



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed El Bachir El Ibrahimi B.B.A.

Faculté des Mathématiques et d'Informatique

Mémoire

Présente en vue de l'obtention du Diplôme

Master en informatique

Spécialité : Technologies de l'information et de la communication

Thème

**Conception et réalisation d'une application basée sur
l'architecture SOA avec Java EE**

**Cas : La gestion d'intégration des bénéficiaires DAIP de
l'ANEM**

Présenté par :

- Fratsa Yassine

- Zitouni Amel

Devant le jury composé:

Président : Moussaoui Ali

Docteur

Université Mohamed El Bachir El Ibrahimi B.B.A.

Examineur : Charikhi Mourad

Docteur

Université Mohamed El Bachir El Ibrahimi B.B.A.

Encadreur: Belazoug Mouhoub

Docteur

Université Mohamed El Bachir El Ibrahimi B.B.A.

Remerciements

Avant tout, nous remercions Dieu qui nous a donné la foi, la santé, le Courage et la volonté pour terminer ce modeste travail.

Nous tenons aussi à exprimer notre reconnaissance et profonde Gratitude à notre promoteur Mr Belazoug Mouhoub , pour son encouragement et sa patience.

Un grand Merci aux membres du jury pour l'honneur qu'ils nous ont fait d'évaluer et de juger notre travail.

Nos sincères sentiments s'adressent à nos familles et amis (es) pour leurs soutiens et leurs encouragements.

Enfin, nos remerciements à tous ceux qui, de près ou de loin, nous ont aidés à l'aboutissement de cette quête

Dédicaces

Je dédie ce travail à :

Mes très chers parents, pour leur amour, soutien et sacrifices, qui m'ont épaulé tout au long de ma vie, qu'ils trouvent ici ma profonde reconnaissance.

*A l'amour éternel de ma vie, mes anges
Kenzy-Rymes et Chamssi.*

A ma famille, mes très chers frères et sœurs, mes amis et tous ceux qui ont contribué de près ou de loin à l'accomplissement de mon travail.

A mon binôme, Yassine

Z. Amel

Résumé

L'ANEM, l'Agence Nationale de l'Emploi, est un établissement national fondé en 1962. Elle est considérée aujourd'hui comme étant le premier intermédiaire entre les employeurs et demandeurs de l'emploi. Les besoins d'accompagnement de ces derniers ne cessent de croître et l'ANEM cherche à mieux les cibler.

Dans ce contexte, une application Desktop, GEST-DAIP avait été mise en place pour gérer le dispositif d'aide à l'insertion professionnelle DAIP.

Néanmoins, l'architecture applicative de « Gest-Daip » empêche l'extensibilité et l'évolution de ses services à cause des difficultés de la mettre à jour et d'ajouter des nouvelles fonctionnalités ce qui à empêcher le développement des nouveaux besoins métiers qui est l'intégration des bénéficiaires DAIP qui impose la collaboration avec les partenaires de l'ANEM pour le partage de données et de services métier.

Le présent projet consiste à concevoir et réaliser une plateforme « TARSSIM » qui sera basée sur une architecture SOA. Il a pour but d'améliorer la qualité, l'extensibilité et la maintenabilité des services existant pour la gestion et le contrôle de l'opération d'intégration.

Cette refonte se fera en premier lieu en unifiant les sources de données, ensuite un ensemble de services web sera développé et mis à la disposition des partenaires de l'ANEM sous forme d'API REST en utilisant la technologie Java-EE.

Mots clés: Architecture Orientée Services (SOA), Application Programming Interface (API), Services Web, Representational state transfer (REST), Java Enterprise Edition (JAVA EE).

Abstract

ANEM, the National Employment Agency, is founded in 1962. Today, it is considered as the first intermediary between employers and jobseekers in Algeria. The needs of the latter for support are constantly growing and ANEM seeks to better target them. In this context, a web platform, « Gest-Daip » has been developed to manage the professional integration assistance system DAIP.

Nevertheless, the application architecture of « Gest-Daip » prevents the extensibility and evolution of new services. Moreover, this system is difficult to add new functionality, hence the need for the integration of DAIP beneficiaries which requires collaboration with ANEM partners for the sharing of data and business services.

This project consists of designing and building a “TARSSIM” platform which will be based on an SOA architecture. It aims to improve the quality, scalability and maintainability of existing services for the management and control of the integration operation.

This redesign will be done first by unifying the data sources, then a set of web services will be developed and made available to ANEM partners in the form of a REST API using Java-EE technology.

Keywords: Service-Oriented Architecture (SOA), Application Programming Interface (API), Web Services, Representational state transfer (REST), Java EE.

ملخص

الوكالة الوطنية للتشغيل هي مؤسسة وطنية أنشأت سنة 1962، تلعب دور الوساطة بين طالبي العمل والمستخدمين، وتسعى إلى تحسين طرق استهداف ودعم احتياجاتهم المتزايدة في مجال التشغيل. في هذا السياق، قامت الوكالة باستحداث الأرضية الرقمية GEST-DAIP لتسيير جهاز المساعدة على الإدماج المهني.

ومع ذلك، فإن الوكالة الوطنية للتشغيل تواجه عقبة أمام تطوير وتنويع هذه الخدمات، نظرا لمحدودية وصعوبة تحديث هذا النظام ليتماشى مع متطلبات المرحلة الجديدة والمتمثلة في ادماج المستفيدين من جهاز المساعدة على الإدماج المهني والذي يتطلب تعاون الوكالة مع الشركاء والمستخدمين لتبادل مجموعة من البيانات والخدمات.

يهدف مشروعنا هذا إلى إزالة الصعوبات التي تعرقل هذه العملية وذلك من خلال إعداد وتصميم أرضية رقمية

« ترسيم Tarim » بالاستناد على هندسة النظم المعلوماتية الموجهة للخدمات SOA.

هذا الإعداد يتم في المرحلة الأولى عن طريق توحيد مصادر البيانات، ثم تطوير مجموعة من خدمات WEB وإتاحتها لشركاء الوكالة كواجهات برمجة تطبيقات API ذات طابع REST مع استخدام تكنولوجيا Java EE

الكلمات المفتاحية: هندسة النظم المعلوماتية الموجهة للخدمات (SOA)، واجهة برمجة التطبيقات (API)، خدمات الويب، REST, Java EE.

Table des matières

<i>Informatique</i>	Erreur ! Signet non défini.
<i>Remerciements</i>	iii
<i>Dédicaces</i>	iv
Résumé	v
Abstract	vi
ملخص	vii
Table des matières	viii
Liste des figures	xi
Liste des tableaux	xii
Liste des sigles et abréviations	xiii
Introduction générale	1
1. Contexte	1
2. Problématique.....	1
3. Objectifs	2
4. Organisation du rapport	2
Partie I : Synthèse bibliographique	4
Chapitre 1 : Architecture Orientée Services	5
1. Introduction	5
2. Notions de base.....	5
2.1. Architecture logicielle	5
2.2. Notion de service.....	6
2.3. Architecture Orientée Services (SOA)	6
3. Web services (WS)	10
3.1. Définition d'un Web service.....	10
3.2. Formalisme de données	10
3.3. Types de Web Services	11
4. Conclusion.....	14
Chapitre 2 : Architecture Java EE	15
1. Introduction	15
2. Un bref historique de Java EE [11].....	15

3.	Présentation de JAVA EE : [12].....	15
4.	L'architecture de JAVA EE: [15].....	16
5.	Les couches logicielles Java EE [11].....	17
6.	Spécifications de la plateforme JAVA EE :.....	18
6.1.	Les APIs.....	18
6.2.	Les serveurs d'applications. [15].....	24
7.	Les avantages d'utiliser JAVA EE :	25
8.	Conclusion.....	25
Partie II : Étude préliminaire		26
Chapitre 3 : Présentation de l'organisme d'accueil.....		27
1.	Introduction	27
2.	Présentation de l'ANEM	27
3.	Organisation de l'ANEM	27
4.	Les missions de l'ANEM	28
5.	Présentation de la DDSI	28
5.1.	Organisation de la DDSI.....	28
5.2.	Missions de la DDSI.....	30
6.	Conclusion.....	30
Chapitre 4 : Étude de l'existant		31
1.	Introduction	31
2.	Présentation de Gest_DAIP.....	31
3.	Acteurs du système GEST DAIP.....	32
4.	Fonctionnalités de Gest_DAIP:.....	32
4.1.	Les fonctionnalités relatives au conseillers de L'ALEM:	32
4.2.	Les fonctionnalités relatives à L'AWEM:	33
4.3.	Les fonctionnalités relatives au DEW	33
5.	Périmètre et objectifs du projet.....	33
6.	Conclusion.....	33
Partie III : Notre contribution		34
Chapitre 5 : Analyse et conception		35
1.	Introduction :	35
2.	Définition d'UML :.....	35

3.	Démarche de l'analyse:	35
3.1.	Phase d'analyse :	36
3.2.	Conception :	46
4.	Conclusion.....	51
	Chapitre 6 : Réalisation.....	52
1.	Introduction	52
2.	Les outils et langages de développement	52
3.	Description de l'application (présentation des différentes interfaces de l'application)	57
3.1.	La page Authentification :	57
3.2.	Interface Accueil (Employeur):	58
3.3.	Interface Accueil (Administrateur):.....	58
3.4.	Interface gestion des contrats DAIP:.....	59
3.5.	Interface gestion des PV Comité ad-hoc:	59
3.6.	L'ajout d'un PV:	60
3.7.	L'impression d'un PV Brouillon:	60
3.8.	L'impression d'un PV Final:.....	61
3.9.	Interface gestion des Besoin en Postes budgétaires:	61
3.10.	L'ajout d'un Besoin:	62
3.11.	Interface gestion des Utilisateurs:.....	62
4.	Déploiement	62
5.	Conclusion.....	63
	Conclusion générale.....	64
	Références bibliographiques	65
	Références webographiques	67
	Annexes	68
	Annexe A : Dispositif d'Aide à l'Insertion professionnelle DAIP	69
	Annexe B : Infrastructure informatique.....	75

Liste des figures

Figure 1 : Les composants d'une architecture orientée services [Zhang Y., Zhou J., 2009].....	7
Figure 2: Couches de l'architecture orientée services [Rosen M. et al., 2012].	9
Figure 3: XML vs JSON	11
Figure 4: Communication dans une architecture orientée services basée sur REST	12
Figure 5 L'architecteur de Java EE.....	17
Figure 6:Architecteur trois tiers	18
Figure 7:Cycle de vie d'une servlet.....	20
Figure 8: Organigramme de la sous-direction de développement et des systèmes d'information	28
Figure 9: Organigramme de l'ANEM.....	29
Figure 10:Page d'accueil de system Gest_DAIP	31
Figure 11:Démarche de modélisation.....	36
Figure 12:Diagramme de contexte	36
Figure 13:Diagramme de cas d'utilisation	45
Figure 14:Diagramme de séquence s'authentifier	46
Figure 15:Diagramme de séquence « consulter PV comité ad-hoc».....	47
Figure 16:Diagramme de séquence « ajouter PV comité ad-hoc»	47
Figure 17: Diagramme de séquence « Consulter Besoin en Postes budgétaires»	48
Figure 18:Diagramme d'activité « authentification »	49
Figure 19:Diagramme d'activité « Imprimer PV Comité ad-hoc »	49
Figure 20:Diagramme de classe	50
Figure 21:Interface de l'IDE Netbeans.....	53
Figure 22:Interface Primefaces	55
Figure 23:Interface de connexion.....	57
Figure 24:Interface Accueil (Employeur)	58
Figure 25:Interface Accueil (Administrateur)	58
Figure 26:Interface gestion des contrats DAIP	59
Figure 27:Interface gestion des PV Comité ad-hoc.....	59
Figure 28:L'ajout d'un PV.....	60
Figure 29:L'impression d'un PV Brouillon.....	60
Figure 30:L'impression d'un PV Final	61
Figure 31:Interface gestion des Besoin en Postes budgétaires	61
Figure 32:L'ajout d'un Besoin.....	62
Figure 33:Interface gestion des Utilisateurs	62
Figure 34: Diagramme de déploiement.....	63
Figure 35: Architecture du réseau de l'ANEM.....	76
Figure 36: Architecture de l'infrastructure informatique du site central de l'ANEM.....	76

Liste des tableaux

Tableau I: Comparatif des Web services SOAP et REST	13
Tableau II: Structuration de cas d'utilisation globale.....	38
Tableau III: Scénarios des tâches d'un Administrateur	40
Tableau IV: Scénarios des tâches d'un employeur.	42
Tableau V: Scénarios des tâches d'un Comité ad-hoc.....	43

Liste des sigles et abréviations

ADSL : Asymmetric Digital Subscriber Line.
PV: Procès Verbal
ALEM : Agence Locale de l'Emploi.
ANEM : Agence Nationale de l'Emploi.
AWEM : Agence Wilaya de l'Emploi.
API : Application Programming Interface.
BDD : Base De Données.
BPM : Business Process Management.
CASNOS : Caisse Nationale de Sécurité Sociale des Non-Salariés.
CNAS : Caisse Nationale des Assurances Sociales.
CRUD: Create Read Update Delete.
CSS: Cascading Style Sheets.
CTA : Contrat de Travail Aidé.
CU : Cas d'utilisation.
DAIP : Dispositif d'Aide à l'Insertion Professionnelle.
DDSI : Direction de Développement et des Systèmes d'Information.
DEW : Direction d'Emploi des Wilayas.
DG : Direction Générale.
DMZ : Demilitarized Zone.
DOM : Document Object Model.
DREM : Direction Régionale de l'Emploi.
EAI : Enterprise Application Integration.
ESB : Enterprise Services Bus.
HTML: Hypertext Markup Language.
HTTP(S) : HyperText Transfer Protocol (Secure).
IDE: Integrated Development Environment.
JS: JavaScript.
JSON : JavaScript Object Notation.
MOM : Message Oriented Middleware.
MVC: Model View Controller.
ORM : Object-Relational Mapping.
REST : Representational State Transfer.
RMS : Réseau Multi Services.
SGBD: Système de Gestion de Base de Données.
SI : Système d'Information.
SOA : Service Oriented Architecture.
SOAP : Simple Object Access Protocol.
SQL : Structured Query Language.
SSMS : SQL Server Management Studio.

T-SQL : Transact - Structured Query Language.
UDDI : Universal Description, Discovery and Integration.
UI : User Interface.
UML : Unified Modeling Language.
URI : Uniform Resource Interface.
VLAN : Virtual Local Area Network.
VPN : Virtual Private Network.
WADL : Web Application Description Language.
WS : Web Service.
WSDL : Web Services Description Language.
XML : Extensible Markup Language.

Introduction générale

1. Contexte

Actuellement, le monde connaît une avancée technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science étudiant les techniques du traitement automatique de l'information. Elle joue un rôle important dans le développement de l'entreprise et d'autres établissements [19].

Avant l'invention de l'ordinateur, nous enregistrions toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières.

Ainsi, jusqu'à présent, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information. Cette invention a permis d'informatiser les systèmes de gestion de données des entreprises, ce qui est la partie essentielle dans leur développement aujourd'hui.

L'ANEM, l'Agence Nationale de l'Emploi, qui nous a accueillies durant notre stage de fin d'études, est consciente de la nécessité de s'adapter à cette révolution. Dans ce contexte, elle avait mis en place un système d'information DESKTOP « Gest_Daip » permettant la gestion de dispositif DAIP (Dispositifs d'Aide à l'Insertion Professionnelle) tel que la gestion des contrats, le paiement, les demandeurs et les employeurs.

2. Problématique

De nos jours, et avec l'explosion du Web, La complexité de l'architecture applicative de la plateforme « Gest_DAIP » constitue un frein pour l'extensibilité et l'évolution de l'opération de l'intégration des bénéficiaires DAIP. À cet égard, l'ANEM veut adopter une « Architecture Orientée Services » (SOA) d'une manière adaptée à son contexte, à ses besoins et à ses contraintes.

De plus, un problème de centralisation et d'extensibilité au niveau local de la wilaya ou :

- Manque de collaboration des différents secteurs ministériels.
- On ne peut pas suivre l'évolution de l'intégration des bénéficiaires DAIP en temps réel
- Il ya une difficulté d'aide à la prise de décision sur l'aspect financier
- Pas d'évaluation et adaptation des procédures d'intégration pour les cas faisant objets d'exception

D'autre part, le besoin d'un module gestionnaire de l'opération d'intégration et de permanisation des bénéficiaires DAIP.

3. Objectifs

Pour pallier ces problèmes, une refonte du système Gest_DAIP est requise afin d'aboutir à un système évolutif, intégré et maintenable. L'objectif de ce projet de fin d'étude et de concevoir et mettre en place une solution complète qui prend en charge les besoins fonctionnels et techniques de l'ANEM. Les objectifs assignés au projet sont :

- ✓ Refonte de l'architecture du système actuel Gest_DAIP, en se basant sur une architecture SOA, qui vise à répondre de manière pertinente aux besoins liés à la flexibilité, l'évolutivité et l'intégration continue ;
- ✓ Centralisation des sources de données de système gest_daip
- ✓ Ajoutant de nouvelles fonctionnalités permettant la gestion de l'opération de l'intégration des bénéficiaires daip ;
- ✓ Rendre disponible l'information liée à l'intégration des bénéficiaires DAIP dans les différentes tutelles

4. Organisation du rapport

Nous aborderons dans le présent rapport les points que nous avons jugés importants pour l'élaboration de notre solution. Il sera organisé comme suit :

PARTIE 1 : Synthèse bibliographique :

C'est une synthèse bibliographique couvrant les notions essentielles à la compréhension de notre sujet. Elle est constituée des chapitres suivants :

- **Chapitre 1 Architecture Orientée Service**

Ce chapitre définit l'architecture orientée services (SOA de l'anglais Service-Oriented Architecture) et les notions théoriques qui lui sont associées.

- **Chapitre 2 : Architecture Java EE**

Nous abordons dans ce chapitre l'essentiel des connaissances permettant de développer des applications web en s'appuyant sur JavaEE.

PARTIE 2 : Etude préliminaire

Dans cette partie, nous exposons les informations relatives à l'étude de l'environnement organisationnel et informatique du projet. Elle comporte les chapitres suivants :

- **Chapitre 3 : Présentation de l'organisme d'accueil**

Etant accueillies par l'ANEM, nous présentons dans ce chapitre sa structure, son organisation et ses missions.

- **Chapitre 4 : Etude de l'existant**

Ce chapitre constitue le compte rendu de notre étude de l'existant concernant l'ancien système Gest_Daip et ses limites. Les attentes du nouveau système seront dévoilées dans ce chapitre.

PARTIE 3 : Notre Contribution

Cette partie du rapport est dédiée à l'aspect pratique de notre projet, dans laquelle nous abordons plus en détails les choix de conception et de réalisation. Elle est composée des chapitres suivants :

- **Chapitre 5 : Conception et analyse des besoins**

Dans ce chapitre, nous décrivons les besoins du client en termes de spécifications fonctionnelles et techniques du nouveau système « Tarsim ».

À travers ce chapitre, nous mettons l'accent sur l'aspect architectural et technique de la solution et nous expliquons de manière plus détaillée comment réaliser l'intégration.

- **Chapitre 6 : Réalisation**

Dans ce chapitre, nous citons les différents aspects de la réalisation, notamment l'environnement de travail, les résultats et l'architecture de déploiement.

Nous clôturons notre mémoire par une conclusion générale ainsi que des perspectives pour notre projet. Un compliment de notre travail a été ajouté sous forme d'annexes.

Partie I :

Synthèse bibliographique

Chapitre 1 : Architecture Orientée Services

1. Introduction

Dans une industrie du logiciel où la complexité des applications ne cesse de croître, un des objectifs du génie logiciel est de conserver un processus intelligible de construction et de maintenance de ces applications. Le concept d'architecture logicielle a émergé comme l'action de concevoir une solution de haut niveau pour pallier cette complexité.

Aujourd'hui, l'évolution des entreprises impacte fortement leurs applications. En effet, si certaines applications restent pérennes, d'autres doivent être rapidement améliorées ou remplacées, car elles constituent un frein à la bonne marche de l'ensemble. Il fallait donc imaginer une architecture modulaire qui saurait incorporer l'existant tout en permettant le développement de nouveaux services performants et mutualisables.

Depuis quelques années, l'Architecture Orientée Services (SOA) s'est rapidement imposée et a été largement adoptée par l'industrie du logiciel. Ce style architectural propose un mécanisme souple d'assemblage d'unités logicielles, indépendant de toute technologie particulière, et guidé principalement par les besoins du métier.

Nous consacrons ce chapitre à l'étude détaillée de l'architecture orientée services. Tout d'abord, nous commençons par définir les notions d'architecture logicielle, de l'architecture orientée services et de service. Ensuite, nous abordons les services Web, en particulier les deux technologies de leur mise en place : SOAP et REST.

2. Notions de base

2.1. Architecture logicielle

Selon [Kruchten, P. et al, 2006], « une architecture logicielle comprend la structure et l'organisation selon lesquelles les composants du système moderne et les sous-systèmes interagissent pour former des systèmes plus complexes permettant ainsi une meilleure conception et analyse de ces composants ».

En effet, l'étude de l'architecture logicielle est l'étude de la conception et de la construction de systèmes logiciels. L'architecture d'un système est l'ensemble des principales décisions de conception prises au cours de son développement et de toute évolution ultérieure. Elle permet de produire des logiciels évolutifs, performants et de grande qualité [Medvidovic N., Taylor R.N., 2010].

L'architecture logicielle se manifeste dans toutes les facettes principales d'un système logiciel, y compris ses éléments structurels - composants (unités de calcul dans un système), ses connecteurs (points d'interaction entre les composants logiciels) et ses configurations (agencements de composants logiciels et de connecteurs, et les règles qui guident leur composition). De ce fait, les architectures logicielles jugées particulièrement utiles pour les familles de systèmes sont souvent codifiées en modèles d'architecture, styles d'architecture et architectures de référence paramétrables et réutilisables [Medvidovic N., Taylor R.N., 2010]. Ainsi, un style d'architecture logicielle est défini comme un ensemble de « modèles organisationnels et idiomes récurrents. » [Garlan, D., 1995]. Il existe plusieurs styles

d'architecture standards, dont les plus connus sont : pipe / filtre, client / serveur, 3 tiers / N-tiers, MVC (Model-View-Controller), SOA et Cloud Computing [Mostefai M.A., 2012].

2.2. Notion de service

Le service au sens SOA est le composant élémentaire de l'architecture logique du système. Tout accès à une donnée et tout accomplissement d'un processus métier doivent passer par un service. Un service comporte deux aspects principaux : l'interface et l'implémentation [Rosen M. et al., 2012].

- **L'interface** du service spécifie les opérations offertes par le service, leurs paramètres d'entrée et/ou de sortie et les protocoles d'utilisation de ces fonctionnalités. Un service propose généralement plusieurs opérations différentes mais sémantiquement liées ;
- **L'implémentation** du service peut être basée sur des applications existantes, sur l'orchestration d'autres services pour combiner leurs capacités ou sur un nouveau code écrit spécifiquement pour le service.

2.3. Architecture Orientée Services (SOA)

2.3.1. Définition de la SOA

Selon [Josuttis N.M., 2007] « La SOA n'est pas une architecture concrète : c'est quelque chose qui mène à une architecture concrète. Vous pourriez appeler cela un style, un paradigme, un concept, une perspective, une philosophie ou une représentation. En d'autres termes, la SOA est une approche, une façon de penser, un système de valeurs qui conduit à certaines décisions concrètes lors de la conception d'une architecture logicielle ».

Alors que [Juric, M. B., 2007] définit la SOA comme un style d'architecture logicielle qui consiste à concevoir des applications et des systèmes distribués pour fournir des services métier dotés d'interfaces auto-descriptives bien définies et qui sont utilisés pour composer les processus métier de l'entreprise.

La SOA permet de concevoir des applications faiblement couplées, modulaires et flexibles. Sa caractéristique la plus distinguée est qu'elle permet de traverser les barrières technologiques entre les différentes plates-formes logicielles [Rosen M. et al., 2012]. Une architecture SOA est décrite par trois acteurs qui sont :

- **Le fournisseur de services** : C'est un système qui implémente un service et publie sa description dans l'annuaire de services. C'est une entité adressable sur le réseau qui reçoit, accepte et exécute les requêtes des consommateurs ;
- **Le consommateur de services** : C'est un système qui a intérêt à consommer un service exposé par le fournisseur de services. Il cherche le service en interrogeant l'annuaire de services, il récupère sa description et l'utilise. Il peut être une application, un service ou un autre type de module logiciel ;
- **L'annuaire de services** : C'est un répertoire réseau contenant les descriptions des services disponibles. Il enregistre, classe et permet la découverte des services publiés par les fournisseurs.

La **figure 1** illustre les différentes étapes de communication, entre les acteurs dans une architecture orientée services. Ces étapes sont détaillées dans ce qui suit :

- (1) Un fournisseur de services fournit ses services selon un mécanisme standardisé en les publiant dans un annuaire de services sur le réseau ;
- (2) Un consommateur de services consulte l'annuaire pour rechercher le service dont il a besoin ;
- (3) L'annuaire localise le service demandé et oriente le consommateur vers le fournisseur en lui envoyant les informations de ce dernier ;
- (4) Le consommateur envoie sa requête au fournisseur afin de consommer le service ;
- (5) Le fournisseur répond à la requête du consommateur et retourne les résultats attendus.

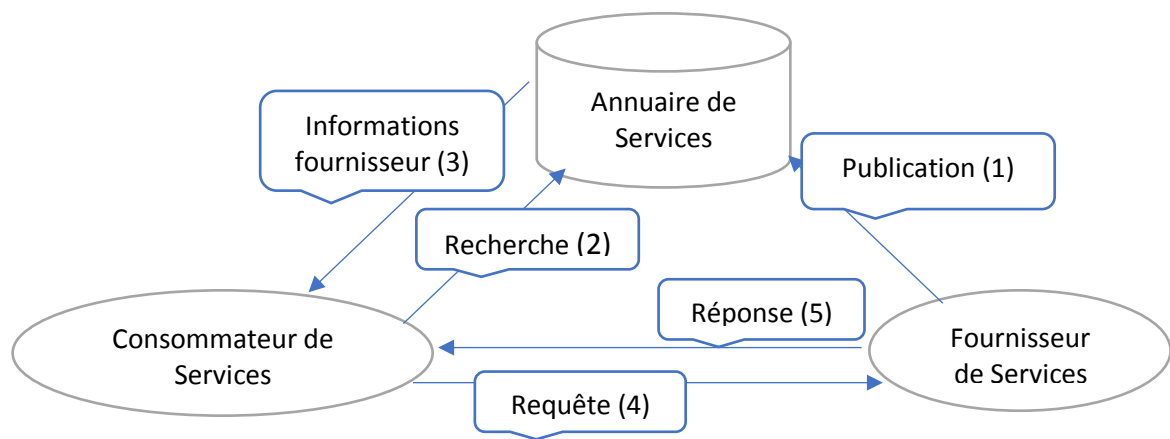


Figure 1 : Les composants d'une architecture orientée services [Zhang Y., Zhou J., 2009]

La SOA permet de limiter le couplage entre les consommateurs et les fournisseurs de services à des interfaces bien définies. Le couplage décrit le nombre de dépendances entre deux composants ou modules logiciels. Les composants faiblement couplés ont peu de dépendances bien connues et gérées. Le degré de couplage affecte directement la flexibilité et l'extensibilité d'un système. En effet, si le système est étroitement couplé, un changement dans un de ses composants nécessitera des changements dans d'autres composants. Dans SOA, les modifications apportées à une fonction n'affectent que les consommateurs connectés au service fournissant cette fonction. Tous les autres consommateurs qui interagissent avec le même fournisseur, mais via des interfaces de services différentes, ne sont pas affectés [Rosen M. et al., 2012].

2.3.2. Typologie des services

La démarche de la conception SOA consiste à faire une typologie de services qui se déclinent en trois catégories : les services métier, les services entités et les services utilitaires [Lemrabet Y., 2012].

- i. **Service métier** : c'est un service fonctionnel offrant un ensemble cohérent de traitements métier. Il correspond à un périmètre fonctionnel exposé à des consommateurs indépendamment des choix d'architecture applicative. Il est défini du

point de vue de l'utilisateur final qui le consomme [Bonnet P., 2005]. On entend par l'architecture applicative : le cycle de développement et les processus utilisés pour construire des applications d'entreprise ou encore les modèles d'application qui montrent les techniques de conception correspondant le mieux aux besoins opérationnels [TenStep, 2019] ;

- ii. **Service entité** : représente un service centré métier qui définit sa frontière fonctionnelle et son contexte autour d'une ou plusieurs entités métier liées, indépendamment de la plupart des services métier parents [Erl T., 2010]. En effet, un service entité est issu d'un travail d'architecture applicative qui prépare l'implémentation des services métier. Les services de ce type permettent de dégager un noyau stable composé de services hautement réutilisables [Lemrabet Y., 2012]. Un service entité, implémente souvent l'interface CRUD (Create, Read, Update et Delete) d'une entité, et ajoute des opérations supplémentaires spécifiques au domaine métier et nécessaires pour prendre en charge les fonctionnalités et les cas d'utilisation de l'application ;
- iii. **Service utilitaire** : contrairement aux deux autres types de services, les services utilitaires ne dépendent pas forcément du métier de l'organisation. Ils peuvent être utilisés pour envelopper les ressources existantes de l'entreprise issues de systèmes patrimoniaux, ou pour assurer des fonctionnalités utiles pour d'autres services comme la journalisation, la sécurité, la gestion des exceptions ou la transformation de données [Papazoglou M.P., 2003]. Ils sont implémentés une seule fois et ils font ensuite partie de l'infrastructure SOA.

2.3.3. Architecture SOA en couche

Selon [Rosen M. et al., 2012], l'architecture SOA se compose des couches suivantes (voir **figure 2**) :

- i. **Couche des ressources d'entreprise et des systèmes opérationnels** : Cette couche comprend les applications existantes, les systèmes patrimoniaux et les données de l'entreprise. Ces applications fournissent des traitements métier. Les données de cette couche résident dans des entrepôts et bases de données existants.
- ii. **Couche de l'infrastructure de services** : Composée des services d'intégration et des services entités. Les services d'intégration permettent l'intégration et l'accès aux applications existantes. Alors que les services entités expriment les fondamentaux du métier de l'entreprise et sont hautement réutilisables. De plus, les services entités servent à composer des services métier plus complexes.
- iii. **Couche des services métier** : Les services métier fournissent des fonctionnalités métier de haut niveau dans l'ensemble de l'entreprise. Cette couche fournit une intégration de la couche inférieure, rompant ainsi la dépendance directe entre les processus métier et les systèmes existants.
- iv. **Couche des processus métier** : Un processus métier consiste en une série d'opérations exécutées dans un ordre donné en fonction d'un ensemble de règles métier. Souvent, le processus métier est décrit dans un modèle de processus métier tel que ceux conformes

à la notation de modélisation de processus métier (BPMN¹) et il est exécuté par un système de gestion de processus métier spécialisé (BPMS). Les processus métier sont composés des services métier de la couche inférieure.

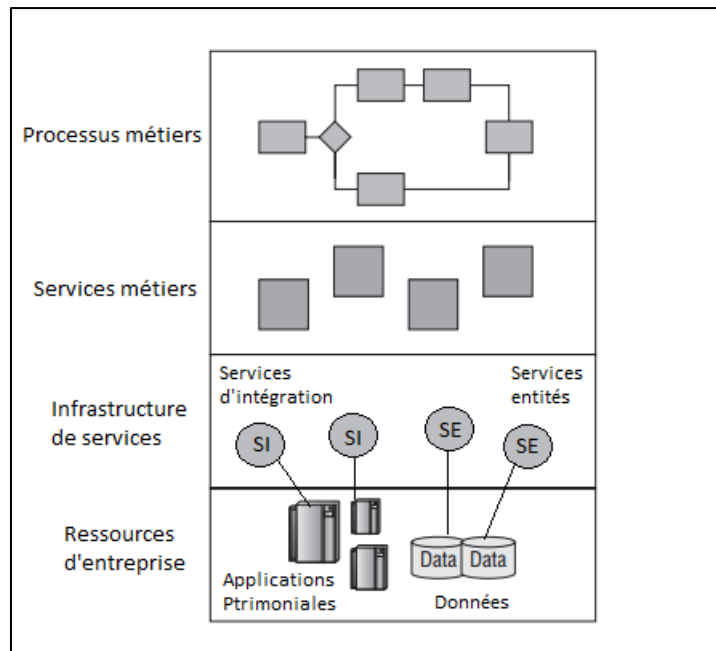


Figure 2: Couches de l'architecture orientée services [Rosen M. et al., 2012].

2.3.4. Approches d'identification de services

La difficulté dans l'identification des services réside dans la délimitation du périmètre fonctionnel de chaque service donc le choix du point de départ est primordial. Trois approches principales de conception des services de l'entreprise sont décrites dans la littérature, à savoir : l'approche top-down, l'approche bottom-up et l'approche middle-out. [Kohlborn T. et al., 2009] définit ces approches comme suit.

2.3.4.1. Approches Top-down

C'est une approche descendante axée sur l'entreprise d'un point de vue stratégique. Elle consiste principalement à décomposer les processus métier en tâches plus fines, afin de permettre l'identification des services de granularités différentes.

Premièrement, l'entreprise doit analyser ses objectifs et ses facteurs afin de comprendre ses besoins fonctionnels généraux. Ensuite, elle doit déterminer comment chacune de ces fonctionnalités pourrait être exprimée non pas comme une application unique, mais comme un ensemble de services. Cela permet de définir une feuille de route identifiant les principales entités, services et groupes de services nécessaires pour soutenir les objectifs de l'entreprise à long terme. Un autre point de départ pour l'approche Top-down consiste à développer des modèles de processus métier BPM en tant qu'ensemble d'exigences métier détaillées [Rosen M. et al., 2012].

¹ **BPMN (Business Process Model and Notation)** : « c'est une notation graphique universelle permettant de documenter les processus métier et de piloter les initiatives d'amélioration de processus » [BPMN, 2019].

2.3.4.2. Approches Bottom-up

C'est une approche ascendante qui démarre avec une perspective des différentes fonctions offertes par le système existant et dont les fonctionnalités seront transformées en services. Elle porte généralement sur un projet spécifique avec des exigences immédiates. On distingue deux manières de réalisation de cette approche. (1) La première ne modifie pas la structure du code des systèmes existants, mais propose de créer des enveloppes autour de leurs composants afin de pouvoir les utiliser dans un environnement orienté services ; (2) La seconde vise à créer des services logiciels autonomes qui encapsulent les fonctionnalités fournies par les systèmes existants en restructurant leur code source.

2.3.4.3. Approches Middle-out

L'approche middle-out est une stratégie intermédiaire qui prend en compte les nouvelles exigences de l'entreprise ainsi que le besoin de continuer à utiliser les systèmes existants. Elle consiste à combiner les deux stratégies Bottom-up et Top-down.

3. Web services (WS)

3.1. Définition d'un Web service

Un Web service est un type spécifique de services présentant les caractéristiques suivantes :

- Il expose ses fonctionnalités sur Internet à l'aide de langages et de protocoles d'Internet standards ;
- Il peut être implémenté via une interface auto-descriptive basée sur des standards d'Internet ouverts (par exemple, des interfaces XML publiées dans des référentiels en ligne).

3.2. Formalisme de données

Deux principaux formats sont utilisés pour le transfert des données produites et traitées par les Web services : XML et JSON.

- **XML (EXtensible Markup Language)** est un format de texte simple et très flexible dérivé de SGML (ISO 8879). Conçu à l'origine pour relever les défis de l'édition électronique à grande échelle, XML joue également un rôle de plus en plus important dans l'échange d'une grande variété de données sur le Web et dans beaucoup de systèmes [W3C, 2016] ;
- **JSON (JavaScript Object Notation)** est un format léger d'échange de données. Il est facile à lire et à écrire pour les humains. Il est aisément analysable et facilement généré par les machines [JSON, 2018].

La **figure 3** illustre la représentation d'un tableau de 3 objets « employee » dans XML et JSON.

XML	JSON
<pre><employees> <employee> <firstName> John</firstName> <lastName> Doe </lastName> </employee> <employee> <firstName> Anna </firstName> <lastName> Smith </lastName> </employee> <employee> <firstName> Peter </firstName> <lastName> Jones </lastName> </employee> </employees></pre>	<pre>{ "employees" : [{ "firstName" : "John", "lastName" : "Doe" }, { "firstName" : "Anna", "lastName" : "Smith" }, { "firstName" : "Peter", "lastName" : "Jones" }]}</pre>

Figure 3: XML vs JSON

3.3. Types de Web Services

Aujourd'hui, deux types de Web services sont les plus populaires et largement utilisés : les Web services basés sur SOAP (**S**imple **O**bject **A**ccess **P**rotocol) et les Web services basés sur REST (**R**epresentational **S**tate **T**ransfer). Ils diffèrent dans la manière d'exposition des services et de traitement de leurs données : SOAP est un protocole d'échange de messages basé sur XML tandis que REST est un principe de conception d'applications Web qui adhère étroitement à l'architecture client-serveur et préconise l'utilisation de méthodes HTTP très simples [Belqasmi F., 2012]. Dans ce qui suit, nous expliquons en détail ces deux types de Web services.

3.3.1. Simple Object Access Protocol (SOAP)

Dans une architecture SOA basée sur des Web services SOAP, la communication entre les trois composants SOA (fournisseur, consommateur et annuaire de services) repose sur le langage XML et utilise le protocole SOAP pour l'échange de messages. Les messages SOAP sont généralement échangés via HTTP. Néanmoins, d'autres protocoles et liaisons peuvent être utilisés. L'UDDI (Universal Description Discovery and Integration) est le registre ou l'annuaire de services défini par des organismes standard pour les Web services basés sur SOAP. Cependant, il est rarement utilisé dans la pratique [Belqasmi F., 2012]

Un message SOAP doit avoir une enveloppe comme élément racine. L'enveloppe SOAP peut contenir deux sous-éléments : l'en-tête et le corps. Le corps est un élément obligatoire utilisé pour coder les informations transmises. L'en-tête est un élément facultatif utilisé pour fournir des informations contextuelles relatives au traitement du message. Il est utilisé pour spécifier des informations liées à la sécurité ou le contexte transactionnel, par exemple. [Mumbaikar S., 2013].

Les descriptions de service sont publiées à l'aide du langage WSDL (Web Services Description Language). WSDL fournit des informations sur l'utilisation d'un Web service, y compris une description des méthodes du service et des informations de liaison (binding) [Belqasmi F., 2012].

3.3.2. Representational State Transfer (REST)

REST est un style architectural de plus en plus populaire en raison des performances et de la facilité d'utilisation qu'il présente par rapport aux Web services SOAP. Les Web services développés selon les principes REST sont appelés services RESTful.

REST utilise une architecture client-serveur dans laquelle le client envoie une requête (demande) au serveur, le serveur traite la demande puis renvoie la réponse adéquate [Mumbaikar S., 2013]. REST ne limite pas la communication client-serveur à un protocole particulier. Cependant, HTTP est le plus souvent utilisé car c'est le protocole principal de transfert sur le Web [Kumari S., Rath S. K., 2015].

Les services RESTful peuvent être décrits à l'aide du langage WADL (Web Application Description Language). Un fichier WADL décrit les requêtes pouvant être adressées à un service, en spécifiant l'identificateur de ressource uniforme (URI) du service et les données d'entrée et de sortie du service [Belqasmi F., 2012].

Un URI est une courte chaîne de caractères identifiant une ressource physique ou abstraite sur un réseau, et dont la syntaxe respecte une norme d'Internet. De même, une URL (Uniform Resource Locator) est une URI qui, en plus, fournit les moyens d'agir sur une ressource ou d'obtenir une représentation de la ressource en décrivant son mode d'accès primaire ou « emplacement » réseau [IETF, 2005].

Chaque requête reçue par un service RESTful entraîne le transfert d'une représentation de la ressource demandée, généralement sous la forme d'un document XML ou d'un objet JSON. Le consommateur de service n'a pas besoin de connaître le processus d'implémentation du service mais doit connaître les formats de données échangées pour pouvoir communiquer avec le service. La **figure 4** illustre la communication entre les consommateurs et les fournisseurs de services dans une SOA basée sur REST.

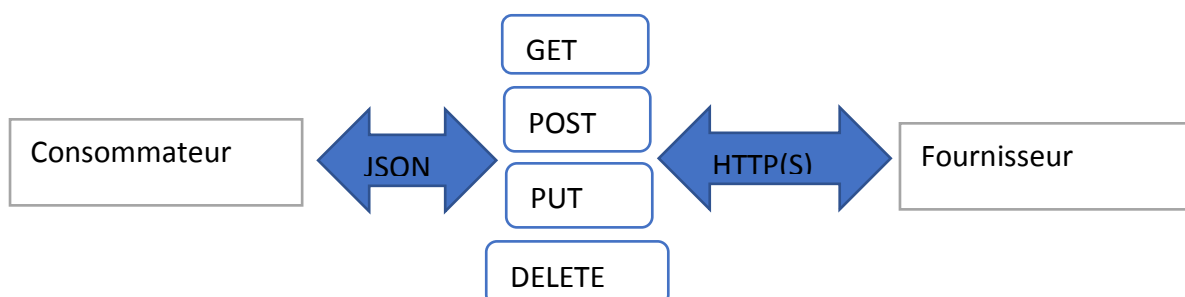


Figure 4: Communication dans une architecture orientée services basée sur REST

REST repose principalement sur trois principes de conception : l'adressabilité, l'interface uniforme et le principe sans état. [Mumbaikar S., 2013] définit ces principes comme suit :

- i. **L’adressabilité** : REST modélise l’ensemble de données manipulées comme ressources et identifie chaque ressource avec un URI. Une ressource est toute forme d’information pouvant être nommée et suffisamment importante pour être référencée (par exemple, un document, une ligne dans une base de données, un résultat de recherche, etc.) ;
- ii. **L’interface uniforme** : Les ressources REST basées sur HTTP sont accessibles via l’interface standard et uniforme du protocole HTTP. REST se base sur les quatre méthodes HTTP suivantes : POST, GET, PUT et DELETE qui représentent les opérations CRUD classiques ;
- iii. **Le principe « sans état »** : Chaque requête REST contient toutes les informations dont le serveur a besoin pour exécuter l’action demandée et les données de la session du client ne sont pas conservées sur le serveur. De plus, chaque requête est indépendante et non liée à la requête précédente. Par conséquent, les réponses du serveur sont également indépendantes.

3.3.3. Comparaison SOAP vs REST

Le tableau I présente un comparatif des avantages et des inconvénients des Web services de type SOAP et de type REST [Wagh K., 2012] [Potti P., 2012].

WS SOAP	WS RESTful
Une technologie bien connue et très utilisée depuis longtemps.	Une technologie plus récente.
Retourne toujours des données de format XML.	Peut renvoyer des données dans plusieurs formats (JSON, XML, etc.).
Consomme plus de bande passante à cause du volume relativement important des messages échangés.	Consomme moins de bande passante car les réponses des services sont légères.
Plus difficile à développer et nécessite des outils complexes.	Relativement facile à mettre en œuvre et à maintenir.
Fonctionne sur n’importe quel protocole de communication, même asynchrone.	Utilise principalement le protocole HTTP.

Tableau I: Comparatif des Web services SOAP et REST

Les Web services sont largement utilisés sur Internet si bien que leurs performances sont devenues un facteur important à considérer lors de la conception et l’implémentation de ces services. Or, les Web services basés sur SOAP génèrent un trafic réseau considérable et une latence élevée, et la taille des messages échangés est également importante. Ceci défavorise leur utilisation, particulièrement avec le nombre croissant de clients mobiles exploitant des services sur Internet qui nécessitent une communication avec des messages plus légers, d’où l’intérêt des WS RESTful qui sont beaucoup plus légers avec une latence moins élevée.

4. Conclusion

À travers ce chapitre, nous avons étudié l'architecture orientée services. Dans un premier temps, nous avons évoqué les notions d'architecture logicielle en général et de l'architecture orientée service et de services en particulier. Ensuite, nous avons décrit les couches de l'architecture SOA et les approches d'identification des services. Enfin, nous avons abordé en détails les Web services basés sur REST et sur SOAP.

Fédérant plusieurs technologies avancées, l'architecture SOA promet une évolution majeure dans la conception des systèmes d'information de l'entreprise. En effet, elle permet de mettre en place des systèmes évolutifs, flexibles, modulaires et faiblement couplés. De plus, elle assure une forte réutilisation des fonctionnalités existantes permettant ainsi la composition rapide de nouveaux systèmes fonctionnels.

En somme, la SOA répond de manière adéquate à toute forme de réutilisation des données de l'entreprise, des applications existantes ou des systèmes patrimoniaux. Cette caractéristique a fait de la SOA une architecture d'intégration d'applications d'entreprise très prometteuse. Dans le chapitre suivant, nous allons étudier la problématique l'intégration d'applications en général, et de l'intégration orientée services en particulier, dans un contexte inter et intra-entreprise.

Chapitre 2 : Architecture Java EE

1. Introduction

La technologie Java est aujourd'hui incontournable, tant pour le développement d'applications autonomes que pour l'utilisation des outils orientés web. La technologie Java est ici employée afin de générer des pages HTML dynamiques (DHTML). Elles permettent l'intégration de l'ensemble des fonctionnalités de Java Enterprise Edition (XML, bases de données, services web, systèmes distribués CORBA, ...etc.), contrairement à d'autres langages web (tel que PHP par exemple).

Dans ce chapitre nous allons présenter l'essentiel des connaissances permettant de développer des applications web en s'appuyant sur Java EE.

2. Un bref historique de Java EE [11]

L'environnement Java pour l'entreprise a commencé à émerger assez rapidement après les débuts de Java au milieu des années 90. A son origine, Java était destiné aux environnements contraints (par exemple des petits équipements électroniques). Il a en fait percé dans l'environnement du Web, notamment dans les navigateurs pour les supports d'interfaces graphiques riches la notion d'appliquette Java (en anglais applet). Les premières déclinaisons de Java dans l'environnement des serveurs sont apparues en 1997 avec les servlets, dont l'objectif est la construction programmatique de pages Web, puis avec les Enterprise Java Bean (EJB) dont l'objectif est le support de code métier (API) nécessitant un contexte d'exécution transactionnel.

Après ces premiers pas et un relatif succès de ces technologies, Sun a structuré l'offre technique autour des serveurs d'applications Java à travers le standard JAVA EE. L'objectif de ce dernier est de fédérer dans un cadre cohérent toutes les technologies nécessaires à la mise en œuvre des applications de l'entreprise (applications orientées <serveur>). Sun a décidé de rebaptiser son langage « Java 2 » et d'introduire la première version 1.2 des spécifications de JAVA EE en 1999. Cette version est suivie alors de la version 1.3 de JAVA EE en 2001, puis la version 1.4 en 2003 incluant un support complet des standards XML et le support des services Web.

Mais pour des raisons de marketing, Sun a décidé de rebaptiser cette plateforme « Java Enterprise Edition » (en supprimant le 2) à compter de la version 1.5, qui elle-même été changée en « 5 » et elle est devenu JEE5.

3. Présentation de JAVA EE : [12]

Qu'est ce que JAVA EE ?

JAVA EE (Java 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 la «plate-forme JAVA EE» pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution. JAVA EE comprend notamment :

- Les spécifications du serveur d'application, c'est-à-dire de l'environnement d'exécution : JAVA EE 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 JAVA EE SDK (JAVA EE Software Development Kit).

Dans la mesure où JAVA EE 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 JAVA EE repose sur des composants distincts, interchangeable et distribués, ce qui signifie notamment :

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

4. L'architecture de JAVA EE: [15]

JAVA EE ajout des nombreuses couches de niveau entreprise au-dessus de la plate-forme J2SE – Java Standard Edition. Chaque couche est conçue pour supporter une différente technologie de développement.

- **Technologie web application:** technologies liées à la production des interfaces web dynamiques, par exemple JSP (Java Server Pages) et servlet.
- **Technologie entreprise application:** technologies plus directement liées à la logique de business : EJB (Enterprise Java Bean), JavaMail, JMS (Java Message Service), JTA (Java Transaction), etc.
- **Technologie web services:** technologies utiles au développement des applications adhérentes au paradigme SOA (Service Oriented Architecture) : web services, JAXWS (java API for XML-based web services), JAX-RPC (Java API for XML-Based RPC).
- **Technologie management and security:** technologies liées à la gestion de la technologie d'entreprise afin de réaliser l'accès et l'échange d'information entre machines et services distribués : JAAS (Java Authentication and Authorization Service), JCA (Java Connector Architecture)

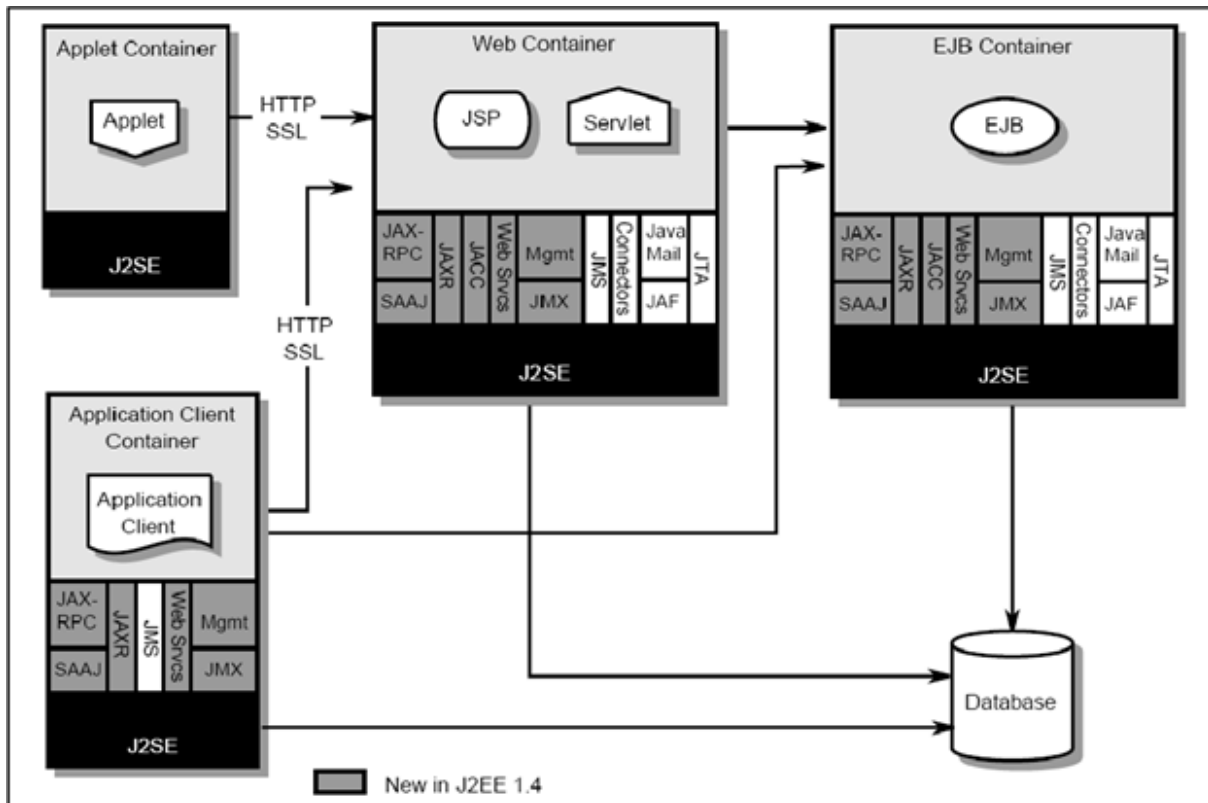


Figure 5 L'architecture de Java EE

5. Les couches logicielles Java EE [11]

La plateforme JEE utilise un modèle d'application distribuée multi tiers pour des applications d'entreprise. La logique d'application est divisée en composants selon la fonction. Les composants divers qui composent l'application JEE sont installés sur des machines différentes en fonction du niveau de l'environnement JEE multi tiers auquel le composant d'application appartient, la figure suivante montre deux types d'applications JEE réparties dans des niveaux décrits comme suis :

- La couche client : ces composants s'exécutent sur la machine du client.
- La couche WEB : ces composants s'exécutent sur le serveur JEE.
- La couche métier : ces composants s'exécutent sur serveur JEE.
- La couche EIS (Entreprise information System) : s'exécutent sur le serveur EIS.

Bien qu'une application JEE peut être constituée par trois ou quatre couches comme montrées dans Figure 2 suivante. Les applications JEE multi tiers sont généralement considérées comme des applications 3-tiers, parce qu'elles sont distribuées sur trois emplacements :

- La machine client.
- La machine pour serveur Java EE.
- Et la machine pour serveur de base de données.

Les applications à 3-tiers qui fonctionnent de cette façon étendent le standard 2-tiers : client/serveur en plaçant un serveur d'application entre l'application de client et le stockage principal.

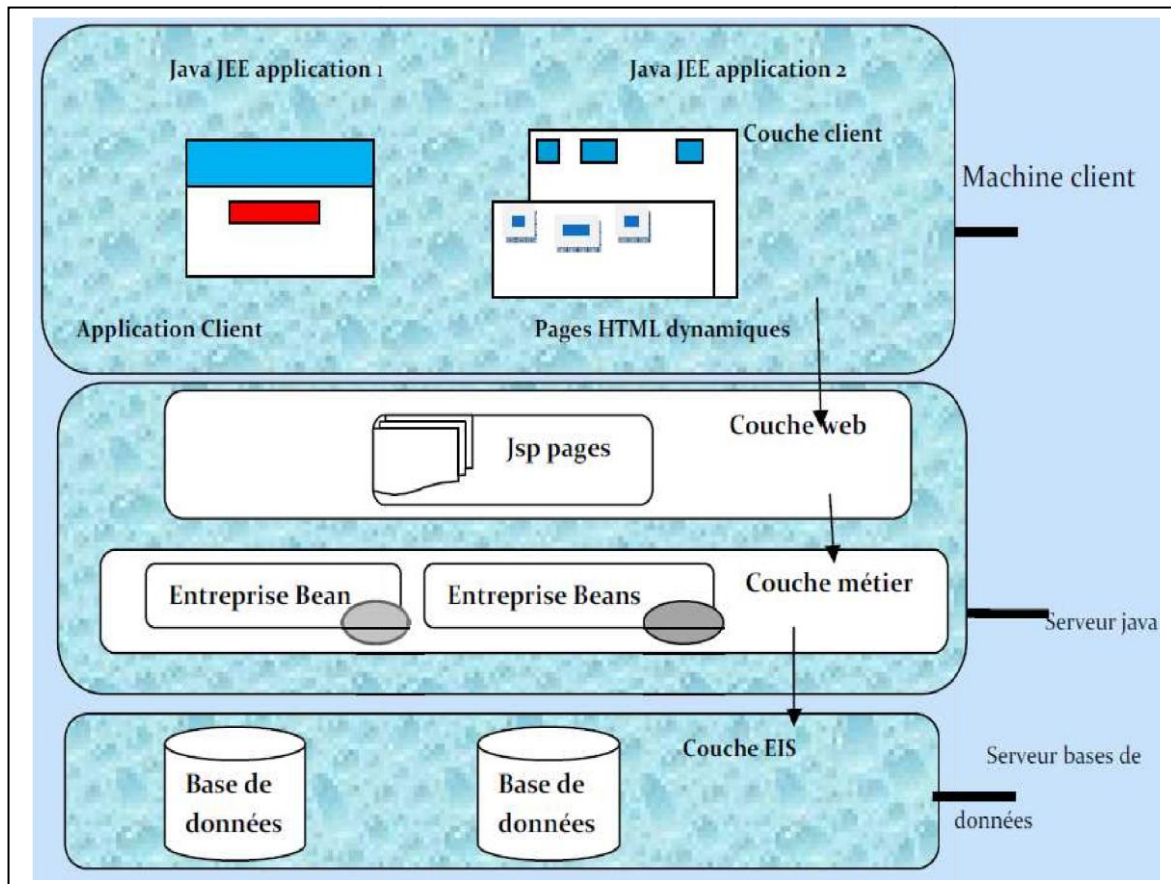


Figure 6:Architecteur trois tiers

6. Spécifications de la plateforme JAVA EE :

JAVA EE comporte des spécifications qu'on peut regrouper en deux grandes parties :

- ✚ Les APIs ;
- ✚ Les serveurs d'applications.

6.1. Les APIs.

Les APIs sont des extensions Java, forment une collection d'outils logiciels permettant d'offrir en standard un certain nombre de fonctionnalités, Sun fournit une implémentation minimale de ces APIs nommée JAVA EE SDK (JAVA EE SOFTWARE DEVELOPMENT KIT). On peut regrouper ces APIs en trois groupes principales :

- Composants ;
- Services d'infrastructures ;
- Services de communications.

6.1.1. Les composants :

Un composant est un fichier simple, contenant généralement du code compilé. Il est possible de créer des composants avec la majorité des langages. Aussi il peut être vu comme unité qui regroupe un certain nombre de fonctionnalités qui peuvent être appelées depuis un programme externe ou client.

Remarque: Pour les composants JAVA EE écrite en java et compilées sont assemblés en une application JAVA EE déployés, exécutés et gérés par le serveur JAVA EE.

On peut faire une séparation entre les composants existants selon leur exécution :

- Composants exécutés sur le client.
- Composants exécutés sur le serveur.

1. **Composants exécutés sur le client:** Un client JAVA EE peut être client web, application client ou applet.

- **Client web**

Un client web est appelé client léger qui n'a généralement pas de requête de BDD. L'utilisation de client web permet de charger les opérations lourds au niveau de l'entreprise beans et sont exécutées sur le serveur JAVA EE où ils peuvent exploiter les avantages des technologies JAVA EE (sécurité, la vitesse, les services ...) sur le serveur. Un client web a besoin de deux fonctionnalités principales :

- Les pages web dynamiques avec les différents langages à balisage (HTML, XML), ces pages sont générés par les composants web au niveau du client web.
- Le navigateur web qui affiche les pages reçus par le serveur.

- **Application client**

L'application client fonctionne au niveau de la couche métier, et peut accéder directement à l'entreprise Bean. Généralement les utilisateurs possède une interface graphique (GUI) pour exécuter l'application client (on utilisant l'API Swing ou Abstract windwos toolkit).

- **Applet**

Une applet est une petite application java qui a les caractéristiques de pouvoir s'exécuter via un navigateur. En effet, la plupart des navigateurs aujourd'hui, dispose d'une machine virtuelle capable d'interpréter les bytecode inclus dans une page HTML. Il existe plusieurs différences entre application et applets .en effet, les applets fournissent un certain nombre de fonctionnalités supplémentaires propres à leur contexte d'utilisation, comme la possibilité de communiquer avec le navigateur ou même entre applets. Mais la plus notable, réside dans la sécurité qui leur est associée.

2. **Composants exécutés sur le serveur [Isabelle V, Louis M,2001]**

Sont des composants web ou bien EJB.

❖ **Composants WEB** : Sont des servlets, JSP ou JSF.

➤ **Servlet** Ou « un peu d'HTML dans beaucoup de Java... »:

Une servlet est un composant du côté serveur, considérée comme une classe java qui permet une communication avec un protocole « requête / réponse ». Autrement dit, une servlet est un programme qui s'exécute côté serveur en tant qu'une extension du serveur. Elle reçoit une requête du client, elle effectue des traitements et renvoie le résultat.

L'utilisation des servlets permet la génération des pages web dynamiques qui utilisent le protocole HTTP, ou aussi n'importe quel autre protocole reposant sur le principe de servlet. Une servlet est une classe java qui implémente les méthodes doPost et/ou doGet .

Cycle de vie d'une servlet :

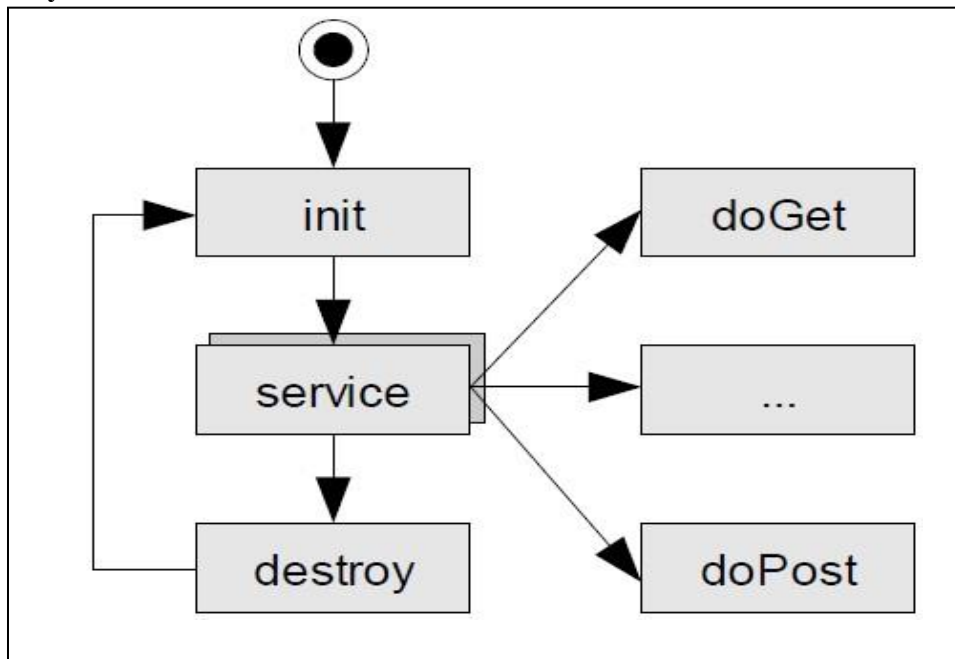


Figure 7: Cycle de vie d'une servlet

- **init()** : initialisation de la servlet chargement du code. Souvent effectué lors de la première requête cliente (doGet, doPost) Allocation d'un pool de threads.

- **doGet ()** : Traitement des requêtes HTTP GET
- **doPut ()** : Traitement des requêtes HTTP PUT
- **doPost ()** : Traitement des requêtes HTTP POST
- **destroy ()** : Destruction de la servlet par le serveur

✚ La méthode Init est appelé une seule fois au chargement de la servlet.

✚ La méthode service est appelée à chaque requête http (client)

✚ La méthode destroy est appelée une seule fois lorsque le conteneur détruit la servlet.

- **Avantage :**

- ✓ Portabilité (car s'exécute sur une JVM (java vertuel machin).
- ✓ S'exécute cote serveur.
- ✓ Maintenabilité.
- ✓ Elle maintient des sessions de plusieurs clients http grâce à l'objet http session.
- ✓ Multithread/contrairement a PHP exécute plusieurs fois.

- **Inconvénients :**

- ✓ Très difficile d'utiliser les outils d'édition HTML pour traiter les aspects graphique.

➤ **JSP : Java Server Page** Ou « un peu de Java dans beaucoup d'HTML... »

Les JSPs (pour Java Server Pages) est une technologie qui permet d'écrire très facilement des pages dynamiques en insérant des portions de code java dans une page html. Destiné à l'origine à la présentation des données, une page JSP permet de définir le design d'une page (en HTML) et d'y intégrer les données (grâce aux portions de code java).cette technologie favorise donc la séparation de la présentation et du contenu.

D'un point de vu plus technique, une page java est donc constitue de code HTML dans lequel sont insérées des code java délimitées par des balises particulières .l'intégralité du code est portable puisque tout comme un code java, il sera compilé et exécuté comme Servlet .

A la différence d'une Servlet, il sera par contre compilé au vol (lors de chaque invocation par un moteur de JSP. Enfin, les pages JSP offrent les possibilités d'importer des classes java et d'accéder à des objets java.

➤ **JSF [14]**

(Java Server Faces) Un cadre standard de composants conçu pour construire des interfaces utilisateur riches pour les applications Java. Java Server Faces exécuté sur le serveur, mais sont affichés sur le client. La technologie JSF représente un Framework basé sur les composants utilisateurs utilisés pour construire des applications web. L'API de JSF permet de représenter les composants, de gérer leurs états et leurs comportements. Elle facilite l'écriture d'interfaces utilisateurs en fournissant une bibliothèque de contrôles

- Simples : zones de saisie classiques, boutons, liens hypertextes.
- Complexes : tableaux de données ...
- Elle permet même de créer ses propres composants, ou d'enrichir les composants existants.

JSF permet principalement de:

- ✚ Faciliter le développement de pages web grâce à un Framework basé sur les composants. représenter les composantes UI et gérer leurs états.
- ✚ Gestion d'événements.
- ✚ Validation coté serveur.
- ✚ Conversion de données.
- ✚ Définition de la navigation des pages.
- ✚ gérer les composantes et les lier aux objets coté serveur.

❖ **Composants métiers (EJB) [Isabelle V, Louis M,2001]**

Enterprise JavaBeans (EJB) est une architecture de composants logiciels côté serveur pour la plateforme de développement JAVA EE.

Cette architecture propose un cadre pour créer des composants distribués (c'est-à-dire déployés sur des serveurs distants) écrit en langage de programmation Java hébergés au sein d'un serveur applicatif permettant de représenter des données (EJB dit *entité*), de proposer des services avec ou sans conservation d'état entre les appels (EJB dit *session*), ou encore d'accomplir des tâches de manière asynchrone (EJB dit *message*). Tous les EJB peuvent évoluer dans un contexte transactionnel.

L'architecture EJB prévoit en effet de manipuler les composants EJB dans un environnement distribué, assurant une complète compatibilité avec tous types de clients : servlet, page jsp, autre EJB, applet ...

Pour assurer la communication réseau entre différents composants distribués les EJB utilisent un mécanisme basé sur RMI. Ce mécanisme est implémenté dans une API qui offre à une machine virtuelle Java les mécanismes nécessaires pour invoquer des méthodes sur d'autres JVM, qu'elles soient ou non sur le même réseau.

Il existe trois types des EJB :

- **Les beans entités**

Ça correspond à des objets réels tels que ces objets sont généralement des enregistrements persistants stockés dans une BDD. On peut aussi dire que les beans entités sont des moyens de modélisation des concepts métier pouvant être désignés par des noms, par exemple : un bean entité peut représenter un client ou un objet stocké.

- **Les beans session**

Ceux sont une extension de l'application client et ils sont responsables de la gestion des processus ou des tâches.

- **Les beans orientés message**

Un bean session fournit une interface distante afin de définir les méthodes que l'on peut invoquer, alors que bean orienté message attend des messages asynchrones spécifiques auquel il répond.

Alors généralement, un EJB récupère également les données stockées, les traite et les envoie au programme du client.

6.1.2. Services d'infrastructures [12]

On trouve les services suivants :

- **JDBC « Java DataBase Connectivity »**

C'est une API qui permet aux programmes Java d'interagir avec les bases de données SQL. Les serveurs fournissent en plus un mécanisme qui permet de réutiliser les connexions créées avec les utilisations.

- **JTA/JTS « Java Transaction Api / Java Transaction Services »**

C'est une API définissant des interfaces standard avec un gestionnaire de transaction, c'est à dire permettant à des applications et à des serveurs Java EE d'accéder à des transactions.

- **JCA « Java EE Connector Architecture »**

C'est une API de connexion au système d'information de l'entreprise, notamment aux systèmes dits "legacy" tel que les ERP (Entreprise Resource Planning).

- **JMX « Java Management Extension »**

Permet de développer des applications WEB de supervision d'applications à partir des extensions.

- **JNDI « Java Naming and Directory Interface»**

C'est une extension Java standard qui fournit une API permettant de localiser et d'utiliser des ressources; il peut y avoir un appel à des services CORBA, DNS, NIS, LDAP...

6.1.3. Services de communication :

➤ **JMS « Java Message Service »** : A pour rôle l'envoi et la réception des messages entre deux composants d'application JAVA EE.

➤ **JAAS « Java Authentication And Autorisation Service »**

Permet aux applications JAVA EE d'authentifier et autoriser l'exécution par les utilisateurs.

➤ **Java Mail TM Technology**

Permet l'envoi des e-mails d'une manière indépendante de la plate forme et du protocole, donc envoi de courrier électronique.

➤ **RMI-IIOP « Remote Method Invocation/Internet Inter-Orb Protocol »** Cette API permet la communication synchrone entre objets.

➤ **Web Services**

Les web services permettent de partager un ensemble de méthodes qui pourront être appelées à distance. Cette technologie utilise XML ce qui permet d'être utilisé par n'importe quel langage et n'importe quelle plate forme.

➤ **XML**

N'est pas vraiment une API de JAVA EE mais son utilisation dans cette plate forme est de plus en plus important comme son utilisation pour les fichiers de configurations. XML est la base d'un nouveau mode de communication entre les applications.

Les conteneurs (en anglais containers) :

Un conteneur est une entité qui fournit aux composants la gestion de cycle de vie, la sécurité, les services de déploiement et d'exécution. Chaque type de conteneur (EJB, WEB, JSP...) fournit aussi des services spécifiques à ces composants.

Il existe plusieurs types de conteneurs qu'on peut citer :

- **Conteneur web**

Il est composé de deux grandes parties : un moteur de servlets et un moteur de JSP, donc ce conteneur permet d'exécuter les composants JSP et les servlets de l'application JAVA EE.

- **Conteneur EJB**

Ce type de conteneur permet de gérer l'exécution d'EJB.

- **Conteneur d'application client**

Il permet de gérer l'exécution des composants de l'application client. Il s'exécute sur le client.

- **Conteneur d'applets**

C'est une composition de deux parties : navigateur web et java plug-in qui fonctionnent ensemble sur le client, donc ce conteneur permet de gérer l'exécution des applets.

6.2. Les serveurs d'applications. [15]

Les API JAVA EE ont été projetés pour travailler avec un particulier type de serveur appelé JAVA EE Application Server. Un serveur d'application est une couche software résidente sur une machine serveur qui fournit les services que la technologie JAVA EE nécessite. Il y a plusieurs serveurs d'applications. Parmi les produits commerciaux on rappelle Bea WebLogic, IBM Websphere, Sun Application Server, ...etc. Parmi les produits libres le plus connu est JBOSS. Les différences principales entre les différents serveurs d'applications sont relatives aux opérations de deploy, clustering, ...etc. Par contre toutes les fonctionnalités qui concernent strictement le fonctionnement des applications JAVA EE adhèrent aux spécifications proposées par la Sun.

Un serveur d'application s'installe et se lance comme un normal serveur web (de fait il met à disposition des services typiques d'un serveur web). En plus il possède un panneau d'administration accessible par un poste distant à travers lequel on peut définir les différents services. On peut considérer le serveur d'application comme une application Java standalone exécutée par une J2SE qui dessert les requêtes provenant des différents clients.

Les serveurs d'application se sont développés depuis la création de JAVA EE. On peut distinguer principalement 2 grandes catégories de serveurs :

- **Open Source** : évolue grâce à la communauté par exemples **Tomcat : Apache, Jonas : ObjectWeb, JBoss : JBoss ...**
- **Propriétaire** : évolue selon l'éditeur par exemples WebSphere : IBM, WebLogic : BEA, WebObject : Apple, Oracle Application Server : Oracle ...

6.2.1. Le rôle d'un serveur d'application (ou bien les services) : [11]

Le rôle d'un serveur d'application est d'héberger des applications distribuées, fabriquées à base de composants Java (Servlet, JSP, EJB) et de les rendre accessibles à des clients web (navigateurs) et à des applications d'entreprise écrites en Java. Le serveur d'application doit être performant et fiable pour satisfaire aux exigences des applications d'entreprise. Le serveur offre différents services utiles pour les applications :

a. Service de nommage :

C'est le service le plus important dans un serveur d'application implémentant l'interface SPI (Service Provider Interface) et utilise généralement le protocole LDAP (Lightweight Directory Access Protocol). Ce dernier fournit des mécanismes de recherche, de localisation et de liaison.

b. Services de gestion de transaction :

Il permet de gérer des transactions locales mais surtout il prend en charge la gestion des transactions distribuées, indispensable pour les applications JAVA EE.

c. Service de gestion de la disponibilité des applications :

Il permet de gérer un grand nombre d'accès et de requêtes simultanés.

d. Service de sécurité :

L'administrateur du serveur d'application peut configurer des domaines de sécurité en définissant des utilisateurs (nom, mot de passe), des groupes d'utilisateurs, puis les permissions

accordées sur différents éléments par rapport aux composants et/ou aux ressources de l'application JAVA EE.

e. Service d'administration :

L'administrateur a la possibilité de consulter, de modifier, et d'ajouter des éléments à la configuration des applications d'entreprises et du serveur d'application.

f. Service d'accès aux données :

Ce service est une implémentation des extensions de L'API JDBC

g. Service de gestion de messages :

C'est une implémentation de l'API JMS, il permet de gérer une file de messages de traitements asynchrones.

7. Les avantages d'utiliser JAVA EE :

L'utilisation de JAVA EE pour développer et exécuter une application représente 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 les serveurs d'applications utilisés qu'ils soient commerciaux ou libres.

8. Conclusion

Dans ce chapitre, nous avons abordé les notions de base de la plate forme JAVA EE, certaines d'entre eux seront utilisées pour le développement de notre application.

Partie II : Étude préliminaire

Chapitre 3 : Présentation de l'organisme d'accueil

1. Introduction

Nous entamons cette partie par la présentation de l'organisme d'accueil qui est l'ANEM, l'Agence Nationale de L'Emploi. Nous commençons dans un premier temps par donner un aperçu global de l'ANEM, son organisation et ses missions. Puis, nous présenterons la sous-direction informatique de développement et des systèmes d'information au niveau de laquelle notre stage s'est déroulé.

2. Présentation de l'ANEM

L'ANEM, Agence Nationale de l'Emploi, est un service public d'intermédiation de l'emploi, qui assure la connaissance, la régularisation, la gestion et l'évolution du marché de l'emploi en Algérie. Elle offre aux demandeurs d'emploi et aux employeurs un service de recrutement personnalisé et efficace [ANEM, 2018a].



En 1991, l'Organisation Internationale du Travail (OIT) a recommandé aux états membres la mise en place du service public de l'emploi. Cette mission a été mise en œuvre par l'Agence Nationale de l'Emploi (ANEM) en succession à l'Office national de la main d'œuvre (ONAMO) qui était instauré en 1962. L'ANEM a changé de statut en 2006 d'une administration publique (EPA) à une Entreprise Publique à Gestion Spécifique (EPGS), ce qui a induit une réforme en profondeur du service public de l'emploi, pour l'adapter à l'évolution du monde du travail et aux besoins des demandeurs d'emploi [ANEM, 2018b].

3. Organisation de l'ANEM

L'ANEM est constituée de différents établissements et structures (voir **figure 10**). Elle compte :

- **La Direction Générale (DG)** : située dans la ville d'Alger, la DG s'occupe de la normalisation, du conseil et du contrôle des activités relatives à l'emploi. De plus, elle se charge du traitement des informations sur le marché de l'emploi captées par le réseau national d'agences.
- **La direction Régionale de l'Emploi (DREM)** : elle constitue la liaison entre la DG et les agences opérationnelles (AWEM et ALEM) et assure la supervision des activités de gestion financière et du personnel des agences appartenant à la région qu'elle contrôle. L'ANEM compte actuellement 11 DREM.
- **Agences de Wilaya de l'emploi (AWEM) et Agences Locales de l'emploi (ALEM)** : constituent les agences opérationnelles. Elles ont comme missions :
 - Accueillir, informer et proposer un emploi à tout demandeur d'emploi se présentant à l'agence ;
 - Prospecter les offres d'emploi au niveau local ;
 - Conseiller et accompagner les employeurs dans le processus de recrutement ;
 - Contribuer à la mise en œuvre des dispositifs d'emploi.

4. Les missions de l'ANEM

L'ANEM a plusieurs fonctions, dont on peut citer :

- Recueillir et mettre en relation les employeurs et les demandeurs d'emploi ;
- Renseigner de manière précise, fiable et régulière les changements du marché de l'emploi et de la main d'œuvre en Algérie ;
- Entreprendre des études et des enquêtes liées à sa mission ;
- Développer et normaliser les outils afin d'améliorer la fonction d'observation du marché de l'emploi.

5. Présentation de la DDSI

Notre stage s'est déroulé au niveau de la sous-direction de **D**éveloppement et des **S**ystèmes d'**I**nformation (DDSI). Cette structure contribue à la modernisation et à l'amélioration de la qualité des services de l'ANEM.

5.1. Organisation de la DDSI

La DDSI est constituée d'un sous-directeur et de plusieurs équipes de développement et d'administration des systèmes applicatifs de l'ANEM. L'organigramme de la DDSI est représenté par la **figure 9**.

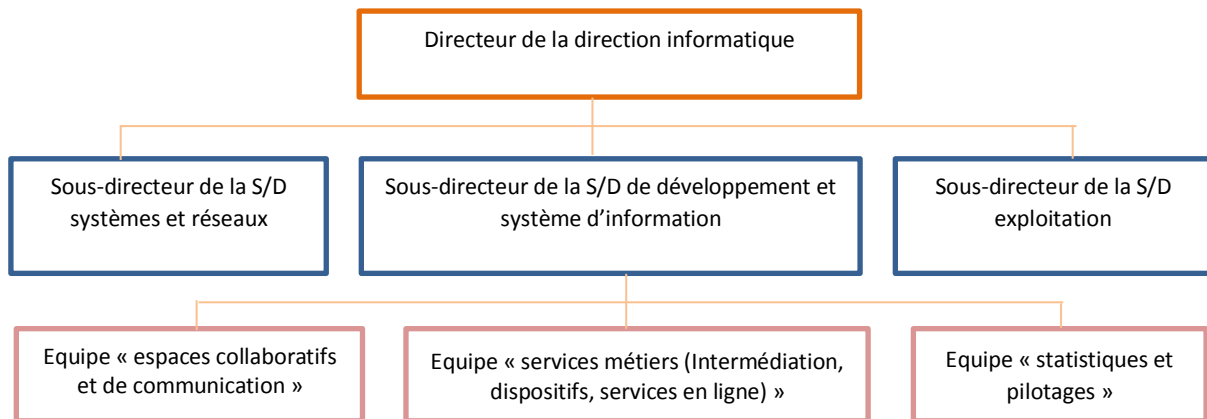


Figure 8: Organigramme de la sous-direction de développement et des systèmes d'information

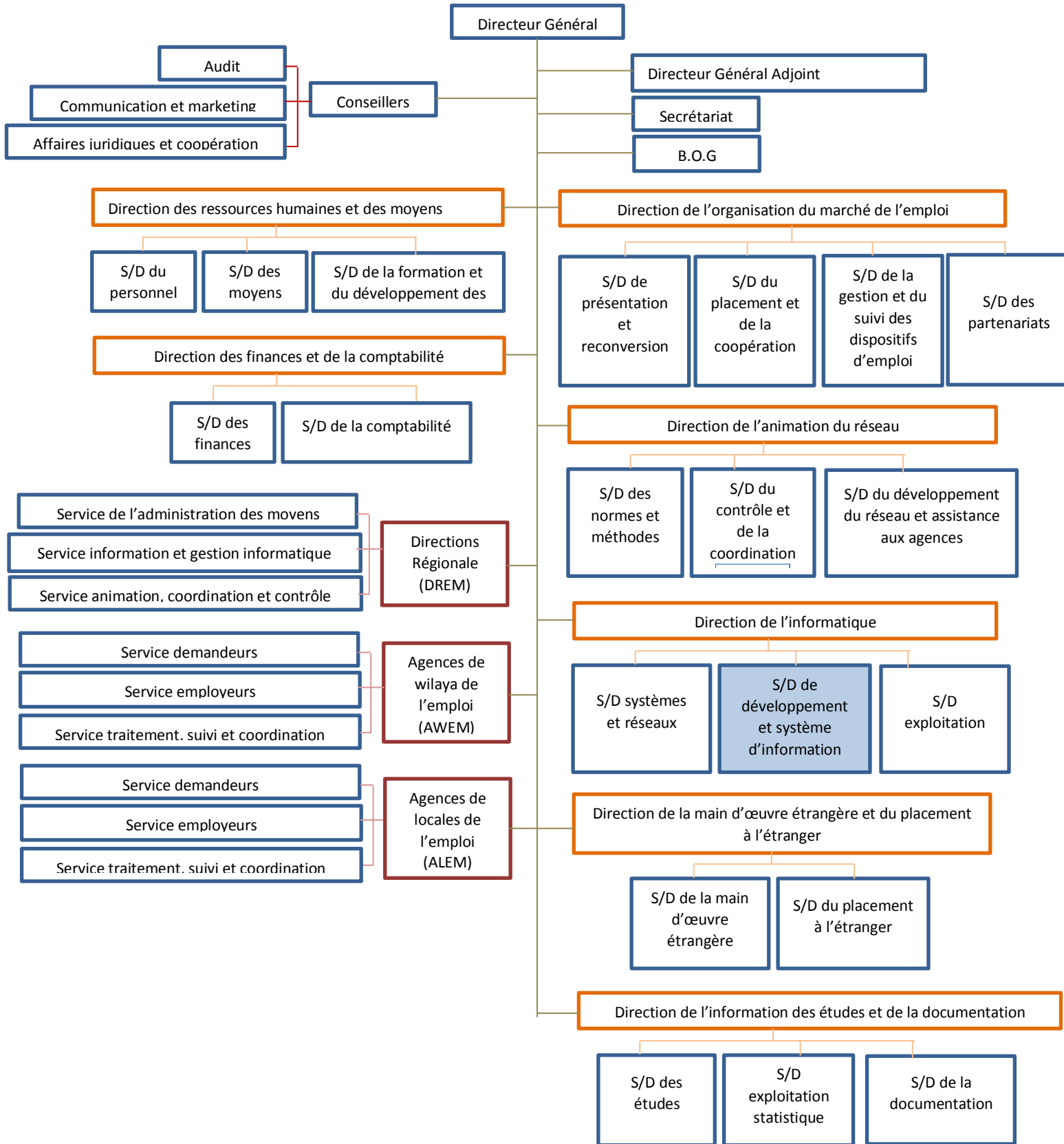


Figure 9: Organigramme de l'ANEM

5.2. Missions de la DDSI

La sous-direction informatique de développement et des systèmes d'information a pour missions :

- L'étude, la conception et la mise en place des différentes applications ;
- La réalisation des projets de collaboration et de communication afin d'assurer une bonne gestion des documents de l'entreprise et de centraliser le flux d'information entre le réseau des agences et la DG ;
- La réalisation de projets BI pour solutionner les problèmes décisionnels de l'ANEM et permettre une meilleure compréhension du marché de l'emploi ;
- L'administration des bases de données et le développement des systèmes de l'ANEM.

6. Conclusion

À travers ce chapitre, nous avons présenté l'organisme d'accueil, qui est l'Agence Nationale de l'Emploi (ANEM), sa structure, ses missions et ses objectifs. Nous avons également évoqué la sous-direction de développement et des systèmes d'information (DDSI), au niveau de laquelle nous avons effectué notre stage.

Dans le prochain chapitre, nous procédons à l'étude de l'existant informatique de l'ANEM, en particulier le système d'information Gest_DAIP, en vue de bien cadrer notre projet et de formuler les objectifs à atteindre.

Chapitre 4 : Étude de l'existant

1. Introduction

La refonte d'un système d'information est un projet complexe impliquant des changements fonctionnels et techniques. Il nécessite une étude approfondie du système à refaire afin de bien cerner ses dysfonctionnements et ses insuffisances, et de proposer une meilleure conception visant à l'améliorer. Dans cette optique, nous avons réalisé une étude détaillée du système existant, ainsi que toute autre source de documentation jugée pertinente.

Dans ce chapitre, nous commençons par la présentation du système Gest_DAIP mis en place par l'ANEM dans sa globalité. Puis, nous procédons à l'examination et l'évaluation du système « Gest_DAIP ». Enfin, nous délimitons le périmètre et les objectifs de notre projet.

2. Présentation de Gest_DAIP

GEST DAIP est un système d'information qui s'occupe du suivi et du traitement des contrats des bénéficiaires du dispositif d'aide d'insertion professionnel gérés entre L'ALEM, L'AWEM et DEW.

Il est destiné aux conseillers de l'emploi ; les informaticiens et les cadres de la direction de l'emploi.

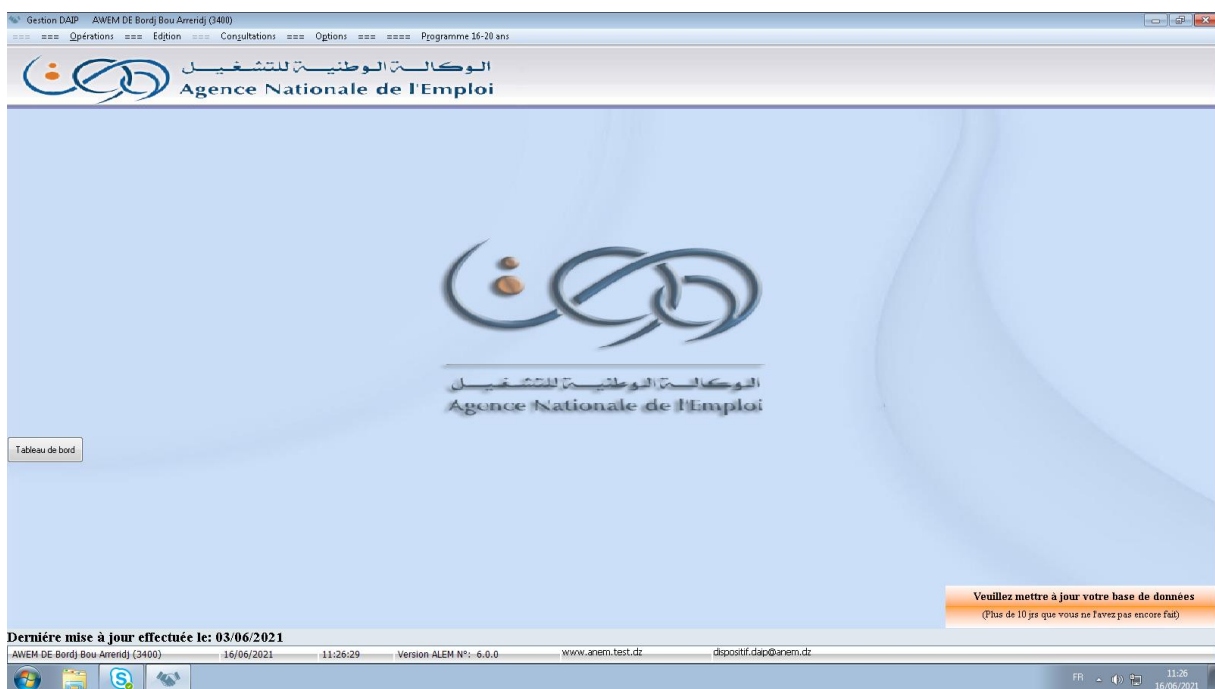


Figure 10: Page d'accueil de system Gest_DAIP

3. Acteurs du système GEST DAIP

Le système GEST DAIP compte trois types d'utilisateurs : Les conseillers de l'ALEM, les informaticiens de l'AWEM et les cadres de la DEW.

- **Administrateur du système:** Ce sont les informaticiens de l'ANEM Gestionnaire des comptes
- **L'ALEM :** Ce sont les cadres chargés de la gestion du dispositif tel que :
L'établissement des contrat et l'est envoyer au service AWEM
La création des changements de situation
- **L'AWEM :** C'est un intermédiaire entre ALEM et DEW elle est chargé du suivi des contrats, suivi du pointage DAIP et des changements de situation
- **DEW :** Ce sont les cadres de la direction de l'emploi qui font:
 - La validation des contrat et l'installation des bénéficiaires DAIP
 - La validation des demandes de changement de situation
 - Le payement des bénéficières .

4. Fonctionnalités de Gest_DAIP:

4.1. Les fonctionnalités relatives au conseillers de L'ALEM:

4.1.1. Inscription du demandeur :

- Saisie de l'état civil et les informations du contact
- Saisie du profil (niveau scolaire, diplôme et poste de travail)
- Autre information (service national, situation familial, pièce d'identité)

4.1.2. Inscription de l'employeur :

- Saisie des informations de l'employeur : raison sociale, les informations du contact,

RIP, NIF, N° bancaire, N°CNAS/CASNOS

4.1.3. Traitement de l'offre :

- Saisie de l'offre (post de travail, nombre)
- Rapprochement entre la demande et l'offre

4.1.4. Changement des contrats:

- Prolongation du contrat
- Création Reconduite CTA (transfert de contrat DAIP au CTA)

4.1.5. Impression :

- Imprimer contrat
- Imprimer prolongation
- Imprimer contrat DAIP
- Imprimer fiche de pointage
- Imprimer PV d'installation
- Imprimer et envoyer des bordereaux

4.2. Les fonctionnalités relatives à L'AWEM:

- Impression des bordereaux
- Mise à jour
- Déclaration des incidents

4.3. Les fonctionnalités relatives au DEW

1.Traitement des contrats :

- Validation des contrat DAIP et CTA
- Rejet pour correction
- Rejet définitif.
- Installation contrat DAIP et CTA
- Validation de la prolongation

2.traitement paye et rappel :

- Impression des pièces paye (état de sortie, CH50, fiche de paye)
- Impression pièces CNAS (ATS - maternité ...)
- Arrêt du contrat (intégration – résiliation- démission - Suspension)

5. Périmètre et objectifs du projet

Dans l'optique de régler une partie des problèmes du système Gest –Daip qui est limité :

- L'application desktop client lourd
- Collaboration hors système
- Les statistiques manuelles
- 360 Bases de données reparties
- Manque de traçabilité
- Processus fonctionnels hétérogène

Les principaux objectifs de notre projet s'articulent sur les points suivants :

- Mise en place d'un nouveau système accessible sur internet pour le suivi de l'opération d'intégration des bénéficiaires DAIP
- Rendre disponible l'information liée à l'intégration des bénéficiaires DAIP dans les différentes tutelles.
- Suivi des différents indicateurs d'intégration des bénéficiaires DAIP

6. Conclusion

À travers ce chapitre, nous avons présenté le système Gest-Daip et ses acteurs principaux nous avons mis l'accent sur les Fonctionnalités de Gest_Daip. Enfin nous avons délimité le périmètre du projet et mis en évidence ses objectifs.

Partie III :

Notre contribution

Chapitre 5 : Analyse et conception

1. Introduction :

Après avoir élaborer les différents concepts nécessaires à la réalisation de notre application, nous abonderons la phase Analyse et réalisation.

Le but principal d'un système logiciel est de rendre services à ses utilisateurs.

Le processus de développement sera donc concentré sur l'utilisateur ; et c'est en passant par l'analyse et conception qu'on peut définir ces besoins.

Pour cette étape de notre application nous avons adopté la conception avec UML et grâce à son extension pour le web, il nous a permis de représenter la dynamique d'une application par une série de diagrammes.

2. Définition d'UML :

UML signifie « Unified Modeling Language » ou Langage de modélisation unifié en français. C'est un langage de modélisation qui permet de représenter graphiquement les besoins des utilisateurs. On utilise pour cela des diagrammes.

UML est une démarche qui se base sur une approche objet. Cette approche s'appuie sur 4 principes fondamentaux.

L'approche objet nécessite une démarche itérative et incrémentale, c'est-à-dire que le concepteur doit faire des allers-retours entre les diagrammes initiaux et, les besoins du client et des utilisateurs perçus au fur et à mesure de la conception du logiciel afin de le modifier si nécessaire.

L'approche objet est guidée par les besoins du client.

L'approche objet est centrée sur le diagramme de classes qui décrit aussi bien des actions que des informations dans une même entité. Les autres diagrammes nous aident à voir clair dans les besoins et dans la solution qui est à développer. Ils permettent de compléter le diagramme de classes.

3. Démarche de l'analyse:

La modélisation comporte deux composants :

1.la phase d'analyse : on cherche d'abord à bien comprendre et à décrire de façon précise les besoins des utilisateurs ou des clients. Que souhaitent-ils faire avec le logiciel ? Quelles fonctionnalités veulent-ils ? Pour quel usage ? Comment l'action devrait-elle fonctionner ? C'est ce qu'on appelle « **l'analyse des besoins** ».

2.la phase de conception : on apporte plus de détails à la solution et on cherche à clarifier des aspects techniques, tels que l'installation des différentes parties logicielles à installer sur du matérielle.

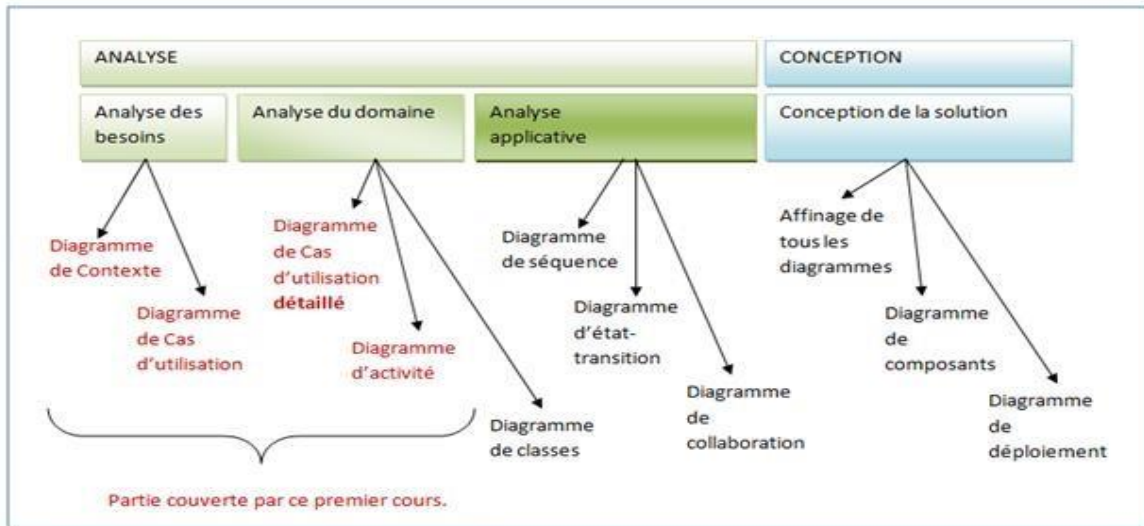


Figure 11: Démarche de modélisation

3.1. Phase d'analyse :

L'étape d'analyse est primordiale pour la réalisation et le développement d'une application, ou d'un système informatique, elle permet de faire une étude approfondie du domaine que couvre notre application et cela dans le but de cerner, de comprendre et de définir les besoins.

3.1.1. Diagramme de contexte :

Le diagramme de contexte est un modèle conceptuel de flux qui permet d'avoir une vision globale des interactions entre le système et les liens avec l'environnement extérieur. Il permet aussi de bien délimiter le champ d'étude. Pour notre cas le diagramme de contexte est donné par la figure suivante :

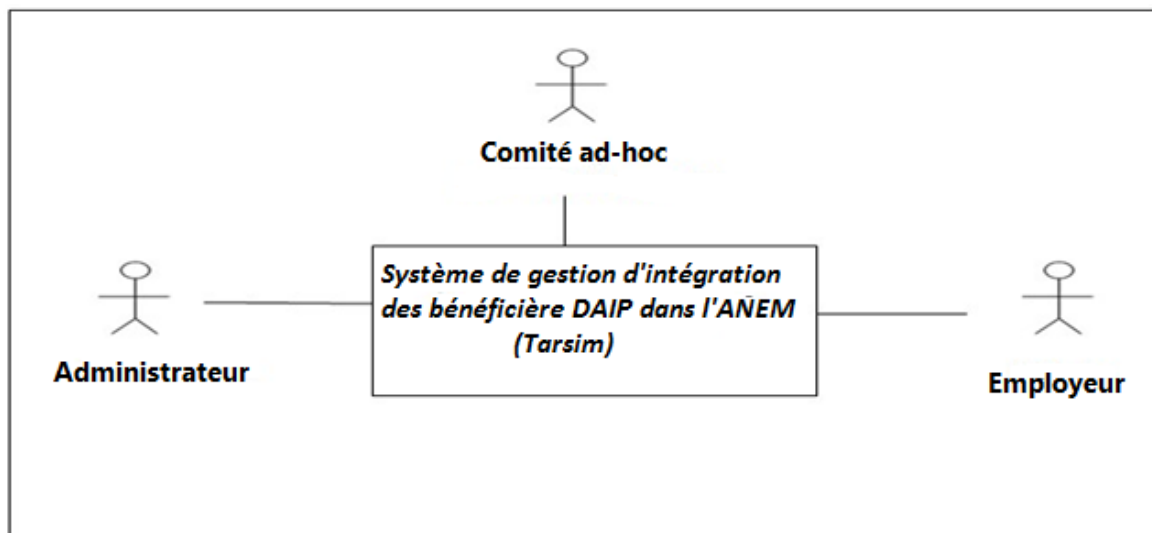


Figure 12: Diagramme de contexte

3.1.2. Identification des acteurs:

Notre système communique avec des acteurs afin de réaliser un ensemble d'opérations. Un acteur représente un ensemble de rôles joués par des entités externes (utilisateur humain, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Il peut consulter et/ou modifier directement l'état du système, en émettant et/ou recevant des messages susceptibles d'être porteur des données.

L'acteur est associé à un cas d'utilisation, c'est-à-dire qu'il peut interagir avec lui et participer à son scénario. Les acteurs candidats sont :

- ✓ **Employeur** : les établissement Publique.
- ✓ **Comité ad-hoc** : représenter par (DEW, AWEM, fonction publique, Contrôleur financier, wali).
- ✓ **Administrateur** : responsable de la gestion des comptes(ANEM).

3.1.3. Identification des cas d'utilisation :

Un cas d'utilisation est une unité cohérente d'une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service.

Chacun des acteurs que nous avons définis effectue un certain nombre de tâches, nous allons donc les résumer dans le tableau suivant :

Acteur	Cas d'utilisation
Employeur	<ul style="list-style-type: none"> ➤ S'authentifier. ➤ Consulte les contrats DAIP. ➤ Rechercher des Contrats DAIP. ➤ MAJ Contrat DAIP. • Modifier Etat Contrat. • Renseigner l'employeur d'intégration. • Renseigner poste d'intégration ➤ Consulter PV Comité ad-hoc. ➤ Rechercher PV Comité ad-hoc ➤ Créer PV Comité ad-Hoc. • Renseigner poste d'intégration ➤ MAJ PV Comité ad-hoc. • Modifier PV. • Ajouter /supprimer des contrats dans PV. ➤ Imprimer PV brouillon. ➤ Consulter Besoin en Postes budgétaires. ➤ MAJ Besoin en Postes budgétaires. ➤ Rechercher Besoin en Postes budgétaires. ➤ Consulter Ses statistique d'intégration ➤ Imprimer PV final
Comité Ad-hoc	<ul style="list-style-type: none"> ➤ S'authentifier. ➤ Valider PV Comité ad-hoc. ➤ Consulter statistique.
Administrateur	<ul style="list-style-type: none"> ➤ S'authentifier. ➤ Gérer les utilisateurs. <ul style="list-style-type: none"> • Ajouter un utilisateur. • Modifier un utilisateur. • Supprimer un utilisateur.

Tableau II: Structuration de cas d'utilisation globale

3.1.4. Identification des scénarios :

Un scénario représente une succession particulière d'enchaînement étant l'unité de description de séquence d'actions.

Le tableau suivant représente les différents scénarios associés à chaque acteur :

⇒ **Administrateur :**

Tâches	Scénarios
T01 : S'authentifier	<p>S01 : Le système affiche la page d'authentification</p> <p>S02 : L'Administrateur saisit son login et son mot de passe et profil, puis clique sur le bouton «Se Connecter».</p> <p>S03 : Le système renvoie la page d'accueil de l'administrateur si les informations saisit sont valide, sinon il renvoie un message d'erreur.</p>
T02 : Consulter la liste des utilisateurs	<p>S04 : L'Administrateur clique sur le lien «Liste Utilisateurs».</p> <p>S05 : Le système retourne la liste des utilisateurs existants.</p>
T03 : Consulter les statistiques d'intégration.	<p>S06 : L'Administrateur clique sur le lien «Statistique»</p> <p>S07 : Le système retourne la page des statistiques à consulter.</p>
T04 : Ajouter un compte utilisateur	<p>S08 : L'administrateur clique sur le bouton «Ajouter un utilisateur».</p> <p>S09 : Le système affiche un formulaire d'ajout.</p> <p>S10 : L'administrateur remplit le formulaire d'ajout puis il clique sur le bouton «Ajouter».</p> <p>S11 : Le système vérifie la validité des informations reçues et affiche un message de confirmation.</p>
T05 : Désactiver un compte Utilisateur	<p>S12 : L'administrateur sélectionne l'utilisateur à désactivé puis il clique sur le bouton «Désactiver un compte» pour désactiver le compte utilisateur.</p>
T06 : Rechercher un utilisateur	<p>S13 : L'administrateur saisi le nom d'un utilisateur et clique sur le bouton «Rechercher».</p> <p>S14 : Le système lui affiche l'utilisateur recherché</p>

T07 : Modifier un utilisateur	<p>S15 : L'Administrateur clique sur le bouton «Modifier utilisateur».</p> <p>S16 : Le système affiche un formulaire qui contient les champ à modifier.</p> <p>S17 : L'Administrateur remplit le champ à modifier et valide.</p> <p>S18 : Le système vérifier la cohérence des données reçus et renvoie un message de confirmation de changement.</p>
T08 : Déconnexion	<p>S19 : L'Administrateur clique sur le bouton « Déconnecter ».</p> <p>S20 : Le système renvoie la page login.</p>

Tableau III: Scénarios des tâches d'un Administrateur

⇒ Employeur :

Tâches	Scénarios
T09 : S'authentifier	<p>S21 : Le système affiche la page d'authentification</p> <p>S22 : L'employeur saisit son login et son mot de passe, puis clique sur le bouton «Se Connecter».</p> <p>S23 : Le système renvoie la page d'accueil du l'employeur si les informations saisit sont valide, sinon il renvoie un message d'erreur.</p>
T10 : Consulter la liste des Contrats DAIP.	<p>S24 : L'employeur clique sur le lien «liste Contrats DAIP».</p> <p>S25 : Le système affiche la liste des contrats de l'employeur.</p>
T11 :Rechercher un Contrat DAIP	<p>S26 : L'employeur saisi les information d'un Contrat et clique sur le bouton «Rechercher».</p> <p>S27 :Le système lui affiche le contrat recherché.</p>
T12 : Modifier l'état d'un contrat.	<p>S28 : L'employeur clique sur « Modifier »</p> <p>S29 : Le système affiche la liste des états à modifier.</p> <p>S30 : L'employeur sélectionne l'état à modifier.</p> <p>S31 : L'employeur modifier et valide.</p> <p>S32 : Le système renvoie un message de succès.</p>

<p>T13 : Renseigner l'employeur d'un contrat.</p>	<p>S33 : L'employeur clique sur « Modifier »</p> <p>S34 : Le système affiche la liste de référentiel employeur.</p> <p>S35 : L'employeur sélectionne l'employeur d'intégration.</p> <p>S36 : L'employeur modifier et valide.</p> <p>S37 : Le système renvoie un message de succès.</p>
<p>T14 : Renseigner le poste d'un contrat.</p>	<p>S38 : L'employeur clique sur « Modifier »</p> <p>S39 : Le système affiche la liste de référentiel grade fonction publique.</p> <p>S40 : L'employeur sélectionne le poste.</p> <p>S41 : L'employeur modifier et valide.</p> <p>S42 : Le système renvoie un message de succès.</p>
<p>T15 : Ajouter PV.</p>	<p>S43 : L'employeur clique sur «Ajouter un PV ».</p> <p>S44 : Le système affiche un formulaire d'ajout d'un PV.</p> <p>S45 : L'employeur remplit le formulaire et valide.</p> <p>S46 : Le système retourne un message de confirmation d'ajout.</p>
<p>T16 : Consulter les PV.</p>	<p>S47 : L'employeur clique sur le lien « Liste des PV ».</p> <p>S48 : Le système affiche la liste des PV de L'employeur.</p>
<p>T17 : Modifier un PV.</p>	<p>S49 : L'employeur clique sur « Modifier un PV »</p> <p>S50 : Le système affiche un formulaire de modification.</p> <p>S51 : L'employeur modifier et valide.</p> <p>S52 : Le système renvoie un message de succès.</p>
<p>T18 :Rechercher un PV</p>	<p>S53 : L'employeur saisi les information d'un PV et clique sur le bouton «Rechercher ».</p> <p>S54 :Le système lui affiche le PV recherché.</p>
<p>T19 : Ajouter un besoin.</p>	<p>S55 : L'employeur clique sur « Ajouter un besoin ».</p> <p>S56 : Le système affiche un formulaire d'ajout d'un besoin.</p> <p>S57 : L'employeur remplit le formulaire et valide.</p>

	S58 : Le système retourne un message de confirmation d'ajout.
T20 : Consulter les besoin.	S47 : L'employeur clique sur le lien « Liste des besoins ». S48 : Le système affiche la liste des besoins de L'employeur.
T21 : Modifier un besoin.	S49 : L'employeur clique sur « Modifier un besoins » S50 : Le système affiche un formulaire de modification. S51 : L'employeur modifier et valide. S52 : Le système renvoie un message de succès.
T22 : Rechercher un besoin.	S53 : L'employeur saisi les information d'un besoin et clique sur le bouton « Rechercher ». S54 :Le système lui affiche le besoin recherché.
T23 : Déconnexion	S55 : L'employeur clique sur le bouton « Déconnecter ». S56 : Le système renvoie la page login.

Tableau IV:Scénarios des tâches d'un employeur.

⇒ Comité ad-hoc :

Tâches	Scénarios
T24 : S'authentifier	S21 ;S22 ;S23 ;
T25 : Consulter la liste des Contrats DAIP.	S57 : le comité ad-hoc clique sur le lien « liste Contrats DAIP ». S58 : Le système affiche la liste des contrats existants.
T26 :Rechercher un Contrat DAIP	S26 ; S27 ;
T27 : Modifier l'état d'un contrat.	S28 ;S29 ;S30 ;S31 ;S32 ;
T28 : Renseigner l'employeur d'un contrat.	S33 ;S34 ;S35 ;S36 ;S37 ;

T29 : Renseigner le poste d'un contrat.	S38 ;S39 ; S40 ;S41 ;S42 ;
T30 : Ajouter PV.	S43 ;S44 ;S45 ; S46 ;
T31 : Consulter les PV.	S59 : le comité ad-hoc clique sur le lien « Liste des PV ». S60 : Le système affiche la liste des PV existants.
T32 : Modifier un PV.	S61 : le comité ad-hoc clique sur « Modifier un PV» S62 : Le système affiche un formulaire de modification. S63 : le comité ad-hoc modifier et valide. S64 : Le système renvoie un message de succès.
T33 :Rechercher un PV	S65 : le comité ad-hoc saisi les information d'un PV et clique sur le bouton «Rechercher ». S66 :Le système lui affiche le PV recherché.
T34 : Ajouter un Besoin en Postes budgétaires.	S55 ; S56 ;S57 ;S58 ;
T35 : Consulter les Besoin en Postes budgétaires.	S67 : le comité ad-hoc clique sur le lien «Liste des Besoin en Postes budgétaires ». S68 : Le système affiche la liste des Besoin en Postes budgétaires de L'employeur.
T36 : Modifier un Besoin en Postes budgétaires.	S49 ; S50 ; S51 ; S52 ;
T37 : Rechercher un Besoin en Postes budgétaires.	S69 : le comité ad-hoc saisi les information d'un Besoin en Postes budgétaires et clique sur le bouton «Rechercher ». S70 :Le système lui affiche le besoin recherché.
T38 : Déconnexion	S71 : le comité ad-hoc clique sur le bouton «Déconnecter ». S72 : Le système renvoie la page login.

Tableau V:Scénarios des tâches d'un Comité ad-hoc

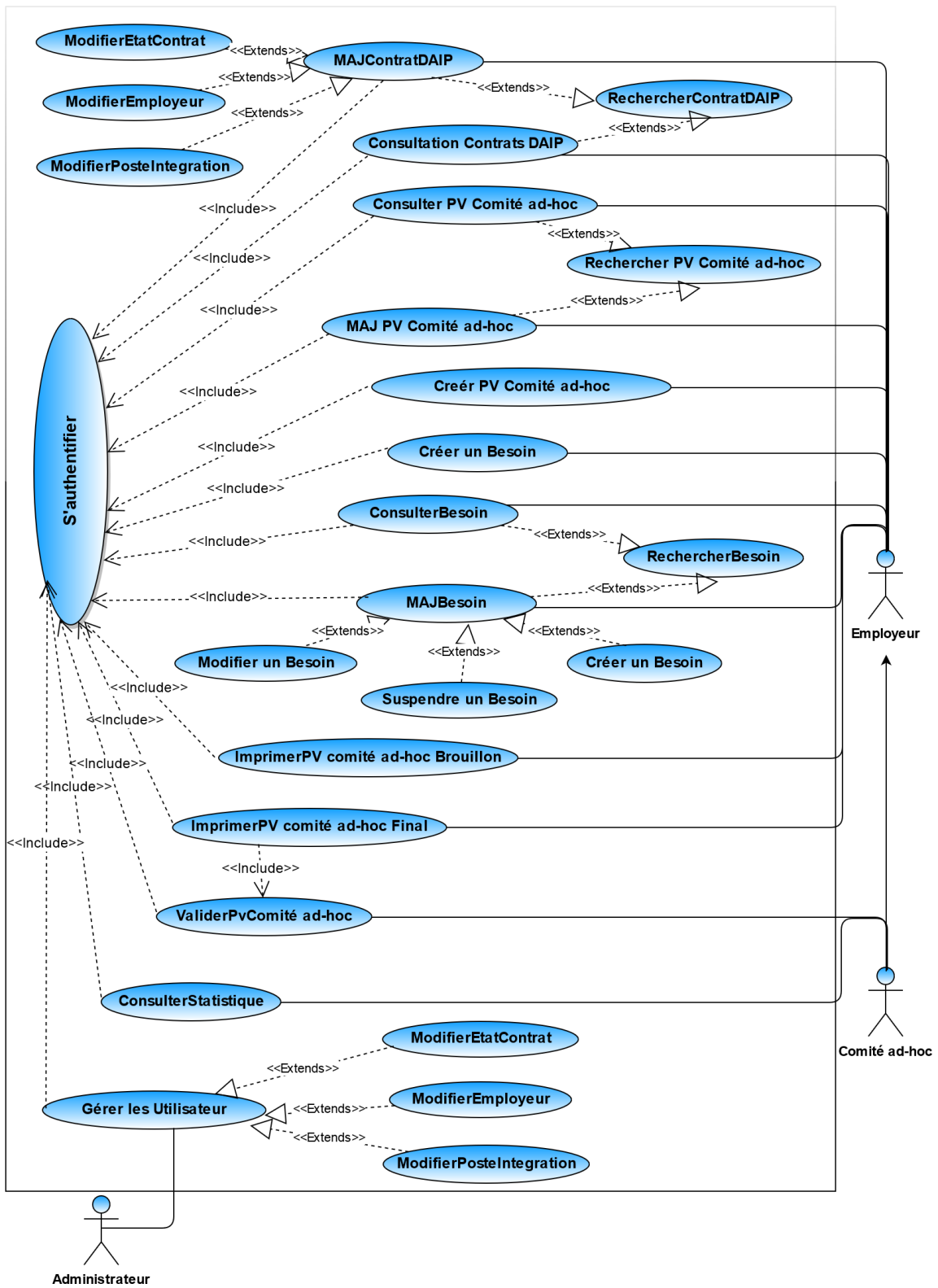
3.1.5. Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation permettent de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Les cas d'utilisation sont décrits de manière textuelle, agrémentée de quelques diagrammes d'interaction. A ce stade de la modélisation, les interactions représentent les principaux événements qui se produisent dans le domaine de l'application. Plus tard, lors de conception, ces événements sont traduits en messages qui déclenchent des opérations.

- ✓ La relation d'inclusion (include) : Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. Cette relation permet de décomposer des comportements et de définir les comportements partageables entre plusieurs cas d'utilisations.
- ✓ La relation d'extension (Extend) : Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à des conditions.

Pour notre projet voici les cas d'utilisation dans la figure ci-après :

Figure 13: Diagramme de cas d'utilisation



3.2. Conception :

3.2.1. Diagramme de séquence

C'est un diagramme qui permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets (David & GABAY, 2008). Il comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction.

Cette partie va dresser le diagramme de séquence de certains cas d'utilisation décrits plus haut.

❖ Diagramme de séquence s'authentifier

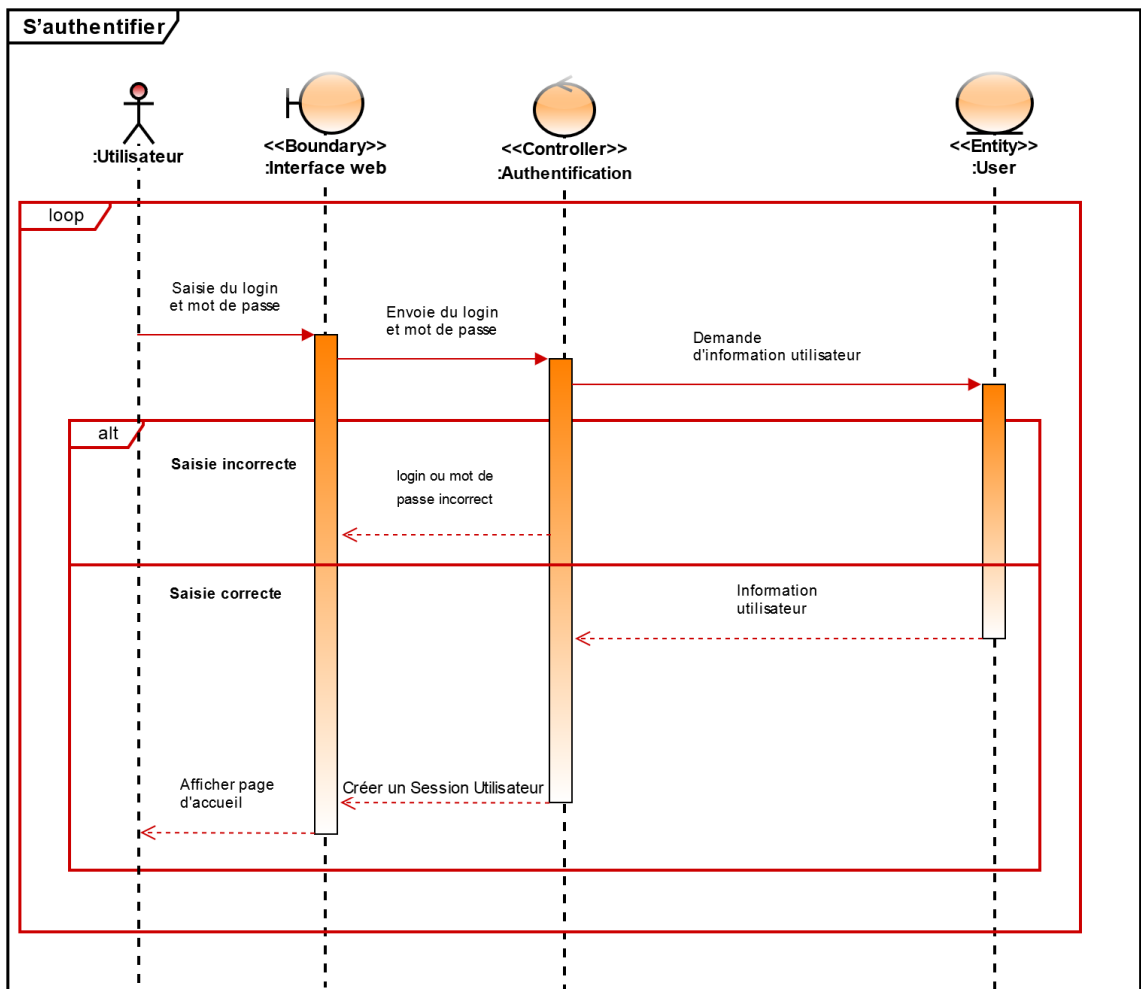


Figure 14: Diagramme de séquence s'authentifier

❖ Diagramme de séquence du cas d'utilisation « consulter PV Comité ad-hoc »

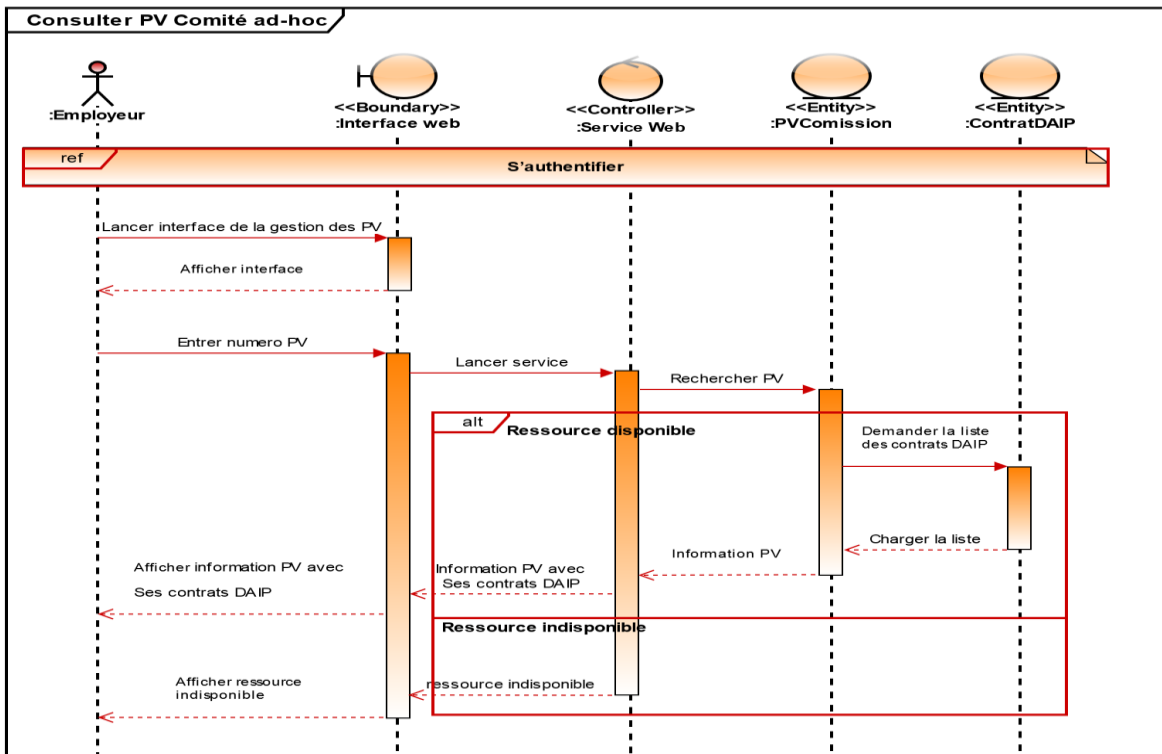


Figure 15: Diagramme de séquence « consulter PV comité ad-hoc »

❖ Diagramme de séquence « ajouter PV Comité ad-hoc »

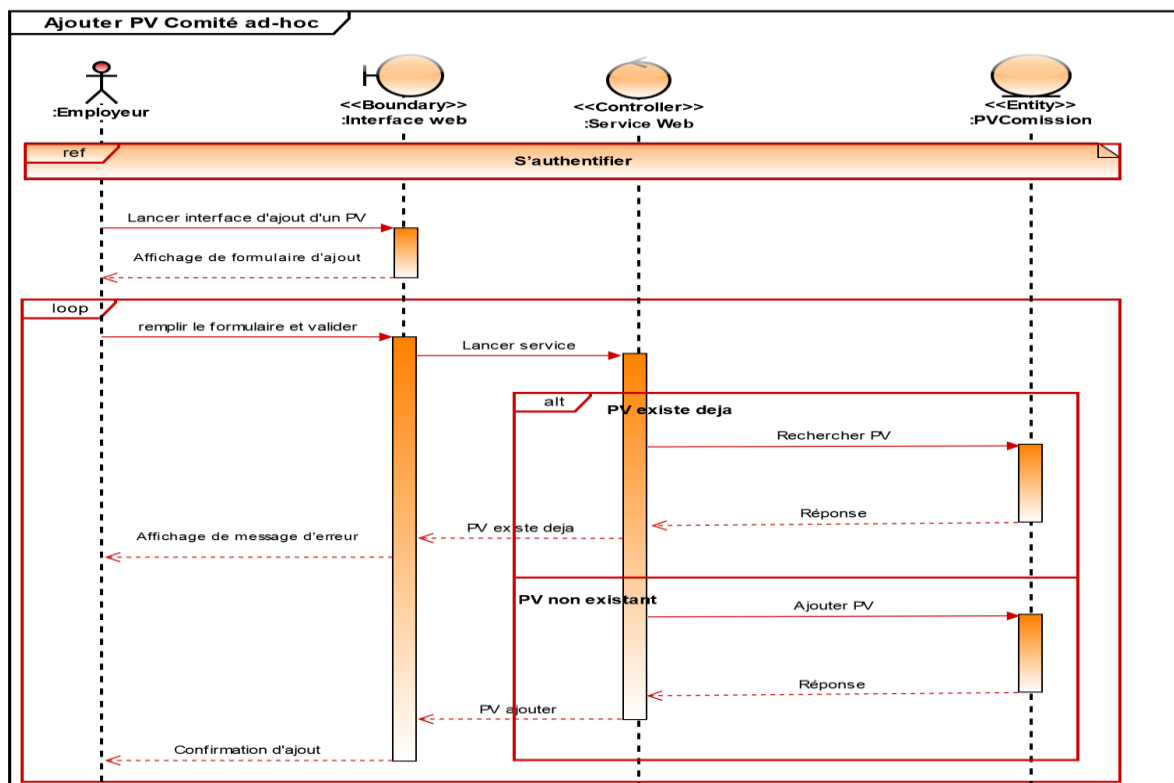


Figure 16: Diagramme de séquence « ajouter PV comité ad-hoc »

❖ Diagramme de séquence « Consulter Besoin en Postes budgétaires »

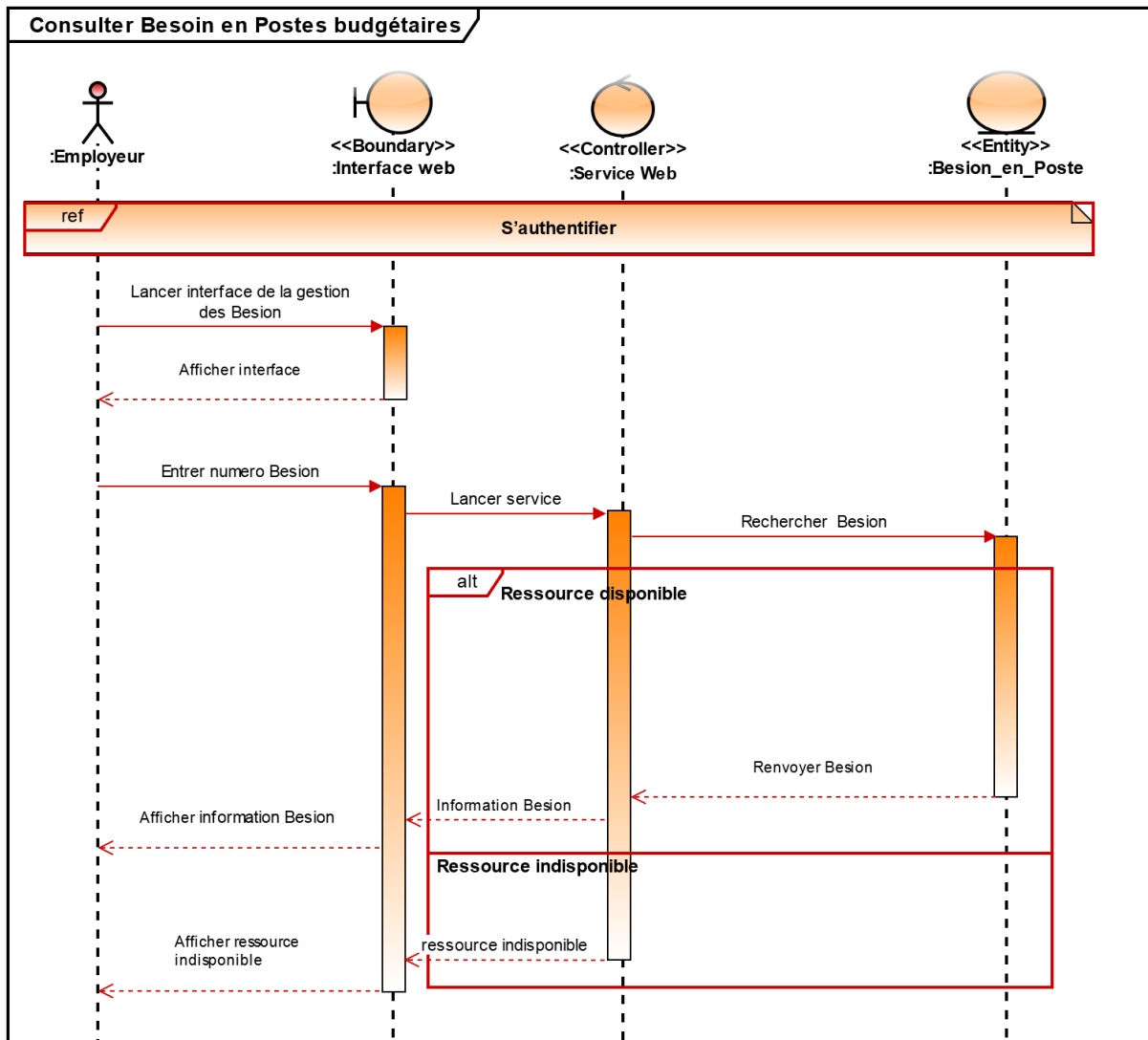


Figure 17: Diagramme de séquence « Consulter Besoin en Postes budgétaires »

3.2.2. Diagramme d'activité

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

❖ Diagramme d'activité « authentification »

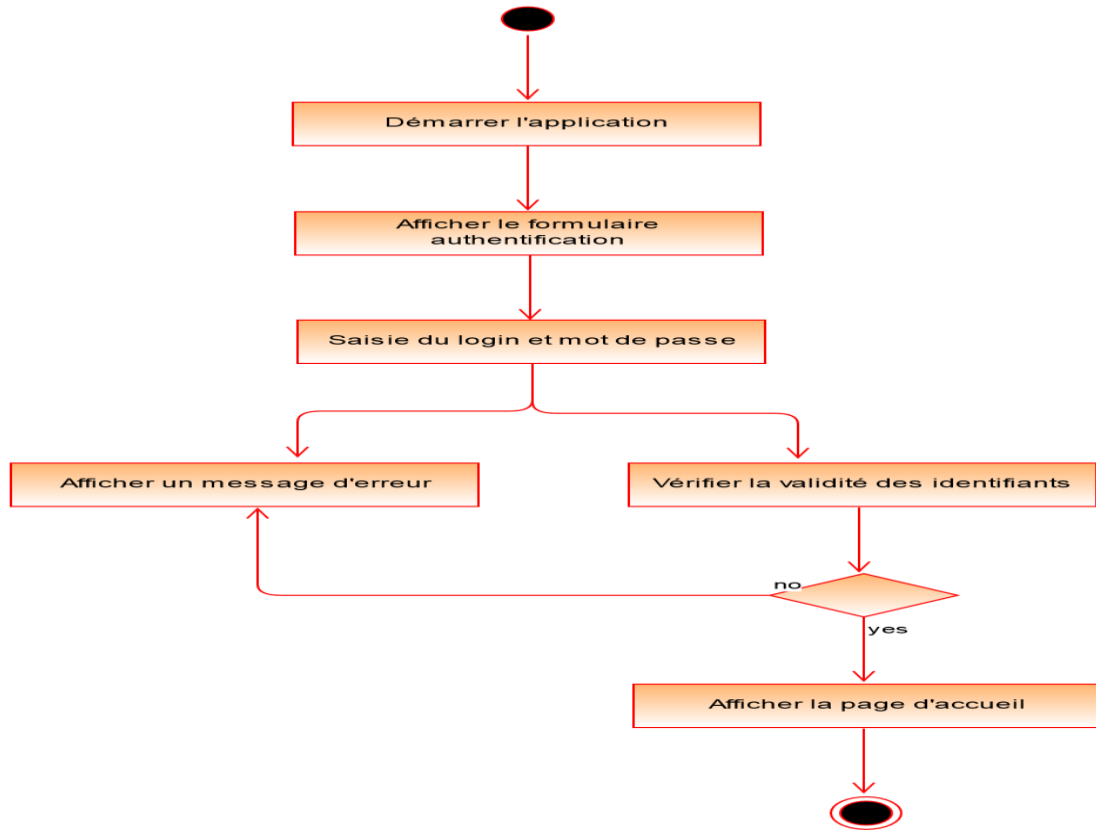


Figure 18:Diagramme d'activité « authentification »

❖ Diagramme d'activité « Imprimer PV Comité ad-hoc »

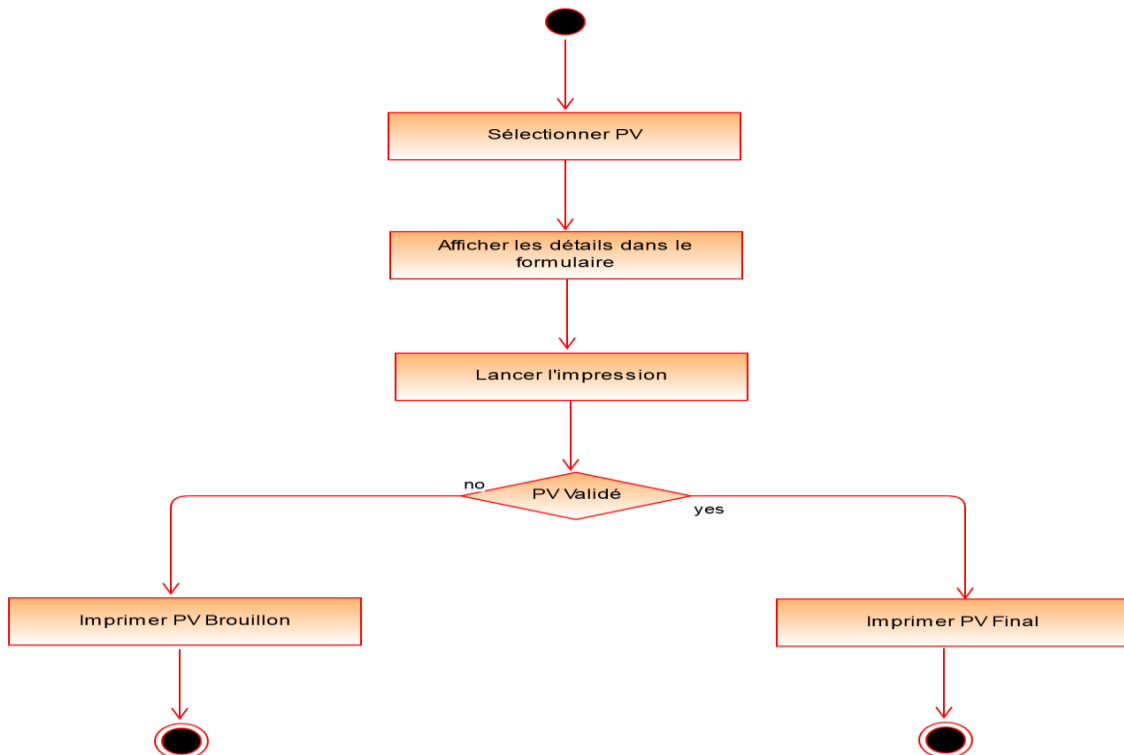


Figure 19:Diagramme d'activité « Imprimer PV Comité ad-hoc »

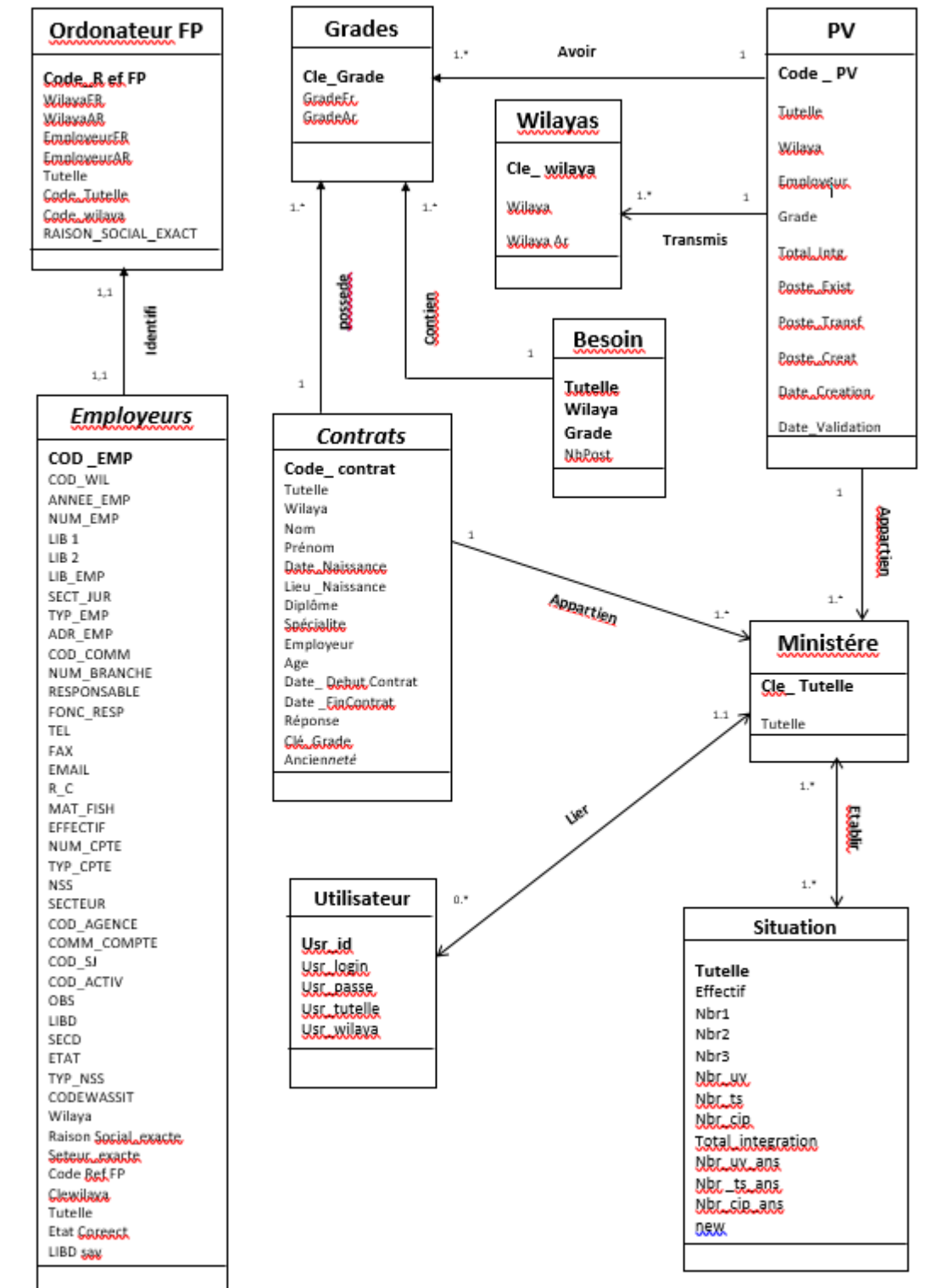


Figure 20: Diagramme de classe

4. Conclusion

Cette partie importante a consisté à l'analyse et la conception du système. Nous avons grâce au diagramme UML modéliser le système depuis les cas d'utilisation jusqu'à l'analyse aboutissant au diagramme d'activité. Le prochain chapitre va maintenant concerner la mise en œuvre du système avec la présentation de quelques interfaces.

Chapitre 6 : Réalisation

1. Introduction

Après avoir présenté la conception de la solution à travers les différents volets : intégration, API REST et application cliente, nous allons dans ce chapitre aborder les différents aspects de réalisation de notre solution.

Nous commençons par la présentation des outils de développement. Puis, nous illustrons les résultats de l'implémentation.

2. Les outils et langages de développement



Java EE : Java Entreprise Edition est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles [17] :

- Un ensemble de spécification pour une infrastructure dans laquelle s'exécutent les composants écrits en Java : un tel environnement se nomme serveur d'application.
- Un ensemble d'APIs 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 Java Entreprise Edition : le Java EE 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 [17].

Java Entreprise Edition permet une grande flexibilité dans le choix de l'architecture de l'application en combinant différents composants. Ce choix dépend des besoins auxquels doit répondre l'application mais aussi des compétences dans les différentes API de Java Entreprise Edition.

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 web ou d'applets.
- **La partie métier** : c'est la partie qui encapsule les traitements (dans des EJB ou des Java Beans).
- **La partie donnée** : c'est la partie qui stocke les données.

L'utilisation de Java Entreprise Edition pour développer et exécuter une application, offre plusieurs avantages :

- Une architecture d'application basée sur des 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, ...
- La possibilité de choisir les outils de développement et les serveurs d'applications utilisés qu'ils soient commerciaux ou libres.



NetBeans IDE: est un environnement de développement intégré (IDE) gratuit, open source, il permet de développer des applications de bureau, mobiles et Web. L'IDE prend en charge le développement d'applications dans divers langages, notamment Java, HTML5, PHP et C ++. L'EDI fournit un support intégré pour le cycle de développement complet, de la création de projet au débogage, au profilage et au déploiement. L'EDI s'exécute sous Windows, Linux, Mac OS X et d'autres systèmes UNIX[22].

L'IDE fournit une prise en charge complète des technologies JDK 8 et des améliorations Java les plus récentes. C'est le premier IDE à prendre en charge JDK 8, Java EE 7 et JavaFX 2. L'EDI prend entièrement en charge Java EE en utilisant les dernières normes pour Java, XML, les services Web et SQL et prend entièrement en charge le serveur GlassFish de Java EE[22].

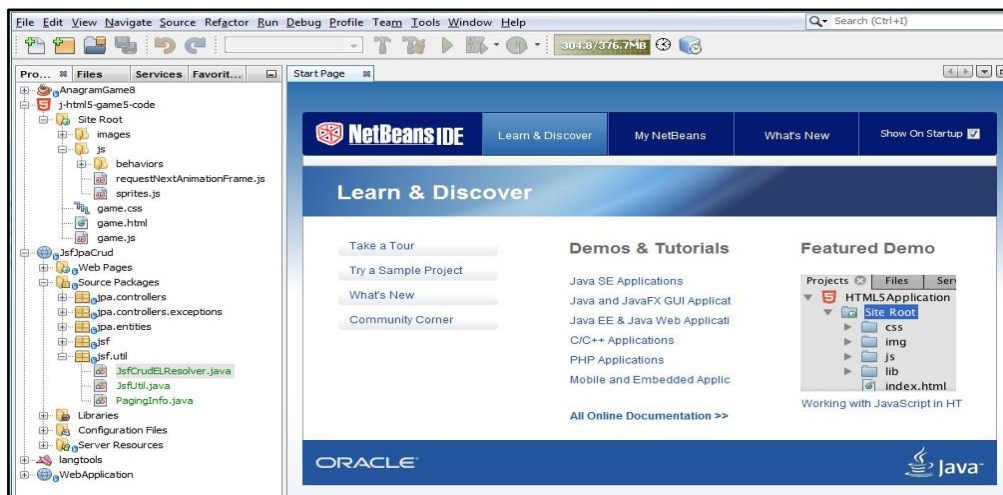


Figure 21: Interface de l'IDE Netbeans



GlassFish : est le nom du serveur d'applications Open Source Java EE 5 et désormais Java EE 7 avec la version 4.1 qui sert de socle au produit Oracle GlassFish Server4 (anciennement Sun Java System Application Server5 de Sun Microsystems). C'est la réponse aux développeurs Java désireux d'accéder aux sources et de contribuer au développement des serveurs d'applications de nouvelle génération. La distribution dite Open Source Edition est placée sous double licence CDDL et GPLv2. GlassFish est certifié Java EE 5 (EJB3 + JPA + JSF + JAX-WS 2.x + ...) et Java EE 6 (EJB 3.1, CDI, JSF 2.0, JAX-RS 1.1, ...), la version utilisée est GlassFish 4.1.1.



JSF (Java Server Faces) : JSF est un Framework de type MVC, destiné aux applications web respectant l'architecture Java EE.

Le premier objectif de JSF, est de procurer un environnement de développement permettant de construire une interface de type web, sans devoir toucher au code HTML et JavaScript. Ceci est réalisé par la mise en place d'un mapping entre l'HTML et les objets concernés. JSF est donc basé sur la notion de composants, comparable à celle de Swing, où l'état de ces composants est sauvegardé puis restauré au retour de la requête[19] .

Le principal atout de JSF est qu'il tient compte des d' différentes expériences acquises non seulement avec des technologies de type standard comme les Servlets, les JSP,... mais aussi avec les technologies de type Framework comme Struts. Dès lors, JSF propose un Framework qui puisse être mis en œuvre par des outils facilitant le développement d'applications web. Rien n'empêche que JSF puisse être utilisé par codage à la volée, néanmoins, l'utilisation d'un outil approprié est vivement recommandé afin d'exploiter rapidement les possibilités de JSF[19].

L'implémentation de JSF nécessite l'implémentation de deux fichiers qui sont les suivants[19] :

- Face-config.xml :

Le Face-config.xml est le fichier de configuration, qui permet de gérer les beans, et les règles de navigation entre les pages.

- Web.xml :

Le web.xml permet de définir l'emplacement des pages JSP, ainsi que certaines contraintes de sécurité concernant l'accès à des pages. Il établit également l'endroit où se situe le Face Servlet, Face Servlet est la classe motrice de chaque application JSF.

Chaque application JSF à sa propre Faces Servlet, qui gère toutes les informations relatives à la requête courante.



JAX-RS (Java API for RESTful Web Services) : JAX-RS est l'API Java API des services web RESTful. Il fournit un support pour construire

des services web avec REST. Les services web RESTful sont conçus pour exposer les API sur le web. Ils sont destinés à améliorer la performance, l'adaptabilité, et la flexibilité par rapport aux services web traditionnels, tout en permettant aux clients d'accéder à des données et à des ressources par des URL prédictibles.



L'ANEM compte parmi ses fournisseurs « Microsoft Algérie » qui lui assure l'acquisition et le renouvellement des licences des différents produits Microsoft. Le SGBD relationnel de Microsoft : SQL Server a été conçu pour prendre en charge le traitement des transactions, d'entreposage de données, de veille stratégique et d'analyse dans les environnements informatiques d'entreprise. Il repose sur le langage SQL mais possède sa propre implémentation, T-SQL (Transact-SQL), qui ajoute un ensemble d'extensions de programmation au langage SQL standard.

Le composant principal de Microsoft SQL Server est le moteur de base de données, SQL Server, qui contrôle le stockage, le traitement et la sécurité des données. Il inclut un moteur relationnel qui traite les commandes et les requêtes et un moteur de stockage qui gère les fichiers de base de données, les tables, les pages, les index, les mémoires tampons de données et les transactions. Les procédures stockées, les déclencheurs, les vues et d'autres objets de base de données sont également créés et exécutés par le moteur de base de données [Rouse M., 2017].



Primefaces Le framework utilisé est JSF (« Java Server Faces »). JSF dispose de nombreux composants variés favorisant la réalisation d'applications Web avec une interface efficace et agréable pour les utilisateurs. En revanche, le JSF est très lourd.

Pourquoi Primefaces ? - Associé à la bibliothèque de composants graphiques PrimeFaces, JSF gagne en productivité et en efficacité. -Les composants Primefaces proposent des fonctionnalités prêtes à l'emploi, nombreux, divers et de qualité. - Il y a beaucoup de Widgets pour la gestion. - Il y a une bonne documentation.

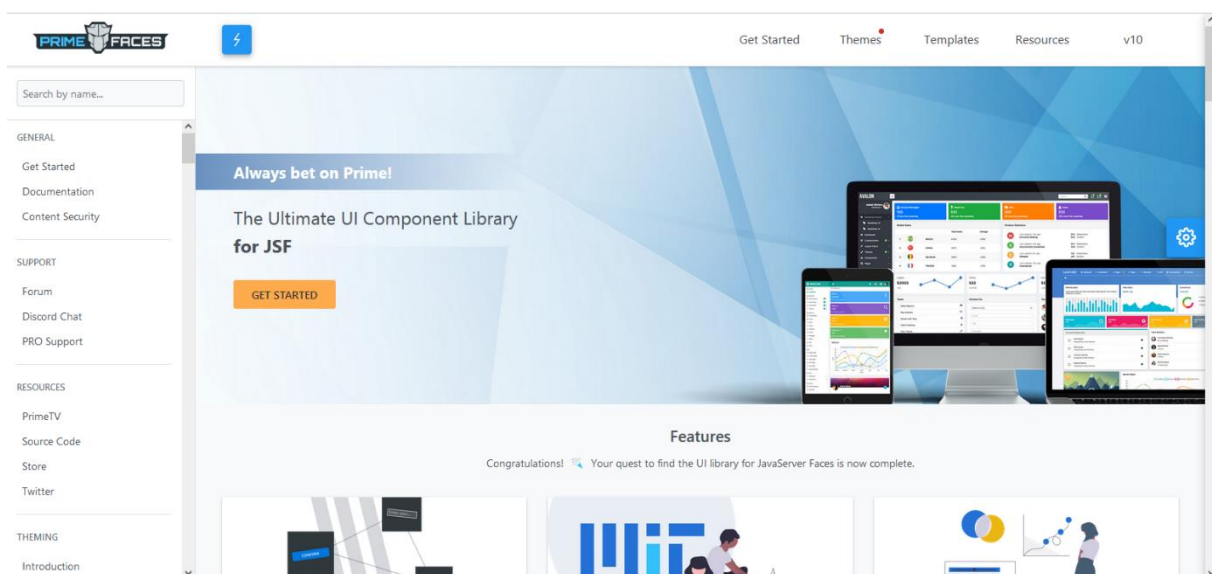


Figure 22: Interface Primefaces



JAVA : Est un langage de programmation informatique orienté objet, La parti central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou Linux, avec peu ou pas du tout de modifications. Pour cela, diverses plateformes associées visent à garantir la portabilité des applications développées en Java.



XHTML (Extensible Hypertext Markup Langage) : est un langage de balisage servant à écrire des pages pour le World Wide Web. Conçu à l'origine comme le successeur de HTML, XHTML se fonde sur la syntaxe définie par SGML sur laquelle repose HTML. Il s'agissait en effet à l'époque de tirer parti des bénéfices techniques attendus de la simplification offerte par XML [25].



Les CSS, Cascading Style Sheets (feuilles de styles en cascade) : servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web. Les feuilles de styles ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle.



JavaScript JavaScript (initialement appelé Live script) a été développé par Netscape puis a été repris par la firme Sun. Les scripts sont des instructions (des lignes de code) interprétées par les navigateurs Netscape et Internet explorer. Le JavaScript n'a cessé d'évaluer avec les versions des navigateurs. JavaScript est un langage de ùScript qui s'incorpore au langage HTML de présentation des pages web. Les scripts vont permettre de rendre une page web plus dynamique.



C'est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. Elle permet la manipulation du DOM (Document Objet Model), des évènements, des effets visuels, et accroît la dynamicit  des pages web. Il permet surtout d' crire moins de code mais adaptable   chaque type de navigateur; ce que ne permet pas JavaScript brut.



Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.



AJAX est l'acronyme d'Asynchronous JavaScript and XML, ce qui, transcrit en français, signifie « JavaScript et XML asynchrones ». C'est un ensemble de technologies destinées à réaliser de rapides mises à jour du contenu d'une page Web, sans qu'elles nécessitent le moindre rechargement visible par l'utilisateur de la page Web

3. Description de l'application (présentation des différentes interfaces de l'application)

3.1. La page Authentification :

Cette page est nécessaire pour l'authentification des différents acteurs qui vont utiliser cette application, ce qui va leur permettre d'accéder à leur espace personnel et pour cela il suffit de cliquer sur le lien <https://integrationdaip.anem.dz/> ensuite remplir le formulaire (Login, Mot de passe) et valider la saisie.

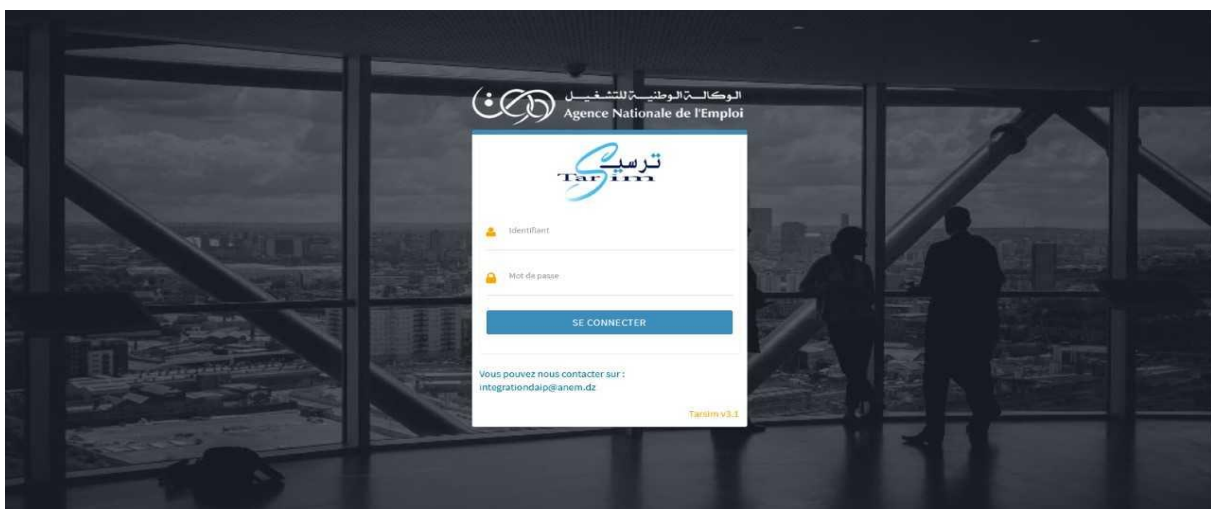


Figure 23: Interface de connexion

3.2. Interface Accueil (Employeur):

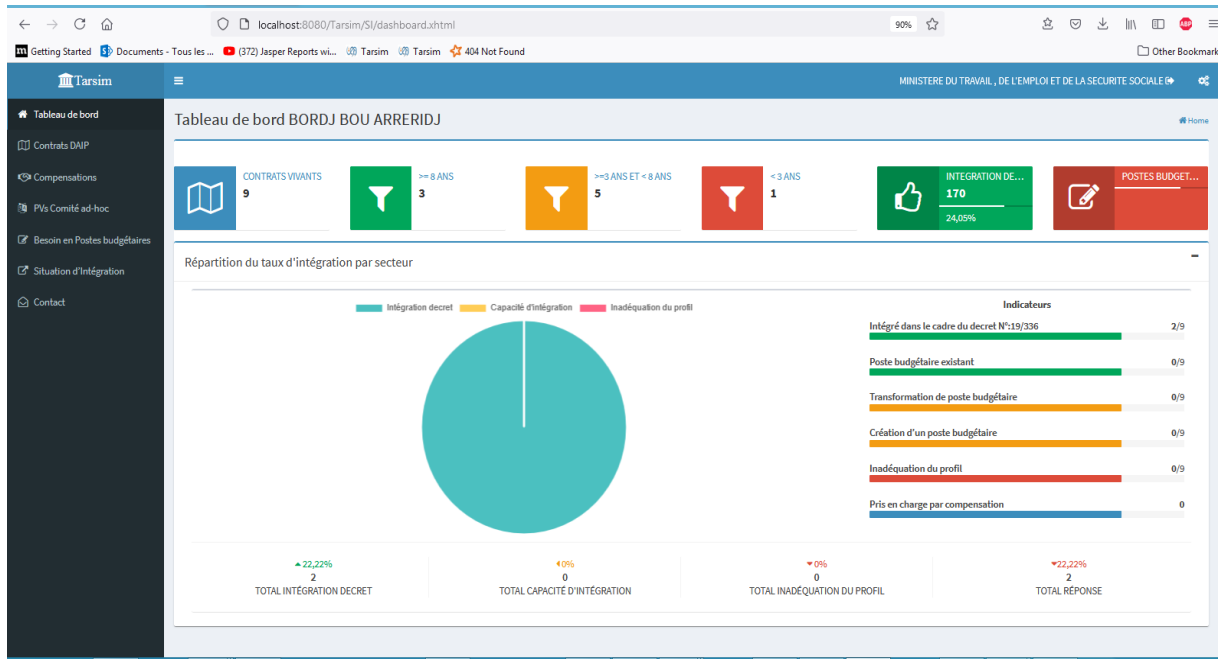


Figure 24:Interface Accueil (Employeur)

3.3. Interface Accueil (Administrateur):

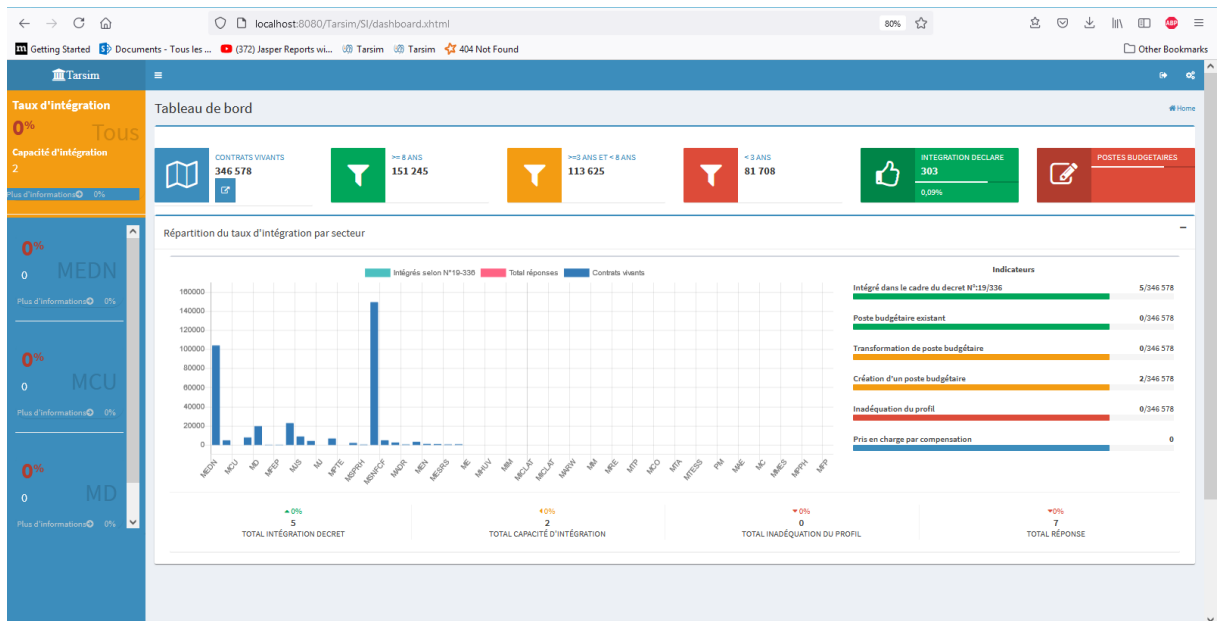


Figure 25:Interface Accueil (Administrateur)

3.4. Interface gestion des contrats DAIP:

The screenshot displays the 'Liste des contrats DAIP' interface. At the top, there is a filter for 'Ancienneté d'insertion : >= 8 ans'. Below this, a table lists three contracts. The table has the following columns: Wilaya, Nom, Prénom, Nom/Pr, Prénom/Pr, D.Naissance, Diplôme, Spécialité, Employeur, Employeur de l'intégration, Ancienneté, Réponse, and Grade FP. The data rows are as follows:

Wilaya	Nom	Prénom	Nom/Pr	Prénom/Pr	D.Naissance	Diplôme	Spécialité	Employeur	Employeur de l'intégration	Ancienneté	Réponse	Grade FP
BORDJ BOU ARR	ADJIL	HANANE	عميل	حان	19/12/1986	LICENCE	SCIENCES JURID	DIRECTION DE L'	DIRECTION DE L'EMPLOI DE LA WILAYA	8	Intégré dans le cadre	Administrateur territorial
BORDJ BOU ARR	AMAROUCHE	ELHAOUES	عماروش	الحواس	13/08/1983	TECHNICIEN	INFORMATIQUE	DIRECTION DE L'	DIRECTION INSPECTION DE TRAVAIL	8		
BORDJ BOU ARR	ABBAS	HAMZA	عاشق	حمزة	20/07/1984	TS	INFORMATIQUE	DIRECTION INSP	DIRECTION INSPECTION DE TRAVAIL	10		

Figure 26: Interface gestion des contrats DAIP

3.5. Interface gestion des PV Comité ad-hoc:

The screenshot displays the 'Procès verbal de comité ad-hoc' interface. It features a form with the following fields: EtatPV (dropdown), CodePV, Ministeres (dropdown), Wilaya (dropdown), Grade Fp, Employeur, and DatePV. Below the form, there is a table with one row of data:

EtatPV	CodePV	Ministeres	Wilaya	Grade Fp	Employeur	DatePV
Select One...	25340004000001	MINISTÈRE DU TRAVAIL, DE L'EMPLOI ET DE LA SECURITE SC	BORDJ BOU ARRERIDJ	Administrateur territorial	DIRECTION DE L'EMPLOI DE LA WILAYA	03/01/2021

Figure 27: Interface gestion des PV Comité ad-hoc

3.6. L'ajout d'un PV:

Créer Nouveau PV

Tutelle
 MINISTERE DU TRAVAIL , DE L'EMPLOI ET DE LA SECURITE SOCIALE

Grade/الرتبة
 Select One... Type de poste/المنصب/تويع المنصب
 Grade

Employeur/ادارة الإدماج
 Select One...

+ ↻ Enregistrer

0 Contrats

Nom	Prénom	D.Naissance	Diplôme	Spécialité	Ancienneté	Réponse
		filtre par année	Sélectionner			Sélectionner

Aucun contrats trouver

Figure 28:L'ajout d'un PV

3.7. L'impression d'un PV Brouillon:

الجمهورية الجزائرية الديمقراطية الشعبية
 Procès verbale de comité ad-hoc (brouillon)

بتاريخ: 04/01/2021 رقم: 25340004000001

القطاع الوزاري: MINISTERE DU TRAVAIL , DE L'EMPLOI ET DE LA SECURITE SOCIALE
 الرتبة: Administrateur territorial
 الولاية: BORDJ BOU ARRERIDJ
 إدارة الإدماج: DIRECTION DE L'EMPLOI DE LA WILAYA

التعداد المدمجين: 2
 عدد المناصب المالية المخصصة: 0
 عدد المناصب المالية محل طلب التحويل: 0
 عدد المناصب المالية الواجب إنشائها: 0

القائمة الاسمية للمدمجين في الرتبة وفق أحكام المرسوم التنفيذي رقم 19-336 المؤرخ في 08 ديسمبر 2019، المتضمن إدماج المستفيدين من جهازي المساعدة على الإدماج المهني والإدماج الاجتماعي للشباب حاملي الشهادات

الرقم التسلسلي	اللقب	الاسم	تاريخ الميلاد	الشهادة	الخبرة المهنية المكتسبة من قبل المترشح في جهاز	ملاحظة
1	ADJIL	HANANE	20/12/1986	LIGENCE SCIENCES JURIDIQUES	8 سنوات	

إمضاء أعضاء اللجنة المختصة

Date : 13/07/2021

Figure 29:L'impression d'un PV Brouillon

3.8. L'impression d'un PV Final:

الجمهورية الجزائرية الديمقراطية الشعبية
Procès verbale de comité ad-hoc

رقم : 17340004000005

بتاريخ : 28/12/2020

القطاع الوزاري : **MINISTÈRE DE L'INTERIEUR, DES COLLECTIVITES LOCALES ET DE**
الولاية : **BORDJ BOU ARRERIDJ**
ادارة الادماج : **APC DE KSOUR**

الرتبة : **Administrateur territorial**
التعداد المدمجين : **1**
عدد المناصب المالية المخصصة : **0**
عدد المناصب المالية محل طلب التحويل : **0**
عدد المناصب المالية الواجب إنشائها : **0**

القائمة الاسمية للمدمجين في الرتبة وفق أحكام المرسوم التنفيذي رقم 19-336 المؤرخ في 08 ديسمبر 2019، المتضمن إدماج المستفيدين من جهازي المساعدة على الإدماج المهني والإدماج الاجتماعي للشباب حاملي الشهادات

الرقم التسلسلي	اللقب	الاسم	تاريخ الميلاد	الشهادة	الخبرة المهنية المكتسبة من قبل المترشح في جهاز	ملاحظة
1	ALLOU	GHANIA	01/01/1978	DEUA GENIE DES PROCEDI	8 سنوات	

إمضاء أعضاء اللجنة المختصة

Date : 28/12/2020

Figure 30:L'impression d'un PV Final

3.9. Interface gestion des Besoin en Postes budgétaires:

MINISTÈRE DU TRAVAIL, DE L'EMPLOI ET DE LA SECURITE SOCIALE

Besoin en Postes budgétaires

Créer Supprimer

1 10

Wilaya II | Grade Fp II | Nbr Poste

Sélectionner

TLEMEN | Comptable principal de l'administration territoriale | 14

1 10

Powered by Anem - © 2021

Figure 31:Interface gestion des Besoin en Postes budgétaires

3.10. L'ajout d'un Besoin:

Figure 32:L'ajout d'un Besoin

3.11. Interface gestion des Utilisateurs:

Id	Login	EtatCompte
91	acc-tiziouzu	
92	acc-boumerdes	
93	acc-bouira	

Figure 33:Interface gestion des Utilisateurs

4. Déploiement

Le déploiement de la solution est l'étape qui permet de mettre en production notre produit final. La réussite de cette étape implique d'autres facteurs liés principalement à l'infrastructure physique de déploiement. Dans notre cas, l'infrastructure physique de l'ANEM est prête car elle est utilisée par l'ensemble des applicatifs de l'agence (voir Annexe B). Pour réussir notre déploiement, nous avons établi un plan de déploiement progressif permettant de mettre en production nos services au fur et à mesure. Ceci nous a permis de découvrir tôt les dysfonctionnements et les mal-configurations éventuels. La **figure 21** illustre les différents nœuds utilisés (ressources), la répartition des composants sur l'ensemble des nœuds, ainsi que les connexions entre ces nœuds.

- Les deux bases de données utilisées par le système sont déployées dans le cluster Production sous le SGBD SQL Server ;
- L'ensemble des applicatifs destinés à être accessibles depuis Internet est déployé dans le cluster DMZ, incluant les différentes API. Ces derniers sont hébergés et exécutés sur un serveur GlassFish. C'est un serveur Web flexible, sécurisé et administrable pour tout hébergement sur le Web ;
- L'application web cliente 'Tarsim' est déployée sur le serveur GlassFish mais exécutée sur un navigateur Web.

Toute communication avec l'ensemble des applicatifs qui passe par Internet est effectuée via le protocole HTTPS. Ce dernier est une variante du protocole HTTP qui ajoute une couche de sécurité aux connexions entre le client et le serveur. Il crypte les données transmises sur le réseau en utilisant le protocole SSL (Secure Sockets Layer) ou TLS (Transport Layer Security).

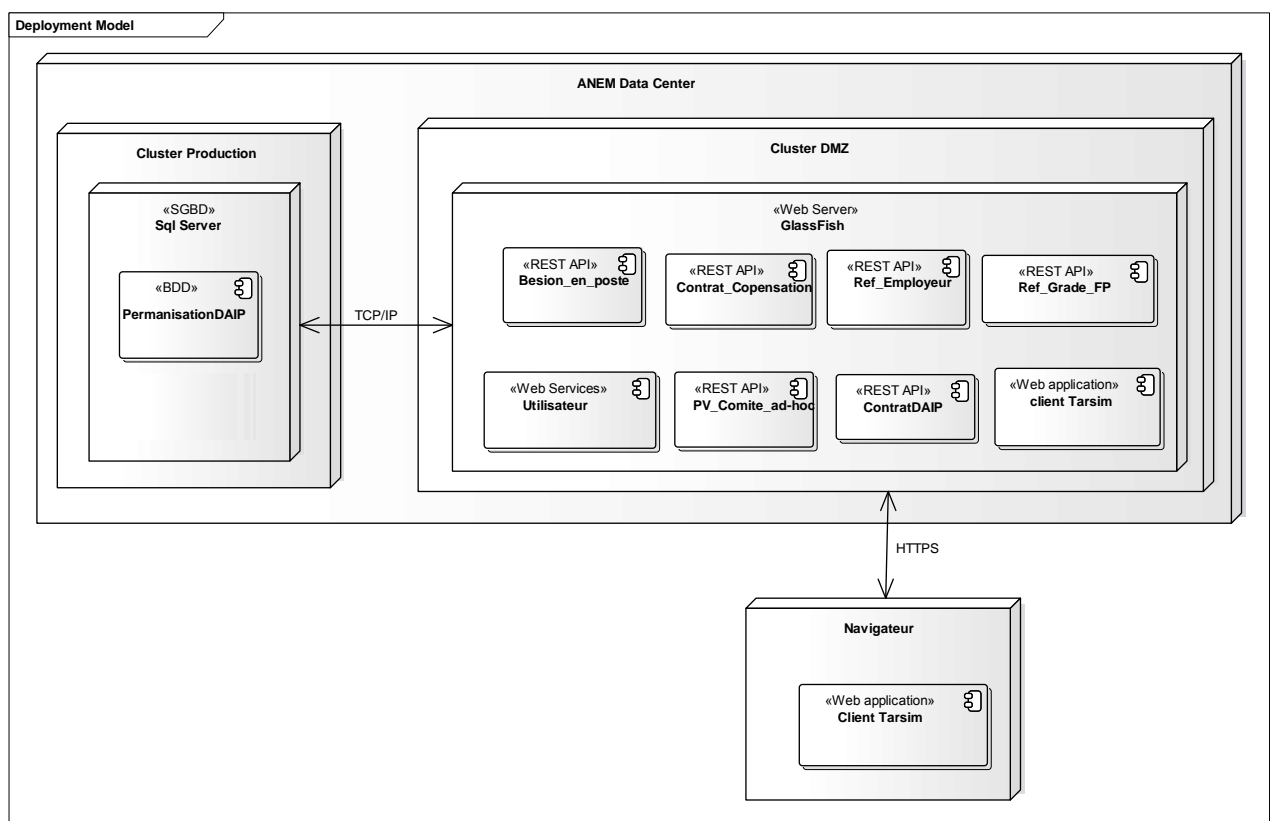


Figure 34: Diagramme de déploiement

5. Conclusion

Dans ce chapitre nous avons, au premier lieu, présenté les différents outils et langages que nous avons utilisé pour implémenter notre application. Par la suite, nous avons présenté quelques interfaces de notre application.

Conclusion générale

Le thème retenu pour notre mémoire : Conception et réalisation d'une application basée sur une architecture SOA avec java EE pour la gestion d'intégration des bénéficiaire DAIP de l'ANEM.

L'objectif majeur de notre projet consiste à développer un système qui permet aux services de l'ANEM de résoudre la problématique d'intégration des bénéficiaire DAIP dans les établissements publics. Ce qui a permis de mettre en place le système « TARSSIM».

Au cours de notre travail, nous avons présenté les différentes étapes de la conception et de réalisation de notre application. Grâce au langage UML, nous avons pu identifier puis modéliser la conception du système.

Ce projet a fait l'objet d'une expérience très enrichissante, car elle nous a permis de nous familiariser avec de nouvelles notions et technologies d'une part, mais aussi elle nous a permis d'améliorer nos connaissances et nos compétences dans le domaine de la programmation en java EE. D'autre part, nous avons pu grâce à ce projet faire beaucoup de recherches par la lecture des articles et mémoires publiés dans la thématique notamment sur l'interopérabilité des systèmes d'information. Ceci nous motive aujourd'hui à entrevoir une carrière de chercheur surtout dans ce domaine afin d'en apporter notre contribution pour l'amélioration de ce concept.

Références bibliographiques

Articles :

- [**Belqasmi F., 2012**] Belqasmi, F., Singh, J., Melhem, S. Y. B., & Glitho, R. H. (2012). Soap-based vs. restful web services: A case study for multimedia conferencing. *IEEE Internet computing*, 16(4), 54-63.
- [**Garlan, D., 1995**] Garlan, D. (1995, February). What is style. In *Proceedings of Dagstuhl Workshop on Software Architecture* (Vol. 100). Saarbruecken, Germany.
- [**Kruchten, P. et al, 2006**] Kruchten, P., Obbink, H., & Stafford, J. (2006). The past, present, and future for software architecture. *IEEE software*, 23(2), 22-30.
- [**Kumari S., Rath S. K., 2015**] Kumari, S., & Rath, S. K. (2015). Performance comparison of soap and rest based web services for enterprise application integration. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1656-1660). IEEE.
- [**Medvidovic N., Taylor R.N., 2010**] Medvidovic, N., & Taylor, R. N. (2010, May). Software architecture: foundations, theory, and practice. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2* (pp. 471-472). ACM.
- [**Mumbaikar S., 2013**] Mumbaikar, S., & Padiya, P. (2013). Web services based on soap and rest principles. *International Journal of Scientific and Research Publications*, 3(5), 1-4.
- [**Papazoglou M.P., 2003**] Papazoglou, Mike P. "Service-oriented computing: Concepts, characteristics and directions." *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*. IEEE, 2003.
- [**Potti P., 2012**] Potti, P. K., Ahuja, S., Umapathy, K., & Prodanoff, Z. (2012). *Comparing performance of web service interaction styles: Soap vs. rest*. Paper presented at the Proceedings of the Conference on Information Systems Applied Research ISSN.
- [**Wagh K., 2012**] Wagh, K., & Thool, R. (2012). A comparative study of soap vs rest web services provisioning techniques for mobile host. *Journal of Information Engineering and Applications*, 2(5), 12-16.
- [**Zhang Y., Zhou J., 2009**] Zhang, Y., & Zhou, J. (2009). Research and Implementation of EAI Based on SOA. Paper presented at the *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*. Disponible sur : <https://ieeexplore.ieee.org/abstract/document/5362586>.
- [12] document : introduction_a_la_plateforme J2EE, fichier PDF qui se trouve sur www.mccours.net/cours/pdf/onfo/introduction_a_la_plateforme_J2EE.pdf
- [15] document : J2EE vs Net Centre d'enseignements de grenoble 2008 /2009 qui se trouve sur membres-liglab.imag.fr/plumejeaud/NFE107.../J2EE%20vs%20NET.pdf
- [25]: www.la-grange.net/w3c/xhtml
- [19]: <https://www.oracle.com/technetwork/.../javaserverfaces>

Thèses :

[11] document : pfe de J2EE, fichier PDF, qui se trouver sur fac-sciences-unv-batna.dz/cs/enseignants/guezouli.../projet_03.pdf

[Lemrabet Y., 2012] Lemrabet Y. Proposition d'une méthode de spécification d'une architecture orientée services dirigée par le métier dans le cadre d'une collaboration inter-organisationnelle. Diss. Ecole Centrale de Lille, 2012.

Livres :

[Isabelle V, Louis M,2001] JAVA Autoformation « Isabelle Valembois, Louis Millecam » Ellipses 2001

[Bonnet P., 2005] Bonnet, P. (2005). Cadre de Référence Architecture SOA-Meilleures Pratiques. Orchestra Networks (February 2005).

[Erl T., 2010] Erl, T. (2010). SOA principles of Service Design. Prentice Hall/PearsonPTR.

[Josuttis N.M., 2007] Josuttis, N. M. (2007). *SOA in practice : the art of distributed system design* (pp. 47-59, 190-207). " O'Reilly Media, Inc.".

[Juric, M. B., 2007] Juric, M. B. (2007). SOA Approach to Integration: XML, Web services, ESB, and BPEL in real-world SOA projects. Packt Publishing Ltd.

[Rosen M. et al., 2012] Rosen, M., B. Lublinsky, et al. (2012). Applied SOA: Service-Oriented Architecture and Design Strategies, Wiley.

Références webographiques

- [ANEM, 2018a] ANEM (2018). « F.A.Q ». Disponible sur : <http://www.anem.dz/fr/contact/faq>.
- [ANEM, 2018b] ANEM (2018). « Historique ». Disponible sur : <http://www.anem.dz/fr/pages/historique.html>.
- [BPMN, 2019] BPMN (2019). « BPMN poster ». Disponible sur : <https://www.trisotech.com/infographics/bpmn-poster>.
- [IETF, 2005] IETF (Jan 2005) «RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax ». Disponible sur : <https://tools.ietf.org/html/rfc3986#section-1.2.2>.
- [JSON, 2018] JSON (2018) « Présentation de JSON ». Disponible sur : <https://www.json.org/json-fr.html>.
- [Mostefai M.A., 2012] Mostefai M.A. (2012). « Génie Logiciels : Introduction aux architectures ». Disponible sur : <https://fr.slideshare.net/mostefaiamine/gnie-logiciels-introduction-aux-architectures>.
- [Oracle, 2010] Oracle (2010) « Message-oriented Middleware (MOM) ». Disponible sur : <https://docs.oracle.com/cd/E19340-01/820-6424/aeraq/index.html>.
- [Rouse M., 2017] Rouse M. (2017). « Microsoft SQL Server ». Disponible sur : <https://searchsqlserver.techtarget.com/definition/SQL-Server>.
- [TenStep, 2019] LifecycleStep (2019) « Méthodologie pour la gestion du cycle de vie d'un projet de développement informatique ». Disponible sur : http://www.tenstep.fr/TSLifecycle/03_Licence/421.1_Architecture_de_developpement_Applicative.htm.
- [W3C, 2016] W3C (2016) « Extensible Markup Language (XML) ». Disponible sur : <https://www.w3.org/XML/>.
- [W3Schools, 2019] W3Schools (2019) « JSON vs XML ». Disponible sur : https://www.w3schools.com/js/js_json_xml.asp.
- [14] Site web: www.wikipédia.fr

Annexes

Annexe A : Dispositif d'Aide à l'Insertion professionnelle DAIP

Ce dispositif, institué par le décret exécutif n°08-126 du 19 avril 2008, modifié et complété, relatif au dispositif d'aide à l'insertion professionnelle, est géré par l'ANEM en relation avec les Directions de l'Emploi de Wilaya (DEW).

Il vise à :

- Favoriser l'insertion professionnelle des jeunes primo-demandeurs d'emploi;
- Encourager toutes autres formes d'actions et de mesures tendant à promouvoir l'emploi des jeunes, à travers notamment des programmes de formation-emploi et de recrutement.

Le dispositif est destiné à trois (3) catégories de primo-demandeurs d'emploi :

- 1ère catégorie : Les jeunes diplômés de l'enseignement supérieur et les techniciens supérieurs issus des établissements nationaux de formation professionnelle ;
- 2ème catégorie : Les jeunes sortant de l'enseignement secondaire de l'éducation nationale, des centres de formation professionnelle, ou ayant suivi un stage d'apprentissage ;
- 3ème catégorie : Les jeunes sans formation ni qualification.

Les contrats d'insertion :

Un contrat est établi pour chaque catégorie à savoir :

- Contrat d'Insertion des Diplômés «CID» pour les diplômés de l'enseignement supérieur et les techniciens supérieurs ;
- Contrat d'Insertion Professionnelle «CIP» pour les sortant de l'enseignement secondaire de l'éducation nationale et des centres de formation professionnelle ;
- Contrat Formation-Insertion «CFI» pour les jeunes sans formation ni qualification ;

Les contrats d'insertion sont établis entre :

1. Les services relevant de l'Administration chargée de l'Emploi (la Direction de l'Emploi de Wilaya DEW) ;
2. L'Employeur ou l'organisme formateur ;
3. Les bénéficiaires.

– Quels sont les organismes employeurs ciblés ?

- Entreprises économiques publiques et privées ;
- Institutions et administrations publiques ;
- Les collectivités locales et les différents secteurs d'activité à travers les chantiers de travaux divers d'utilité publique ;
- les Maîtres artisans.

- Quelle est la durée des contrats d'insertion ?
- Trois (03) années renouvelable dans le secteur des institutions et administrations publiques ;
- Une (01) année renouvelable dans le secteur économique ;
- Une (01) année non renouvelable pour les formations auprès des maîtres artisans ;
- Six (06) mois renouvelables une fois à la demande de l'employeur pour les chantiers d'utilité publique ;
- Une (01) année non renouvelable dans les entreprises de production.

- Quel est le nombre de bénéficiaires pouvant être affecté par employeur ?

Le nombre ne doit pas dépasser :

- 15 % de l'effectif en activité pour les wilayas du Nord et des hauts Plateaux ;
- 25 % de l'effectif en activité dans les wilayas du sud.

Pour les Micro- entreprises créées dans le cadre des dispositifs de soutien à la création d'activité, deux (02) primo demandeurs d'emploi en CID, CIP ou CFI peuvent leur être affectés durant la phase de démarrage.

NB. 1 : Les employeurs qui n'auront pas procédé au recrutement d'au moins 25 % des bénéficiaires de contrats d'insertion qui leur ont été affectés, ne peuvent prétendre à de nouvelles affectations de jeunes primo-demandeurs d'emploi dans le cadre du dispositif.

NB. 2 : Les employeurs ayant réalisé un taux de recrutement égal ou supérieur à 25% peuvent bénéficier d'une affectation supplémentaire dans la limite de 30% de leur effectif total.

- Quelles sont les conditions d'éligibilité des demandeurs d'emploi ?

- Etre primo demandeur d'emploi ;
- De nationalité Algérienne ;
- Agé de 18 à 35 ans ;
- Titres et diplômes, justifications de niveau d'instruction, de qualification et d'acquis professionnels ;
- Inscrit comme demandeur d'emploi auprès des agences de l'ANEM du lieu de résidence ;
- A titre dérogatoire pour les jeunes âgés de 16 ans, à condition d'accepter de suivre une formation dans les filières déficitaires sur le marché de l'emploi.

- Rémunérations et bourses :

1. Les bénéficiaires des CID perçoivent une rémunération mensuelle nette fixée à :

- 15.000 DA pour les diplômés de l'enseignement supérieur ;

- 10.000 DA net pour les techniciens supérieurs.
2. Les bénéficiaires des CIP perçoivent une rémunération mensuelle nette fixée à 8.000 DA;
3. Les bénéficiaires des CFI perçoivent une rémunération mensuelle dont le montant est fixé comme suit :
- 12.000 DA pour les jeunes insérés dans le cadre des chantiers divers initiés par les secteurs ou les communes ;
 - Salaire de poste de travail occupé comprenant une contribution de l'Etat dont le montant est fixé à 6.000 DA lorsqu'ils sont placés dans les entreprises économiques devant réaliser des projets d'utilité publique. Le différentiel avec le salaire de poste est versé par l'employeur ;
 - Rémunération mensuelle d'un montant de 6.000 DA versée intégralement sur le budget de l'Etat pendant une (1) année lorsqu'ils sont placés dans des entreprises de production ;
 - Bourse mensuelle de 4.000 DA pour les jeunes placés auprès de maîtres artisans.

Accompagnement par la formation pour les jeunes primo demandeurs d'emploi (Contrat Formation –Emploi):

Les jeunes insérés dans le cadre du dispositif peuvent bénéficier de formation complémentaire, recyclage ou perfectionnement en vue de leur adaptation au poste de travail et de l'amélioration de leurs qualifications.

Les jeunes insérés dans le cadre des Contrats d'Insertion des Diplômés (CID) ou des Contrats d'Insertion Professionnelle (CIP) ou des Contrats Formation/Insertion (CFI), à l'exclusion de ceux placés auprès des maîtres artisans, peuvent bénéficier de contrats Formation-Emploi (CFE) financés à hauteur de 60% par le dispositif pendant une période maximale de six (6) mois dans le cas où l'employeur s'engage à recruter le bénéficiaire pour une durée minimale d'une (1) année à l'issue de la formation.

Le contrat de formation-Emploi (CFE) est établi entre l'ANEM, l'employeur et le bénéficiaire après accord du directeur de l'emploi de wilaya.

Une prime d'encouragement à la recherche de formation fixée à 3.000 DA par mois, est accordée aux bénéficiaires du dispositif qui auront réussi à s'inscrire dans un stage de formation d'une durée maximale de six (6) mois dans des filières ou spécialités en déficit sur le marché de l'emploi.

Des formations en partenariat avec le secteur de la formation professionnelle sont organisées pour les jeunes âgés de 16 à 20 ans dans les métiers déficitaires dans le marché du travail, pour une durée de six (6) mois, durant laquelle le jeune perçoit une bourse de 3.000 DA.

Le contrat de travail aide (CTA) :

– Qu'est-ce que le CTA ?

Le Contrat de Travail Aidé (CTA) est un contrat de travail au sens de la loi n°90-11 du 21 avril 1990 relative aux relations de travail. Il entre dans le cadre de l'encouragement au recrutement durable des jeunes par les entreprises publiques et privées.

Les jeunes primo demandeurs d'emploi recrutés dans ce cadre, sont rémunérés conformément à la grille des salaires de l'organisme employeur.

L'Etat contribue pendant une durée de (03) années non renouvelable, au salaire de poste, à travers une subvention fixée respectivement à :

- 12.000 DA par mois pour les universitaires ;
- 10.000 DA par mois pour les techniciens supérieur ;
- 8.000 DA par mois pour les sortants de la formation professionnelle et du système éducatif ;

Concernant les CFI entreprise, la contribution de l'Etat au salaire de poste , est fixée à 6.000 DA par mois pendant une (1) année non renouvelable.

Obligations des jeunes insérés :

- Toute rupture du contrat d'insertion par les jeunes insérés, sans motif dûment justifié, entraîne la perte de leur éligibilité au dispositif.
- Les jeunes s'engagent à achever la période d'insertion fixée dans le contrat. Dans le cas où la rupture du contrat est due à des motifs dûment justifiés, les jeunes peuvent prétendre au bénéfice d'un nouveau contrat d'insertion.
- Les jeunes insérés dans le cadre du dispositif sont tenus de respecter le règlement intérieur de l'organisme employeur.
- Dans le cas où il est proposé au jeune diplômé, durant la période d'insertion auprès des institutions et administrations publiques, un contrat de travail aidé en entreprise conforme à ses qualifications, il lui est fait obligation d'accepter cette proposition sous peine de perdre le droit au maintien en contrat d'insertion des diplômés ou en contrat d'insertion professionnelle.

Obligations de l'employeur :

- Toute rupture du contrat d'insertion par l'employeur, sans motif dûment justifié, entraîne la perte de son éligibilité au dispositif.
- Dans le cas où l'employeur désire rompre le contrat d'insertion, il doit aviser par écrit le jeune bénéficiaire et les services compétents de l'Agence Nationale de l'Emploi (ANEM), sept (7) jours au moins avant la rupture du contrat.
- L'employeur doit préciser par écrit les motifs de résiliation du contrat d'insertion.
- La rupture du contrat d'insertion entraîne la suspension du versement de la rémunération ou de la bourse.

MESURES D'ENCOURAGEMENT AU RECRUTEMENT :

Abattement de la part patronale de cotisation à la sécurité sociale pendant trois (3) ans:

- 40% si l'employeur recrute un demandeur d'emploi ayant déjà travaillé ;
- 80% si l'employeur recrute un primo-demandeur d'emploi ;

- 95% pour tout recrutement dans les wilayas des hauts plateaux et du sud.

Les employeurs des secteurs du tourisme, de l'artisanat, de la culture, de l'agriculture, du BTPH ou dans des sociétés de service, peuvent recruter pour une durée de six (6) à douze (12) mois. Dans ce cas la part patronale de cotisation sera de :

- 20% s'il recrute un demandeur d'emploi ayant déjà travaillé ;
 - 18% s'il recrute un primo-demandeur d'emploi ;
 - 16% s'il recrute dans les wilayas des haut plateaux du sud.
- Outre les abattements de la part patronale, l'employeur peut bénéficier pendant trois (3) ans d'une subvention mensuelle à l'emploi pour chaque recrutement lorsque le contrat de travail est conclu pour une durée indéterminée.
 - Exonération de la cotisation globale pour tout employeur qui engage des actions de formation ou de perfectionnement en faveur de ses travailleurs.

La caisse nationale d'assurance-chômage prend en charge la cotisation globale de sécurité sociale pendant une période pouvant aller jusqu'à trois (3) mois.

- Recrutement en CTA :

- Bonification d'une cotisation de la part patronale de 80%, ce taux est porté à 95% pour les wilayas des hauts plateaux et du sud.

Ancrage juridique:

- Loi n°90-11 du 21 avril 1990 relative aux relations de travail ;
- Loi n°06-21 du 20 Dhou El Kaada 1427 correspondant au 11 décembre 2006 relative aux mesures d'encouragement et d'appui à la promotion de l'emploi, notamment ses articles 6, 11 et 12 ;
- Loi n°06-24 du 6 Dhou El Hidja 1427 correspondant au 26 décembre 2006 portant loi de finances pour 2007, notamment son article 59 ;
- Loi n°09-09 du 30 décembre 2009 portant loi de finances pour 2010 ;
- Décret exécutif n°94-187 du 26 Moharrem 1415 correspondant au 6 juillet 1994, modifié et complété, fixant la répartition du taux de cotisation de sécurité sociale ;
- Décret exécutif n°07-386 du 25 Dhou El Kaâda 1428 correspondant au 5 décembre 2007 fixant le niveau et les modalités d'octroi des avantages prévus par la loi n°06-21 du 20 Dhou El Kaada 1427 correspondant au 11 décembre 2006 relative aux mesures d'encouragement et d'appui à la promotion de l'emploi ;
- Décret exécutif n°08-126 du 19 avril 2008 relatif au dispositif d'aide à l'insertion modifié et complété par le décret exécutif n°10-277 du 4 novembre 2010, le décret exécutif n°11-105 du 6 mars 2011 et le décret exécutif n°13-142 du 10 avril 2013.

- Convention MTESS – MFEP (25 février 2009), relative à la formation des primo demandeurs d'emploi âgés de 16-20 ans, dans les métiers déficitaires.

Annexe B : Infrastructure informatique

L'infrastructure physique (réseau et Data Center) de l'ANEM est son support vital de communication. Elle représente la structure de base sur laquelle sont construits les systèmes applicatifs et permet le transfert de l'information en toute sécurité.

La communication entre le site central (DG de l'ANEM) et l'environnement externe est assurée par deux mécanismes :

- Des lignes spécialisées et des lignes ADSL reliant le réseau des AWEM et des ALEM avec le site central, en passant par un réseau RMS (Réseau Multi Services) que l'ANEM loue auprès du fournisseur national Algérie Télécom. Les employés des différentes agences réparties à travers le territoire national ont accès aux applications métier, qui se trouvent au niveau du site central, grâce à des tunnels VPN. L'ANEM est aussi reliée au Ministère de Travail et de la Sécurité Sociale via une ligne spécialisée.
- Les employeurs, les demandeurs d'emploi et les internautes accèdent aux services à distances de l'ANEM via Internet à travers la plateforme web « WassitOnline ». De même, les collaborateurs de l'ANEM ont accès à des services dédiés via un extranet accessible sur Internet.

Au niveau du site central se trouve le Data Center dans lequel sont hébergés tous les applicatifs métier. Ce centre est constitué de plusieurs clusters, nous citons principalement :

- **Le cluster Production** : une gamme de serveurs non accessible de l'extérieur, hébergeant uniquement les systèmes internes (tels que l'intranet 3amali et le système Wassit) destinés à être utilisés par les employés de l'ANEM dans les différentes agences incluant la direction générale ;
- **Le cluster DMZ** : une zone démilitarisé accueillant les données provenant de l'extérieur notamment à travers les services à distance (WassitOnline) et l'extranet ;
- **Le cluster Management** : qui héberge les outils et les applications de gestion du réseau informatique ;
- **Baie de stockage** : conserve les données et les backups des différents systèmes applicatifs ;
- **Active Directory** : un annuaire de gestion de l'identité de l'ANEM : les informations sur les personnes, les applications et les ressources réparties sur le réseau informatique de l'ANEM. Créé par Microsoft, l'Active Directory fournit des services d'authentification et d'identification centralisés pour les réseaux d'ordinateurs sous Windows.

La **figure 53** illustre l'architecture de l'infrastructure physique de l'ANEM.

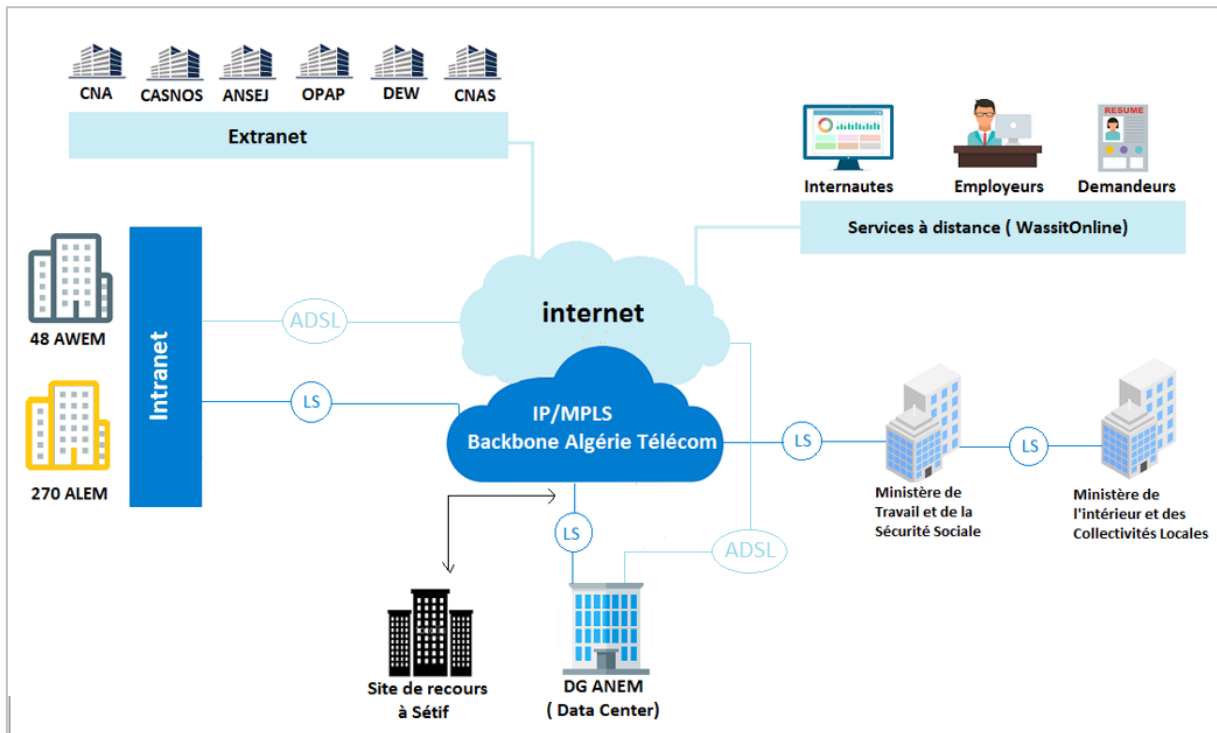


Figure 35: Architecture du réseau de l'ANEM.

Les différentes sous-directions du site central sont organisées en VLAN (Réseau Local Virtuel) reliés entre eux et avec le Data Center à l'aide de switch. Les flux de données entrantes et sortantes passent par un cluster de plusieurs pare-feu qui régulent la communication entre le site central et le monde extérieur (Internet et les lignes spécialisées). Voir la **figure 54**.

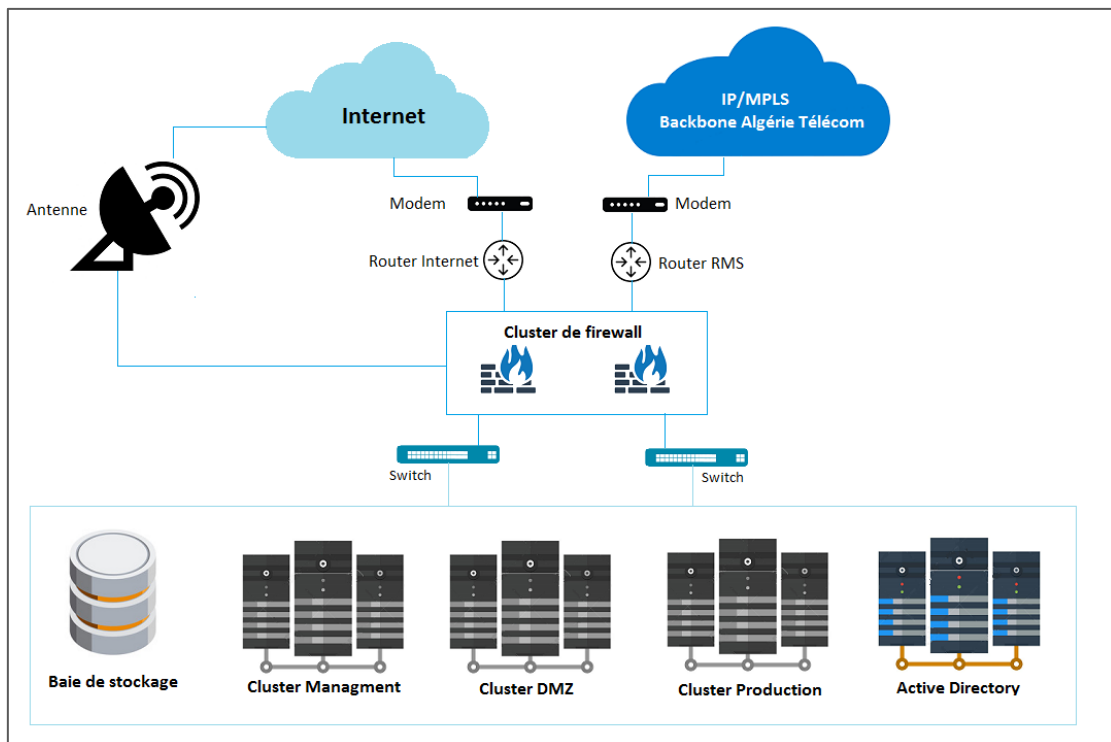


Figure 36: Architecture de l'infrastructure informatique du site central de l'ANEM