Conclusion.....	28

## **Chapitre II : Le traitement coopératif**

I. Introduction.....	29
II. Les aspects coopérants des applications réparties.....	29
III. Les caractéristiques du traitement coopératif réparti.....	31
IV. Organisation et traitement coopératif.....	32
Conclusion.....	33

## **Chapitre III : Généralités sur UML**

I. Historique et origine de l'UML.....	34
II. L'intérêt d'utilisation UML.....	34
III. Stades de développement.....	34
IV. Style de cycle de vie.....	35
IV. 1. Développement en cascade.....	35
IV. 2. Développement itératif.....	36
V. Présentation des diagrammes UML.....	36
Conclusion.....	37

## **Chapitre IV: Principe des patrons de conception logiciels**

### **Partie1 : Les patrons de conception**

I.1.Introduction.....	39
I.2.Historique des patrons logiciels.....	39
I.3. L'utilité des patrons de conceptions dans le domaine du génie logiciel.....	39
I.4. Les avantages des patrons de conceptions .....	40
I.5. Les inconvénients des patrons de conceptions.....	40
I.6. Classification du GoF pour les patrons de conception.....	40
I.7. Les parties essentielles des patrons de conception.....	43

### **Partie 2 : Le patron de conception logiciel J2EE**

I. Introduction à J2EE.....	44
II. Présentation de J2EE.....	44
III. Les API de J2EE.....	45
IV. L'environnement d'exécution des applications J2EE.....	47
IV. 1. Les services proposés par la plate-forme J2EE .....	47
IV. 2. L'assemblage et le déploiement d'applications J2EE .....	48
V. Principes de J2EE .....	48
VI. Composants J2EE.....	49
Conclusion.....	49

## **Chapitre V: Présentation de l'organisme d'accueil**

I. Définition de sonelgaz.....	51
II. Décision de la direction générale de distribution.....	51
III. L'organigramme de la direction de distribution générale.....	52
IV. Les missions.....	52
V. Les attributions.....	52
VI. L'organigramme de la direction régionale.....	54
VII. Les attributions de la direction régionale.....	55
Conclusion.....	55

## **Chapitre IV: Etude de l'existant**

I. Présentation du champ d'étude.....	56
I.1. Organigramme du champ d'étude.....	56
I.2. Description des éléments du domaine d'étude.....	56
I.3. Présentation des fonctions de la DGSII.....	56
II. Analyse du champ d'étude.....	57
II.1. Introduction.....	57
II.2. Etude des postes de travail.....	57
II.3. Etude des documents.....	69
II.3.1. Définitions.....	69
II.3.2. Les types d'un document.....	69
II.3.3. Liste des documents.....	69
II.3.4. Etude détaillée de chaque document.....	69
II.4. Etude des fichiers.....	79
II.5. Etude des procédures.....	87
III.5.1. Définition d'une procédure.....	87
III.5.2. Liste des procédures.....	87
III.5.3. Notation graphique.....	87
III.5.4. Etude détaillée de chaque procédure de travail.....	89
Conclusion.....	106

## **Chapitre V: Analyse et conception**

### **Partie I : Etude préliminaire**

I. Modélisation de contexte.....	107
I.1. Les acteurs.....	107
I.2. Les messages.....	108
I.3. Le diagramme de contexte.....	109
II. Représentation des besoins exprimés par les utilisateurs.....	110
II.1. Identification des cas d'utilisation.....	110
II.1.1. Cas d'utilisation « Gestion de droit d'accès ».....	111
II.1.1.1. Description textuelle de cas d'utilisation.....	111
II.1.1.2. Description de cas d'utilisation à l'aide du tableau.....	111
II.1.1.3. Description graphique de cas d'utilisation« Gestion de droit d'accès »	



II.1.1.4. Scénario d cas d'utilisation « Gestion des droits d'accès ».....	112
II.1.2. Cas d'utilisation « Identification des utilisateurs »	
II.1.2.1. Description textuelle des cas d'utilisation.....	
II.1.2.2. Description de cas d'utilisation à l'aide du tableau.....	112
II.1.2.3. Description graphique de cas d'utilisation « Identification des utilisateurs»	113
II.1.2.4. Scénario du cas d'utilisation « Identification des utilisateurs».....	113
II.1.3. Cas d'utilisation « Approvisionnement papier ».....	113
II.1.3.1. Description textuelle des cas d'utilisation.....	113
II.1.3.2. Description de cas d'utilisation à l'aide du tableau.....	114
II.1.3.3. Description graphique de cas d'utilisation« Approvisionnement papier»...	114
II.1.3.4. Scénario du cas d'utilisation « Approvisionnement papier».....	115
II.1.4. Cas d'utilisation « Transfert papier ».....	115
II.1.4.1. Description textuelle de cas d'utilisation.....	115
II.1.4.2. Description de cas d'utilisation à l'aide du tableau.....	116
II.1.4.3. Description graphique de cas d'utilisation : « Transfert papier».....	116
II.1.4.4. Scénario du cas d'utilisation « Transfert papier».....	117
II.1.5. Cas d'utilisation « Demande et recensement de matériel entré à la DGSI»..	117
II.1.5.1. Description textuelle de cas d'utilisation.....	117
II.1.5.2. Description de cas d'utilisation à l'aide du tableau.....	118
II.1.5.3. Description graphique de cas d'utilisation : «Demande et recensement de matériel entré à la DGSI».....	118
II.1.5.4. Scénario du cas d'utilisation «Demande et recensement de matériel entré à la DGSI».....	119
II.1.6. Cas d'utilisation « Recensement de matériel sorti de la DGSI».....	119
II.1.6.1. Description textuelle de cas d'utilisation.....	119
II.1.6.2. Description de cas d'utilisation à l'aide du tableau.....	120
II.1.6.3. Description graphique de cas d'utilisation : «Recensement de matériel sorti de la DGSI».....	120
II.1.6.4. Scénario du cas d'utilisation « Recensement de matériel sorti de la DGSI».	121
II.1.7. Cas d'utilisation « Suivi des rapports d'intervention».....	122
II.1.7.1. Description textuelle de cas d'utilisation.....	122
II.1.7.2. Description de cas d'utilisation à l'aide du tableau.....	122
II.1.7.3. Description graphique de cas d'utilisation : «Suivi des rapports d'intervention».....	122
II.1.8. Cas d'utilisation « gestion des documents».....	123
II.1.8.1. Description textuelle de cas d'utilisation.....	123
II.1.8.2. Description de cas d'utilisation à l'aide du tableau.....	123
II.1.8.3. Description graphique de cas d'utilisation : «Gestion des documents».....	124
III. Relations entre cas d'utilisation.....	125
A. Inclusion.....	125
1. Inclusion de cas d'utilisation « identification d'utilisateur ».....	125



2. Inclusion de cas d'utilisation « Gestion des documents ».....	126
B. Extension	
1. Extension de cas d'utilisation « Gestion de parc informatique ».....	126
2. Extension de cas d'utilisation « Gestion de stock papier ».....	127
C. Diagramme de cas d'utilisation.....	127
III. Le packaging des cas d'utilisation.....	128
IV. Capture des besoins techniques.....	131
IV.1. Spécifications techniques de point de vue matériel.....	131
IV. 1.1. Les différentes architectures réseaux d'un système d'information.....	132
IV. 1.2. L'infrastructure réseau.....	132
IV. 1.2.1. Le principe de fonctionnement de réseau VPN.....	132
IV. 1.3. Déploiement de la solution.....	132
IV.2. Spécification techniques de point de vue logicielle.....	133
IV.2.1. Les outils de développement.....	133
IV.2.2. Identification des cas d'utilisation technique.....	134
1) Cas d'utilisation « Manipuler les objets ».....	135
2) Cas d'utilisation « Gérer l'intégrité ».....	135
3) Cas d'utilisation « Gérer la sécurité ».....	135
4) Cas d'utilisation « Gérer les erreurs ».....	136
IV.2.3) Elaboration du modèle de spécification logicielle.....	136
❖ Organisation en couches du modèle de spécification.....	136
 Partie II : Analyse de nouveau système.....	138
Section 1 : Analyse statique.....	138
Construction de dictionnaire de données.....	138
Section 2 : Analyse dynamique.....	139
I. Analyse des états de transition.....	139
☞ Objet demande.....	139
☞ Objet Approvisionnement.....	140
☞ Objet transfert (papier ou Matériel).....	140
☞ Objet Document.....	140
 Partie 3 : Conception.....	142

<b>I. Identification des scénarios.....</b>	<b>142</b>
I.1. Scénario de « Gestion des droits d'accès ».....	142
I.2. Scénario de « Identification Utilisateur ».....	143
I.3. Scénario de « Approvisionnement Papier ».....	143
I.4. Scénario de « Transfert Papier ».....	144
I.5. Scénario de « Recensement de matériel entré à la DGSi ».....	144
I.6. Scénario de « Matériel sorti ».....	145
I.7. Scénario de « Gestion des interventions ».....	146
I.8. Scénario de « Gestion des documents ».....	147
<b>II. Conception du système.....</b>	<b>149</b>
II. 1. La conception des couches logicielles .....	150
II.1.1. La couche Interface Homme Machine.....	150
II.1.2. La couche métier.....	151
II.1.3 La couche Infrastructure.....	151
II.2. Affinage de diagramme des classes.....	151
II.2.1. Conception des méthodes.....	151
<b>III. Diagramme de classes.....</b>	<b>153</b>
III.1.Liste des classes.....	153
III.2.Liste des classes d'association.....	156
III.3. Le diagramme des classes.....	156
IV. Le passage vers le modèle relationnel .....	158
V. Conception des objets.....	159
VI. La sécurité.....	159
<b>Conclusion.....</b>	<b>160</b>

## Chapitre VI: réalisation

<b>Section : I : Environnement et outils de développement.....</b>	<b>161</b>
<b>Introduction.....</b>	<b>161</b>
I.1. L'environnement de développement.....	161
I.1.1.Le langage de programmation java.....	161
I.1.2. Le langage XML .....	164
I.1.3.IDE (NetBeans 6.7).....	165
I.1.4. Le serveur d'application (Glassfish V2.1).....	166
I.1.5.Le SGBD (MYSQL 5).....	166
I.1.6.Le <u>Framework</u> (Hibernate).....	167
<b>Section II : Architecture de notre application</b>	
II.1. Introduction.....	168
II.2. Architecture de notre application.....	168
<b>II.3 : Implémentation des différentes parties de l'application.....</b>	<b>170</b>
II.3.1: Partie stockage.....	170
II.3.2. Partie Traitement.....	171

II.3.2.1. le projet EAGestionSonelgaz.....	171
II.3.2.2 : Le module EJBGestionSonelgaz.....	172
II.3.2.2.1.Configuration Hibernate de serveur Glassfish .....	173
II.3.2.2.2.Ajout d'une ressource JDBC au serveur Glassfish.....	174
II.3.2.2.3-Création d'une unité de persistance.....	175
II.3.2.2.4-Création des entités JPA .....	176
II.3.2.2.5-Création de la couche [dao] d'accès aux entités JPA.....	181
II.3.2.2.6-Création d'une classe exception.....	182
II.3.2.2.7.Le module TraitementsGestionSonelgaz.....	183
Section 3 : Les fonctionnalités de l'application.....	185
Conclusion.....	191
Conclusion générale.....	192

# Introduction générale

Comme tout le monde savait, que la naissance de la gestion des entreprises est une réponse à leurs besoins fonctionnels ; il faut encore savoir qu'elles sont en relation avec leurs usines, leurs agences, et leurs divisions avec les services qu'elles possèdent. Au préalable chacune de l'entreprise et sa division travaille séparément, est de même pour un service et l'autre de cette division, contrairement à aujourd'hui où le management a décidé de les connecter afin d'extraire et de corréler des informations portant sur toute l'entreprise et afin de générer un travail coopératif et collaboratif entre les utilisateurs distants. D'où la mise en place d'un réseau d'interconnexion pour un nombre important d'ordinateurs opérationnels qui sont souvent fort distants ; afin d'avoir un nouveau système d'information installé autour d'un réseau.

Généralement l'objectif dans ce cas est le partage des ressources, c'est-à-dire, rendre accessible à chaque membre du réseau les programmes, les données et les équipements indépendamment de leur localisation physique par rapport à l'utilisateur. Récemment les réseaux informatiques occupent une place prépondérante dans l'évolution technologique, les services qu'ils offrent font partie de la vie courante des entreprises et administrations (banques, gestion, commerce,...). Ainsi leur application est une obligation au domaine de la technologie de l'information et de la communication. Plus de détails, grâce au réseau il sera possible de mettre en place une véritable architecture de système d'information organisée autour de serveurs de taille adaptée à des besoins spécifiques et de clients distribués. C'est une des avancées technologiques qui a conduit à une redéfinition complète de l'architecture des moyens en passant d'un modèle centralisé autour de centraux puissants vers un modèle distribué où chaque service de l'entreprise gère le serveur dont il a la responsabilité. C'est dans ce sens qu'aujourd'hui nos développeurs et nos architectes logiciels se basent initialement sur les architectures réseaux pour développer leurs applications et leurs logiciels au niveau de tout type d'entreprises.

D'ailleurs, ces applications et ces logiciels doivent être réutilisables, extensibles, et performants avec une compétitivité accrue pour garantir leur qualité et évaluer les produits de services d'entreprises qui les demandent afin d'engendrer leur bonne marche. D'où la nécessité d'utilisation d'une bonne démarche et d'un bon patron pour concevoir, et développer ces applications.

Dans ce contexte, la direction régionale de distribution de gaz et d'électricité doit servir elle aussi à respecter les principes, les règles, et les techniques universelles de la profession pour s'adapter au nouvel environnement de la gestion mis en place, et évaluer la qualité de son système d'information (adéquation au changement, l'usage de l'information, aide aux processus etc.)

Au fur et à mesure que les évolutions technologiques ont le pouvoir de disposer les moyens d'actions efficaces, la mise en place d'un réseau dans l'entreprise a un effet structurant sur son organisation ; D'où la nécessité d'adaptation de l'entreprise à ces évolutions est obligatoire au domaine de la restructuration pour modifier son organisation et reconfigurer ses processus afin d'être rentable.

Dans le cadre de partage et d'échange d'information, de collaboration, de coordination, de communication, et de prise de décision les utilisateurs et les membres d'entreprise doivent amener à se réunir grâce à un réseau et la reprise d'une nouvelle génération de système d'information.

Au terme de productivité, réassemblage de solutions, et méthode de résolution de problème qui s'est déjà posé par le passé ; le suivi d'un patron lors de conception et de développement des applications d'entreprises est obligatoire.

La réponse à cette problématique est engendré grâce à notre projet qui été proposé par la division de gestion des systèmes informatiques de la direction régionale de distribution de gaz et d'électricité de Tizi Ouzou, et qu'on va élaborer avec le langage de modélisation **UML** et selon le plan suivant :

**Chapitre I** : Réseaux et client serveur.

**Chapitre II** : Traitement coopératif

**Chapitre III** : Généralités sur UML


**Chapitre IV** : Les patrons de conception.

**Chapitre V** : Présentation de l'organisme d'accueil.

**Chapitre VI** : Etude de l'existant.

**Chapitre VII** : Analyse et conception.

**Chapitre VIII** : Réalisation.



# CHAPITRE I :

## Réseaux et

## Architecture

## Client/Serveur

## Partie 1 : Les réseaux

### I. 1.1. Introduction :

Les applications et les systèmes distribués font de plus en plus partie intégrante du paysage d'un grand nombre d'entreprises. Ces technologies ont pu se développer grâce aux performances toujours plus importantes des réseaux locaux. En effet si les applications distribuées deviennent le principal outil du système d'information de l'entreprise alors une tendance récente des logiciels consiste à répartir les tâches entre plusieurs personnes via un réseau de communication.

### I. 1.2 : Définition des réseaux informatique: [Tanenbaum95]

Un réseau informatique est un ensemble interconnecté d'ordinateurs autonomes permettant d'échanger des informations sous forme de données numériques (valeurs binaires, c'est-à-dire codées sous forme de signaux pouvant prendre deux valeurs : 0 et 1). Cette connexion ne nécessite pas forcément un câble en cuivre (une ligne physique), mais elle peut être réalisée par laser, par ondes courtes ou par satellite.

L'autonomie des ordinateurs exclut les relations de type maître/esclave. Si un calculateur peut forcer le démarrage (ou l'arrêt) d'un autre ou le commander, ces calculateurs ne sont pas autonomes. Un système composé d'une unité de contrôle et de plusieurs unités esclaves n'est pas un réseau ; il en est de même pour un grand ordinateur ayant des lecteurs de carte, des imprimantes et des terminaux distants.

La plus grande confusion règne dans la littérature entre réseau et systèmes distribué. La distinction fondamentale réside dans le fait que, dans un système distribué, la répartition en ordinateurs autonomes est transparente, invisible à l'utilisateur. On tape une commande de lancement de programme et ça marche. C'est la tâche du système d'exploitation de sélectionner le bon processeur, trouver et transporter les fichiers d'entrée vers ce processeur et déposer les résultats au bon endroit. En d'autres termes, l'utilisateur d'un système distribué n'a pas connaissance de la multiplicité des processeurs ; il lui apparaît tel un système à un seul processeur virtuel unique.

Dans le cas d'un réseau, un utilisateur doit se signer explicitement sur une machine, soumettre une tâche à distance explicitement, faire des déplacements de fichiers explicitement et, de façon générale, gérer tout le réseau lui-même. Avec un système distribué, rien ne doit être fait de façon explicite ; tout est réalisé automatiquement par le système sans que l'utilisateur le sache.

En faite, un système distribué est un cas particulier de réseau dont le logiciel fournit un niveau élevé de cohérence et de transparence. Ainsi, la distinction entre système distribué et réseau réside dans le logiciel plutôt que dans le matériel. Cependant les deux systèmes ont de nombreux points communs. Ainsi tous les deux ont besoin de déplacer des fichiers. La différence réside dans la source du déplacement, le système ou l'utilisateur.



### **I. 1.3 : Les domaines d'application des réseaux:**

#### **1. L'accès à un programme distant : [Tanenbaum95]**

Une société ayant réalisé un modèle simulant l'économie mondiale doit permettre à ses clients de se connecter au réseau et de lancer le programme pour voir comment différents taux d'inflation projetés, taux d'intérêt, et fluctuations des monnaies pourraient affecter leur activité. Cette approche est souvent préférable à la vente des droits d'usage du programme, particulièrement dans le cas où le modèle est perpétuellement ajusté ou requiert l'exécution sur un très gros site ordinateur.

#### **2. L'accès à des bases de données distantes : [S.Ghernaouti 94]**

L'exécution de n'importe quelle application aura, peut être besoin de données sauvegardées et gérées par un système de gestion de base de données (SGBD) distant. Encore cette application devra entrer en communication avec le système informatique sur lequel est implantée la base de données et exprimer sa requête de consultation par l'activation des protocoles d'application appropriés.

#### **3. Communication :**

Quand plusieurs systèmes sont connectés entre eux par un réseau de communication, les utilisateurs des divers sites ont la possibilité d'échanger des informations. A bas niveau on transfère les messages entre les systèmes de la même façon que dans le système de message d'un ordinateur unique. Ainsi toutes les fonctionnalités de niveau supérieur retrouvées dans les systèmes autonomes peuvent être étendues afin d'envelopper le système réparti. Ces fonctions incluent le transfert de fichiers, la connexion, la messagerie et les appels de procédures à distance (RPC).

L'avantage d'un système réparti est que ces fonctions peuvent s'exécuter sur de grandes distances. Par exemple deux personnes peuvent effectuer un projet dans des sites géographiquement séparés. En transférant des fichiers du projet, en se connectant aux systèmes éloignés de chacun d'entre eux pour exécuter des programmes et en échangeant du Courrier pour coordonner le travail.

### **I. 1.4 : Les objectifs des réseaux :**

Il existe quatre objectifs principaux des réseaux d'ordinateurs à lesquels les organisations et les entreprises s'intéressent ; dans ce paragraphe nous allons développer chacun d'entre eux.

#### **1. Partage de ressources :**

Si un certain nombre de sites différents (avec des capacités différentes) sont connectés entre eux, un utilisateur dans un site peut donc utiliser les ressources disponibles dans un autre site. En général, le partage de ressources dans un réseau fournit des mécanismes non seulement pour partager des fichiers dans des sites éloignés, traiter l'information dans une base de données réparties, imprimer des fichiers dans des sites éloignés en employant des dispositifs matériels spécialisés, mais également pour effectuer d'autres opérations comme le partage de charge qui est une autre facette du partage des ressources.

## 2. Fiabilité :

Si un site tombe en panne dans un réseau, les sites restants peuvent continuer à fonctionner. Si le système est composé d'un certain nombre de grandes installations autonomes (c'est-à-dire des ordinateurs à usage général), la défaillance de l'un d'entre eux ne devrait pas affecter le reste ; Si en revanche, le système est constitué d'un certain nombre de petits ordinateurs, chacun d'entre eux est responsable d'une fonction cruciale du système (comme les entrées/sorties à caractères des terminaux ou le système de fichiers), une défaillance unique peut arrêter le fonctionnement de tout le système. En général, s'il existe suffisamment de redondance dans le système (autant dans les données que dans le matériel), le système peut poursuivre son opération, même si certains sites sont tombés en panne.

La défaillance d'un site doit être détectée par le système et l'on peut avoir besoin d'une action appropriée de reprise après défaillance. Le système ne doit plus utiliser les services de ce site. De plus si la fonction du site en panne peut être prise en charge par un autre site, le système doit assurer que le transfert de la fonction s'effectue correctement.

Enfin quand le site en panne reprend ou est réparé, on doit disposer de mécanismes permettant de le réintégrer doucement dans le système.

## 3. Réduction de cout:

D'après [Tanenbaum95] Les petits ordinateurs ont un bien meilleur ratio prix/performance que les gros. Les grandes installations sont environ dix fois plus rapides que les microprocesseurs à une seule puce, mais elles coûtent mille fois plus cher. Cette évaluation comparée a amené beaucoup de concepteurs de systèmes à construire des systèmes à base de micros personnels puissants, un par utilisateur, les données étant partagées sur un ou plusieurs serveurs de fichiers.

## 4. Accélération du calcul et augmentation de performances :

La possibilité d'augmenter graduellement les performances du système par adjonction de processeurs lorsque la charge de travail croît est en relation étroite avec un tel principe.

Si un calcul particulier peut être découpé en un certain nombre de sous-calculs pouvant s'exécuter en concurrence, la disponibilité d'un système réparti peut nous permettre de répartir le calcul entre les divers sites, afin d'exécuter le calcul de façon concurrente. De plus, si un site particulier est surchargé de travaux, certains d'entre eux peuvent être déplacés vers d'autres sites légèrement chargés. Ce déplacement de travaux est appelé *répartition de la charge*.

D'ailleurs, le système d'information installé autour d'un réseau va permettre une circulation plus rapide de l'information de l'entreprise dont les effets auront des conséquences sur la gestion quotidienne. Cette circulation rapide en masse de l'information va conduire les acteurs à disposer de données conduisant à des réactions rapides de tous les niveaux hiérarchiques face aux événements de la gestion quotidienne. Donc, c'est une technologie qui, si elle est bien organisée, va générer des gains de productivité et améliorer le service rendu. C'est la raison pour laquelle la décision de mettre en place un réseau local doit être considérée comme une décision d'ordre stratégique.

## I. 1.5 : Typage et classification des réseaux informatique:

# Chapitre I : Réseaux et architecture client serveur

---

Les réseaux informatiques sont généralement classifiés suivant le type de liaisons utilisées et la surface géographique qu'ils couvrent (distance).

## I. 1.5.1. Réseaux de type filaire :

C'est un ensemble d'hôtes (ordinateurs par exemple) reliés soit directement par des liaisons filaires soit via un sous-réseau de communication (ou réseau de transport). On trouve : **les réseaux locaux, et Les réseaux métropolitains ou MAN.**

### A. les réseaux locaux :

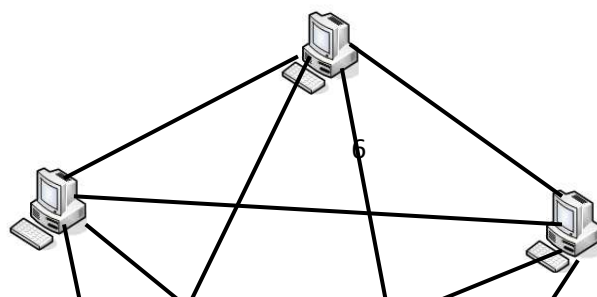
Les réseaux locaux apparaissent au début des années 1970, comme remplaçant des gros systèmes informatiques. Il est devenu évident que, pour plusieurs entreprises, il est plus économique de posséder un certain nombre de petits ordinateurs, chacun avec ses propres applications autonomes, plutôt qu'un seul gros système. Comme il est probable que chaque petit ordinateur ait besoin d'un complément complet de périphériques (comme disques et imprimantes) et qu'une certaine forme de partage de données se produise dans une entreprise, on a tout naturellement connecté les petits ordinateurs au réseau.

Les réseaux locaux sont habituellement conçus pour couvrir une petite zone géographique (comme un immeuble unique ou quelques immeubles adjacents) et généralement utilisés dans un environnement de bureau. Tous les sites des systèmes sont proches les uns des autres, les liaisons de communication ont donc tendance à avoir une vitesse supérieure et un taux d'erreur inférieur à leurs équivalents dans les réseaux à grande distance. Puisque l'on peut atteindre cette vitesse et cette fiabilité supérieures, des câbles de haute qualité (chers) sont nécessaires. Il est également possible d'employer le câble exclusivement pour la circulation du réseau informatique. Sur des distances plus longues, le coût de l'emploi de câbles de qualité supérieure est énorme et l'utilisation exclusive de câble tend à être prohibitive.

### ➤ Topologie des réseaux locaux :

#### a. Réseaux complètement connectés :

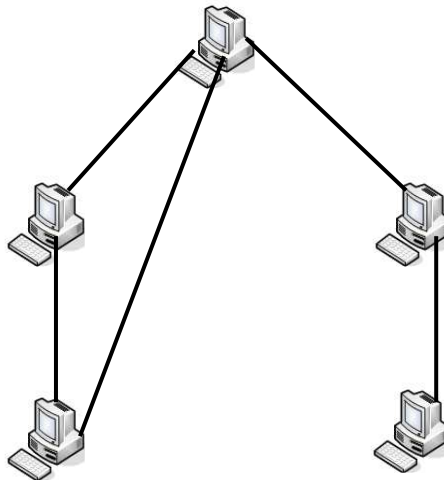
Dans un réseau complètement connecté, chaque site est directement connecté avec tous les autres sites du système. Cette configuration possède un haut coût de base, puisque une ligne de communication directe doit être disponible entre chaque paire de sites. Le coût de base augmente selon le carré du nombre de sites. Cependant, dans cet environnement, on peut rapidement envoyer les messages entre les sites ; un message doit utiliser seulement une liaison pour voyager entre deux sites. De plus ces systèmes sont fiables, car plusieurs liaisons devraient tomber en panne pour que le système soit subdivisé. Un système est subdivisé s'il a été découpé en deux (ou plus) sous-systèmes ne possédant aucune connexion entre eux.



### **FIG1 : RÉSEAU COMPLÈTEMENT CONNECTÉ**

#### **b. Réseaux partiellement connectés :**

Dans un réseau partiellement connecté, les liaisons directes existent entre certaines paires de sites, mais pas toutes. Le cout de base de cette configuration est donc inférieur à celui d'un réseau complètement connecté. Cependant on peut devoir envoyer un message d'un site à un autre en traversant plusieurs sites intermédiaires, provoquant ainsi une communication plus lente. De plus, un système partiellement connecté n'est pas aussi fiable qu'un réseau complètement connecté. La défaillance d'une liaison peut subdiviser le réseau



### **FIG2 : RÉSEAU PARTIELLEMENT CONNECTÉ**

#### **c. Réseaux hiérarchiques :**

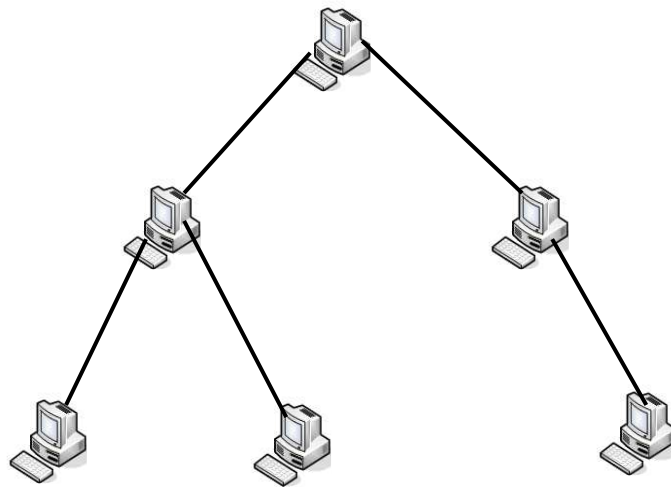
Dans un réseau hiérarchique, les sites sont organisés comme un arbre. Cette organisation est communément utilisée pour des réseaux d'entreprise. Des bureaux individuels sont connectés au bureau principal local. Les bureaux principaux sont connectés aux bureaux régionaux ; Les bureaux régionaux sont connectés à la maison-mère de l'entreprise.

Chaque site (sauf la racine) possède un seul père et un certain nombre de fils (peut être aucun). Le cout de base de cette configuration est généralement inférieur à celui du schéma partiellement connecté. Dans cet environnement, un père et un fils communiquent directement. Les nœuds ayant le même père peuvent communiquer entre eux seulement à travers leur père commun. Un message d'un de ces nœuds à l'autre doit être envoyé au père

## Chapitre I : Réseaux et architecture client serveur

pour qu'il le transmette au fils correspondant. De même, les cousins peuvent communiquer entre eux par l'intermédiaire de leur grand-père commun. Cette configuration correspond bien à la généralisation du fait que les systèmes proches entre eux communiquent plus que ceux qui sont éloignés. Par exemple, il est plus probable que les systèmes d'un immeuble transfèrent des données que ceux des installations séparées.

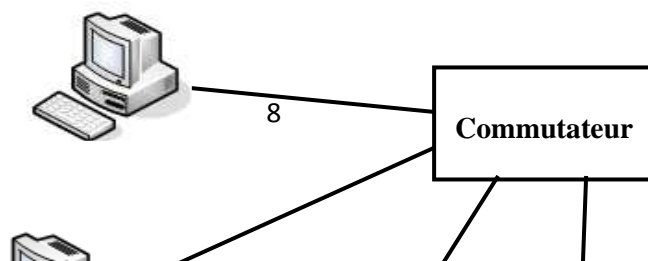
Si un site père tombe en panne, ses fils ne peuvent donc plus communiquer entre eux ou avec d'autres processeurs. En général la défaillance d'un nœud (sauf une feuille) subdivise le réseau en plusieurs sous-arbres disjoints.



**FIG3 : RÉSEAU À STRUCTURE ARBORESCENTE**

### d. Réseaux en étoile :

Dans un réseau en étoile, l'un des sites du système est connecté à tous les autres sites. Aucun des autres sites ne sont connectés entre eux. Le coût de base de ce système est linéaire selon le nombre de sites. Le coût de communication est également faible, car un message de site A vers B demande au moins deux transferts (de A au site centrale et ensuite du site centrale à B). Cependant ce schéma de transferts de messages nécessaires soit bas, le temps requis pour envoyer ces messages peut être long. Dans plusieurs systèmes en étoile, le site central se consacre donc complètement à la tâche de l'échange de messages, et si le site central tombe en panne, le réseau est complètement subdivisé. Dans cette topologie chaque ligne est raccordée à un concentrateur ou un commutateur dont le rôle sera de fournir un service de mise en relation d'une ligne à l'autre. En principe cette topologie est adaptée aux paires torsadées.

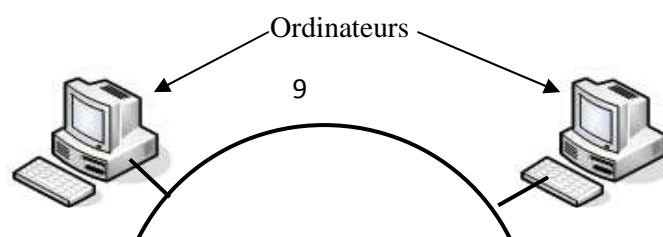


### FIG4 : RÉSEAUX EN ÉTOILE

#### e. Réseaux en anneau :

Dans un réseau en anneau, chaque site est physiquement connecté à exactement deux autres sites. L'anneau peut être unidirectionnel ou bidirectionnel. Dans une architecture unidirectionnelle, un site peut transmettre des informations à un seul de ses voisins. Tous les sites doivent envoyer de l'information dans la même direction. Dans une architecture bidirectionnelle, un site peut transmettre des informations à ses deux voisins. Le coût de base d'un anneau est linéaire en fonction du nombre de sites. Cependant, le coût de communication peut être élevé. Un message d'un site à un autre voyage dans l'anneau jusqu'à arriver à destination. Dans un anneau unidirectionnel, ce processus peut demander  $n-1$  transferts. Dans un anneau bidirectionnel, au plus  $n/2$  transferts sont nécessaires.

Dans un anneau bidirectionnel, deux liaisons devraient tomber en panne avant de subdiviser le réseau. Dans un anneau unidirectionnel, la défaillance d'un seul site (ou d'une liaison) subdiviserait le réseau. Pour résoudre ce problème, on peut enrichir l'architecture en fournissant des liaisons doubles.

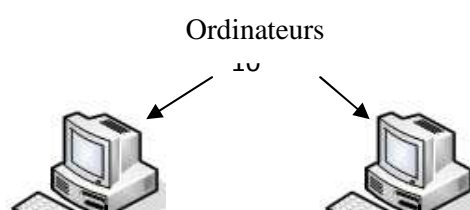


## FIG5 : RÉSEAUX EN ANNEAU

### f. Topologie en bus :

Un bus désigne une organisation matérielle se présentant comme un câble logique sur lequel se connectent les stations (ordinateurs). Son rôle est d'acheminer les informations d'un équipement à l'autre. Ces derniers sont identifiés par une adresse unique. L'avantage de cette topologie réside dans sa simplicité et son coût. L'inconvénient réside dans la façon dont sont traités les conflits d'accès et les questions de surcharge. En effet, à un instant donné, un seul message peut être présent sur le bus (donc une seule machine est autorisée à transmettre à un moment donné). Si deux trames sont présentes simultanément (deux machines ou plus émettent en même temps), elles génèrent une pollution de transmission –que l'on désigne sous le nom de collision– qui a pour effet de détruire les deux trames. Pour pallier ces difficultés, deux familles de solutions ont été réalisées :

- Les solutions basées sur les méthodes d'accès au bus. Le principe consiste à s'assurer, avant l'émission d'une trame, que le bus est libre ;
- Les autres solutions consistent à organiser le réseau local sous la forme de segments indépendants qui communiquent entre eux par des ponts. La propagation des trames se limite à un segment du réseau et fait donc décroître la probabilité de créer une collision (Ethernet et Token-Bus sont conçus sur cette topologie).





## FIG6 : RÉSEAUX EN BUS

### B. Les réseaux métropolitains ou MAN : selon [Pujolle03]

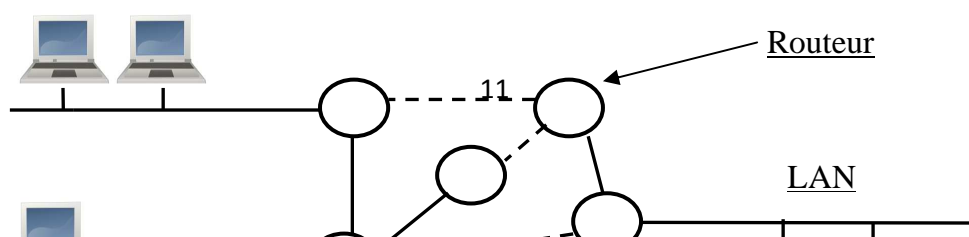
Un MAN (Metropolitan Area Network), est un système de communication offrant des services de transfert très divers comme les données à haut débit, la voix ou la vidéo, entre des équipements situés dans un environnement géographique étendu, dont la dimension est celle d'une métropole. Ces réseaux peuvent également relier un certain nombre de bâtiments se trouvant sur un même campus. Il s'agit dans ce cas d'un réseau métropolitain privé.

Les réseaux métropolitains permettent l'interconnexion de réseaux locaux situés à moins d'une centaine de kilomètres les uns des autres. La vitesse de transport doit atteindre un minimum de 100 Mbits/s pour permettre ce type d'interconnexion. Ces réseaux métropolitains sont plus ou moins adaptés au multimédia, certains d'entre eux ne prenant en charge que les données, d'autres tenant d'intégrer voix et données.

### C. Les réseaux étendus ou WAN :

Les WANs (Wide Area Network), appelé aussi réseaux longues distances, selon [Pujolle03] sont destinés, comme leur nom l'indique, à transporter des données numériques sur des distances à l'échelle d'un pays, voire d'un continent ou de plusieurs continents

Interconnecte plusieurs LANs à travers de grandes distances géographiques. Ce type de réseaux s'étend sur une vaste surface (pays, continent). Il englobe un grand ensemble d'ordinateurs (Hôtes) appartenant à des clients. Ces hôtes sont reliés par un sous-réseau de communication (réseau de transport) propriété d'un fournisseur d'accès à Internet, qui en assure le fonctionnement



### **FIG7 : RÉSEAU LONGUE DISTANCE.**

#### **I. 1.5.2. Réseaux sans fils :**

Ce sont des réseaux filaires dans lesquels au moins une liaison filaire (câbles, fibres optiques) est remplacée par une liaison radio, permettant ainsi la mobilité de l'ordinateur concerné par cette liaison. Du fait de la mobilité des ordinateurs, les réseaux sans fils sont aussi appelés réseaux mobiles. On trouve :

##### **I. .1.5.2.1 Le réseau personnels ou Bluetooth :**

Appelé aussi WPAN (Wireless Personal Area Network). C'est généralement un réseau d'interconnexion des différents composants d'un ordinateur à l'aide d'une liaison radio très courte distance. Ce système permet donc de connecter un ordinateur et ses périphériques (clavier, souris, imprimante,...) en les plaçant dans la zone de couverture radio du système. Ce réseau utilise le modèle Maître-Esclave. C'est l'ordinateur (Maître) qui gère la communication avec les différents périphériques (Esclaves).

##### **I. .1.5.2.2 Les réseaux locaux sans fils (WLAN) :**

Généralement appelés Wireless LAN. Dans ce cas chaque ordinateur est doté d'une antenne radio grâce à laquelle il peut communiquer avec les autres ordinateurs via une antenne commune appelée *Station de base* leur donnant ainsi accès au réseau filaire.

Toutefois, si les ordinateurs sont très proches l'un de l'autre, ils peuvent communiquer entre eux en mode peer to peer (poste à poste).

L'architecture la plus connue dans ce type de réseaux est le WIFI.

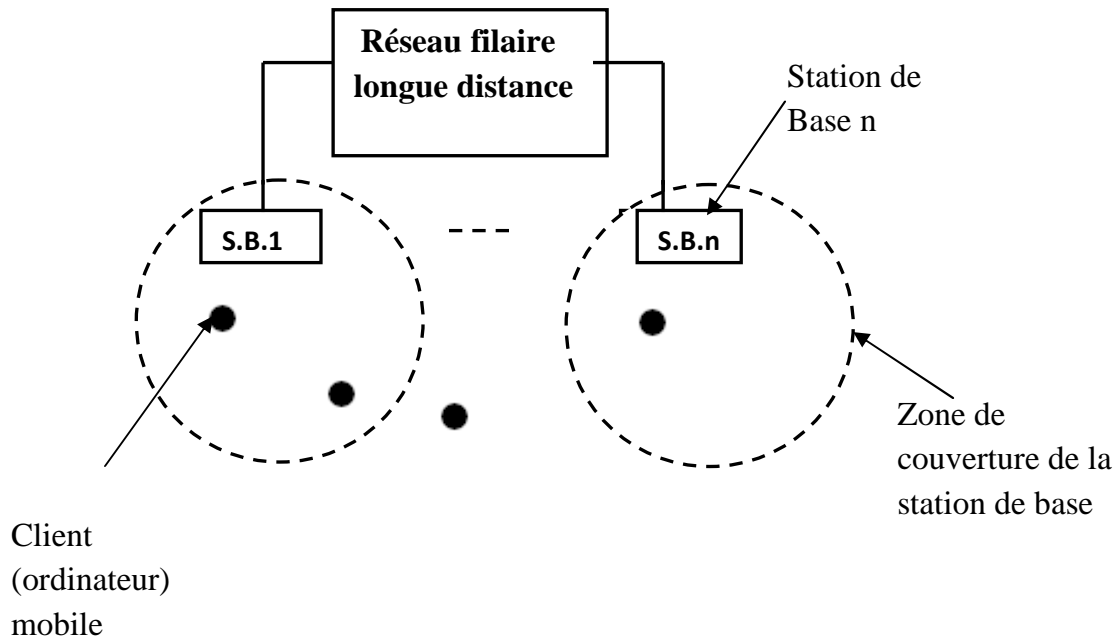
##### **I. 1.5.2.3 Les réseaux métropolitains sans fils (WMAN) :**

Appelés aussi Wireless MAN ou Boucle Local Radio (BLR). La technologie la plus connue est le WIMAX qui offre des débits de l'ordre 70 Mbps pour une portée de plusieurs kilomètres.

##### **I. .1.5.2.4 Les réseaux longue distance (WWAN) :**

Appelés aussi Wireless WAN) ou le réseau cellulaire mobile. Il est généralement constitué d'un réseau de transport longue distance filaire auquel accèdent les clients

(ordinateurs ou téléphones mobiles) via des antennes stations de base fixes (Figure 8). De plus les clients peuvent être mobiles. En d'autre terme, durant une session de travail, un client peut sortir de la zone de couverture de l'antenne d'une station de base puis passer à la zone de couverture d'une autre station de base.



**FIGURE 8: RÉSEAU LONGUE DISTANCE**

### **I. 1.6: Architecture des réseaux et stratégies de conception :**

Lors de la conception d'un réseau de communication il faut affronter la complexité inhérente à la coordination des opérations asynchrones communiquant dans un environnement potentiellement lent et sujet aux erreurs. Il est également nécessaire que les systèmes du réseau soient d'accord sur un protocole ou un ensemble de protocoles pour déterminer les noms des machines, localiser les machines dans le réseau, établir des connexions. Le problème peut être simplifié par le découpage de système en plusieurs couches. Chaque couche d'un système communique avec la couche équivalente dans les autres systèmes.

Chaque couche peut posséder ses propres protocoles ou peut être une segmentation logique. On peut implémenter les protocoles dans le matériel que dans le logiciel.

En accord avec l'International Standards Organisation (ISO), nous présentons les couches par les descriptions suivantes :

1. **La couche physique :** La couche physique est responsable de la manipulation des détails mécaniques et électriques de la transmission physique d'un flot de bits. Dans cette couche, les systèmes de communication doivent être d'accord sur la représentation électrique des binaires 0 et 1 afin que les données soient envoyées comme un flot de signaux électriques, alors le récepteur soit capable d'interpréter les données correctement comme les données binaires. La couche physique est implémentée dans le matériel du périphérique de gestion de réseau.
2. **Couche liaison de données :** cette couche est responsable de la manipulation des *trames* ou des parties de longueur fixe des paquets y compris toute détection et reprise en cas d'erreur qui se serait produite dans la couche physique.
3. **Couche de réseau :** La couche de réseau a la responsabilité de fournir des connexions et d'effectuer le routage des paquets dans le réseau de communication, y compris la manipulation des adresses des paquets sortant, le décodage des adresses des paquets entrant et le maintien de l'information de routage pour obtenir une réponse correcte quand on change le niveau de charge. Encore les routeurs travaillent au niveau de cette couche.
4. **Couche de transport :** la couche de transport est responsable des accès de bas niveau au réseau et du transfert des messages entre clients, y compris le découpage des messages en paquets, le maintien de l'ordre des paquets, le contrôle du flux et la génération des adresses physiques.
5. **Couche de session :** elle est responsable de l'implémentation des sessions ou des protocoles de communication de processus à processus.  
En général, ces protocoles sont les communications réelles pour les connexions à distance et pour le transfert de Fichiers et du courrier.
6. **Couche de présentation :** Responsable de la résolution des différences de format entre les divers sites d'un réseau, y compris la conversion de Caractères et les modes semi duplex, duplex (écho caractère).
7. **Couche applicative :** responsable de l'interaction directe avec les utilisateurs. La couche applicative traite le transfert de fichiers, les protocoles de connexions à distance et le courrier électronique ainsi que les schémas pour les bases de données répartis.

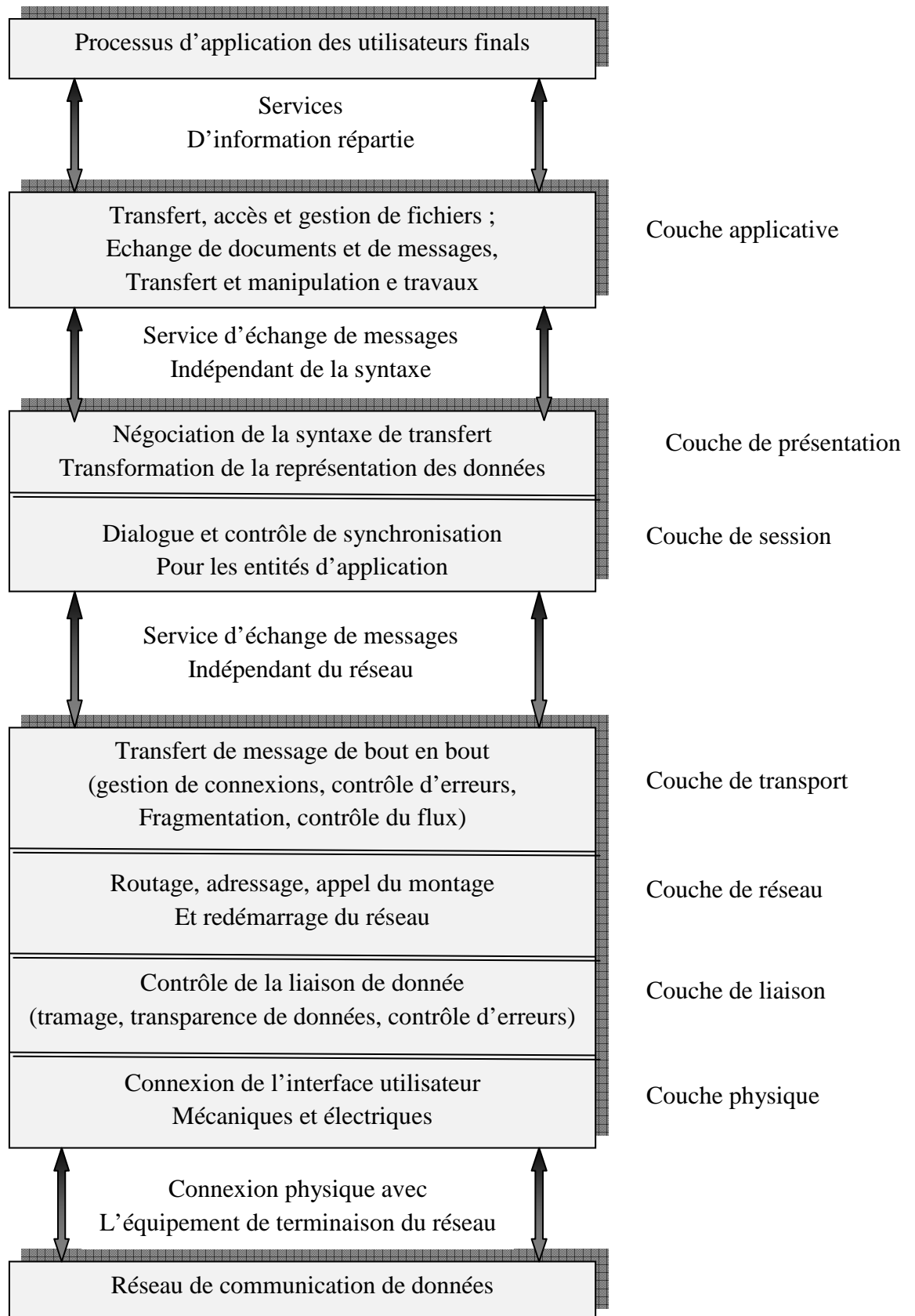
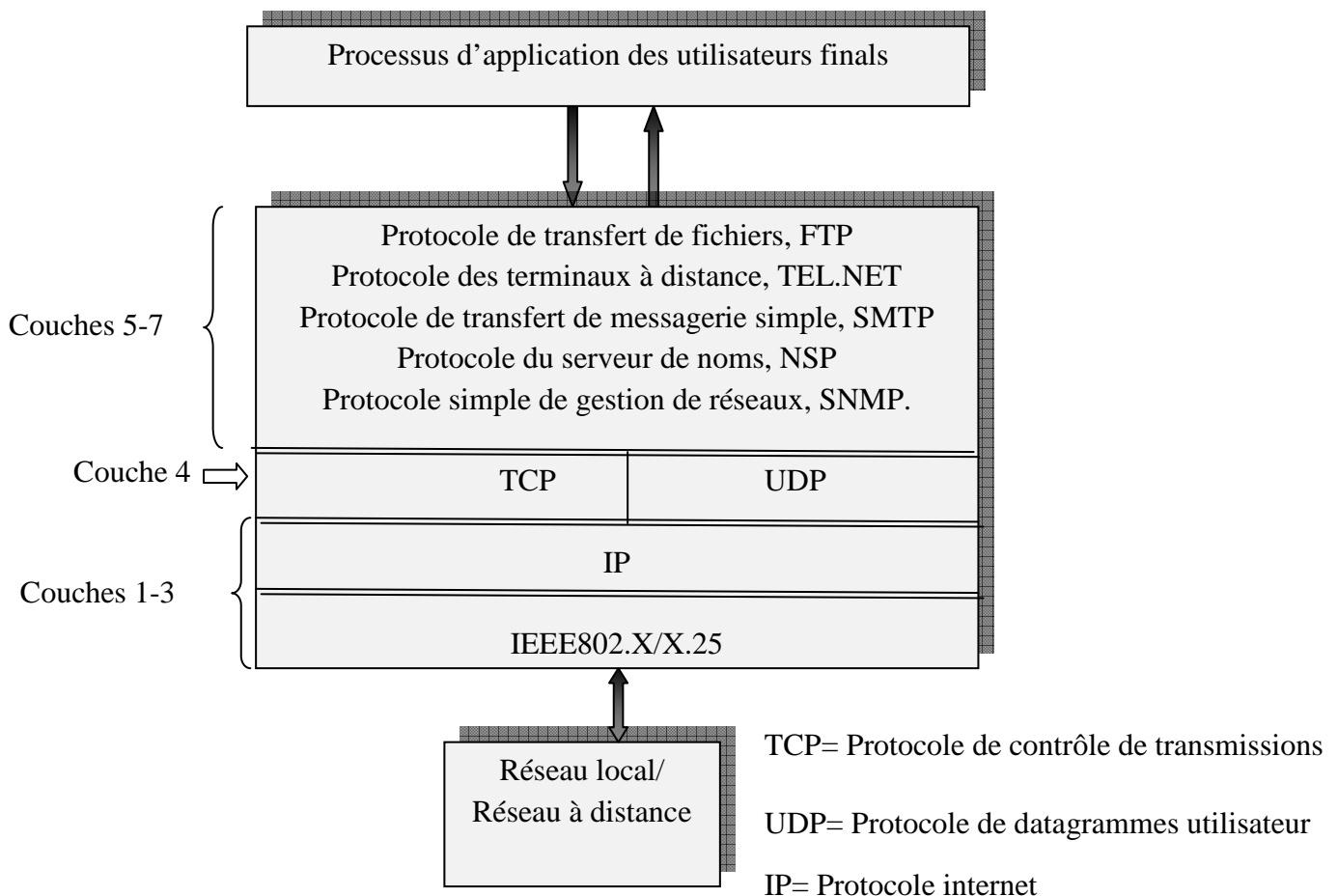


FIGURE 9 : RÉSUMÉ DES COUCHES DE PROTOCOLES ISO

## Chapitre I : Réseaux et architecture client serveur

La plus part des sites internet communiquent encore par l'intermédiaire du *Transmission Contrôle Protocole / Internet Protocole*, connu par TCP/ IP. La pile de protocole TCP/IP possède moins de couches que le modèle ISO, comme il combine plusieurs fonction dans chaque couche, il est plus difficile à implémenter mais efficace que la gestion de réseaux ISO.

Dans un réseau TCP/IP, chaque machine possède un nom et un numéro internet à 32 bits associé (pour l'identification de machine). Tous ces deux derniers doivent être uniques et pour que l'espace de noms puisse être géré, ils sont segmentés. Le nom décrit le nom de la machine et ensuite les organisations avec lesquelles la machine est associée. L'identification de machine est découpée en un numéro de réseau et un numéro de machine. La proportion de la séparation varie selon la taille du réseau. Une fois que les administrateurs de l'internet ont affecté un numéro de réseau, le site ayant ce numéro est libre d'affecter des identifications de machines.



**FIGURE 10 : RÉSUMÉ DES COUCHES DU PROCESSUS TCP/IP**

## 2<sup>ème</sup> partie : l'architecture client serveur

### **I. 2.1. L'apparition du modèle client serveur :**

Le concept du client-serveur est apparu dans les années 1990 pour pouvoir allier les avantages de l'informatique centralisée (puissance et sécurité) et de l'informatique individuelle (convivialité).

Le modèle client-serveur s'oppose aux vieux systèmes informatiques centralisés autour d'un gros ordinateur, basés sur une architecture propriétaire. Le système central traite la requête d'un terminal, puis lui envoie la réponse. Le modèle client-serveur s'oppose également au mode autonome de travail des réseaux PEER to PEER. Dans une organisation de type égal à égal (ou poste à poste), les fichiers sont répartis sur les disques durs de tous les ordinateurs (c'est la pagaille décentralisée) et tous les utilisateurs «voient » les autres ordinateurs qui sont connectés en même temps, c'est ainsi qu'ils peuvent savoir si une ressource localisée sur telle machine est accessible ou non. Par contre, dans une organisation de type client-serveur, les fichiers sont généralement centralisés sur un serveur, et les utilisateurs « voient » le serveur mais ne « voient » pas les autres machines utilisateur, tout passe par l'intermédiaire du serveur. Dans un réseau de type client-serveur, les ordinateurs ne devront jamais avoir besoin des ressources d'une autre station, parce qu'ils ne pourront simplement pas y accéder. La totalité de l'architecture du réseau repose sur un ou plusieurs serveurs dédiés.

Le modèle client-serveur est apparu grâce aux progrès technologiques qui ont permis de transformer les terminaux dépourvus d'intelligence, en de véritables ordinateurs avec une véritable capacité de traitement, de stockage, de présentation et de communication... Pour autant, quand un client fait appel à un serveur de base de données, par exemple, le serveur ne transmet pas toute la base de données au client (même si cela est envisageable, cette solution encombrerait beaucoup le réseau). En réalité, la situation la plus courante est que le client et le serveur se partagent le travail.

### **I. 2.2. Environnement client/ serveur :**

#### **I. 2.2.1. Présentation du modèle client/ serveur :**

De nombreuses applications fonctionnent selon le modèle client/serveur ; cela signifie que des **machines clientes** (les machines faisant partie de réseau) contactent un **serveur**, une machine généralement très puissante en termes de capacités d'entrées-sortie, qui leur fournit des **services**. Ces services sont des programmes fournissant des données.

Les services sont exploités par des programmes, **appelés programmes clients**, s'exécutants sur les machines clientes.

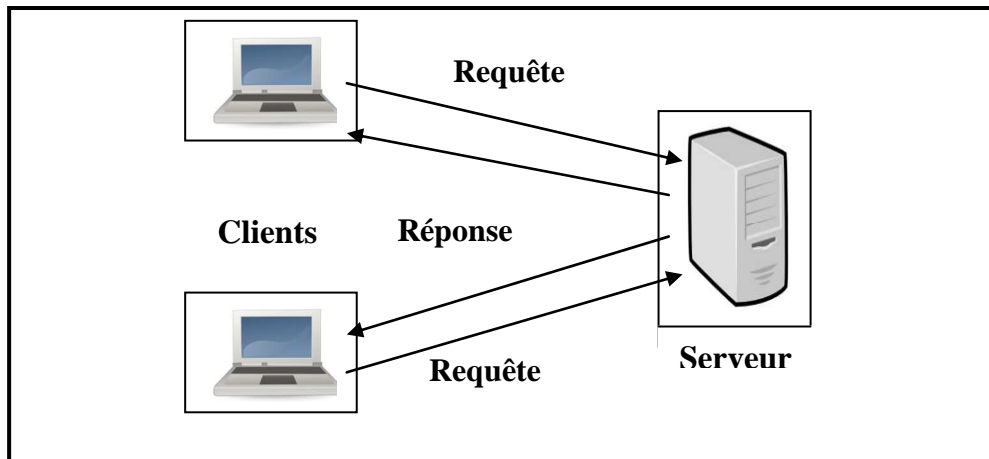
Le client / serveur est une architecture de réseau dans laquelle toutes les informations sont localisées sur des ordinateurs ayant le rôle de serveur et accessible aux ordinateurs clients. Une application est développée selon cette architecture lorsqu'elle est composée de deux programmes coopérant l'un avec l'autre à la réalisation d'un même traitement, l'un tient le



## Chapitre I : Réseaux et architecture client serveur

rôle de serveur conçu de façon à pouvoir répondre à des requêtes provenant des programmes clients qui attendent des réponses.

Dans le modèle client-serveur, les deux acteurs sont le client et le serveur. Les machines clientes (en général des micro-ordinateurs) se chargent des fonctions d'affichage et de traitement "logique" des données. Les serveurs sont dédiés aux fonctions de stockage et de gestion des données. L'architecture Client/serveur repose sur l'utilisation des liaisons réseau pour distribuer les traitements sur ces divers systèmes. Le dialogue entre les clients et les serveurs se fait sous forme de requêtes.



Les architectures Client/serveur cherchent à répondre au problème suivant :

- où héberger les données ?
- où héberger les applications qui gèrent les données ?
- où héberger la logique applicative, c'est à dire les applications chargées de la gestion du dialogue avec l'utilisateur (exemple : les formulaires de saisie, les états, etc.) ?

### I. 2.2.2. Structure du mode client/ serveur : [S.Gheraouti 94]

Le modèle client /Serveur permet de supporter les applications coopérantes. Il rend transparent et fait abstraction des diverses techniques de mise en œuvre du transport des questions/réponses (RPC, *Remote procedure Call*, transfert de messages, mode conversationnels, etc.). Il permet également d'intégrer  $n$  application sur une même plate forme.

L'approche Client/serveur repose sur trois types d'acteurs : le « *client* », le « *serveur* », et le « *réseau de transport* ». Le client réalise la manipulation, la saisie et l'affichage des données. Le serveur effectue le stockage, l'accès, qui sera possible sécurisé, aux ressources demandées, qu'elles soient matérielles ou logicielles, passives ou actives. Le réseau effectue le transport des requêtes du client vers le ou les serveurs et en rapporte les résultats d'exécution.

### I. 2.2.3. : Avantages du modèle client/ serveur : [S.Gheraouti 94]

Le modèle client /serveur est particulièrement recommandé pour des raisons nécessitant un grand niveau de fiabilité, ses avantages sont :

- **Le réseau évolutif:**

L'architecture réseau la plus répandue.

Une plate-forme relativement bon marché.

Une capacité de traitement comparable aux grands systèmes.

Le traitement d'un grand nombre de requêtes simultanées.

La réduction du trafic réseau.

La répartition des tâches entre les clients et les serveurs exploite plus efficacement les ressources du réseau dans son ensemble.

Le stockage des données sur une ou quelques machines améliore leur sécurité.

L'ajout et suppression des clients sans perturber le fonctionnement de réseau et sans modification majeure.

- **L'administration au niveau serveur :**

Une configuration personnalisable.

L'installation d'un grand nombre d'applications partagées, notamment les serveurs de base de données.

La centralisation de l'administration, de la stratégie de sécurité, de la maintenance, des sauvegardes, ...La centralisation des fichiers.

L'utilisation des serveurs d'applications qui réduisent le coût des licences d'exploitation (à partir de plus 20 stations utilisant la même application, il n'est plus rentable d'avoir 20 licences, et 20 fois le même logiciel...).

Les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.

- **Le matériel :**

L'économie en mémoire RAM et ROM sur les ordinateurs clients.

La concentration de la puissance sur les serveurs.

- **Une meilleure sécurité :** Car le nombre de points d'entrée permettant l'accès aux données est moins important.

- **Des ressources centralisées :** étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction.

## I. 2.2.4. : Inconvénients du modèle client/ serveur :

L'architecture client/serveur a de même quelques lacunes parmi lesquelles :

- **Un cout élevé :** dû à la technicité du serveur.
- **Un maillon faible :** Le serveur est le seul maillon faible de réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui ! Heureusement, le serveur a une grande tolérance aux pannes.

### **I. 2.2.5. : Le processus de fonctionnement d'une requête SQL en client/serveur :**

L'application la plus utilisée en mode client-serveur est la base de données. Les bases de données permettent d'organiser fonctionnellement un très grand nombre d'informations, et de les trier au fur et à mesure des besoins. Le modèle client-serveur permet de centraliser les informations de la base de données et de répondre à un grand nombre de requêtes simultanées de la part des clients. Le langage pour exprimer une requête auprès de la plupart des bases de données est le SQL (Structured Query Language). Le langage SQL est un « langage d'interrogation structuré » qui a été conçu par la société IBM. SQL est devenu une norme, un standard dans le monde des bases de données. Le processus d'une requête SQL en client-serveur :

- L'utilisateur émet une demande
- La commande est traduite en SQL
- Le redirecteur intercepte la requête SQL et l'envoie à la carte réseau qui la transmet au support de communication
- La requête circule sur le réseau jusqu'au serveur de base de données
- Le serveur de base de données accepte la requête, la traite (recherche, extrait et trie les informations contenues dans les tables de la base de données) et envoie une réponse (un enregistrement SQL)
- La réponse circule sur le réseau
- Le client reçoit la réponse du serveur
- L'utilisateur visualise la réponse

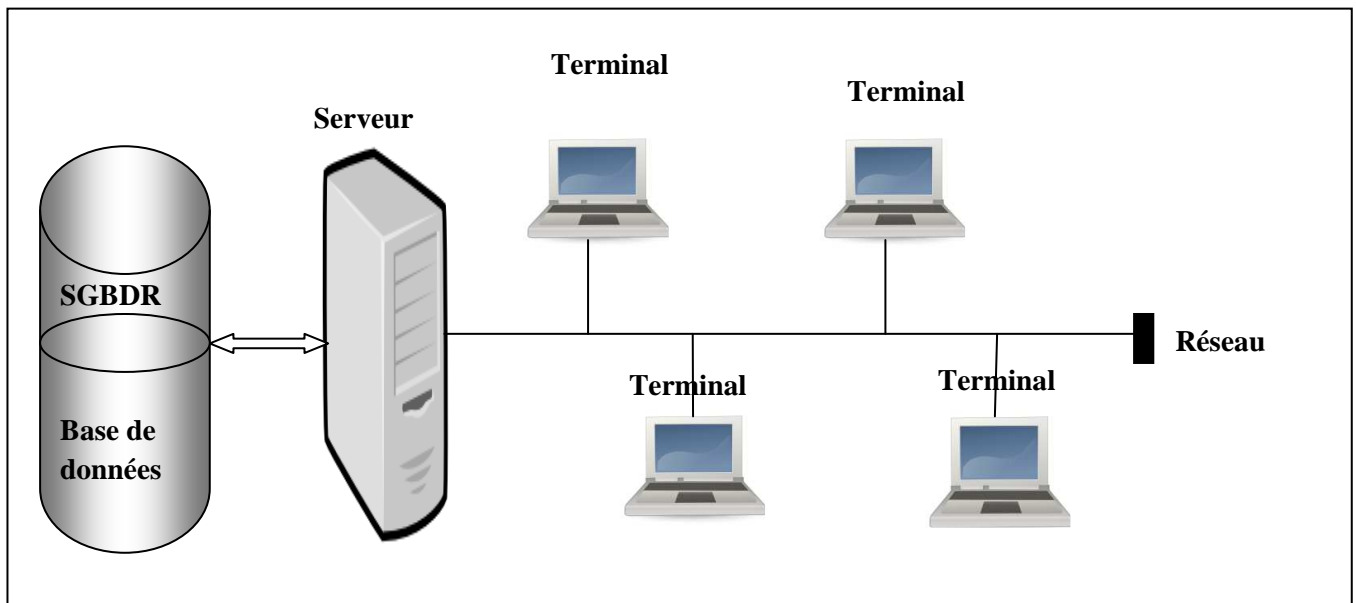
L'application de base de données MICROSOFT ACCESS possède son propre langage SQL, mais son interface utilisateur permet d'interroger la plupart des autres bases de données

### **I. 2.2.6. : Le paradigme client/ serveur :**

La majorité des réseaux informatiques se fondent sur le paradigme client / serveur. Le qualificatif client / serveur est souvent employé dans deux contextes ; physique et logique. Dans le premier contexte, clients et serveurs désignent des systèmes physiques (machine), tandis que dans le second, ils relèvent du domaine de logiciel et données, donc ils sont assimilés à deux processus constituant une application client / serveur. Ces deux processus peuvent être implantés sur une même machine ou sur différentes machines.

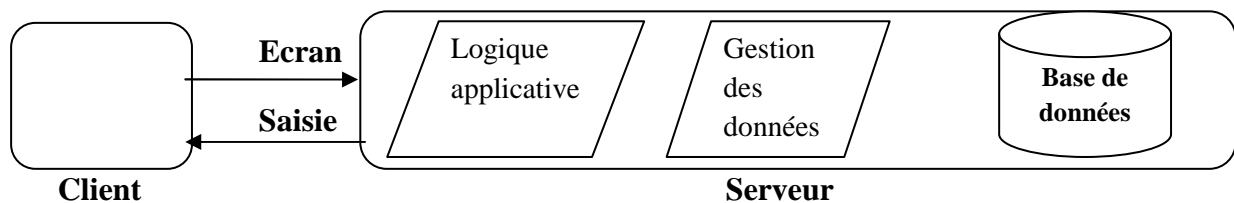
### I. 2.3 : Classification des architectures clients/serveur :

#### I. 2.3.1. L'architecture centralisée :



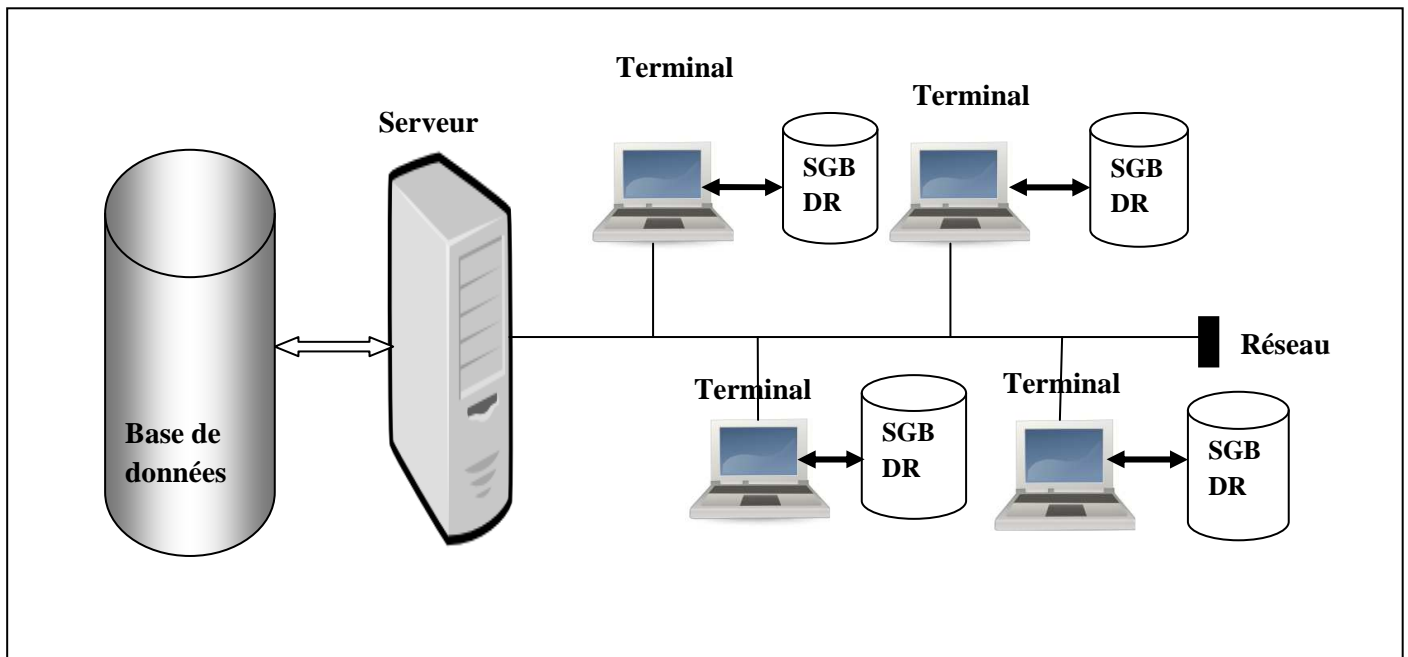
**FIGURE 11 : L'ARCHITECTURE CLIENT/SERVEUR CENTRALISÉE**

Le serveur héberge la base de données, il effectue la totalité des traitements et assure, au travers d'interfaces, le dialogue avec les clients (terminaux ou Emulation terminal en réseau).



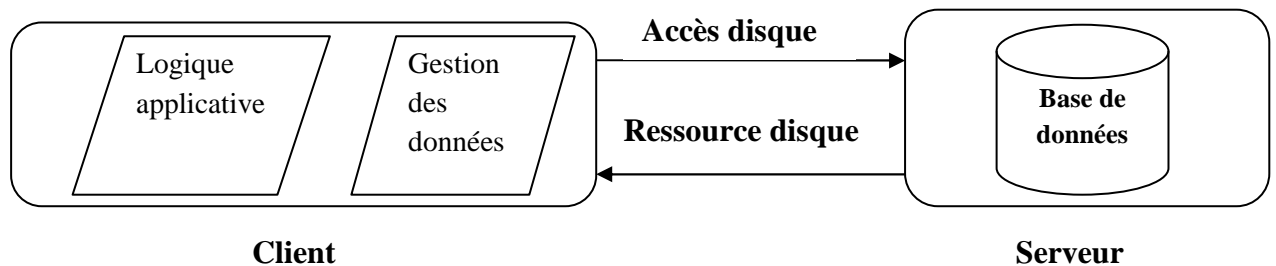
Avantages	Inconvénients
<ul style="list-style-type: none"><li>- Sécurité très élevée</li><li>- Fiabilité très élevée</li><li>- Hébergement de bases de données de très grande taille (milliers de Go)</li><li>- Gestion centralisée performante</li></ul>	<ul style="list-style-type: none"><li>- Serveur et système d'exploitation propriétaires (Ibm, Bull...), d'où une forte dépendance au fournisseur et un coût du système élevé.</li><li>- Coût de développement des applications élevé.</li><li>- Interface utilisateur sans souplesse.</li><li>- Le terminal est un poste de travail sans intelligence dont l'usage est limité aux applications du serveur auquel il est relié.</li></ul>

### I. 2.3.2. L'architecture à serveur de fichiers :



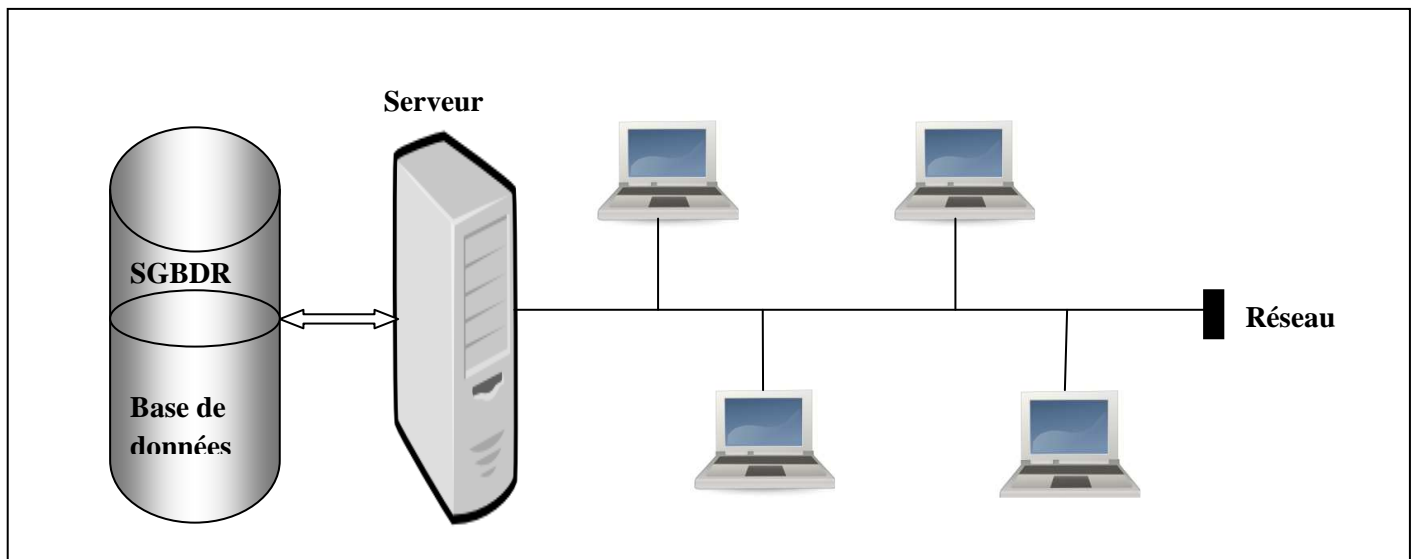
**FIGURE 12 : L'ARCHITECTURE À SERVEUR FICHIER**

Le serveur de fichiers héberge l'ensemble des données, mais n'effectue aucun traitement. C'est une fonction du client, tout comme la logique applicative.



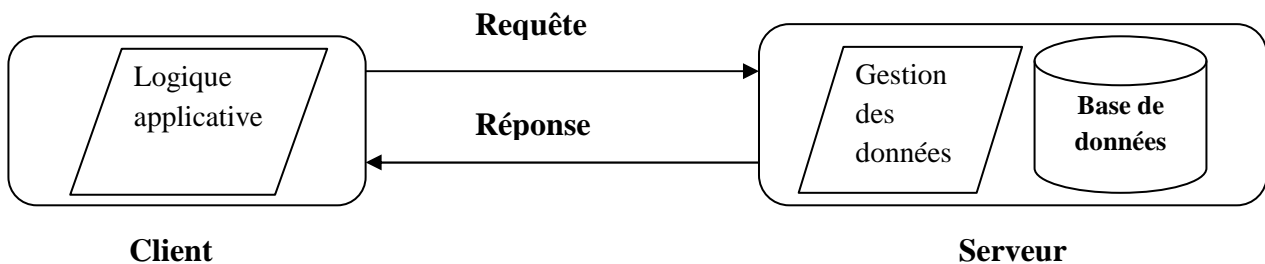
Avantages	Inconvénients
<ul style="list-style-type: none"><li>- Coût faible.</li><li>- Développement d'applications avec peu de compétences.</li><li>- Grande souplesse de l'interface utilisateur.</li></ul>	<p>Inadapté dans le cas d'un grand nombre d'utilisateurs connectés</p> <ul style="list-style-type: none"><li>- La mise à jour des applications est lourde (les applications sont en effet à mettre à jour au niveau de chaque client)</li><li>- Sécurité, performance et fiabilité faibles</li></ul>

## I. 2.3.3.L'architecture à serveur de bases de données :



**FIGURE 13 : L'ARCHITECTURE À SERVEUR DE BASE DE DONNÉES**

Le serveur prend en charge non seulement l'accès aux données mais aussi l'intégralité des traitements. Le client est dit « léger » car il ne comporte que l'interface de présentation des données (logique applicative).



Pour alléger la charge du serveur, il est possible de répartir la charge de travail sur deux machines différentes :

- une machine « serveur d'applications »
- une machine « serveur de bases de données »

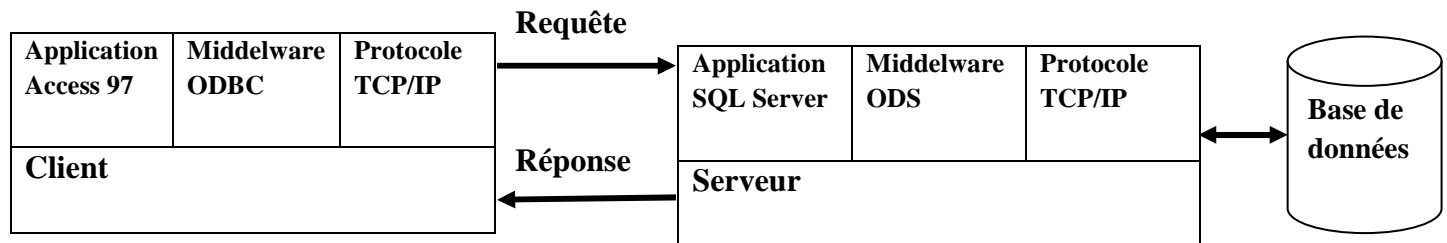
Il est également possible de répartir les traitements.

Avantages	Inconvénients
<ul style="list-style-type: none"><li>-Sécurité et fiabilité élevées.</li><li>- Hébergement de base de données de très grande taille (milliers de Go).</li><li>- Souplesse de l'interface utilisateur.</li><li>- Gestion centralisée performante.</li></ul>	<ul style="list-style-type: none"><li>-Coût élevé.</li><li>- Risque de forte sollicitation du serveur (échanges nombreux)</li><li>- Le serveur doit être surdimensionné (en mémoire vive pour l'exécution des programmes et en disque pour les données).</li></ul> <p>Dans les grandes organisations, le serveur</p>

# Chapitre I : Réseaux et architecture client serveur

	peut être multiprocesseur
--	---------------------------

Plus précisément, la communication est organisée en couches , garantissant l'indépendance de la solution client par rapport à la solution serveur.

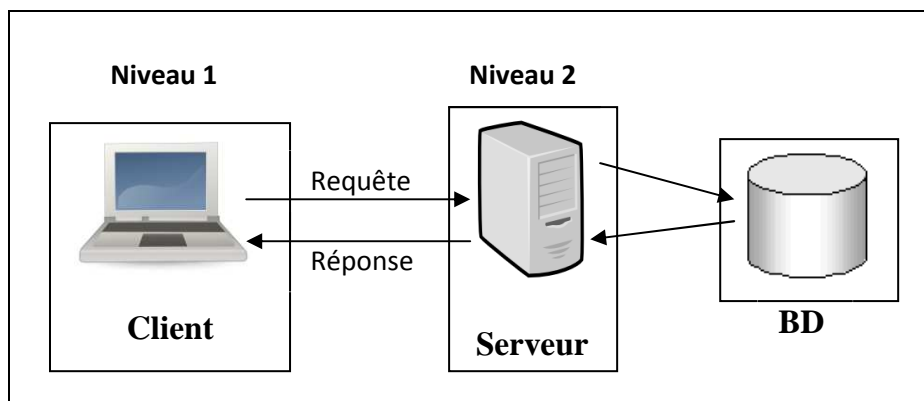


Le middelware est un programme permettant au poste client et au poste serveur de communiquer.

## I. 2.4. Types d'architectures client/serveur [Lefevre06] :

### I. 2.4.1. L'architecture à deux niveaux :

L'architecture à deux niveaux, appelée aussi architecture 2-tiers (tiers signifiant rangée en anglais) est l'architecture la plus classique. Caractérise les systèmes Client /Serveur pour lesquels le client demande une ressource et le serveur la lui fournit directement en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie de service. Elle décrit les systèmes client/ serveur dans lesquels, la logique applicative est enfouie soit dans l'interface utilisateur chez le client, soit dans la base de données chez le serveur (ou dans les deux à la fois).



**FIGURE 14 : ARCHITECTURE À DEUX NIVEAUX.**

Ce type d'architecture est une bonne solution d'informatique distribuée lorsque le nombre d'utilisateurs ne dépasse pas une centaine d'utilisateurs, cependant il existe d'une part une limite tenant au fait que la connexion est maintenue en permanence entre le client et le serveur, même si aucun travail n'est effectué, d'autre part les procédures d'accès aux données étant spécifiques aux moteurs de base de données, la flexibilité et le choix d'une base de données sont réduites. L'architecture à 3 niveaux, que nous allons détailler dans le point

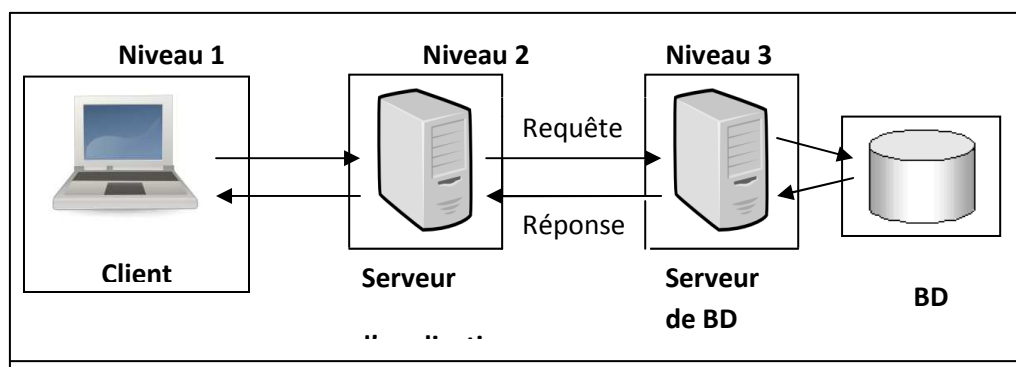


suivant, permet de dépasser ces limites, et d'apporter une meilleur réactivité de l'entreprise en cas de changements.

### I. 2.4.2. L'architecture à trois niveaux :

Dans l'architecture à trois niveau, appelée aussi architecture à 3-tiers, il existe un niveau intermédiaire (supplémentaire), entre es deux niveaux précédents alors on a généralement une architecture partagée entre :

1. **Un client** : C'est-à-dire le poste client (l'ordinateur demandeur de ressources, équipé d'une interface utilisateur (généralement un navigateur web) chargé de la présentation. Constitué de l'affichage et des traitements locaux (contrôle de saisie, mise en forme e données).
2. **Un serveur d'application** : appelé également **middeware**, chargé de fournir la ressource mais faisant appel à un autre serveur (niveau 3). Constitué des traitements applicatifs globaux.
3. **Le serveur de données** : Fournit au serveur d'application (niveau 2) les données dont il a besoin. Ses données sont localisées dans des bases de données pris en charge par un SGBD.



**FIGURE 15 : ARCHITECTURE À TROIS NIVEAUX.**

Le modèle à trois niveaux est recommandé si l'application possède l'une des caractéristiques suivantes :

- Nombreux service.
- Application écrite dans plusieurs langages ou par différentes organisations.
- Sources de données multiples et hétérogènes.
- Application évolutive (modification et ajout).
- Importante charge de traitement (transactions, accès simultané de plusieurs utilisateurs à la même base de données).
- Importante communication inter applications.

### **I. 2.4.3. Comparaison entre les architectures à deux et à trois niveaux :**

L'architecture à deux niveaux est une architecture Client /Serveur dans laquelle le serveur est polyvalent, c'est-à-dire qu'il est capable de fournir directement l'ensemble de ressources demandées par le client.

Par contre dans l'architecture à trois niveaux, les applications au niveau serveur sont délocalisées, c'est-à-dire que chaque serveur est spécialisé dans une tâche (serveur web /serveur de base de données par exemple). Ainsi cette architecture permet :

- Une plus grande flexibilité/souplesse.
- Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau.
- De meilleures performances, tant donné le partage de tâches entre les différents serveurs.

Le tableau ci-dessus représente la comparaison des deux architectures :

## Chapitre I : Réseaux et architecture client serveur

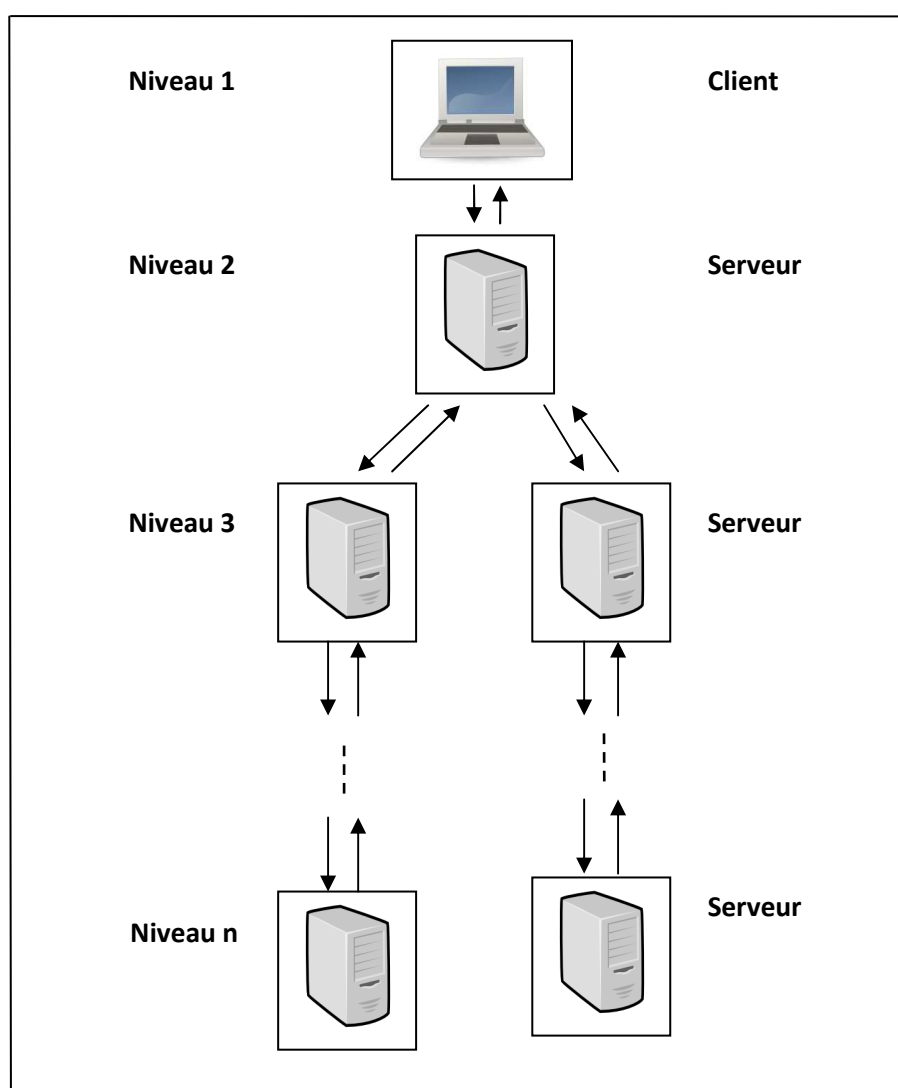
Architecture à deux niveaux	Architecture à trois niveaux
<ul style="list-style-type: none"><li>✓ L'application tourne sur le client et toute modification doit être opérée sur tous les clients d'où la complexité d'administration du système.</li><li>✓ Une sécurité faible au niveau des données.</li><li>✓ De nombreuses instructions SQL transitent le réseau, ce qui engendre une faible performance.</li><li>✓ Réutilisation faible (les programmes sont monolithiques).</li><li>✓ Facilité de développement.</li><li>✓ Ne supporte pas les bases de données hétérogènes.</li><li>✓ La souplesse de l'architecture matérielle est très limitée.</li><li>✓ Le support de l'Internet est faible (bande passante limitée).</li><li>✓ Une gestion limitée des liens de communications des clients, ce qui rend difficile le changement d'échelle.</li></ul>	<ul style="list-style-type: none"><li>✓ l'application est gérée de manière centralisée d'où l'administration du système est moins complexe.</li><li>✓ Une sécurité importante au niveau du service, méthode ou type d'objet.</li><li>✓ Réduction du trafic grâce aux procédures implantées sur le serveur d'où une bonne performance.</li><li>✓ Réutilisation importante (les programmes sont constitués de plusieurs composants).</li><li>✓ Difficulté de développement.</li><li>✓ Supporte les bases de données hétérogènes.</li><li>✓ La souplesse de l'architecture matérielle est excellente.</li><li>✓ Excellent support de l'internet.</li><li>✓ Possibilité de la répartition de la charge sur plusieurs serveurs, ce qui facilite le changement d'échelle.</li></ul>

**Tab 1 : Comparaison entre les architectures à deux et à trois niveaux.**

### I. 2.4.4 L'architecture à n niveaux :

L'architecture à n niveau, appelée aussi architecture à n-tiers, a été pensée pour concevoir des applications puissantes et simples à maintenir. Ce type d'architecture permet de distribuer plus librement la logique applicative, ce qui facilite la répartition de la charge entre tous les niveaux.

Cette architecture permet de pallier aux Limites des architectures précédentes.



**FIGURE 16 : ARCHITECTURE À N NIVEAUX.**

### Conclusion :

Dans ce chapitre on a présenté les réseaux informatique qui sont un moyen pour minimiser le coût de transport des informations, et d'augmenter les performances des systèmes ; On a vu ensuite l'architecture Client/Serveur qui est un environnement qu' on va utiliser pour la réalisation de notre système et qui est encore l'un des aspect de traitement coopératif .

Dans le prochain chapitre on va essayer de présenter ce traitement coopératif qui est un système d'organisation du travail et de distribution des roles des acteurs de l'entreprise.

# CHAPITRE II :

Le

traitement

coopératif

### I. Introduction :

Les nouvelles technologies disponibles pour la mise en œuvre des télécommunications essentiellement liées aux possibilités offertes par des transmissions hautes débit ont conduit à repenser l'architecture des applications. La transparence des facteurs temps et répartition est devenue possible. Les utilisateurs peuvent exprimer leur besoins en intégrant la dimension temporelle pour un environnement géographique étendu, pour la manipulation à distance des données multimédia. La répartition doit être comprise au sens large : répartition des données ; des traitements, des services et des accès utilisateurs. Les applications réparties sont non seulement communicantes mais aussi coopératives. Les processus d'applications distribués sur un espace géographique plus ou moins large doivent coopérer ensemble, avec cohérence et de façon optimale pour réaliser des tâches souvent complexes.

La coopération des composants logiciels pour la réalisation d'un système de traitement coopératif n'a de sens que dans la mesure où cela permet de réaliser et de traduire des activités humaines coopérantes. C'est ce qui a poussé les gens à penser au manière de définition de travail coopératif des individus appartenant à un groupe et d'identification des interrelations plus ou moins dynamiques et simultanées qui peuvent exister.

La télé-ingénierie coopérative est fortement d'actualité aujourd'hui car elle semble constituer un moyen pour améliorer la qualité et le temps de conception des produits logiciels. En effet, les besoins croissants dans ce domaine et les progrès technologiques récents en matière de réseaux informatiques longues distances, systèmes opératoires, technologies multimédias et applications distribuées ont ouvert la voix à de nouvelles façons de travailler. Ainsi la conception d'un système de traitement coopératif est avant tout un problème d'organisation du travail et de distribution des rôles des acteurs de l'entreprise.

### II. Les aspects coopérants des applications réparties :

Notre objectif dans ce paragraphe est d'identifier les besoins des applications coopérantes qui déterminent les composants architecturaux de ces applications. La satisfaction de ces exigences nécessite un ensemble d'outils et de moyens qui supportent les traitements répartis coopératifs ou non. Il est essentiellement consacré aux aspects liés à l'association de processus d'application répartis, à la représentation des données ainsi qu' à la synchronisation des processus.

Les premiers environnements de travail coopératif à être apparus sont les environnements pour le travail distribué asynchrone qui intègrent sous une interface unifiée des outils de courrier électronique textuels et/ou multi- médias, de transfert de fichier, de consultation de base de données distribuées, etc.

L'aspect coopérant d'une application peut avoir plusieurs résonances selon l'angle sous lequel on se place. Ainsi, « Hommes –Système » s'intéressera aux interactions logicielles

## Chapitre II : Le traitement coopératif

---

entre modules applicatifs répartis ou non. Notons qu'une application peut être coopérante et non distribué : c.-à-d. plusieurs entités logicielles interagissent entre elles, sur un même site sans invoquer des services de communication. La notion d'application coopérante n'est pas nouvelle, ce qui est récent, c'est l'intégration dans ce mode les moyens de communication étendus qui offrent une transparence temporelle et géographique de la répartition aux utilisateurs.

La perception des utilisateurs d'une application coopérante sera limitée à la visibilité externe des applications et surtout aux moyens qui sont à la fois importants et très spécifique pour invoquer une application coopérante. Il s'agira de doter les applications d'interfaces interactives avec les utilisateurs. Sachant que le niveau de complexité de cette interface externe des applications est corrélé à la nature de l'information manipulée (voix, données images). De plus, on devrait pouvoir appréhender une application toujours de la même façon indépendamment de la nature multimédia des données. L'interface doit rendre transparente la diversité de représentation de l'information et offrir des modalités de dialogue adaptées à l'utilisateur. Le degré d'interaction avec l'utilisateur est qualitativement et quantitativement variable d'une application à une autre.

Un « télécommunicant », seuls les aspects liés à la mise en œuvre des télécommunications seront retenus. Cette mise en œuvre se résume souvent à la maîtrise des aspects de transport de données et de gestion de systèmes. Rare sont ceux qui en plus disposent d'une vision globale de la répartition (critères et algorithmes de répartition, prise en compte des aspects sécuritaires et maîtrise des coûts), c'est-à-dire d'une vision de gestionnaire.

Une application ne devient pas coopérante, elle est conçue comme telle. Le développeur d'application aura lui aussi sa propre vision de l'aspect coopératif d'une application. Il est donc indispensable qu'il puisse disposer de méthodes et de langages de programmation adaptés. Pour l'informaticien, le réseau est transparent et n'est qu'un outil à disposition pour accéder à des ressources matérielles et logicielles distantes. Les contraintes temporelles, de fiabilité, de disponibilité et d'intégrité sont variables d'un domaine applicatif à un autre.

La figure suivante présente les différentes perceptions du traitement coopératif réparti.

## Chapitre II : Le traitement coopératif

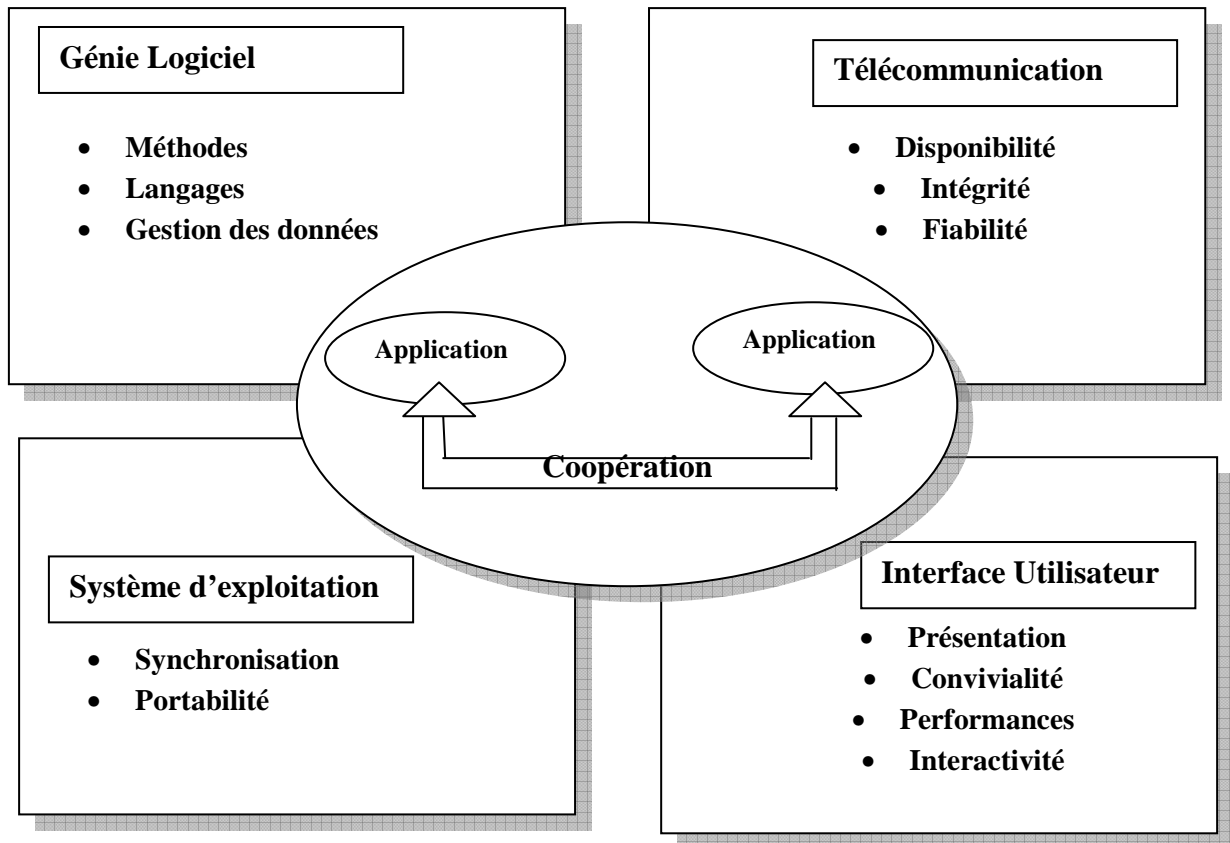


FIGURE : DIFFÉRENTES PERCEPTIONS DU TRAITEMENT COOPÉRATIF RÉPARTI

### III. Les caractéristiques du traitement coopératif réparti :

Quel est le produit logiciel qui, aujourd'hui, peut être conçu, réalisé, fabriqué et diffusé par un seul individu ? Quel que soit le domaine, quel que soit le produit ou le service à fournir, il est le fruit du travail d'une équipe ou d'un groupe d'individus souvent géographiquement distribués.

Une des caractéristiques importantes du travail coopératif est qu'il fait intervenir un groupe d'utilisateurs, et qu'en ce sens il est obligatoire de sortir du schéma classique des communications point à point. Il faut également ajouter que dans ce groupe d'utilisateurs, chaque individu a une fonction, un rôle et une importance qui lui sont propres par rapport au travail qu'il a à fournir. De ce fait, le traitement coopératif réparti consiste à coordonner des services répartis qui concourent à la réalisation d'une activité complexe à partir des tâches autonomes, tout en assurant la cohérence du système informatique qui les intègre. La notion de travail de groupe (Groupware) peut résumer ce qui est le traitement coopératif. Un groupe de personnes réalisent ensemble et en commun une activité déterminée (par exemple la constitution d'un document).

Comme la répartition concerne la délocalisation des traitements, des ressources, des utilisateurs et le contrôle de ces éléments, dont tout ou partie est fragmenté, il faut disposer



## Chapitre II : Le traitement coopératif

---

d'outils de coordination, de synchronisation, de communication et de gestion de cet ensemble. De plus, la nature multimédia de l'information relative au contexte applicatif d'appréhension impose de disposer d'interfaces d'accès aux services de coopération adaptés.

Mais en basant sur ces caractéristiques, le travail coopératif est rendu possible grâce aux nouvelles technologies de l'information et de communication (services réseaux, ordinateurs, composants pour les traitements multimédias, logiciels de communication et de coopération, etc.) qui permettra de créer des plateaux, ou même des entreprises virtuelles. Le but espéré par les entreprises est un gain en compétitivité, en offrant la possibilité à des ingénieurs et techniciens qui doivent coopérer d'enrichir et de multiplier le nombre de leurs interactions. Ainsi, il devrait être possible d'améliorer le facteur QCD (Qualité/Coût/Délai) des produits logiciels.

### IV. Organisation et traitement coopératif :

Les technologies de l'information et des communications ont profondément modifié les métiers à leur maîtrise. La structure des entreprises s'en trouve affectée. Les anciennes directions informatiques sont devenues, dans leur grande majorité, des directions des systèmes d'information. L'organisation, jusque-là ignorée ou considérée comme un sous produit des applications informatiques a retrouvé son rôle essentiel, par la mise en œuvre de structures d'organisation relationnelle et opérationnelle assurant la gestion de projet. La notion d'équipe pluridisciplinaire, appliquant des méthodes de conduite de projets, modélisée sur des objectifs clairement exprimés, est fondamentale au développement harmonieux des applications informatiques.

Pour développer un système d'information, il faut disposer d'une méthode de développement qui assure une analyse globale et une cohérence de toutes les applications spécifiques dont a besoin l'entreprise pour se réaliser. Toutes les caractéristiques de l'ingénierie classique doivent donc être appliquées dans le domaine de la réalisation des systèmes d'information (notion de l'ingénierie de l'information).

Ainsi le développement d'une application informatique est similaire au développement de tout autre produit. C'est-à-dire qu'il y a des phases de conception, de réalisation et d'industrialisation auxquelles on peut associer des plans d'assurance qualité. Le traitement coopératif étant partie prenante du système d'information est lui aussi à organiser lors de la phase de conception de ce dernier.

L'on pourrait résumer les phases générales d'une méthode de conception par ces quelques caractéristiques :

- Planification stratégique du système d'information en fonction de la stratégie de l'entreprise. Identification des grandes fonctions (domaines), données de l'entreprise et de leurs interactions. C'est à la direction générale de l'entreprise qu'appartient la planification des moyens de sa stratégie, en identifiant les objectifs généraux et les facteurs clés de succès pour y parvenir.

## Chapitre II : Le traitement coopératif

---

- Analyse des domaines précédemment identifiés séparément, une réflexion sur ce qui est fait et non sur la façon de le faire (Analyse du quoi et non du comment). Etude des systèmes existants et identification de la nature des nouvelles applications  
Qu'on aura besoin pour recouvrir un domaine.
- Conception fonctionnelle des applications informatiques indépendamment de la technologie qui les supporte. Informatisation possible des fonctions naturelles de l'entreprise et non plus d'un organigramme d'entreprise, dont les frontières d'informatisation sont non pas fixées par les fonctions naturelles de l'entreprise mais par l'homme. Ainsi, l'informatique n'est plus un frein à la réorganisation de l'entreprise.
- Réalisation technique des applications.
- Mise en œuvre des applications, préparation psychologique de formation des utilisateurs. Favoriser l'acceptation de nouveaux outils, dont l'introduction induit souvent des modes et des conditions de travail différent, pour une utilisation optimale du système d'information en minimisant les réactions de rejets.
- Maintenance, contrôle opérationnel, assurance de la qualité de service, des performances et de la satisfaction des utilisateurs.

### **Conclusion :**

Dans ce chapitre on a présenté le traitement coopératif avec ses aspects, ses caractéristiques, comme étant un système d'organisation du travail et de distribution des rôles des acteurs de l'entreprise.

Dans le prochain chapitre on va essayer de citer quelques généralités sur le langage de modélisation UML qu'on utilisera comme un outil de conception et technique de modélisation pour notre système.

# CHAPITRE III :

Généralité

Sur

UML

# Chapitre III : Généralité sur UML

---

## Introduction :

De nombreuses techniques de modélisation existent dans le monde de l'entreprise, Ses techniques s'agissent de méthodes, de langages de modélisation, ou d'outils informatiques intégrés. Notre choix dans le développement de notre système est basé sur **UML** qui est un langage de notation graphique standardisé qui peut être utilisé dans différents domaines de modélisation et notamment pour décrire des processus de gestion. UML, Unified Modeling Language, est un langage de modélisation orienté objets. Ce langage est le fruit du rapprochement des trois méthodes orientées objet : **Booch** conçue par **Grady Booch**, **OMT Object Modeling Technique** de **James Rumbaugh** et **OOSE Object Oriented Software Engineering** de **Ivar Jacobson**. Ces trois experts ont uni leurs expériences et compétences afin de définir un langage de modélisation standard et universel UML qui est supporté par l'**OMG Object Management Group**.

## I. Historique et origines de l'UML :

**UML (Unified Modeling Language)** est un langage standards de modélisation objet issu en 1996 de la pratique industrielle et de la modélisation des systèmes logiciels.

UML est un langage de notation orienté objet qui a été développé et standardisé par Rational Software et Object Management Group. Il a vu le jour en **1997** et est très rapidement devenu un standard de l'industrie pour spécifier, visualiser, développer et documenter des logiciels. Unified Modeling Language est né, comme son nom l'indique, du regroupement de trois techniques de modélisation objet, la méthode **Booch**, **Object Modeling Technique (OMT)** et **Objectory (OOSE)** Process, respectivement développées par **Grady Booch**, **James Rumbaugh** et **Ivar Jacobson**. Ces trois chercheurs travaillent actuellement chez Rational Software.

## II. L'intérêt d'utilisation UML :

L'intérêt d'utiliser un langage tel que UML est de:

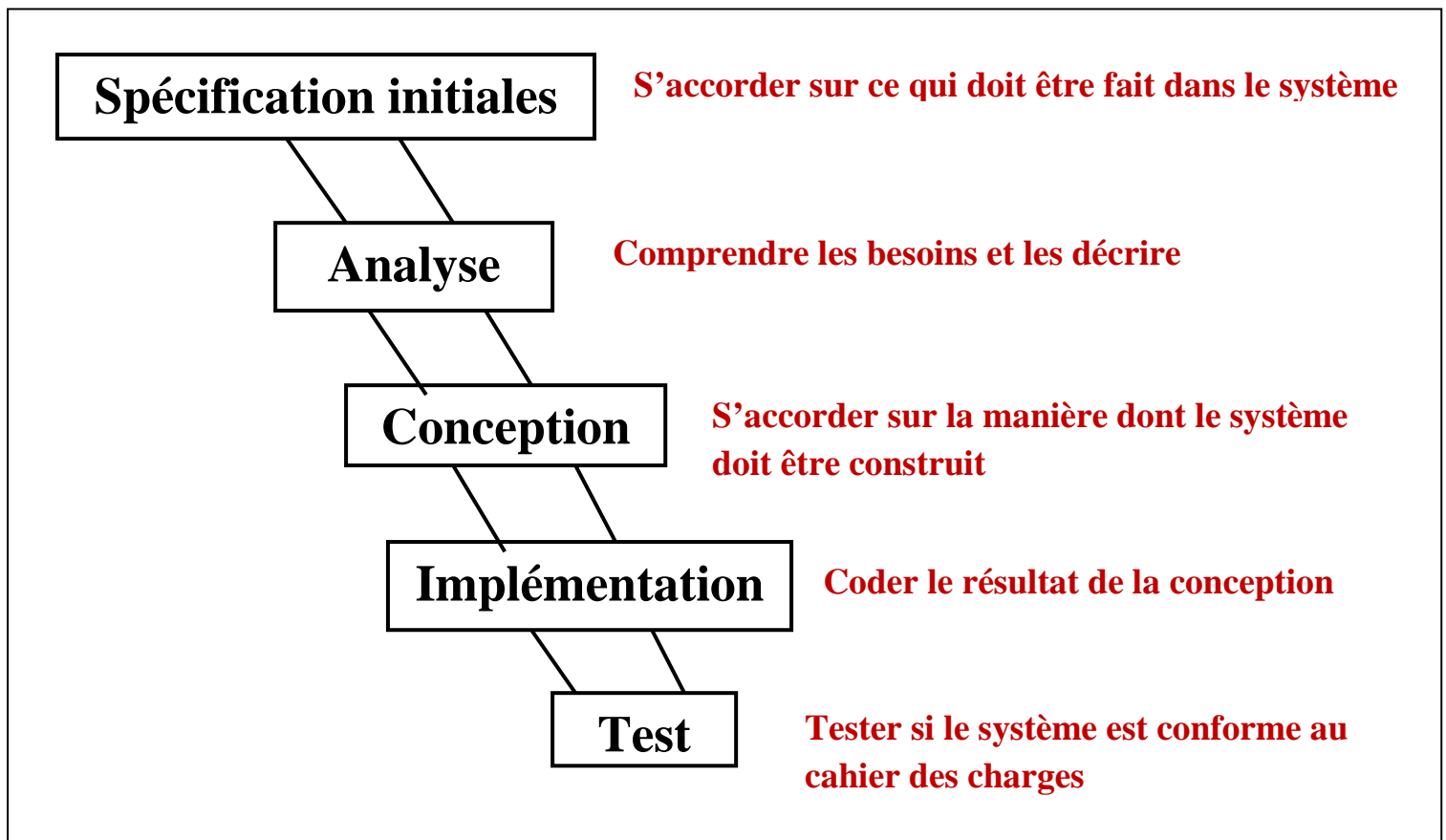
- Disposer d'un langage visuel permettant la représentation simple des fonctions et leurs relations,
- Utiliser un langage reconnu comme standard
- Utiliser des techniques éprouvées
- Utiliser des concepts similaires (tout pourra être rapporté à des objets, relations et interactions entre objets)
- Utiliser la même notation que celle utilisée pour la modélisation du système et resserrer ainsi l'écart entre les deux modélisations.

## III. Stades de développement :

UML couvre les différentes phases d'un développement objet (Spécification initiales, analyse, conception, implémentation et test) en offrant neuf types de diagrammes :

## Chapitre III : Généralité sur UML

- **Spécifications initiales du système:** définition et formulation des exigences provisoires.
- **Analyse:** Compréhension en profondeur des exigences à partir de la construction de modèles.
- **Conception du système:** Mise au point de l'architecture du système en instaurant les politiques de conception des classes.
- **Conception des classes:**
  - Augmentation et ajustement des modèles du monde réel issus de l'analyse en vue d'une compatibilité avec une implémentation informatique.
  - Détermination des algorithmes des opérations.
- **Implémentation:** Traduction de la conception en code.
- **Test:** Vérification du bon fonctionnement de l'application.



### IV. Styles de cycle de vie :

- **Développement en cascade:**
  - Séquence linéaire des différents stades
  - Pas de retour en arrière

## Chapitre III : Généralité sur UML

---

- Passage au stade suivant après la fin complète du stade précédent
- Pour des applications bien comprises avec des exigences bien stabilisées et des résultats d'analyse et de conception prévisibles
- Pas de livraison d'un système utilisable avant la finalisation complète du système

### ▪ Développement itératif:

- Développement en cascade du noyau du système
- Élargissement du périmètre du système par ajout successif de propriétés et de comportement aux objets existants et de nouveaux types d'objets
- Plusieurs itérations avant le livrable final –chaque itération comprenant un ensemble complet de stades
- Pas de construction du système dans son intégralité en une seule fois
- Valable pour la plupart des applications.

### V. Présentation des diagrammes UML:

UML possède neuf diagrammes principaux, dans le paragraphe qui suit nous allons développer chacun d'entre eux:

- **Le diagramme de cas d'utilisation** : représente les comportements d'un système du point de vue de l'utilisateur.

- **Le diagramme de classes** : représente la structure statique d'un système sous la forme de classes et de relations et ne contient pas d'informations temporelles. Une classe est une représentation abstraite d'un ensemble d'éléments similaires.

- **Le diagramme d'objets** : représente les objets et leurs relations, un objet étant un élément particulier d'une classe.

- **Le diagramme de séquence** : représente les objets et leurs interactions selon une ligne temporelle.

- **Le diagramme de collaboration** : représente les objets, leurs liens et leurs interactions de manière structurelle. Les diagrammes de séquence et de collaboration sont isomorphes.

- **Le diagramme de transition : d'états** exprime le comportement dynamique d'un objet en termes d'états, d'activités, de transitions et d'événements.

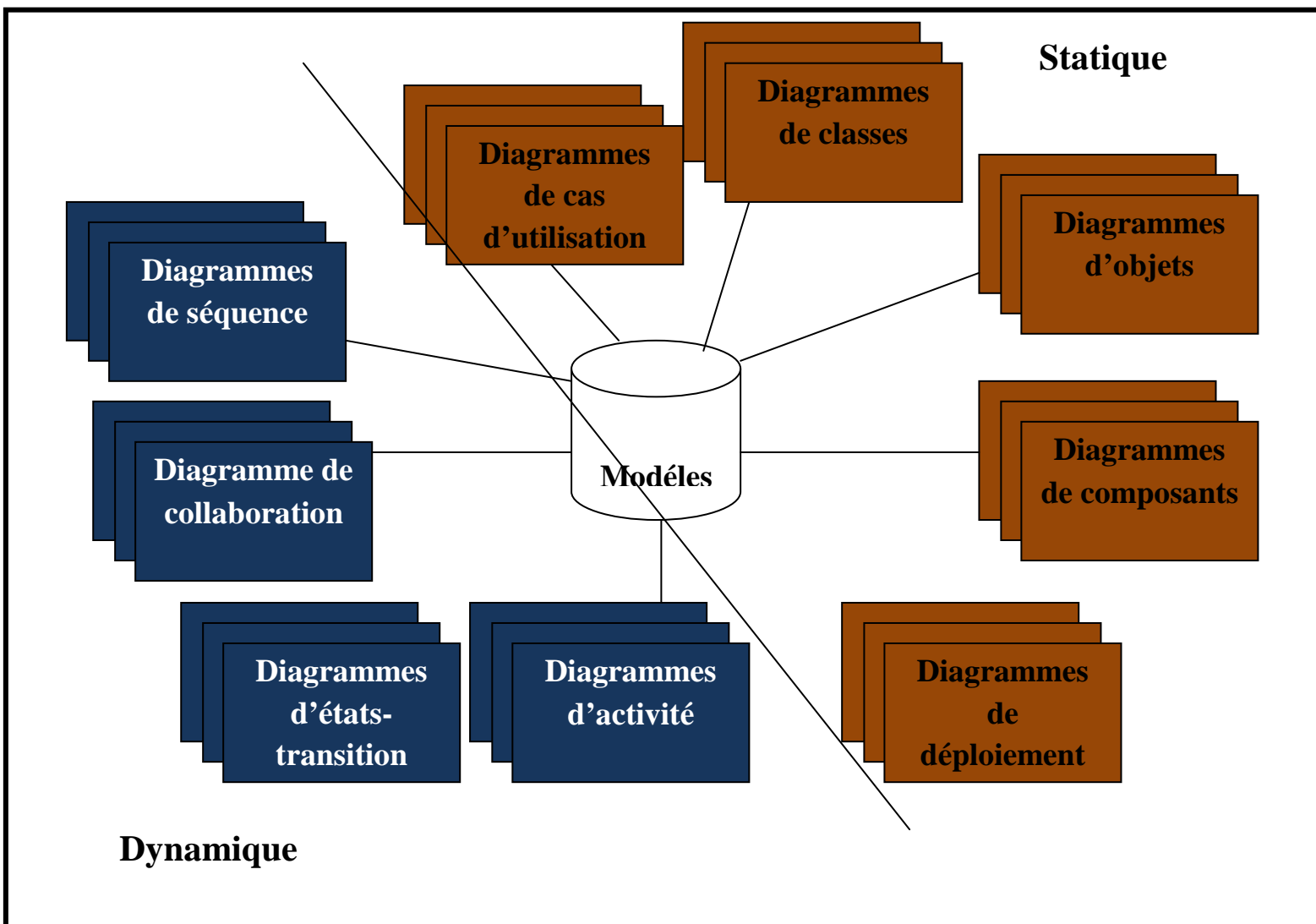
- **Le diagramme d'activités** : décrit les flux entre activités au sein d'un système. Cela permet de représenter le déroulement d'une procédure ou d'une fonction.

- **Le diagramme de composants** : montre l'implémentation physique d'un système, en termes de composants logiciels.

- **Le diagramme de déploiement** : décrit la configuration des éléments de traitement à l'exécution et les composants qui leur sont rattachés.

## Chapitre III : Généralité sur UML

➤ *Le schéma qui suit présente les différents diagrammes UML*



### RÉSUMÉ DE DIAGRAMMES UML

#### Conclusion :

Durant ce chapitre on a fait une bref description de langage de modélisation UML qu'on va utiliser dans la partie « Analyse et conception » de notre système comme étant un langage de modélisation standard.

Alors on passera au prochain chapitre « Principe de patron de conception » qui va enrichir notre mémoire par des connaissance importantes concernant l'utilité et le vrai sens des patrons

## Chapitre III : Généralité sur UML

---

logiciels qui jouent un rôle magique dans le domaine de l'architecture et de la génie logicielle.



# CHAPIRE IV :

Principe

Des

Patrons

De

Conception

### Partie 1 : Les patrons de conception :

#### I. 1 : Introduction :

Dans le domaine du génie logiciel, « *Le patron de conception* » (*Design pattern* en anglais) est une solution générique d'implémentation répondant à un problème spécifique. En général, un patron de conception décrit une structure de classes utilisant des interfaces, et s'applique donc à des **développements logiciels** utilisant **la programmation orientée objet**.

Encore est un concept destiné à résoudre les problèmes récurrents suivant le paradigme objet. En français on utilise aussi le terme motif de conception qui est une mauvaise traduction de «design pattern».

Les patrons de conception décrivent des solutions standards pour répondre à des problèmes d'architecture et de conception des logiciels. A la différence d'un algorithme qui s'attache à décrire d'une manière formelle comment résoudre un problème particulier, les patrons de conception décrivent des procédés de conception généraux. On peut considérer un patron de conception comme une formalisation de bonnes pratiques, ce qui signifie qu'on privilégie les solutions éprouvées (un patron de conception n'est considéré comme « prouvé » qu'une fois qu'il a été utilisé avec succès au moins dans trois cas).

Il ne s'agit pas de fragments de code, puisque les patrons de conception sont le plus souvent indépendants du langage de programmation, mais d'une méthode de conception, c'est-à-dire d'une manière standardisée de résoudre un problème qui s'est déjà posé par le passé. Le concept de patron de conception a donc une grande influence sur l'architecture logicielle d'un système.

On peut donc considérer les patrons de conception comme un outil de capitalisation de l'expérience appliqué à la conception logicielle.

#### I. 2 : Historique des patrons logiciels:

- 1987 - Cunningham et Beck utilisent les idées d'Alexander (Alexander a étudié les manières d'améliorer le processus de conception de bâtiments et des zones urbaines) pour développer un petit langage de patrons pour Smalltalk.
- 1990 – Le Gang des 4 (« Gang of Four » : Gamma, Helm, Johnson and Vlissides) commence à travailler à la compilation d'un catalogue de patrons de conceptions
- 1991 - Bruce Anderson donne le premier workshop de Patrons à l'OOPSLA
- 1993 - Kent Beck et Grady Booch sponsorisent la première réunion de ce qui est maintenant connu comme le groupe Hillside
- 1995 - Le Gang des 4 (GoF) publie le livre des *Patrons de conception*

#### I. 3 : L'utilité des patrons de conceptions dans le domaine du génie logiciel:

Un patron est « *Une solution à un problème dans un contexte* », et peuvent être utilisés dans de nombreux domaines différents, y compris le développement logiciel avec les intérêts suivants :

## Chapitre IV : Principe des patrons de conception

---

- ❖ **Devenir un champion aux échecs, en étudiant le jeu d'autres champions**
  - Ces jeux contiennent des patrons qui doivent être compris, mémorisés, puis appliqués de manière répétée.
- ❖ *"Concevoir un logiciel orienté-objet est difficile, et concevoir un logiciel orienté-objet réutilisable est encore plus difficile."* - Erich Gamma
- ❖ Les concepteurs expérimentés réutilisent des solutions qui ont fonctionné dans le passé
- ❖ Les systèmes orientés-objet bien structurés suivent des patrons récurrents pour les classes et objets
- ❖ Les patrons qui ont fonctionné dans le passé permettent d'être plus productif. Les conceptions qui en résultent sont plus flexibles et réutilisables.

### **I. 4 : Les avantages des patrons de conceptions :**



- Capturent l'expertise et la rendent accessible à des non-experts
- Réduisent le temps de développement
- Facilitent la communication entre les développeurs en fournissant un langage commun
- Facilitent la réutilisation réussie de conceptions
- Améliorent la documentation de conception
- Améliorent la compréhensibilité des conceptions

### **I. 5 : Les inconvénients des patrons de conceptions :**

- ❖ Les patrons sont simplistes, mais ils doivent être appliqués à votre problème
- ❖ Utiliser des patrons requiert la capacité de faire le lien entre un patron et votre problème!
- ❖ Les patrons sont validés par l'expérience, plutôt que par des tests automatiques

### **I. 6 : Classification du GoF pour les patrons de conception:**

#### **1. Patrons de création :**

-  Concernent le processus de la création d'objets
-  Les patrons de création aident à créer des objets pour vous, au lieu d'avoir à instancier les objets directement.

#### ❖ **Exemples de patrons de création :**

- **Fabrique :**
  - Une méthode dans une classe dérivée crée les instances associées
- **Fabrique abstraite :**

- *Fabrique pour construire des objets liés*

➤ **Monteur :**

- *Fabrique pour construire des objets complexes de manière incrémentale*

➤ **Prototype :**

- *Fabrique pour cloner de nouvelles instances d'un prototype*

➤ **Singleton :**

- *Fabrique pour n'avoir qu'une seule et unique instance*

### 2. Patrons de structure :



Concernent la composition de classes et d'objets



Les patrons de structure aident à composer des groupes d'objets en des structures plus larges, telles que des interfaces utilisateur complexes.

#### ❖ Exemples de patrons de structure :

➤ **Adaptateur :**

- Un traducteur qui adapte une interface de serveur pour un client

➤ **Pont :**

- Découpler l'interface d'une classe et son implémentation

➤ **Objet composite :**

- Structure pour construire des agrégats récursifs

➤ **Décorateur :**

- Etend un objet de manière transparente

➤ **Façade :**

- Façade simplifie l'interface pour un sous-système

➤ **Poids-mouche :**

- De nombreux objets partagés efficacement

➤ **Proxy :**

- Un objet est l'approximation d'un autre

### 3. Patrons de comportement :



Concernent l'interaction des classes et des objets



Les patrons de comportement aident à définir la communication entre les objets du système et définir comment le flux est contrôlé.

#### ❖ Exemples de comportement :

##### ➤ Chaine de responsabilité :

- Requête déléguée au fournisseur de service responsable

##### ➤ Commande :

- Requête comme objet de première classe

##### ➤ Interpréteur :

- Interpréteur de langage pour une petite grammaire

##### ➤ Itérateur :

- Eléments d'un agrégat sont atteints séquentiellement

##### ➤ Médiateur :

- Médiateur coordonne les interactions entre ses associés

##### ➤ Memento :

- Une photo qui capture et restaure des états d'objets

##### ➤ Observateur :

- Les observateurs sont mis au courant des changements des observés

##### ➤ Etat :

- Object dont le comportement dépend de son état

##### ➤ Stratégie :

- Abstraction pour la sélection d'un parmi plusieurs algorithmes

##### ➤ Patron de méthode :

- Algorithme avec des pas fournis par une classe dérivée

##### ➤ Visiteur :

- Opérations appliquées aux éléments d'une structure d'objet hétérogène

### **I. 7 : Les parties essentielles des patrons de conception:**

Chaque patron est une règle en 3 parties, qui exprime une relation entre un certain *contexte*, un *problème* et une *solution*.”

➤ **Nom de patron :**

- Un nom court et parlant améliore la communication entre développeurs

➤ **Problème :**

- Quel est le problème et le contexte pour lesquels nous utiliserions ce patron?
- Quelles sont les conditions pour l'utilisation de ce patron?

➤ **Solution :**

- Une description des éléments constituant le patron de conception

➤ **Conséquences :**

- Le pour et le contre de l'utilisation du patron
- Inclus les impacts sur la réutilisabilité, la portabilité et l'extensibilité

### Partie 2 : Le patron de conception logiciel J2EE

#### I. Introduction à J2EE :

**J2EE** (*Java 2 Enterprise Edition*) est une norme proposée par la société Sun, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants.

On parle généralement de «plate-forme J2EE» pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution. J2EE comprend notamment :

- Les spécifications du **serveur d'application**, c'est-à-dire de l'environnement d'exécution : J2EE définit finement les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées. Ces recommandations permettent ainsi à des entreprises tierces de développer des serveurs d'application conformes aux spécifications ainsi définies, sans avoir à redévelopper les principaux services.
- Des services, au travers d'**API**, c'est-à-dire des extensions Java indépendantes permettant d'offrir en standard un certain nombre de fonctionnalités. *Sun* fournit une implémentation minimale de ces API appelée **J2EE SDK** (*J2EE Software Development Kit*).

Dans la mesure où J2EE s'appuie entièrement sur le **java** il bénéficie des avantages et inconvénients de ce langage, en particulier une bonne portabilité et une maintenabilité du code.

De plus, l'architecture J2EE repose sur des composants distincts, interchangeables et distribués, ce qui signifie notamment :

- qu'il est simple d'étendre l'architecture ;
- qu'un système reposant sur J2EE peut posséder des mécanismes de haute-disponibilité, afin de garantir une bonne qualité de service ;
- que la maintenabilité des applications est facilitée.

#### II. Présentation de J2EE :

J2EE est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles :

- Un ensemble de spécifications pour une infrastructure dans laquelle s'exécutent les composants écrits en java : un tel environnement se nomme serveur d'application.

## Chapitre IV : Principe des patrons de conception

---

- Un ensemble d'API qui peuvent être obtenues et utilisées séparément. Pour être utilisées, certaines nécessitent une implémentation de la part d'un fournisseur tiers.

Sun propose une implémentation minimale des spécifications de J2EE : le J2EE SDK. Cette implémentation permet de développer des applications respectant les spécifications mais n'est pas prévue pour être utilisée dans un environnement de production. Ces spécifications doivent être respectées par les outils développés par des éditeurs tiers.

L'utilisation de J2EE pour développer et exécuter une application propose plusieurs avantages:

- Une architecture d'application basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement.
- La possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA ...
- La possibilité de choisir les outils de développement et le ou les serveurs d'applications utilisés qu'ils soient commerciaux ou libres

J2EE permet une grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants. Ce choix dépend des besoins auxquels doit répondre l'application mais aussi des compétences dans les différentes API de J2EE. L'architecture d'une application se découpe idéalement en au moins trois tiers :

- La partie cliente : c'est la partie qui permet le dialogue avec l'utilisateur. Elle peut être composée d'une application standalone, d'une application web ou d'applets.
- La partie métier : c'est la partie qui encapsule les traitements (dans des EJB ou des JavaBeans).
- La partie donnée : c'est la partie qui stocke les données.

### **III. Les API de J2EE :**

Les API de J2EE peuvent se répartir en trois grandes catégories :

- Les **composants**. On distingue habituellement deux familles de composants :
  - Les composants web : [Servlets](#) et [JSP](#) (Java Server Pages). Il s'agit de la partie chargée de l'interface avec l'utilisateur (on parle de *logique de présentation*).
  - Les composants métier : EJB (Enterprise Java Beans). Il s'agit de composants spécifiques chargés des traitements des données propres à un secteur d'activité (on parle de *logique métier* ou de *logique applicative*) et de l'interfaçage avec les bases de données.
- Les **services**, pouvant être classés par catégories :

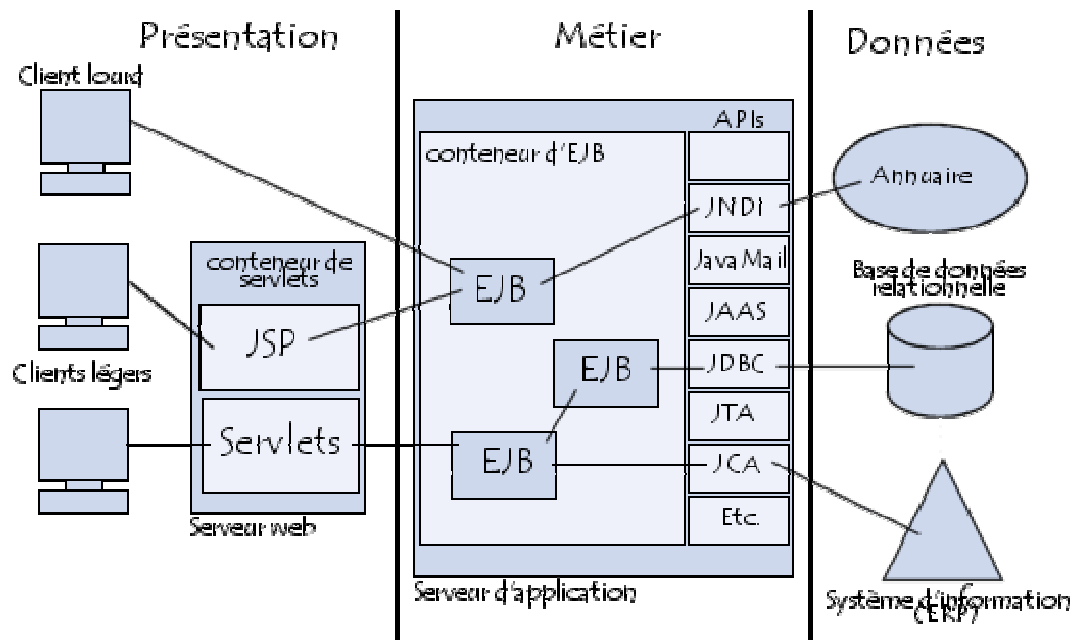


## Chapitre IV : Principe des patrons de conception

---

- Les **services d'infrastructures** : il en existe un grand nombre, définis ci-dessous :
  - **JDBC** (*Java DataBase Connectivity*) est une API d'accès aux bases de données relationnelles.
  - **JNDI** (*Java Naming and Directory Interface*) est une API d'accès aux services de nommage et aux annuaires d'entreprises tels que DNS, NIS, LDAP, etc.
  - **JTA/JTS** (*Java Transaction API/Java Transaction Services*) est un API définissant des interfaces standard avec un gestionnaire de transactions.
  - **JCA** (*J2EE Connector Architecture*) est une API de connexion au système d'information de l'entreprise, notamment aux systèmes dits «Legacy» tels que les [ERP](#).
  - **JMX** (*Java Management Extension*) fournit des extensions permettant de développer des applications web de supervision d'applications.
- Les **services de communication** :
  - **JAAS** (*Java Authentication and Authorization Service*) est une API de gestion de l'authentification et des droits d'accès.
  - **JavaMail** est une API permettant l'envoi de courrier électronique.
  - **JMS** (*Java Message Service*) fournit des fonctionnalités de communication asynchrone (appelées *MOM* pour *Middleware Object Message*) entre applications.
  - **RMI-IIOP** est une API permettant la communication synchrone entre objets.

L'architecture J2EE permet ainsi de séparer la couche présentation, correspondant à l'interface homme-machine (IHM), la couche métier contenant l'essentiel des traitements de données en se basant dans la mesure du possible sur des API existantes, et enfin la couche de données correspondant aux informations de l'entreprise stockées dans des fichiers, dans des bases de données relationnelles ou XML, dans des annuaires d'entreprise ou encore dans des systèmes d'information complexes.



## API J2E

### IV. L'environnement d'exécution des applications J2EE :

J2EE propose des spécifications pour une infrastructure dans laquelle s'exécutent les composants. Ces spécifications décrivent les rôles de chaque élément et précisent un ensemble d'interfaces pour permettre à chacun de ces éléments de communiquer.

Ceci permet de séparer les applications et l'environnement dans lequel il s'exécute. Les spécifications précisent à l'aide des API un certain nombre de fonctionnalités que doivent implémenter l'environnement d'exécution. Ces fonctionnalités sont de bas niveau ce qui permet aux développeurs de se concentrer sur la logique métier.

Pour exécuter ces composants de natures différentes, J2EE définit des conteneurs pour chacun de ces composants. Il définit pour chaque composant des interfaces qui leur permettront de dialoguer avec les composants lors de leur exécution. Les conteneurs permettent aux applications d'accéder aux ressources et aux services en utilisant les API.

Les appels aux composants se font par des clients via les conteneurs. Les clients n'accèdent pas directement aux composants mais sollicitent le conteneur pour les utiliser.

#### IV.1. Les services proposés par la plate-forme J2EE :

Une plate-forme d'exécution J2EE complète implémentée dans un serveur d'application propose les services suivants :

- service de nommage (naming service).
- service de déploiement (deployment service).

- service de gestion des transactions (transaction service).
- service de sécurité (security service).

Ces services sont utilisés directement ou indirectement par les conteneurs mais aussi par les composants qui s'exécutent dans les conteneurs grâce à leurs API respectives.

### **IV.2. L'assemblage et le déploiement d'applications J2EE :**

J2EE propose une spécification pour décrire le mode d'assemblage et de déploiement d'une application J2EE.

Une application J2EE peut regrouper différents modules : modules web, modules EJB ... Chacun de ces modules possède son propre mode de packaging. J2EE propose de regrouper ces différents modules dans un module unique sous la forme d'un fichier EAR (Entreprise ARchive).

Le format de cette archive est très semblable à celui des autres archives :

- un contenu : les différents modules qui composent l'application (module web, EJB, fichier RAR ...).
- un fichier descripteur de déploiement.

Les serveurs d'application extraient chaque module du fichier EAR et les déploient séparément un par un.

## **V. Principes de J2EE :**

- L'architecture J2EE est une architecture d'application distribuée à base de composants.
- Elle identifie et donne les spécifications des composants de l'application :
  - composants logiciels ou beans (EJB),
  - conteneur
  - serveurs
  - clients
- Les conteneurs isolent les beans du client et d'une implémentation spécifique du serveur.
- Les beans sont installés dans la partie serveur d'une application J2EE.
- Les conteneurs et serveurs implémentent les mécanismes de bas niveau utilisés par les applications:
  - transactions,

- persistance,
- gestion de la mémoire,
- Sécurité
- Les spécifications J2EE s'intéressent aux activités d'une application liées:
  - au développement,
  - au déploiement,
  - à l'exécution

### **VI. Composants J2EE :**

- Un composant est une unité logicielle de niveau applicatif.
- En plus des JavaBeans, qui font partie du J2SE, J2EE supporte les types de composants suivants :
  - ❖ applets,
  - ❖ application clientes,
  - ❖ composants Enterprise JavaBeans (EJB),
  - ❖ composants Web,
  - ❖ composants adaptateurs de ressource.
- ❖ Les applets et applications clientes sont exécutées sur le poste du client tandis que les composants EJB, Web et adaptateurs de ressources fonctionnent sur le serveur.
- ❖ A l'exception des adaptateurs de ressources, les concepteurs et développeurs d'application développent les composants d'une application J2EE.
- ❖ Les adaptateurs de ressources et logiciels associés sont en général vendus par les fournisseurs de systèmes d'information de l'entreprise et ensuite déployés sur les serveurs pour accéder aux données.


### **Conclusion :**

On vient de terminer notre présent chapitre, dans lequel on a présenté les patrons de conception logiciels qui sont des solutions génériques d'implémentation répondant à des problèmes spécifiques ; ensuite on a vu le patron J2EE, qui est une norme de définition d'un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants.

## Chapitre IV : Principe des patrons de conception

---

Notre suivant chapitre va identifier et présenter notre organisme d'accueil « Direction de distribution de gaz et d'électricité de tizi-ouzou ».



# **CHAPITRE v :**

## **Présentation**

### **De**

## **L'organisme**

### **D'accueil**

### **I. Définition de sonelgaz** : Société nationale d'électricité et de gaz

Une entreprise spécialisée dans la distribution des énergies électriques et gazières.

### **II. Décision de la direction générale de distribution :**

- Vue la loi N° 02 .01 de 05 /02/2002 relative à l'électricité et à la distribution de gaz par canalisation.
- Vue le statut de groupe sonelgaz.
- Vue la décision N° 463.DG/2004 du 27 juin 2004, portant création de la direction générale centre.
- Vue la décision N°049.DG/2005 du 23 janvier 2005 portant création du comité De suivi de l'organisation de distribution.
- Vue les conclusions de la réunion du comité de suivi de l'organisation de distribution.
- Vue la résolution N° 03/2005 du conseil d'administration de sonelgaz Spa du 30 avril 2005 portant organisation de la distribution.

# Décide

**Article01/** Organisation de la direction générale de distribution centre : relative au niveau centrale.

**Article02/** Organisation de la direction générale de distribution centre ; relative au niveau : Direction régionale, District [Service] Technique Electricité, District [Service] Technique Gaz ou [Subdivision] Exploitation Gaz, Agence [Service ou Subdivision] Commerciale.

**Article 03/** L'organisation adapté de la direction générale de distribution centre ; relative au cas de la direction régionale d'ILLIZI.

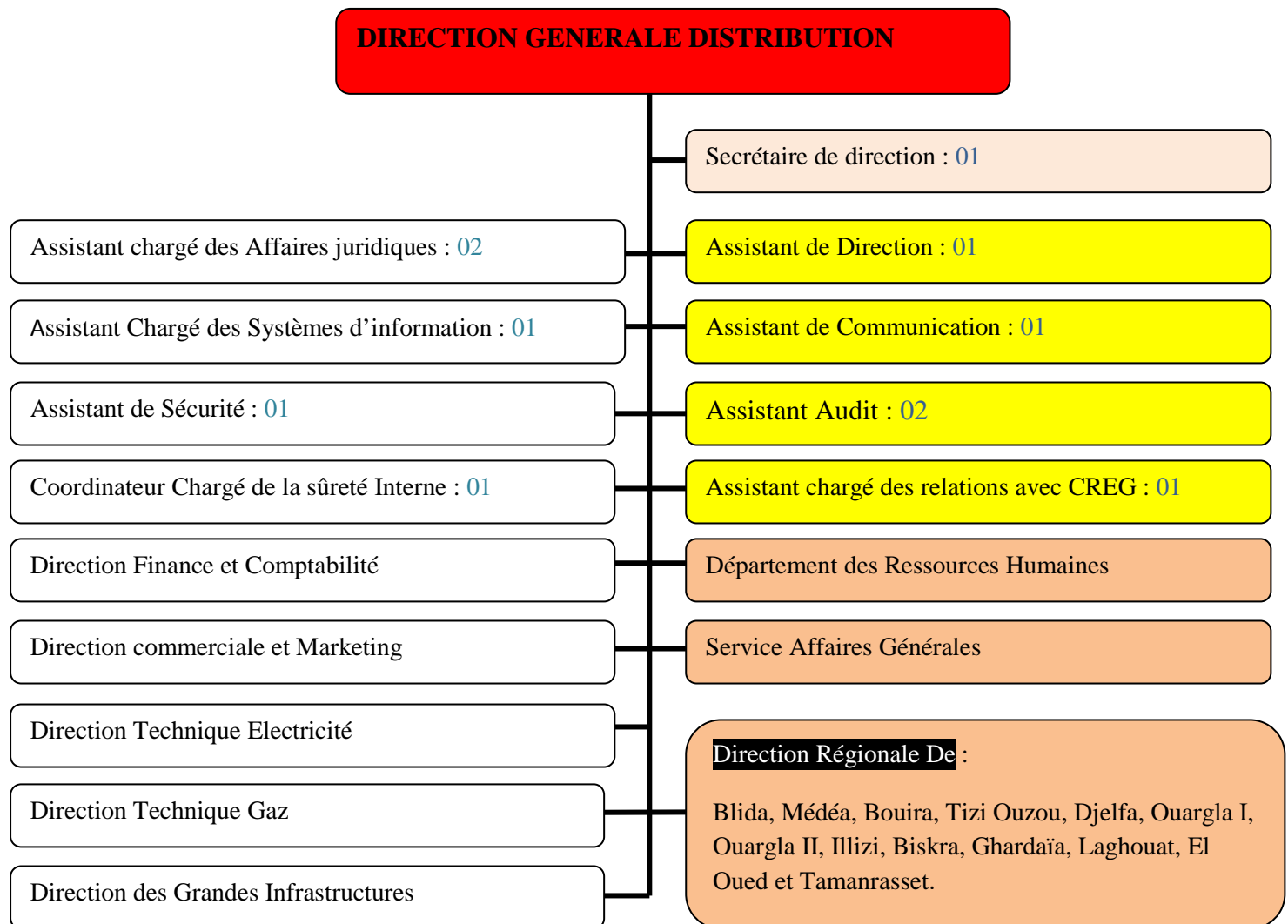
**Article 04/** Dans le cadre de la mise à jour des organigrammes ; la création de structures District Electricité ou District ou Exploitation Gaz ainsi que l'agence commerciale sont de ressort du conseil d'orientation et de surveillance sur proposition de directeur général de distribution.

**Article 05/** La présente décision annuelle et remplace la décision N° 187 DG et N° 100 DOI du 08 Avril 1998.

**Article06/** Le directeur général de distribution centre, Le secrétaire du groupe, Le directeur des ressources humaines Groupe, Le directeur de finance et comptabilité Groupe, Le directeur général de développement et système sont chargés, chacun en ce qui le concerne, de l'application de cette décision.

**Article07/** La date d'effet de cette décision est : 02 Juillet 2005.

### III. L'organigramme de la direction de distribution générale :



**IV. Les missions :** La direction générale de distribution est chargée, dans les limites de ses attributions, de la distribution de l'énergie électrique est gazière et de la satisfaction des besoins de la clientèle aux conditions requises de coût, de qualité de service, et de sécurité.

### **V. Les attributions :**

- Acheter l'énergie électrique et gazière et la revendre aux clients HT, MT, BT, HP, MP, et BP.
- Participer à l'élaboration de la politique commerciale du groupe (en matière de tarification, prestation aux clients, développement des ventes, conditions commerciales,...).
- Mettre en œuvre la politique commerciale du groupe en élaborant les règles de la gestion de la clientèle et les règles commerciales et en contrôler l'application.

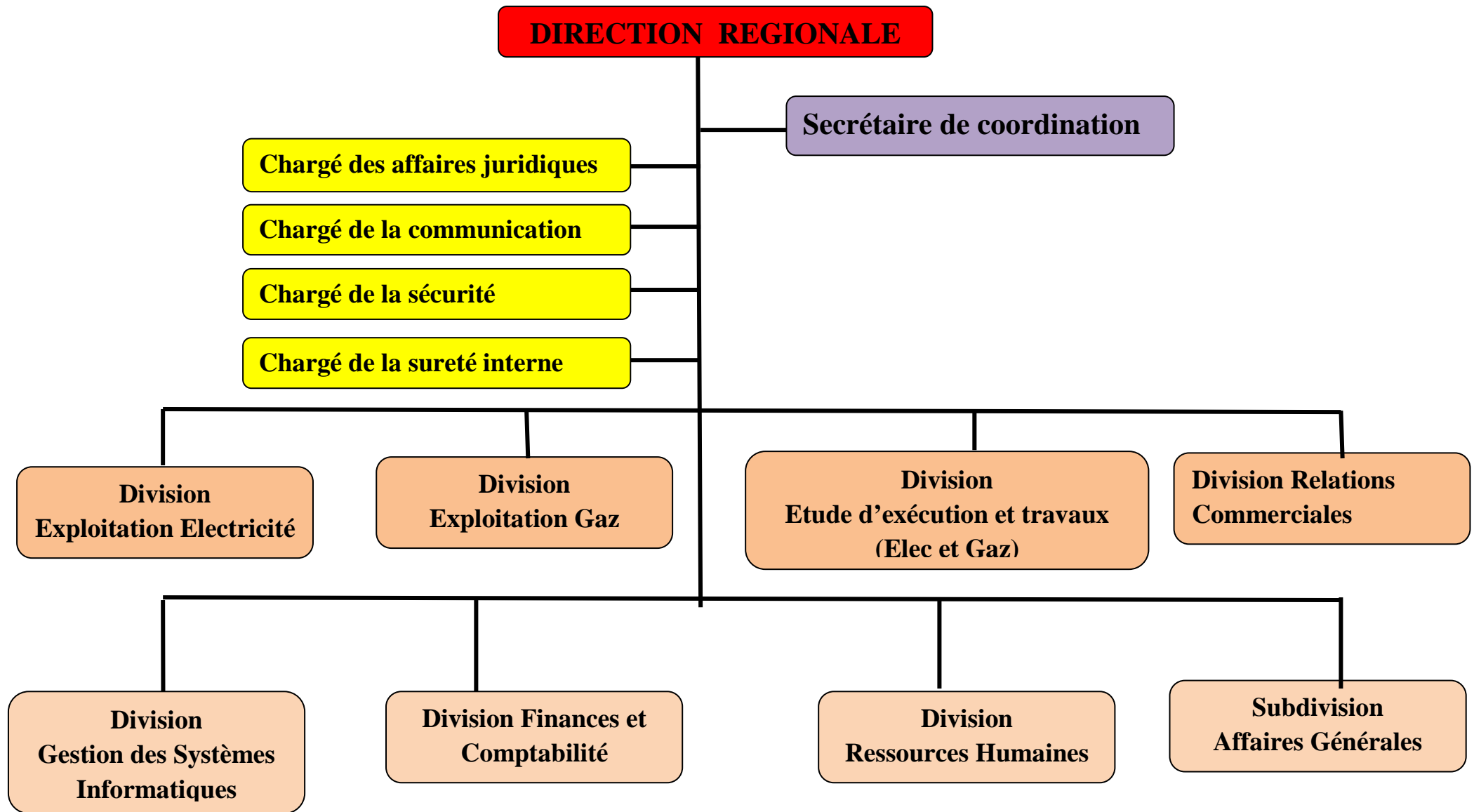


## Chapitre V : Présentation de l'organisme d'accueil

---

- Satisfaire aux meilleures conditions la demande de raccordement des clients électricité et gaz et leur apporter conseil et assistance dans les limites du cahier des charge et des règles en vigueur.
- Assurer la gestion (conduite, exploitation, et maintenance), et le développement des réseaux MT/BT, MP/BP et des installations annexes.
- Participer à l'élaboration des spécifications techniques, aux choix des matériels et leur homologation.
- Valider les programmes d'investissement.
- Participer à l'élaboration du PNE et des DPN et en assurer la maîtrise d'œuvre.
- Participer à la maintenance des systèmes de gestion.
- Elaborer les études d'organisation propre à la distribution en cohérence avec les décisions du groupe.
- Assurer la gestion et le développement de la ressource humaine et des moyens matériels nécessaires au fonctionnement de la distribution.
- Assurer la sécurité des personnes et des biens en rapport avec les activités de la distribution.

### VI. L'organigramme de la direction régionale :



### **VII. Les attributions de la direction régionale :**

- Participer à l'élaboration de la direction générale de distribution (en matière de prestations rendus aux clients, développement des ventes, recouvrement des créances,.....).
- Mettre en œuvre la politique commerciale de l'entreprise et en contrôler l'application
- Satisfaire aux meilleures conditions de coûts et de délais la demande de raccordement des clients MT/BT, MP/BP et leur apporter conseil et assistance
- Assurer la gestion (conduite, exploitation, et maintenance), et le développement des réseaux MT/BT, MP/BP et des installations annexes.
- Elaborer et mettre en œuvre le développement de la construction et la maintenance et l'exploitation des ouvrages.
- Etablir les programmes travaux qui se rapportent à ses missions et en assurer la maîtrise d'œuvre.
- Assurer la gestion et le développement de la ressource humaine et des moyens matériels nécessaires au fonctionnement du centre.
- Assurer la sécurité des personnes et des biens en rapport avec les activités de la distribution.
- Assurer la représentation de sonelgaz au niveau local.

### **Conclusion :**

Nous venons de terminer la présentation de notre organisme qui est une société nationale d'électricité et de gaz spécialisée dans la distribution des énergies électriques et gazières.

Dans le suivant chapitre, on va procéder à l'étude de notre existant en faisant une bref étude pour :

- 1. Les postes de travail.**
- 2. Les documents**
- 3. Les fichiers (Registres)**
- 4. Les procédures de travail**



**CHAITR VI :**

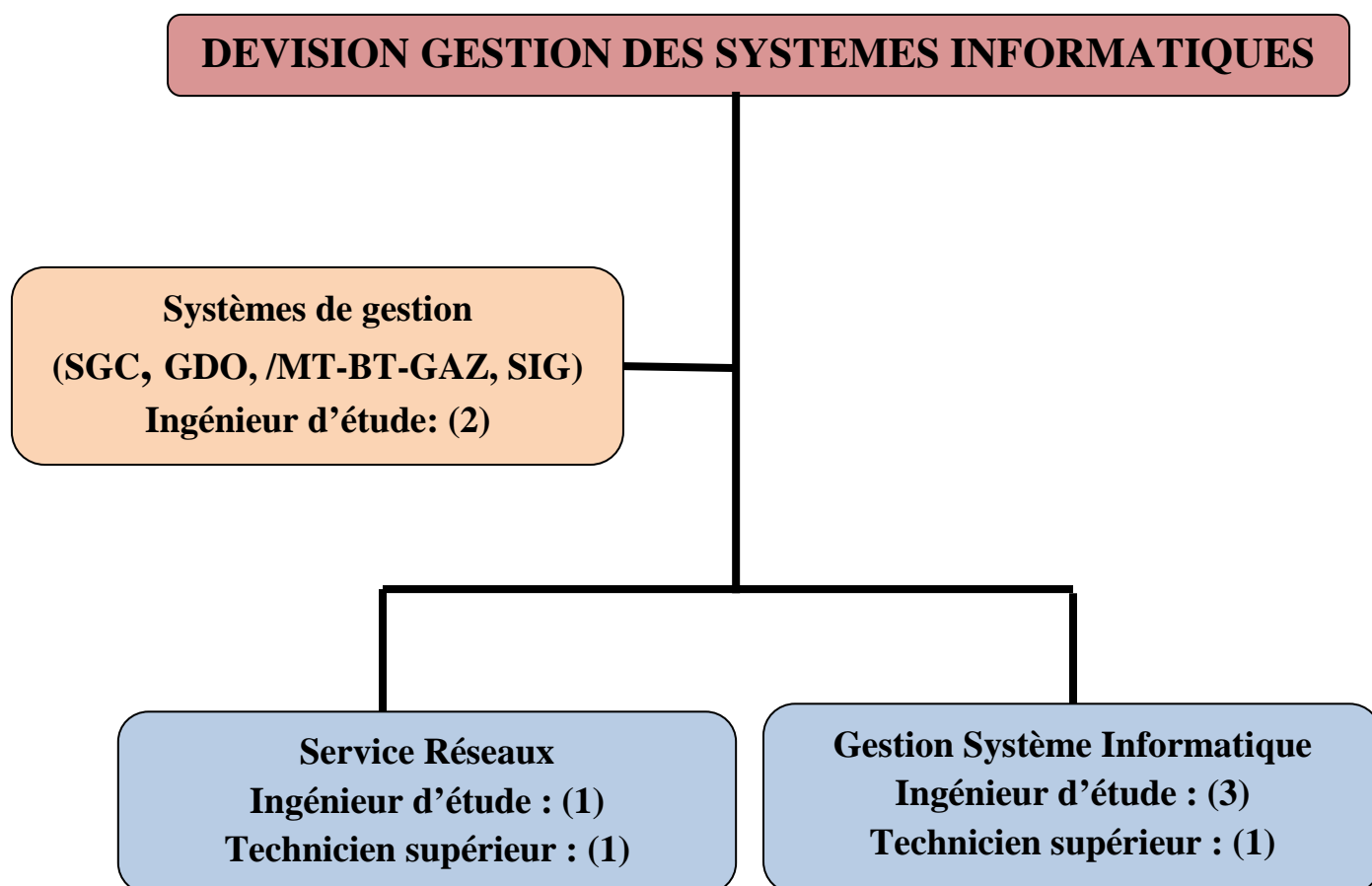
**Etude**

**De**

**L'existant**

### I. Présentation du champ d'étude :

#### I. 1 .Organigramme du champ d'étude :



#### I. 2. Description des éléments du domaine d'étude :

Le domaine d'étude est décomposé en services qui sont :

- **Service de systèmes de gestion** : L'ensemble des systèmes de gestion (Ex : Système de gestion de base de données SGBD, ...).
- **Service de gestion des systèmes informatiques** : service qui assure la gestion de papier, de parc informatique, de documents.
- **Service Réseaux** : Service d'assurance de bon fonctionnement des réseaux informatique de l'entreprise.

#### I. 3. Présentation des fonctions de la DGSI :

- Assurer la gestion du centre de traitement informatique.
- Gérer l'ensemble du matériel et périphériques affecté à la direction régionale.
- Approvisionner et contrôler la fourniture du consommable.
- Veiller à la maintenance des systèmes.
- Développer les applications propres à la direction régionale.

### **II. Analyse du champ d'étude :**

#### **II. 1. Introduction :**

La conception d'un logiciel implique la bonne connaissance de l'organisme qui va l'utiliser, et puisque le terme conception au niveau des applications correspond à la formalisation d'informations ; Donc l'objectif de notre chapitre « **Etude de l'existant** » est de recueillir les informations et de formaliser les données.

Durant ce chapitre nous allons étudier :

- Les postes du travail
- Les documents
- Les registres (fichiers)
- Le diagramme des flux d'informations
- Les procédures de travail

#### **II. 2. Etude des postes de travail :**

➤ **Définition d'un poste de travail :** Un poste de travail est l'un des composants principaux de l'entreprise ; est une unité d'action dans une organisation. Autrement dit est un lieu où un ensemble de tâches est exécutées dans cette organisation.

➤ **Caractérisation d'un poste de travail :**

Chaque poste de travail est caractérisé par :

- **Un type de lieu :** Représente l'ensemble de lieux où les actions d'une opération peuvent s'effectuer.
- **Un responsable :** L'ensemble des personnes ayant la responsabilité de certaines actions d'une opération.
- **Des ressources :** Se sont les moyens permettant de réaliser certaines actions d'une opération ; on peut distinguer : des ressources humaines, des ressources matérielles, des ressources logicielles, et des ressources humaines- matérielles (les systèmes personne-machine)

## Chapitre VI : Etude de l'existant

### Fiche d'analyse du poste N° 1 : (Administrateur des SGBD)

<b>1. Identifications</b>	
<p><b>Code :</b> AdminSGBD.</p> <p><b>Désignation :</b> Administrateur des SGBD.</p> <p><b>Profil :</b> Ingénieur d'étude.</p> <p><b>Service de rattachement :</b> Service Systèmes de gestion</p> <p><b>Effectif :</b> 01.</p>	
<b>02 Responsabilités</b>	
<ul style="list-style-type: none"><li>Assurer l'administration et la surveillance des systèmes de gestion des bases de données.</li></ul>	
<b>03 Tâches</b>	
Tâche	Fréquence
-Administration des SGBD.	Chaque jour
-Surveillance des données.	Chaque jour
-Installation des nouvelles versions des SGBD.	Aléatoire
-Installation de certaines nouveautés.	Aléatoire
-Modifier la structure de la table dans la base de données.	Aléatoire

### Fiche d'analyse du poste N° 2 : (Administrateur des systèmes d'exploitation)

<b>1. Identifications</b>	
<p><b>Code :</b> AdminOS.</p> <p><b>Désignation :</b> Administrateur des Systèmes d'exploitation.</p> <p><b>Profil :</b> Ingénieur d'étude.</p> <p><b>Service de rattachement :</b> Service Systèmes de gestion</p> <p><b>Effectif :</b> 01.</p>	
<b>02 Responsabilités</b>	
<ul style="list-style-type: none"><li>Assurer l'administration et la surveillance des systèmes d'exploitation.</li></ul>	
<b>03 Tâches</b>	
Tâche	Fréquence
-Administration des systèmes d'exploitation.	Chaque jour
-Surveillance des fichiers.	Chaque jour
-Création d'utilisateurs.	Aléatoire
-Installation des nouvelles versions des systèmes d'exploitation.	Aléatoire
-Installation de certaines nouveautés.	Aléatoire



## Chapitre VI : Etude de l'existant

### Fiche d'analyse du poste N° 3 :

<b>1. Identifications</b>
<p><b>Code :</b> GestStockPapier</p> <p><b>Désignation :</b> Gestionnaire de stock papier.</p> <p><b>Profil :</b> Ingénieur d'étude</p> <p><b>Service de rattachement :</b> Service Gestion des Systèmes Informatiques</p> <p><b>Effectif :</b> 01.</p>
<b>02 Responsabilités</b>
<ul style="list-style-type: none"> <li>Définir les besoins en papier nécessaire.</li> </ul>

03 Tâches	
Tâche	Fréquence
<ul style="list-style-type: none"> <li>-Gestion des approvisionnements de papier.</li> <li>-Gestion des transferts de papier.</li> <li>-Demande d'état de stock.</li> </ul>	<p>Aléatoire</p> <p>Aléatoire</p> <p>Après chaque transfert ou approvisionnement</p>

#### Documents et fichiers sortants :

Désignation	Destination	Observations
<ul style="list-style-type: none"> <li>Demande de prestation de service</li> <li>Demande d'édition d'état de stock</li> </ul>	<ul style="list-style-type: none"> <li>-Le responsable, Service d'affaires général (SAG)</li> <li>-Le gestionnaire de documents</li> </ul>	<ul style="list-style-type: none"> <li>-Pour cerner tous les aspects de la demande de papier.</li> <li>-Pour connaître l'état de stock après chaque approvisionnement ou transfert papier.</li> </ul>

## Chapitre VI : Etude de l'existant

### Documents et fichiers entrants :

Désignation	Origine	Observations
<ul style="list-style-type: none"><li>• Accusé de réception</li><li>• Décision d'affectation</li><li>• Demande de prestation de service</li><li>• Décharge interne ou externe</li></ul>	<ul style="list-style-type: none"><li>-Le responsable</li><li>-Service d'affaires général (SAG</li><li>-Autres divisions de la direction ou bien les agences locales.</li><li>-Gestionnaire de document</li></ul>	<ul style="list-style-type: none"><li>-Accord de la commande de papier</li><li>-Confirmation de transmission de papier.</li><li>-Vient d'autres divisions de DRD, ou agences locaux.</li><li>-Les décharges éditent par le gestionnaire de document après chaque transfert ou l'entrée d'un produit (Papier, Matériel).</li></ul>

### Documents et fichiers établis :

Désignation	Destination	Observations
<ul style="list-style-type: none"><li>• Fichier courrier arrivé (mise à jour)</li><li>• Fichier courrier sorti (mise à jour)</li></ul>	Aucune	/

## Chapitre VI : Etude de l'existant

### Fiche d'analyse du poste N° 4:

<b>1. Identifications</b>	
<p><b>Code :</b> GestParcInfo</p> <p><b>Désignation :</b> Gestionnaire de parc informatique.</p> <p><b>Profil :</b> Ingénieur d'étude</p> <p><b>Service de rattachement :</b> Service Gestion des Systèmes Informatiques</p> <p><b>Effectif :</b> 01.</p>	
<b>02 Responsabilités</b>	
<ul style="list-style-type: none"><li>• Définir les besoins en matériel informatique</li></ul>	
<b>03 Tâches</b>	
Tâche	Fréquence
-Gestion du matériel sorti de la DGSI	Aléatoire
-Gestion des réparations (Les interventions)	Aléatoire
-Gestion de nouveau matériel à arriver (les dotations)	Aléatoire

## Chapitre VI : Etude de l'existant

### Documents et fichiers sortants :

Désignation	Destination	Observations
<ul style="list-style-type: none"> <li>• Demande de prestation de service.</li> </ul>	-Service d'affaires général (SAG)	-Pour cerner tous les aspects de la demande de matériel.
<ul style="list-style-type: none"> <li>• Demande d'édition d'une décharge interne ou externe.</li> </ul>	-Le gestionnaire de documents	-Pour garder la traçabilité de matériel entré ou sorti de la DGSI.
<ul style="list-style-type: none"> <li>• Accusé de réception</li> </ul>	-Autres postes ou Service de la DGSI	-Comme Accord pour une demande.
<ul style="list-style-type: none"> <li>• Décharge d'intervention non remplie.</li> </ul>	-Technicien supérieur	-Pour cerner l'état de l'intervention.

### Documents et fichiers entrants :

Désignation	Origine	Observations
<ul style="list-style-type: none"> <li>• Décharge interne matériel</li> </ul>	-GesDoc	-Pour garder la traçabilité de matériel entré à la DGSI.
<ul style="list-style-type: none"> <li>• Décharge externe matériel</li> </ul>	-GestDoc	-Pour garder la traçabilité de matériel sorti de la DGSI
<ul style="list-style-type: none"> <li>• Décharge interne réparation</li> </ul>	- GestDoc	-Pour garder la traçabilité de matériel réparé au niveau de la DGSI.
<ul style="list-style-type: none"> <li>• Décharge externe réparation</li> </ul>	-GestDoc	-Pour garder la traçabilité de matériel réparé pour les autres divisions de DRD, et les agences locales.
<ul style="list-style-type: none"> <li>• Demande de prestation de service</li> </ul>	-GestStokPap, GestDoc, Responsable, Agences, Division DRD	-Pour la demande d'intervention, de matériel,

## Chapitre VI : Etude de l'existant

### Fiche d'analyse du poste N° 5 :

<b>1. Identifications</b>												
<p><b>Code :</b> GestDoc</p> <p><b>Désignation :</b> Gestionnaire des documents.</p> <p><b>Profil :</b> Ingénieur d'étude</p> <p><b>Service de rattachement :</b> Service Gestion des Systèmes Informatiques</p> <p><b>Effectif :</b> 01.</p>												
<b>02 Responsabilités</b>												
<ul style="list-style-type: none"><li>• Gérer les documents internes et externes.</li></ul>												
<b>03 Tâches</b>												
<table border="1"><thead><tr><th>Tâche</th><th>Fréquence</th></tr></thead><tbody><tr><td>-Réception des documents.</td><td>Aléatoire</td></tr><tr><td>-Sauvegarde des documents.</td><td>Aléatoire</td></tr><tr><td>-Edition des documents.</td><td>Aléatoire</td></tr><tr><td>-Classement des documents</td><td>Aléatoire</td></tr><tr><td>-Transfert des documents.</td><td>Aléatoire</td></tr></tbody></table>	Tâche	Fréquence	-Réception des documents.	Aléatoire	-Sauvegarde des documents.	Aléatoire	-Edition des documents.	Aléatoire	-Classement des documents	Aléatoire	-Transfert des documents.	Aléatoire
Tâche	Fréquence											
-Réception des documents.	Aléatoire											
-Sauvegarde des documents.	Aléatoire											
-Edition des documents.	Aléatoire											
-Classement des documents	Aléatoire											
-Transfert des documents.	Aléatoire											

### Documents et fichiers sortants :

Désignation	Destination	Observations
<ul style="list-style-type: none"><li>• Etat de stock</li><li>• Décharge interne ou externe.</li></ul>	<ul style="list-style-type: none"><li>-GestStockPap</li><li>-GestParcInfo</li></ul>	<ul style="list-style-type: none"><li>-Après la demande d'édition.</li><li>-Après la demande d'édition par GestParcInfo.</li></ul>

## Chapitre VI : Etude de l'existant

---

### Documents et fichiers entrants :

Désignation	Origine	Observations
<ul style="list-style-type: none"><li>Décharge externe.</li></ul>	- Le responsable	-Pour classement, sauvegarde, et garder traçabilité des documents sortis.
<ul style="list-style-type: none"><li>Demande de prestation de service.</li></ul>	-Le responsable	- Pour classement, sauvegarde, et garder traçabilité des demandes.

### Documents et fichiers établis :

Désignation	Destination	Observations
<ul style="list-style-type: none"><li>Etat de stock</li><li>Décharge interne</li></ul>	-GestStockPap -GestParcInfo	Réponse à la demande

## Chapitre VI : Etude de l'existant

### Fiche d'analyse du poste N° 6 :

<b>1. Identifications</b>	
<p><b>Code :</b> ReparInfo</p> <p><b>Désignation :</b> Réparateur informatique</p> <p><b>Profil :</b> Technicien supérieur.</p> <p><b>Service de rattachement :</b> Service Gestion de Systèmes Informatiques et service réseaux.</p> <p><b>Effectif :</b> 02.</p>	
<b>02 Responsabilités</b>	
<ul style="list-style-type: none"><li>• . Suivre les différentes pannes matérielles et logicielles</li><li>• Assurer le bon fonctionnement de réseaux informatique de l'entreprise.</li><li>• Assurer la maintenance du matériel</li></ul>	
<b>03 Tâches</b>	
<b>Tâche</b>	<b>Fréquence</b>
<ul style="list-style-type: none"><li>▪ Suivi de la réparation</li><li>▪ Tourner aux agences pour voir l'état de matériel</li><li>▪ Maintenance du système</li><li>▪ Vérifier si le matériel est en bon état</li></ul>	

## Chapitre VI : Etude de l'existant

### Fiche d'analyse du poste N° 7 :

<b>1. Identifications</b>	
<p><b>Code :</b> AdminRéseau</p> <p><b>Désignation :</b> Administrateur réseaux</p> <p><b>Profil :</b> Ingénieur d'étude</p> <p><b>Service de rattachement :</b> Service Réseaux</p> <p><b>Effectif :</b> 01.</p>	
<b>02 Responsabilités</b>	
<ul style="list-style-type: none"><li>Assurer le bon fonctionnement de réseaux informatique de l'entreprise</li></ul>	
<b>03 Tâches</b>	
<b>Tâche</b>	<b>Fréquence</b>
<ul style="list-style-type: none"><li>Maintenance et développement réseaux (Développement = remplacer les périphériques qui ne répond pas aux besoins)</li><li>Configurer d'une manière adéquate le réseau afin d'assurer une sécurité optimale (vérification et amélioration).</li><li>Faire des extensions s'il faut</li></ul>	Selon les besoins des utilisateurs et selon les pannes Chaque année.



### Fiche d'analyse du poste N° 8 :

<b>1. Identifications</b>	
<p><b>Code :</b> ResponDGSI</p> <p><b>Désignation :</b> Le responsable de la DGSI</p> <p><b>Profil :</b> Ingénieur d'étude + l'expérience</p> <p><b>Service de rattachement :</b> /</p> <p><b>Effectif :</b> 01.</p>	
<b>02 Responsabilités</b>	
<ul style="list-style-type: none"><li>• Prend la responsabilité de toutes les activités et les employeurs de la DGSI.</li></ul>	
<b>03 Tâches</b>	
Tâche	Fréquence
<ul style="list-style-type: none"><li>▪ Gérer les demandes</li><li>▪ Consultation de la base de données</li><li>▪ </li></ul>	Aléatoire Aléatoire

### **II. 3. Etude des documents :**

#### **II. 3.1. Définitions:**

Un document est une fiche ayant un ensemble d'informations concernant un élément bien précis.

#### **II. 3.2: Les types d'un document :** on peut distinguer deux types de documents :

- ❖ **Document interne :** est le document qui n'est utilisé qu'à l'intérieur du domaine d'étude ; on distingue deux types :
  - a. **Document interne de position :** est le document qui est utilisé qu'à l'intérieur de poste où il a été créé.
  - b. **Document interne de liaison :** Circule entre les postes de domaine d'étude.
- ❖ **Document externe :** est le document qui provient ou est destiné aux acteurs externes.

#### **II. 3.3. Liste des documents:**

- Demande de prestation de service.
- Décharge matériel interne
- Décharge matériel externe
- Décharge réparation interne
- Décharge réparation externe
- Rapport d'intervention
- Ordre de mission
- Certificat de réforme
- PV constat de la réforme

#### **II. 3.4. Etude détaillé de chaque document:**

## Chapitre VI : Etude de l'existant

<b>Fiche d'analyse du document N° 01</b>			
<p><b>Code :</b> Alphanumérique</p> <p><b>Désignation :</b> Demande de prestation de service.</p> <p><b>Support :</b> Papier.</p> <p><b>Type :</b> externe.</p> <p><b>Nombre d'exemplaires :</b> 01.</p> <p><b>Poste créateur :</b> GestParcInfo, Autres division de DRD, Les agences locales</p> <p><b>Poste destinataire :</b> SAG, DGSI</p> <p><b>Rôle :</b> Demander la prestation de service (ex : le papier, le matériel,...)</p>			
<b>Désignation du document</b>			
<b>Rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Code de document</li> <li>▪ Nom de service émetteur</li> <li>▪ Nom de service récepteur</li> <li>▪ Objet de la demande</li> <li>▪ Imputation des dépenses</li> <li>▪ Date de la demande</li> <li>▪ Nom et signature de l'agent demandant.</li> </ul>	AN A A A A Date	10 04 04 40 10 08	Ex : PS0526/13/14 Ex : DGSI Ex : DGSI  JJ/MM/AAAA
<b>Total</b>		<b>76</b>	

## Chapitre VI : Etude de l'existant

<b>Fiche d'analyse du document N° 02</b>			
<p><b>Code :</b> Alphanumérique</p> <p><b>Désignation :</b> Décharge interne matériel.</p> <p><b>Support :</b> Papier.</p> <p><b>Type :</b> interne.</p> <p><b>Nombre d'exemplaires :</b> 01.</p> <p><b>Poste créateur :</b> GestMat, Autres division de DRD, Les agences locales.</p> <p><b>Poste destinataire :</b> DGSI</p> <p><b>Rôle :</b> Suivre et garder la traçabilité du matériel reçu par la DGSI.</p>			
<b>Désignation du document</b>			
<b>Rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Nom de la division</li> <li>▪ Numéro de document</li> <li>▪ Lieu et date d'établissement de document</li> <li>▪ Nom et prénom de récepteur du matériel</li> <li>▪ Date de réception du matériel</li> <li>▪ Nom du matériel</li> <li>▪ Code barre/N° de série</li> <li>▪ Observation</li> <li>▪ Signature de responsable</li> </ul>	A AN AN  A Date AN AN A	04 10 20  20 08 10 10	Ex : DGSI Ex : 09/DGSI/2011 Ex : Tizi-Ouzou, le 28/03/11.  JJ/MM/AAAA
<b>Total</b>		<b>82</b>	

## Chapitre VI : Etude de l'existant

<b>Fiche d'analyse du document N° 03</b>			
<p><b>Code :</b> Alphanumérique</p> <p><b>Désignation :</b> Décharge externe matériel.</p> <p><b>Support :</b> Papier.</p> <p><b>Type :</b> externe.</p> <p><b>Nombre d'exemplaires :</b> 01.</p> <p><b>Poste créateur :</b> Autres division de DRD, Les agences locales.</p> <p><b>Poste destinataire :</b> DGSI</p> <p><b>Rôle :</b> Suivre et garder la traçabilité du matériel sortant de la DGSI.</p>			
<b>Désignation du document</b>			
<b>Rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Nom de la division ou d'agence</li> <li>▪ Numéro de document</li> <li>▪ Lieu et date d'établissement de document</li> <li>▪ Nom et prénom de récepteur du matériel</li> <li>▪ Date de réception du matériel</li> <li>▪ Nom du matériel</li> <li>▪ Code barre/N° de série</li> <li>▪ Observation</li> <li>▪ Signature de responsable</li> </ul>	A AN AN  A Date AN AN A	04 10 20  20 08 10 10	Ex : DEM Ex : 09/DGSI/2011 Ex : Tizi-Ouzou, le 28/03/11.  JJ/MM/AAAA
<b>Total</b>		<b>82</b>	

## Chapitre VI : Etude de l'existant

<b>Fiche d'analyse du document N° 04</b>			
<p><b>Code :</b> Alphanumérique</p> <p><b>Désignation :</b> Décharge interne réparation.</p> <p><b>Support :</b> Papier.</p> <p><b>Type :</b> interne.</p> <p><b>Nombre d'exemplaires :</b> 01.</p> <p><b>Poste créateur :</b> GestMat, Autres division de DRD, Les agences locales.</p> <p><b>Poste destinataire :</b> DGSI</p> <p><b>Rôle :</b> Suivre et garder la traçabilité des interventions et des réparations reçu par la DGSI.</p>			
<b>Désignation du document</b>			
<b>Rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Nom de la division</li> <li>▪ Numéro de document</li> <li>▪ Lieu et date d'établissement de document</li> <li>▪ Nom et prénom de récepteur de réparation.</li> <li>▪ Date de réception de la réparation</li> <li>▪ Nom du matériel</li> <li>▪ Code barre/N° de série</li> <li>▪ Observation</li> <li>▪ Signature de responsable</li> </ul>	A AN AN  A  Date AN AN A	04 10 20  20  08  10 10	Ex : DGSI Ex : 09/DGSI/2011 Ex : Tizi-Ouzou, le 28/03/11.   JJ/MM/AAAA
<b>Total</b>		<b>82</b>	

### Fiche d'analyse du document N° 05

**Code :** Alphanumérique

**Désignation :** Décharge externe réparation.

**Support :** Papier.

**Type :** externe.

**Nombre d'exemplaires :** 01.

**Poste créateur :** Autres division de DRD, Les agences locales.

**Poste destinataire :** DGSI

**Rôle :** Suivre et garder la traçabilité des interventions et des réparations reçu par les autres divisions de DRD et agences locales.

### Désignation du document

Rubrique	Type	Taille	Observation
<ul style="list-style-type: none"> <li>▪ Nom de la division</li> <li>▪ Numéro de document</li> <li>▪ Lieu et date d'établissement de document</li> <li>▪ Nom et prénom de récepteur de réparation.</li> <li>▪ Date de réception de la réparation</li> <li>▪ Nom du matériel</li> <li>▪ Code barre/N° de série</li> <li>▪ Observation</li> <li>▪ Signature de responsable</li> </ul>	A AN AN A Date AN AN A	04 10 20 20 08 10 10 10	Ex : DGSI Ex : 09/DGSI/2011 Ex : Tizi-Ouzou, le 28/03/11.  JJ/MM/AAAA
<b>Total</b>		<b>82</b>	

## Chapitre VI : Etude de l'existant

<b>Fiche d'analyse du document N° 06</b>			
<p><b>Code :</b> Alphanumérique</p> <p><b>Désignation :</b> Rapport d'intervention.</p> <p><b>Support :</b> Papier.</p> <p><b>Type :</b> externe et interne.</p> <p><b>Nombre d'exemplaires :</b> 01.</p> <p><b>Poste créateur :</b> DGSI</p> <p><b>Poste destinataire :</b> DGSI</p> <p><b>Rôle :</b> Suivre et garder la traçabilité des interventions et des réparations concernant la DGSI, les autres divisions de DRD et agences locales.</p>			
<b>Désignation du document</b>			
<b>Rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Nom de la division qui établit le rapport.</li> <li>▪ Numéro de document</li> <li>▪ Lieu et date d'établissement de document.</li> <li>▪ Nom de la structure.</li> <li>▪ Heure de déplacement</li> <li>▪ Durée.</li> <li>▪ Objet d'intervention</li> <li>▪ Equipement ou autre</li> <li>▪ Observation</li> <li>▪ Problème réglé</li> <li>▪ Si non pourquoi</li> <li>▪ Signature de responsable de structure.</li> </ul>	<p>A</p> <p>AN</p> <p>AN</p> <p>A</p> <p>AN</p> <p>AN</p> <p>A</p> <p>A</p> <p>A</p> <p>A</p> <p>A</p>	<p>04</p> <p>12</p> <p>20</p> <p>04</p> <p>05</p> <p>08</p> <p>03</p>	<p>Ex : DGSI</p> <p>Ex : /DGSI/DD/TO/2011</p> <p>Ex : Tizi-Ouzou, le 28/03/11.</p> <p>Ex : 11h :15</p> <p>Ex : Oui</p>
<b>Total</b>		<b>56</b>	



## Chapitre VI : Etude de l'existant

### Fiche d'analyse du document N° 07

**Code :** Alphanumérique

**Désignation :** Ordre de mission.

**Support :** Papier.

**Type :** externe.

**Nombre d'exemplaires :** 01.

**Poste créateur :** DGSI

**Poste destinataire :** DGSI

**Rôle :** Suivre et garder la traçabilité des interventions et des réparations concernant la DGSI, les autres divisions de DRD et agences locales.

### Désignation du document

Rubrique	Type	Taille	Observation
<ul style="list-style-type: none"><li>▪ Nom de la division qui établit le rapport.</li><li>▪</li></ul>	A	04	Ex : DGSI
<b>Total</b>		<b>56</b>	

<b>Fiche d'analyse du document N° 08</b>			
<p><b>Code :</b> Alphanumérique</p> <p><b>Désignation :</b> Certificat de réforme.</p> <p><b>Support :</b> Papier.</p> <p><b>Type :</b> externe.</p> <p><b>Nombre d'exemplaires :</b> 01.</p> <p><b>Poste créateur :</b> DGSI</p> <p><b>Poste destinataire :</b> DGSI</p> <p><b>Rôle :</b> Suivre et garder la traçabilité des interventions et des réparations concernant la DGSI, les autres divisions de DRD et agences locales.</p>			
<b>Désignation du document</b>			
<b>Rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Nom de la division qui établit le rapport.</li> <li>▪ </li> </ul>	A	04	Ex : DGSI
<b>Total</b>		<b>56</b>	

## Chapitre VI : Etude de l'existant

### Fiche d'analyse du document N° 09

**Code :** Alphanumérique

**Désignation :** PV de constat de réforme.

**Support :** Papier.

**Type :** externe.

**Nombre d'exemplaires :** 01.

**Poste créateur :** DGSI

**Poste destinataire :** DGSI

**Rôle :** Suivre et garder la traçabilité des interventions et des réparations concernant la DGSI, les autres divisions de DRD et agences locales.

### Désignation du document

Rubrique	Type	Taille	Observation
<ul style="list-style-type: none"><li>▪ Nom de la division qui établit le rapport.</li><li>▪</li></ul>	A	04	Ex : DGSI
<b>Total</b>		<b>56</b>	

## II. 4. Etude des fichiers :

➤ **Définition d'un fichier :**

Un fichier est un ensemble d'informations concernant un ensemble d'individus ou d'objets.

<b>Fiche d'analyse du fichier N° 1</b>			
<b>Identifications</b>			
<b>Désignation :</b> Registre gestion de matériel. <b>Support :</b> Papier. <b>Poste utilisateur :</b> GestParcInfo. <b>Rôle :</b> Suivre le matériel arrivé et sorti.			
<b>Les opérations réalisées sur le fichier</b>			
<b>Type d'opération</b>		<b>Période</b>	
-Consultation. -Mise à jour		-Variable. -Aléatoire.	
<b>Description des articles</b>			
<b>Désignation rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Code à barre.</li> <li>▪ Structure.</li> <li>▪ Service.</li> <li>▪ Désignation.</li> <li>▪ Marque.</li> <li>▪ Type.</li> <li>▪ Modèle.</li>   <li>▪ Numéro de série</li> <li>▪ Etat</li> <li>▪ Nom net Bios</li> <li>▪ Adresse IP</li> <li>▪ Domaine</li> <li>▪ Année d'acquisition</li> <li>▪ Est ondulé ?</li> <li>▪ Remarque</li> </ul>	N AN A AN AN AN AN  AN A AN  AN A A	10          08	 Ex : interne Ex : Accueil Ex : Ecran Ex : Dell Ex : PC Ex : dimension 2400 Ex : Bon Ex : 448- attaches1  Ex : Oui/Non  Oui/Non Ex : Avec modem

## Fiche d'analyse du fichier N° 2

Identifications	

<p><b>Désignation :</b> Registre matériel à réformé.</p> <p><b>Support :</b> Papier.</p> <p><b>Poste utilisateur :</b> GestParcInfo.</p> <p><b>Rôle :</b> Faire des recensements.</p>
---

**Rôle :** Faire des recensements.

## Les opérations réalisées sur le fichier

Type d'opération	Période
-Consultation. -Mise à jour	-Variable. -Aléatoire.

Description des articles
--------------------------

Désignation rubrique	Type	Taille	Observation
<ul style="list-style-type: none"> <li>▪ Code à barre.</li> <li>▪ Structure.</li> <li>▪ Service.</li> <li>▪ Désignation.</li> <li>▪ Marque.</li> <li>▪ Type.</li> <li>▪ Modèle.</li> </ul>	N AN A AN AN AN AN	10	Ex : interne Ex : Accueil Ex : Ecran Ex : Dell Ex : PC Ex : dimension 2400
<ul style="list-style-type: none"> <li>▪ Numéro de série</li> <li>▪ Etat</li> <li>▪ Nom net Bios</li> <li>▪ Adresse IP</li> <li>▪ Domaine</li> <li>▪ Année d'acquisition</li> <li>▪ Est ondulé ?</li> <li>▪ Remarque</li> </ul>	AN A AN  AN A A	08	Ex : Bon Ex : 448-attaches l Ex : Oui/Non Oui/Non Ex : Avec modem

<b>Fiche d'analyse du fichier N° 3</b>			
<b>Identifications</b>			
<p><b>Désignation :</b> Fichier réformé.</p> <p><b>Support :</b> Papier.</p> <p><b>Poste utilisateur :</b> GestParcInfo.</p> <p><b>Rôle :</b> Faire des recensements.</p>			
<b>Les opérations réalisées sur le fichier</b>			
<b>Type d'opération</b>		<b>Période</b>	
-Consultation. -Mise à jour		-Variable. -Aléatoire.	
<b>Description des articles</b>			
<b>Désignation rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Structure.</li> <li>▪ Service.</li> <li>▪ Désignation.</li> <li>▪ Marque.</li> <li>▪ Type.</li> <li>▪ Modèle.</li>   <li>▪ Numéro de série</li> <li>▪ Rôle de réformé</li> </ul>	AN A AN AN AN AN  A		Ex : interne Ex : Accueil Ex : Ecran Ex : Dell Ex : PC Ex : dimension 2400

## Chapitre VI : Etude de l'existant

<b>Fiche d'analyse du fichier N° 4</b>			
<b>Identifications</b>			
<b>Désignation :</b> Registre réparation. <b>Support :</b> Papier. <b>Poste utilisateur :</b> GestParcInfo. <b>Rôle :</b> Traçabilité et suivi.			
<b>Les opérations réalisées sur le fichier</b>			
<b>Type d'opération</b>	<b>Période</b>		
-Consultation. -Mise à jour	-Variable. -Aléatoire.		
<b>Description des articles</b>			
<b>Désignation rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Code à barre.</li> <li>▪ Service.</li> <li>▪ Désignation.</li> <li>▪ Marque.</li> <li>▪ Type.</li> <li>▪ Modèle.</li>   <li>▪ Numéro de série</li> <li>▪ Etat</li> <li>▪ Remarque</li> <li>▪ Date d'entrée</li> <li>▪ Date sortie réparation</li> <li>▪ Date de retour réparation</li> <li>▪ Date de sortie</li>   <li>▪ Réparateur</li> <li>▪ Réparé ?</li> <li>▪ Localisation</li>   <li>▪ Diagnostic</li> </ul>	N A AN AN AN AN  AN A A Date Date Date Date  A A A  A	10        08 08 08 08	Ex : Accueil Ex : Ecran Ex : Dell Ex : PC Ex : dimension 2400  Ex : Bon Ex : retourné JJ/MM/AAAA JJ/MM/AAAA JJ/MM/AAAA JJ/MM/AAAA  Oui/Non Pour traçabilité (DGSI) Ex : problème de tête.

<b>Fiche d'analyse du fichier N° 5</b>			
<b>Identifications</b>			
<b>Désignation :</b> Registre papier arrivé. <b>Support :</b> Papier. <b>Poste utilisateur :</b> GestStockPap. <b>Rôle :</b> Traçabilité et suivi.			
<b>Les opérations réalisées sur le fichier</b>			
<b>Type d'opération</b>	<b>Période</b>		
-Consultation. -Mise à jour	-Variable. -Aléatoire.		
<b>Description des articles</b>			
<b>Désignation rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Date d'arrivé</li> <li>▪ Date de correspondance</li> <li>▪ Numéro de correspondance</li> <li>▪ Expéditeur</li> <li>▪ Objet</li>   <li>▪ Date de la réponse</li> <li>▪ Numéro de la réponse</li> </ul>	Date Date AN A AN  Date AN	08 08     08	JJ/MM/AAAA JJ/MM/AAAA  Ex : 20 Rames 3exp JJ/MM/AAAA



## Chapitre VI : Etude de l'existant

<b>Fiche d'analyse du fichier N° 6</b>			
<b>Identifications</b>			
<b>Désignation :</b> Registre papier sortie. <b>Support :</b> Papier. <b>Poste utilisateur :</b> GestStockPap. <b>Rôle :</b> Traçabilité et suivi.			
<b>Les opérations réalisées sur le fichier</b>			
<b>Type d'opération</b>	<b>Période</b>		
-Consultation. -Mise à jour	-Variable. -Aléatoire.		
<b>Description des articles</b>			
<b>Désignation rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Date de départ</li> <li>▪ Numéro de correspondance</li> <li>▪ Destinataire</li> <li>▪ Objet</li>   <li>▪ Date de la réponse</li> <li>▪ Numéro de la réponse</li> </ul>	Date AN A AN  Date AN	08      08	JJ/MM/AAAA  Ex : Tizi genniff Ex : 2 Rames 2exp JJ/MM/AAAA

## Chapitre VI : Etude de l'existant

<b>Fiche d'analyse du fichier N° 7</b>			
<b>Identifications</b>			
<b>Désignation :</b> Registre déplacement. <b>Support :</b> Papier. <b>Poste utilisateur :</b> GestParcInfo. <b>Rôle :</b> Traçabilité et suivi.			
<b>Les opérations réalisées sur le fichier</b>			
<b>Type d'opération</b>	<b>Période</b>		
-Consultation. -Mise à jour	-Variable. -Aléatoire.		
<b>Description des articles</b>			
<b>Désignation rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Numéro de déplacement</li> <li>▪ Nom d'agent déplacé</li> <li>▪ Destination</li> <li>▪ Mission</li>   <li>▪ Date de déplacement</li> <li>▪ Autorisé par</li> </ul>	N A A A  Date A	08     08	Ex : 13/06/004  Ex : LNI Ex : Maintenance informatique JJ/MM/AAAA

## Chapitre VI : Etude de l'existant

<b>Fiche d'analyse du fichier N° 8</b>			
<b>Identifications</b>			
<b>Désignation :</b> Registre suivi des connexions aux agences. <b>Support :</b> Papier. <b>Poste utilisateur :</b> GestParcInfo. <b>Rôle :</b> suivre les statistiques chaque mois.			
<b>Les opérations réalisées sur le fichier</b>			
<b>Type d'opération</b>	<b>Période</b>		
-Consultation. -Mise à jour	-Variable. -Aléatoire.		
<b>Description des articles</b>			
<b>Désignation rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<ul style="list-style-type: none"> <li>▪ Agence</li> <li>▪ Etat</li> <li>▪ Date et heure</li> <li>▪ Signalé ?</li>   <li>▪ Date rétablissement</li> <li>▪ Observation</li> </ul>	A A Date A  Date A	08	DEM  Ex : Oui/Non  JJ/MM/AAAA Ex :      coupure technique

### **III. 5. Etude des procédures:**

#### **III. 5.1. Définition d'une procédure :**



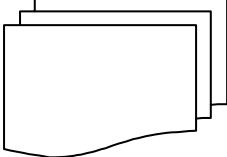
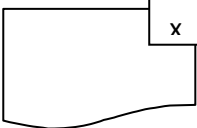

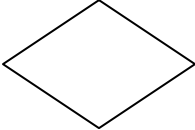
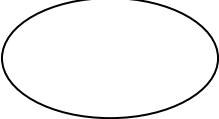
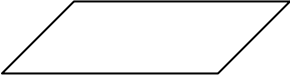
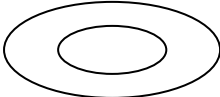
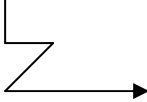

Une procédure est un enchaînement de phases concourant à une même finalité.

#### **III. 5.2. Liste des procédures :**

- Procédure approvisionnement papier
- Procédure transfert papier (Aux autres divisions de la DRD, et aux agences)
- Procédure de suivi des statistiques du matériel arrivé et sorti.
- Procédure de suivi des rapports d'intervention.
- Procédure de gestion et de suivre des documents entrés/sortis.
- Procédure de la gestion des demandes.

#### **III. 5.3. Notation graphique**

Le schéma ci-dessous montre les notations graphiques utilisées pour la modélisation des procédures

Symboles	Désignations
	Opération
	Document
	Dossier
	Document en x exemplaires
	Flux direct
	alternative
	Indication / Demande verbale
	Registre
	Jeton
	Appel par ligne téléphonique
	Chemise.

### II.5.4 Etude détaillée de chaque procédure de travail :

#### L'approvisionnement de papier

##### ▪ Diagramme des flux :

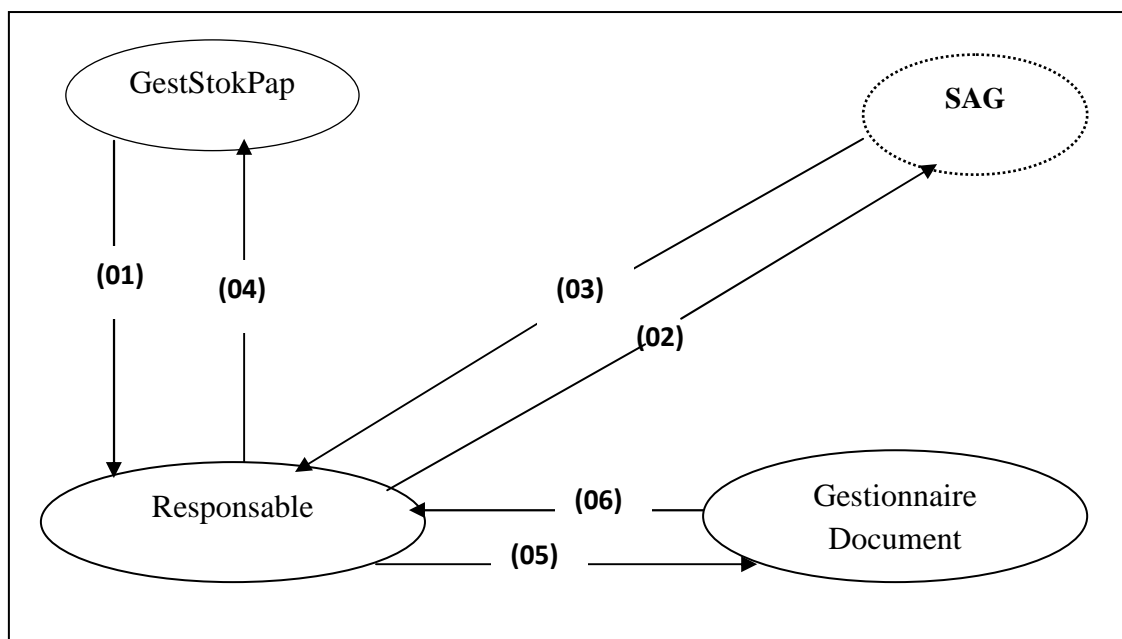
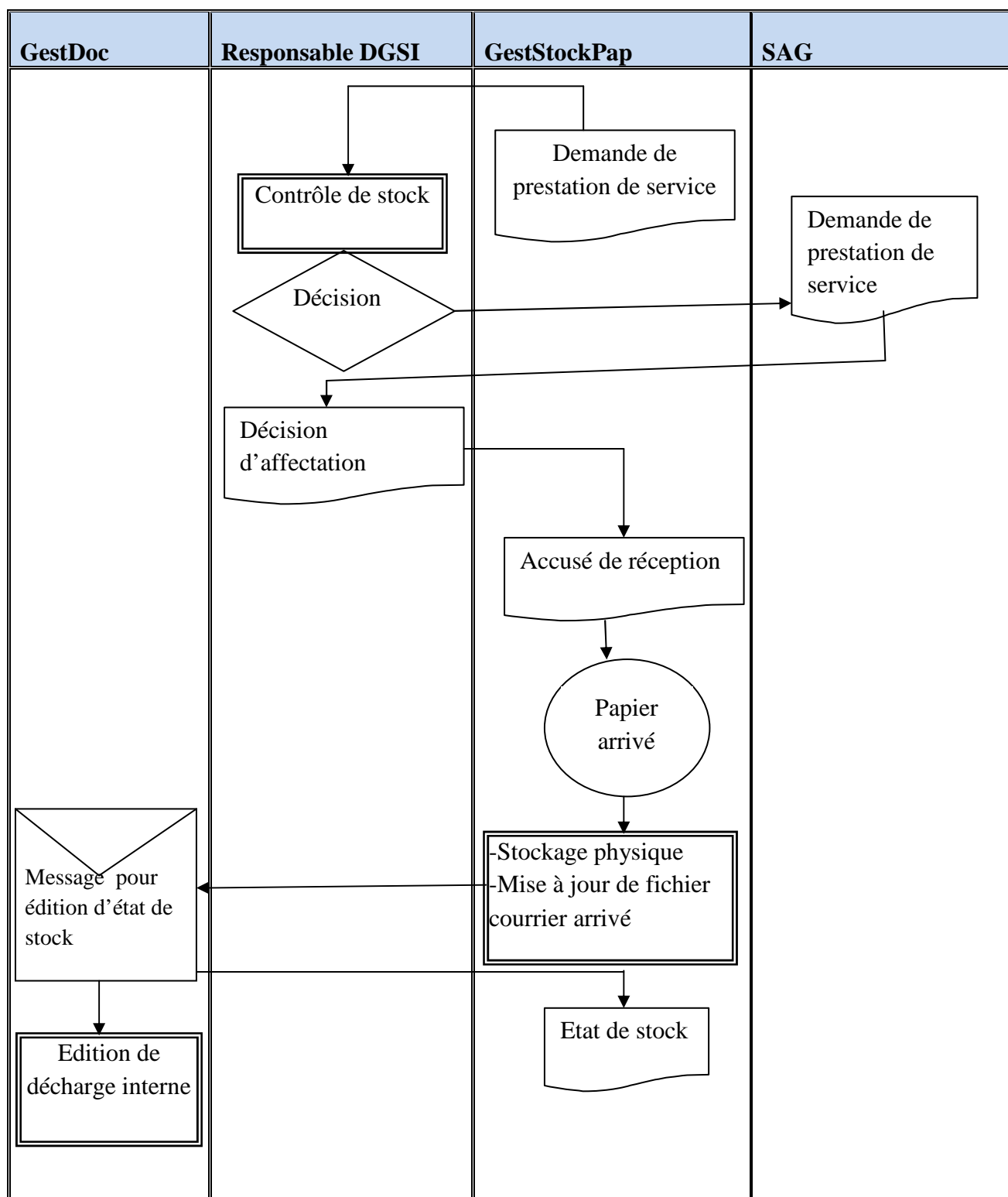


Figure 2.2.1 : Diagramme des flux « Approvisionnement papier »

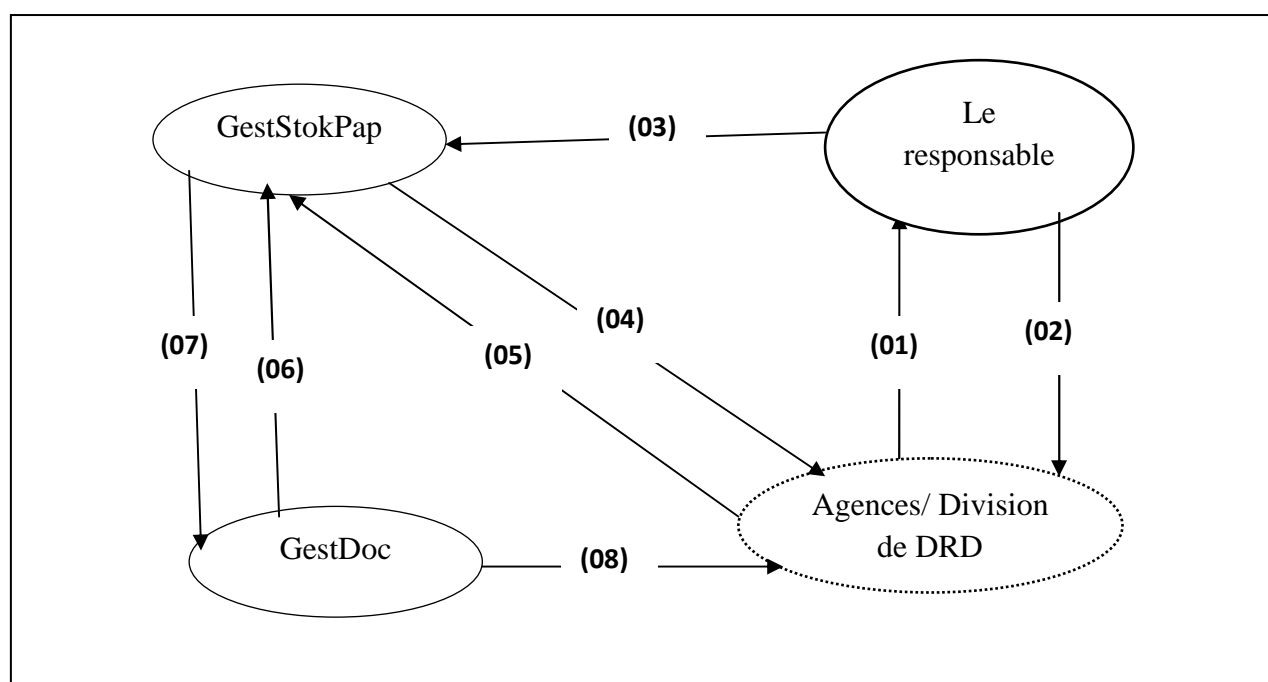
##### ▪ Description des flux :

N° Flux	Descriptions
01	Demande de prestation de service
02	Demande de prestation de service
03	Décision d'affectation
04	Accusé de réception
05	Demande d'édition d'état de stock
06	Etat de stock



Nom de l'opération	Description des tâches
Contrôle de stock	Dans cette opération le responsable de la DGSi vérifie et contrôle si le stock physique de papier.
Stock physique et mis à jour de fichier courrier arrivé	Durant cette opération le gestionnaire de stock papier stock et classe physiquement le papier arrivé, ensuite met à jour toutes les informations concernant la quantité de papier arrivé.

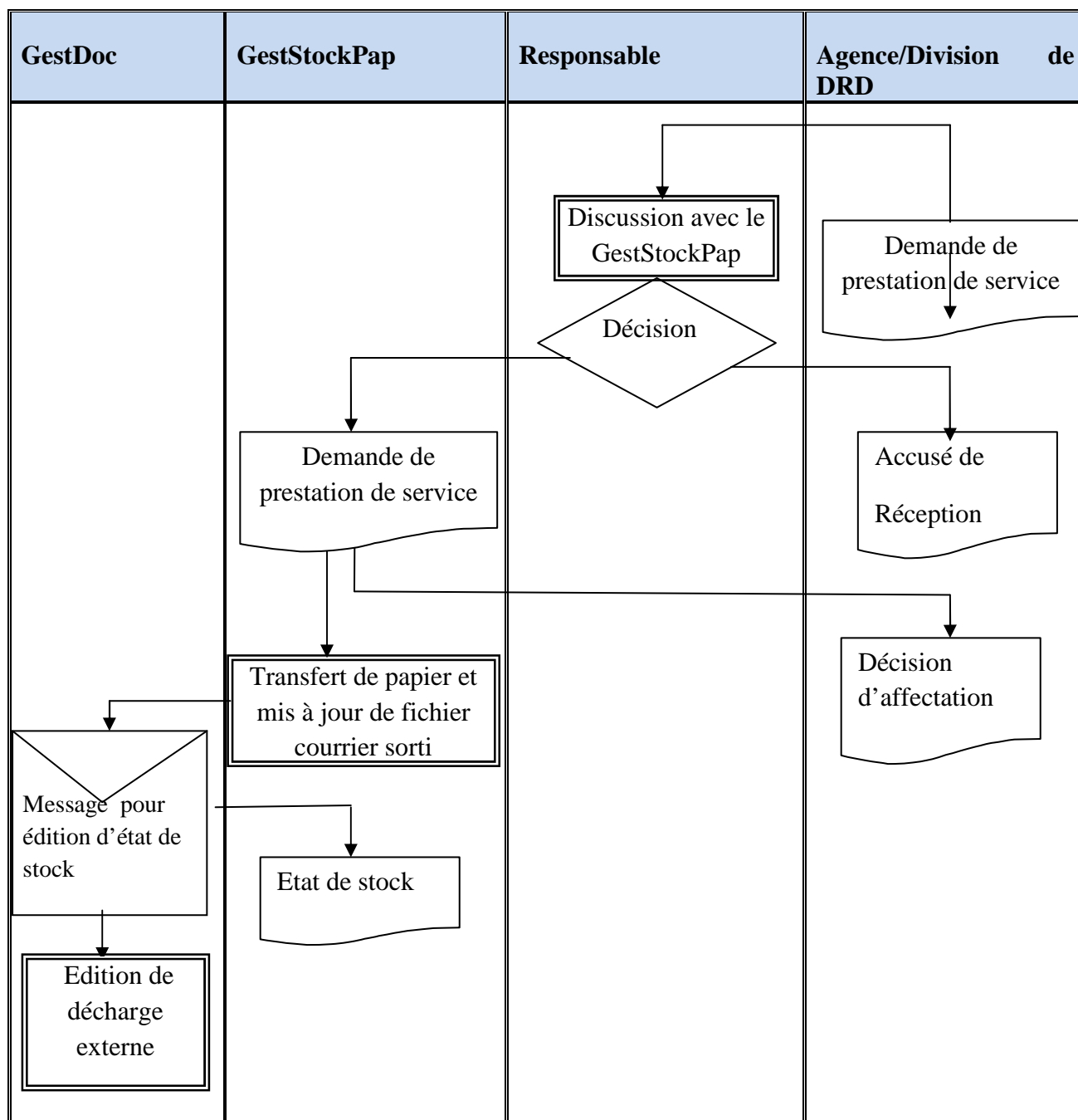
### Transfert papier



**Figure 2.2.1 : Diagramme des flux « Transfert papier »**



N° Flux	Descriptions
01	Demande de prestation de service
02	Accusé de réception
03	Demande de prestation de service
04	Décision d'affectation
05	Décharge externe
06	Demande d'état de stock
07	Etat de stock
08	Décharge externe



## Chapitre VI : Etude de l'existant

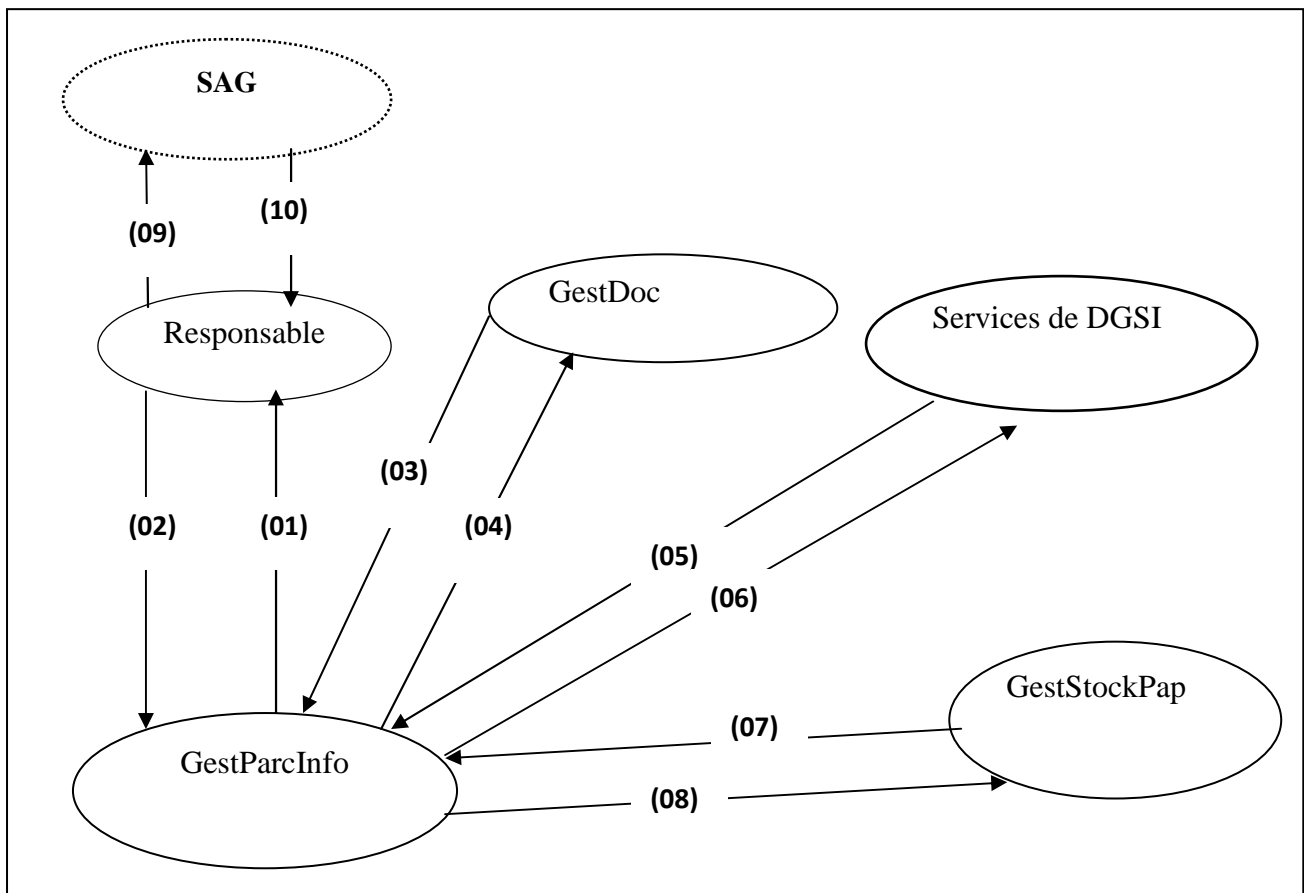
---

Nom de l'opération	Description des tâches
Discussion avec le GestStockPap	Dans cette opération le responsable de la DGSI et le gestionnaire de stock papier se discutent pour le transfert de papier.
Transfert de papier et mis à jour de fichier courrier sorti	Durant cette opération le gestionnaire de stock papier met à jour toutes les informations concernant la quantité de papier sorti, et ça après le transfert de papier.
Edition de décharge externe	Pendant cette opération le gestionnaire de documents édite la décharge externe concernant le transfert papier.

### Procédure de suivi des statistiques du matériel arrivé et sorti

#### ❖ Matériel arrivé à la DGSI :

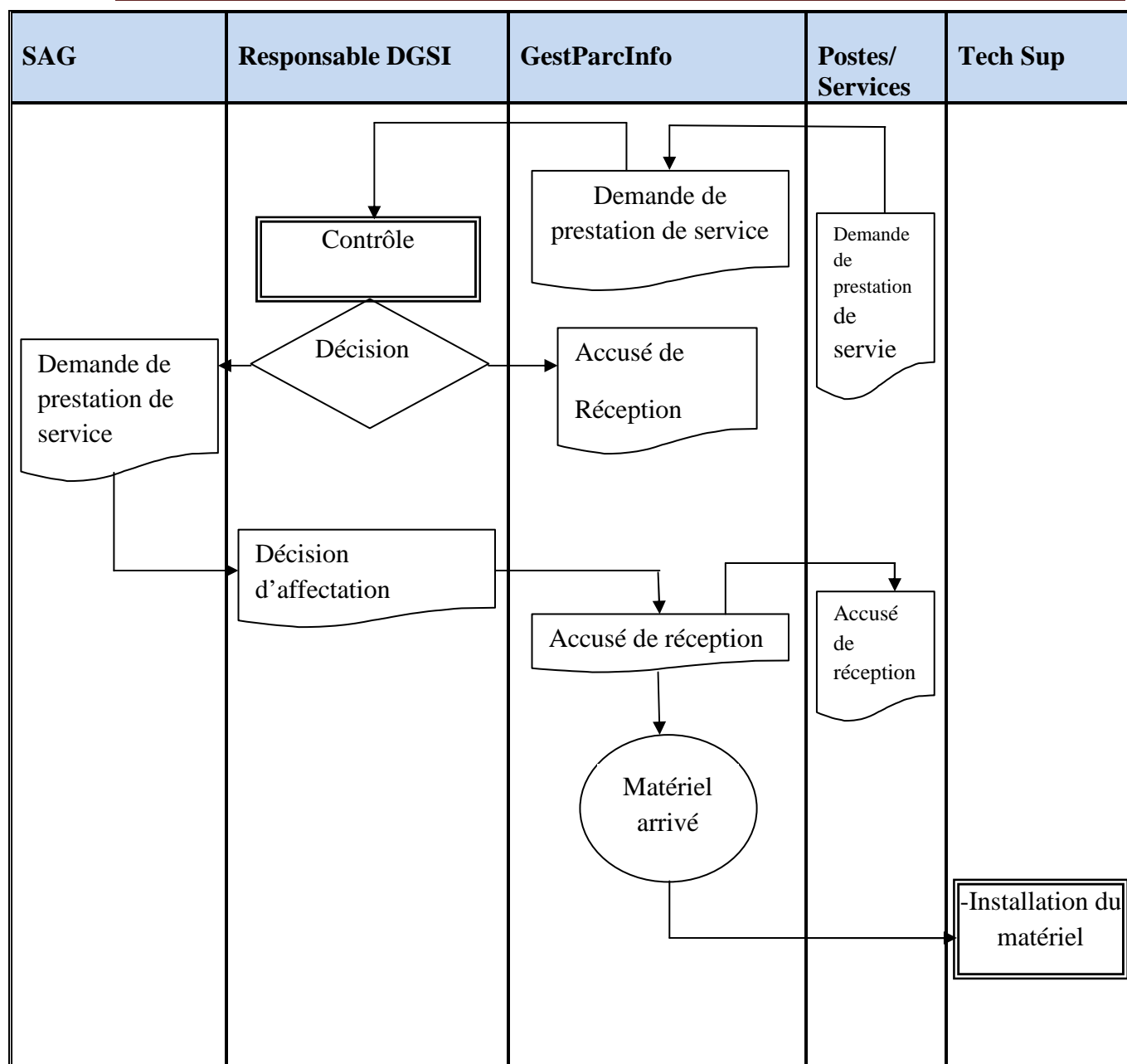
##### ▪ Diagramme des flux :



▪ **Description des flux :**

<b>N° Flux</b>	<b>Descriptions</b>
01	Demande de prestation de service
02	Accusé de réception
03	Demande de prestation de service
04	Accusé de réception
05	Demande de prestation de service
06	Accusé de réception
07	Demande de prestation de service
08	Accusé de réception
09	Demande de prestation de service
10	Décision d'affectation

## Chapitre VI : Etude de l'existant

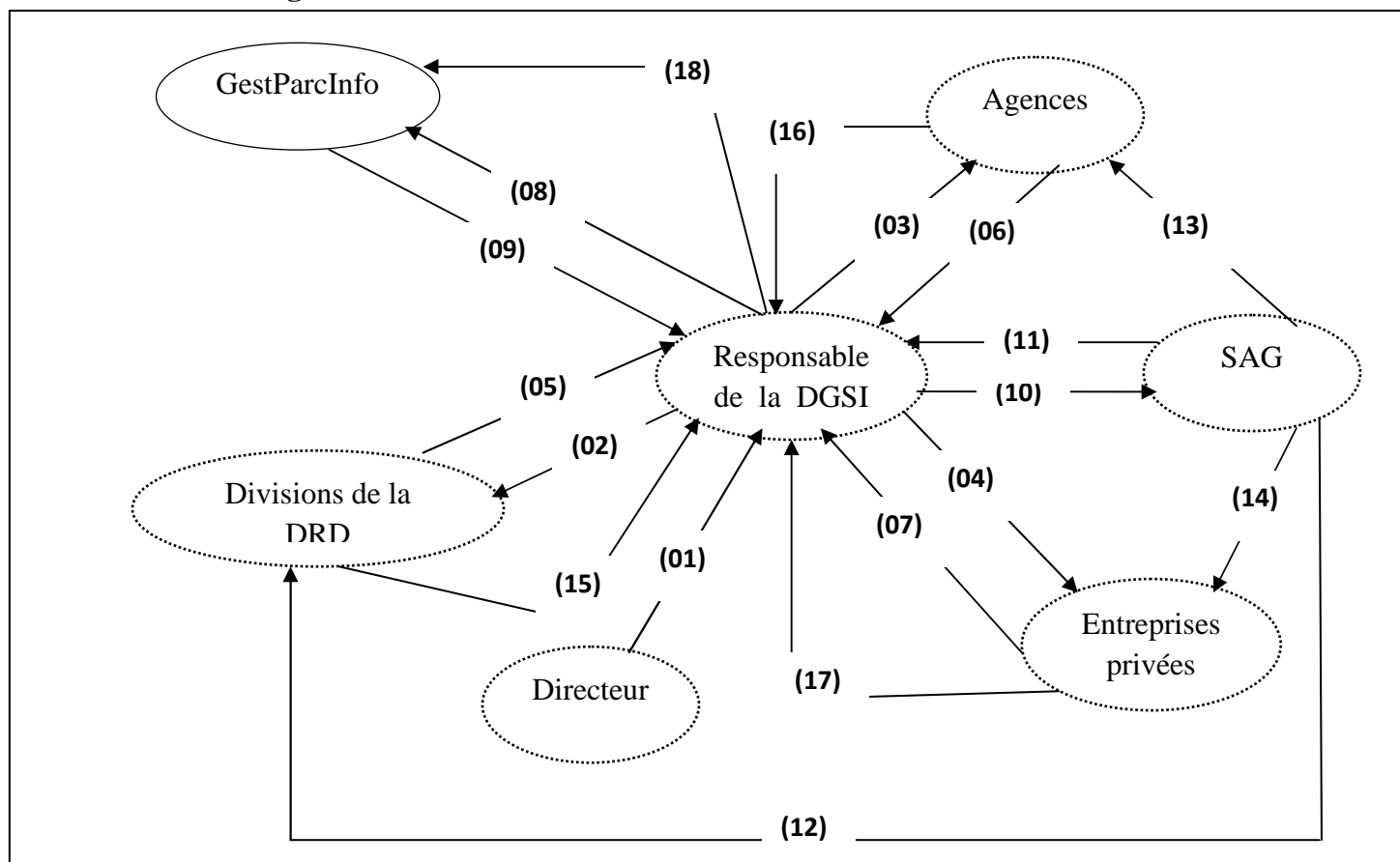


## Chapitre VI : Etude de l'existant

Nom de l'opération	Description des tâches
Contrôle	Durant cette opération le responsable de la DGSI vérifie la nécessité ou non d'avoir le matériel demandé
Installation du matériel	Dans cette opération le technicien supérieur de la DGSI procède à l'installation du matériel arrivé au niveau de service dont il a besoin.

### ❖ Matériel sorti de la DGSI :

#### ▪ Diagramme des flux :



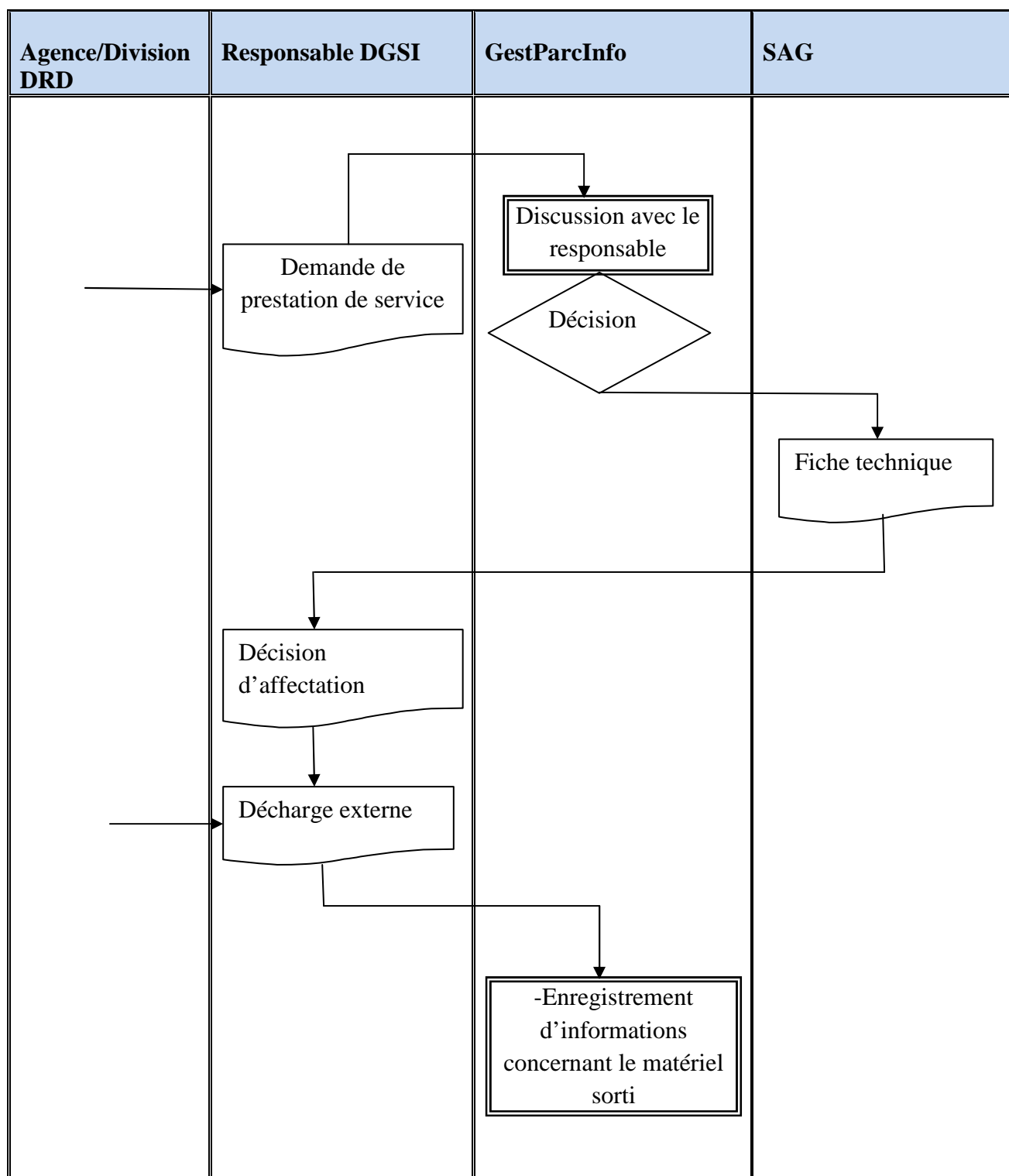
## Chapitre VI : Etude de l'existant

---

### ▪ Description des flux :

N° Flux	Descriptions
01	Accord de directeur
02	Appel d'offre
03	Appel d'offre
04	Appel d'offre
05	Demande de prestation de service
06	Demande de prestation de service
07	Demande de prestation de service
08	Information sur le matériel sorti
09	Fiche technique
10	Fiche technique
11	Décision d'affectation
12	Bon de livraison
13	Bon de livraison
14	Bon de livraison
15	Décharge externe
16	Décharge externe
17	Décharge externe
18	Décharge externe



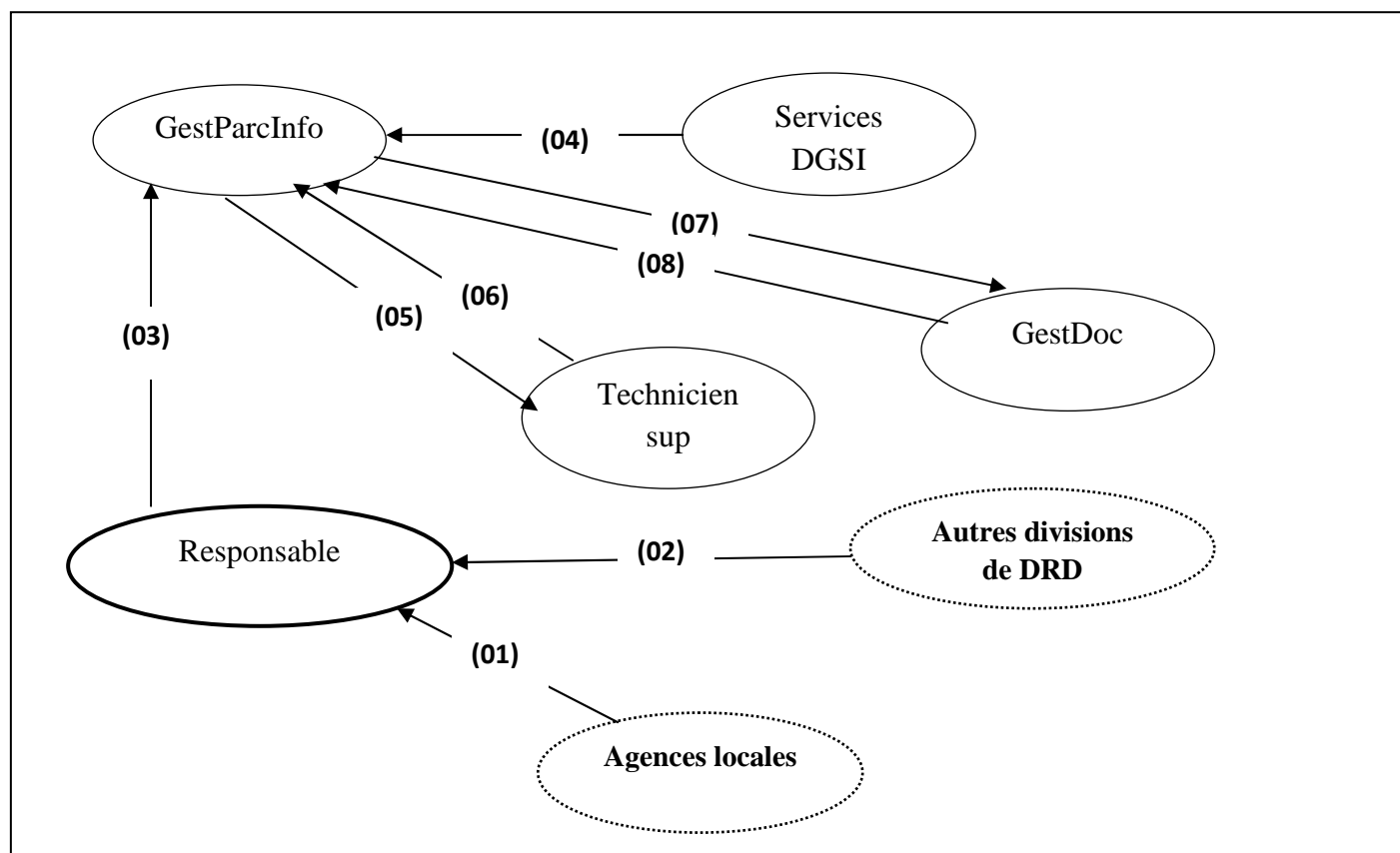


## Chapitre VI : Etude de l'existant

Nom de l'opération	Description des tâches
Discussion avec le responsable	Dans cette opération le responsable de la DGSI et le gestionnaire du matériel se discutent pour le transfert.
Enregistrement d'informations concernant le matériel sorti	Durant cette opération le gestionnaire de matériel enregistre toutes les informations concernant la quantité de matériel sorti.

### Procédure de suivi des rapports d'intervention

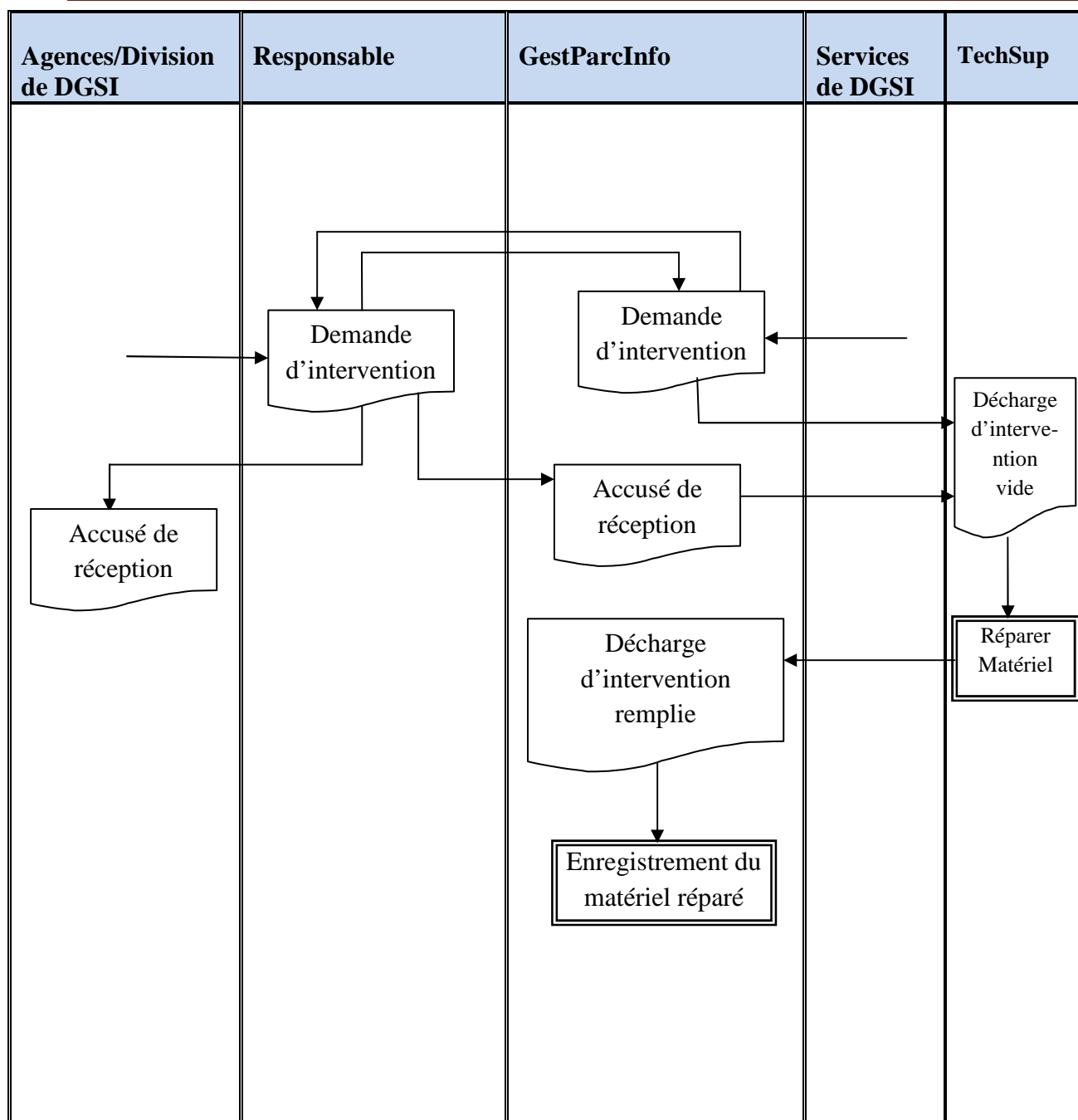
#### ▪ Diagramme des flux :



▪ Description des flux :

N° Flux	Descriptions
01	Demande d'intervention
02	Demande d'intervention
03	Demande d'intervention
04	Demande d'intervention
05	Décharge d'intervention vide
06	Décharge d'intervention remplie
07	Demande d'édition de rapport d'intervention
08	Rapport d'intervention

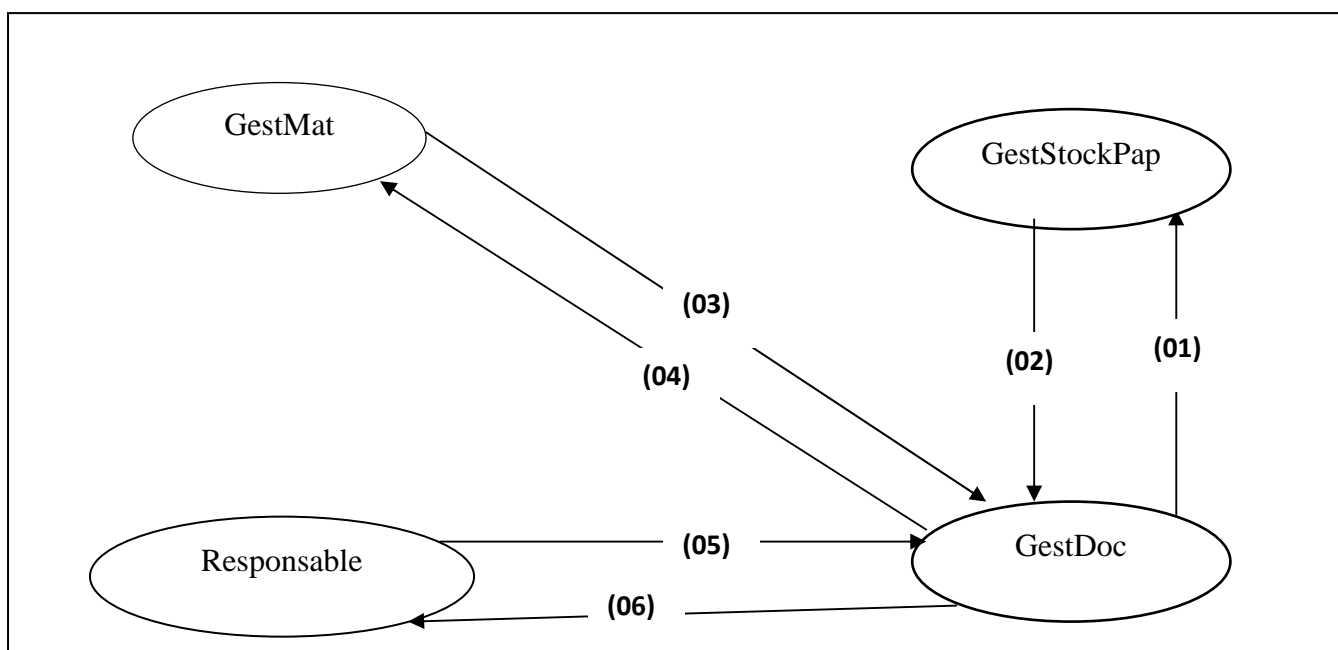
## Chapitre VI : Etude de l'existant



Nom de l'opération	Description des tâches
Réparer le matériel	Durant cette opération le technicien supérieur de la DGSI procède à la réparation de matériel qui est en panne
Enregistrement du matériel réparé	Dans cette opération le gestionnaire de matériel DGSI procède à l'enregistrement du matériel réparé au niveau de la DGSI, des divisions de DRD, et des agences locales.

### Procédure de gestion des documents

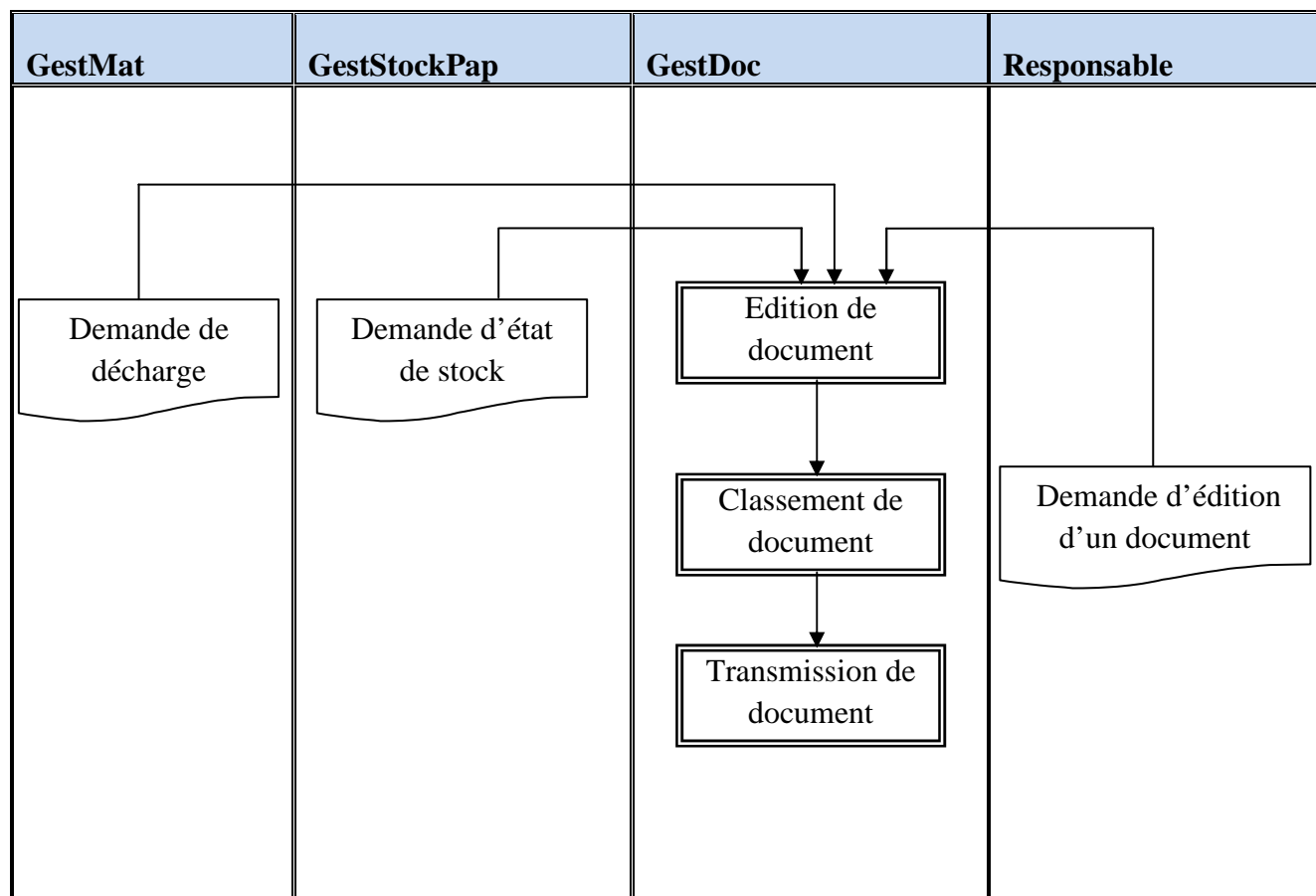
#### ▪ Diagramme des flux :



#### ▪ Description des flux :

## Chapitre VI : Etude de l'existant

N° Flux	Descriptions
01	Demande d'édition d'état de stock
02	L'état de stock
03	Demande d'édition des décharges interne ou externe
04	Décharge interne ou externe
05	Demande d'édition d'un document
06	Document édité



## Chapitre VI : Etude de l'existant

---

### ▪ Description des opérations :

Nom de l'opération	Description des tâches
Edition de document	Dans cette opération le gestionnaire de document DGSI édite le document demandé.
Classement de document	Dans cette opération le gestionnaire de document classe les différents documents édités et entrés.
Transmission de document	Transmettre le document au poste correspondant

### **Conclusion :**

Notre présent chapitre vient de se terminer avec une étude globale de l'existant d'organisme d'accueil, que se soit les postes de travail, les documents, les fichiers, et les procédures du travail.

Le suivant chapitre aura lieu d'analyser et de concevoir notre futur système à réaliser, en utilisant l'outil de conception UML.

# CHAITR VII :

**Analyse**

**et**

**Conception**



### Section I : Etude préliminaire

#### **I. Modélisation du contexte :**

La modélisation est la conception d'un modèle selon son objectif et les moyens utilisés, elle consiste à structurer et à représenter visuellement les activités de l'entreprise.

La modélisation de contexte de notre projet est au cœur même de la démarche d'analyse dynamique de notre organisme, elle met en évidence notre champ d'application et les acteurs interviennent.

Cette modélisation permet de formaliser le fonctionnement précis de la DGSI en utilisant un langage standard et aisément compréhensible UML.

La modélisation comprend le domaine de l'application, l'environnement du projet et identifier une solution viable pour le problème.

#### **I. 1. Les acteurs :**

Un acteur représente l'abstraction d'un rôle joué par des entités externe (utilisateurs, dispositif matériel ou autre système) qui interagissent directement avec le système

Dans notre système nous allons identifier les acteurs principaux suivant :

- 1. Le responsable :** Le responsable a pour mission le contrôle et la prise de décision pour les différentes affaires de la DGSI.
- 2. Le gestionnaire de stock papier :** Il s'occupe de l'approvisionnement et le transfert de papier à partir de la DGSI.
- 3. Le gestionnaire de parc informatique :** Chargé de suivre les statistiques du matériel arrivé et sorti de la DGSI ainsi que les interventions.
- 4. Le gestionnaire de documents :** Chargé d'établissement, d'enregistrement, et de transmission des documents nécessaires.

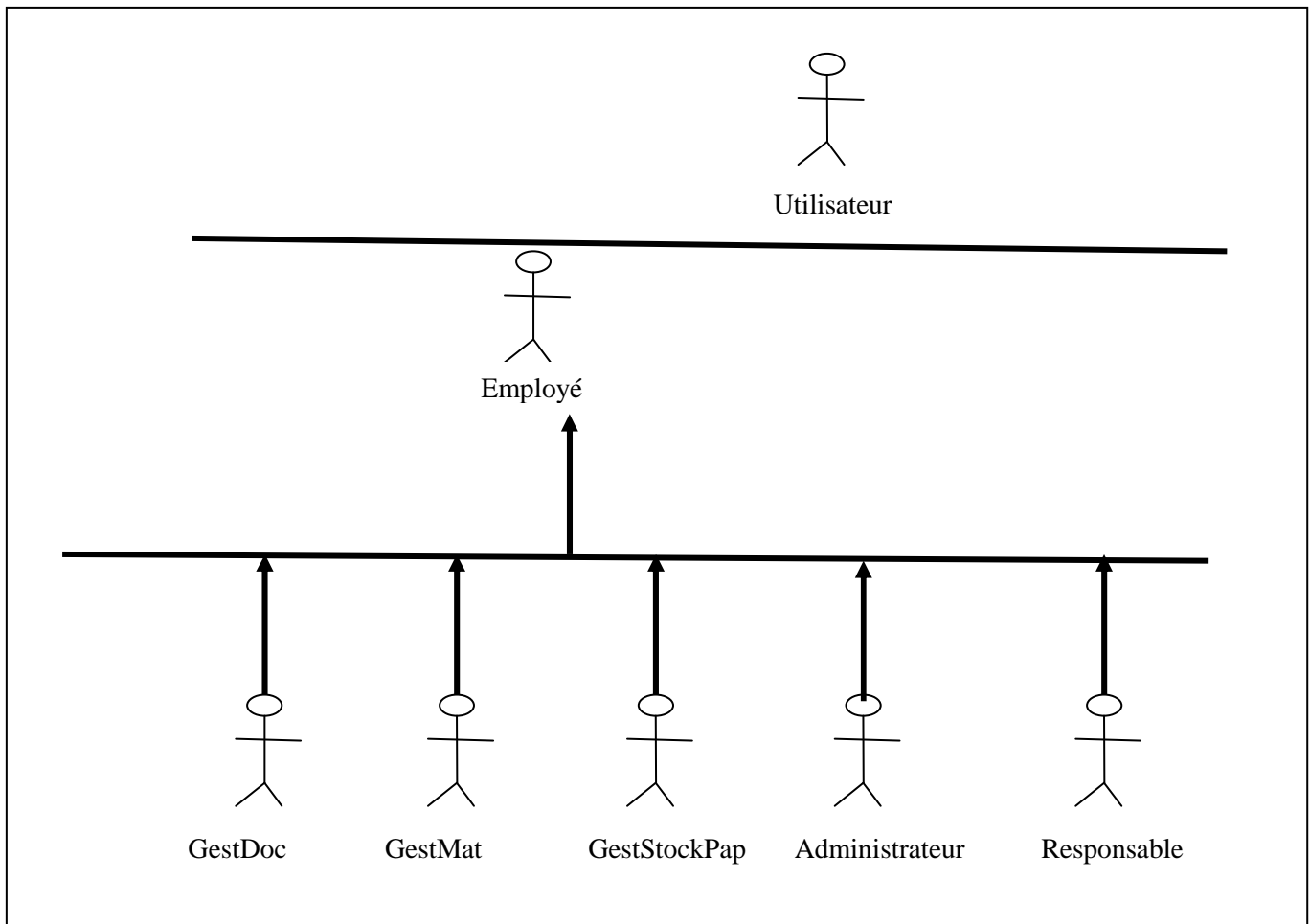


FIG1 : LES ACTEURS DE SYSTÈME

### I. 2. Les messages :

Un message représente la spécification d'une communication entre objets qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur.

Le système et les acteurs communiquent entre eux à travers des messages, alors chacun de ces derniers joue le rôle d'émetteur et récepteur à la fois.

D'où les messages de notre système sont les suivants :

- Le système reçoit :
  - La création des réponses des demandes (Réponses aux demandes de prestation de service)
  - La validation des réponses.
  - La création des documents.
  - La validation des documents.
  - La création d'approvisionnement papier.
  - La validation d'approvisionnement.
  - La création d'un transfert de papier.

- Validation d'un transfert.
  - Création d'une livraison matériel, sortie matériel, et interventions.
  - Validation de livraison matériel, de sortie matériel, et d'interventions
  - Demande des statistiques (Affichage).
  - La création, la suppression ou la mise à jour de droit d'accès.
- Le système émet:
1. Confirmation sur la saisie.
  2. Accusé de réception de la demande.
  3. Confirmation sur validation d'approvisionnement et stockage papier.
  4. Confirmation sur validation de transfert papier.
  5. Confirmation sur validation de livraison matériel, de sortie matériel, d'intervention.
  6. Edition des documents.
  7. Edition des statistiques (Affichage de ce qui est demandé par l'utilisateur).
  8. Confirmation sur la gestion de droit d'accès.

### I. 3. Le diagramme de contexte :

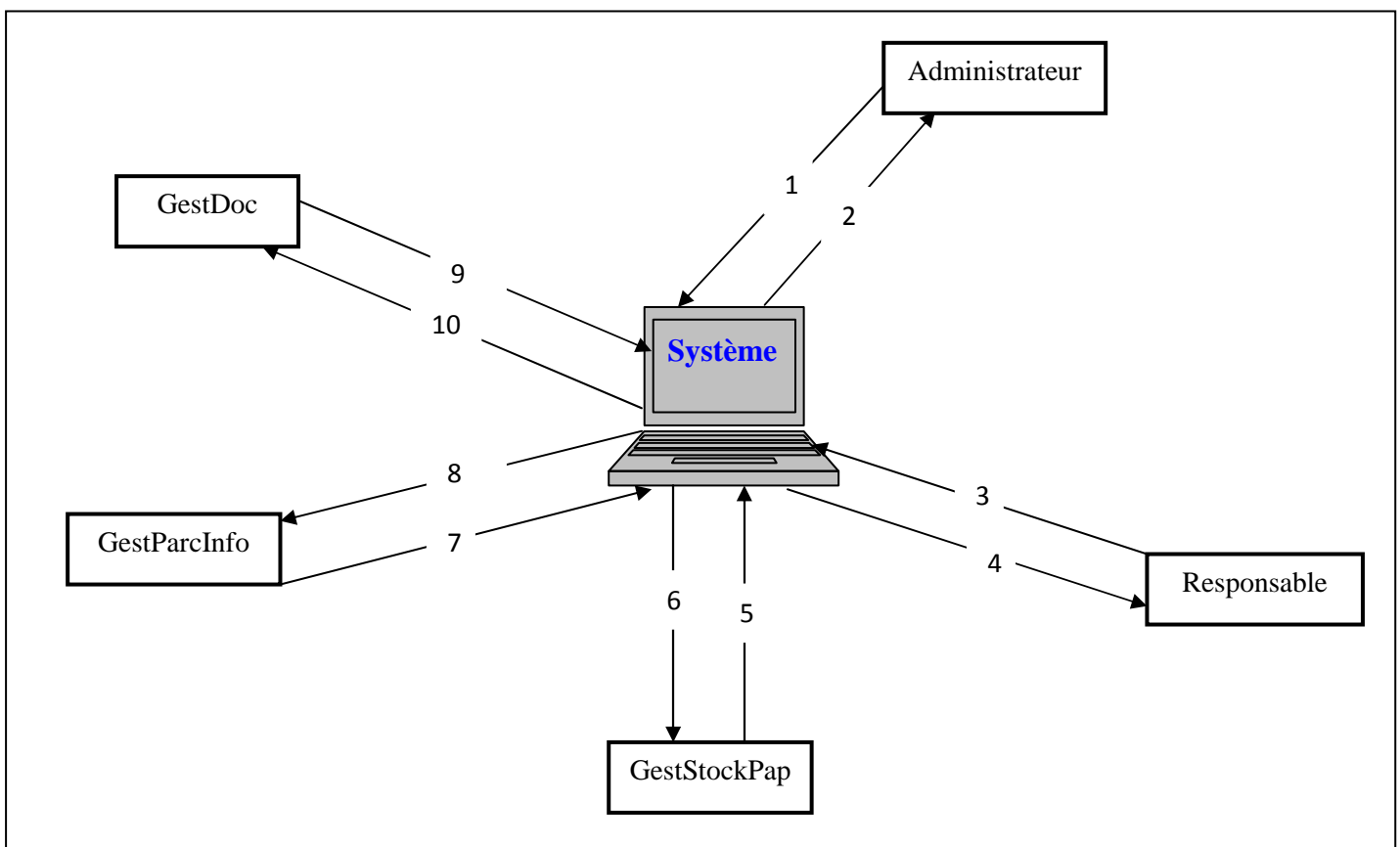


FIG2 : LE DIAGRAMME DE CONTEXTE

## Chapitre VII : Analyse et conception

### Les messages échangés :

N° message	L'émetteur	Le récepteur	Désignation
1	Administrateur	Le système	Création et modification des profils utilisateurs.
2	Le système	Administrateur	Des confirmations sur la gestion des profils utilisateurs (mot de passe).
3	Le responsable	Le système	Contrôle; validation des réponses des demandes et Demandes des statistiques (Affichage).
4	Le système	Le responsable	Confirmation de la validation et affichage des statistiques.
5	GestStockPap	Le système	Saisie d'approvisionnement, Transfert papier, et stockage après chaque transfert et approvisionnement.
6	Le système	GestStockPap	Confirmation sur la saisie, et sur la validation d'approvisionnement et de transfert.
7	GestParcInfo	Le système	Saisie de matériel arrivé et sorti et de toutes les interventions.
8	Le système	GestParcInfo	Confirmation sur la saisie et sur la validation.
9	GestDoc	Le système	Demande de création des documents.
10	Le système	GestDoc	Edition des documents.

## **II. Représentation des besoins exprimés par les utilisateurs :**

Première description des fonctionnalités du système, on représente les acteurs et leurs besoins. La description dynamique de notre système commencera par la description et le recensement des besoins des utilisateurs en décrivant l'interaction entre l'utilisateur et le système grâce au diagramme de cas d'utilisation.

### **II. 1. Identification des cas d'utilisation:**

## Chapitre VII : Analyse et conception

Un cas d'utilisation synthétise un ensemble de scénarios avec des traitements optionnels et des traitements d'erreurs. Un scénario est une instance de cas d'utilisation (une description de ce qui doit se passer). Le scénario nominal est le scénario le plus court, celui où tout se passe correctement.

Les traitements optionnels sont des extensions du scénario nominal. Ils peuvent apparaître dans le diagramme sous forme de traitements d'extension.

Mettre en évidence des parties de scénario qu'on peut isoler conduit aux inclusions.

Mettre en évidence des cas particuliers conduit à des spécialisations.

### II. 1.1. Cas d'utilisation « Gestion de droit d'accès »:

#### II. 1.1.1. Description textuelle de cas d'utilisation:

1. Demande de création ou de modification d'un compte utilisateur.
2. Afficher la liste des profils.
3. Choisir le profil de compte à créer ou à modifier.
4. Afficher la liste des fonctionnalités.
5. Choisir les fonctionnalités correspondantes.
6. Validation de l'opération de création ou de modification.

#### II. 1.1.2. Description de cas d'utilisation à l'aide du tableau:

Cas	<b>Administration des utilisateurs (Gestion de droit d'accès)</b>
Acteur primaire	Administrateur de système
Acteurs secondaires	Aucun
Pré-condition	Il faut que soit une mise à jour nécessaire concernant les comptes utilisateurs.
Résultats	Création de nouveaux comptes utilisateurs, et spécification de leurs profils ; mis à jour de comptes utilisateurs.
Description de scénario	-Un utilisateur se recrute, se quitte ou décide de modifier ses coordonnées. -Mise à jour des coordonnées. -Validation de mise à jour.
Exception	Aucune

#### II. 1.1.3. Description graphique de cas d'utilisation : « Gestion de droit d'accès »

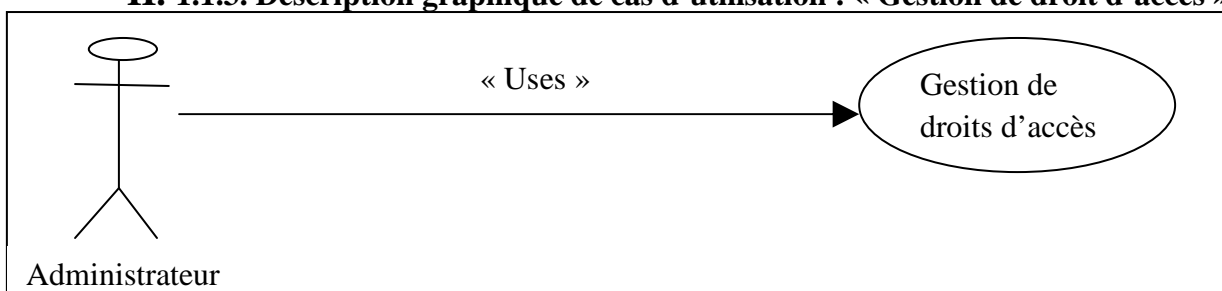


FIG1.3 : CAS D'UTILISATION « GESTION DE DROIT D'ACCÈS »

### II. 1.1.4. Scénario du cas d'utilisation « Gestion des droits d'accès »

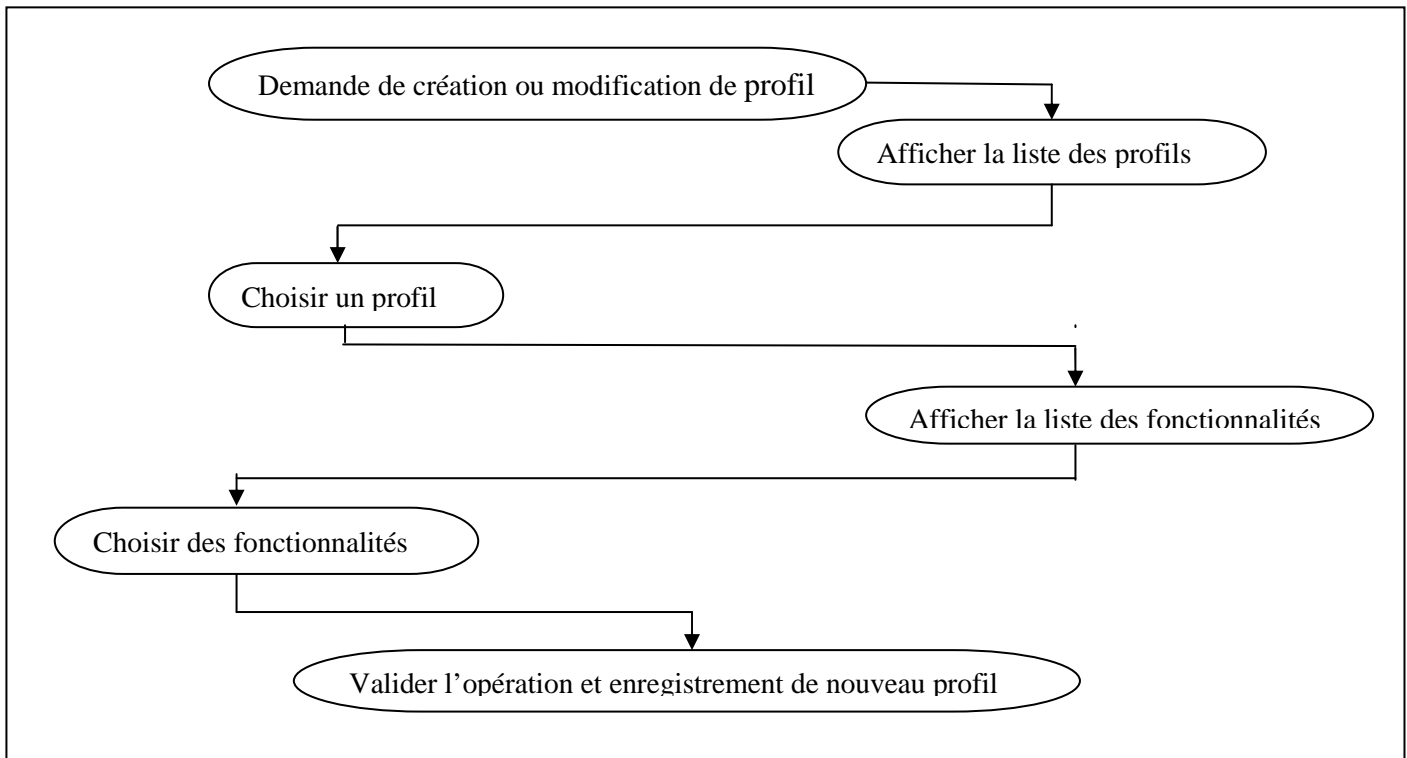


FIG. : 1.4. SCÉNARIO DU CAS D'UTILISATION « GESTION DES DROITS D'ACCÈS »

### II. 1.2. Cas d'utilisation « Identification des utilisateurs »:

#### II. 1.2.1. Description textuelle des cas d'utilisation:

1. Un utilisateur qui veut accéder à son espace s'identifie par son login et son password.
2. Le système vérifie l'existence de cet utilisateur.
3. L'utilisateur se connecte ; est alors dans son espace.

#### II. 1.2.2. Description de cas d'utilisation à l'aide du tableau:

Cas	Identification des utilisateurs
Acteurs	Tous les utilisateurs
Pré-condition	L'utilisateur doit être enregistré dans le système
Résultats	Authentification d'un utilisateur se connectant au système ; Présentation de l'interface utilisateur.
Description de scénario	-Saisie de login et mot de passe pour accéder à l'espace utilisateur.
Exception	Chaque utilisateur a son propre mot de passe et son propre login

### II. 1.2.3. Description graphique de cas d'utilisation : « Identification des utilisateurs »

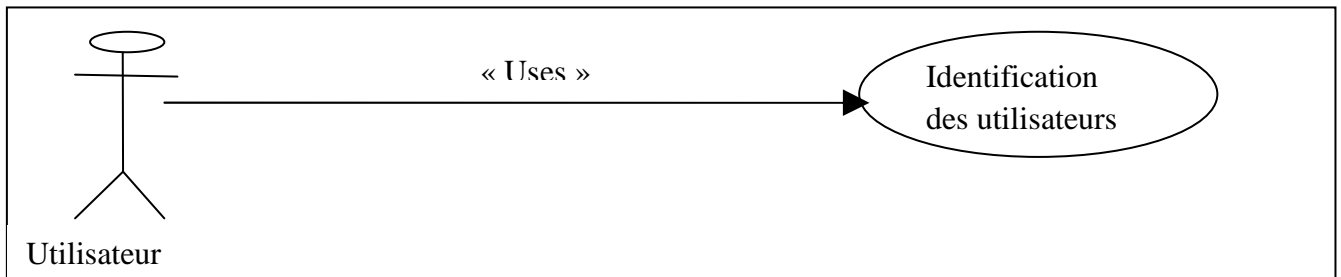


FIG2.3 : CAS D'UTILISATION « IDENTIFICATION DES UTILISATEURS »

### II. 1.2.4. Scénario du cas d'utilisation « Identification des utilisateurs »

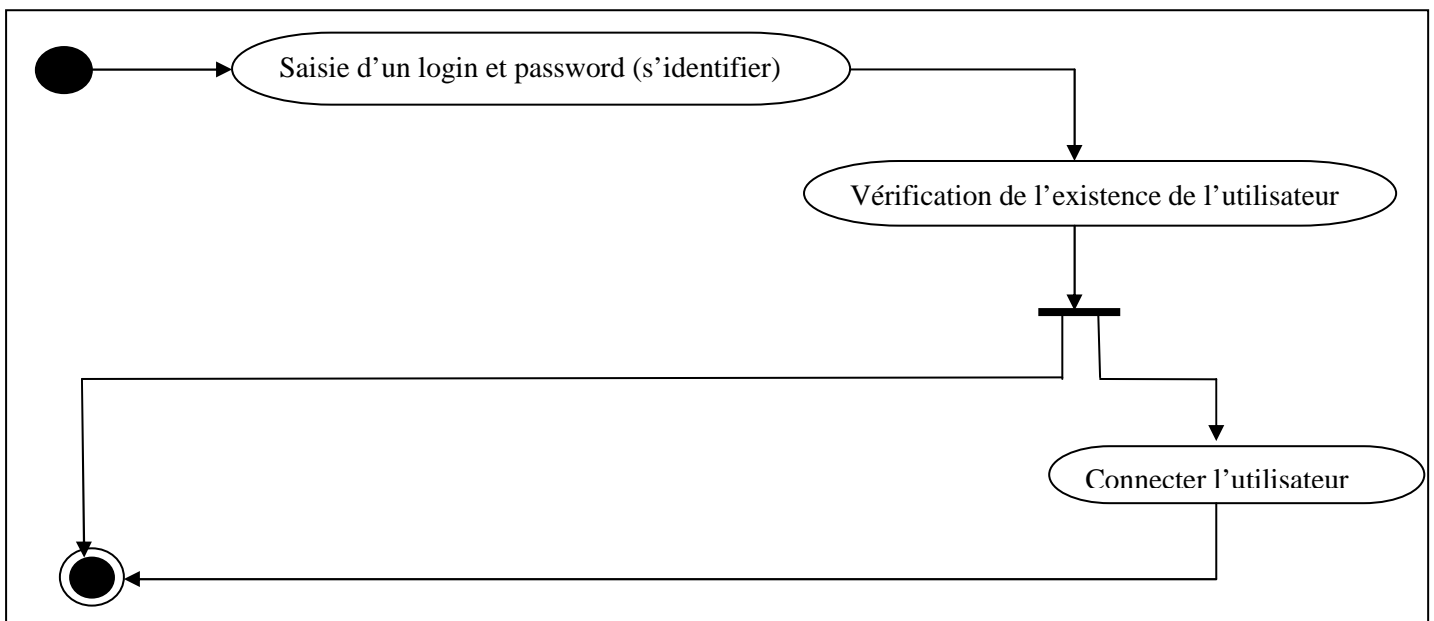


FIG.2 4. SCÉNARIO DU CAS D'UTILISATION « IDENTIFICATION DES UTILISATEURS »

## II. 1.3. Cas d'utilisation « Approvisionnement papier » :

### II. 1.3.1. Description textuelle des cas d'utilisation:

1. Le gestionnaire de stock papier fait une demande de prestation de service au SAG, et doit être passé tout d'abord par le responsable.
2. Le responsable de la DGSI consulte l'état du stock pour se confirmer, et envoie l'accusé de réception au GestStockPap, ainsi que la demande de prestation de service au SAG.
3. Le SAG répond par une décision d'affectation au responsable qui va la transmettre au GestStockPap.
4. Le papier est alors arrivé, et le GestStockPap procède au stockage physique de papier, saisi l'approvisionnement, met à jour le registre

## Chapitre VII : Analyse et conception

courrier arrivé ainsi que l'opération de stockage papier après l'approvisionnement.

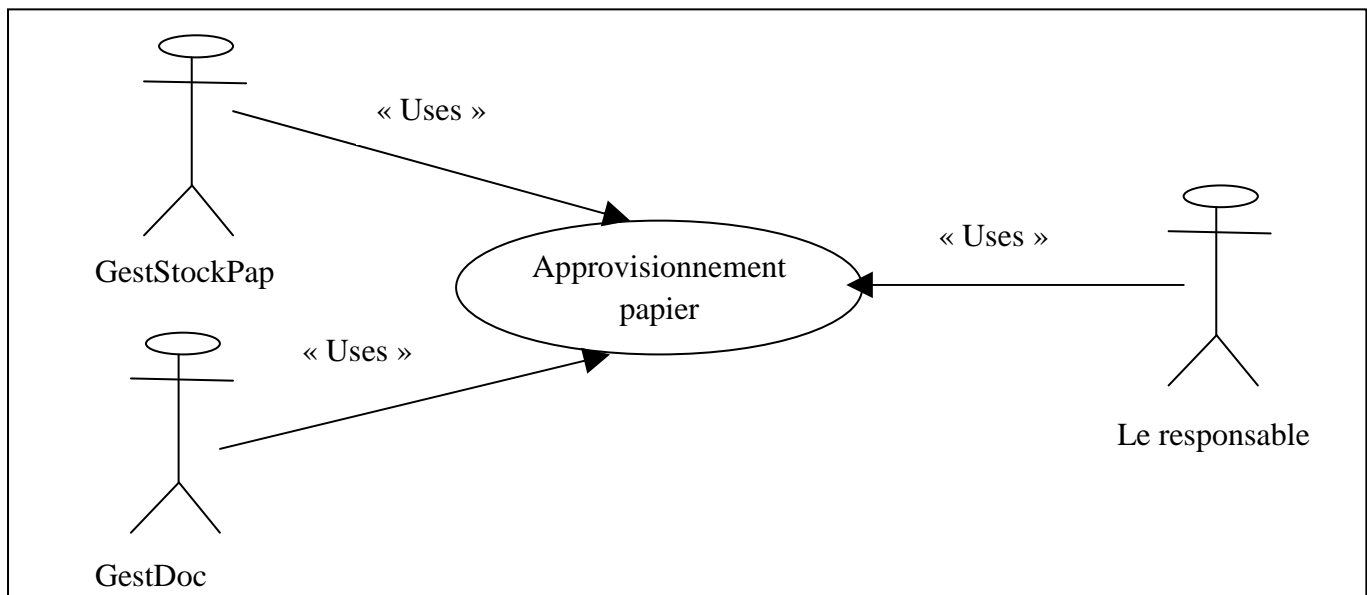
5. Validation de l'opération de saisie par le GestStockPap.

6. Edition d'état de stock par le GestDoc.

### II. 1.3.2. Description de cas d'utilisation à l'aide du tableau:

Cas	<b>Approvisionnement papier</b>
Acteur primaire	Gestionnaire de stock papier
Acteurs secondaires	Le responsable, Le gestionnaire de documents.
Pré-condition	Le stock de sécurité est atteint.
Résultats	Le papier commandé s'arrivé.
Description de scénario	-demande de prestation de service, réception de décision d'affectation et de papier, stockage physique de papier, Mis à jour de fichier courrier arrivé, saisie d'un approvisionnement, mise à jour de l'opération de stockage et validation de l'opération de saisie, édition d'état de stock.
Exception	L'état de stock doit être faite par le GestDoc, et la demande de papier se fait après l'atteint de stock de sécurité.

### II. 1.3.3. Description graphique de cas d'utilisation : « Approvisionnement papier »





### II. 1.3.4. Scénario du cas d'utilisation « Approvisionnement papier»

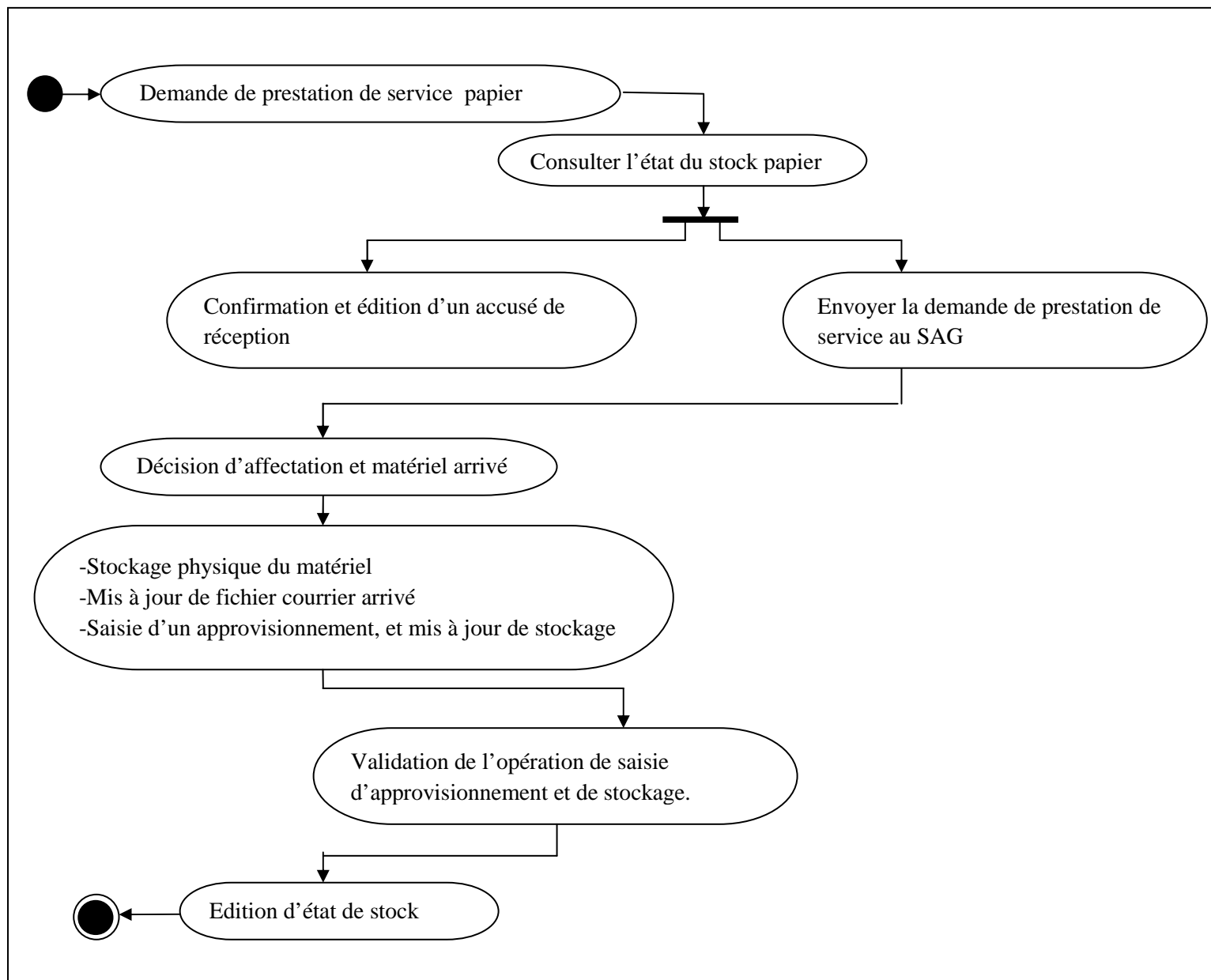


FIG.3 4. SCÉNARIO DU CAS D'UTILISATION « APPROVISIONNEMENT PAPIER»

### II. 1.4. Cas d'utilisation « Transfert papier »:

#### II. 1.4.1. Description textuelle de cas d'utilisation:

1. Le demandeur de papier (Division de DRD ou Agence) dépose la demande de prestation de service chez le responsable de la DGSI.
2. Le responsable de la DGSI transmet la demande au GestStockPap pour qu'il offre le papier dont ils ont besoin.
3. Le gestionnaire de stock papier met à jour le fichier courrier sorti, et saisi le transfert ainsi que l'opération de stockage après le transfert.
4. Validation de l'opération de saisie par le GestStockPap.
5. Edition d'état de stock par le GestDoc.

### III. 1.4.2. Description de cas d'utilisation à l'aide du tableau:

Cas	<b>Transfert papier</b>
Acteur primaire	Gestionnaire de stock papier
Acteurs secondaires	Le responsable, Le gestionnaire de documents.
Pré-condition	La quantité de papier demandée est inférieure à celle stockée.
Résultats	Etat de stock.
Description de scénario	- Réception de demande de prestation de service, transfert de papier, Mis à jour de fichier courrier sorti, saisie d'un transfert et stockage après transfert, et validation de l'opération de saisie, édition d'état de stock.
Exception	L'état de stock doit être fait par le GestDoc, et après chaque transfert.

### III. 1.4.3. Description graphique de cas d'utilisation : « Transfert papier »

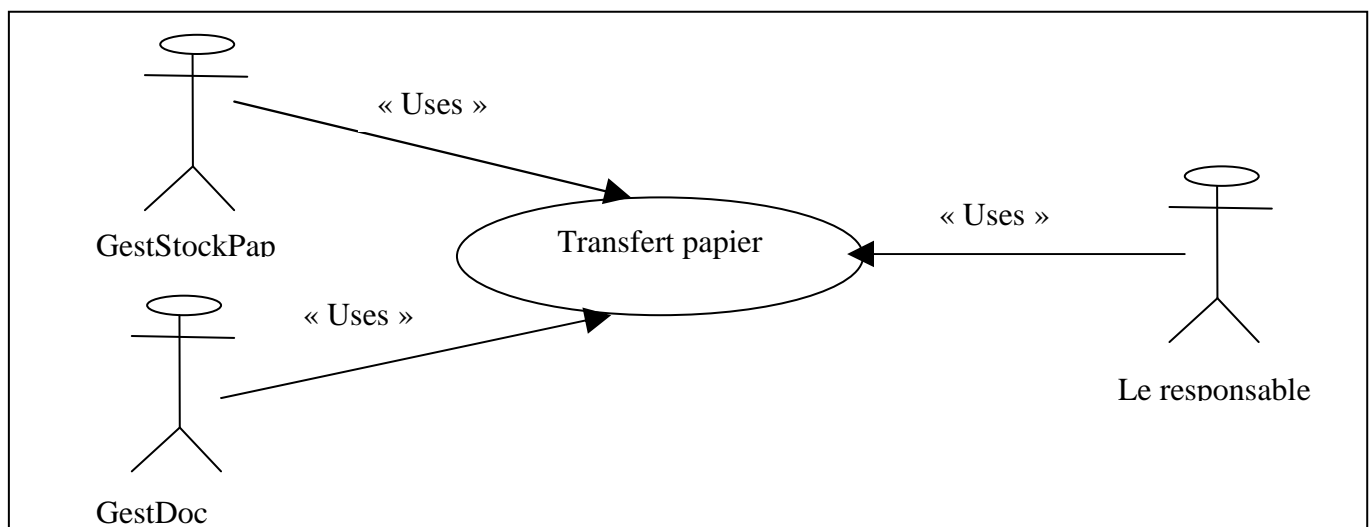


FIG4.3 : CAS D'UTILISATION « TRANSFERT PAPIER »

### II. 1.4.4. Scénario du cas d'utilisation « Transfert papier»

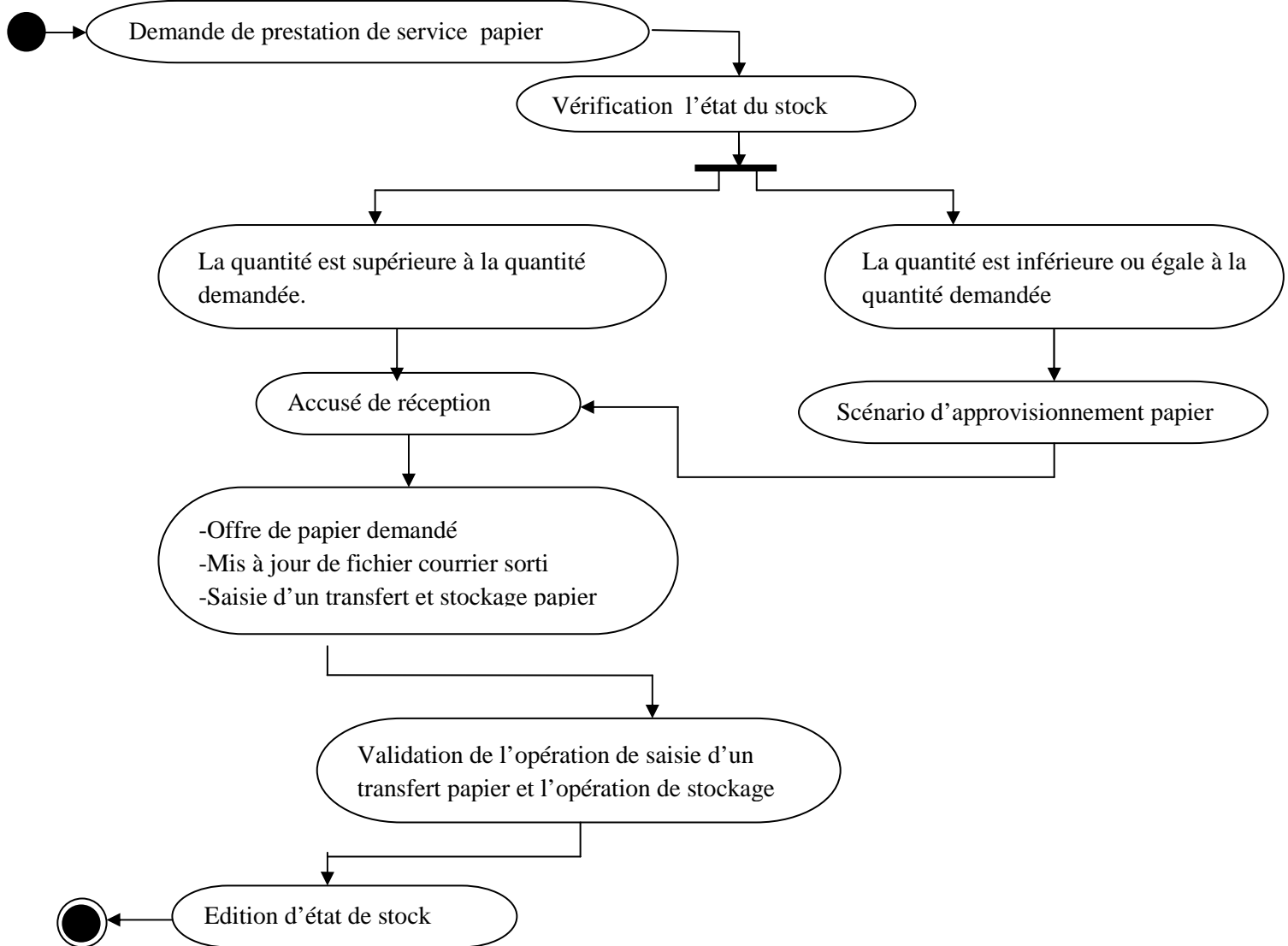


FIG.4. 4. SCÉNARIO DU CAS D'UTILISATION « APPROVISIONNEMENT PAPIER»

### II. 1.5. Cas d'utilisation « Demande et recensement de matériel entré à la DGSII»:

#### II. 1.5.1. Description textuelle de cas d'utilisation:

1. Le gestionnaire de parc informatique fait une demande de prestation de service au SAG, et doit être passé tout d'abord par le responsable.

2. Le responsable de la DGSI se confirme sur la nécessité de matériel demandé, et envoie l'accusé de réception au GestParcInfo, ainsi que la demande de prestation de service au SAG.
3. Le SAG répond par une décision d'affectation au responsable qui va la transmettre au GestParcInfo.
4. Le matériel est alors arrivé, et le technicien supérieur de la DGSI procède à son installation physique.
5. Saisie de matériel entré et validation de l'opération de saisie par le GestParcInfo.
6. Edition de décharge matérielle interne par le GestDoc.

### II. 1.5.2. Description de cas d'utilisation à l'aide du tableau:

Cas	<b>Demande et recensement de matériel entré à la DGSI</b>
Acteur primaire	Gestionnaire de parc informatique
Acteurs secondaires	Le responsable, Le gestionnaire de documents.
Pré-condition	Le matériel demandé est vraiment indispensable.
Résultats	Le matériel commandé s'arrive, décharge matériel interne.
Description de scénario	-demande de prestation de service, réception de décision d'affectation et de matériel nécessaire, installation physique de matériel, Mis à jour de fichier courrier arrivé, saisie de matériel entré, et validation de l'opération de saisie, édition d'un décharge matériel interne.
Exception	La décharge matériel interne doit être faite par le GestDoc, et après chaque arrivé de matériel.

### II. 1.5.3. Description graphique de cas d'utilisation : «Demande et recensement de matériel entré à la DGSI»

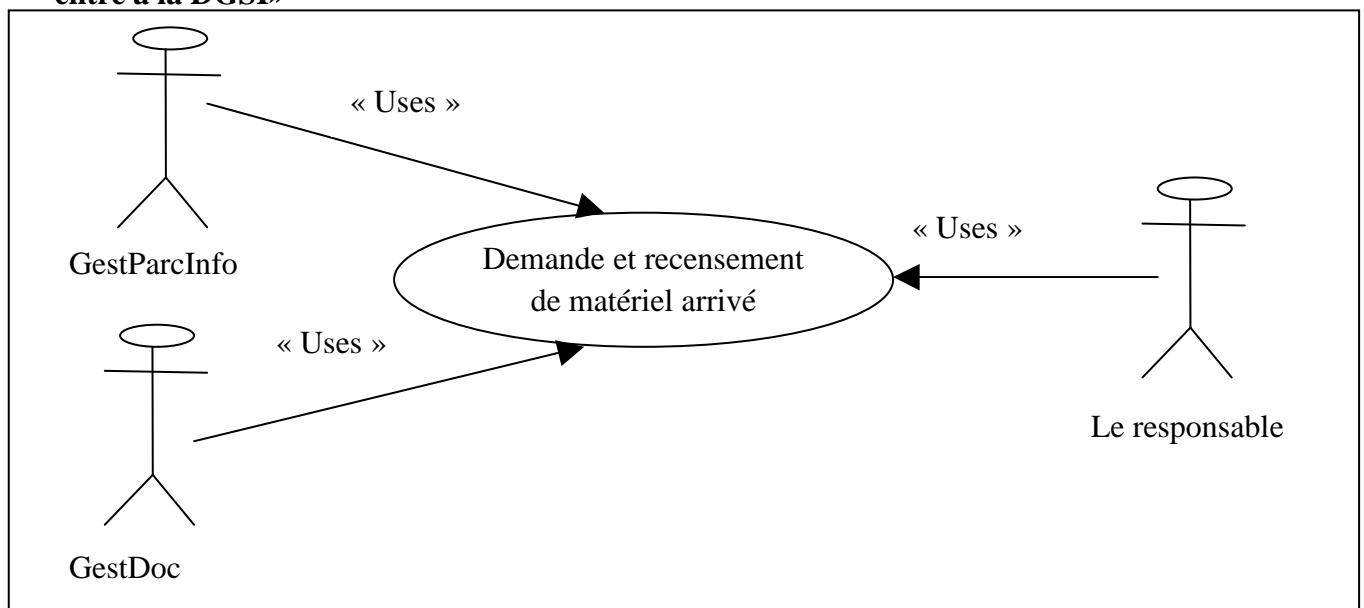
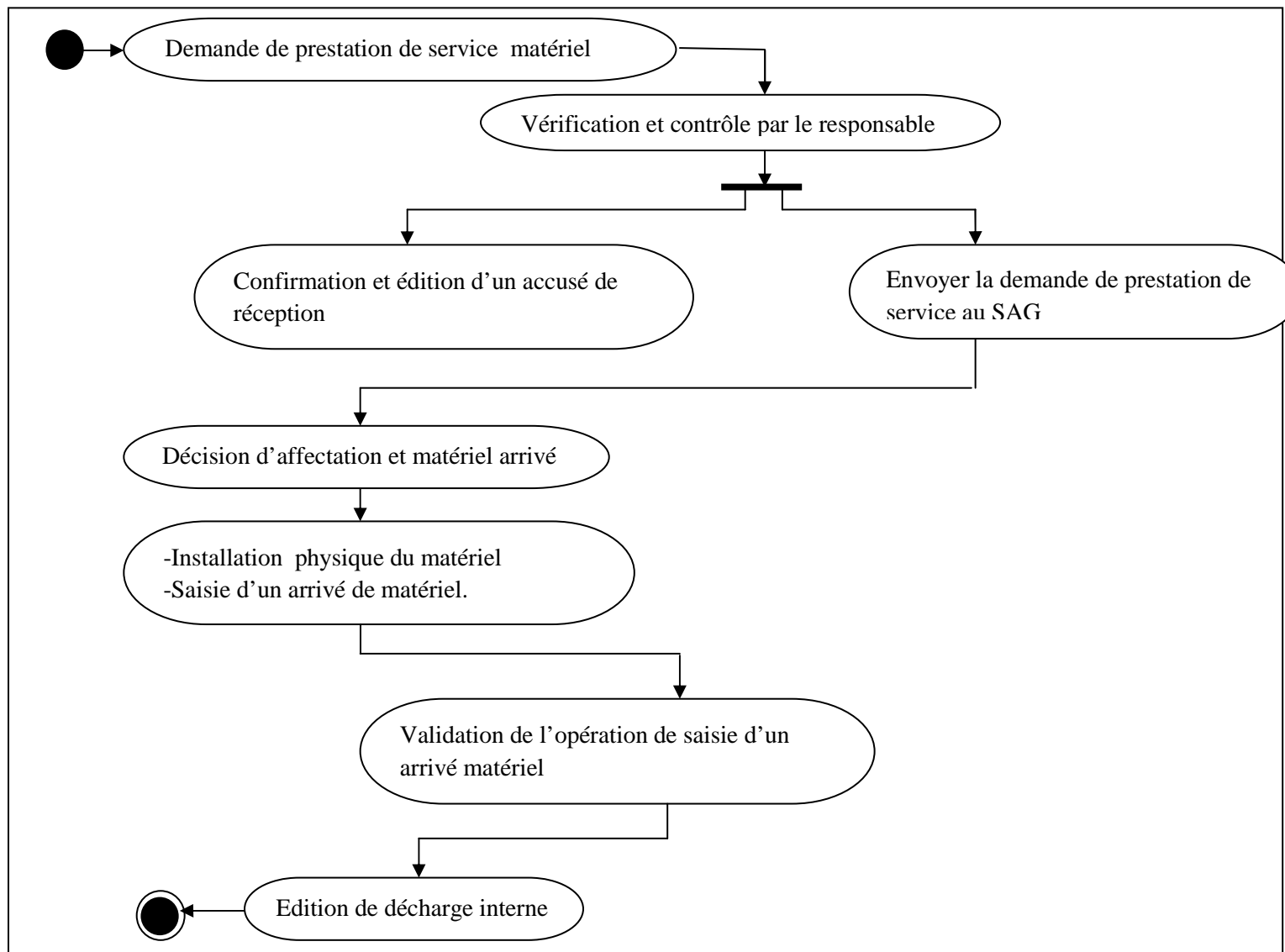


FIG.5.3 : CAS D'UTILISATION « Demande et recensement de matériel entré à la DGSI »

### II. 1.5.4. Scénario du cas d'utilisation «Demande et recensement de matériel entré à la DGSI»



### II. 1.6. Cas d'utilisation « Recensement de matériel sorti de la DGSI »:

#### II. 1.6.1. Description textuelle de cas d'utilisation:

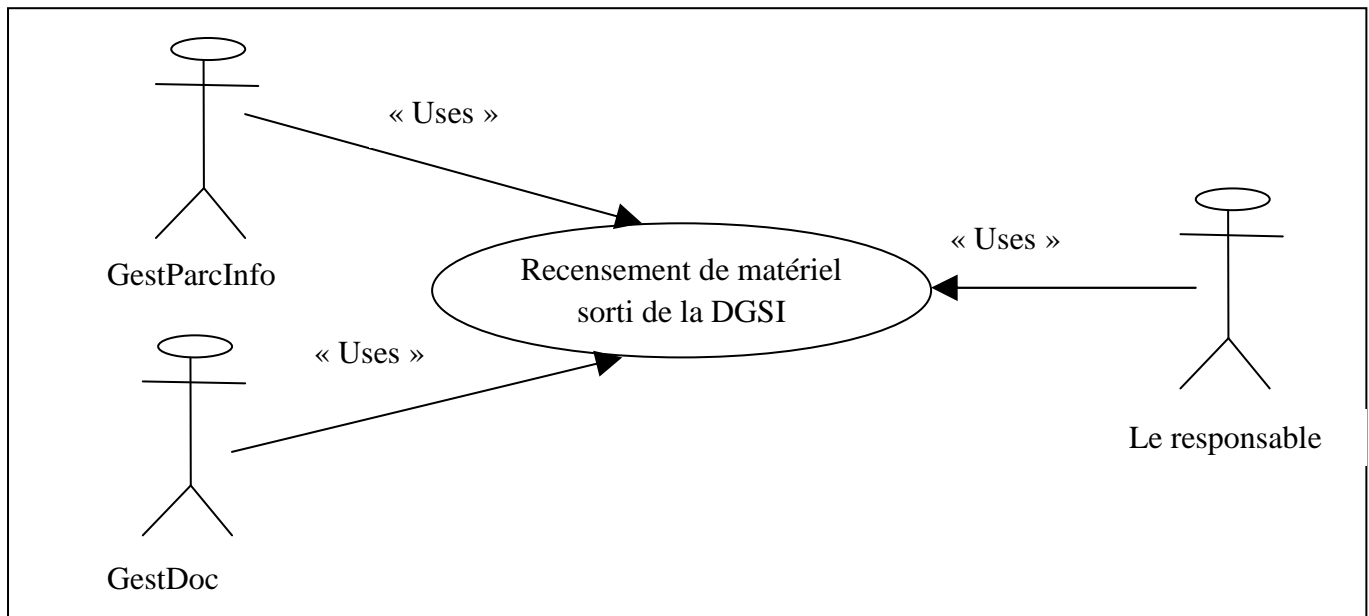
1. Le demandeur de matériel (Division de la DRD, Agence, ou bien entreprises privé en relation avec la DRD) dépose sa demande de prestation de service au niveau de la DGSI chez le responsable.
2. Le responsable de la DGSI répond par un accusé de réception et transmet la demande au SAG qui répond encore par une décision d'affectation et offre le matériel demandé.
3. Le récupérateur de matériel établit une décharge matériel externe et la dépose chez le responsable de la DGSI.

4. Le responsable de la DGSI transmet la décharge matériel externe au GestParcInfo pour qu'il puisse saisir le matériel sorti.
5. Validation de l'opération de saisie par le GestParcInfo.
6. Classement de la décharge externe par le GestDoc.

### II. 1.6.2. Description de cas d'utilisation à l'aide du tableau:

Cas	Recensement de matériel sorti de la DGSI
Acteur primaire	Gestionnaire de parc informatique
Acteurs secondaires	Le responsable, Le gestionnaire de documents.
Pré-condition	Les entreprises privées en relation avec la DRD doivent déposer leurs dossiers au préalable au niveau de la DGSI.
Résultats	Décharge matériel externe.
Description de scénario	-demande de prestation de service, réception de décision d'affectation et l'offre de matériel, saisie de matériel sorti, et validation de l'opération de saisie, classement de la décharge matérielle externe.
Exception	La décharge matérielle externe doit être reçue par la DGSI, après chaque sortie de matériel.

### II. 1.6.3. Description graphique de cas d'utilisation : «Recensement de matériel sorti de la DGSI»



**FIG.6.3 : CAS D'UTILISATION «Recensement de matériel sorti de la DGSI»**

### II. 1.6.4. Scénario du cas d'utilisation « Recensement de matériel sorti de la DGSI »

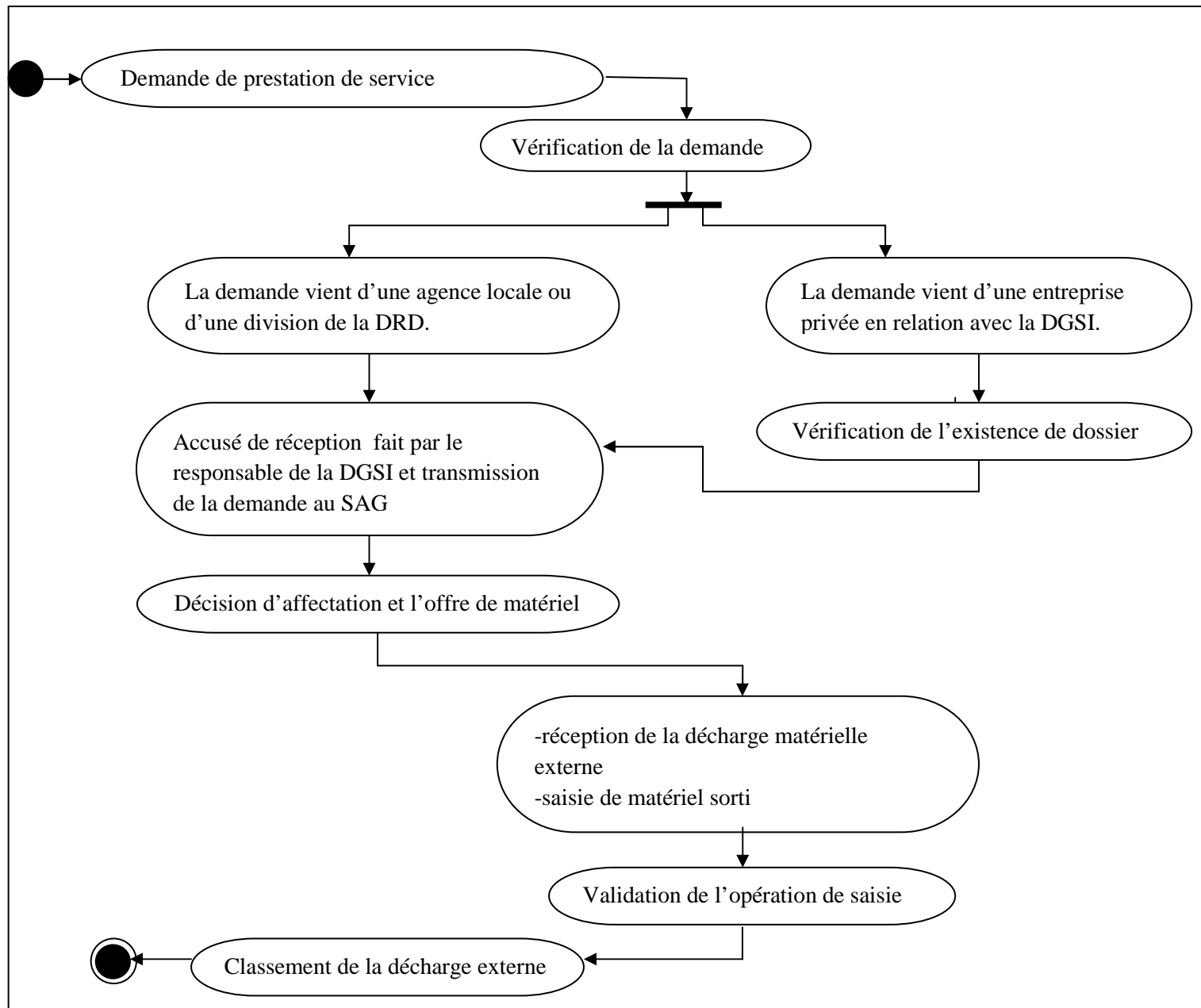


FIG.6. 4. SCÉNARIO DU CAS D'UTILISATION « RECENSEMENT DE MATÉRIEL SORTI DE LA DGSI »

### II. 1.7. Cas d'utilisation « Suivi des intervention»:

#### II. 1.7.1. Description textuelle de cas d'utilisation:

1. Réception des décharges réparation internes ou externes par le gestionnaire de parc informatique.
2. Saisie d'intervention par le GestParcInfo.
3. Validation de l'opération d'intervention.
4. Edition e rapport d'intervention par le gestionnaire des documents.

#### II. 1.7.2. Description de cas d'utilisation à l'aide du tableau:

Cas	Suivi des rapports d'intervention
Acteur primaire	Gestionnaire de parc informatique
Acteurs secondaires	Le gestionnaire de documents.
Pré-condition	La panne nécessite l'intervention d'un technicien.
Résultats	Rapport d'intervention.
Description de scénario	-Réception des décharges réparation par le GestParcInfo, saisie des interventions, et édition des rapports d'intervention.
Exception	Le rapport d'intervention doit être réalisé par le GestDoc.

#### II. 1.7.3. Description graphique de cas d'utilisation : «Suivi des intervention»

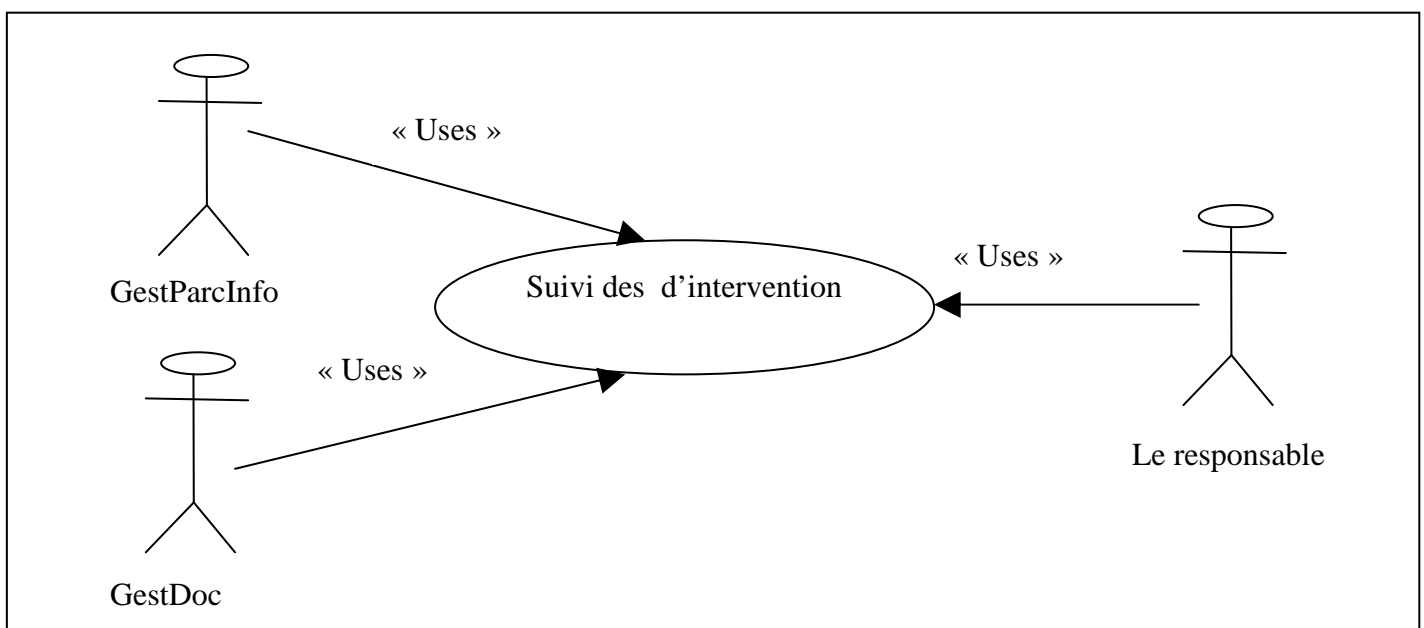


FIG. 7.3 : CAS D'UTILISATION «SUIVI DES RAPPORTS D'INTERVENTION»



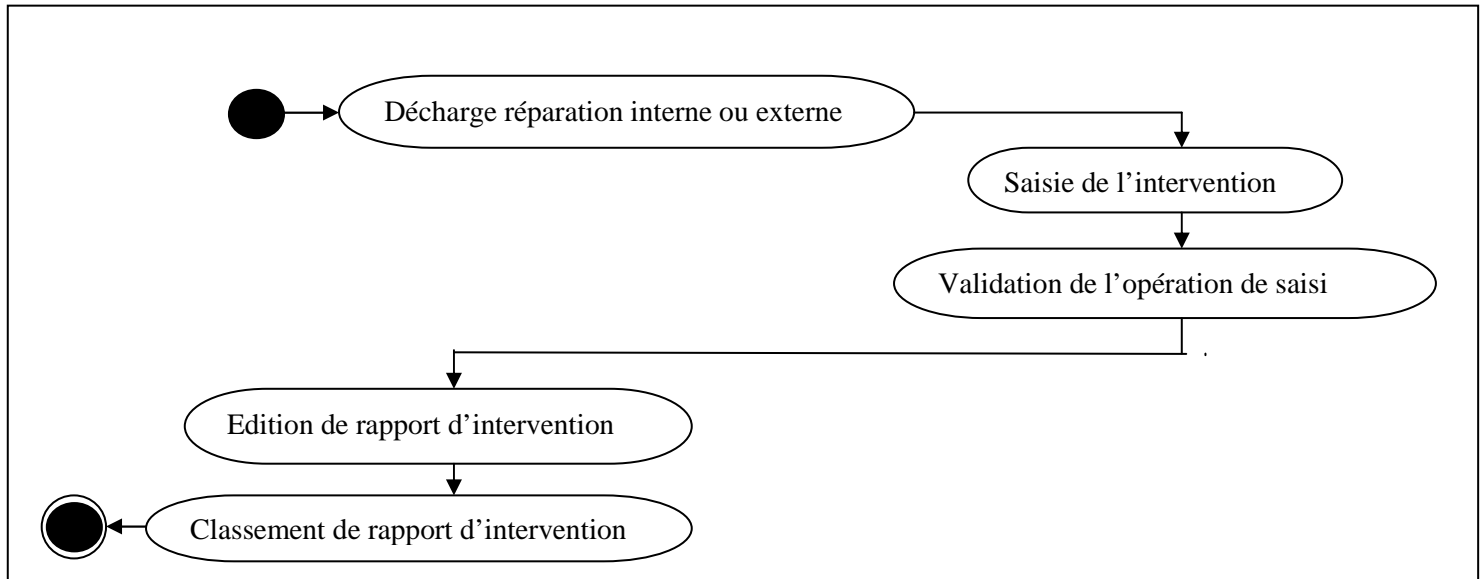


FIG. 7.4. SCÉNARIO DU CAS D'UTILISATION « SUIVI DES RAPPORTS D'INTERVENTION »

### II. 1.8. Cas d'utilisation « gestion des documents »:

#### II. 1.8.1. Description textuelle de cas d'utilisation

1. Réception de message pour l'édition de document.
2. Création et édition de document.
3. Classement de document.
4. Transmission de document.

#### II. 1.8.2. Description de cas d'utilisation à l'aide du tableau:

Cas	Suivi des rapports d'intervention
Acteur primaire	Gestionnaire des documents
Acteurs secondaires	Le responsable, GestParcInfo, GestStockPap.
Pré-condition	L'opération d'édition d'un document se déclenche par l'envoi de message par ceux qui ont besoin.
Résultats	Documents souhaités se réalisent.
Description de scénario	-Réception de message pour l'édition de document, Création et édition de document, classement de document, et transmission de document
Exception	Les documents doivent être réalisés que par le GestDoc.

### II. 1.8.3. Description graphique de cas d'utilisation : «Gestion des documents» :

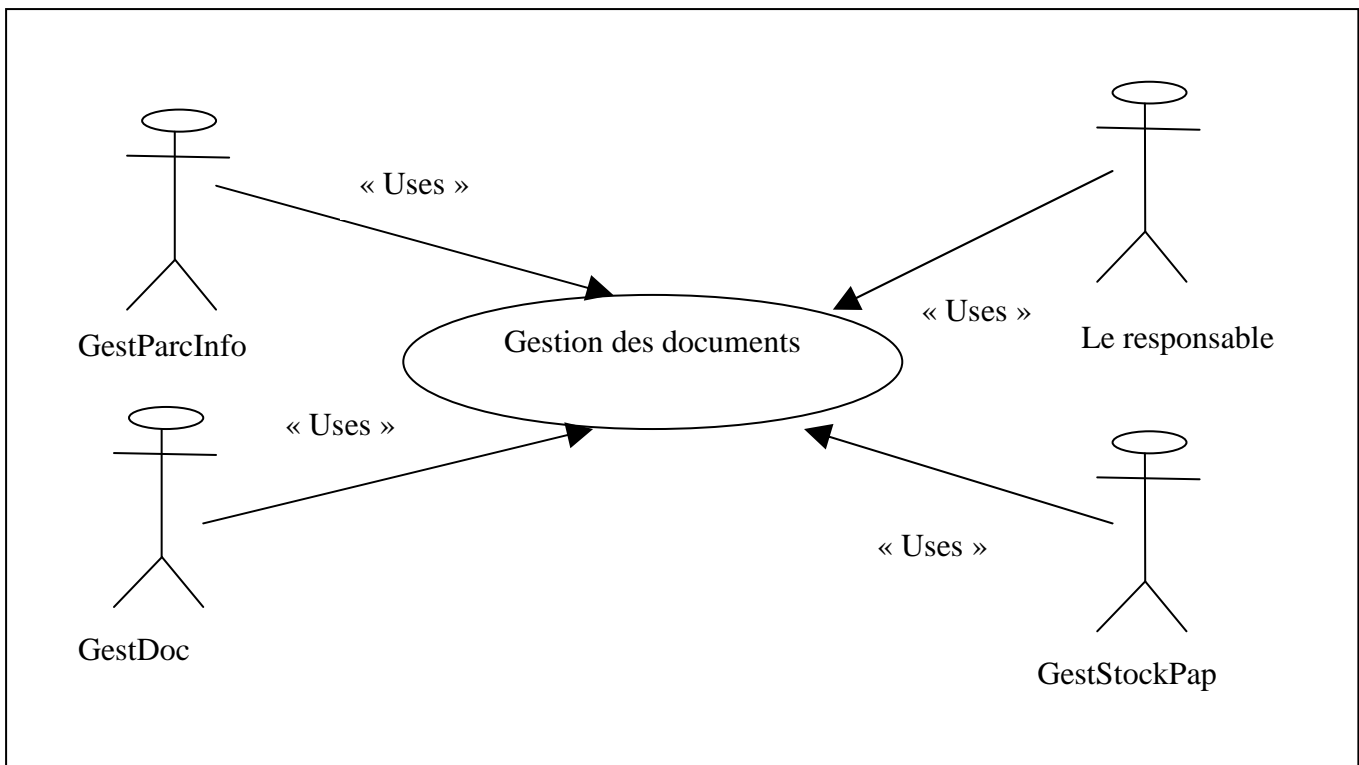


FIG.8.3 : CAS D'UTILISATION «GESTION DES DOCUMENTS»

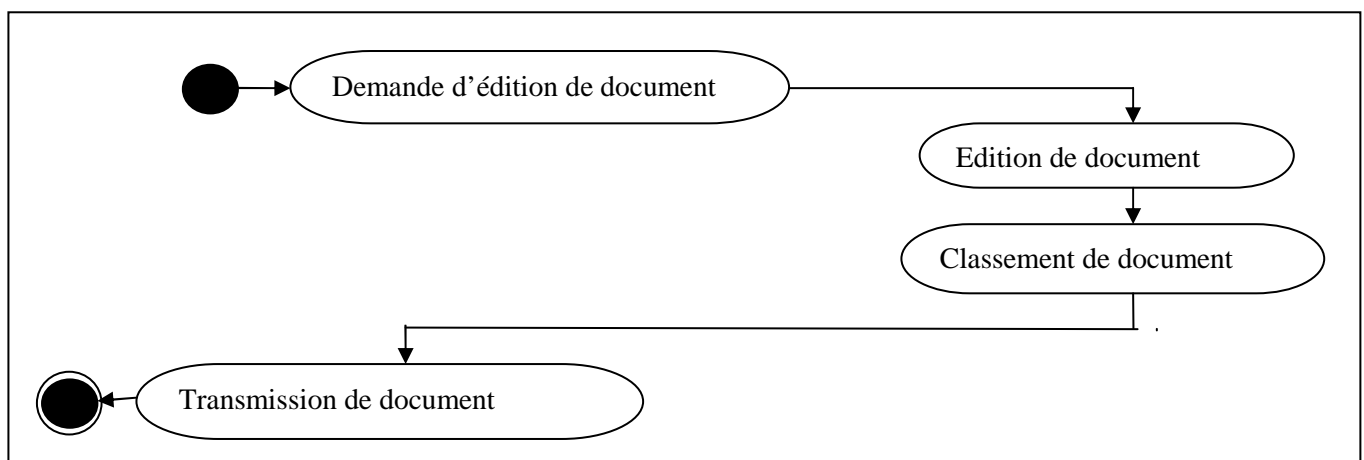


FIG.8.4. SCÉNARIO DU CAS D'UTILISATION « GESTION DES DOCUMENTS»

### III. Relations entre cas d'utilisation :

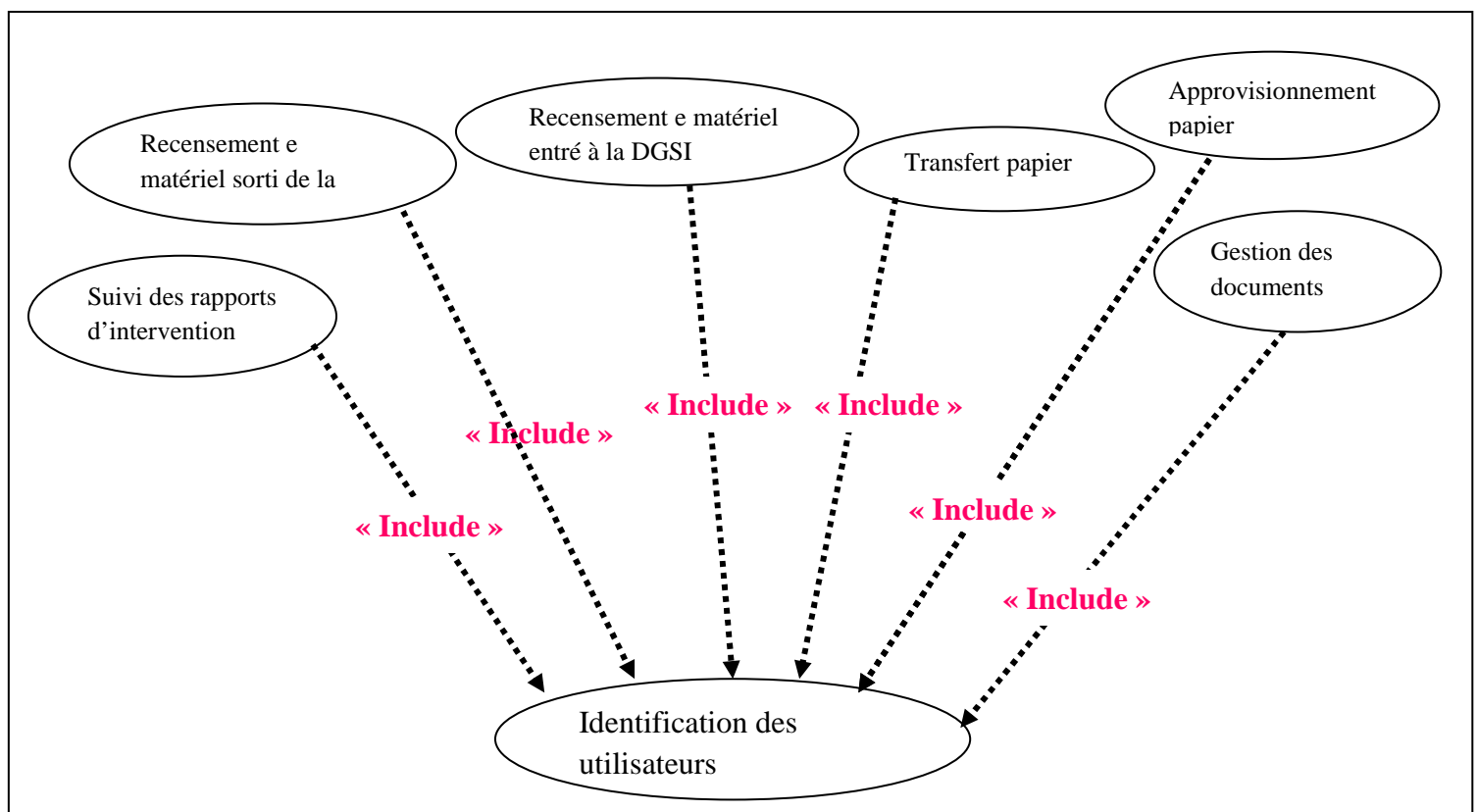
–Réutilisation de cas en utilisant les relations

- d'inclusion include
- d'extension extend
- de généralisation

#### A. Inclusion

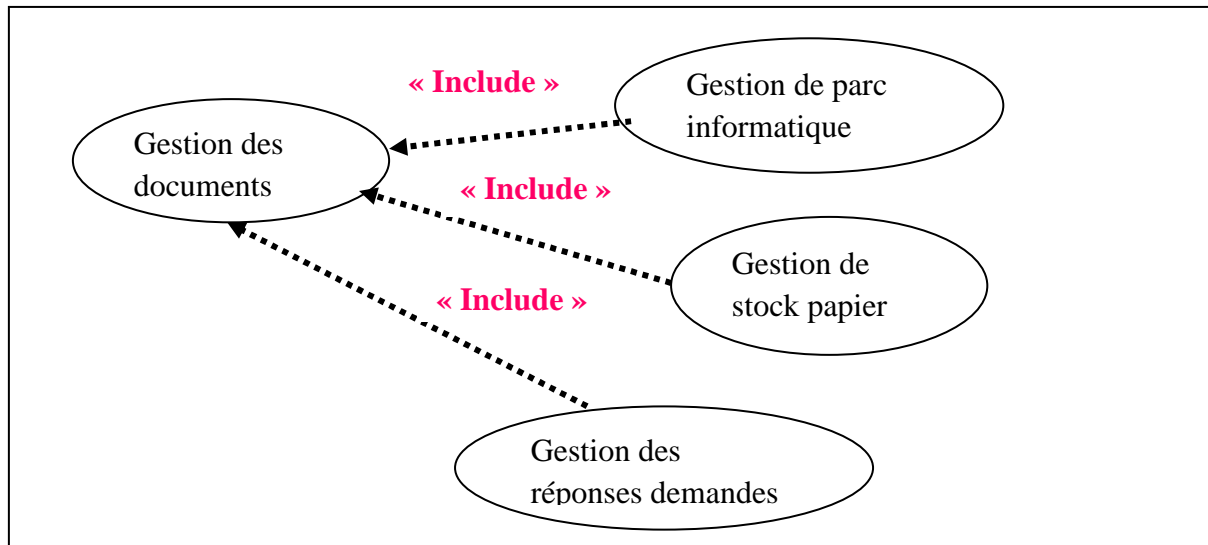
–Une instance contient le comportement d'une autre instance.

##### 1. Inclusion de cas d'utilisation « identification d'utilisateur »



**FIG. III.A.1 : INCLUSION DE CAS D'UTILISATION « IDENTIFICATION D'UTILISATEUR »**

### 2. Inclusion de cas d'utilisation « Gestion des documents »

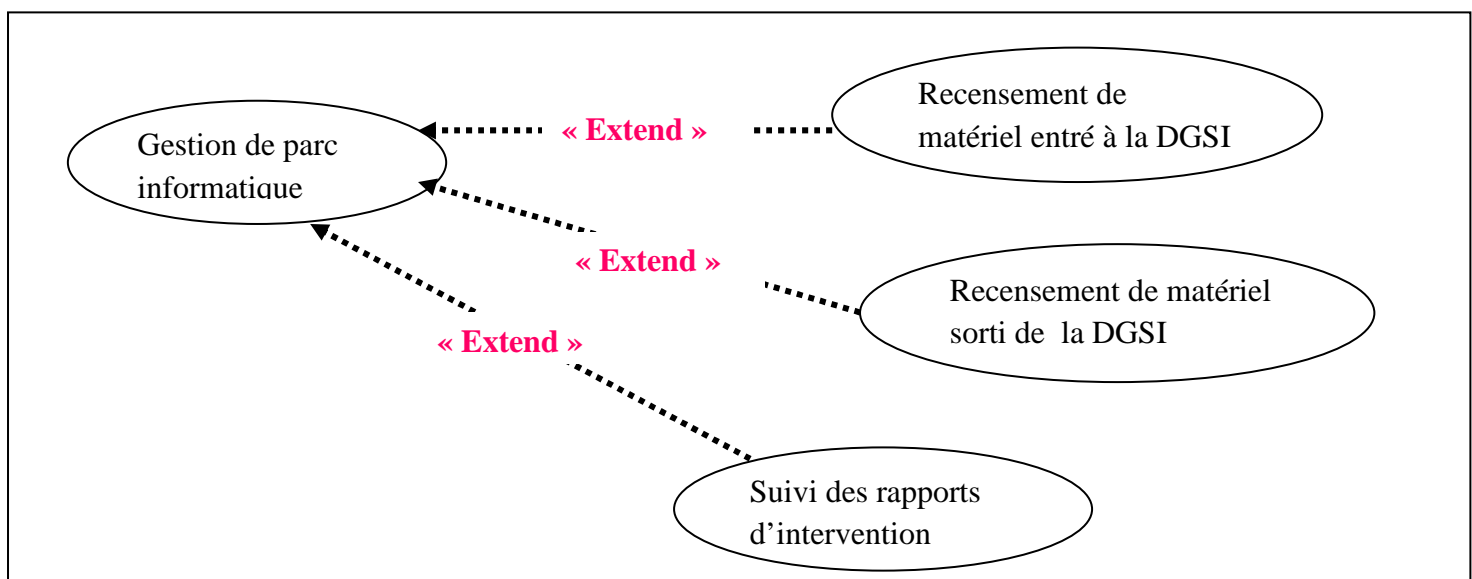


**FIG.III.A.2 : INCLUSION DE CAS D'UTILISATION « GESTION DES DOCUMENTS »**

### B. Extension :

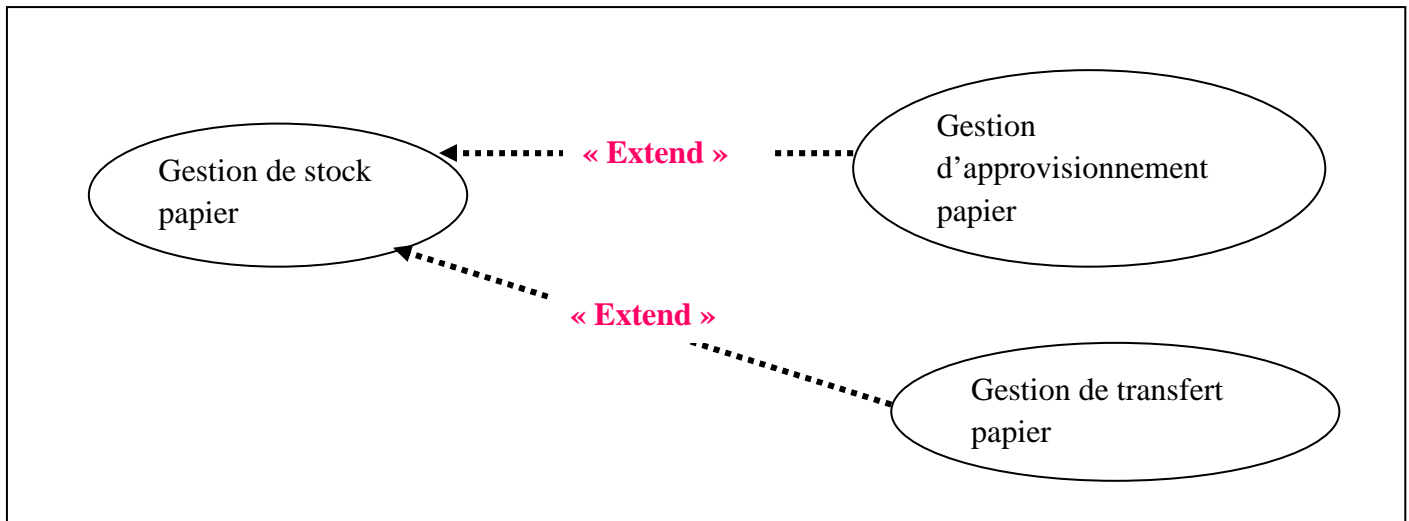
Le comportement d'une instance peut être étendu par le comportement d'une autre instance, et le Point d'extension mentionné dans le cas d'utilisation.

#### 1. Extension de cas d'utilisation « Gestion de parc informatique »



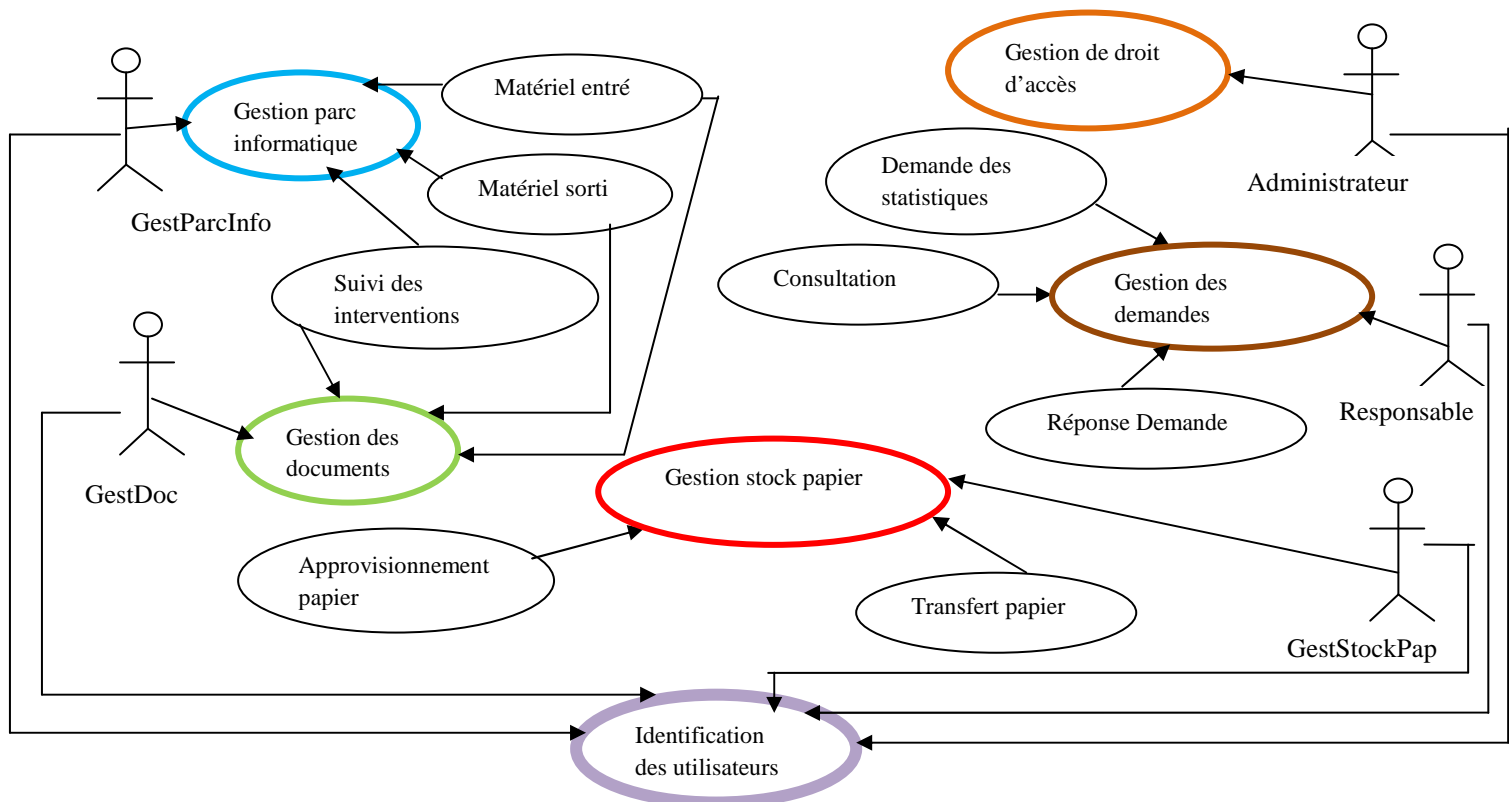
**FIG.III.B.1 : EXTENSION DE CAS D'UTILISATION « GESTION DE PARC INFORMATIQUE »**

## 2. Extension de cas d'utilisation « Gestion de stock papier »



**FIG.III.B.2 : EXTENSION DE CAS D'UTILISATION « GESTION DE STOCK PAPIER »**

### C. Diagramme de cas d'utilisation :



**Diagramme des cas d'utilisation**

Légende :

Rouge : package service stock papier

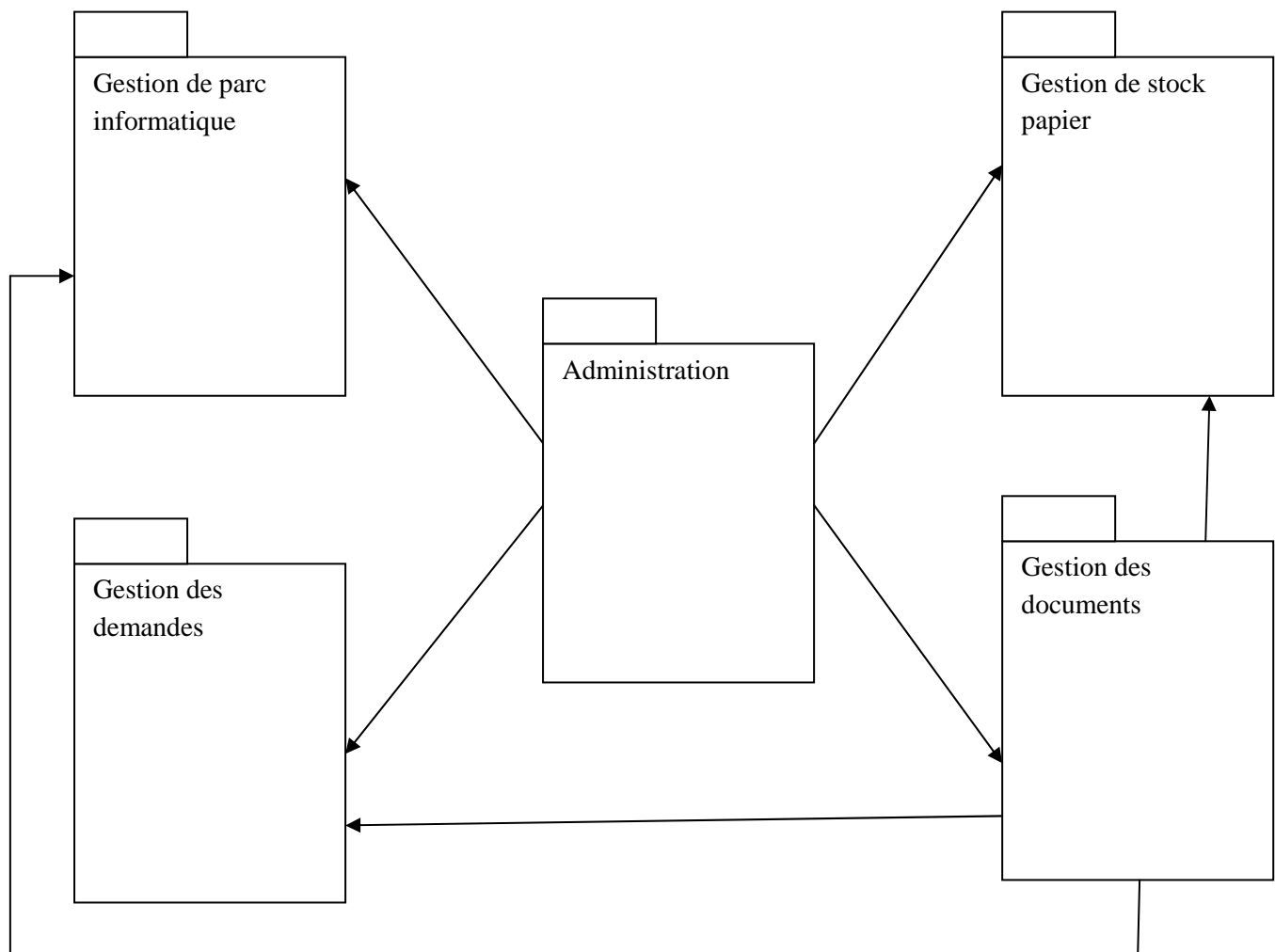
Orange : Gestion des droits d'accès

Bleu : package service parc informatique

Vert : package gestion des documents

Marron : package gestion des demandes

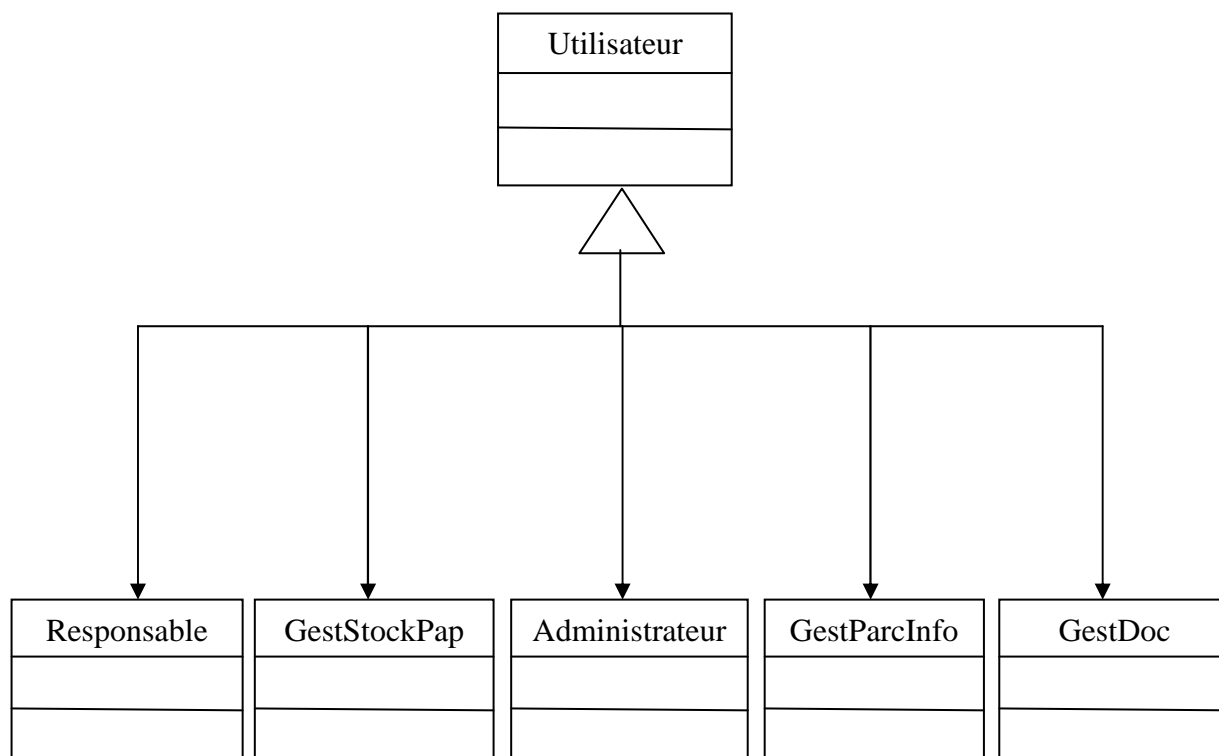
### III. Le paquetage des cas d'utilisation :



**Regroupement en packages**

Le package	Les cas d'utilisation inclus
<i>Administration du système</i>	<ul style="list-style-type: none"> <li>• Gestion des droits d'accès</li> <li>• Identification Utilisateur</li> </ul>
<i>Gestion stock papier</i>	<ul style="list-style-type: none"> <li>• Approvisionnement papier</li> <li>• Transfert papier</li> <li>• Mise à jour de stockage papier après chaque approvisionnement et chaque transfert.</li> </ul>
<i>Gestion parc informatique</i>	<ul style="list-style-type: none"> <li>• Recensement de matériel entré à la DGSI</li> <li>• Recensement de matériel sorti de la DGSI</li> <li>• Suivi des interventions</li> </ul>
<i>Gestion des documents</i>	<ul style="list-style-type: none"> <li>• Edition des documents</li> <li>• Classement de document</li> <li>• Transmission de document</li> </ul>
<i>Gestion des demandes</i>	<ul style="list-style-type: none"> <li>• Réponses aux demandes</li> <li>• Consultation des données</li> <li>• Afficher les statistiques</li> </ul>

**Diagramme des classes participantes au package «Administration du système» :**



**diagramme de Classes « administration du système»**

### Diagramme des classes participantes au package « gestion stock papier »

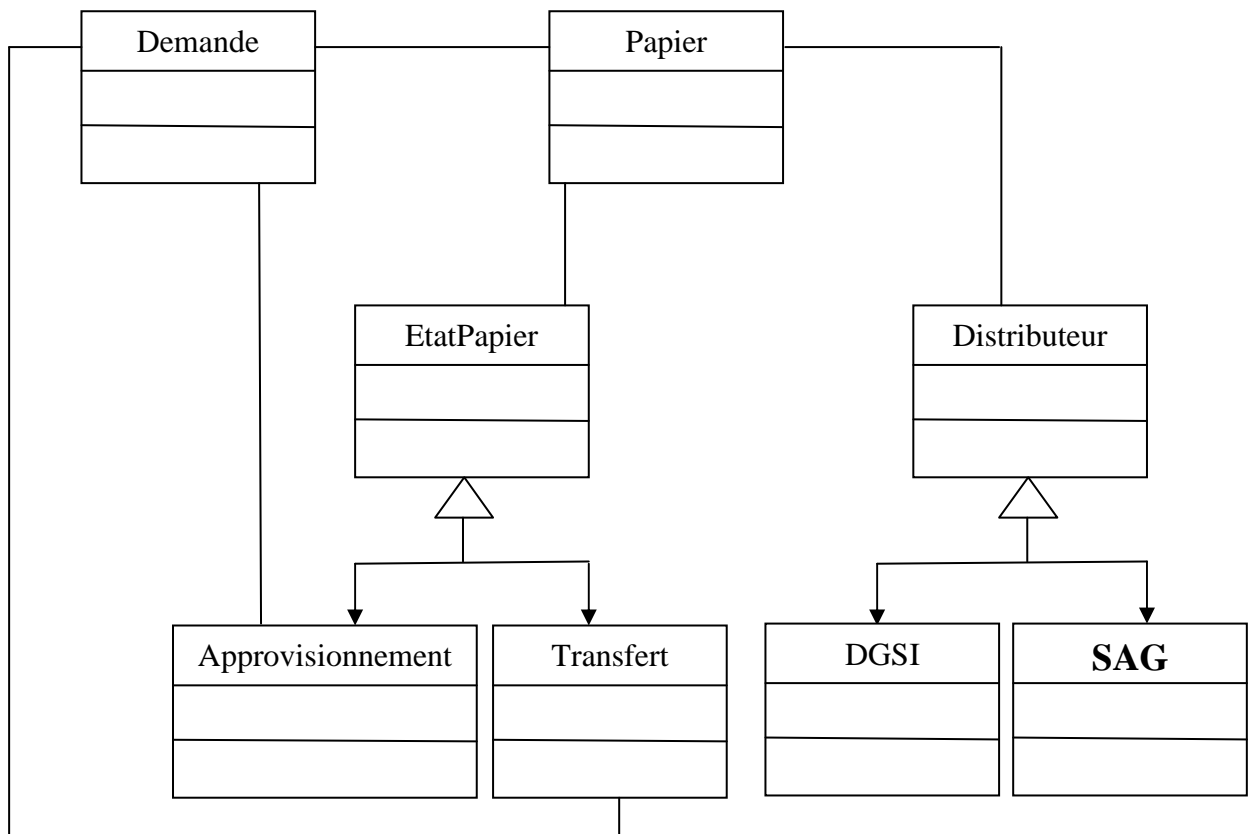


diagramme de Classes « Gestion de stock papier»



### Diagramme des classes participantes au package « gestion de parc informatique »

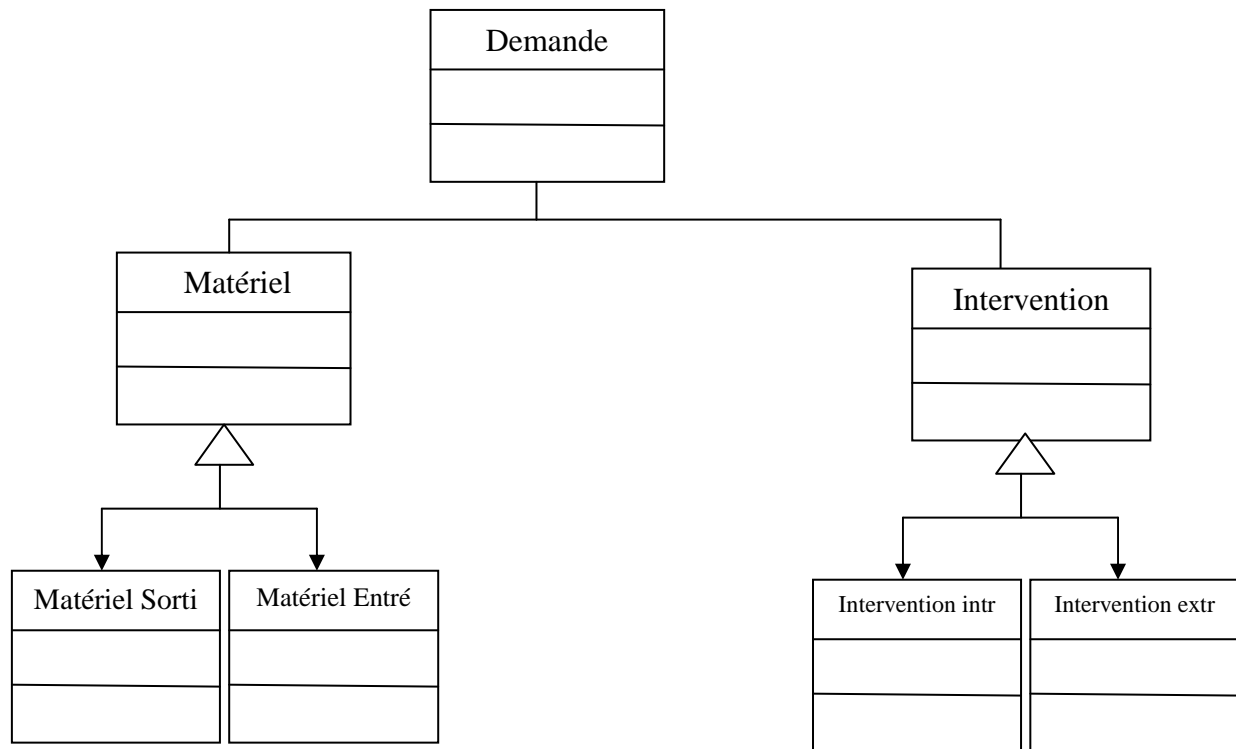


diagramme de Classes « Gestion de parc informatique»

#### IV. Capture des besoins techniques :

La spécification des besoins techniques est une activité complémentaire à la spécification des besoins fonctionnels que toutes les deux contribuent à la conception de l'architecture d'un logiciel.

##### IV. 1. Spécifications techniques de point de vue matériel :

Durant ce paragraphe on va citer les contraintes : géographiques, organisationnelles et techniques relatives à la configuration du réseau matériel. Ces contraintes concernent les performances d'accès aux données, la sécurité du système, l'interopérabilité, la volumétrie et le mode d'utilisation du système.

##### IV. 1.1. Les différentes architectures réseaux d'un système d'information :

## Chapitre VII : Analyse et conception

---

Avant tout réalisation d'une application informatique, et définition de la solution mise en place pour un système d'information distribué au sein d'un réseau, on s'intéresse aux différentes infrastructures existantes qui sont illustrées dans le chapitre « Réseau et architecture Client/serveur » (Les infrastructures présentées sont : la 3tiers ; 2tiers, ntiers).

### IV. 1.2. L'infrastructure réseau :

Si les applications distribuées deviennent le principal outil du système d'information de l'entreprise, comment assurer leur accès sécurisé au sein de structures parfois réparties sur de grandes distances géographiques ? Concrètement comment une succursale d'une entreprise peut-elle accéder aux données situées sur un serveur de la maison mère distant de plusieurs milliers de kilomètres ? Les VPN ont commencé à être mis en place pour répondre à Ce type de problématique. Mais d'autres problématiques sont apparues et les VPN ont aujourd'hui pris une place importante dans les réseaux informatique et l'informatique distribuées. On verra ici quel est le principe de fonctionnement de ce réseau.

#### IV. 1.2.1. Le principe de fonctionnement de réseau VPN :

Un réseau VPN repose sur un protocole appelé "protocole de tunneling" qui a comme principe la construction d'un chemin virtuel après l'identification d'émetteur et de destinataire. En suite, la source chiffre les données et les achemine en empruntant Ce chemin virtuel. Ce protocole permet de faire circuler les informations de l'entreprise de façon cryptée d'un bout à l'autre du tunnel. Ainsi, les utilisateurs ont l'impression de se connecter directement sur le réseau de leur entreprise.

Afin d'assurer un accès aisé et peu coûteux aux intranets ou aux extranets d'entreprise, les réseaux privés virtuels d'accès simulent un réseau privé, alors qu'ils utilisent en réalité une infrastructure d'accès partagée, comme Internet.

Les données à transmettre peuvent être prises en charge par un protocole différent d'Ip. Dans Ce cas, le protocole de tunneling encapsule les données en ajoutant une en-tête. Le tunneling est l'ensemble des processus d'encapsulation, de transmission et de désencapsulation.

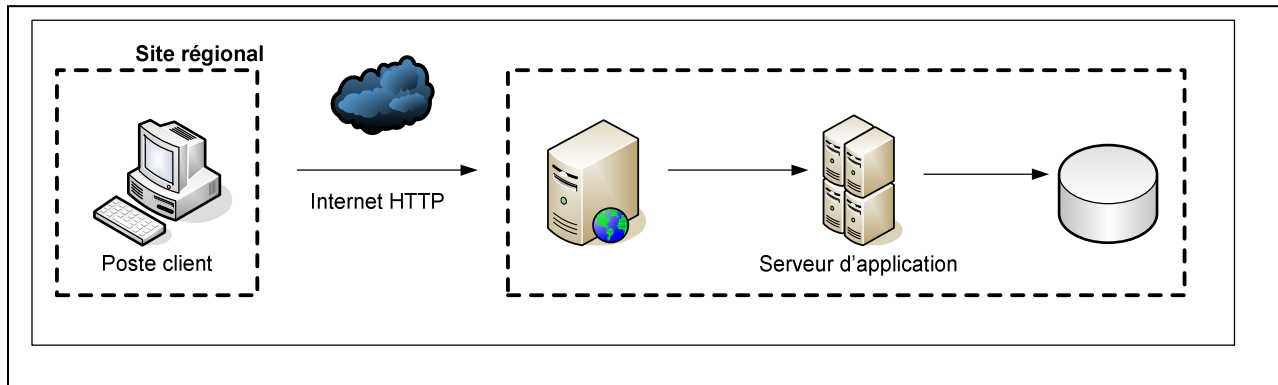
### IV. 1.3. Déploiement de la solution :

En tenant compte des différents schémas de déploiement et de stratégie de déploiement de données et en considérant les points suivants :

- ✓ L'étendue géographique importante des agences de la sonelgaz et les différents services de la DGSI imposent l'utilisation d'un protocole de transmission à longue distances.
- ✓ Le nombre important de services de la DGSI impose d'utiliser une configuration économique en matière de déploiement des fonctionnalités.
- ✓ La politique de la sonelgaz visant à simplifier l'accès à l'information.

## Chapitre VII : Analyse et conception

Nous avons opté pour un schéma de déploiement multi tiers avec une centralisation des données.



**Figure 47 : Déploiement de la solution**

**Le client** : est de type Web (léger), il est donc doté d'un navigateur web.

**Le serveur Web** est chargé d'intercepter les requêtes http et de ressusciter les résultats.

**Le serveur d'application** a deux rôles principaux :

- ✓ Serveur de présentation : chargé de gérer la logique de la présentation et la navigation.
- ✓ Serveur d'application métier : gère l'ensemble des composants métiers.

**La base de données** assure la persistance des données.

### IV. 2. Spécification techniques de point de vue logicielle :

Après avoir exprimé les spécifications techniques et d'architecture on s'intéresse aux fonctionnalités propres du système technique en procédant à une spécification logicielle. On commence par la citation des outils de développement, après cela on passe aux cas d'utilisation pour décrire les besoins purement techniques du système.

#### IV. 2.1) Les outils de développement :

Les outils de développement concernent le système de gestion des bases de données et le langage de développement (de programmation) ainsi que le système d'exploitation.

En convention avec les membres de la DGSI on a choisi les outils suivants :

**SGBDR** : pour le système de gestion de base de données, le choix a été fixé pour **MYSQL**.

**Environnement de programmation** : le choix de langage de programmation est fixé sur

**Java2.**

**Le système d'exploitation :** le système d'exploitation qui sera installé est : **Windows7 Edition intégrale.**

### IV. 2.2) Identification des cas d'utilisation technique :

Un cas d'utilisation technique est destiné à l'exploitant. C'est une séquence d'actions produisant une valeur ajoutée opérationnelle ou purement technique.

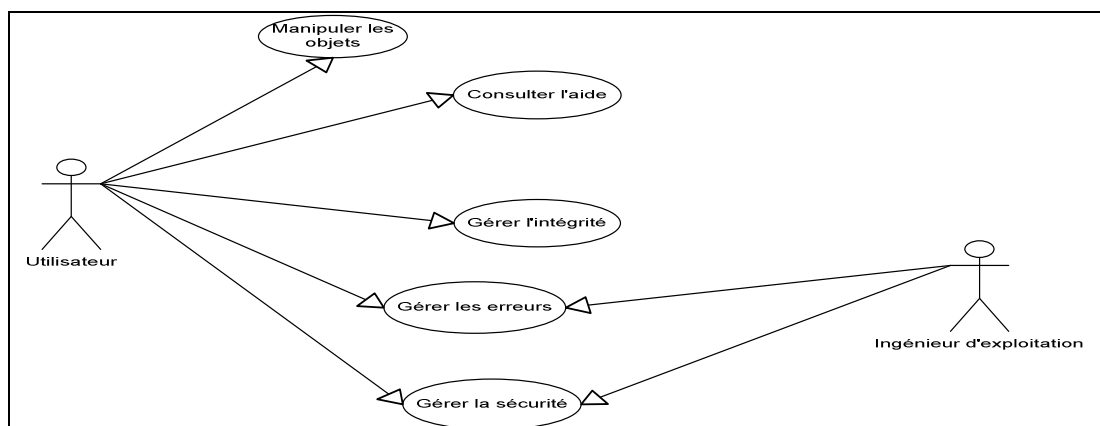
**Les exploitants du système :** les exploitants de notre système sont :

- L'utilisateur : qui utilise l'application, la majorité des acteurs de la branche fonctionnelle sont donc des utilisateurs dans la dimension technique.
- L'ingénieur d'exploitation : qui est chargé de déployer et de maintenir le système.

**Les cas d'utilisation techniques :** les cas d'utilisation du système sont identifiés en considérant l'attente opérationnelle de chaque exploitant :

- Manipulation des objets : l'utilisateur va manipuler des entités sous forme d'objets, ce qui implique la mise en œuvre de mécanismes de persistance et de gestion de cycle de vie des objets.
- Gérer l'intégrité : plusieurs utilisateurs peuvent travailler en parallèle, l'intégrité est le mécanisme qui empêche la mise à jour simultanée d'une même entité par deux utilisateurs différents.
- Gérer la sécurité : l'ingénieur d'exploitation ainsi que les utilisateurs sont soumis à des règles de sécurité. Dans un système/client serveur ces aspects recouvrent l'authenticité, l'habilitation, le cryptage, la non-répudiation et l'audit.
- Gérer les erreurs : le système doit être exploitable, A ce titre, il faut qu'il soit en mesure de générer des traces et des alertes qui vont faciliter sa maintenance au sein du système informatique global de l'entreprise. C'est cette analyse technique du problème qui permet d'introduire l'ingénieur d'exploitation comme autre exploitant.

**Diagramme de cas d'utilisation techniques :**



**Figure 48 : Diagramme de cas d'utilisation technique**

### Description des cas d'utilisation techniques :

#### 1) Cas d'utilisation « Manipuler les objets » :

**Titre :** manipulation des objets de système.

**Exploitant :** L'utilisateur

**Intention :** l'utilisateur désire agir sur le cycle de vie d'un ou plusieurs objets

#### 2) Cas d'utilisation « Gérer l'intégrité » :

**Titre :** Gestion de l'intégrité des objets

**Exploitant :** L'utilisateur

**Intention :** L'utilisateur désire gérer l'intégrité des objets de système, pour ne pas avoir un problème sur l'accès aux données.

**Action :** avant la sauvegarde des informations l'utilisateur doit s'assurer que l'objet est libre (malgré que cette gestion est assurée par le système de gestion de base de données mais l'utilisateur ne doit pas forcer le système à exécuter une telle action, cela veut dire que si le système a un retard dans l'exécution d'une requête l'utilisateur doit attendre son tour).

#### 3) Cas d'utilisation « Gérer la sécurité » :

**Titre :** Gérer la sécurité.

**Exploitant :** L'utilisateur, Ingénieur d'exploitation.

**Intention :** les exploitants doivent gérer la sécurité du système pour éviter les accès non autorisés et éviter les pannes .

**Action :** les utilisateurs doivent s'authentifier par un mot de passe et un nom d'utilisateur, et ils doivent mettre en place des moyens et des mécanismes pour assurer la sécurité.

**Exemple :** utiliser des pare feux, les extincteurs, ne pas autoriser les gens à utiliser les PC accédants au système.

### 4) Cas d'utilisation « Gérer les erreurs » :

**Titre :** Gestion des erreurs

**Exploitant :** L'utilisateur, l'ingénieur d'exploitation.

**Intention :** Gérer les erreurs du système, ces erreurs peuvent avoir plusieurs origines : elle peuvent être dues au système lui-même, au système de gestion de base de données , ou au réseau.

**Action :** Chaque fois qu'une erreur se produit, l'utilisateur doit essayer de la régler sinon il doit contacter l'ingénieur d'exploitation pour détecter l'origine du problème et le régler.

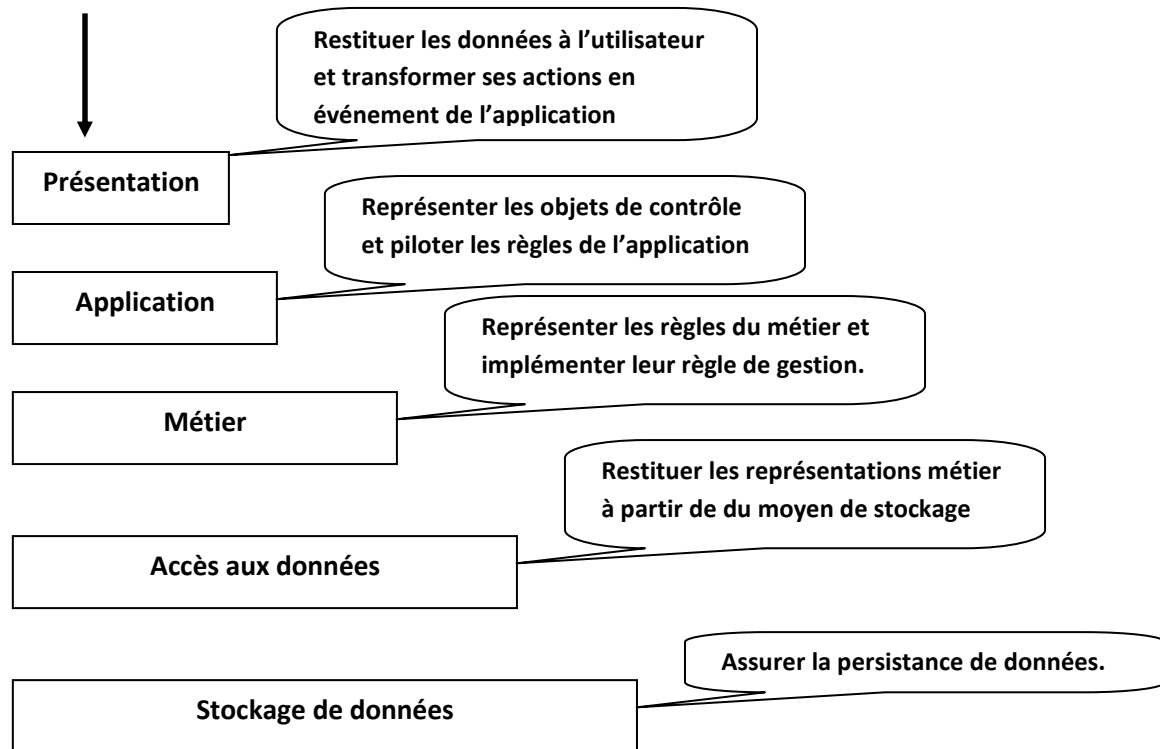
**Exemple :** Si le système affiche une erreur de type *Proxy* il doit contacter l'ingénieur se trouvant à la direction d'informatique pour voir la cause de cette erreur et dépanner le réseau.

### IV. 2.3) Elaboration du modèle de spécification logicielle :

A l'usage, le modèle de spécification logicielle obtenu est cependant trop sommaire pour permettre une spécification technique détaillée. Tous les cas d'utilisation tels que « manipuler les objets » concernant différentes responsabilités de traitement qui vont de l'interface utilisateur à la base de données. Dans ce contexte, il est difficile de pouvoir préciser de manière détaillée les comportements techniques attendus, si l'on n'organise pas la spécification suivant les différentes responsabilités de traitement.

#### ❖ Organisation en couches du modèle de spécification :

Le schéma suivant montre les différentes couches logicielles d'un système, il donne un aperçu sur les différentes responsabilités de traitement :



Le recours aux couches logicielles va nous permettre d'affiner la spécification technique en divisant le problème en sous-parties spécialisées. Nous allons voir ces couches d'une manière plus détaillée dans le paragraphe de conception.

### Partie II : Analyse de nouveau système

#### Section 1 : Analyse statique

##### Construction de dictionnaire de données :

Le dictionnaire de données est un tableau, regroupant toutes les données du système d'information ; il fournit des détails sur toute entité enregistré par ce système, contient toutes les informations recueillies dans les documents et les registres manipulés, et prend le nom brut car il peut contenir des synonymes, des polysémies et certaines information inutiles.

Attribut	Type	Taille
Adresse IP	N	16
Année Acquisition Matériel	AN	08
Code de document	AN	10
Code Matériel	AN	10
Date approvisionnement	Date	08
Date de document	Date	08
Date Déplacement	Date	08
Date Intervention	Date	08
Date Transfert	Date	08
Destination Déplacement	A	10
Durée Intervention	AN	08
Etat Matériel	AN	15
Heure Déplacement	AN	08
Marque Matériel	AN	15
Mission Déplacement	A	10
Modèle Matériel	AN	10
Nom Bios	AN	10
Nom de division	A	04
Nom de service émetteur	A	50
Nom de service récepteur	A	50
Nom Matériel	AN	10
Nom Structure	A	04
Numéro Approvisionnement	N	10
Numéro Déplacement	N	08
Numéro Transfert	N	10
Objet de document	A	40
Objet Intervention	A	10
Type Matériel	AN	15



### Section 2 : Analyse dynamique

#### I. Analyse des états de transition :

Le diagramme des états et transitions décrit les changements d'états d'un objet. L'état d'un objet à un instant  $t$  peut changer à l'instant  $t+1$ . Un objet est toujours dans un état donné pour un certain temps et ne peut être dans un état inconnu ou non défini.

Le passage d'un état à un autre est une transition, une transition est un changement d'état, provoqué par un événement. La transition peut être gardée par une condition (garde).

On peut lui associer une action. Une action est une opération instantanée, sans interruption.

On peut associer une activité à l'état. Une activité est une opération que fait l'objet quand il est dans cet état. Une activité peut être interrompue.



#### Objet demande :

Dans notre cas l'objet demande peut changer d'état et cela dépend d'un facteur principal qui est l'acceptation d'une demande (papier, matériel). Donc une demande peut passer sur les trois états suivants :

- **En attente** : la demande rentre dans cet état dès sa création et demeure dedans jusqu'à la prise de décision par le responsable de la DGS.
- **acceptée** : dans le cas où la décision est positive on change l'état de la demande en acceptée.
- **refusée** : dans le cas où la décision est négative on change l'état de la demande en refusée.

#### Diagramme d'états transitions :

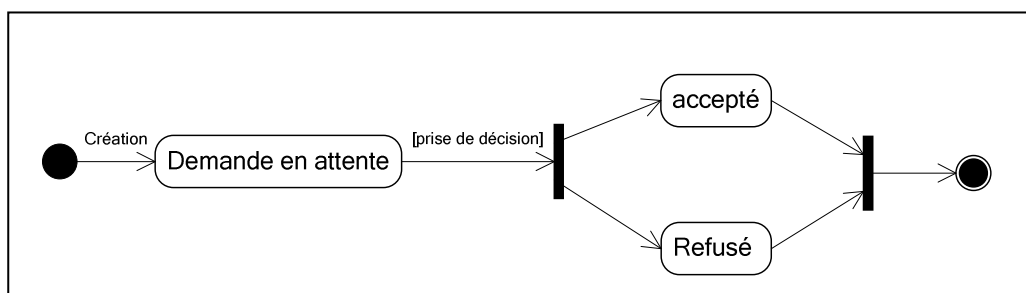


Figure 49 : Diagramme d'états de transition de l'objet « demande »

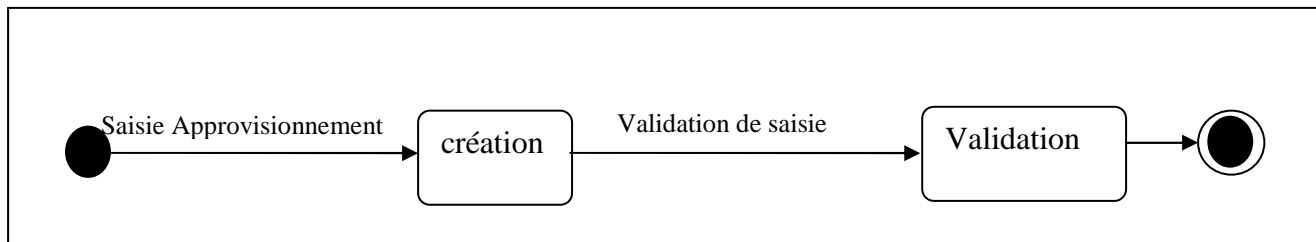
## Chapitre VII : Analyse et conception

### Objet Approvisionnement :

L'approvisionnement transite entre deux états différents :

- **création** : l'approvisionnement passe dans cet état dès l'arrivée de produit, jusqu'à la fin de la saisie.
- **Validation** : l'approvisionnement passe à cet état dès que le système se confirme.

#### Diagramme d'états transitions :



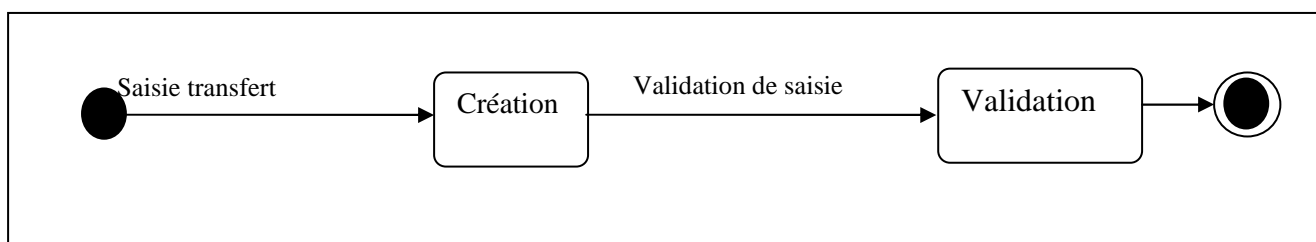
#### Diagramme d'états transitions de l'objet Approvisionnement

### Objet transfert (papier ou Matériel) :

Le transfert transite encore entre deux états différents :

- **Création** : le transfert passe dans cet état dès que le produit (matériel, papier) sort de la DGSI, jusqu'à la fin de la saisie.
- **Validation** : l'état de transfert est valide à la confirmation de système.

#### Diagramme d'états transitions :



#### Diagramme d'états transitions de l'objet Transfert

### Objet Document :

L'objet document transite entre quatre états différents :

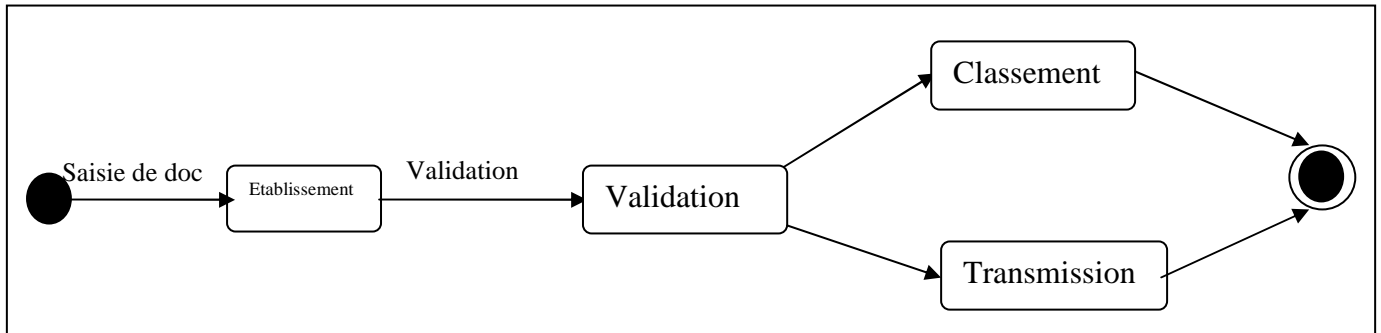
- **Etablissement** : le document passe dans cet état dès que le message de demande d'édition s'arrive, jusqu'à la fin de la saisie des informations nécessaires.
- **Validation** : Le document passe à l'état validation dès que le système se confirme.

## Chapitre VII : Analyse et conception

---

- **Transmission** : Lors de transmission de document aux postes correspondants
- **Classement** : le document passe à l'état classement dès qu'il sera sauvegardé dans le fichier correspondant.

### Diagramme d'états transitions



### Partie 3 : Conception

#### I. Identification des scénarios :

Un cas d'utilisation représente un ensemble de scénarios. Ces scénarios devraient être décrits de façon textuelle.

Pour formaliser la description d'un scénario, il est possible d'utiliser une forme simplifiée de diagramme de séquence.

Le diagramme de séquence montre les interactions entre objets en insistant sur la séquence des interactions. Il va donc permettre de décrire les interactions entre les acteurs et le système, sans rentrer dans la description du système. Autrement dit : un scénario représente un groupe de séquences et d'interactions entre le système et ses acteurs, le système est alors considéré comme une boîte noire.

##### I. 1. *Scénario de « Gestion des droits d'accès » :*

La séquence d'actions d'un scénario du cas d'utilisation «Administration » débute lorsque l'administrateur demande au système de créer un utilisateur ou de le mettre à jour.

#### Diagramme de séquence :

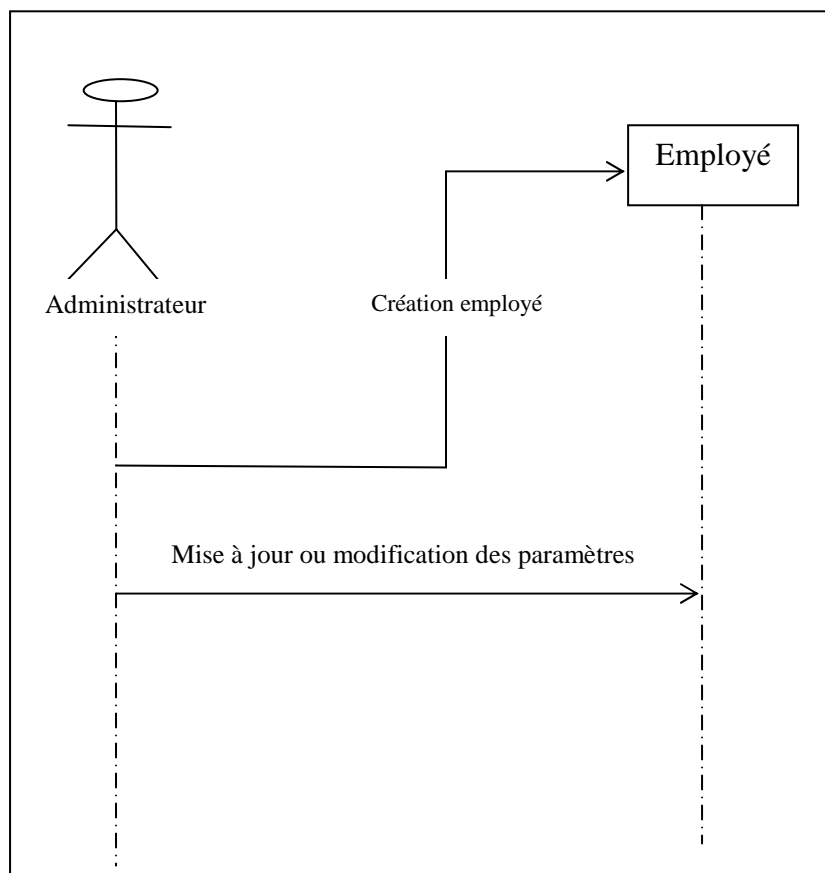


Figure 53 : diagramme séquence gestion « droits d'accès »

### I. 2. Scénario de « Identification Utilisateur » :

Le scénario débute lorsqu'un utilisateur veut accéder au système.

#### Diagramme de séquence

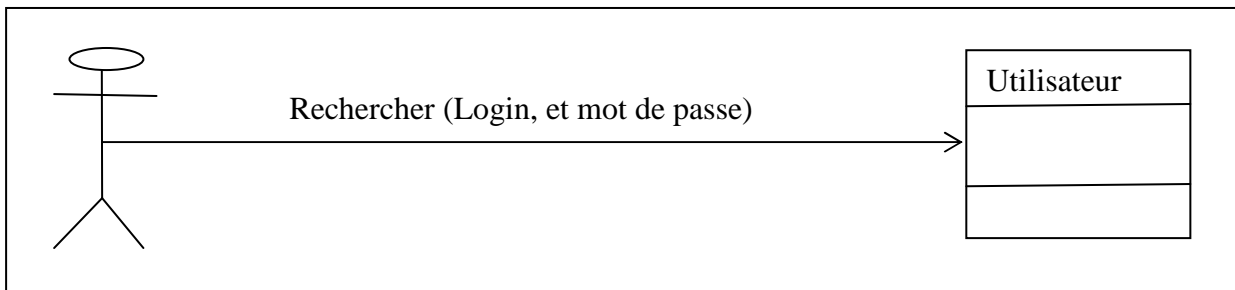


Figure 53 : diagramme séquence gestion « Identification des utilisateurs »

### I. 3. Scénario de « Approvisionnement Papier » :

Ce scénario commence lorsque le gestionnaire de stock papier décide de demander le papier

#### Diagramme de séquence

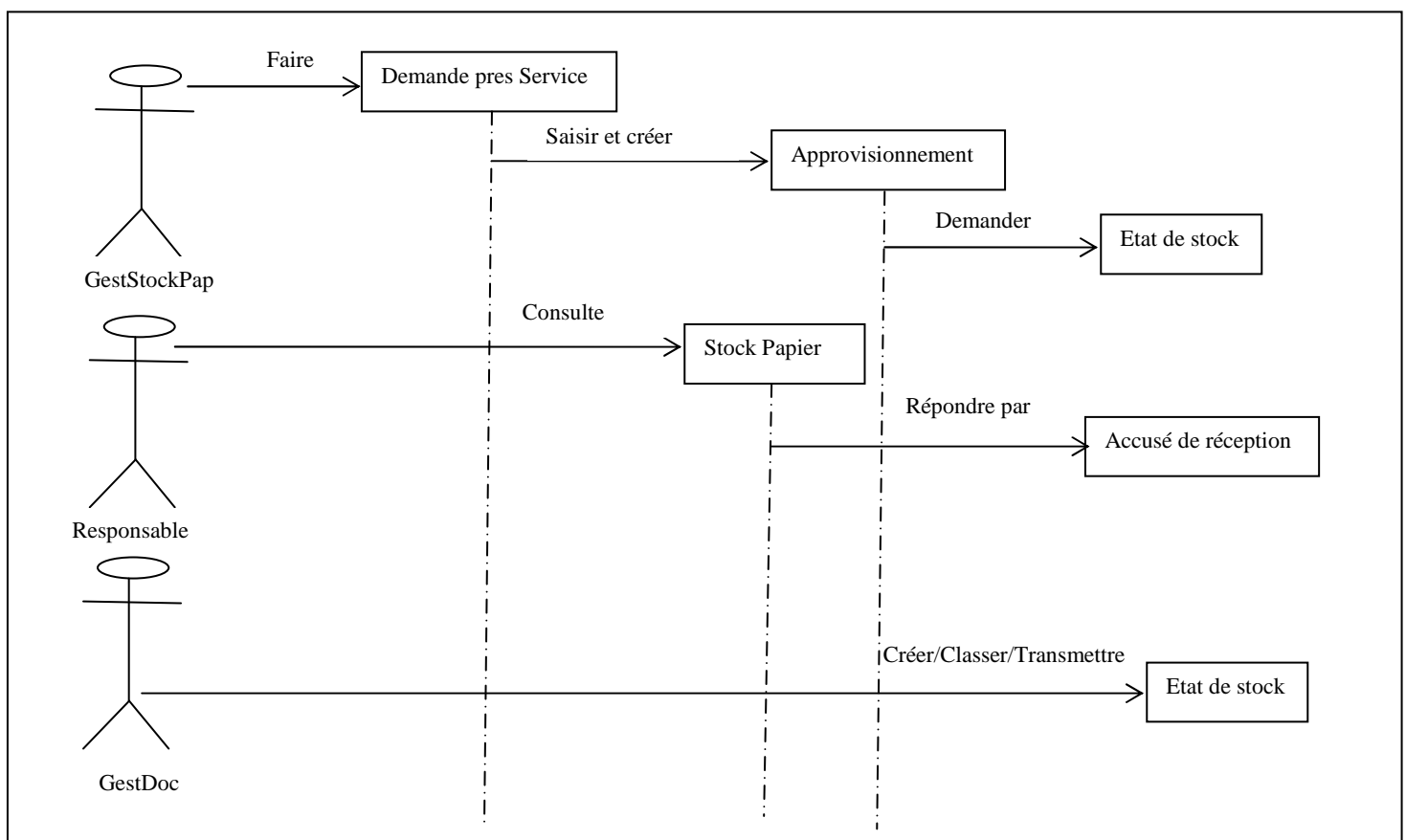


Figure 53 : diagramme séquence gestion « Approvisionnement papier »

### I. 4. Scénario de « Transfert Papier » :

Ce scénario commence lorsque les autres divisions de la direction sonelgaz ou bien les agences locaux demandent de papier à la DGSJ.

#### Diagramme de séquence

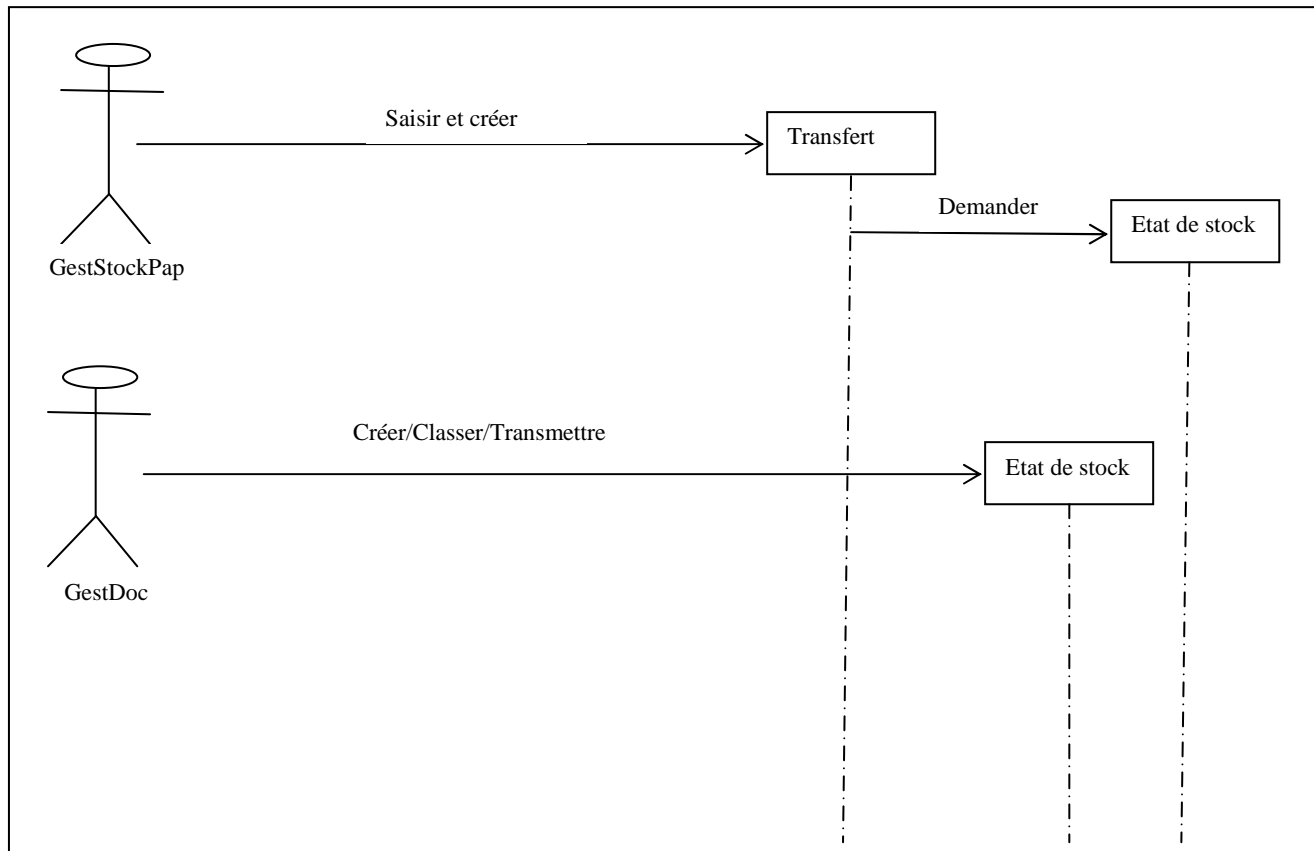
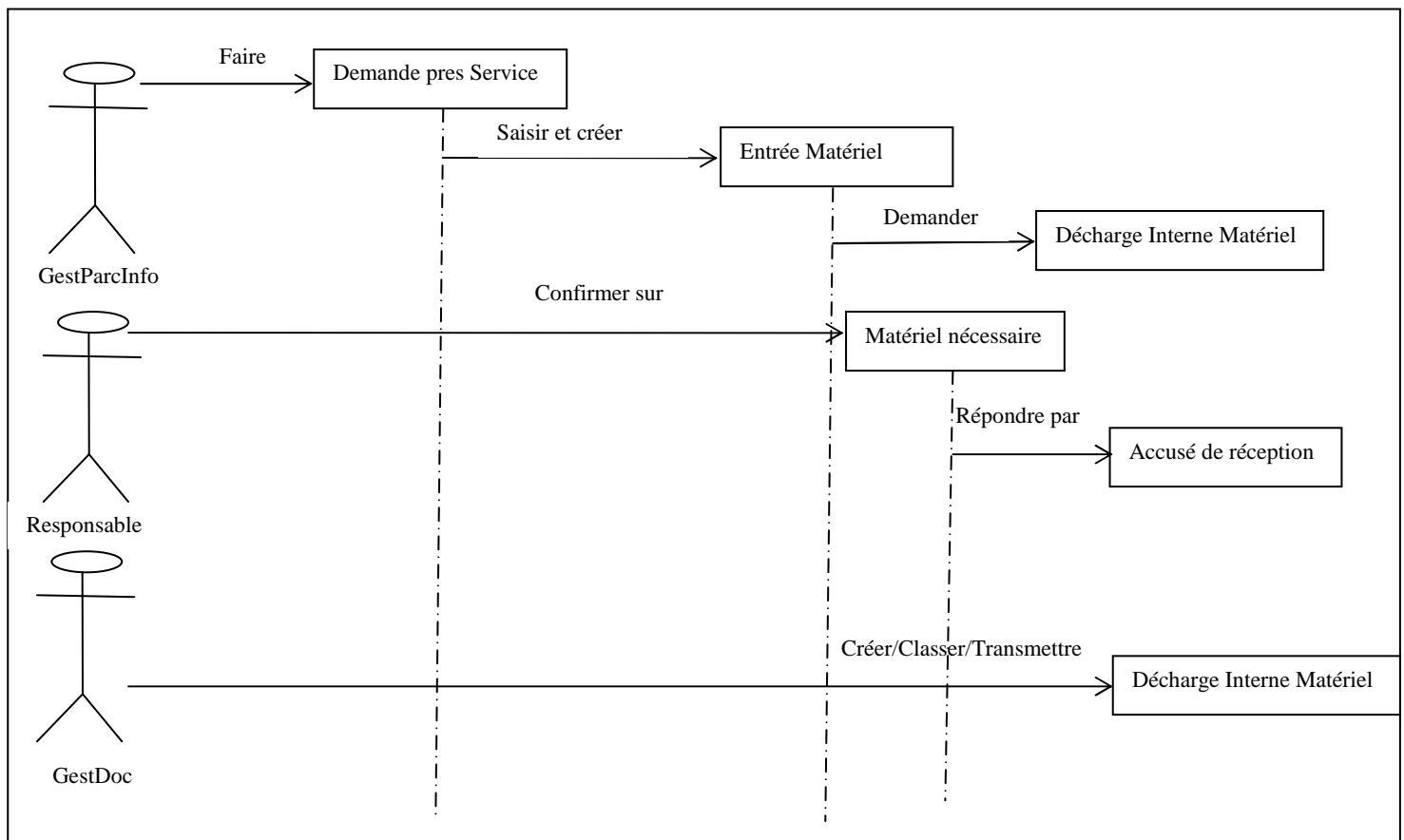


Figure 53 : diagramme séquence gestion « Transfert papier »

### I. 5. Scénario de « Recensement de matériel entré à la DGSJ » :

Ce scénario débute lorsque le gestionnaire de parc informatique veut avoir le matériel nécessaire à la DGSJ.

#### Diagramme de séquence

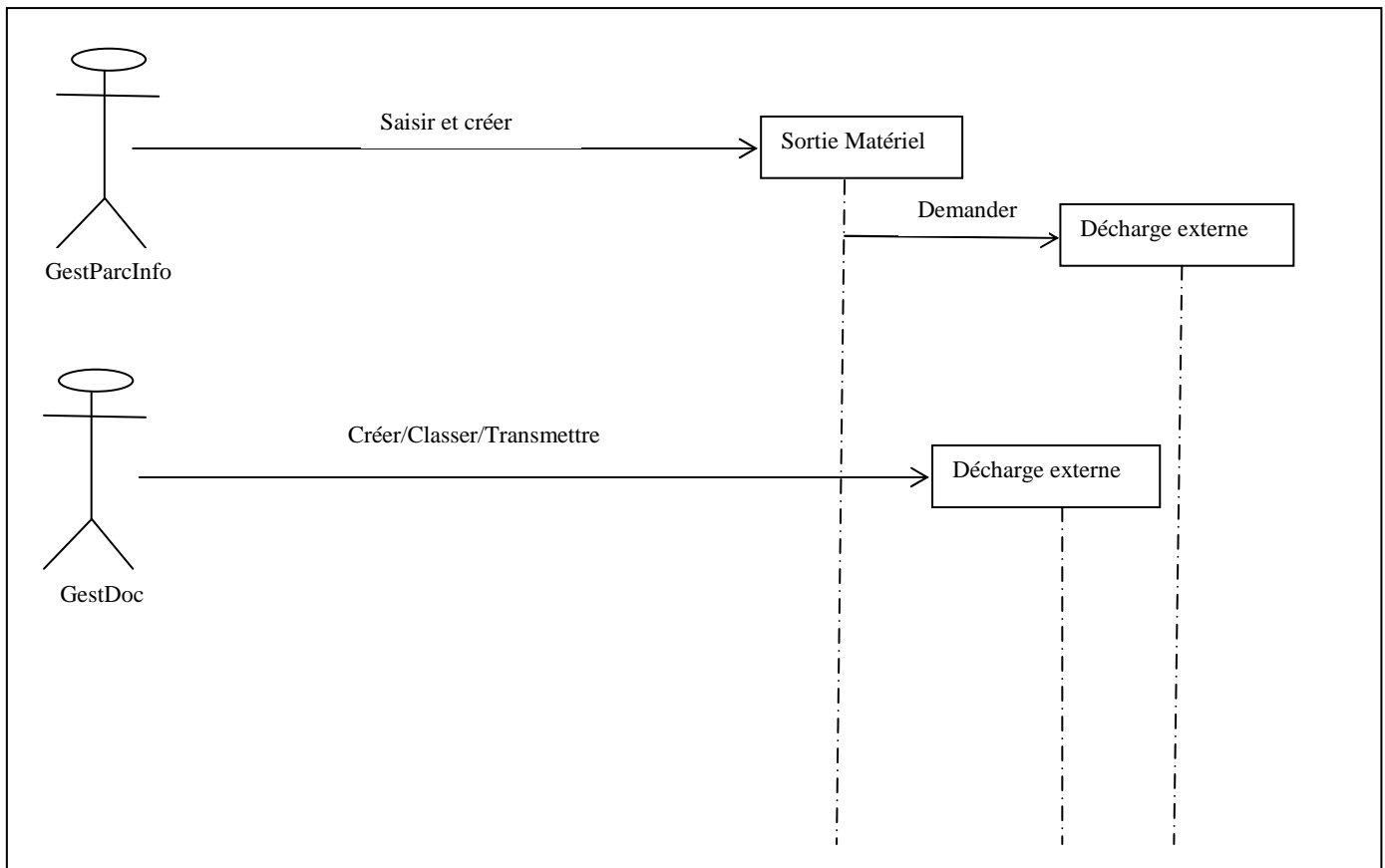


**Figure 53 : diagramme séquence gestion « Recensement de matériel entré à la DGSI »**

### **I. 6. Scénario de « Matériel sorti » :**

Ce scénario commence lorsque les autres divisions de la direction sonelgaz ou bien les agences locaux demandent de matériel à la DGSI.

### Diagramme de séquence



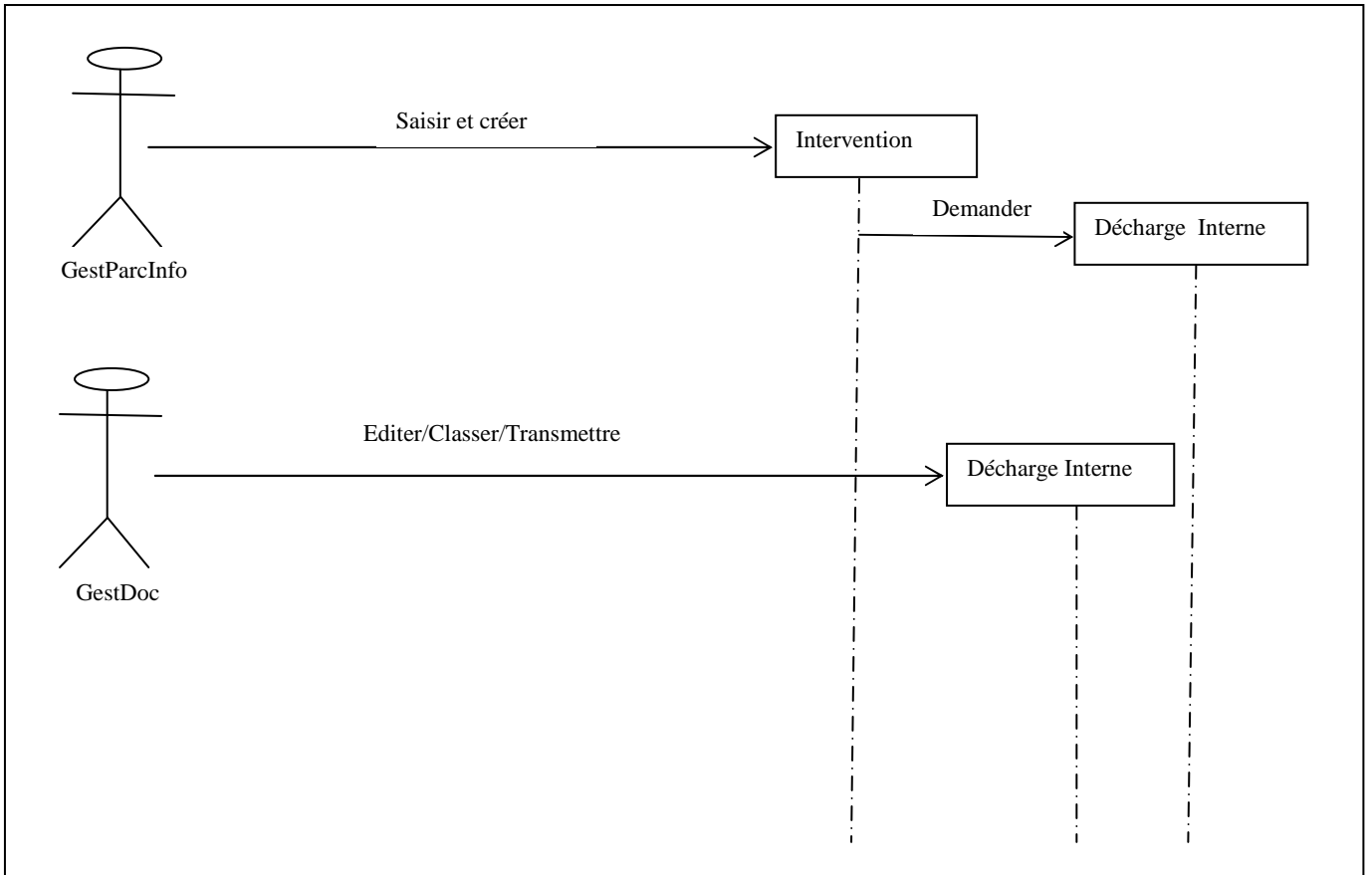
**Figure 53 : diagramme séquence gestion «Sortie matériel de la DGSI »**

#### **I. 7. Scénario de « Gestion des interventions » :**

Ce scénario débute à la réparation de matériel par les techniciens de la DGSI.

### Diagramme de séquence

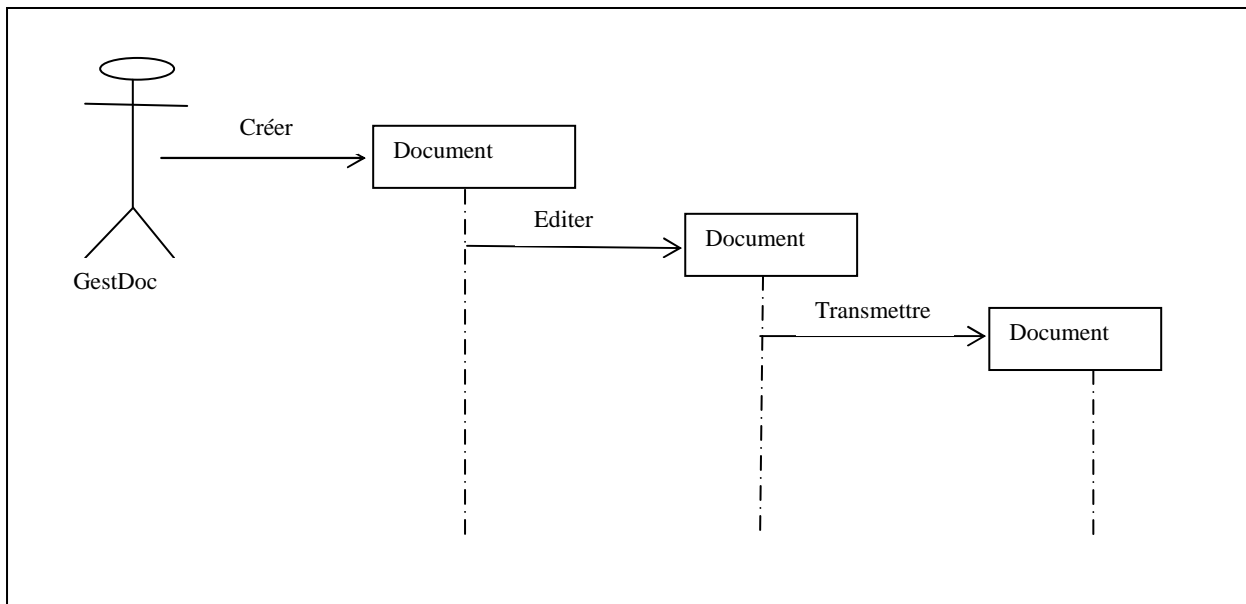




### I. 8. Scénario de « Gestion des documents » :

Ce scénario débute à la demande d'édition d'un document par les autres services.

#### Diagramme de séquence



## Chapitre VII : Analyse et conception

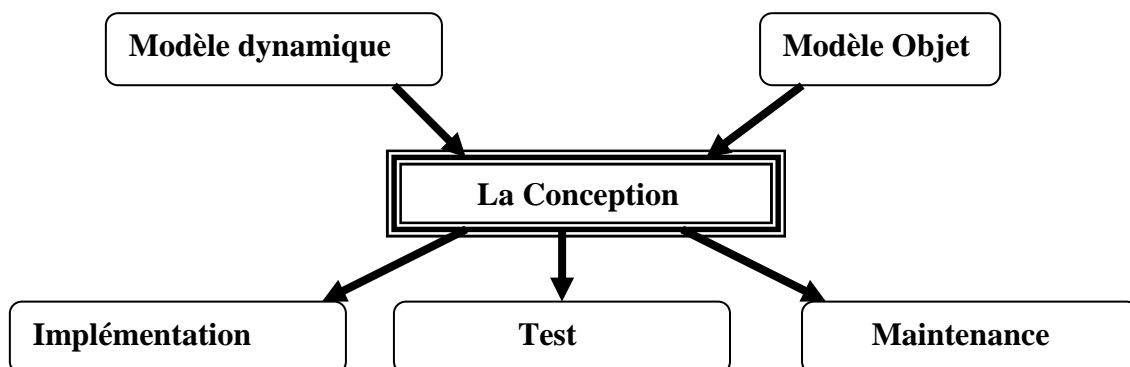
---

La conception est une solution de modélisation récurrente et documentée, applicable dans un contexte donné.

La phase de conception a pour objectif de s'accorder sur la manière dont le système doit être construit, avec des prises de décisions, dans un premier temps de haut niveau, puis à des niveaux de plus en plus détaillés. Donc s'accorde sur le comment, mais pas sur le quoi.

En d'autres termes, elle vise à trouver des solutions informatiques et techniques pour mettre en œuvre et construire le système analysé au cours des phases précédentes. Elle doit permettre d'élaborer les différentes couches de système analysé au cours de différentes couches du système et leurs interactions, d'abord à un niveau plus général puis à un niveau plus détaillé, en tenant compte des contraintes informatiques et techniques : langage, base de données, matériel, ...etc. selon le cycle de développement suivi. Cette phase succède à la phase d'analyse, qui se base sur les différents modèles de l'analyse (modèles statiques et le modèle dynamique). Elle fait intervenir à la fois les objets métiers revus et corrige pour les adapter à l'implémentation, ainsi que des objets techniques. Elle permet de transférer la connaissance vers l'implémentation qui est donc directement utilisée dans les étapes suivantes :

1. la phase de test
2. la phase de maintenance.



La conception de notre système passe par les étapes suivantes :

- Conception de nouveau système
- Conception des objets
- La sécurité du système

### **II. Conception du système :**

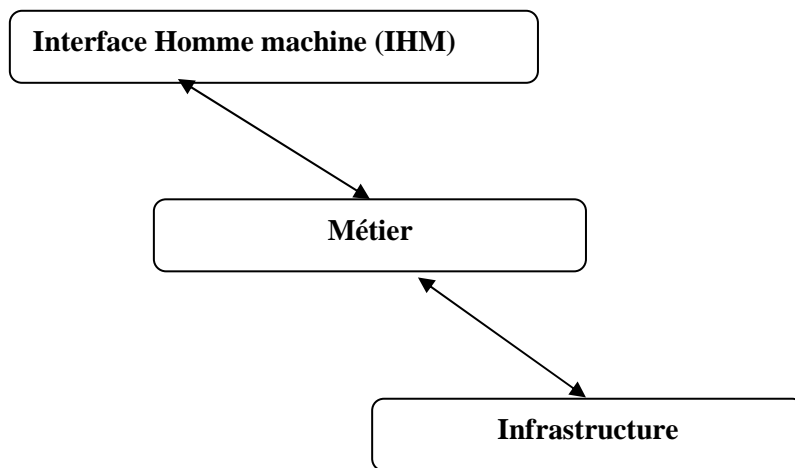
La conception du système s'étale de la conception des couches logicielles jusqu'au passage du modèle objet vers le modèle relationnel

### II. 1. La conception des couches logicielles

Une couche est un composant qui englobe des aspects communs à tout le système :

- Des fonctionnalités semblables
- Un même emplacement physique

Les couches de système ont été citées dans le paragraphe de la capture des besoins techniques dans un modèle en cinq couches, pour les détailler dans cette phase on a essayé d'englober les couches dans un autre modèle qui contient seulement trois couches. Dans ce modèle la couche présentation et la couche application ont été englobées dans une seule couche dite couche Interface Homme Machine (communément appelée IHM), la couches accès aux données et la couche stockage de données sont englobées dans une couches appelée Infrastructure. Le schéma suivant dans la présentation du système en trois couches :



#### II. 1.1. La couche Interface Homme Machine :

IHM est objet ou groupe d'objets fournissant aux utilisateurs d'un système un moyen cohérent d'accès aux objets du domaine, aux commandes et aux options de l'application

La couche IHM comprend les séquences d'écran permettant la communication entre l'application et l'utilisateur

Le tableau suivant présente les interfaces appropriées à chacun des utilisateurs (les utilisateurs et le système qui permettent la communication entre le système et ses acteurs)

## Chapitre VII : Analyse et conception

### Les interfaces utilisateurs

Poste de travail	Les interfaces utilisateur
L'administrateur	<ul style="list-style-type: none"><li>• Interface d'identification utilisateur</li><li>• Interface d'accueil</li><li>• Interface de création des employés</li><li>• Interface de mis à jours de profils employés</li></ul>
Poste de responsable de la DGSI	<ul style="list-style-type: none"><li>• Interface d'identification utilisateur</li><li>• Interface d'accueil</li><li>• Interface de saisie d'une réponse aux demandes et d'affichage des statistiques.</li></ul>
Poste de gestionnaire de stock papier	<ul style="list-style-type: none"><li>• Interface d'identification utilisateur</li><li>• Interface d'accueil</li><li>• Interface de saisie des approvisionnements et de stockage</li><li>• Interface de saisie des transferts et de stockage</li></ul>
Poste de gestionnaire de parc informatique	<ul style="list-style-type: none"><li>• Interface d'identification utilisateur</li><li>• Interface d'accueil</li><li>• Interface de saisie de matériel entré à la DGSI</li><li>• Interface de saisie de matériel sorti de la DGSI</li><li>• Interface de saisie des interventions</li></ul>
Poste de gestionnaire des documents	<ul style="list-style-type: none"><li>• Interface d'identification utilisateur</li><li>• Interface d'accueil</li><li>• Interface de création des documents</li></ul>

### II. 1.2. La couche métier :

La couche métier est constituée de l'ensemble des objets impliqués dans l'application. Cette couche fait l'objet de diagramme de classe qui sera détaillé dans le paragraphe suivant.

### II. 1.3 La couche Infrastructure :

Comprenant les couches de base du système telle que les bases de données, la gestion des architectures à objets distribués. Dans notre cas le SGBD demandé c'est « MySQL 5.4 ».

### II. 2. Affinage de diagramme des classes :

Dans cette phase on va procéder à l'affinage des classes, cet affinage se fait principalement par l'ajout des méthodes à chaque classe.

#### II. 2.1) Conception des méthodes :

Les méthodes d'une classe sont déduites du modèle d'analyse dynamique, chaque méthode est présentée dans le diagramme de collaboration par message provenant à cette classe.

Le tableau suivant représente la liste des méthodes de chacune des classes :

## Chapitre VII : Analyse et conception

### Les méthodes des classes objets :

Classe	Opérations
Document	Créer, Transmettre, Rechercher
Distributeur	Créer
Employé	Créer, modifier, Rechercher, Supprimer
Réponse_Demande	Créer, Afficher
Papier	Créer, Afficher
Matériel	Créer, Afficher
ApprovPap	Créer, Afficher, Rechercher
Livr_Mat	Créer, Afficher
Transf_Pap	Créer, Afficher, Rechercher
Sortie_Mat	Créer, Afficher
Intervention_Intr	Créer, Rechercher
Intervention_Extr	Créer, Rechercher

### 4) Méthode de classes d'associations :

Classe	désignation
Ligne_DemandePap	Créer, Afficher, Rechercher
Ligne_Demande_Mat	Créer, Afficher, Rechercher
DonneLieu_ApprovPap	Créer, Afficher, Rechercher
DonneLieu_LivrMat	Créer, Afficher, Rechercher
DonneLieu_TransfPap	Créer, Afficher, Rechercher
DonneLieu_SortieMat	Créer, Afficher, Rechercher
LigneApprov_Pap	Créer, Afficher, Recherche
LigneLivr_Mat	Créer, Afficher, Recherche
LigneTransf_Pap	Créer, Afficher, Recherche
LigneSortie_Mat	Créer, Afficher, Recherche

### III. Diagramme de classes :

#### 1. Liste des classes :

Classe	Attributs	Opérations
<b>Document</b>	<u>code_doc</u> , <u>id_emp</u> *, genre_doc, nom_doc, date_doc, Destinataire, motif_doc	Créer, Transmettre
<b>Distributeur</b>	<u>code_distributeur</u> , nom_distributeur	Créer
<b>Employé</b>	<u>id_emp</u> , nom_emp, Prenom_emp, date_naiss, fonction, adresse, login, pw, téléphone, actif	Créer, modifier, Afficer
<b>Réponse_Demande</b>	<u>Num réponse</u> , nom_réponse, type_réponse, date_réponse, destinataire, motif_réponse	Créer, Afficher

## Chapitre VII : Analyse et conception

---

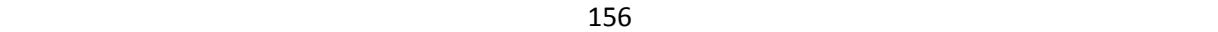
<b>Papier</b>	<u>code_papier</u> , <u>code_distributeur*</u> , type_papier, quantité_s_papier  date_Stock	Créer, Rechercher, Afficher
<b>Matériel</b>	<u>code_mat</u> , <u>code_distributeur*</u> , type_mat, Quantité_s_mat, Etat_matériel, Date	Créer, Rechercher, afficher
<b>ApprovPap</b>	<u>numapprov_pap</u> , date_approv_pap	Créer, Afficher
<b>Livr_Mat</b>	<u>numliv_rmat</u> , <u>datelivr_mat</u>	Créer, Afficher
<b>Transf_Pap</b>	<u>numtransf_pap</u> , <u>datetransf_pap</u>	Créer, Afficher
<b>Sortie_Mat</b>	<u>numsortie_mat</u> , <u>datesortie_mat</u>	Créer, Afficher
<b>Intervention_Intr</b>	num_intervention, date_intervention, motif_intervention, Structure	Créer, Afficher, Rechercher
<b>Intervention_Extr</b>	num_intervention, date_intervention, motif_intervention, Structure	Créer, Afficher, Rechercher



### 2. Liste des classes d'association :

Classe	Attributs	Opérations
<b>Ligne_DemandePap</b>	<u>code demande</u> , <u>code papier</u> , <u>quantité_d</u>	Créer, supprimer, modifier, Rechercher
<b>Ligne_Demande_Mat</b>	<u>code demande</u> , <u>code mat</u> , <u>quantité_d</u>	Créer, supprimer, modifier, Rechercher
<b>DonneLieu_ApprovPap</b>	<u>code demande</u> , <u>numapprov pap</u>	Créer, supprimer, modifier, Rechercher
<b>DonneLieu_LivrMat</b>	<u>code demande</u> , <u>numlivr mat</u>	Créer, supprimer, modifier, Rechercher
<b>DonneLieu_TransfPap</b>	<u>code demande</u> , <u>numtransf pap</u>	Créer, supprimer, modifier, Rechercher
<b>DonneLieu_SortieMat</b>	<u>code demande</u> , <u>numsorti mat</u>	Créer, supprimer, modifier, Rechercher
<b>LigneApprov_Pap</b>	<u>code pap</u> , <u>numapprov pap</u> , <u>quantité_approv</u>	Créer, modifier, Recherche
<b>LigneLivr_Mat</b>	<u>code_liv</u> , <u>numlivr_mat</u> , <u>quantité_livr</u>	Créer, modifier, Recherche
<b>LigneTransf_Pap</b>	<u>code pap</u> , <u>numtransf pap</u> , <u>quantité_transf</u>	Créer, modifier, Recherche
<b>LigneSortie_Mat</b>	<u>code_mat</u> , <u>numsortie_mat</u> , <u>quantité_sortie</u>	Créer, modifier, Recherche

### 3. Le diagramme des classes :



### IV. Le passage vers le modèle relationnel :

Une base de données est une collection de données sur un domaine d'application, elle sert à répondre aux besoins d'accès selon des multiples critères, aux besoins d'intégration, comme elle permet encore de montrer les relations existantes entre les données. Autrement dit une base de données une collection de classe qui nécessite la traduction en un ensemble de **tables relationnelles**.

Cette traduction a été basée sur un ensemble de règles de passage et a permis d'aboutir au modèle logique suivant. Le Modèle Logique des Données « **MLD** » consiste à transformer le diagramme de classes fonction des logiciels de gestion de base de données.

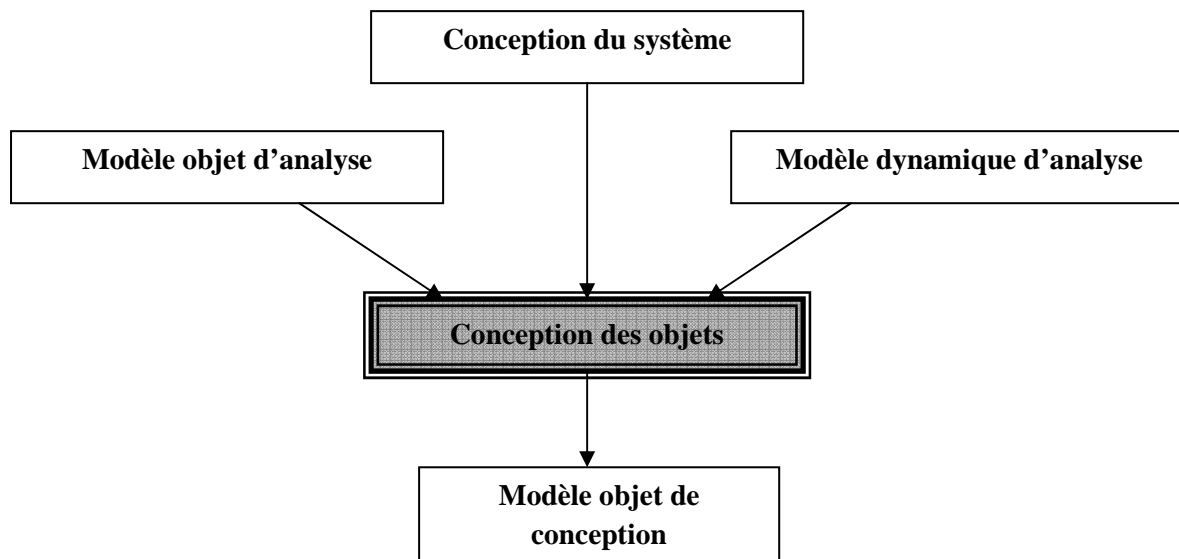
#### Liste des relations :

- Employés (id\_emp, nom\_emp, Prenom\_emp, date\_naiss, fonction, adresse, login, pw, téléphone, actif).
- Document (code\_doc, id\_emp \*, genre\_doc, nom\_doc, date\_doc, Destinataire, motif\_doc)
- Distributeur (code\_distributeur, nom\_distributeur, ....)
- Réponse\_Demande (Num\_réponse, nom\_réponse, type\_réponse, date\_réponse, destinataire, motif\_réponse)
- Papier (code\_papier, code\_distributeur\*, type\_papier, quantité\_s\_papier, Date)
- Matériel (code\_mat, code\_distributeur\*, type\_mat, Quantité\_s\_mat, etat\_matériel, date).
- Ligne\_DemandePap (code\_demande, code\_papier, quantité\_d)
- Ligne\_Demande\_Mat (code\_demande, code\_mat, quantité\_d)
- DonneLieu\_ApprovPap (code\_demande, numapprov\_pap)
- DonneLieu\_LivrMat (code\_demande, numlivr\_mat)
- DonneLieu\_TransfPap (code\_demande, numtransf\_pap)
- DonneLieu\_SortieMat (code\_demande, numsorti\_mat)
- ApprovPap (numapprov\_pap, date\_approv\_pap,)
- Livr\_Mat (numliv\_rmat, datelivr\_mat)
- Transf\_Pap (numtransf\_pap, datetransf\_pap,)
- Sortie\_Mat (numsortie\_mat, datesortie\_mat)
- Intervention\_Intr (num\_intervention, date\_intervention, motif\_intervention, Structure)
- Intervention\_Extr (num\_intervention, date\_intervention, motif\_intervention, structure)
- LigneApprov\_Pap (code\_pap, numapprov\_pap, quantité\_approv)
- LigneLivr\_Mat (code\_l\_mat, numlivr\_mat, quantité\_livr)
- LigneTransf\_Pap (code\_pap, numtransf\_pap, quantité\_transf)
- LigneSortie\_Mat (code\_mat, numsortie\_mat, quantité\_sortie)

### V. Conception des objets :

## Chapitre VII : Analyse et conception

La conception des objets s'appuie essentiellement sur les modèles de l'analyse et sur la conception du système



Les principales tâches de la conception objet visent à ajouter aux modèles objet de l'analyse, les détails liés à l'implémentation du système.

UML ne propose pas de formalisme particulier pour décrire un modèle de conception, il s'agit donc d'ajouter les éléments nécessaires au modèle d'analyse en fonction des options proposées par l'outil de modélisation utilisé.

### **VI. La sécurité :**

L'importance de la sécurité en informatique est plus élevée, on pourra ne jamais la négliger du fait qu'elle doit prendre en compte le système informatique et l'environnement (les personnes, les immeubles, les affaires, les objets de valeur et les menaces) dans lequel le système est utilisé.

La sécurité consiste à protéger les données stockées dans le système informatique contre les accès non autorisés, contre l'incohérence des données vis-à-vis des contraintes d'organisation, et contre toute panne matérielle ou logicielle se produisant dans le système de gestion de base de données ou les supports.

Notre application peut être victime de piratage ou de manipulation non autorisées. Il faut donc protéger la base de données d'une part et protéger le source de notre application d'autre part.

Pour cela, il faut donc protéger la base de donnée sonelgz et les accès direct à l'application, la sécurité vise donc deux module de l'application, l'accès à l'application elle même et l'accès à la base de donnée.

## Chapitre VII : Analyse et conception

---

**L'application** : Chaque utilisateur dispose d'une session propre à lui qui empêche toute intrusion non autorisée. Chaque utilisateur doit donc s'identifier avant d'avoir accès à sa propre session

**La base de données** : Il faut aussi protéger l'accès à la base de données, les bases de données sont des composants incontournables des serveurs web et des applications en ligne, qui fournissent du contenu dynamique. Des données secrètes ou critiques peuvent être stockées dans les bases de données : il est donc important de les protéger efficacement.

### **Les droits d'accès :**

Nous avons créé différents utilisateurs pour chaque aspect de notre application, avec des droits limités aux seules actions planifiées. Cela permet de limiter les dégâts s'il y a des intrus à la base de données avec l'un de ces jeux de droits, ils ne pourront pas donc affecter toute l'application.

Il y a là deux types de profil :

**L'administrateur** : qui a le droit d'accès à toutes les parties de l'application, et c'est à lui que revient le droit de créer des comptes et de définir leur droit d'accès dans le profil. En général, pour des raisons de sécurité il ne peut y avoir qu'un seul administrateur, cela évite les manipulations par plusieurs utilisateurs qui peuvent créer une confusion lors de la gestion de l'application.

**Gestionnaire** : ce profil est constitué d'un ensemble de membres présidés par un responsable, chargé de suivre leurs activités ; ayant tous les droits d'intervention.

### **Conclusion :**

Lors de l'étude conceptuelle on a procédé à spécifier en détail la solution informatique conforme aux choix de gestion et technique, alors c'est le temps de passer à la partie réalisation qui consiste à réaliser la solution informatique et passer au nouveau système d'information dans des meilleures conditions possibles.

# CHAITR VIII :

## Réalisation

### Section : I : Environnement et outils de développement

#### Introduction :

Après avoir conçu notre système, on arrive maintenant au dernier chapitre « Réalisation » qui décrit l'implémentation de la solution proposée durant la conception.

Ce chapitre réalisation sera développé en trois sections bien précises, on commencera par la description de l'environnement et les outils de développement, puis on passe à la présentation de l'architecture de notre application, ensuite les fonctionnalités de l'application, enfin on terminera par une conclusion.

#### I. 1. L'environnement de développement :

##### I. 1.1. Le langage de programmation java :

Java est un langage de programmation récent le plus puissant actuellement sur le marché mondiale, java a été créé en 1991 par Sun, mais il a été créé accidentellement car Sun voulait créer un petit programme qui pouvait tourner sur les plates formes différentes pour pouvoir contrôler des boîtes de commutation de télévision. Alors ils ont pensé que les gens avaient besoin de programmer ces boîtes de commutation ; pour pouvoir les programmer il devait concevoir un programme qui pouvait tourner sur plusieurs plates formes différentes. Vers la fin de 1995 java a obtenu un très grand succès parce que c'est un langage qui pouvait travailler sur plusieurs plates formes (Windows, Unix, Linux, Mac, Solaris,...). Donc java est apparu en fin d'années 1995 et début 1996 et développé par Sun Microsystems, inspiré des langages C et C++ dont il hérite partiellement la syntaxe mais non ses défauts, et fait partie de la « famille » des *langages orientés objets*. Il répond donc aux trois principes fondamentaux de l'approche orientée objet (POO) : l'**encapsulation**, le **polymorphisme** et l'**héritage**. Au départ, java est destiné à la programmation de centraux téléphoniques sous l'appellation de langage "oak". La société Sun a eu l'idée de le recentrer sur les applications de l'internet et des réseaux ; Java C'est un langage en évolution permanente Java 2 est la version stabilisée de java fondée sur la version initiale 1.2.2 du JDK (Java Development Kit de Sun : <http://java.sun.com>). Encore il s'est très rapidement taillé une place importante en particulier dans le domaine de l'internet et des applications client-serveur.

#### ➤ **Les objectifs de langage java :**

Java a été développé dans le but d'augmenter la productivité des programmeurs. Pour cela, ses quatre raisons principales de développement ont été mises en place :

- C'est un langage orienté objet dérivé du C, mais plus simple à utiliser et plus « pur » que le C++. On entend par « pur » le fait qu'en Java, on ne peut faire que de la programmation orienté objet contrairement au C++ qui reste un langage hybride, c'est-à-dire autorisant plusieurs styles de programmation. C++ est hybride pour assurer une compatibilité avec le C.
- Il est doté, en standard, de bibliothèques de classes très riches comprenant la gestion des interfaces graphiques (fenêtres, boîtes de dialogue, contrôles, menus, graphisme), la programmation multi-threads (multitâches), la gestion des exceptions, les accès aux

fichiers et au réseau ... L'utilisation de ces bibliothèques facilitent grandement la tâche du programmeur lors de la construction d'applications complexes.

- Il est doté, en standard, d'un mécanisme de gestion des erreurs (les exceptions) très utile et très performant. Les exceptions représentent le mécanisme de gestion des erreurs intégré au langage Java. Il se compose d'objets représentant les erreurs et d'un ensemble de trois mots clés qui permettent de détecter et de traiter ces erreurs ( try, catch et finally ) et de les lever ou les propager (throw et throws). Ce mécanisme, inexistant en C, existe en C++ sous la forme d'une extension au langage beaucoup moins simple à utiliser qu'en Java. Ces mécanismes permettent de renforcer la sécurité du code Java
- Il est multi plates-formes : les programmes tournent sans modification sur tous les environnements où Java existe (Windows, Unix et Mac). c-à-dire c'est un langage de programmation qu'on puisse compiler et exécuter sur énormément de plates formes.

### ➤ Les caractéristiques de langage java :

✚ **Java est interprété** : Le code source est compilé en pseudo code ou Byte code puis exécuté par un interpréteur Java qui est : la Java Virtual Machine (JVM). Ce concept est à la base du slogan de Sun pour Java : WORA (Write Once, Run Anywhere : écrire une fois, exécuter partout). En effet, le Byte code, s'il ne contient pas de code spécifique à une plate-forme particulière peut être exécuté et obtenir les mêmes résultats sur toutes les machines disposant d'une JVM. Autrement dit le Byte code est normalisé, peut être exécuté sur plusieurs plates formes, et exploitable par tous les systèmes d'exploitations.

✚ **Java est portable : il est indépendant de toute plate-forme** : il n'y a pas de compilation spécifique pour chaque plate forme. Le code reste indépendant de la machine sur laquelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine. Cette indépendance est assurée au niveau du code source grâce à Unicode et au niveau du byte code.

✚ **Java est orienté objet** : comme la plupart des langages récents, Java est orienté objet. Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées les unes avec les autres pour former une application. Java n'est pas complètement objet car il définit des types primitifs (entier, caractère, flottant, booléen,...).

✚ **Java est simple** : le choix de ses auteurs a été d'abandonner des éléments mal compris ou mal exploités des autres langages tels que la notion de pointeurs (pour éviter les incidents en manipulant directement la mémoire), l'héritage multiple et la surcharge des opérateurs, ...

✚ **Java est fortement typé** : toutes les variables sont typées et il n'existe pas de conversion automatique qui risquerait une perte de données. Si une telle conversion doit être réalisée, le développeur doit obligatoirement utiliser un cast ou une méthode statique fournie en standard pour la réaliser.



✚ **Java assure la gestion de la mémoire** : l'allocation de la mémoire pour un objet est automatique à sa création et Java récupère automatiquement la mémoire inutilisée grâce au garbage collector qui restitue les zones de mémoire laissées libres suite à la destruction des objets.

✚ **Java est robuste et sûr** :

- Peu de pièges.
- Pas de pointeurs, pas de fonctions d'arguments variables.
- Compilateurs très stricts car toutes les valeurs doivent être initialisées.
- Le traitement des exceptions est obligatoire.
- Les erreurs à l'exécution sont vérifiées tout comme les limites des tableaux.

✚ **Java est économe** : le pseudo code a une taille relativement petite car les bibliothèques de classes requises ne sont liées qu'à l'exécution.

✚ **Java est multitâche** : il permet l'utilisation de threads qui sont des unités d'exécution isolées. La JVM, elle même, utilise plusieurs threads. Un thread est une unité d'exécution faisant partie d'un programme. Cette unité fonctionne de façon autonome et parallèlement à d'autres threads. En fait, sur une machine mono processeur, chaque unité se voit attribuer des intervalles de temps au cours desquels elles ont le droit d'utiliser le processeur pour accomplir leurs traitements.

Le principal avantage des threads est de pouvoir répartir différents traitements d'un même programme en plusieurs unités distinctes pour permettre leur exécution "simultanée".

La classe `java.lang.Thread` et l'interface `java.lang.Runnable` sont les bases pour le développement des threads en java. Par exemple, pour exécuter des applets dans un thread, il faut que celles ci implémentent l'interface `Runnable`.

### *Pourquoi le langage java ?*

Notre choix est basé sur le langage de programmation **java**, d'une part parce qu'il est un langage de programmation orientée objet qui n'est qu'un ensemble d'objets autonomes et responsables, qui s'entraînent pour résoudre un problème final en s'envoyant des messages, d'autre part les principes de la programmation orientée objet sont nettement plus élaborés dans le langage **java** ; Bien sur ne sont pas spécifiques à **java**, mais de nombreux programmeurs les découvrent au même temps que le langage **java**. Comme on pourra justifier notre choix encore, par l'adéquation de **java** aux concepts du monde réel ; en **java** tout est objet ! **Java** est le rare langage qui est proche aux concepts de la vie réelle

### I. 1.1XML :

XML est l'acronyme de «eXtensible Markup Language». Sous-ensemble simplifié du langage SGML (Standard Generalized Markup Language), permet de décrire des informations organisées sous une forme hiérarchique à l'aide de balises entre < > de façon similaire au langage HTML.

XML permet d'échanger des données entre applications hétérogènes car il permet de modéliser et de stocker des données de façon portable.

XML est extensible dans la mesure où il n'utilise pas de tags prédéfinis comme HTML et il permet de définir de nouvelles balises : c'est un métalangage.

Le format HTML est utilisé pour formater et afficher les données qu'il contient : il est destiné à structurer, formater et échanger des documents d'une façon la plus standard possible..

XML est utilisé pour modéliser et stocker des données. Il ne permet pas à lui seul d'afficher les données qu'il contient.

Pourtant, XML et HTML sont tous les deux des dérivés d'un langage nommé SGML (Standard Generalized Markup Language). La création d'XML est liée à la complexité de SGML. D'ailleurs, un fichier XML avec sa DTD correspondante peut être traité par un processeur SGML.

XML et Java ont en commun la portabilité réalisée grâce à une indépendance vis à vis du système et de leur environnement.

Au premier abord, XML est un langage assez déroutant car il nous laisse la liberté de décider du nom des balises à utiliser et n'impose que peu de règles de construction. Cette souplesse lui a permis d'être adopté très rapidement par la communauté informatique, qui voit XML comme un des meilleurs formats pour structurer lisiblement des données et les échanger dans le monde ouvert d'Internet.

Un document XML débute généralement par le prologue :

```
<? Xml version="1.0"?>
```

Ce prologue peut comporter aussi un attribut encoding qui spécifie le type d'encodage des caractères utilisé dans le document ; comme encoding est égal par défaut à UTF-8, la ligne précédente est donc équivalente à :

```
<? Xml version="1.0" encoding="UTF-8"?>
```

Suivent ensuite les données du document décrites sous une forme arborescente d'*éléments* inclus les uns dans les autres, un document XML ne devant contenir qu'un seul élément racine. Chaque *élément* commence par une *balise (tag)* de début notée entre les symboles < et > et se termine par une balise de fin notée entre les symboles </ et >. Chaque balise qui porte le nom de votre choix, est utilisée pour nommer un élément et décrire ses données avec :

- une suite d'*attributs* dans sa balise de début, chaque attribut s'écrivant avec un nom suivi du signe = et d'une valeur entre ' ' ou entre " " (par exemple <messages sujet="chat" >) ;
- et/ou des *données textuelles* et/ou d'autres éléments *enfants* (*child*) entre ses balises de début et de fin.

### I. 1.3.IDE (NetBeans 6.7) :

C'est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et des pages web). NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X et Open VMS.

NetBeans est à l'origine un EDI Java. NetBeans fut développé à l'origine par une équipe d'étudiants à Prague, racheté ensuite par Sun Microsystems. Quelque part en 2002, Sun a décidé de rendre NetBeans open-source. Mais NetBeans n'est pas uniquement un EDI Java. C'est également une plateforme, nous permettant d'écrire nos propres applications Swing. Sa conception est complètement modulaire : Tout est module, même la plateforme. Ce qui fait de NetBeans une boîte à outils facilement améliorable ou modifiable. La license de NetBeans permet de l'utiliser gratuitement à des fins commerciales ou non. Elle permet de développer tout types d'applications basées sur la plateforme NetBeans. Les modules que nous pourrions écrire peuvent être open- source comme ils peuvent être closed-source, Ils peuvent être gratuits, comme ils peuvent être payants.

### *Pourquoi l'IDE Net Beans ?*

#### *Signification et définition d'un IDE :*

**IDE** (Integrated Development Environment) = (Environnement de développement Intégré)

Un IDE est un outil qui nous permettra d'augmenter notre productivité pour les développements d'entreprise.

On a doté de choisir l'IDE « *Net Beans* », grâce à son adéquation aux développements d'entreprise, il nous permettra de créer une application sans écrire une ligne de code, en utilisant des composants prêts et en générant automatiquement le code nécessaire.

### **I. 1.4. Le serveur d'application (Glassfish V2.1) :**

Glassfish est le nom du *serveur d'applications* **"Open Source"** *Java EE 5* et désormais *Java EE 6* avec la version 3 et qui sert de fondation au produit "Oracle GlassFish Server" (anciennement *Sun Java System Application Server* de ***Sun Microsystems***). Sa partie *Toplink persistence* provient d'**Oracle**. C'est la réponse aux développeurs Java désireux d'accéder aux sources et de contribuer au développement des serveurs d'applications de nouvelle génération.

### **I. 1.5. Le SGBD (MYSQL 5) :**

#### ➤ **Présentation de système de gestion de base de données MySQL :**

MySQL, le plus populaire des serveurs de bases de données SQL Open Source, est développé, distribué et supporté par **MySQL AB**. **MySQL AB** est une société commerciale, fondée par les développeurs de MySQL, qui développent leur activité en fournissant des services autour de MySQL.

- **MYSQL est un système de gestion de base de données SQL :**

Il se compose d'un langage de définition de données et de droits ainsi qu'un langage de manipulation des données. En effet, si (on a besoin d'un système de gestion de bases de données tel que le serveur MySQL pour ajouter, lire et traiter des données dans une base de données et comme les ordinateurs sont très bons à manipuler de grandes quantités de données) alors le SGBD MySQL joue un rôle central en informatique.

- **MYSQL est disponible sous Windows et sous Unix.**

- **MySQL est un serveur de bases de données relationnelles :**

Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données. SQL est le plus populaire langage de base de données dans le monde. SQL est un langage standardisé qui rend facile le stockage, la mise à jour et l'accès à l'information. Par exemple, on peut utiliser le SQL pour récupérer des informations sur un produit ou stocker des informations client sur un site web.

- **MySQL** est un véritable serveur (ou gestionnaire) de base de données SQL multi-utilisateur et multi-threadé.
- **MySQL** est une configuration client/serveur.
- **MySQL** est suffisamment rapide et flexible pour gérer des historiques et des images.

### ➤ Histoire de MySQL :

Nous ont débuté avec l'intention d'utiliser mSQL pour se connecter aux tables en utilisant leurs propres routines bas niveau ISAM. Cependant, après quelques tests, ils sont arrivés à la conclusion que mSQL n'était pas assez rapide et flexible pour leurs besoins. Cela les a conduit à créer une nouvelle interface SQL pour leur base de données, mais en gardant la même API que mSQL. Cette API a été choisie pour la facilité de port des programmes de tiers.

Les liens avec le nom MySQL ne sont pas parfaitement établis. Leur dossier de base et un grand nombre de bibliothèques et outils étaient préfixés par ``my" depuis plus de 10 ans. Mais la fille de Monty, plus jeune que lui, était aussi appelée My. Lequel des deux a conduit au nom de MySQL est toujours un mystère, même pour eux.

Le nom du dauphin MySQL (notre logo) est Sakila, qui a été choisi par les fondateurs de MySQL AB à partir d'une grande liste de noms suggérés par les utilisateurs dans le concours "Name the Dolphin" ("Nommez le dauphin"). Le nom a été suggéré par Ambrose Twebaze, un développeur de logiciels libres au Swaziland, en Afrique. D'après Ambrose, le nom Sakila puise ses origines du SiSwati, la langue locale du Swaziland. Sakila est aussi le nom d'une ville en Arusha, Tanzanie, près du pays d'origine d'Ambrose, Uganda.

### I. 1.6. Le Framework (Hibernate) :

Hibernate est un framework open source gérant la *persistance* des *objets* en *base de données relationnelles*.

Hibernate est adaptable en termes d'architecture, il peut donc être utilisé aussi bien dans un *développement* client lourd, que dans un environnement web léger de type *Apache Tomcat* ou dans un environnement *J2EE* complet : *WebSphere*, *JBoss Application Server* et *Oracle WebLogic Server*.

Hibernate apporte une solution aux problèmes d'adaptation entre le *paradigme objet* et les *SGBD* en remplaçant les accès à la base de données par des appels à des *méthodes* objet de haut niveau.

### Section II : Architecture de notre application :

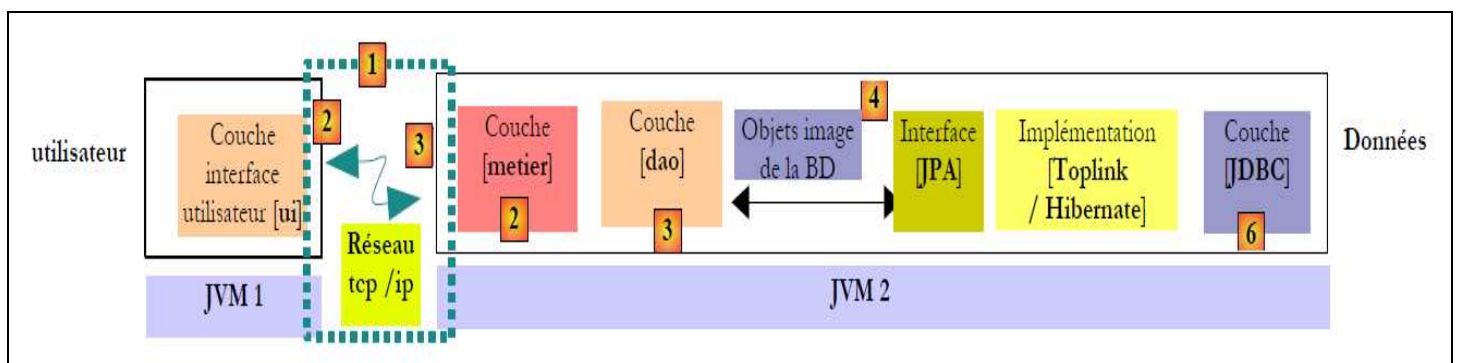
#### II. 1. Introduction :

J2EE est un patron de conception qui consiste en plusieurs type de projets différents qu'on va présenter tous par la suite dans la présentation de l'architecture de l'application. Par patron de conception on entend la description d'un problème récurrent et ses solutions, les solutions ne sont jamais exactement les mêmes pour chaque occurrence du problème mais le patron de conception donne une solution générale au problème qu'il suffit d'adapter.

#### II. 2. Architecture de notre application :

Pour la réalisation proprement dite de notre système, on a opté pour la technologie **Swing** pour la couche graphique cliente avec un support **spring**

Notre application suivra donc l'architecture JEE en couche suivante :



D'après cette figure on peut dire que notre application est une application client/serveur multi-tiers. Elle est constituée essentiellement de trois grandes parties :

- **Une partie stockage :** cette partie contient une base de données qui nous permet de stocker et sauvegarder nos données.
- **Une partie traitement :** dont laquelle on pourra effectuer les différents traitements sur nos données, cette dernière sera décomposée selon l'architecture JEE utilisées aux couches suivantes :
  - La couche [2], appelée ici [metier] est la couche qui applique les règles dites métier, c.à.d la logique spécifique de l'application, sans se préoccuper de savoir d'où viennent les données qu'on lui donne, ni où vont les résultats qu'elle produit.

- La couche [3], appelée ici [dao] (Data Access Object) est la couche qui fournit à la couche [2] des données pré-enregistrées (fichiers, bases de données, ...) et qui enregistre certains des résultats fournis par la couche [2].

La couche [4] des objets, image de la BD est appelée "contexte de persistance". Une couche [dao] s'appuyant sur Hibernate fait des actions de persistance (CRUD, create - read - update - delete) sur les objets du contexte de persistance, actions traduites par Hibernate en ordres SQL exécutés par la couche Jdbc. Pour les actions d'interrogation de la base (le SQL Select), Hibernate fournit au développeur, un langage HQL (Hibernate Query Language) pour interroger le contexte de persistance [4] et non la BD elle même.

- La couche [5], appelée ici [Hibernate] se place entre la couche [dao] écrite par le développeur et la couche [JDBC]. Hibernate est un ORM (Object Relational Mapping), un outil qui fait le pont entre le monde relationnel des bases de données et celui des objets manipulés par Java. Le développeur de la couche [dao] ne voit plus la couche [Jdbc] ni les tables de la base de données dont il veut exploiter le contenu. Il ne voit que l'image objet de la base de données, image objet fournie par la couche [Hibernate]. Le pont entre les tables de la base de données et les objets manipulés par la couche [dao] est fait principalement de deux façons :
  - par des fichiers de configuration de type XML
  - par des annotations Java dans le code, technique disponible seulement depuis le JDK 1.5

La couche [Hibernate] est une couche d'abstraction qui se veut la plus transparente possible. L'idéal visé est que le développeur de la couche [dao] puisse ignorer totalement qu'il travaille avec une base de données. C'est envisageable si ce n'est pas lui qui écrit la configuration qui fait le pont entre le monde relationnel et le monde objet.

- La couche [6], appelée ici [JDBC] est la couche standard utilisée en Java pour accéder à des bases de données. C'est ce qu'on appelle habituellement le pilote Jdbc du SGBD (dans notre application c'est MySQL).
- **Une partie présentation :** Comme son nom l'indique, cette partie s'occupe de la présentation graphique (interfaces graphiques) du système pour les utilisateurs; elle est implémentée par des clients java.

### **II. 3 : Implémentation des différentes parties de l'application**

#### **II. 3.1: Partie stockage :**

Pour l'implémentation de cette partie, Nous avons créé avec le SGBD MYSQL une base de données [dbsonelgaz] contenant les différentes tables.

Pour créer les tables on a utilisé le script [MySQL Command Line Client] fourni avec le SGBD MySQL. Voici un exemple de script de création d'une des tables

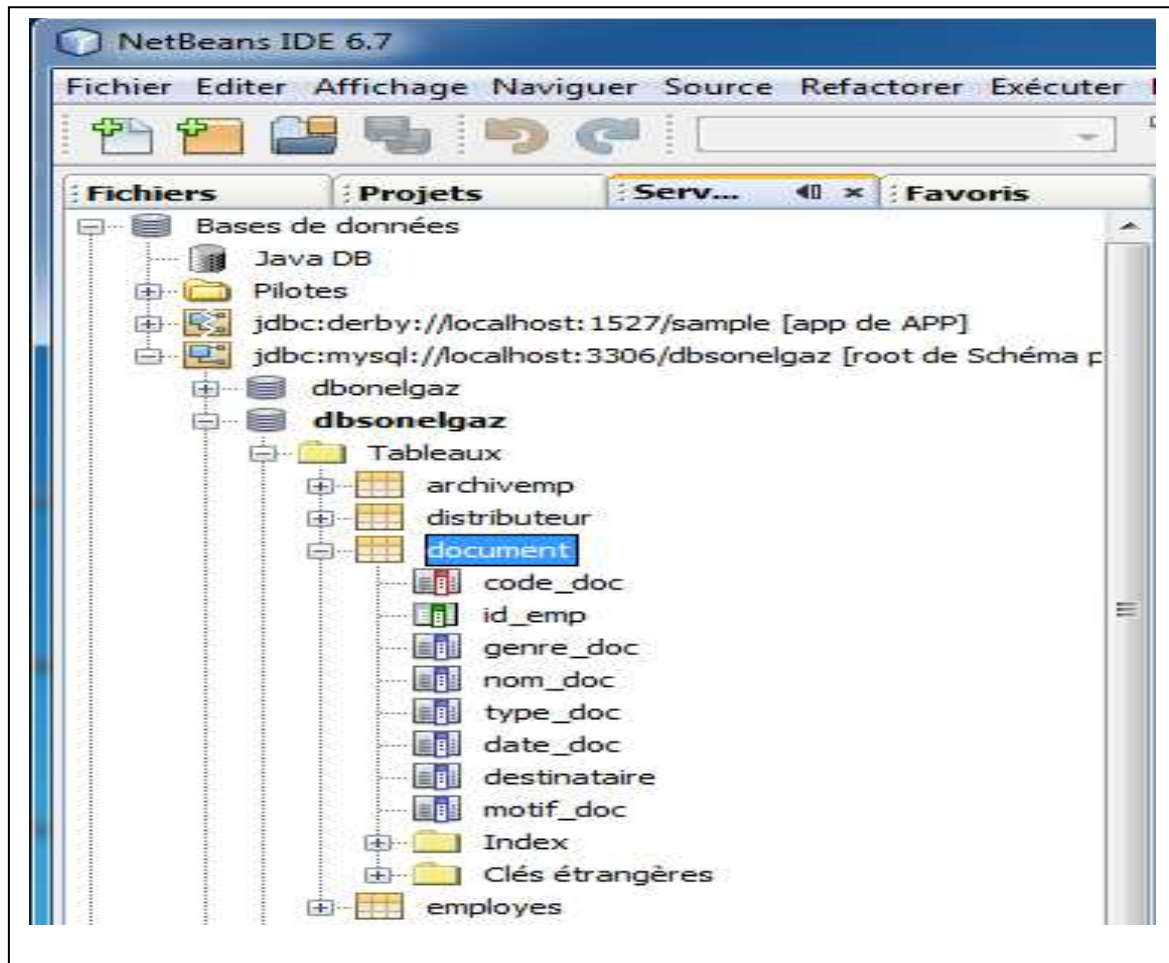
```
CREATE TABLE `document` (  
  `code_doc` varchar(15) NOT NULL,  
  `id_emp` int(10) NOT NULL,  
  `genre_doc` varchar(20) NOT NULL,  
  `nom_doc` varchar(20) NOT NULL,  
  `type_doc` varchar(10) NOT NULL,  
  `date_doc` date NOT NULL,  
  `destinataire` varchar(30) NOT NULL,  
  `motif_doc` varchar(30) NOT NULL,  
  PRIMARY KEY (`code_doc`),  
  CONSTRAINT id_empfk FOREIGN KEY (id_emp) REFERENCES employes(id_emp)  
) ENGINE=InnoDB;
```

**Figure II.3.1: script de la création de la table Document**

Après avoir créé cette base de données, on a établie une connexion NetBeans /MYSQL à l'aide du pilote Jdbc de MySQL.

L'image suivante est une capture d'écran montrant la connexion établie, ainsi que les différentes tables de notre BDD.

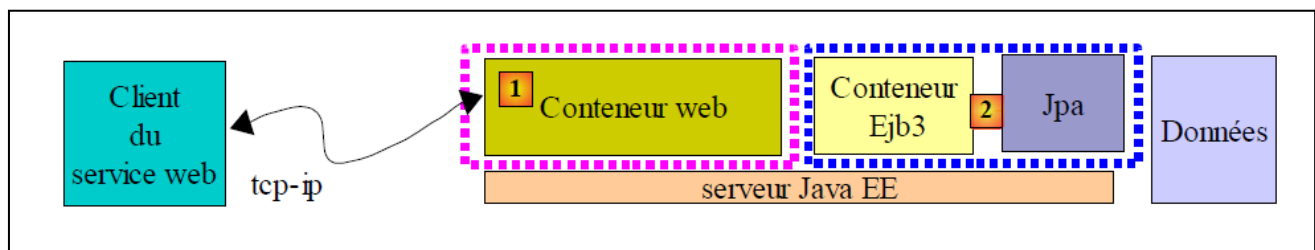




### II. 3.2. Partie Traitement :

Dans cette section, nous allons faire une brève description des différents projets NetBeans implémentant cette partie, ainsi les différentes couches JEE qui la constituent.

#### II. 3.2.1. le projet EAGestionSonelgaz :



Ce projet est un projet **JEE** de type Entreprise Application, c'est lui qui nous a permis le déploiement de notre application client/serveur 3-tiers au sein du serveur Glassfish v2.1

Ce dernier est composé de deux modules :

- Un module EJB qui implémente les couches [JDBC], [JPA/Hibernate], [DAO] de notre application.
- Un module de type web application qui implémente la couche [metier] et déploie nos Pdf.

La figure suivante est une capture d'écran montrant ce projet NetBeans.

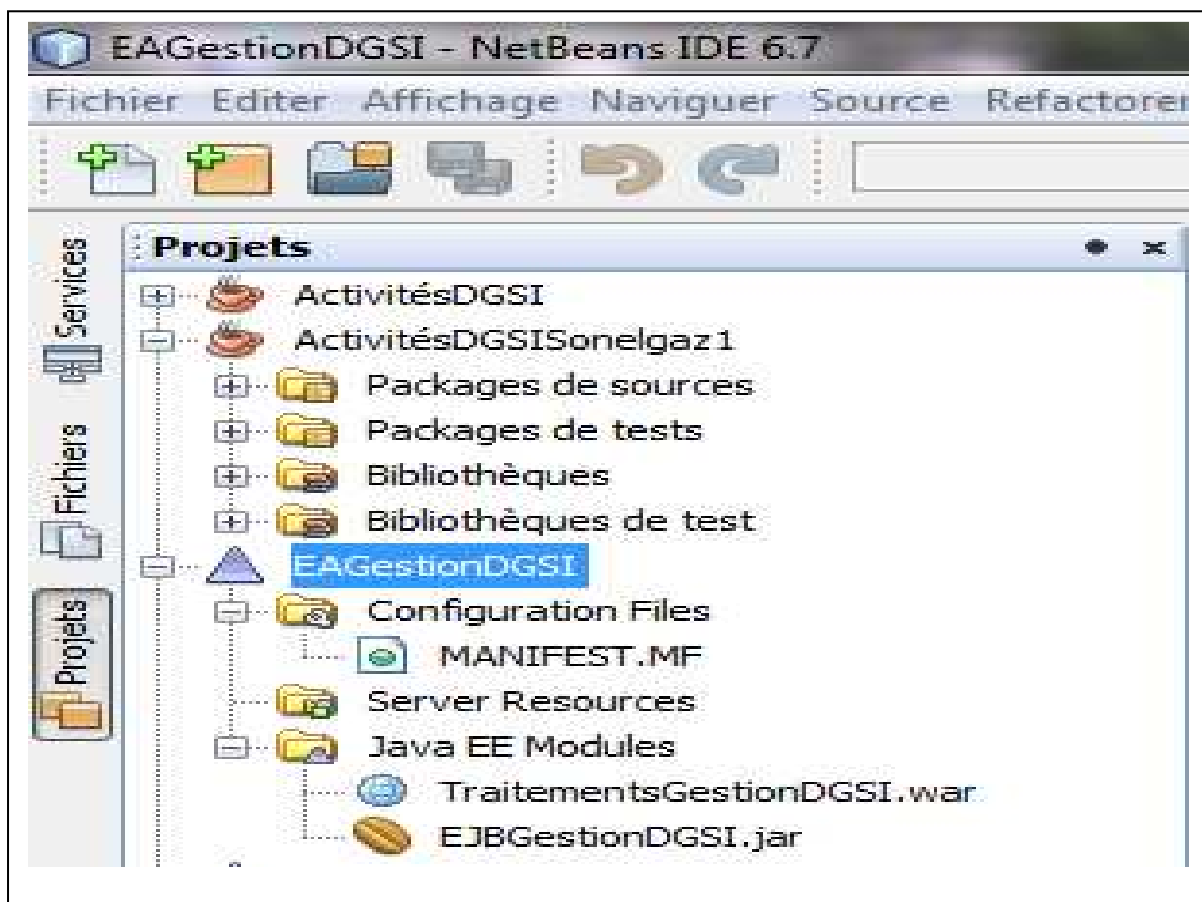


Figure II.3.2.1: Projet EAGestion Sonelgaz

### II. 3.2.2 : Le module EJBGestionSonelgaz :

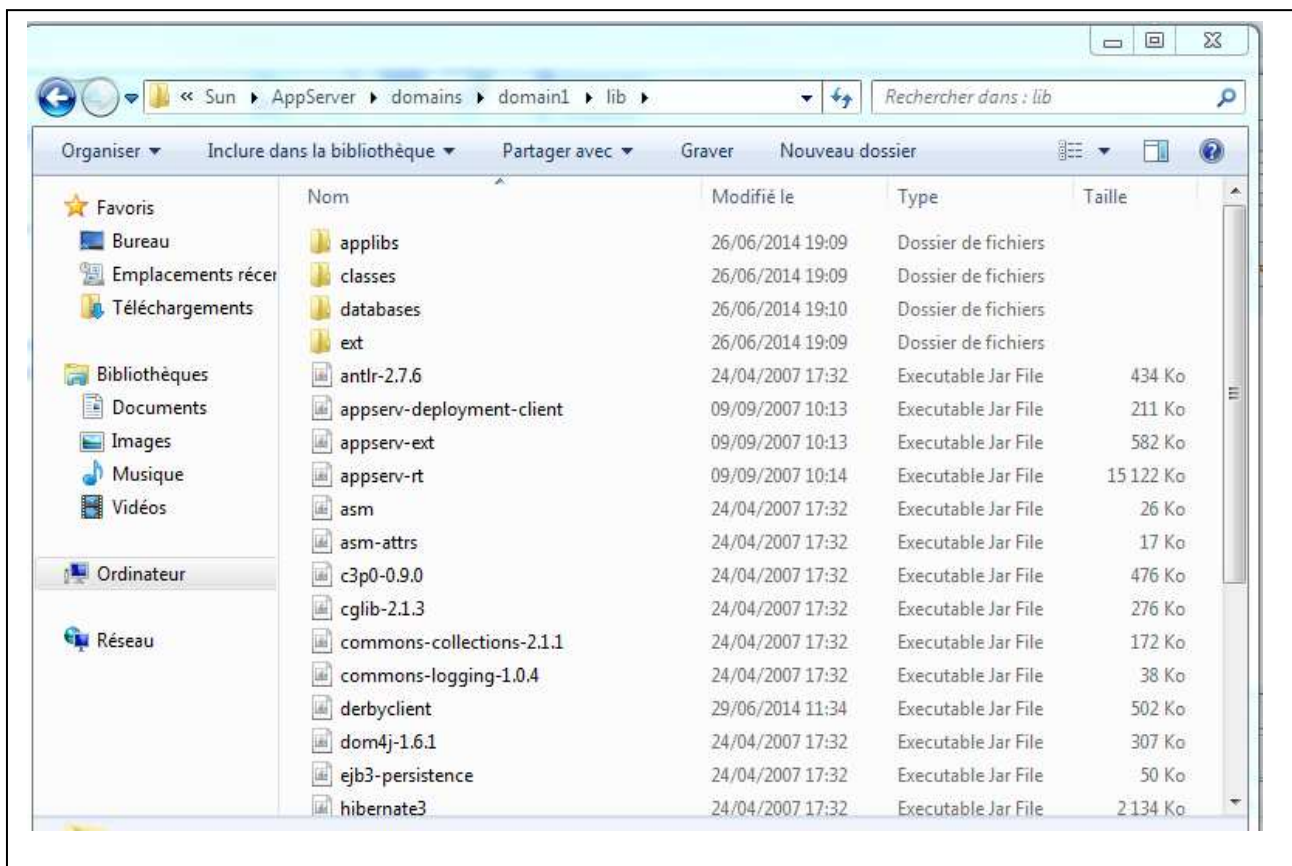
C'est un projet de type EJB Module de la catégorie Java EE des projets NetBeans.

Ce module construit un composant logiciel distribué (c'est-à-dire déployé sur le serveur distant Glassfish), il implémente les couches : [JDBC], [JPA/Hibernate], [DAO] de notre application.

### II. 3.2.2.1. Configuration Hibernate de serveur Glassfish :

Afin que le serveur Glassfish V2.1 livré avec Netbeans 6.7 nous permet d'implémenter la couche JPA/Hibernate, on a ajouté quelques bibliothèques, dans le dossier [`<AppServer>/domains/domain1/lib`] qui se trouve dans le répertoire d'installation de ce dernier.

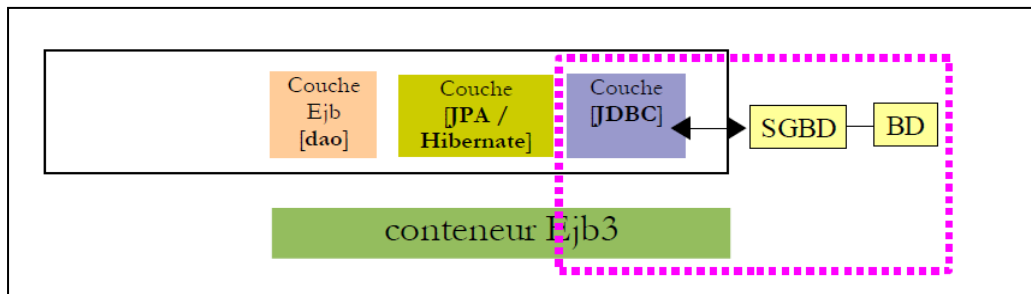
La figure suivante montre l'ensemble de ces bibliothèques ajoutées



**Figure II.3.2.2.1 : Les bibliothèques java configurant le serveur Glassfish**

Revenons maintenant à notre module EJB, afin que ce dernier puisse exploiter les données de notre base de données, en assurant leurs persistances, on a configuré ça sous forme des fichiers XML qui ont été créés dans ce module.

### II. 3.2.2.2. Ajout d'une ressource JDBC au serveur Glassfish :



Afin que le serveur Glassfish puisse accéder à notre base de données [dbsonelgaz], on a ajouté une ressource JDBC au serveur. Ce qui génère le fichier de configuration [sun-resources.xml] dans le répertoire « Server Resources » du module EJB, dont le contenu est le suivant :

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE resources PUBLIC "-//Sun Microsystems, Inc./DTD Application Server 9.0
   Resource Definitions //EN" "http://www.sun.com/software/appserver/dtds/sun-
   resources_1_3.dtd">
3. <resources>
4.   <jdbc-resource enabled="true" jndi-name="jdbc/dbsonelgaz" object-type="user" pool-
      name="dbsonelgazPool">
5.     <description/>
6.   </jdbc-resource>
7.   <jdbc-connection-pool allow-non-component-callers="false" associate-with-thread="false"
      connection-creation-retry-attempts="0" connection-creation-retry-interval-in-seconds="10"
      connection-leak-reclaim="false" connection-leak-timeout-in-seconds="0" connection-
      validation-method="auto-commit" datasource-
      classname="com.mysql.jdbc.jdbc2.optional.MysqlDataSource" fail-all-connections="false"
      idle-timeout-in-seconds="300" is-connection-validation-required="false" is-isolation-level-
      guaranteed="true" lazy-connection-association="false" lazy-connection-enlistment="false"
      match-connections="false" max-connection-usage-count="0" max-pool-size="32" max-wait-
      time-in-millis="60000" name="dbsonelgaz" non-transactional-connections="false" pool-
      resize-quantity="2" res-type="javax.sql.DataSource" statement-timeout-in-seconds="-1"
      steady-pool-size="8" validate-atmost-once-period-in-seconds="0" wrap-jdbc-
      objects="false">
8.     <property name="URL" value="jdbc:mysql://localhost:3306/dbsonelgaz"/>
9.     <property name="User" value="root"/>
10.    <property name="Password" value="()/"/>
11.  </jdbc-connection-pool>
12. </resources>
```

**Figure II.3.2.2.2 : Le code source du fichier de configuration [sun-resources.xml].**

- Lignes 8-10 : les caractéristiques Jdbc de la source de données (Url de la base, nom et mot de passe de l'utilisateur). La base de données MySQL *dbsonelgaz* est celle décrite précédemment.
- Ligne 7 : les caractéristiques du pool de connexions associé à cette source de données

Le fichier ci-dessus reprend toutes les informations saisies dans l'assistant sous un format XML. Il sera utilisé par l'IDE Netbeans 6.7 pour demander au serveur Glassfish v2.1 de créer la ressource "jdbc/dbsonelgaz" définie ligne 4.

### II. 3.2.2.3-Création d'une unité de persistance :

Après avoir implémenté et configuré la couche JDBC, on passe à la couche [JPA]. Cette couche a été configurée par l'unité de persistance [persistence.xml] : cette unité indique l'implémentation JPA utilisée (Hibernate dans notre cas), elle nous permet aussi d'indiquer à Hibernate le type de SGBD à gérer (dans notre cas MySQL 5). Ce fichier est créé dans le répertoire « Configuration Files » du module EJB, son contenu est le suivant :

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <persistence          version="1.0"          xmlns="http://java.sun.com/xml/ns/persistence"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
   http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">
3.   <persistence-unit name="dbsonelgaz" transaction-type="JTA">
4.     <provider>org.hibernate.ejb.HibernatePersistence</provider>
5.     <jta-data-source>jdbc/dbsonelgaz</jta-data-source>
6.     <properties>
7.       <!-- Dialecte -->
8.       <property                                name="hibernate.dialect"
   value="org.hibernate.dialect.MySQL5InnoDBDialect"/>
9.     </properties>
10.  </persistence-unit>
11. </persistence>
```

**Figure II.3.2.2.3 : Le code source du fichier de configuration [persistence.xml].**

- Ligne 3 : le type de transactions est JTA : les transactions seront gérées par le conteneur Ejb de Glassfish.
- Ligne 4 : une implémentation Jpa / Hibernate est utilisée. Pour cela, la bibliothèque Hibernate a été ajoutée au serveur Glassfish (voir précédemment).
- Ligne 5 : la source de données JTA utilisée par la couche Jpa a le nom JNDI « jdbc/dbsonelgaz ». La source de données "jdbc/dbsonelgaz" est configurée dans le fichier [sun-resources.xml] précédent.
- Ligne 8 : cette ligne n'est pas générée automatiquement. Elle doit être ajoutée à la main. Elle indique à Hibernate, que le SGBD utilisé est MySQL5.

### II. 3.2.2.4-Création des entités JPA :

Pour donner une image objet à notre base de données, on a créé un package [Sonelgaz.entites] dans le répertoire « Source Packages » du module EJB, ce package implémente la couche JPA de notre architecture, il contient des entités JPA.

Chaque entité de ce package encapsule les lignes d'une table de la base de données, donc il y'a autant d'entités dans ce package que de tables de la base de données.

Voici le code d'une de ces entités :

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package Sonelgaz.entites;
import java.io.Serializable;
import java.util.Date;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
 *
 * @author Galleze
 */
@Entity
```

```
@Table(name = "document")
@NamedQueries({ @NamedQuery(name = "Document.findAll", query = "SELECT d FROM
Document d"), @NamedQuery(name = "Document.findByIdEmp", query = "SELECT d
FROM Document d WHERE d.idEmp = :idEmp"), @NamedQuery(name =
"Document.findByIdByNomDoc", query = "SELECT d FROM Document d WHERE d.nomDoc =
:nomDoc"), @NamedQuery(name = "Document.findByIdBygenreDoc", query = "SELECT d
FROM Document d WHERE d.genreDoc = :genreDoc"), @NamedQuery(name =
"Document.findByIdByDateDoc", query = "SELECT d FROM Document d WHERE d.dateDoc =
:dateDoc"), @NamedQuery(name = "Document.findByIdByTypeDoc", query = "SELECT d
FROM Document d WHERE d.typeDoc = :typeDoc"), @NamedQuery(name =
"Document.findByIdByDestinataire", query = "SELECT d FROM Document d WHERE
d.destinataire = :destinataire"), @NamedQuery(name = "Document.findByIdByMotifDoc", query
= "SELECT d FROM Document d WHERE d.motifDoc = :motifDoc"))})
public class Document implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "code_doc")
    private String codeDoc;
    @Basic(optional = false)
    @Column(name = "id_emp")
    private String idEmp;
    @Basic(optional = false)
    @Column(name = "genre_doc")
    private String genreDoc;
    @Basic(optional = false)
    @Column(name = "nom_doc")
    private String nomDoc;
    @Column(name = "type_doc")
    private String typeDoc;
    @Column(name = "date_doc")
    @Temporal(TemporalType.DATE)
```

```
private Date dateDoc;
@Column(name = "destinataire")
private String destinataire;
@Basic(optional = false)
@Column(name = "motif_doc")
private String motifDoc;

public Document() {
}

public Document(String codeDoc) {
    this.codeDoc = codeDoc;
}

public Document( String codeDoc, Integer idEmp,String genreDoc, String nomDoc, String
typeDoc , Date dateDoc, String destinataire, String motifDoc ) {
    this.codeDoc = codeDoc;
    /*this.idEmp = idEmp;*/
    /* this.genreDoc = genreDoc;
    this.nomDoc = nomDoc;
    this.typeDoc = typeDoc;
    this.dateDoc = dateDoc;
    this.destinataire = destinataire;
    this.motifDoc = motifDoc;

}

public String getCodeDoc() {
    return codeDoc;
}
```



```
public void setCodeDoc(String codeDoc) {
    this.codeDoc = codeDoc;
}

public String getIdEmp() {
    return idEmp;
}

public void setIdEmp(String idEmp) {
    this.idEmp = idEmp;
}

public String getGenreDoc() {
    return genreDoc;
}

public void setGenreDoc(String genreDoc) {
    this.genreDoc = genreDoc;
}

public String getNomDoc() {
    return nomDoc;
}

public void setNomDoc(String nomDoc) {
    this.nomDoc = nomDoc;
}

public String getTypeDoc() {
    return typeDoc;
}

public void setTypeDoc(String typeDoc) {
```

```
this.typeDoc = typeDoc;
}

public Date getDateDoc() {
    return dateDoc;
}

public void setDateDoc(Date dateDoc) {
    this.dateDoc = dateDoc;
}

public String getDestinataire() {
    return destinataire;
}

public void setDestinataire(String destinataire) {
    this.destinataire = destinataire;
}

public String getMotifDoc() {
    return motifDoc;
}

public void setMotifDoc(String motifDoc) {
    this.motifDoc = motifDoc;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (codeDoc != null ? codeDoc.hashCode() : 0);
    return hash;
}
```

```
}

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Document)) {
        return false;
    }
    Document other = (Document) object;
    if ((this.codeDoc == null && other.codeDoc != null) || (this.codeDoc != null &&
!this.codeDoc.equals(other.codeDoc))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Sonelgaz.entites.Document[codeDoc=" + codeDoc + "]";
}

}
```

**Figure II.3.2.2.4 : Code de l'une des entités JPA**

### **II. 3.2.2.5-Création de la couche [dao] d'accès aux entités JPA :**

Afin de pouvoir exploiter nos objets (donc manipuler nos données), on a créé un package [dao]. La couche [dao] gérant les entités Jpa précédente est implémentée par les classes [IDao]. Cette classe implémente deux interfaces :

- L'une dite locale préfixée ici par I et suffixée par le terme *Local* [IDaoLocal]
- L'une dite distante préfixée ici par I et suffixée par le terme *Remote* [IDaoRemote]

L'interface locale [IDaoLocal] de l'Ejb se contente de dériver l'interface [IDao] est le suivant :

```
1. Package Gs.dao;
2.
3. import javax.ejb.Local;
4.
5. @Local
6. public interface IDaoLocal extends IDao {
7. }
```

L'interface distante [IDaoRemote] de l'Ejb se contente de dériver l'interface [IDao]

```
1. package Gs.dao;
2.
3. import javax.ejb.Remote;
4.
5. @Remote
6. public interface IDaoRemote extends IDao {
7. }
```

### **II. 3.2.2.6-Création d'une classe exception :**

On a crée un package [Sonelga.exception] qui contient une classe [ExceptionPersonne], dérivée de la classe [RuntimeException] de Java pour gérer les exceptions.

La figure suivante est une capture d'écran montrant cet EJB ainsi que ses différents composants.

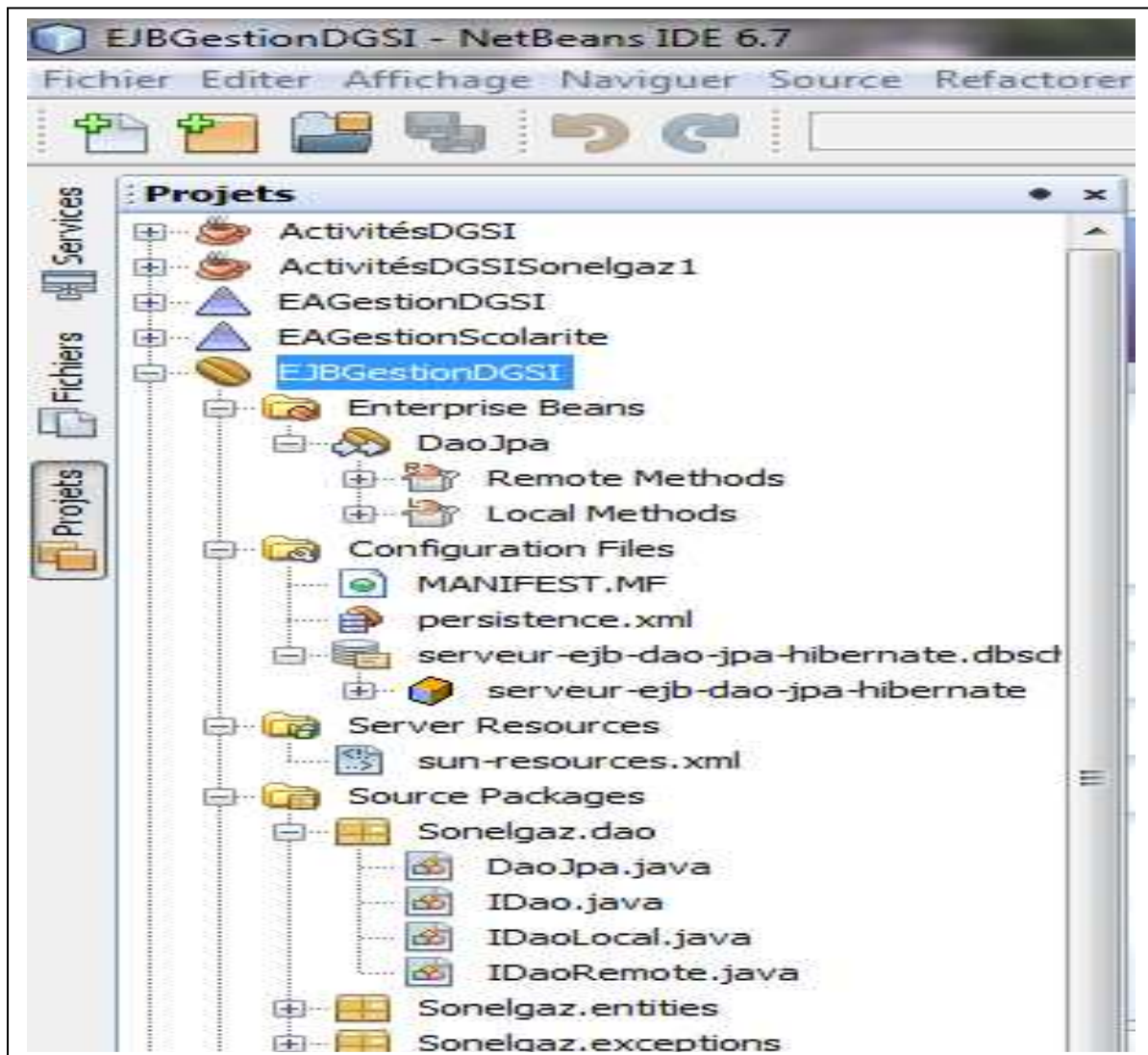


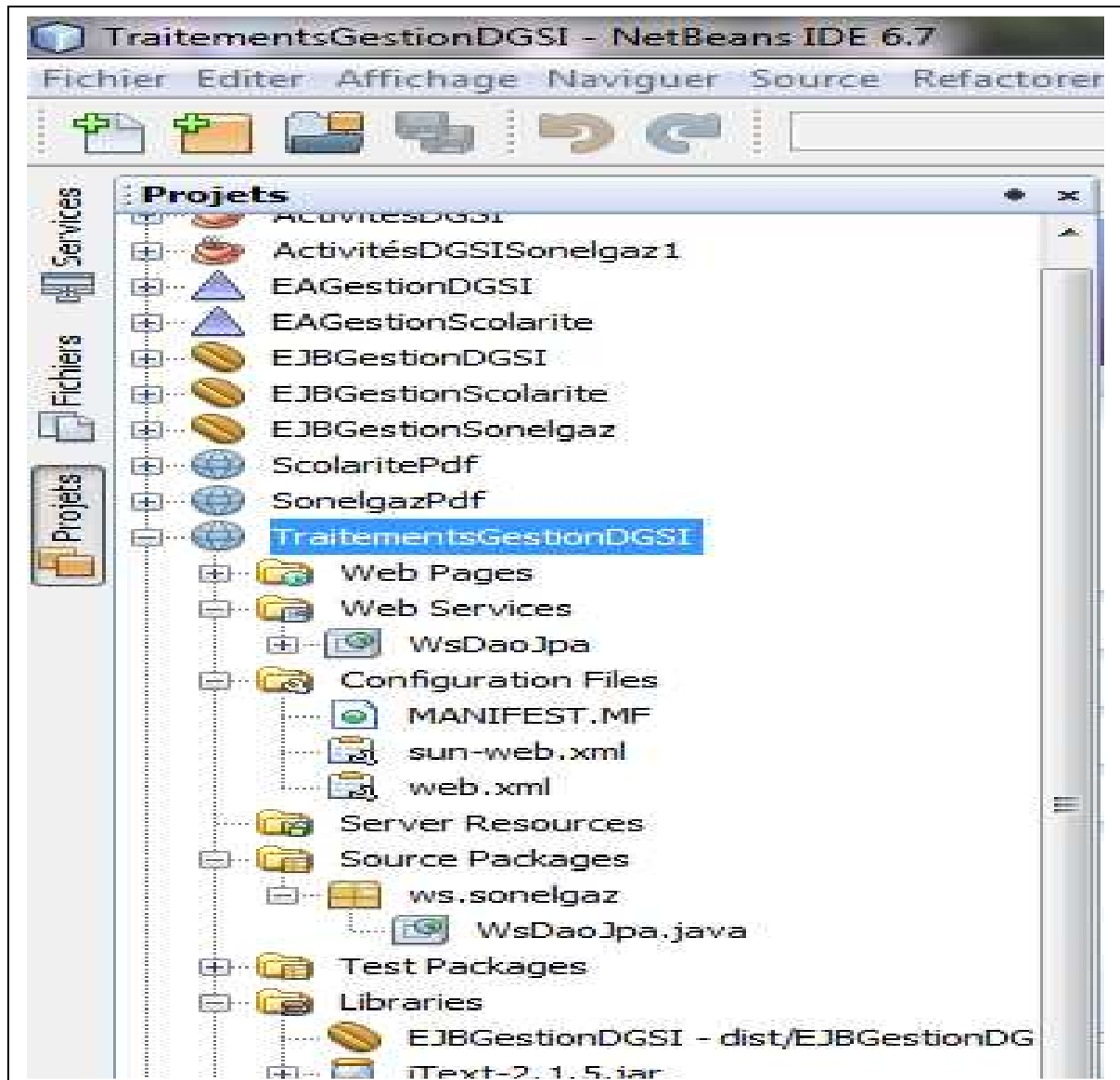
Figure II.3.2.2.6. Le module EJBGestionSonelgaz.

### II. 3.2.2.7.. Le module TraitementsGestionSonelgaz :

C'est un projet de type web application qui s'exécute sur le serveur Glassfish.

Ce projet web que nous construisons va utiliser l'EJB précédent, et il a aussi besoin de référencer son archive .jar (.jar de EJB). Donc on a ajouté le module EJB aux bibliothèques de ce projet web ainsi que la bibliothèque iText-2.1.5.jar pour le déploiement de nos Pdf.

Ce module web nous permettra d'implémenter la couche [metier] et de déployer nos Pdf



*Figure II.3.2.2.7. Le module TraitementsGestionSonelgaz*

L'implémentation de la couche [metier] sera faite dans un paquetage [Gs.metier] (voir la figure ci-dessus). Ce paquetage comprendra outre les interfaces [IMetierLocal] et [IMetierRemote] et leurs implémentation [IMetier].

Afin de pouvoir déployer nos Pdf, on a créé un package [PDF] dans le répertoire « Source Packages » du module WEB (voir la figure ci-dessus).

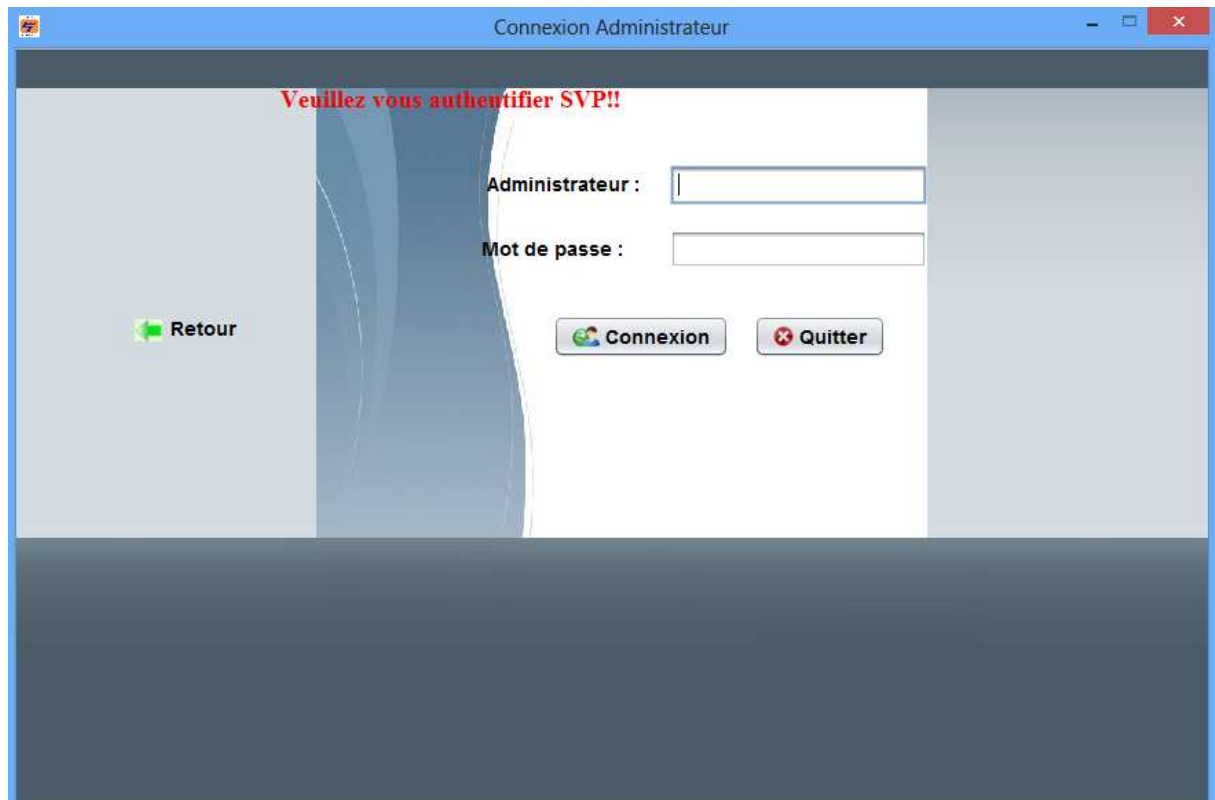
### **Section 3 : Les fonctionnalités de l'application :**

#### **Introduction :**

Pour illustrer l'échange de données entre le client et notre système, nous avons réalisé une application client. Cette application fait appel à des interfaces graphiques swing qu'on a développée dans un projet Java.

À travers les interfaces présentées ci-dessous, nous visons à donner une vue générale de notre application conçue.

## Chapitre IV : Réalisation



L'interface d'authentification (figure 4.22) est la première fenêtre télécharger et visualisé par l'utilisateur. Il devra taper son nom utilisateur, son mot de passe ainsi qu'il doit choisir son profil pour pouvoir accéder à la suite de l'application.







## Chapitre IV : Réalisation

Gestion des utilisateurs (Employés de la DGS)

Modifier/Supprimer Ajouter

Nom :

Prénom :

Date de naissance : Jour Mois Année

Fonction : Choisissez


Adresse :

Téléphone :

Actif : Choisissez

Enregistrer Editer la carte

Fermer



## Chapitre IV : Réalisation

Gestion des utilisateurs (Employés de la DGSI)

Ajouter Afficher/Supprimer

**Veuillez saisir l'identifiant de l'employé à supprimer et afficher la liste des utilisateurs**

Saisir l'identifiant Ici


**X Supprimer**

**Afficher**

Identifiant	Nom	Prénom	Dateden...	Fonction	Adresse	Login	Password	Téléphone	Actif	Code
5	Galleze	Ramdane	1973-08-...	Respons...	Tizi Gheniff	gal8	ram8	0778671...	Oui	
7	Belkacem	Fatima	1977-10-...	GestStoc...	Tizi gheniff	bel10	fat11	0789653...	Oui	
9	Galleze	hdjila	1989-04-...	GestDoc	Tizi ouzou	gal4	hdj14	0559451...	Oui	
10	Galleze	Quardia	1987-10-...	GestParc...	Tizi Ouzou	gal10	oua8	0779892...	Oui	

**Retour** **Fermer**

Espace Responsable



Nom :  Fonction :  Code utilisateur :

**Modifier le profil** **Deconnexion**

**Bienvenue sur votre espace Responsable**

- Répondre aux demande**
- Consultation de stockpapier**
- Consultation de matériel**
- Retour**

Sur quelle intervention voulez-vous s'assurer **Intervention interne** **Intervention externe**

Saisissez ici: Mois Année

Num	dat	Mot	Struc	Durée

**Fermer**

## Chapitre IV : Réalisation

Réponss des demandes

Désaccord Accord

**Code Réponse :**

**Identifiant Employé :**  ✓ validation Accord

**Destinataire :**

**Motif de réponse:**

**Genre réponse :**

**Nom Réponse :**

**Type Réponse :**

**Date Réponse :**

Confirmation  
enregistrement avec succès.  
OK

📄 Edition interne ✓ Accusé de réception 📄 Edition Externe

🏠 Retour ✖ Fermer

Espace Gestionnaire de stock papier

 **الشركة الجزائرية للكهرباء و الغاز**  
**Société algérienne de l'électricité et du gaz**

Nom :  Fonction :  Code utilisateur :

👤 Modifier le profil 🚪 Deconnexion

**Bienvenue sur votre espace Gestionnaire de stock papier**

- ✓ Gestion des Approvisionnements papier
- ✓ Gestion de transfert papier

## Chapitre IV : Réalisation

Gestion des interventions

Intervention interne Intervention externe

Date Intervention : Jour Mois Année

Motif Intervention :

Nom structure :

Durée :

Enregistrer

Demander rapport d'intervention

Gestion des approvisionnements papier

Stockage Approvisionnement

Etat03 Chercher d'abord

Nouvelle Quantité :

Quantité Entrée :

Quantité Existante :

Nouvelle Quantité :

Supprimer Valider le stockage

Codestockagepap	Quantitestock	Codepapier
7	15	Etat03

Enregistrer

Demander l'état de stock papier

Retour Fermer

### Conclusion :

Durant ce chapitre « Réalisation », on a présenté d'abord les outils de développement nécessaires à la réalisation de notre application, puis l'architecture J2EE suivie, et enfin les fonctionnalités de l'application.

# Conclusion générale

Le principal objectif de ce travail est le développement d'une architecture reconfigurable pour les applications coopératives et distribuées à base de service web et réseaux.

Ce travail propose un système de gestion afin de faciliter la communication entre les différents employés de la DGSI, d'éviter des pertes de temps et finalement pour assurer une gestion plus rapide et plus efficace. configuration

On a vu à travers ce travail comment implémenté une architecture 3-tiers avec une centralisation des données, ainsi les majeures contributions de ce projet peuvent se résumer comme suit :

1. Comprendre le fonctionnement de la technologie Swing utilisé.
2. Maîtriser au profond les concepts essentiels et nécessaires de langage de programmation « **java** ».
3. Connaitre le principe et l'utilité des patrons logiciels, et savoir les appliquer dans un contexte.
4. Avoir expérience pour la conduite d'un projet qui m'a permis de :
  - ✓ Apprendre à travailler en équipe ;
  - ✓ se familiariser avec plusieurs langages à savoir : JAVA, SQL, XML ;
  - ✓ enrichir mes connaissances sur l'architecture client/serveur 3-tiers
  - ✓ se familiariser avec les environnements de développement orienté web, à savoir : Netbeans, HTML, et MySQL pour les bases de données.

Enfin, souhaitons que ce travail soit utile pour la mise en œuvre d'une application distribuée pour la gestion des activités de la DGSI.

# Bibliographie

---

# Bibliographie

---

## Référence bibliographique :

### Les ouvrages :

1) **Pratique des systèmes d'information**

Par M.C BELAID

*ET*

D. TAOURI Née Bouyacoub.

Edition de juillet 2005.

2) **[OSMAN EYROLLES MULTIMEDIA] : Le développeur Java™ 2, Mise et œuvre et solution.**

Par : Antoine Mirecourt et pierre

3) **[Bruller98] :** Bertrand Bruller, "Les réseaux locaux informatiques"

Edition Presses Universitaires de France, 1998.

Côte d'ouvrage : RES 84.

Lieu : Bibliothèque du Département d'Informatique de T.O

4) **[Pujolle03] :** Guy Pujolle / Olivier Salvatori, "Les Réseaux"

Edition Eyrolles Paris, 2003.

Côte d'ouvrage : RES 80.

Lieu : Bibliothèque du Département d'Informatique de T.O

5) **[S.Ghernaouti94]** "Communication En Mode Client / Serveur, les outils du traitement réparti coopératif" Edition Masson, Paris, 1994.

Côte d'ouvrage : RES 100.

Lieu : Bibliothèque du Département d'Informatique de T.O

6) **[Tanenbaum95] :** Andrew S. Tanenbaum, "Réseaux Architectures Protocoles, Application" InterEditions Paris, 1995.

Côte d'ouvrage : RES 79.

Lieu : Bibliothèque du Département d'Informatique de T.O



# Bibliographie

---

7) [G. Gardarin]. Bases de Données. Objet & Relationnel. Eyrolles, 1999.

8) [OSMAN EYROLLES MULTIMEDIA] : Le développeur Java<sup>tm</sup> 2

Edition 1999

Par Antoine Mirecourt

1, rue Thénard 75005 Paris

Fax : 01 44 41 11 85

<http://www.eyrolles.com>

9) [Abraham Silberschatz et Peter B.Galvin]. Structure des réseaux.

10) [John L.Hennessy et David A.Pattersob]. Réseaux d'interconnexion

## Les sites web :

11) Les fondements du langage Java

<http://www.berti-editions.com>

12) **Programmation orienté objet** : Disponible sur

<http://www.developpeur.c.la>

13) **Persistance Java 5 par la pratique** : Disponible sur :

[serge.tahe@istia.univ-angers.fr](mailto:serge.tahe@istia.univ-angers.fr)

14) **Unified Modeling Language** : Disponibles sur

<http://laurent-audibert.developpez.com/Cours-UML/>

15) **Cours Client/serveur** : Disponibles sur

<http://www-licence.ufr-info-p6.jussieu.fr/lmd/licence/2005/ue/LI373-2006fev/>

16) **Cours « Bases de données »** : Disponibles sur : <http://ouranos.inapg.inra.fr/wiki/>

17) **Cours MYSQL** : Disponibles sur <http://cyberzoide.developpez.com> et

<http://www.mysql.com>.

## THÈSES

18) DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE ET DE L'UNIVERSITÉ DE SFAX

# Bibliographie

---

Conception, implantation et expérimentation d'une architecture en bus pour l'auto-réparation des applications distribuées à base de services web

**Présentée et soutenue par**

**Riadh BEN HALIMA**

Le Jeudi 14 Mai 2009