REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur

et de la Recherche Scientifique

Université "Yahia FARES" Médéa

Faculté des Sciences

Département de Mathématiques et Informatique

Projet de fin d'étude

Présenté pour l'obtention du diplôme

De Master en Informatique

Spécialité : Ingénierie des Systèmes et Technologie de Web

Thème

Elaboration d'une plateforme de e-gouvernement basée sur la composition dynamique des services web

Proposé et dirigé par :

Mr. TOUBAL El Bahi

Préparé et présenté par :

* *BENYAHIA Amina Aicha*

* BOUIHI Faiza

Promotion 2019

Remerciement

Nous remercions tout d'abord, Allah qui nous a donné la force et le courage afin de parvenir à élaborer ce modeste travail.

Pour tous les bienfaits et les faveurs qu'il nous a attribuées et sans lesquels nous ne serons pas arrivés là où l'on est.

Nous tenons aussi à remercier : Mr. Toubal El Bahi, notre encadreur, pour ses conseils, ses orientations, sa disponibilité et son encouragement qui nous ont permis de réaliser ce travail.

Nous remercions tous les enseignants de l'université Yahia Fares.

Nous tenons à exprimer notre gratitude à toutes les personnes qui nous consacrés à leur temps et de leur énergie afin de faciliter le déroulement de notre projet de fin d'étude.



Dédicace

Je remercie ALLAH

Tout puissant qui m'a donné toute la volonté et la

Patience pour aboutir à la réalisation de ce travail.

Je dédie mon travail à :

Ma très chère mère.

Mon très cher père.

Mes sœurs.

Toute ma famille.

Tous mes amis.

Tous ceux qui me sont chers, …



Faiza



Dédicace

Je dédie ce modeste travail à :

Ma mère et Mon père

Spécialement à Mon mari Mustafa pour son

Encouragement, sa patience et mes enfants la lumière

de ma vie marwa, sara, imad, youcef

Mes frères et leur petite famille,

Tous mes amis,

Tous mes collègues.

Amina

Résumé

Résumé :

L'identité numérique peut être définie comme la représentation numérique d'informations sur une personne, une entité ou une organisation. Pour offrir des services, une institution peut avoir besoin de données provenant d'autres institutions afin de construire la totalité ou une partie d'une identité numérique nécessaire. Cependant, même si ces données sont disponibles dans chaque institution, elles ne sont pas exploitables par les autres institutions et aucune collaboration systématique ne peut être traitée pour accorder l'accès à ces données en l'absence d'une plate-forme offrant des services à cette fin. Dans ce mémoire, nous proposons une plate-forme inter-institutions basée sur des services Web permettant à une institution d'apporter des données d'autres institutions afin de construire l'identité numérique nécessaire pour offrir ses services. La plateforme s'appuie sur un algorithme permettant d'effectuer une composition dynamique des services Web.

Mot clé : service web, identité numérique, profil, e-gouvernement, composition de services web

الملخص:

يمكن تعريف الهوية الرقمية على أنها التمثيل الرقمي للمعلومات المتعلقة بشخص ما أو كيان أو منظمة.

قد تحتاج المؤسسة لتقديم خدماتها إلى بياناتٍ من مؤسسات أخرى لتبنيَ عليها بعض أو كل الهوية الرقمية.

و مع ذلك, حتى و ان كانت هذه البيانات متوفرة في كل مؤسسة فإنها غير قابلة للاستغلال من طرف المؤسسات الاخرى و لا يمكن لأي تعاون منهجي ان يقوم بمعالجتها ليتيح الوصول الى هاته البيانات في ظل غياب منصة تقدم خدمات لهذا الغرض.

تقترح هذه المذكرة منصة مشتركة بين المؤسسات تعتمد على خدمات شبكة الانترنت تتيح لمؤسسة ما جلب بيانات مؤسسات اخرى من اجل بناء هويتها الرقمية الازمة ليتسنى لها تقديم خدماتها.

تعتمد المنصة على خوارزميات نقترحها لتقوم بتركيب ديناميكي لخدمات الويب المقدمة.

الكلمات المفتاحية : خدمة الويب ، الهوية الرقمية ، الملف التعريفي ، الحكومة الإلكترونية ، تركيب خدمة الويب.

Résumé

Abstract:

Digital identity can be defined as the digital representation of information about a person, entity or organization. To offer its services, an institution may need data from other institutions to construct a full or a part of a needed digital identity. However, even if, those data are available in each institution, the other institutions can't use them and no systematic collaboration can be handled to grant access to such data in the absence of a platform offering services for this purpose. We propose an inter-institutions web services-based platform allowing to an institution bringing data from others institutions in order to construct the needed digital identity for offering its services. The platform a new algorithm which we propose to perform a dynamic composition of the offered web services.

Keyword: web service, Digital Identity, profile, e-government, web service composition.

Sommaire

Sommaire:

Résumé i

Liste des figures v

Liste des tableaux vii

Liste des acronymes viii

Introduction générale 1

Chapitre I : gouvernement électronique « E-gouvernement »

I.1. Introduction 3

I.2. Définition et compréhension de l'e-gouvernement 3

I.2.1. Définition de gouvernement ---------- 3

I.2.2. Définition de l'e-gouvernement ---------- 3

I.2.3.Définition de l'e-gouvernance ---------- 4

I.3. Les objectifs et les domaines d'E-gouvernement 4

I.3.1. Les objectifs d'e-gouvernement ---------- 4

I.3.2. Domaine d'application de l'E-gouvernement ---------- 4

I.3.2.1. L'E-administration ---------- 5

I.3.2.2. L'E-société ---------- 5

I.3.2.3. L'E-démocratie ---------- 6

I.3.3. Domaine d'application d'E-gouvernance ---------- 6

I.4. Les niveaux de l'e-gouvernement 7

I.4.1. Administration a Citoyen physique (G2C) ---------- 7

I.4.2. Administration a Administration (G2G) ---------- 7

I.4.3. Administration a Entreprise (G2B) ---------- 7

I.5. Les avantages et difficulté d'e -gouvernement 8

I.6. Exemples d'e-gouvernement 9

I.6.1. E-gouvernement au Maroc ---------- 9

I.6.2. E-gouvernement en Algérie ---------- 10

I.7. Conclusion 11

Chapitre II : Les Services web et la composition des services web

II.1. Introduction 14

II.2. l'architecture Oriente Service (SOA) 14

II.2.1.Definition ---------- 14

II.2.2. Les principes généraux de SOA ---------- 14

II.2.3. Les avantages de SOA ---------- 16
II.3.Les Services Web 16
II.3.1. Définition de Service web ---------- 16
II.3.2. Les caractéristiques des services web ---------- 17
II.4. Description du modèle fonctionnel des services web 17
II.4.1. Fonctionnement des services web ---------- 17
II.4.2. Architecture en couches des services web ---------- 18
II.5.Les technologies des services Web 19
II.5.1.XML (eXtansible Markup Language) ---------- 19
II.5.1.1.Définition ---------- 19
II.5.1.2. Les avantages du XML ---------- 19
II.5.2.SOAP (Simple Object Acces Protocol) ---------- 20
II.5.2.1.Définition ---------- 20
II.5.2.2.Etapes d'invocation d'objet distant avec soap ---------- 20
II.5.3.WSDL (Web Services Description Langage) ---------- 21
II.5.3.1.Définition ---------- 21
II.5.3.2.Organisation d'un document WSDL ---------- 22
II.5.4. UDDI (Universal Descovery and Interogation) ---------- 23
II.5.4.1.Définition ---------- 23
II.5.4.2.Structure de donnée de l'annuaire UDDI ---------- 24
II.6. Avantages et inconvénients des services web 25
II.6.1. Les avantages des services web ---------- 25
II.6.2. Les inconvénients des services web ---------- 25
II.7.La composition des services web 26
II.7.1 Définition ---------- 26
II.7.2. Cycle de vie d'une composition de services ---------- 27
II.8. Description et fonctionnement 28
II.8.1. L'orchestration de Service Web ---------- 28
II.8.2. La chorégraphie de Service Web ---------- 29
II.9. Type de Composition des services web 29
II.9.1.La composition Statique de services web ---------- 29
II.9.2.La composition Dynamique de service web ---------- 29
II.10.Langages de la composition des services web 30
II.10.1.BPEL4WS (Businiss Process Execution Language For Web Services) ---------- 30
II.10.2.WS-CDL (Web Services Choreography Description Language) ---------- 30

II.10.3.BPML (Business Process Modeling Language) ---------- 30
II.10.4.WSFL (Web Service Flow Language) ---------- 31
II.11.Les différentes approches de composition des services web 31
II.11.1.Approche oriente Workflow ---------- 31
II.11.2.Approches sémantiques ---------- 31
II.11.3.Approches basées sur les automates et réseaux de Pétri ---------- 31
II.12. Conclusion 32
Chapitre III :Etude conceptuelle
III.1 Introduction 34
III.2 Approche proposée 34
III.3. L'architecture conceptuelle de la plateforme 34
III.4. Description des modules 37
III.4.1. Le module de gestion des utilisateurs ---------- 37
III.4.2. Le module de publication ---------- 39
III.4.3. Le module de découverte ---------- 41
III.4.4. Le module de composition ---------- 41
III.5. Les entités de la plateforme 47
III.6. Les types d'utilisateur 48
III.6.1. Administrateur de la plateforme ---------- 48
III.6.2. Administrateur d'institution ---------- 48
III.6.3. Fournisseur ---------- 48
III.6.4. Demandeur ---------- 48
III.7. Conclusion 48
Chapitre IV : Implémentation et étude de cas
IV.1. Introduction 50
IV.2. Les outils et les langages utilisés en développement 50
IV.2.1. Langage JAVA ---------- 50
IV.2.2.Environnement de développement NetBeans ---------- 50
IV.2.3. Les JSP (Java Server Pages) ---------- 51
IV.2.4.Glass Fish Server ---------- 51
IV.2.5. MY SQL ---------- 51
IV.2.6. JDOM ---------- 51
IV.2.7. HTM L (L'HyperText Markup Language) ---------- 51
IV.2.8.Les CSS (feuilles de styles en cascade) ---------- 52

Sommaire

IV.3.La mise en œuvre de la plateforme 52

IV.4. Etude de cas d'utilisation 55

IV.4.1.Description des services web utilisés ---------- 55

IV.4.2.Scenario de la composition ---------- 57

IV.4.3.Resultat et discurion.----------58

Conclusion générale 64

Bibliographie ……..65

ANNEXE 69

Liste des figures

Liste des figures :

Figure I.1 : Domaine d'application de l'e-gouvernement---------- 5

Figure II.1 : Les principes généraux de SOA. ---------- 15

Figure II.2 : Fonctionnement des services web. ---------- 17

Figure II.3 : Architecture en couches des services web. ---------- 19

Figure II.4 : Structure d'un message SOAP. ---------- 20

Figure II.5: Etapes d'invocation d'objets distant avec Soap ---------- ---------- 21

Figure II.6 : Organisation d'un document WSDL. ---------- 22

Figure II.7 : Le contenu de l'annuaire UDDI. ----- ---------- 24

Figure II.8: La structure de donnée de l'annuaire UDDI. ---------- 25

Figure II.9: Exemple de composition des services web. ---------- 26

Figure II.10 : Cycle de vie d'une composition de services web. ---------- 27

Figure II.11 : Principe de l'orchestration des services web. ---------- 28

Figure II.12 : Principe de la chorégraphie des services web---------- 29

Figure III.1 : Architecture globale de notre plateforme---------- 35

Figure III.2 : Fonctionnement de la plateforme---------- 36

Figure III.3 : Module de gestion d'utilisateurs---------- 38

Figure III.4 : Module de publication ---------- 40

Figure III.5 : Module de composition---------- 42

Figure III.6 : Similarité entre les paramètres de services web d'entrée et de sortie---------- 43

Figure III.7 : Algorithme de composition---------- 45

Figure III.8 : Diagramme de classe de la plateforme---------- 47

Figure IV.1 : Schéma globale de la plateforme ---------- 53

Liste des figures

Figure IV.2: L'organigramme d'utilisation de la plateforme---------- 54

Figure IV.3 : Service état civil ---------- 55

Figure IV.4 : Service diplôme universitaire. ---------- 56

Figure IV.5 : Service expérience de travail. ---------- 56

Figure IV.6 : Service de santé. ---------- 56

Figure IV.7 : Service casier judiciaire. ---------- 57

Figure IV.8 : Schéma explicatif du système. ---------- 58

Figure 01 : Fenêtre authentification ---------- 79

Figure 02 : Fenêtre d'accueil. ---------- 79

Figure 03: Fenêtre recherche . ---------- 80

Liste de graphes :

Graphe 01 : Pourcentage des requetes calculant par similarite syntaxique---------- 61

Graphe 02 : Pourcentage des requetes calculant par similarite semantique---------- 62

Graphe 03 : Similarité syntaxique vs Similarité sémantique---------- 62

Liste des tableaux :

Tableau I.1 : Classement d'Algérie une-goverment Knowledgebase ----------10
Tableau I.2: Comparaison une-government Knowledgebase----------11
Tableau IV.1 : Similarité de 37 requêtes ---------- 61
Tableau IV.2 : Requêtes SIM>=50 ---------- 61
Tableau IV.3 : Pourcentage de requêtes ----------61

Liste des acronymes

Liste des acronymes :

Mot	Description
API	Application Programming Interface
BPML	Business Process Modeling Language
CSS	Cascading Style Sheets
CORBA	Common Object Request Broker Architecture
DOM	Document Object Model
DAML-S	Dorpa Agent Markup Language for Services
FTP	File Transfer Protocol
G2B	Government to Business
G2C	Government to Citizen
G2G	Government to Government
HTTP	Hyper Text Transfer Protocol
IBM	International Business Machines Corporation
JAVAEE	Java Enterprise Edition
JDOM	Java Document Object Model
JSP	Java Server Page
RMI	Remote Method Invocation
SMTP	Simple Mail Transfer Protocol
SAP	Server Access Protocol

Liste des acronymes

SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
TIC	Technology of Information for communication
URL	Uniform Resource Identifier
UDDI	Universal Description Discovery and Integration
OWL-S	Web Ontologiy Language for Web Services
W3C	Consortium of World Wide Web
WSDL	Web Service Description Language
WSFL	Web Service Flow Language
XSLT	Extensible Stylesheet Language Transformation
XML	Extensible Markup Language

Introduction générale :

L'informatisation de l'administration a considérablement simplifié les tâches administratives et a par conséquent aidé les institutions à améliorer la prestation de leurs services aux clients et aux citoyens. Cependant, une communication efficace entre les systèmes d'information des institutions est nécessaire pour rendre leurs services plus accessibles. Cela nécessite de traiter une donnée importante et de poser un gros problème d'intégration d'applications. En effet, la communication entre les systèmes d'information et les applications des différentes institutions devient de plus en plus difficile à cause des technologies hétérogènes utilisées, ce qui amène les utilisateurs à travailler dans un environnement incohérent, mal adapté et / ou incompatible.

Les services Web sont des logiciels d'application, identifiés par l'URI (Uniform Resource Identifier), parmi lesquels les interfaces et les connexions peuvent être définies, décrites et découvertes sous la forme de documents XML (Extensible Markup Language).

Les services Web peuvent être utilisés pour mettre en œuvre l'interaction directe entre les applications en messagerie du XML sur des protocoles Internet.

En utilisant les services Web et en prenant en compte la notion d'identité numérique, nous s'intéressons en particulier à la découverte et à la composition de services Web en fonction de l'identité du citoyen, de l'institution et du service demandé.

L'objectif de notre projet est de créer une plate-forme basée sur la composition dynamique des services web, qui permet à une institution de récupérer des données fournies par d'autres institutions pour construire une partie de l'identité numérique d'un client ou d'un citoyen afin de lui offrir un service.

Nous avons divisé ce mémoire en quatre chapitres :

Chapitre I : dans ce chapitre nous allons illustrer la notion d'e-gouvernement, ainsi les objectifs principaux et les domaines, les niveaux, les avantages et difficultés de cette dernière. Ensuite, nous allons présenter la relation d'e-gouvernement avec les citoyens et les entreprises.

Chapitre II : dans le deuxième chapitre, nous allons présenter l'architecture oriente services, la définition des services web, le fonctionnement, les avantages et les inconvénients, et Enfin,

un aperçu sur la composition des services, les types de composition, les langages et les différentes approches de cette dernier.

Chapitre III : dans le troisième chapitre, nous allons présenter la conception de l'approche proposée de la plateforme et l'architecteur de cette dernière.

Chapitre IV : dans le quatrième chapitre, nous allons représenter le schème global de notre plateforme, la mise en œuvre de cette dernière et l'étude de cas.

Chapitre I :

gouvernement électronique

« E-gouvernement »

I.1. Introduction :

L'information et la communication sont deux technologies qui sont devenues partie intégrante de la vie quotidienne dans l'ensemble des domaines de l'activité humaine. Les services générés par les TIC (Technologies de l'Information et de la Communication) facilitent la dématérialisation via la numérisation de l'information et la circulation de l'information via les réseaux. Leur interconnexion, notamment Internet, constitue un outil formidable pour les gouvernements afin de rendre leurs activités plus ouvertes et plus transparentes.

Le gouvernement électronique ou gouvernement numérique est un moyen qui utilise la TIC afin de rendre les services publics plus accessibles aux usagers. Les TIC permettent de suivre toutes les étapes des activités administratives et de voir là où les décisions sont préparées et prises.

Ce chapitre présente une vue globale de l'e-gouvernement, des définitions, les objectifs, les avantages et les inconvenants.

I.2. Définition et compréhension de l'e-gouvernement :

Il existe plusieurs définitions de l'e-gouvernement de plusieurs points de vue internationaux.

I.2.1. Définition de gouvernement :

- « De manière générale, la gouvernance n'est pas un concept nouveau car elle fait partie du langage de la philosophie et des sciences politiques depuis Platon et Aristote » [1].
- Selon l'Office de la langue française la gouvernance correspond à la « manière d'orienter, de guider, de coordonner les activités d'un pays, d'une région, d'un groupe social ou d'une organisation privée ou publique » [2].
- Selon James Rosenau qui définit la gouvernance comme « un ensemble de mécanismes de régulation dans une sphère d'activité, qui fonctionnent même s'ils n'émanent pas d'une autorité officielle » [3].

I.2.2. Définition de l'e-gouvernement :

L'e-gouvernement implique l'automatisation ou l'informatisation des procédures existantes non-informatisées, ce qui conduit à de nouveaux styles de leadership, de nouvelles façons de débattre et de décider des stratégies, d'interagir avec les sociétés commerciales,

d'écouter les citoyens et les communautés, d'organiser et de livrer l'information, et donc essentiellement à de nouvelles façons de gouverner. Par conséquent, le processus qui consiste à utiliser, améliorer, inventer et gérer les outils d'e-gouvernement pour gouverner est appelé l'e-gouvernance. [4]

I.2.3.Définition de l'e-gouvernance :

La gouvernance électronique est également définie comme « l'utilisation des technologies de l'information et des communications (TIC) dans les administrations publiques, associée à des changements au niveau de l'organisation et de nouvelles aptitudes du personnel ».

L'objectif est d'améliorer les services publics, renforcer les processus démocratiques et de soutien aux politiques publiques. La gouvernance électronique signifie, l'ensemble des mesures ou décisions qui contribuent à définir le cadre dans lequel évoluent la société du savoir, l'administration électronique et plus généralement les TIC. [4]

I.3. Les objectifs et les domaines d'E-gouvernement :

Les objectifs d'e-gouvernement sont divers :

I.3.1. Les objectifs d'e-gouvernement :

Selon [5] on peut spécifier comme suit :

- Transformer le gouvernement comme organisation plus responsable, transparente et efficace.
- D'améliorer la fourniture d'informations et de services.
- D'encourager la participation du citoyen au processus de décision.
- Rendre le gouvernement ou l'administration plus responsable.
- Fournir des meilleurs services, moins couteux et plus fiables.
- Toujours fournir les informations en ligne sur le gouvernement.
- Donner aux utilisateurs le choix de décider quand, d'où et comment, ils souhaitent interagir avec le gouvernement.

I.3.2. Domaine d'application de l'E-gouvernement :

De manière générale, l'e-gouvernement se divise en 3 grands blocs, illustré dans la **(Figure I.1) :**

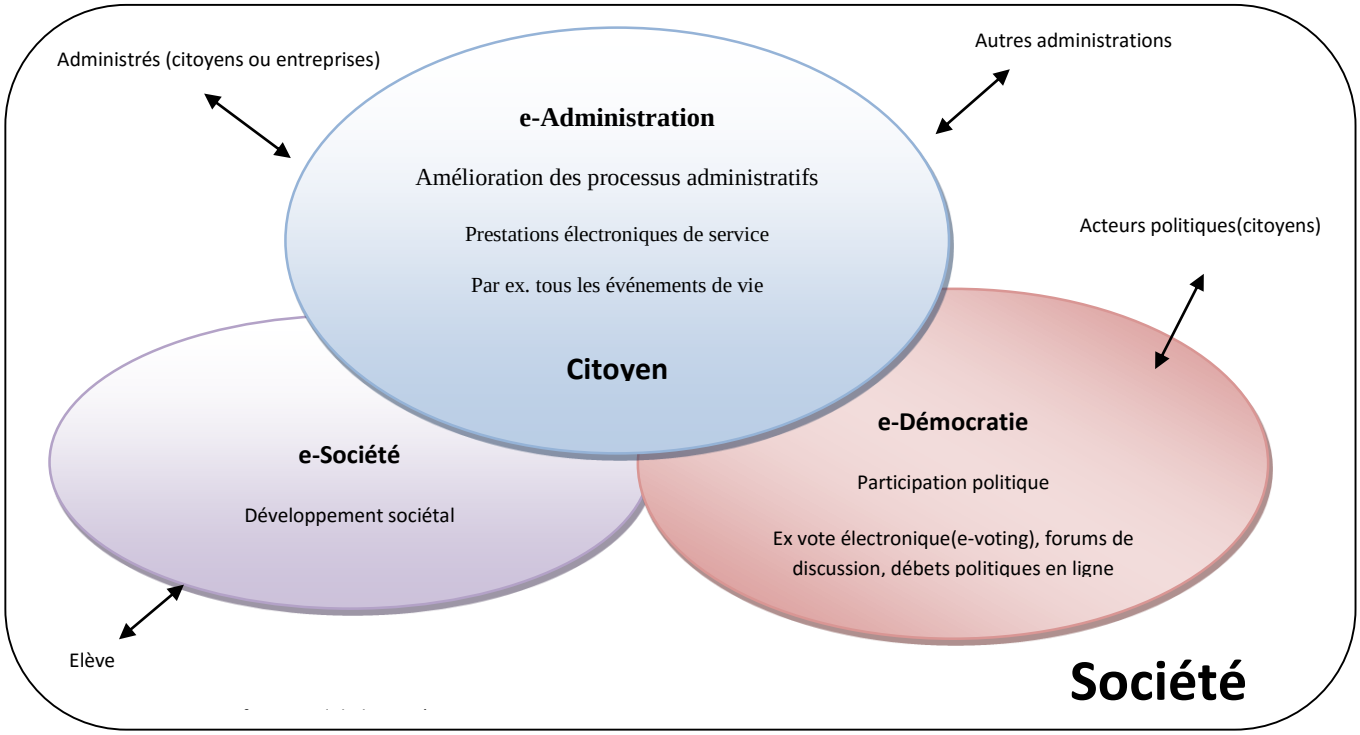


Figure I.1 : Domaine d'application de l'e-gouvernement

I.3.2.1. L'E-administration :

L'objectif global de l'E-administration est l'amélioration de la performance, de la cohérence, de la transparence et de la compétence interne de l'administration publique par l'informatisation de tous les services administratifs aussi bien au niveau central qu'au niveau décentralisé. L'E-administration vise particulièrement à [1] :

- Promouvoir l'accès à l'information des citoyens sur leurs droits et devoirs dans les domaines variés (santé, éducation, etc…) de l'assainissement et d'autres prestations de base des services publics.
- Assurer la transparence des informations, des décisions, des actions de l'état et des collectivités territoriales.

I.3.2.2. L'E-société :

De telles initiatives portent, en particulier, sur les rapports entre les agences publiques et les autres institutions d'autres agences publiques, les sociétés du secteur privé, les organisations à but non lucratif et communautaires. Comme pour les connexions entre citoyens, ces initiatives pourraient bien incorporer les améliorations des procédures. Toutefois, elles pourraient inclure des attributions plus larges comme [5] :

• ***Mieux travailler avec les entreprises*** **:** amélioration de l'interaction entre le gouvernement et les entreprises. Cela pourrait comprendre la digitalisation de la réglementation sur l'approvisionnement et les services en direction des entreprises pour en améliorer la qualité, la commodité et le coût.

• ***Développer les communautés :*** renforcer les capacités sociales et économiques ainsi que le capital des communautés locales.

• ***Renforcer les partenariats :*** créer des groupements d'organisations pour atteindre les objectifs économiques et sociaux. Le secteur est presque toujours l'un des partenaires, même si occasionnellement, il n'agit qu'en facilitateur pour les autres.

I.3.2.3. L'E-démocratie : [2]

L'E-démocratie vise le développement et l'amélioration des relations avec le citoyen en tant qu'acteur politique et détenteur de la légitimité ainsi que les relations intergouvernementales, distingue 3 versions de la démocratie électronique : une version populiste, une version libérale, une version républicaine.

Son analyse de la littérature sur la démocratie électronique montre que celle-ci s'organise autour de 3 questions principales :

* l'information des citoyens ?

*Le débat et la discussion ?

* la délibération et la prise de décision publique ?

I.3.3. Domaine d'application d'E-gouvernance :

- **Engagement électronique (C2G) :**

Il y a un intérêt croissant pour l'utilisation des applications des TIC afin de faciliter et encourager une contribution réelle des citoyens au cycle de formulation des politiques gouvernementales. Néanmoins, ces applications sont encore limitées dans la pratique et par leurs résultats, comme le montrent les résultats de l'enquête sur l'e-gouvernement menée par l'ONU (UN 2008). [6]

- **Consultation électronique (G2C) :**

Afin de faciliter les interactions des responsables du gouvernement avec les citoyens et les groupes de lobbies, des canaux informatisés sont de plus en plus déployés. Le but est de faciliter le processus de participation des électeurs dans les consultations liées à l'élaboration et à la promulgation des lois. Cependant, cette application en est aussi à ses débuts. [6]

- **Contrôle électronique (G2G) :**

Le contrôle est un ensemble de procédures et de mécanismes d'audit interne (ou externe) pour faire le suivi de divers éléments du fonctionnement organisationnel tels que les mesures des processus de dépenses dans la chaîne de valeur (entrées, flux et sorties), la performance et les résultats. Avec le développement des TIC, des moyens électroniques sont de plus en plus utilisés pour mesurer la performance des applications mises à la disposition des clients à travers les réseaux de télécommunication. [6]

I.4. Les niveaux de l'e-gouvernement :

L'e-gouvernance est vue comme l'application de moyens électroniques, on trouve trois principaux niveaux d'e-gouvernement, la relation entre ces trois niveaux :

I.4.1. Administration à Citoyen physique (G2C) :

L'interaction entre le gouvernement et les citoyens englobe les initiatives visant à faciliter l'interaction des gens avec le gouvernement en tant que consommateurs de services publics et en tant que citoyens. Cela comprend les interactions liées à la prestation de services publics, ainsi que la participation à la consultation et au processus de prise de décision. [4]

I.4.2. Administration à Administration (G2G) :

L'interaction entre le gouvernement et le gouvernement implique le partage de données et la conduite des échanges électroniques entre acteurs gouvernementaux. Cela implique des échanges à la fois intra- et interinstitutionnels au niveau national, ainsi que des échanges entre les niveaux national, provincial et local. [4]

I.4.3. Administration à Entreprise (G2B) :

L'interaction entre le gouvernement et les entreprises englobe les transactions spécifiques aux affaires (par exemple les paiements effectués en ce qui concerne la vente et achat de biens et de services) ainsi que la fourniture en ligne de services axés sur les entreprises. [4]

I.5. Les avantages et inconvénient d'e -gouvernement :

Selon [7] L'e-gouvernement a beaucoup d'avantages dont on peut citer :

• Le gouvernement devient moins cher grâce à l'utilisation des technologies de l'information et de communication.

• Le gouvernement produit plus de services au même coût ou à un moindre prix.

• Les opérations du gouvernement deviennent plus rapides à la différence de traitement manuel qui repose sur le facteur humain qui exige un temps d'exécution plus long.

• Le gouvernement produit des services de qualité supérieure. En effet, le gouvernement électronique assure une communication de qualité et une transparence de l'information.

• Les gouvernements deviennent inventifs et plus créatifs.

• Le gouvernement électronique peut aider à bâtir la confiance par l'engagement des citoyens, et ce de plus en plus dans les politiques publiques.

• Le développement de la démocratie numérique : l'utilisation du courrier électronique permet de se connecter avec les élus, de voter par voie de l'Internet, et de participer aux réunions gouvernementales grâce à la technologie des téléconférences.

• Les technologies de l'information contribuent au développement 1économique : l'économie dépend plus en plus de l'accès à l'information et de l'utilisation de la technologie pour améliorer la prestation des services.

Les inconvénients rencontrés qui ont conduit à des échecs dans l'adoption d'un gouvernement électroniques sont :

- La résistance que certains citoyens pourraient manifester face à une telle transformation fondamentale des systèmes de fonctionnement de l'administration publique.
- L'échec de l'infrastructure des technologies de l'information, qui peuvent résulter d'une programmation défectueuse ainsi que de l'absence d'une infrastructure propice.
- Les possibles malentendus entre l'équipe responsable des technologies de l'information et des équipes gouvernementales.

I.6. Exemples d'e-gouvernement :

I.6.1. E-gouvernement au Maroc :

Le Maroc a œuvré pour l'intégration de la gouvernance électronique à travers plusieurs projets et initiatives, preuve de la grande volonté politique des dirigeants du Royaume [8].

Selon le document publié par les nations unies « l'État de l'information pour les activités de développement en Afrique du nord », La volonté du Royaume du Maroc d'entrer dans l'ère de l'information s'exprime à travers la stratégie nationale e-Maroc, qui poursuit selon la commission économique pour l'Afrique attachée aux nations unies, cinq objectifs : [2]

1. La généralisation des TIC.
2. Le déploiement accéléré des infrastructures.
3. L'accélération de la libéralisation et de la concurrence.
4. La redéfinition du rôle de l'État.
5. La mobilisation et la mise en place des moyens de la stratégie.

En consultant les anciens plans stratégiques, notamment le plan quinquennal TIC 1999-2003, qui était à la base de la stratégie e-Maroc 2010, on retrouve que celui-ci s'articulait autour des axes de développement suivants [2] :

- Élaborer une législation capable de répondre aux problématiques actuelles (protection des noms de domaines commerciaux, protection des données nominatives, besoins du commerce électronique, etc.).
- Amorcer un tournant au niveau de l'éducation (alphabétisation, intégration des outils informatiques et d'Internet dans les écoles, etc.).
- Réduire, à défaut d'éviter, la fracture numérique par des politiques qui favorisent le déploiement des TIC dans les régions et permettent leur utilisation même par les couches les plus démunies de la population.
- Financer la création d'entreprises de la nouvelle économie du savoir (capital risque, prêts au démarrage des entreprises à fort contenu technologique, etc.).

I.6.2. E-gouvernement en Algérie :

Le classement international de l'Algérie au niveau de l'indice des Nations Unies pour l'administration électronique constate un retard accusé par le pays. L'Algérie a été classé classée 132ème en 2012 et 136ème en 2014 parmi 193 pays.

Ce classement international tient compte de trois critères : l'infrastructure et les réseaux des télécoms, le capital humain et les services en ligne (e-gouvernance, e-participation, open gouvernement data, etc.). En raison de ce retard, la thématique de l'e-gouvernement a été mise au cœur de la stratégie e-Algérie 2013.

Le gouvernement a lancé différents chantiers depuis l'année 2000 pour encourager l'utilisation des technologies de l'information par les administrations et les organismes publics, espérant ainsi opérer une transformation des modes de travail et d'organisation et donner un meilleur accès des informations aux citoyens. Malgré le lancement de sites internet et de portails thématiques, notamment dans les secteurs de l'éducation, la justice et la santé, des efforts demeurent nécessaires pour améliorer le nombre de sites disponibles, assurer une actualisation régulière de leurs contenus et faciliter l'accès des citoyens à l'information. [4]

L'Algérie a amélioré son classement de 20 places dans le domaine de l'administration électronique, selon la dernière édition en 2016 de l'enquête bisannuelle de l'Organisation des Nations Unies sur l'indice de développement l'E-Gouvernement en 2018 dans le monde, publié jeudi 19 juillet [9].

Le (**Tableau I.1**) présentent le classement de l'Algérie au niveau de l'indice des Nations Unies Pour l'administration électronique (une E-Government Knowledgebase) sur 193 pays.

EGDI Rank	2018	2016	2014	2012	2010	2008
Algérie	130	150	136	132	131	121

Tableau I.1 : classement d'Algérie une E-Government Knowledgebase

Le tableau suivant (**Tableau I.2**) présente la comparaison de site une E-Government Knowledgebase entre Maroc, Tunisie, Algérie et Égypte depuis 2008 jusqu'à 2018.

EGDI Rank	2008	2010	2012	2014	2016	2018
Maroc	140	126	120	82	85	110
Tunisie	124	66	103	75	72	80
Algérie	121	131	132	136	150	130
Égypte	79	86	107	80	108	114

Tableau 1.2 : comparaison une E-Government Knowledgebase

I.7. Conclusion :

Dans le chapitre suivant, nous avons commencé par les définitions des termes gouvernement et l'e-gouvernement, ensuite nous avons déterminé les objectifs principaux et les éléments stratégiques (la stratégie, financement, leadership, etc.) de L'e-gouvernement. Aussi, nous avons vu les domaines de l'e-gouvernement (e-administration, e-société et e-démocratie) et les Domaine de l'e-gouvernance (Engagement électronique, Consultation électronique, Contrôle électronique).

Nous avons présenté aussi les avantages d'e-gouvernement et certains risques et défis de cette technologie. Enfin nous avons présenté quelques exemples sur l'e-gouvernement.

Chapitre II :

Les Services web

II.1. Introduction :

Dans les dernières années, les services web sont devenus le nouveau point de convergence technologique de l'industrie du logiciel, permettre de la communication à travers un réseau internet entre applications et systèmes hétérogènes écrites dans de langage de programmation différentes sur différentes plateformes, dans le but principale d'échanger des données.

Un bon nombre d'éditeurs de serveurs d'applications, d'outils de développement, ont placé ces services web au cœur de leur stratégie [10].

Depuis les dernières années, l'utilisation de Services web a connu une popularité grandissante. Ces Services sont très utilisés notamment par les entreprises pour rendre accessible leurs métiers ou leurs données via le Web. Les Services web, tels qu'ils sont présentés, sont conceptuellement limités à des fonctionnalités relativement simples qui sont modélisées par une collection d'opérations. Toutefois, il est nécessaire de construire des nouvelles applications par le biais de la composition de services afin de répondre à des exigences plus complexes. [11]

Dans ce chapitre nous présentons les définitions, types de composition et en fin les langages de composition.

II.2. l'architecture Oriente Service (SOA) :

II.2.1.Definition : [12]

C'est une architecture orientée services (notée SOA) est une architecture logicielle s'appuyant sur un ensemble de services simples appelés Services Web.

L'objectif d'une architecture orientée services est donc de décomposer une fonctionnalité en un ensemble de fonctions basiques, appelées services, fournies par des composants et de décrire finement le schéma d'interaction entre ces services.

L'idée sous-jacente est de cesser de construire la vie de l'entreprise autour d'applications pour faire en sorte de construire une architecture logicielle globale décomposées en services correspondant aux processus métiers de l'entreprise.

II.2.2. Les principes généraux de SOA : [13]

Il n'existe pas à proprement parler de spécifications officielles d'une architecture SOA, néanmoins les principales notions fédératrices que l'on retrouve dans une telle architecture sont les suivantes :

- **La notion de service,** c'est-à-dire une fonction encapsulée dans un composant que l'on peut interroger à l'aide d'une requête composée d'un ou plusieurs paramètres et fournissant une ou plusieurs réponses. Idéalement chaque service doit être indépendant des autres afin de garantir sa réutilisabilité et son interopérabilité.
- **La description du service,** consistant à décrire les paramètres d'entrée du service et le format et le type des données retournées.
- **Le principal format de description de services** est WSDL (Web Services Description Language), normalisé par le W3C (Consortium du World Wide Web).
- **La publication** (en anglais advertising). La publication consiste à publier dans un registre (en anglais registry ou repository) les services disponibles aux utilisateurs, tandis que la notion de découverte recouvre la possibilité de rechercher un service parmi ceux qui ont été publiés. Le principal standard utilisé est UDDI (Universal Description Discovery and Integration).
- **L'invocation,** représentant la connexion et l'interaction du client avec le service. Le principal protocole utilisé pour l'invocation de services est SOAP (Simple Object Access Protocol).

La figure suivante **(Figure II.1)** représente les principes généraux de SOA :

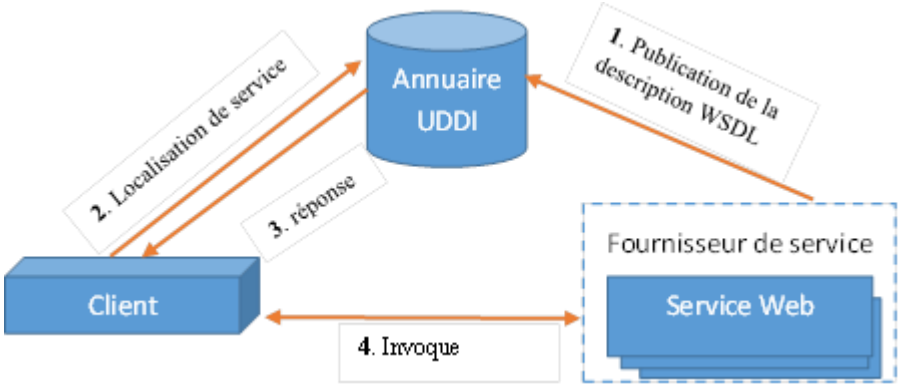


Figure II.1 : Les principes généraux de SOA

II.2.3. Les avantages de SOA : [12]

Une architecture orientée services permet d'obtenir tous les avantages d'une architecture client-serveur et notamment :

- Une modularité permettant de remplacer facilement un composant (service) par un autre.
- Une réutilisabilité possible des composants (par opposition à un système tout-en-un fait sur mesure pour une organisation).
- De meilleures possibilités d'évolution (il suffit de faire évoluer un service ou d'ajouter un nouveau service)
- Une plus grande tolérance aux pannes, une maintenance facilitée.

II.3.Les Services Web :

Dans les sections suivantes, nous présentons les concepts des services web qui proposent une solution pour implémenter l'architecture orientée service.

II.3.1. Définition de Service web :

Nous allons citer quelques définitions :

- **Un service Web selon W3C (World Wide Web Consortium)** :

  Est une application ou un composant logiciel conçu pour permettre une interaction interopérable de machine à machine via un réseau. Identifié par un URI (Uniform Resource Identifier), il dispose d'une interface décrite dans un format exploitable par des machines. Il peut également interagir directement avec d'autres services Web à travers le langage XML et en utilisant des protocoles Internet. [12]

- **Selon IBM :**

  « Les services web sont la nouvelle vague des applications web. Ce sont des applications modulaires, auto-contenues et auto-descriptives qui peuvent être publiées, localisées et invoquées depuis le web. Les services web effectuent des actions allant de simples requêtes à des processus métiers complexes. Une fois qu'un service web est déployé, d'autres applications (y compris des services web) peuvent le découvrir et l'invoquer » [13]

II.3.2. Les caractéristiques des services web :

La technologie des services Web repose essentiellement sur une représentation standard des données (interfaces, messageries) au moyen du langage XML. Cette technologie est devenue la base de l'informatique distribuée sur Internet et offre beaucoup d'opportunités au développeur Web. Un service Web possède les caractéristiques suivantes [14] :

- ❖ Il est accessible via le réseau ;
- ❖ il dispose d'une interface publique (ensemble d'opérations) décrite en XML ;
- ❖ ses descriptions (fonctionnalités, comment l'invoquer et où le trouver ?) sont stockées dans un annuaire ;
- ❖ il communique en utilisant des messages XML, ces messages sont transportés par des protocoles Internet (généralement http (Hyper Text Transfer Protocol), mais rien n'empêche d'utiliser d'autres protocoles de transfert tels : SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), ... ;
- ❖ l'intégration d'application en implémentant des services Web produit des systèmes faiblement couplés.

II.4.: Description du modèle fonctionnel des services web

II.4.1. Fonctionnement des services web :

Le fonctionnement des services web passe par plusieurs étapes clarifiées dans (**Figure II.2**) :

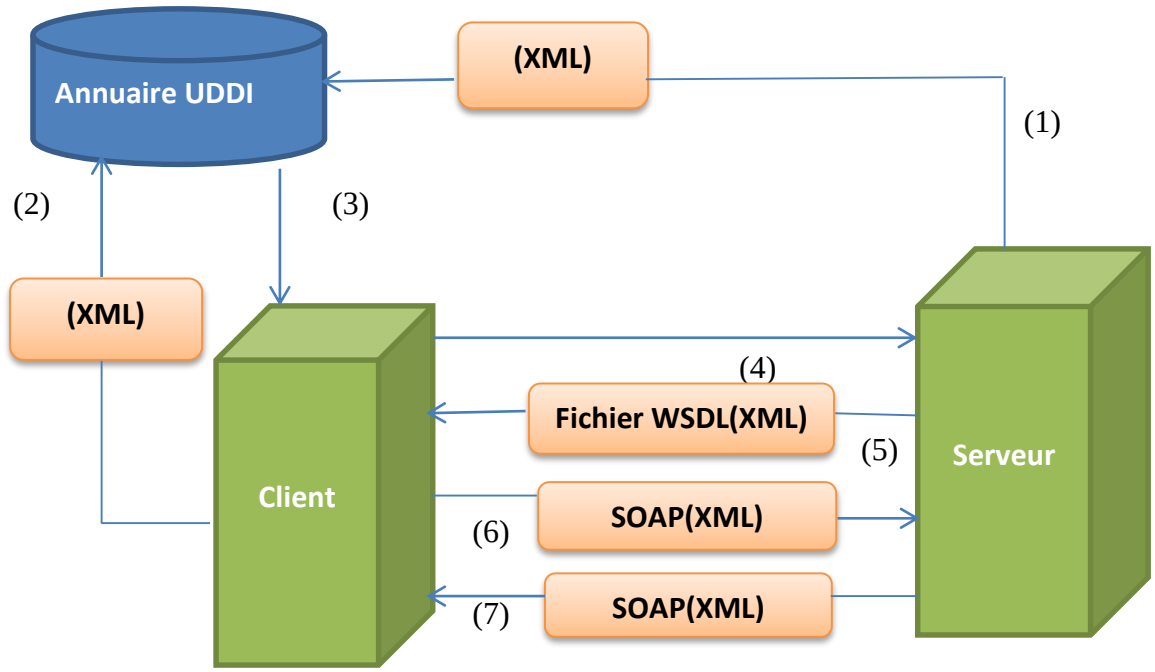


Figure II.2 : Fonctionnement des services web.

(1) : Le fournisseur de services web publie ses services web.

(2) : Le client recherche un service web avec des caractéristiques.

(3) : L'annuaire a trouvé un service avec les caractéristiques de client, il envoie l'information du serveur qui héberge.

(4) : Le client testé : quel est le format d'appel du service que tu proposes ?

(5) : Voici mon contrat (WSDL).

(6) : J'ai compris comment invoquer ton service et je t'envoie un document XML représentant ma requête.

(7) : J'ai exécuté ta requête et je te retourne le résultat.

II.4.2. Architecture en couches des services web :

Les Services Web se basant sur l'architecture orientée services, emploient un ensemble de technologies à base d'une architecture en couches. Selon [15] Cette structure se base sur quatre couches.

- **La couche de transport :** couche de transport des messages SOAP (les différentes communications) assuré par le protocole HTTP, SMTP, FTP.
- **La couche de communication :** couche qui assure la communication utilise le protocole SOAP basée sur XML pour accéder à des Services Web.
- **La couche description de Servies :** cette couche présente une description de Services Web, représente les informations nécessaires afin de facilite l'interaction entre les clients et les fournisseurs de Service Web, et en même temps leur utilisation.
- **La couche découverte de Service ou couche publication de Service :** cette couche à pour centraliser et regrouper les Services Web dans un registre commun, assurer par un annuaire UDDI, pour but de simplifier la recherche et de publications des Services Web.

La figure suivante (**Figure II.3**) représente l'architecture en couches des services web :

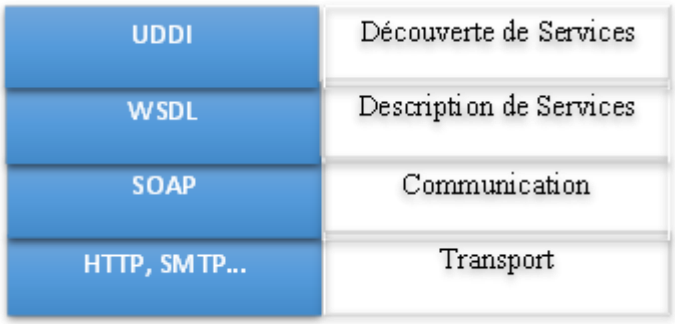


Figure II.3 : Architecture en couches des services web

II.5.Les technologies des services Web :

Dans cette section nous vous présentons les grandes lignes de chacun des standards de bases de technologies des Services Web, SOAP, WSDL et, ces trois standards sont basés sur le langage XML.

II.5.1.XML (eXtansible Markup Language) :

II.5.1.1.Définition :

XML c'est un langage pour structurer des contenus et définir une classe d'objets de données, le langage base sur le concept de balisage de données. La structure d'un document XML est souvent présentée sous forme d'un arbre. [15]

II.5.1.2. Les avantages du XML :

Les avantages de documents XML sont multiples nous vous citons : [16]

• **Lisibilité :** il est facile de lire un document XML car le code structuré et simple à comprendre.

• **Disponibilité :** on peut créer un document XML à partir d'un simple éditeur de traitement de texte.

• **Interopérabilité :** quel système d'exploitation peut lire ce langage.

• **Extensibilité :** on peut ajouter des nouvelles balises en fonction de nos besoins.

II.5.2.SOAP (Simple Object Acces Protocol) :

II.5.2.1.Définition :

C'est le protocole de messagerie qui est devenu le standard utilisé pour les services web. Il définit la façon de faire des invocations de services et d'échanger des données structurées d'une manière standardisée en utilisant XML et HTTP comme protocole de transport des messages SOAP. La première version de SOAP a été acceptée par W3C en 2000. Il se compose de deux parties :

- **L'en-tête du protocole de transport (Protocol Header).**
- **Une enveloppe :** qui définit un Framework pour décrire le contenu d'un message, et contient trois sous composant, SOAP Header, SOAP Body, SOAP Fault. [15]

La figure suivante **(Figure II.4)** représente la structure d'un message SOAP :

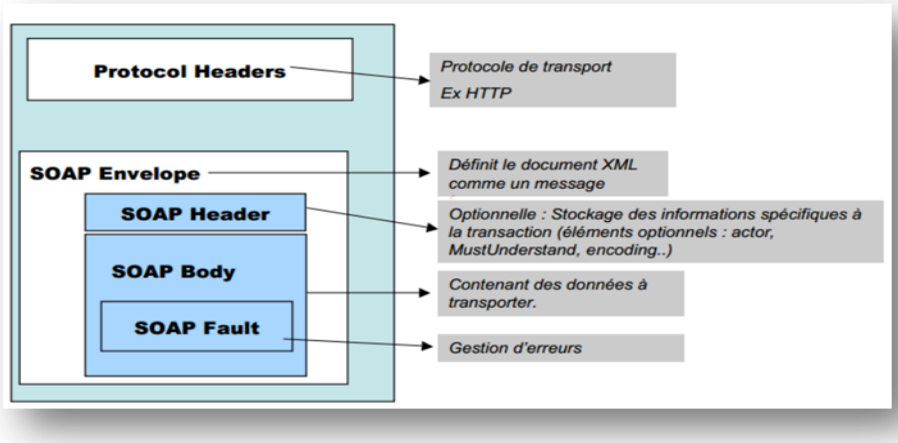


Figure II.4 : Structure d'un message SOAP

II.5.2.2.Etapes d'invocation d'objet distant avec soap :

La figure suivante (**Figure II.5)** représente les étapes d'invocation d'objet distant avec SOAP :

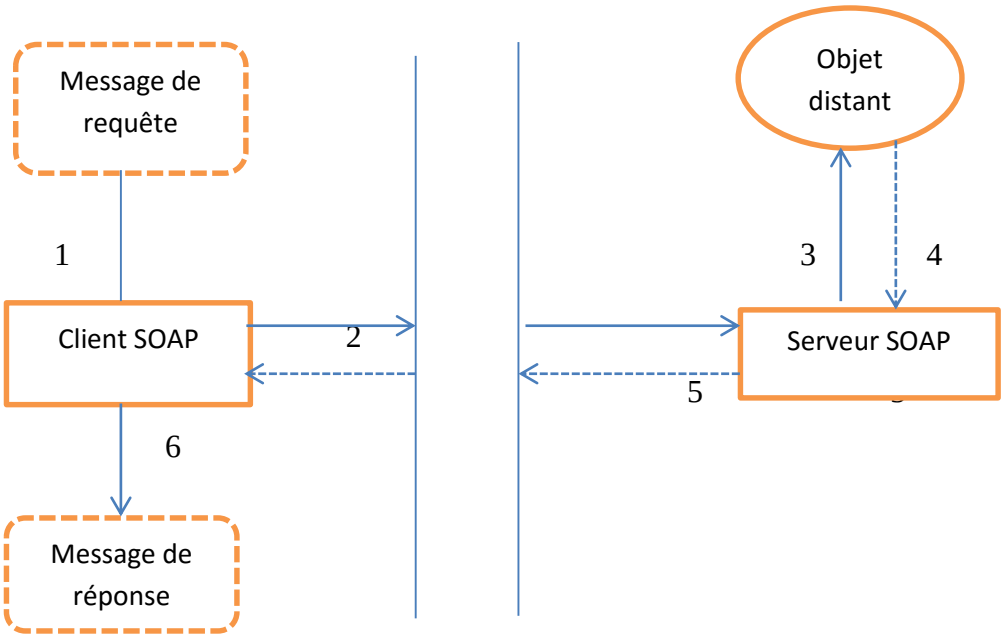


Figure II.5 : Etapes d'invocation d'objets distants avec SOAP

1) Le client SOAP crée un document XML qui contient les informations nécessaires pour invoquer les services d'un objet distant. Ensuite ce document est inséré dans une enveloppe SOAP avant d'être transmis sous forme d'une requête HTTP.

2) Le message est transmis via une connexion HTTP.

3) Le message est reçu et analysé par le serveur SOAP, ensuite envoyé à l'objet distant.

4) L'objet fait le traitement de la requête et envoie la réponse au serveur SOAP.

5) La réponse est envoyée, sous forme d'un document SOAP, au client via le HTTP.

6) Le client reçoit la réponse, ouvre l'enveloppe et envoie le résultat au demandeur initial.

II.5.3.WSDL (Web Services Description Langage) :

II.5.3.1.Définition :

WSDL introduit une grammaire commune pour la description des services. Le langage WSDL (W3C), est un langage basé sur XML permettant de décrire et de publier les interfaces et protocoles des services Web d'une manière standard. [17]

II.5.3.2.Organisation d'un document WSDL :

WSDL est utilisé pour définir les interfaces des services web. [18] Il décrit aussi deux différents aspects de service. Ces signatures qui se représentent le nom et les paramètres et ses détails de liaison et du déploiement.

Dans un fichier WSDL, chaque service web est une entité, qui est définie par des ports, qui sont les points finaux du service. Ces mêmes ports peuvent recevoir et répondre à un ensemble de messages définis par le type du port .Chaque port est, en réalité, la liaison entre le (portType) et un protocole d'accès, qui nous dit comment les messages doivent être encodés et envoyés au port.

Un document WSDL est divisé en deux parties (partie abstraite et partie concrète), qui contient sept sections qui est représenté dans la (**Figure II.6**) :

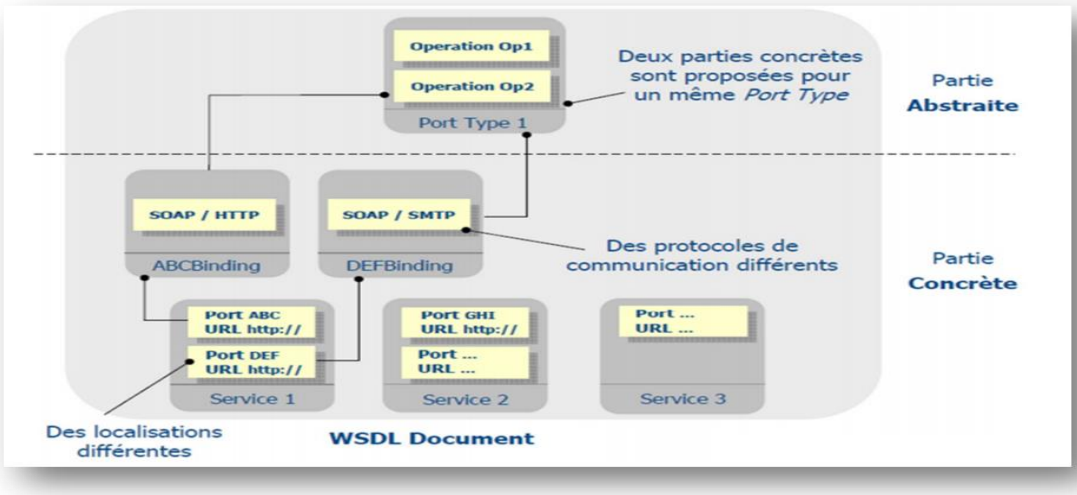


Figure II.6 : Organisation d'un document WSDL

Dans la suite, nous définissons chacun des éléments mis en évidence dans (la Figure II.6) :

a. **Au niveau abstrait :** la description du Service Web consiste à définir les éléments de l'interface du service web tel que les types de données (Data Types), les messages (Message), les types de ports (Port Type). Ces parties décrivent des informations abstraites indépendantes au contexte de mise en œuvre. On y trouve les types de données envoyées et reçues, les opérations utilisables et le protocole qui sera utilisé. [18]

b. **Au niveau concret :** le Service Web est défini grâce aux éléments Bindings et Service & Port. Ces deux derniers décrivent des informations liées à un usage contextuel du Service

Web. On y trouve l'adresse du fournisseur implémentant le service, et le service qui est représenté par les adresses des fournisseurs. [18]

II.5.4. UDDI (Universal Descovery and Interogation) :

II.5.4.1.Définition :

• **UDDI** est l'une des pièces maîtresses des services web. Cette technologie joue le rôle d'un annuaire spécialisé dans les services applicatifs distribués par internet [20].

• **UDDI** est une norme d'annuaire de services Web appelée via le protocole SOAP. Pour publier un nouveau service Web, il faut générer un document appelé **Business Registry**, il sera enregistré sur un UDDI Business Registry Node (IBM (International Business Machines), Microsoft et SAP (Server Access Protocol) en hébergent chacun un) [14].

L'annuaire UDDI est accessible soit via un navigateur Web qui dialogue avec une application Web dédiée, interface spécifique à l'annuaire accédé, soit par programme, en utilisant l'API (Application Programming Interface).

L'API comporte deux groupes de fonctions :

• **Les fonctions de recherche :** navigation, recherche et consultation des informations de l'annuaire ;

• **Les fonctions de publication :** publication, création, modification ou suppression des informations de l'annuaire.

❖ **Consultation de l'annuaire :** L'UDDI peut être vu comme un annuaire contenant (voir la **Figure II.7**) :

- **Les pages blanches** comprennent la liste des entreprises ainsi que des informations associées à ces dernières (coordonnées, description de l'entreprise, ...).
- **Les pages jaunes** recensent les services Web de chacune des entreprises sous le standard WSDL.
- **Les pages vertes** fournissent des informations techniques précises sur les services fournis [22].

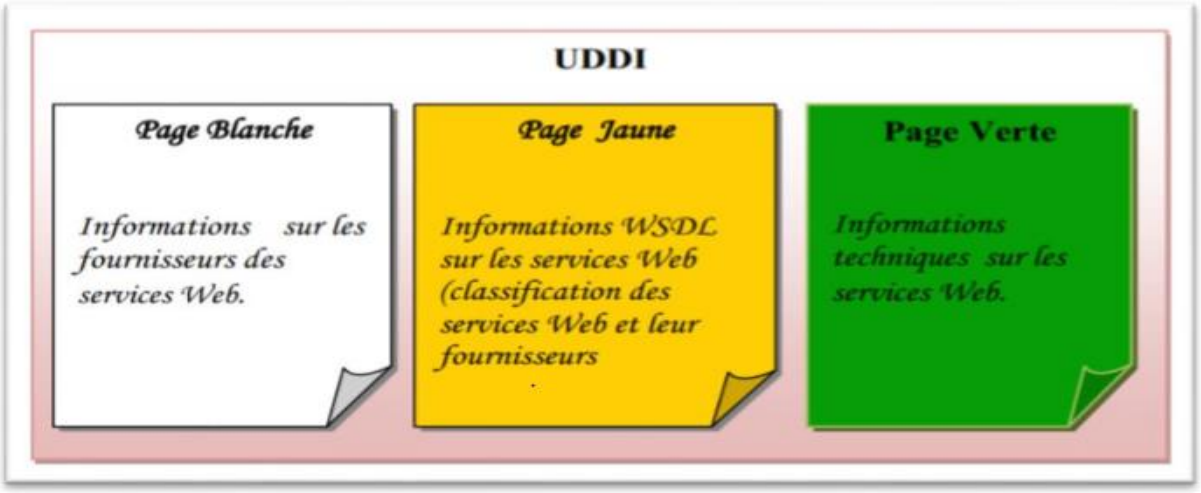


Figure II.7 : Le contenu de l'annuaire UDDI

II.5.4.2.Structure de donnée de l'annuaire UDDI :

L'annuaire UDDI se compose de quatre types de structures de données qui sont : [15]

- **BusinessEntity (entité d'affaires) :** on trouve les informations qui concernent les fournisseurs des Services Web, le nom de l'organisation, ses adresses physiques …, sont en quelque sorte les pages blanches de l'annuaire UDDI.

- **BusinessService (service d'affaires) :** on quelque sorte sont les pages jaunes de l'annuaire UDDI, il s'agit d'informations de description concernant les Services Web (nom du Service Web, sa description, son code).

- **BindingTemplate (information de liaison) :** on quelque sorte les pages vertes de l'annuaire UDDI, offre les informations techniques sur les Services Web et la référence de la description de WSDL.

- **Tmodel (Information techniques) :** a pour rôle de décrire l'ensemble des informations sur le mode d'accès aux Services Web.

La figure suivante **(Figure II.8)** représente la structure de donnée de l'annuaire UDDI :

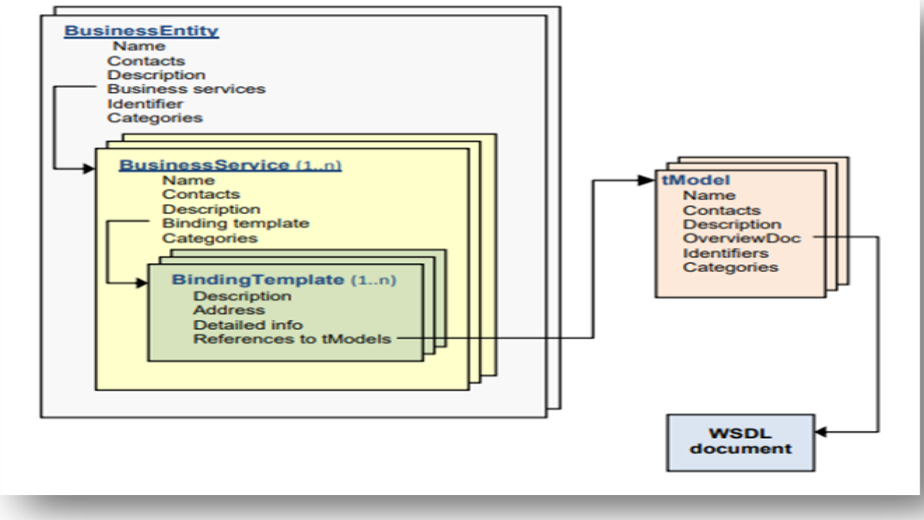


Figure II.8 : Structure de donnée de l'annuaire UDDI

II.6. Avantages et inconvénients des services web :

Les services web ont des avantages et inconvénients :

II.6.1. Les avantages des services web :

- Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- Les Services Web utilisent des standards et protocoles ouverts.
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges.
- Basés sur le protocole HTTP, les Services Web peuvent fonctionner au travers de nombreux pare-feu sans nécessiter des changements sur les règles de filtrage.
- Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les Services Web existants [10].

II.6.2. Les inconvénients des services web :

- Les normes de Services Web dans certains domaines sont actuellement récentes.
- Les Services Web ont de faibles performances par rapport à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM.
- En l'utilisation du protocole HTTP, les Services Web peuvent contourner les mesures de sécurité mises en place à travers des pare-feu [10].

II.7.La composition des services web :

II.7.1 Définition :

• La composition de services est une technique permettant d'assembler des Services Web afin d'atteindre un objectif particulier, par l'intermédiaire de primitives de contrôle (boucle, test, traitement d'exception, etc.) et d'échange (envoi et réception de messages). Les services composants existent au préalable et peuvent ne pas être fournis par la même organisation [11].

• La composition de services est le mécanisme qui permet l'intégration des services dans une application. Le résultat de la composition des services peut être un nouveau service, appelé **service composite** [23].

La figure suivante **(Figure II.9)** représente un exemple de la composition des services web :

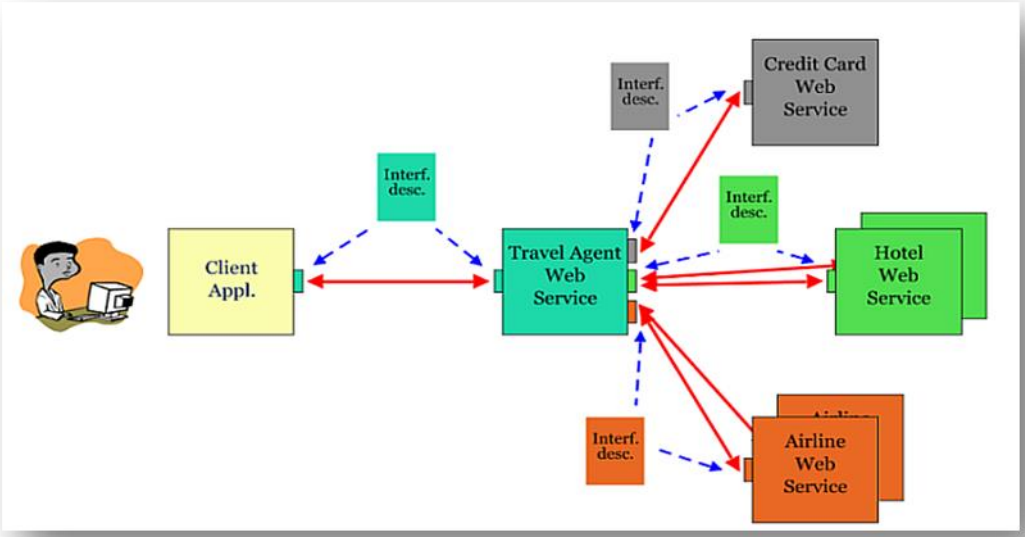


Figure II.9 : Exemple de composition des Services Web

L'utilisateur fournit les informations (voir Figure II.9) nécessaires en entrée et attend un ensemble de concept en sortie, la réponse n'est pas individuelle pas un seul service qui répond mais l'ensemble des Services Web (trois services) qui permet de générer une composition.

II.7.2. Cycle de vie d'une composition de services :

Le cycle de vie d'une composition de services Web reposant à partir de six activités [24] **(voir la Figure II.10) :**

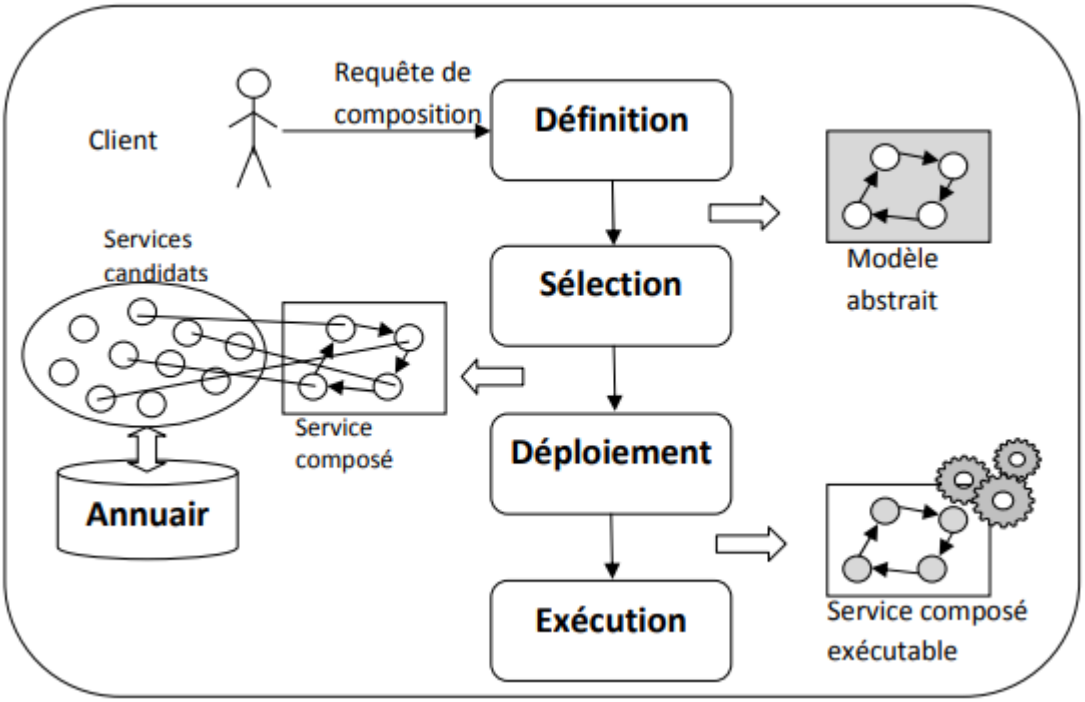


Figure II.10 : Cycle de vie d'une composition de services web

- **L'encapsulation de services natifs** : cette première activité permet de s'assurer que tout service peut être appelé lors d'une composition, indépendamment de son modèle de données, de son format de message, et de son protocole d'interaction.
- **L'établissement d'accord d'externalisation :** cette seconde activité consiste à négocier, établir, et appliquer des obligations contractuelles entre les services.
- **L'assemblage de services composants :** cette activité permet de spécifier, à un haut niveau d'abstraction, l'ensemble des services à composer afin d'atteindre l'objectif attendu. Cet assemblage comporte une phase d'identification des services et de spécification de leurs interactions conformément aux descriptions et aux accords entre services.
- **L'exécution de services composants :** cette activité consiste en l'exécution des spécifications de la composition précédemment définies.
- **Le contrôle de l'exécution de services composites :** la phase de contrôle permet de superviser l'exécution de la composition en vérifiant, par exemple, l'accès aux services,

les changements de statut, les échanges de messages. Ce contrôle permet de détecter des violations de contrats, de mesurer les performances des services appelés et de prédire des exceptions.

- **L'évolutivité des services :** cette dernière phase permet de faire évoluer la composition en modifiant les altérations de l'organisation de services, en utilisant de nouveaux services, ou en prenant en compte les retours de la phase de contrôle.

II.8. Description et fonctionnement :

La composition des services web peut être décrite sous deux angles différents : l'orchestration ou la chorégraphie.

II.8.1. L'orchestration de Service Web :

La composition des services web, en utilisant la technique de l'orchestration, se base sur l'exécution d'un procédé métier central. Ce dernier est lui-même un service web qui prend le contrôle de tous les services web impliqués dans le processus de la composition et coordonne l'exécution des différentes opérations des services composants. Ces derniers n'ont pas de connaissance (et n'ont pas besoin de l'avoir) d´être mêlés dans une composition et de faire partie d'un processus métier. Seulement le coordinateur de l'orchestration a besoin de cette connaissance.

Le principe de l'orchestration est illustré par la **(Figure II.11)** [11]

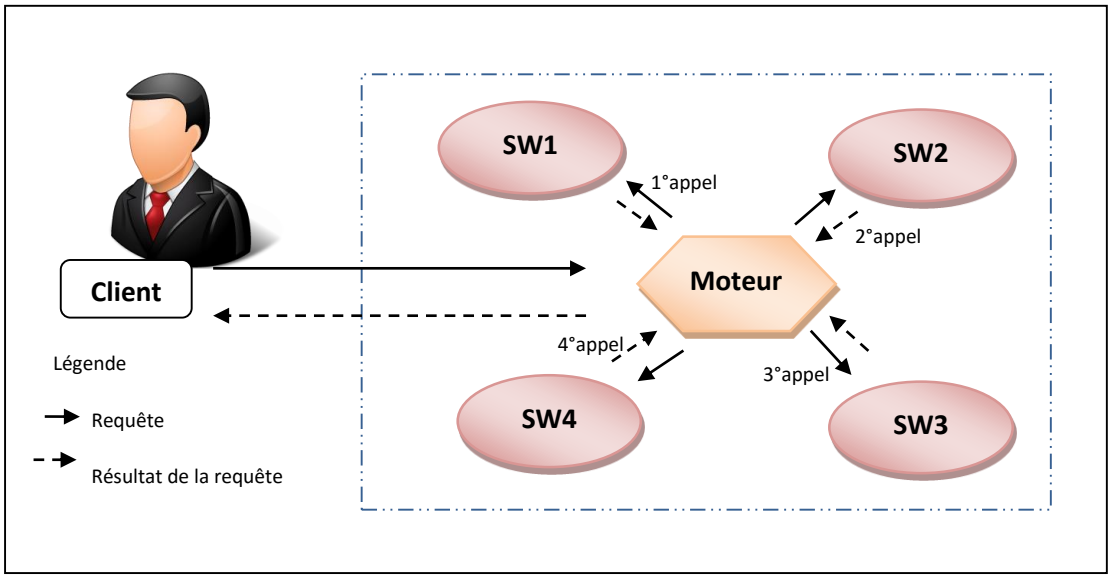


Figure II.11 : principe de l'orchestration des services web.

II.8.2. La chorégraphie de Service Web :

Contrairement à l'orchestration, la chorégraphie offre une vision décentralisée de la composition (elle ne repose pas sur un procédé central pour gérer la composition).

Chaque service web mêlé dans la chorégraphie connaît exactement quand ses opérations doivent être exécutées et avec qui l'interaction doit avoir lieu. Le principe de la chorégraphie est illustré par **(la Figure II.12)** : [11]

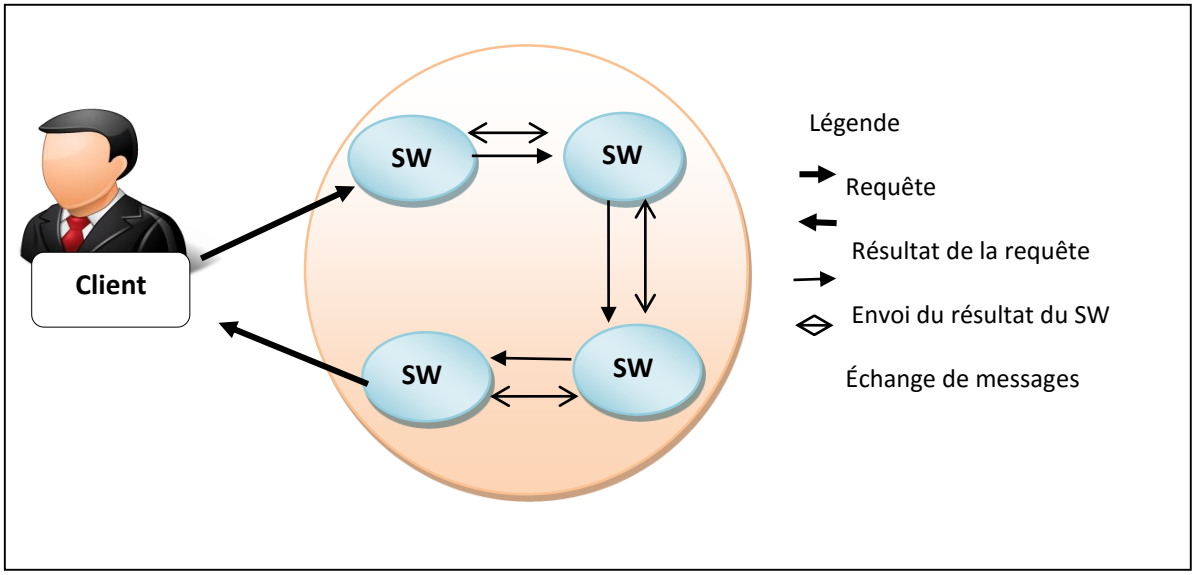


Figure II.12 : Principe de la chorégraphie des Services Web.

II.9. Type de Composition des services web :

II.9.1.La composition Statique de services web :

La composition statique a lieu au moment de la conception d'une application. Ainsi, les composants concernés sont choisis, liés et assemblés avant d'être déployés. Une telle composition est adaptée pour les environnements fermés où les composants n'évoluent pas souvent. Ce type de composition engendre des applications peu flexibles, parfois inappropriées avec les exigences des clients. [11]

II.9.2. La composition Dynamique de service web :

La composition dynamique a lieu au moment de l'exécution. Elle permet de créer de manière autonome des services complexes en combinant des composants à la volée en fonction

des demandes de l'utilisateur et du contexte. Elle évolue dans des environnements flexibles et ouverts où la sélection et la combinaison des composants sont effectuées à la demande. [11]

La technologie de la composition dynamique est généralement contestée par :

- Le grand nombre de services qui deviennent disponibles chaque jour.
- La nature volatile de services web (par exemple : ils peuvent disparaître, être modifiés ou être temporairement indisponibles).
- Le nombre sans cesse croissant de fournisseurs de service.

II.10.Langages de la composition des services web :

De nombreux langages ont vu le jour pour tenter de solutionner le problème de la composition des services web. Ces langages décrivent les interactions entre différents fournisseurs de services web et leurs clients. [11]

II.10.1.BPEL4WS (Businiss Process Execution Language For Web Services) :

Le langage BPEL, nommé aussi BPEL4WS ou WS-BPEL, est un standard OASIS d'orchestration de services Web. Ce langage est largement considéré comme étant le plus abouti du marché dans le cadre de la mise en œuvre des architectures orientées services. BPEL définit le processus, l'enchaînement et l'ordonnancement des actions qui seront exécutées par le moteur d'orchestration, agissant comme une machine virtuelle capable d'exécuter le code BPEL.

II.10.2.WS-CDL (Web Services Choreography Description Language) :

WS-CDL, à l'instar des standards de services Web, est basé sur XML. Il complète la description WSDL des services Web afin de décrire les interactions entre les participants (les autres services) de la composition. L'élément Package englobe cette description. La description des interactions du service étant complexe, nous décomposons ici l'exemple de syntaxe en illustrant les éléments que nous jugeons importants.

II.10.3.BPML (Business Process Modeling Language):

C'est un métalangage de modélisation des processus collaboratifs qu'engage l'entreprise avec ses partenaires sur Internet. Il prend en compte aussi les aspects les plus complexes de la gestion et de la coordination de ces processus, en abordant les transactions, la sécurité et la liaison dynamique avec les services Web en cours d'exécution.

II.10.4.WSFL (Web Service Flow Language) :

C'est une approche IBM qui permet d'intégrer la composition de services Web comme un processus métier statique. C'est un langage basé XML permettant de décrire des orchestrations de services Web.

II.11.Les différentes approches de composition des services web :

II.11.1.Approche oriente Workflow :

Un workflow comme une solution permettant l'automatisation d'un processus métier. Un processus se définit comme un enchaînement coordonné d'un ensemble de tâches aboutissant à un résultat bien déterminé. La coordination spécifie le mode de séquencement des tâches ainsi que les données échangées entre les tâches. En d'autres termes, un processus métier peut être vu comme une application. Cette dernière est construite par composition temporelle de tâches, avec éventuellement des dépendances de données entre ces tâches [26].

II.11.2.Approches sémantiques :

Différentes approches ont été proposées pour la résolution du problème de découverte sémantique des services web, toutes ces approches implantent en fait une découverte approximative, car il n'est pas réaliste d'imaginer qu'il y a toujours un service, qui correspond exactement aux besoins spécifiés, ces approches diffèrent par le langage de description de service utilisé (DAML-S, OWL-S, logique de description…etc.)[26].

II.11.3.Approches basées sur les automates et réseaux de Pétri :

Selon [25] Les automates et Les réseaux de Pétri (Pétri-Nets) offrent une sémantique riche, précise et adaptée à la représentation de la composition des services Web.

Un réseau de Pétri est un ensemble de places et de transitions reliées par des arcs. L'aspect dynamique du réseau est décrit par un ensemble de règles. Chaque règle définit :

- les conditions à valider pour qu'une transaction ait lieu.

- les effets de cette transaction.

II.12. Conclusion :

Dans ce chapitre nous avons donné une vue générale sur les architectures orientées services (SAO) et les notions et les concepts concernent les services web, donc nous avons décrit, plus particulièrement les fonctionnements des services web et les technologies qui leur sont associés, et les avantages et les inconvénients des services web tel que le protocole SOAP, l'interface WSDL, et l'annuaire UDDI.

Ensuite nous avons présenté le problème de composition des services avec l'étude de cycle de vie de ce dernier. À la fin nous avons décrit les types de composition et de langage proposé pour l'implémentation de cette composition.

Chapitre III :

Etude conceptuelle

III.1 Introduction :

Au niveau de ce chapitre nous allons présenter notre approche, commençons par la conception de notre plateforme, son architecture conceptuelle de fonctionnement, les modules qui composent la plateforme, l'architecture globale, les utilisateurs qui peuvent utiliser la plateforme.

III.2 Approche proposée :

Notre solution consiste à proposer une plate-forme basée sur des services Web qui délivre des services interinstitutionnels basés sur l'identité numérique des citoyens.

La plateforme offre un environnement qui permet à une institution de construire une partie de l'identité numérique du citoyen en consultant automatiquement les services fournis par d'autres institutions. Ainsi, la plate-forme permet à une institution d'offrir des services à d'autres institutions. Cela est possible en permettant aux institutions de publier leurs services Web et en proposant une interface de recherche des services propose par la plateforme pour utiliser ces services Web en tant que services atomiques ou composites.

Une institution peut accéder, au moyen de services Web, supposés publiés par les autres institutions, aux données personnelles des citoyens. Une partie des identités numériques des citoyens peut être construite par une institution en apportant des données provenant d'autres institutions. Les données des citoyens (profil du citoyen) d'une institution donneuse sont stockées dans un cloud.

III.3. L'architecture conceptuelle de la plateforme :

L'architecture proposée suppose l'accès de plusieurs institutions à notre plateforme. Les données personnelles des citoyens sont stockées dans le cloud (chaque institution peut stocker dans le cloud une partie des informations personnelles identifiables concernant ses citoyens). L'accès à ces données est effectué uniquement par le biais de services Web. Les services Web sont publiés par les institutions et ils sont enregistrés dans un répertoire UDDI. La plate-forme est organisée en cinq modules, module de gestion des utilisateurs, module de recherche de services Web, module de publication de services Web, module de composition de services Web. Nous connectons avec les Clouds pour récupérer les données.

La figure suivante (**Figure III.1**) montre les différents composants et les entités participées à notre plateforme :

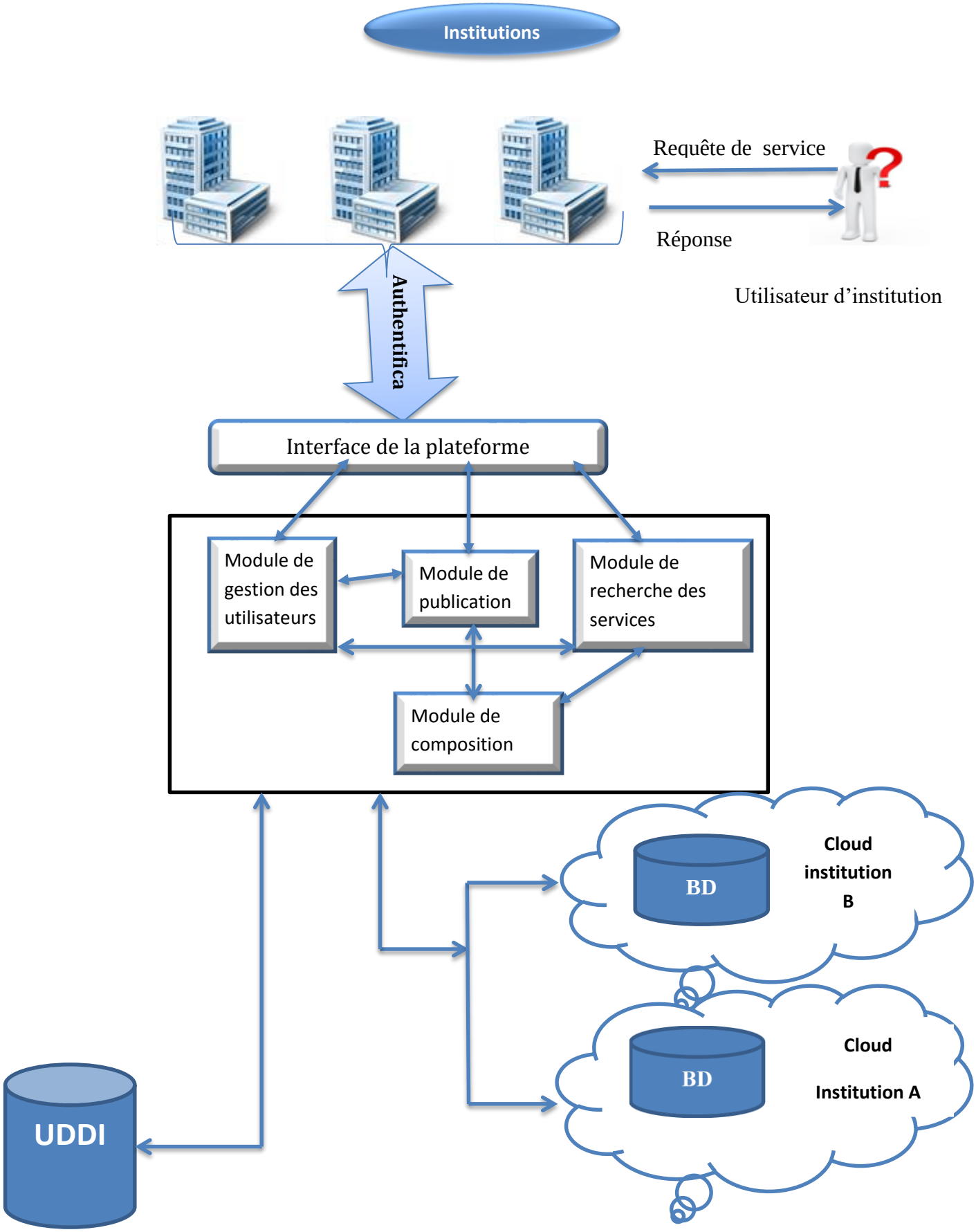


Figure III.1 : L'architecture globale de notre plateforme.

Après l'explication de l'architecture de la plateforme nous présentons le fonctionnement de cette dernière par **(la Figure III.2) :**

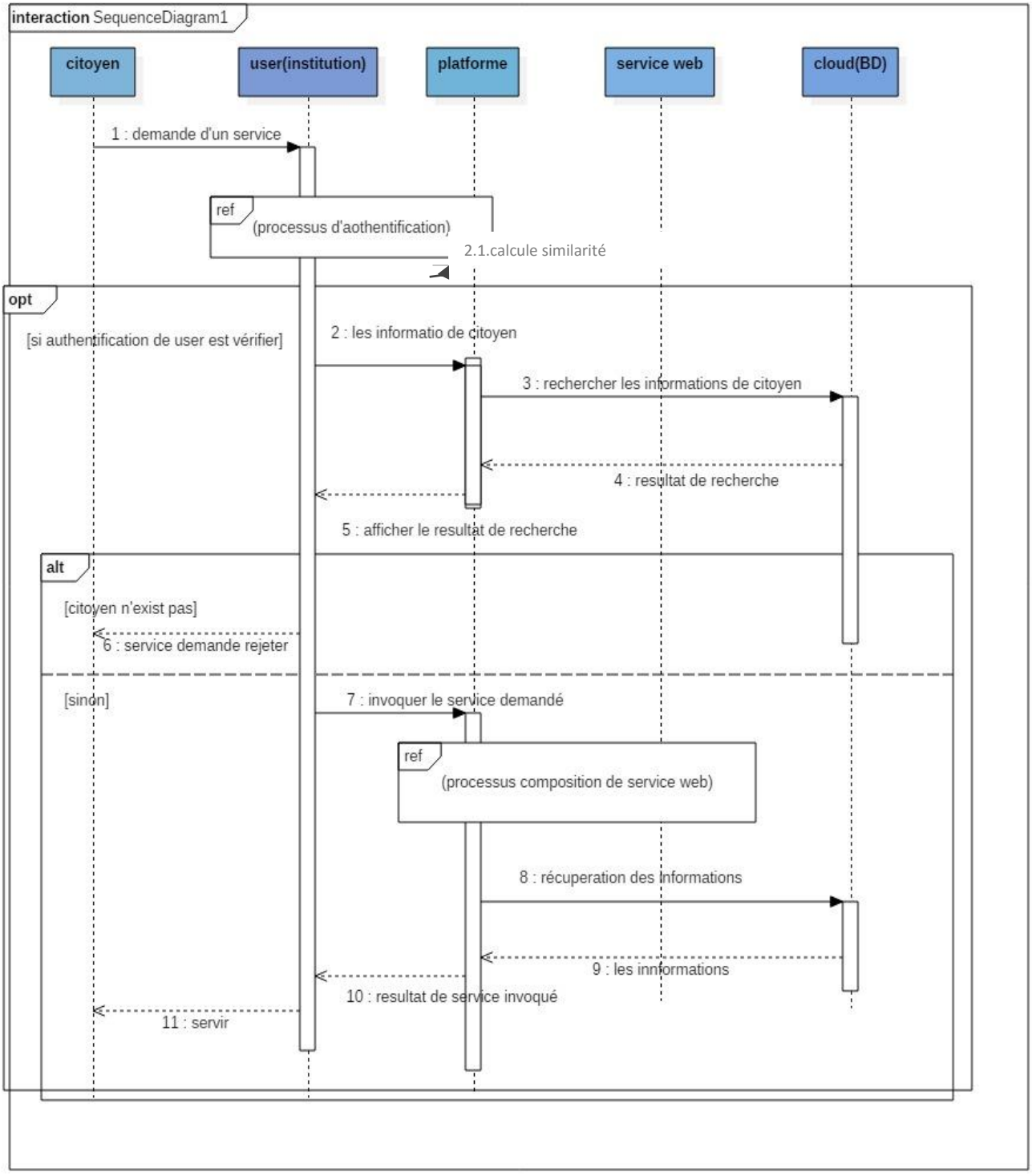


Figure III.2 : Fonctionnement de la plateforme.

Comme illustré dans **(la Figure III.2)** notre plateforme fonctionne comme la suite : dans une première étape le citoyen (candidat) demande un service à une institution. L'institution va contacter notre plateforme qui poura répondre à cette demande. Donc l'institution exécute le processus d'authentification pour accéder à la plateforme. Si l'authentification est valide, l'institution va rédiger une requête et les données de citoyen et la plateforme recherche des informations concernant cette requête dans un cloud (BD). Si les informations de la recherche n'existent pas, l'institution rejet le service demandé, sinon invoque les services demandés. Alors la plateforme va exécuter le processus de composition si les services existes ne sont pas satisfaits, puis récupérés les informations de citoyen du cloud, et affichée le résultat à l'institution, ce dernier serve le citoyen (candidat).

III.4. Description des modules :

La plate-forme est conçue comme un ensemble de modules qui interagissent pour offrir des services aux institutions. La plateforme offre la possibilité de publier, rechercher, composer et accéder à des services Web en fonction du service souhaité (par son nom, sa description et sa date de publication). L'invocation de services permet à l'utilisateur d'institution de récupérer une partie de l'identité numérique du citoyen des autres institutions.

III.4.1. Le module de gestion des utilisateurs :

Ce module géré par l'administrateur de la plateforme, il permet d'organiser l'accès à la plateforme par les utilisateurs des différentes institutions pour publier des services et servir des citoyens. Ce module inclut l'authentification et la gestion des autorisations pour la publication et l'accès aux services web. La figure suivante **(La Figure III.3)** présente un exemple d'accès à la Plateforme :

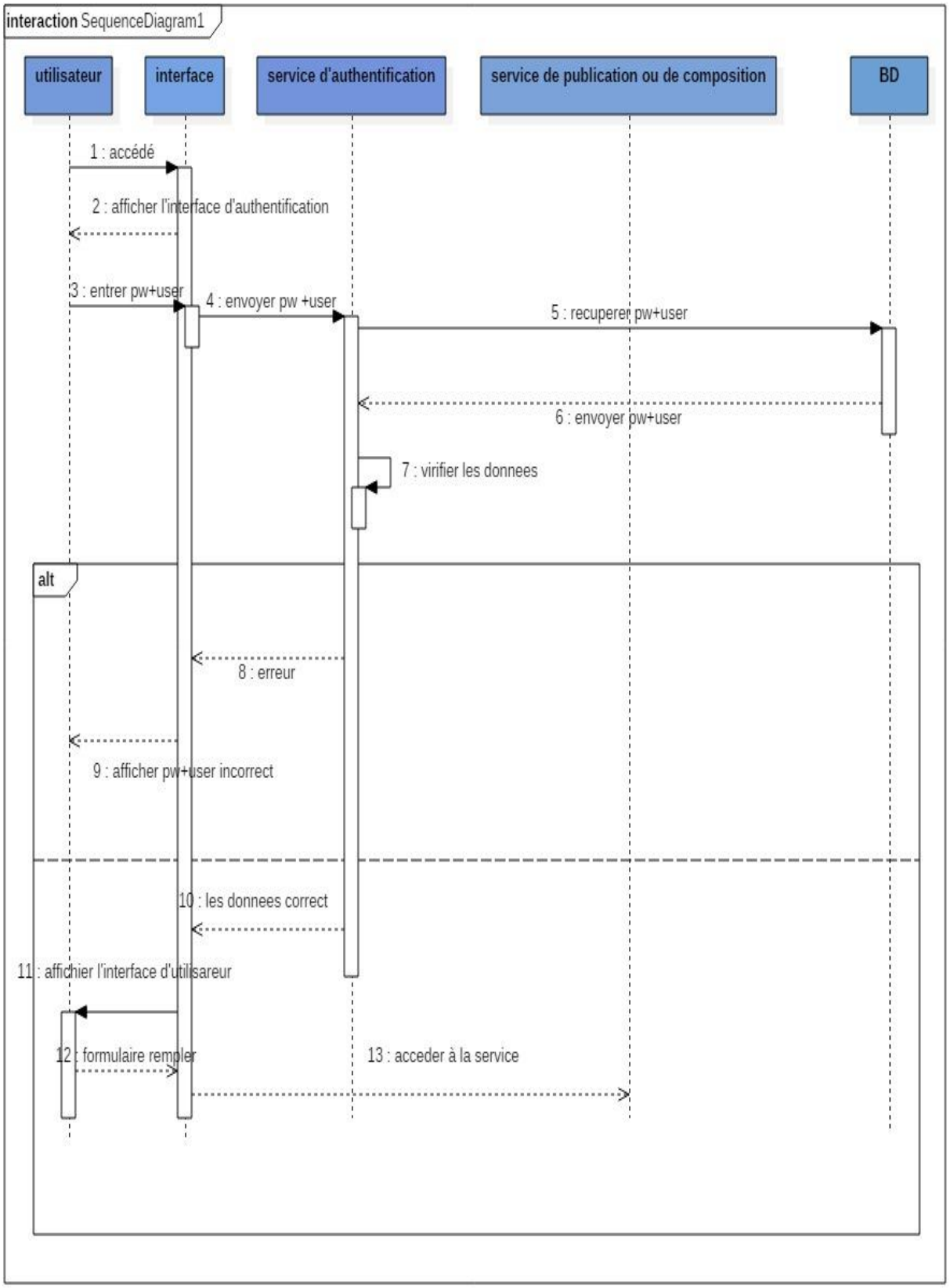


Figure III.3 : module de gestion d'utilisateur.

Les points ci-dessous représentent les étapes pour l'accès à la plateforme :

1. L'utilisateur demande l'accès à la plateforme.
2. La plateforme affiche l'interface de l'authentification.
3. L'utilisateur entre son nom utilisateur et son mot de passe.
4. La plateforme récupère les données d'authentification qui se trouvent à la BDD.
5. La plateforme vérifie la validité de l'authentification.
6. Si l'authentification est incorrecte, la plateforme affiche un message d'erreur.
7. Sinon la plateforme affiche les services autorisés à l'utilisateur authentifié.
8. L'utilisateur sélectionne les services envisagés pour les utiliser.

III.4.2. Le module de publication :

Le module de publication sert d'interface entre la plate-forme et les fournisseurs de services Web (institutions). Il permet à une institution de publier et de mettre à jour des services Web dans le répertoire UDDI. Après l'authentification, l'utilisateur de l'institution (administrateur des services Web) demande la publication ou la mise à jour des services Web. Il peut spécifier et mettre à jour les droits d'accès sur les services Web d'autres institutions. La publication d'un service Web permet de rendre accessibles à des citoyens des informations personnelles (données concernant le service Web publiées et proposées par l'institution), disponibles dans le Cloud. L'accès à ces données (profils de citoyens) est autorisé pour les autres institutions selon les autorisations préconfigurées.

La figure suivante **(Figure III.4)** représente le processus de publication d'un service :

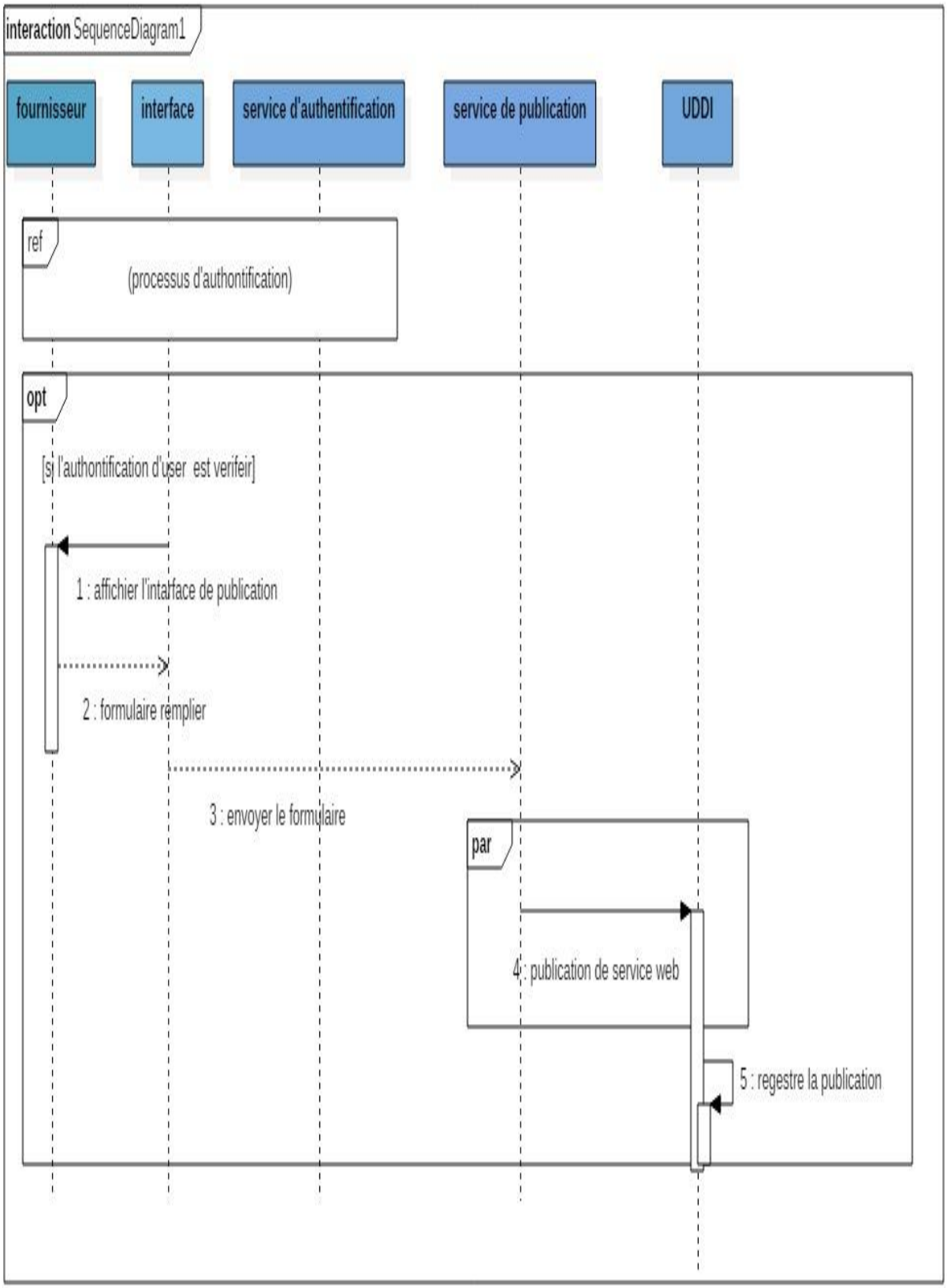


Figure III.4 : module de Publication.

Ces points ci-dessous représentent les étapes pour publier un service Web :

1. Le fournisseur s'authentifie par son nom d'utilisateur et son mot de passe.
2. Le service d'authentification vérifie si le nom 'utilisateur et le mot de passe sont corrects.
3. La plateforme vérifie l'authentification.
4. Si l'authentification est vérifiée : la plateforme affiche l'interface de publication.
5. Le fournisseur remplie le formulaire de publication et publie le service dans.
6. La plateforme invoque le service web de publication.
7. L'annuaire UDDI registre la publication.

III.4.3. Le module de découverte :

Le module de découverte sélectionne les services Web nécessaires pour satisfaire les citoyens et les utilisateurs d'institutions. Le module de découverte est basé sur la description des services Web publiés par les autres institutions et sur les termes clés attribués au point d'accès sélectionné. Il ne renvoie que les services pour lesquels l'établissement dispose des autorisations nécessaires. Ce module peut être utilisé pour la composition des services web.

III.4.4. Le module de composition :

Le module de composition combine de manière dynamique les services Web renvoyés par le module de découverte pour former un service Web composite qui renvoie les données demandées. En fait, une fois que le module de découverte a sélectionné les services pertinents. Comme indiqué dans **(la Figure III.5)** :

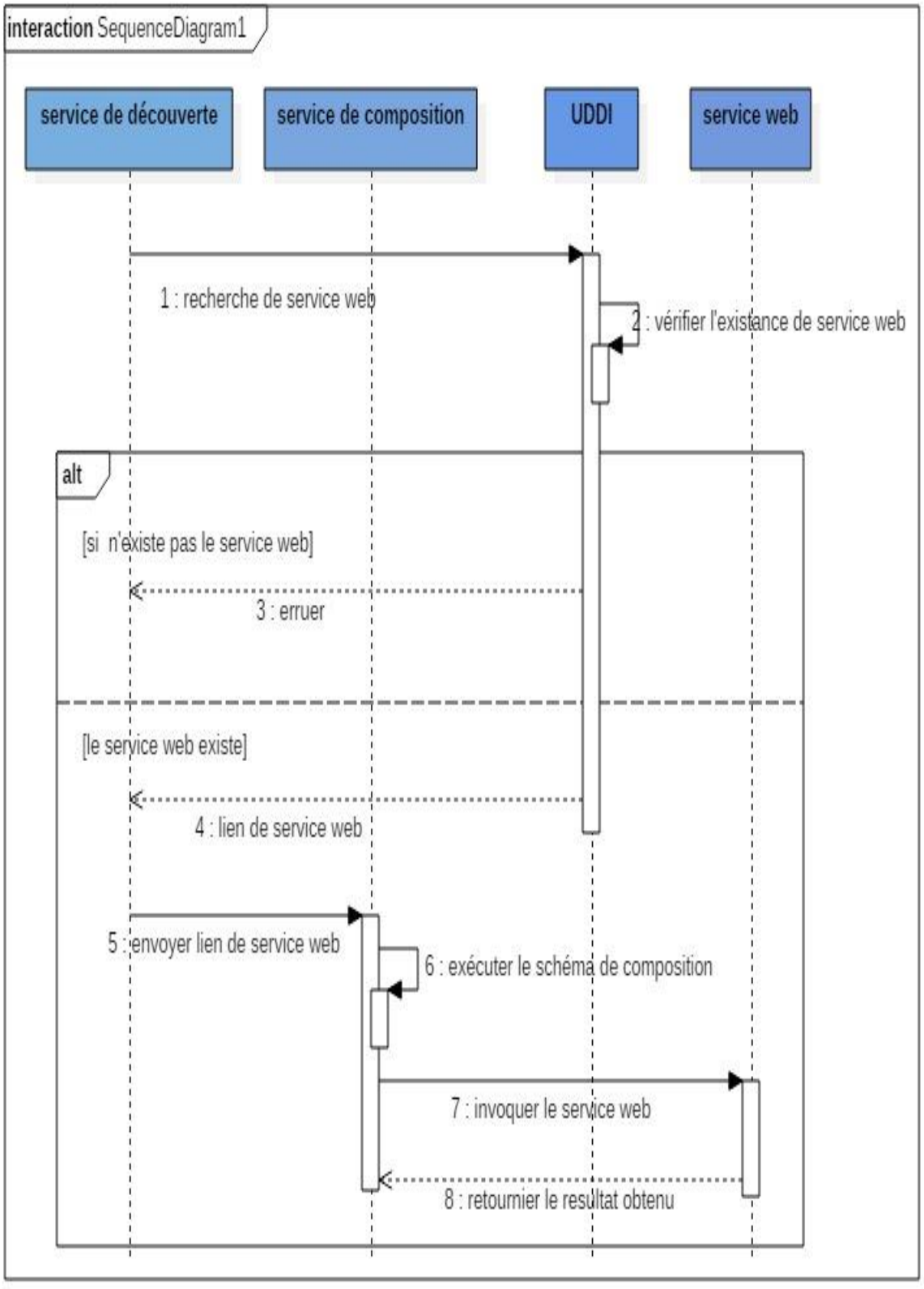


Figure III.5 : module de Composition.

Ces points ci-dessous représentent les étapes de composition des services Web :

1. L'administrateur de la plateforme reçoit des demandes du service composé.
2. Après le processus de l'authentification et l'accès à la plateforme.
3. L'administrateur invoque le service de découvert pour rechercher si les services demandés disponible sur la plateforme.
4. Si les services sont indisponibles, la plateforme affiche un message d'erreur.
5. Sinon si les services sont disponibles, il récupère les liens.
6. Invoque le service composé et retourne le résultat obtenu.

Le module de composition détermine les paramètres d'entrée et de sortie de chaque service sélectionné. Des similitudes entre les services Web sont ensuite calculées pour déterminer le schéma de composition en fonction de l'algorithme **(la Figure III.7)**. Les similitudes entre les paramètres d'entrée et de sortie sont calculées sur la base, où nous supposons que chaque service fait référence à une seule opération et que les liens sémantiques entre les services Web sont utilisés pour calculer les similarités entre les paramètres d'entrée et de sortie, comme indiqué dans **(la Figure III.6)** :

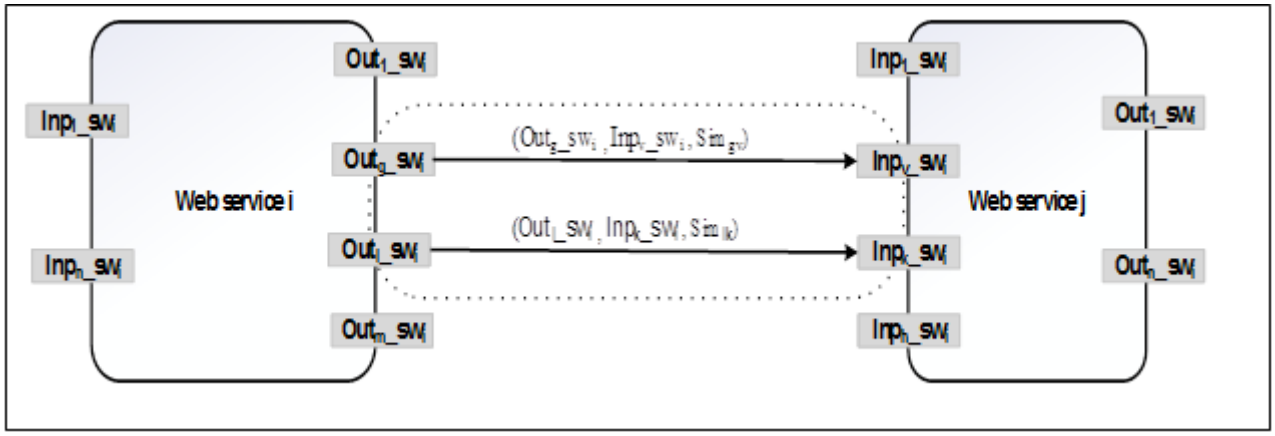


Figure III.6 : Similarités entre les paramètres de service Web d'entrée et de sortie.

L'algorithme utilisé pour construire le schéma de composition prend en entrée un ensemble de services Web sélectionnés (module de découverte) et construit un graphe représentant le schéma de composition. Pour cela, l'algorithme recherche les services Web, qui ne nécessitent que les données fournies dans la requête de l'utilisateur de l'institution. Ces services Web seront injectés dans une liste appelée liste de services Web indépendants **(L_independt).** Le reste des services Web sera injecté dans **une liste (L_middle)**. Ensuite, pour chaque service Web de la liste des

services indépendants. Cette fonction permet de déterminer une étape dans la composition. Sur la base de similitudes entre les services Web **(Figure III.6)**,

La fonction renvoie un service Web et construit une partie du graphe représentant la composition.

Dans l'algorithme principal, chaque service Web traité sera supprimé de la liste des services Web indépendants. Le service Web renvoyé sera supprimé de la liste des services de et inséré dans la liste des services Web indépendants. L'algorithme arrête le traitement des services Web lorsque la liste des services Web indépendants sera vide.

Enfin, l'algorithme élimine les compositions impossibles en supprimant les arcs du graphe entre les services Web où le nombre de paramètres d'entrée nécessaires est inférieur au nombre de paramètres de sortie.

La figure suivante **(Figure III.7)** illustre algorithme de composition :

Algorithme de construction de schéma de composition

Entree:

L_SW:SW_1,…SW_n: La liste de services Web sélectionnés;

L_input_user : $input_user_1$,…$input_user_m$ La liste des entrées saisie par l'utilisateur.

Sortie: G: un graphe représentant le schéma de composition.

Debut

L_independant← ∅;

L_middle ← ∅;

pour (Swi dans L_SW) **faire**

pour ($input_j$ dans Swi) **faire**

Calcule similarite(L_input_user, inputj**)**

Si (similarit<0.50**) alors**

L_middle ←SWi;

break;

Fisi

Fin

L_independant←Swi;

G←(swi);

Fin

pour (SWi dans L_middle**) faire**

pour (SWj dans L_independant**) faire**

Calcule similarite(input(Swi), output (Swj)**)**

Si (similarite>=0.50**) alors**

Deplacer SWi de L_middle à L_independant;

G←(swj,swi);

Finsi

Fin

Fin

Figure III.7 : Algorithme de composition

Calcule similarite:[33] pour calculer la similarité nous avons utilisez le coefficient de Dice est une mesure de similarité, ce coefficient est utilisé en statistique pour déterminer la similarité entre deux échantillons

Présentation du coefficient de Dice :

Le coefficient est toujours compris entre 0 et 1. Il se définit comme le double de l'intersection de deux lots (échantillons de valeurs) divisé par l'union de ces deux lots. La formule ci-dessous résume la méthode de calcul :

$$s = \frac{2 * |X \cap Y|}{|X| + |Y|}$$

Dans cette formule, X et Y sont des lots. Pour calculer le coefficient de Dice de deux chaines, il est possible de définir X et Y comme l'ensemble des bi-grammes de chacune des chaines respectives. Un bi-gramme étant la réunion de deux lettres dans un mot.

III.5. Les entités de la plateforme :

La figure suivante **(La figure III.8)** présente le diagramme de classes de notre plateforme

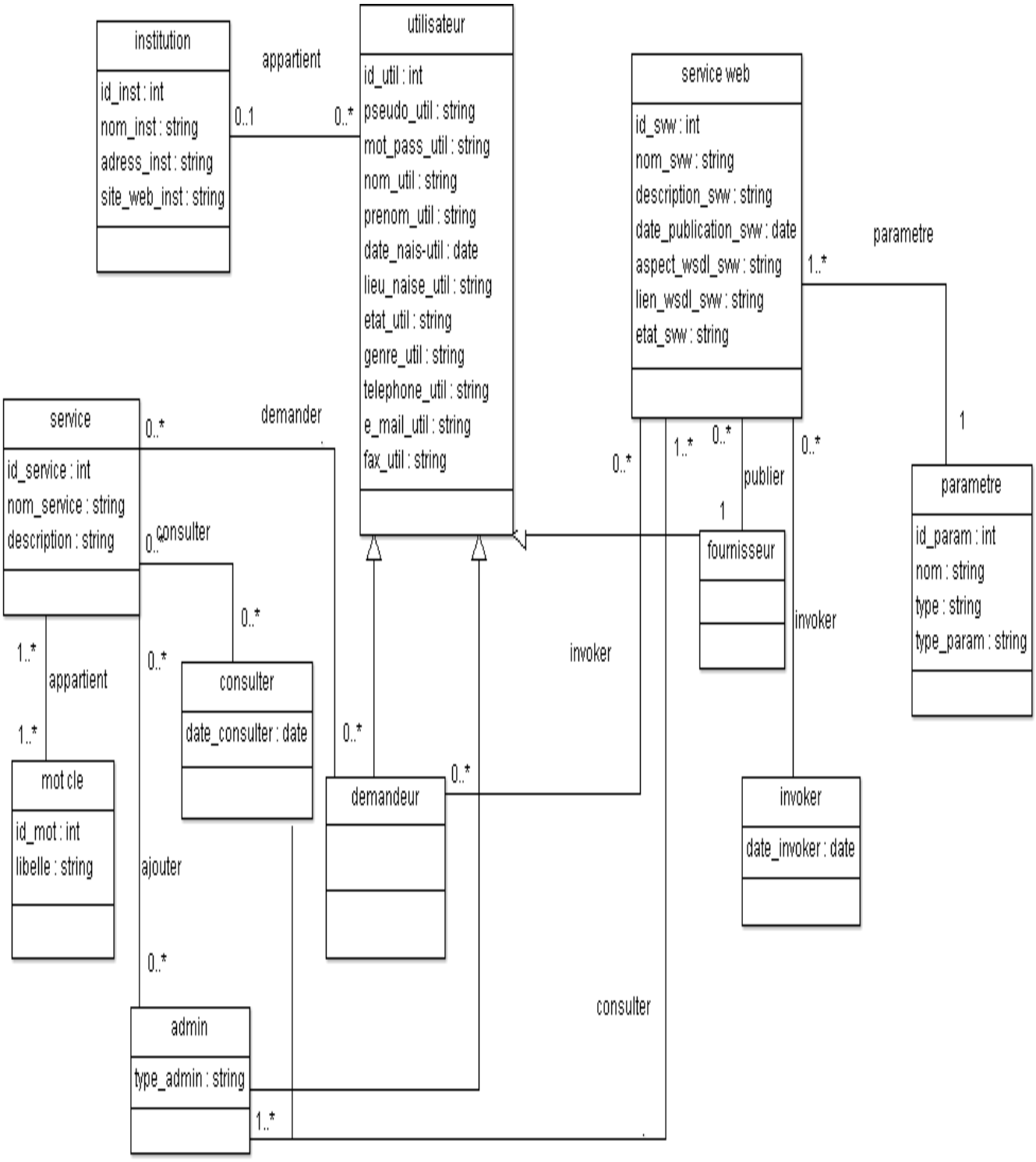


Figure III.9 : Diagramme de classe de la plateforme.

III.6. Les types d'utilisateur :

Nous avons défini quatre types d'utilisateurs qu'ils peuvent utiliser la plateforme :

III.6.1. Administrateur de la plateforme :

C'est la personne qui a le droit d'accès à tous les ressources de la plateforme.
Les tâches effectuées par l'administrateur sont :

- Ajouter ou mis a jours une institution.
- Ajouter ou mis a jours un administrateur d'institution.
- Rechercher des services web.

III.6.2. Administrateur d'institution :

Est un utilisateur moins prioritaire qu'un administrateur de plateforme. Pour utiliser la plateforme doit s'authentifier afin que la plateforme déterminée ses droits, les tâches effectuées par ce type d'utilisateur sont :

- Ajouter ou mis a jours un utilisateur de l'institution de cet administrateur.
- Consulter les services web publié de cette institution.
- Rechercher des services web.

III.6.3. Fournisseur :

Est un utilisateur appartient à une institution. Pour utiliser la plateforme doit S'authentifier afin que la plateforme déterminer ses droits. Il accède à la plateforme pour publier ou modifier un service.

III.6.4. Demandeur :

Est un utilisateur appartient à une institution. Pour utiliser la plateforme doit S'authentifier afin que la plateforme déterminer ses droits. Il accède à la plateforme pour demander un service.

III.7. Conclusion :

Dans ce chapitre nous avons représenté l'approche proposée de la plateforme, comme nous avons présenté l'architecture conceptuelle, la description des modules et l'interaction entre ces modules.

Chapitre IV : Implémentation et étude de cas

IV.1. Introduction :

Dans ce chapitre nous allons présenter l'implémentation et la réalisation de notre plateforme, mais avant de commencer l'implémentation de notre application, nous allons tout d'abord spécifier les outils utilisés qui nous ont semblés être un bon choix vu les avantages qu'ils offrent.

IV.2. Les outils et les langages utilisés en développement :

IV.2.1. Langage JAVA :

Nous avons opté le langage JAVA pour la programmation et cela pour divers raisons : [25]

- **JAVA** est un langage de programmation orienté Object mis au point en 1991 par la firme SUN Microsystems (souvent juste appelée "SUN"), et reprise depuis par la société Oracle.
- **JAVA** est portable, c'est un langage multiplateforme. Il peut être utilisé sous Windows, Macintosh et sur d'autres plates-formes.
- **JAVA** a été développée dans un souci de sécurité maximale, donc on peut dire que **JAVA** est langage à haute sécurité.
- **JAVA** possède une riche bibliothèque de classe comprenant des fonctions diverses.

IV.2.2.Environnement de développement NetBeans :

Nous avons chosiez NetBeans comme un environnement de développement, il possède de nombreux avantage [26] :

- Un environnement de développement intégré (IDE).
- **NetBeans IDE** est un logiciel de développement open source sur JAVA.
- **NetBeans IDE** fonctionne sur n'importe qu'elle système d'exploitation.
- **NetBeans IDE** nous permet de créer des applications de bureau, des applications mobiles et web pour téléphones intelligents et ordinateurs de poche, sans avoir à changer la programmation.
- **NetBeans IDE** support de différents langages de programmation exemple JAVA, PHP, C++, Python et d'autre , accès et gestion de bases de données, serveurs WEB, ressources partagées.

IV.2.3. Les JSP (Java Server Pages) :

Définit par [27] :

- ✓ **JSP** c'est une technologie JAVA qui permet de générer des pages web dynamiques.

  JSP permet :
- ✓ Une page **JSP** contient des balises standards, comme du HTML, comme toute page web normale.
- ✓ D'introduire du code JAVA dans des tags prédéfinis à l'intérieur d'une page HTML.
- ✓ Le mélange la puissance de JAVA côté serveur et la facilité de mise en page d'HTML côté client.

IV.2.4.Glass Fish Serveur :

C'est un serveur d'application Open Source JAVA multiplateforme crée par Sun Microsystems en 2005. [28]

IV.2.5. MY SQL:

Est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde [29]

IV.2.6. JDOM :

JDOM, est une bibliothèque open source pour manipulation des fichiers XML en Java. Elle intègre DOM et SAX, et supporte XPath et XSLT. Elle utilise des analyses syntaxiques externes pour construire les documents. [30]

IV.2.7. HTM L (L'HyperText Markup Language) :

C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est

souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS). [31]

IV.2.8.Les CSS (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.).

Les feuilles de styles ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle. [32]

IV.3.La mise en œuvre de la plateforme :

Dans cette partie nous allons présenter le fonctionnement de la plateforme E-gouvernement pratiquement.

La figure suivante **(Figure IV.1)** représente le schéma global de la plateforme :

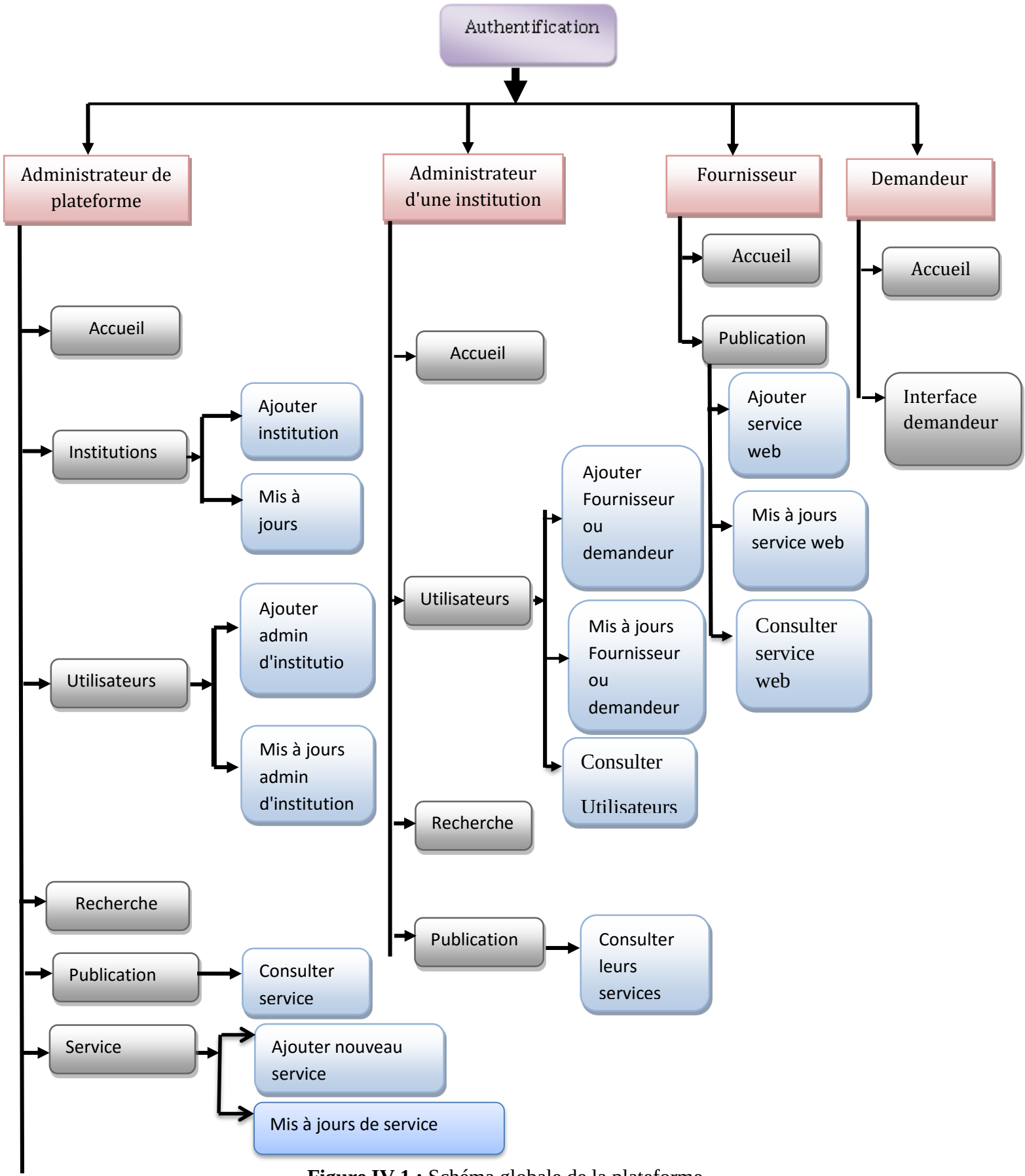


Figure IV.1 : Schéma globale de la plateforme

(Figure IV.2) présente l'organigramme d'utilisation de la plateforme :

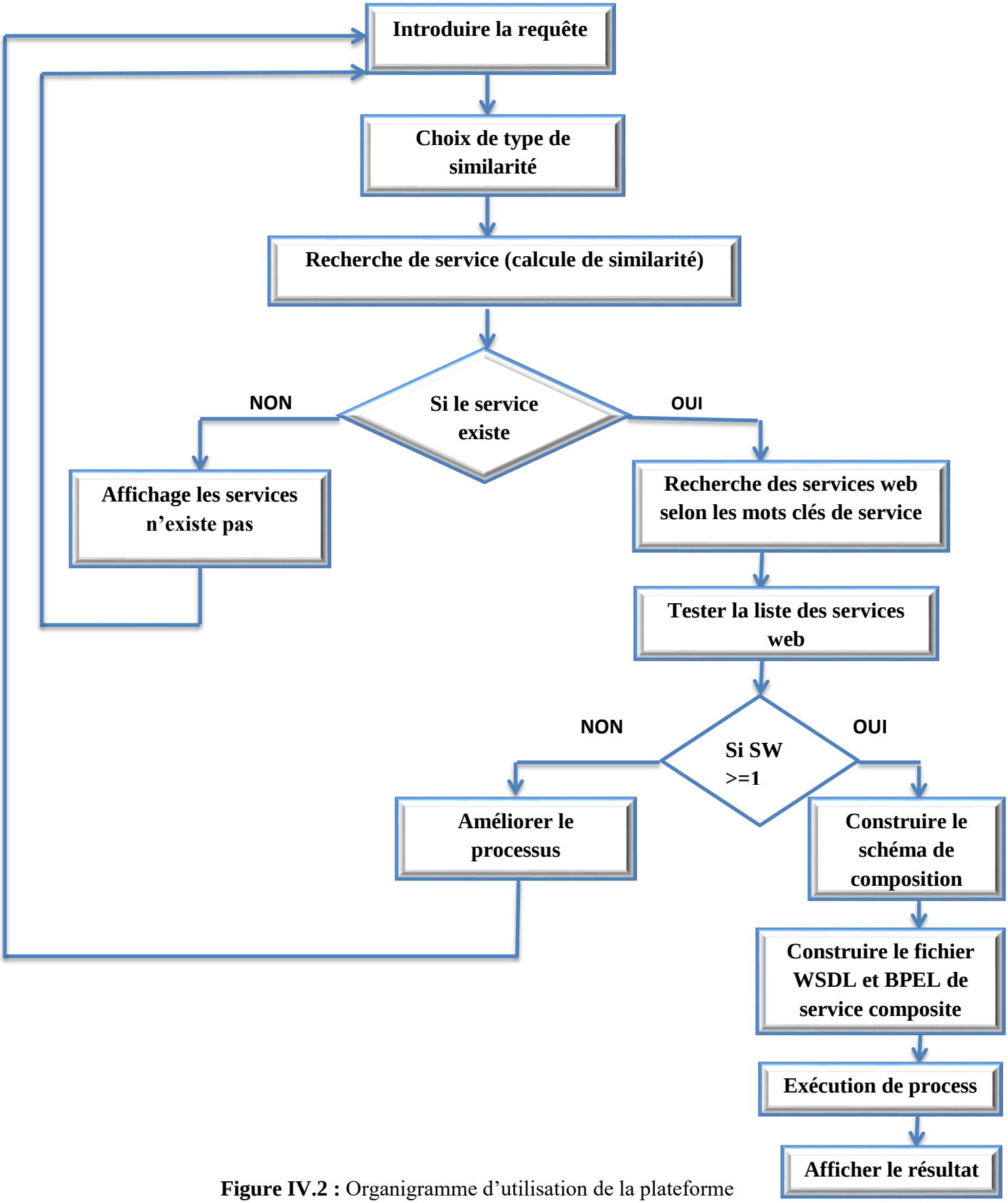


Figure IV.2 : Organigramme d'utilisation de la plateforme

IV.4. Etude de cas d'utilisation :

Pour illustrer la solution proposée dans ce travail, nous proposons une étude qu'une institution qui souhaite précéder à un recrutement d'une personne. Pour le processus de recrutement d'un personnel, nous devons récupérer les informations du candidat (état civil, diplômes, casier judiciaire, expérience de travail, etc…).Ces informations peuvent être transmises par d'autres institutions (ministère de l'Intérieur, ministère de l'Éducation, ministère de la Justice,…), chaque institution est un moyen de Service Web.

IV.4.1.Description des services web utilisés :

Nous allons donner dans cette partie la description complète de chaque service Web et qu'elles sont les paramètres pour l'entrée et sortie.

- **Service «Etat civil » :**

Service basé sur numéro d'identification national qui renvoie l'état civil d'un citoyen donné (prénom, nom de famille, date de naissance, sexe, âge, situation familiale)

(Voir la Figure IV.3) :

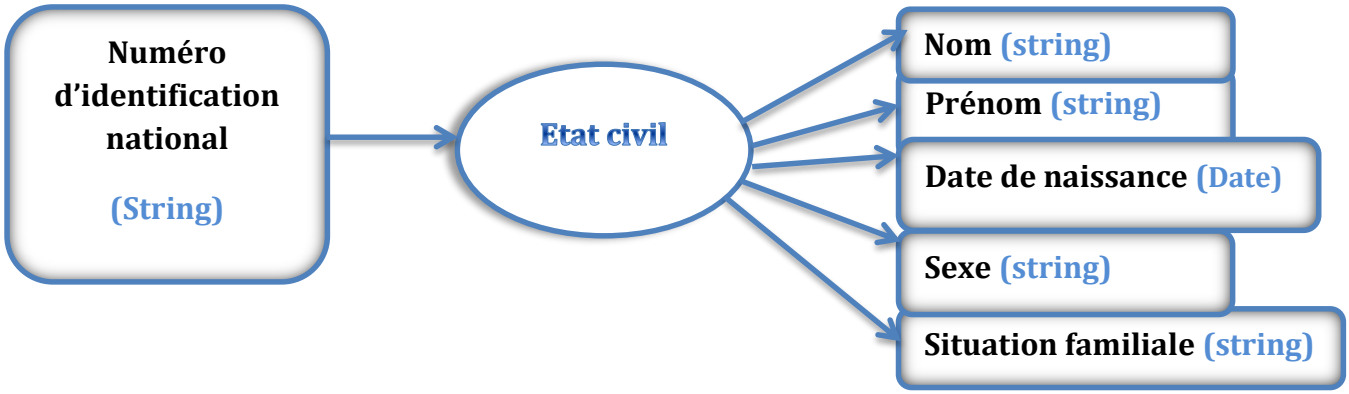


Figure IV.3 : service Etat civil

- **Service «Diplôme universitaire» :**

Ce service permet de restituer le diplôme universitaire obtenu par un citoyen donné. Le service suit les recherches en se basant sur le nom de famille, le prénom et la date de naissance du citoyen **(voir la Figure IV.4)** :

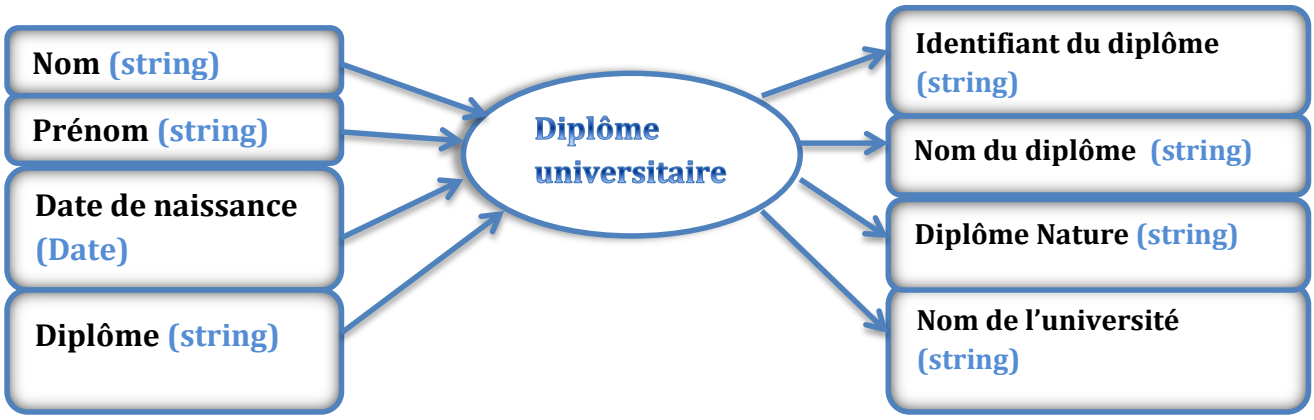


Figure IV.4 : Service Diplôme universitaire

❖ **Service «Expérience de travail» :**

Ce service permet de restituer toutes les expériences professionnelles d'un citoyen donné. Le service consiste en des recherches basées sur l'identifiant du diplôme. La recherche est faite pour un diplôme et un travail donnés **(voir la Figure IV.5)** :



Figure IV.5 : Service Expérience de travail

❖ **Service «Santé» :**

Ce service permet de renvoyer les informations médicales d'un citoyen donné. Le service consiste en des recherches basées sur le nom de famille, le prénom et la date de naissance

(Voire la Figure IV.6) :

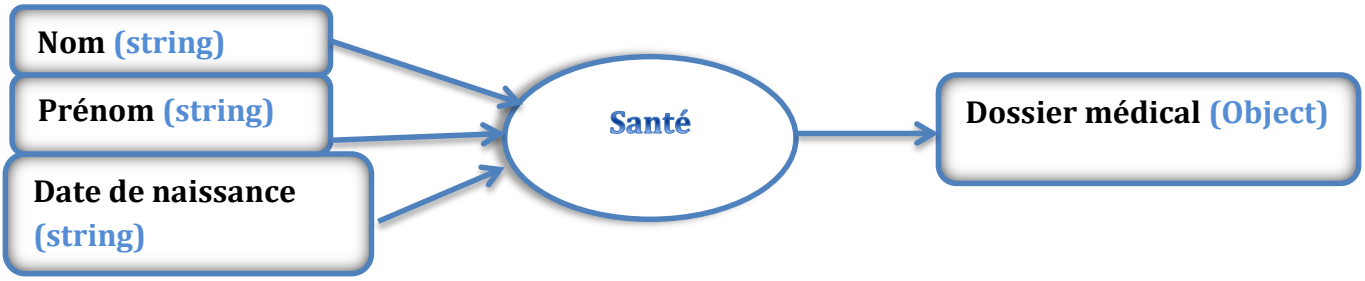


Figure IV.6 : Service Santé

❖ **Service «Casier judiciaire» :**

Ce service restitue, selon le nom et prénom du citoyen et sa date de naissance, un casier judiciaire selon le type de casier judiciaire demandé **(voir la Figure IV.7)** :

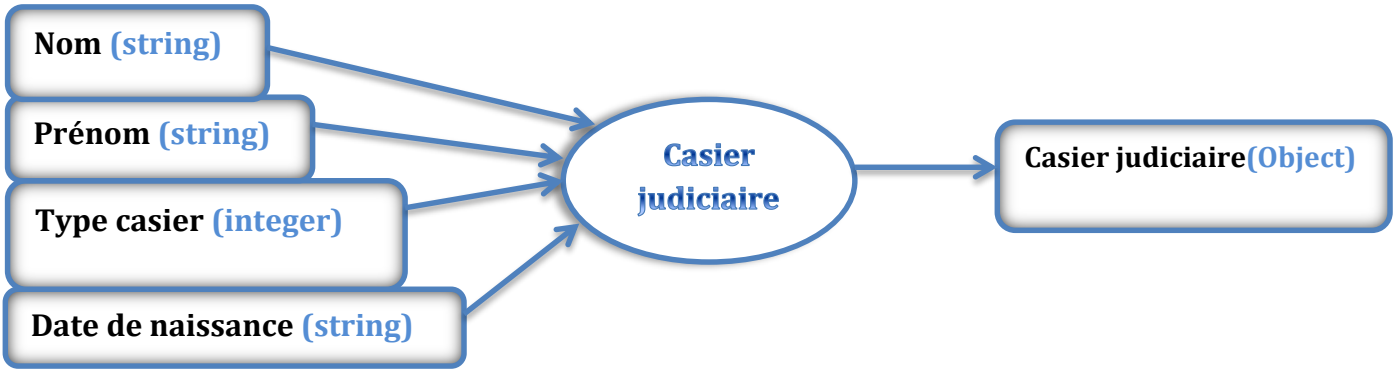


Figure IV.7 : Service Casier judiciaire

IV.4.2.Scenario de la composition :

Pour récupérer les informations d'un candidat donné, l'utilisateur de l'institution doit introduire une requête et les données du candidat afin de spécifier le service qui fournit les données nécessaires. Par exemple, la requête peut ressembler à « **recrutement d'un master** » et les données du candidat peut ressembler à «**numéro national, 1, diplôme, master, type casier judiciaire, 2**».

Le module de sélection détermine les services Web pertinents en choisissant et calculant la similarité (sémantique ou algorithme de Dice) entre la description de services et la requête introduite par l'utilisateur.

Ensuite nous calculant la similarité entre les mots clés de service sélectionne aves les services web uniquement présentant une similarité supérieure à 0,5.

Pour notre exemple, les services sélectionnés sont les suivants : <<État civil, diplôme universitaire, travail, santé et casier. >>

Une fois la sélection effectuée, le module de composition effectue une composition dynamique pour construire un service Web composite. Premièrement, il détermine tous les services qui nécessitent uniquement des entrées formulées dans la demande de l'utilisateur de

l'institution. Dans notre cas, le seul service Web dont toutes les entrées ont été formulées dans la demande de l'utilisateur de l'institution est le service Web "Etat civil".

(La figure IV.8) illustre une vue globale du système et les interactions entre les différentes tâches.

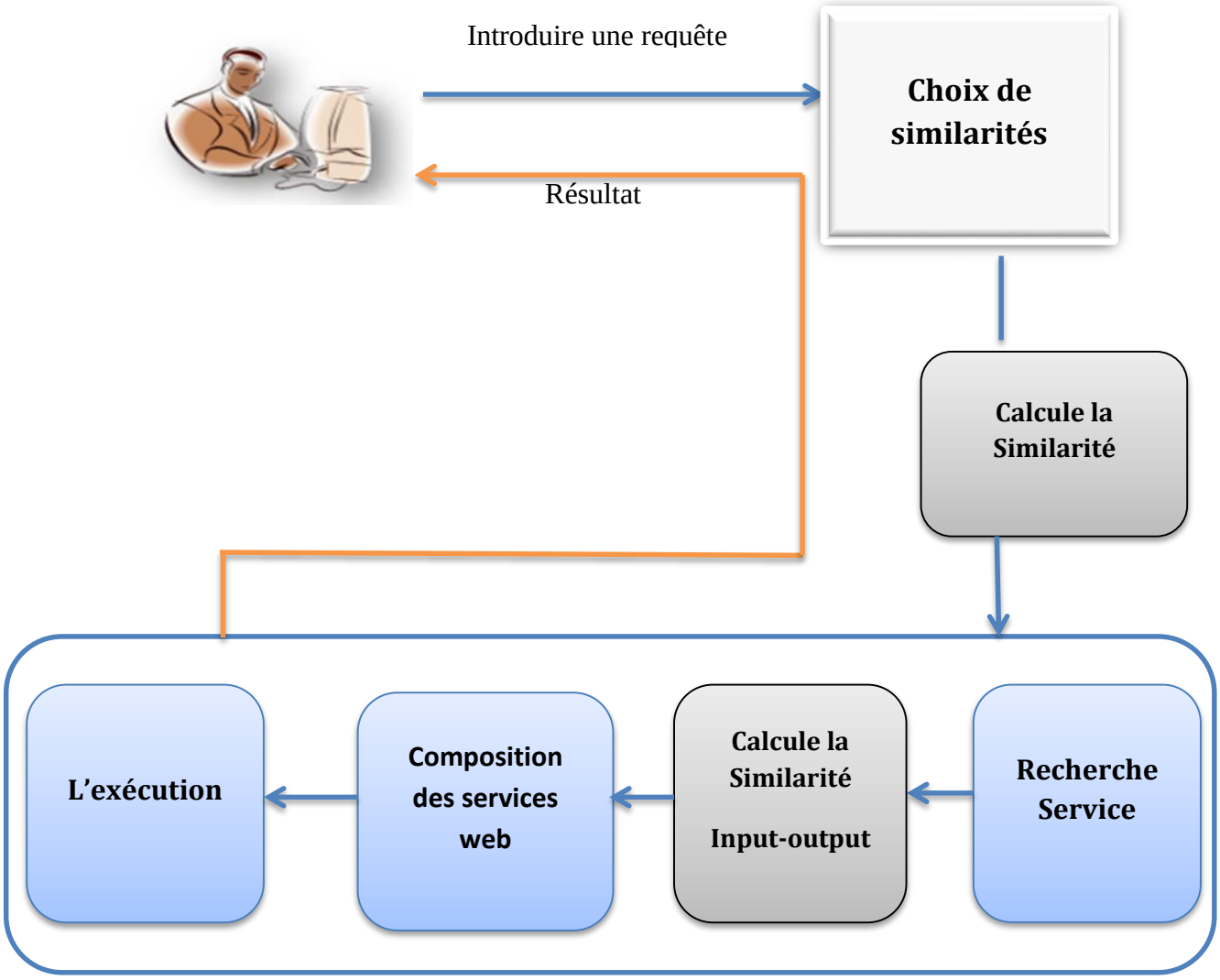


Figure IV.8 : Schéma explicatif du système.

IV.4.3.Résultat et discussion :

En mathématiques et en informatique, une mesure permettant de comparer des documents textuels, consiste à comparer des chaines de caractères. Une telle mesure sur les chaines de caractères fournit une valeur obtenue algorithmiquement.

Nous présentons certains exemples utilisant deux approches permettant de comparer des textes, similarité syntaxique parmi de telles mesures de similarité, citons par exemple, le coefficient de Dice, l'indice de Jaccard, la distance euclidienne, le cosinus, ... et similarité syntaxique

Similarité syntaxique : [33]

Une mesure de similarité syntaxique permet de comparer des documents textuels en se basant sur les chaines de caractères qui les composent. Par exemple, les chaînes de caractères "voiture" et "voiturier" peuvent être considérées comme très proches, alors que "voiture" et "automobile" pourront être considérées comme très différentes. Dans ce chapitre, nous présentons la mesure de similarité syntaxique utilisant Dice pour faire le calcule de similarité pour faire la composition dynamique de service web.

Similarité sémantique :

La similarité sémantique consiste à quantifier le degré de ressemblance de deux concepts issus d'une base de connaissances. Les plus connus sont ceux qui utilisent une ontologie comme WordNet.

Nous avons utilisé WordNet qui est une base de données lexicographique. Le choix de WordNet été cause de diverses raisons :

- C'est la base la plus riche et la plus générale qui contient tous les domaines,
- Il utilise la langue anglaise qui est la langue la plus utilisée dans le monde.

Les requêtes testées:

Le tableau (VI.3) suivant présente la similarité de 37 requêtes on utilisant la similarité syntaxique et sémantique. On calcule la similarité de requête avec l'intitulé de service, dans notre cas nous allons faire calcule de similarité avec le service «**recrute master enginner** », ou la similarité sera entre 0 et 1.

N°	Requête	Sémantique	Syntaxique
01	Master recrutement request	0.99	0.62
02	Executive recruitement	0.73	0.44
03	IT recrutement	0.22	0.39
04	Job application	0.98	0
05	Master work request	0.99	0.39
06	Engagement master	0.98	0.63
07	Vacation enginner	0.98	0.50
08	Recruitement application	0.41	0.25
09	Search job	0.43	0
10	Post a job	0.89	0.11
11	Recrutement process	0.79	0.25
12	Employee to organization	0.42	0.07
13	Managment development	0.66	0.12
14	Managment development master	0.59	0.42
15	Age discrimination	0.56	0.05
16	Assignment request	0.60	0.20
17	Business craft	0.65	0.24
18	Engagement in activity	0.67	0.21
20	Staff recrutement	0.12	0.36
21	Recruitment administrator	0.51	0.43
22	Employment, recruitment and qualifications.	0.24	0.29
23	Executive recruitment companies	0.32	0.29
24	Enginner recrutement request	0.08	0.52
25	Commissioning works	0.37	0.10
26	Finding human resources	0.27	0.23
27	Demand work	0.32	0.10
28	Demand employment	0.25	0.16
29	Recrutement demand for enginners	0.99	0.41
30	Enginnering jobs demand	0.13	0.37
31	Demand job engineering	0.21	0.37
32	Employment agency	0.24	0.14
33	Demande appointment master	0.99	0.48
34	Undertaking jobs demand	0.15	0.18
35	High demand recrutment	0.24	0.31
36	Demand recruitment services	0.22	0.37
37	Demand recruiting	0.15	0.38

Tableau VI.3 : La similarité de 37 requêtes.

Parmi les requêtes tester, il a trois (03) requêtes calculer avec similarité syntaxique et similarité sémantique leurs similarité et SIM>=0.50 **(le tableau VI.4)** présente ces requêtes avec leurs similarité.

N°	Requête	Sémantique	Syntaxique
01	Master recrutement request	0.99	0.62
02	Engagement master	0.98	0.63
03	Vacation enginner	0.98	0.50

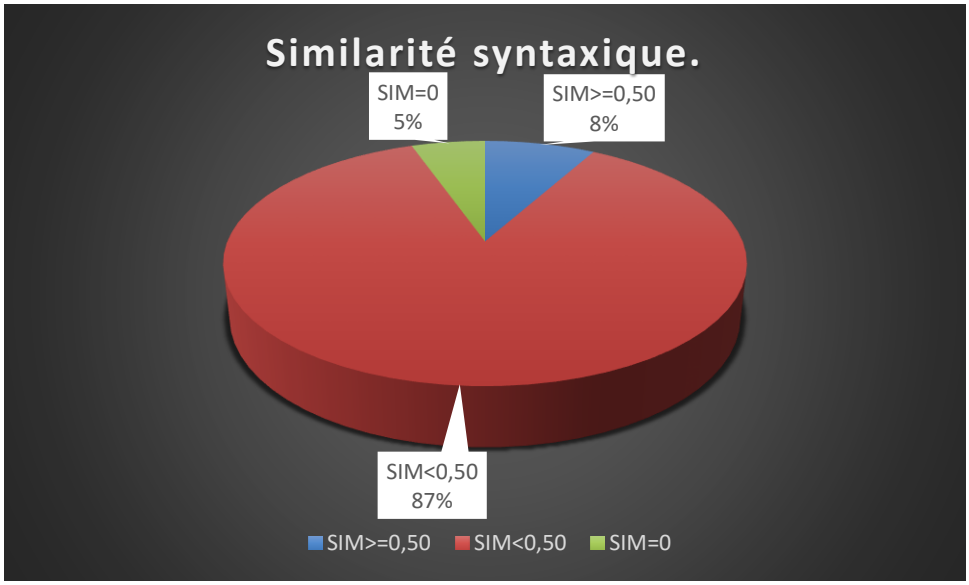
Tableau VI.4 : Requêtes SIM>=0.50

(Le tableau VI.5) suivant présente le pourcentage des requêtes basant sur les deux types de similarité utilisées.

	SIM >=0.50	SIM < 0.50	SIM =0	%>=0.50	% < 0.50	% =0
Similarité syntaxique	3	32	2	**0,08**	**0,86**	**0,05**
Similarité sémantique	17	20	0	**0,45**	**0,54**	**0**

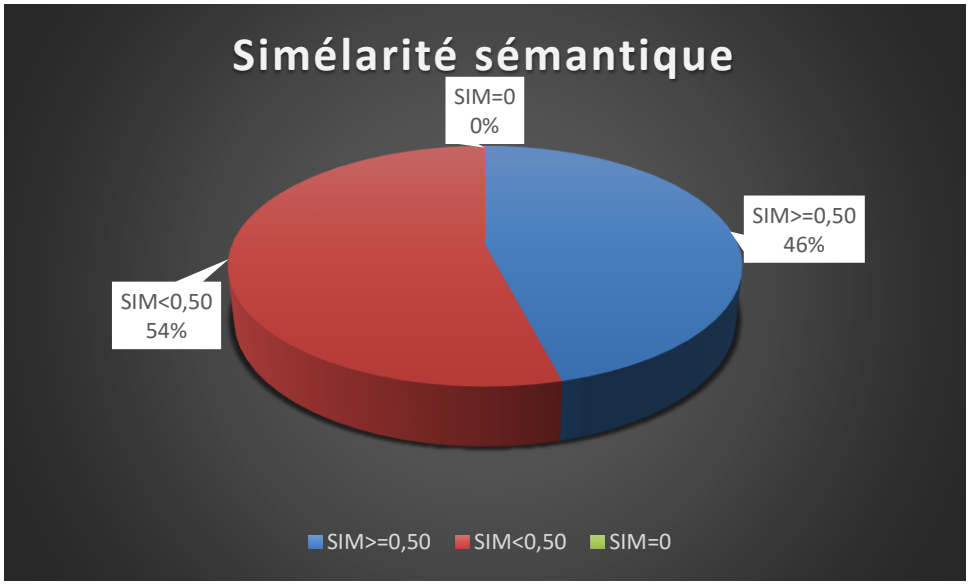
Tableau VI.5 : Pourcentage des requêtes .

Le graphe (N°01) présente le pourcentage des requêtes calculé par similarité syntaxique.



Graphe N° :01 Pourcentage des requêtes calculant par similarité syntaxique.

Le graphe (N°02) présente le pourcentage des requêtes calculé par similarité sémantique.



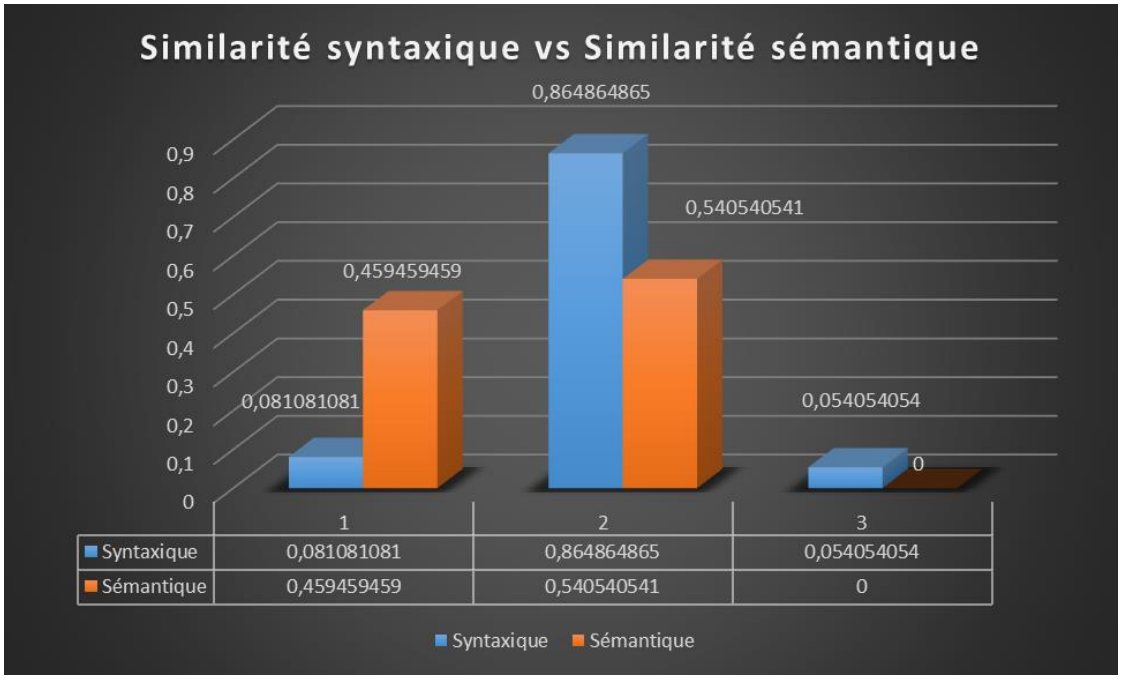
Graphe N° :02 Pourcentage des requêtes calculant par similarité sémantique.

Le graphe suivant présente les résultats représenté dans (le tableau VI.5).

On constate tout d'abord que SIM>=0.50 de similarité sémantique et plus élevé que celle de similarité syntaxique.

On note que SIM<0.50 et plus élevé dan syntaxique que la sémantique.

Et on voit que SIM=0 dans sémantique et nul par contre en trouve deux requêtes leurs SIM=0 dans syntaxique.



Graphe N° :03 Similarité syntaxique vs Similarité sémantique

Points de vue :

Pour obtenir le plus grand nombre des requêtes on diminue la similarité de sémantique à 0.40 :

- **Sémantique :** on prend la similarité à partir de **0.40.**
  Quand Similarité >=0.50 -> le nombre de requêtes : 17
  Quand Similarité >= 0.40 -> le nombre de requêtes sera : 20

Pour obtenir le plus grand nombre des requêtes on diminue la similarité de syntaxique à 0.35 :

- **Syntaxique :** on prend la similarité à partir de **0.35.**
  Quand Similarité >= 0.50 -> le nombre de requêtes : 3
  Quand Similarité >= 0.35 -> le nombre de requêtes : 15

IV.5. Conclusion :

Ce chapitre a été consacré à la mise en œuvre et la validation de notre approche de créer une plateforme basée sur les services Web. Nous avons commencé par la représentation des outils et les langages utiliser durant le développement. Ensuite, nous avons présenté les détails des travaux expérimentaux de l'implémentation des différents modules de l'architecture ainsi que les composants qui les comportent. Nous avons exprimé aussi l'outil de la composition des services web qui nous avons développé. Cette étape a été suivie par une présentation de notre cas d'étude .Enfin nous avons présenté le résultat finale qui exprime la différence entre les similarités sémantique et syntaxique et présenté quelque graphes de pourcentage de chaque similarité.

Conclusion générale :

Dans ce mémoire, nous avons proposé une plate-forme interinstitutionnelle pour faciliter la collaboration et la communication entre institutions afin de fournir des services aux citoyens. La plate-forme suppose que chaque institution peut transmettre les données des citoyens à d'autres intuitions au moyen de services Web.

La plate-forme est composée de modules qui interagissent pour assurer la publication, la découverte et la composition de services Web. Les données personnelles (profils de citoyens) et les profils de services Web sont stockés dans le cloud. L'accès aux services Web est géré par le module de gestion des utilisateurs.

Un algorithme est proposé pour effectuer la composition dynamique de services Web en fonction de la demande de l'utilisateur de l'institution. Le module de découverte utilise des similitudes sémantiques entre la demande d'établissement et les descriptions des services Web. L'algorithme de composition dynamique est basé sur les similitudes sémantiques entre les entrées et les sorties des services Web.

Pour illustrer l'utilisation et l'utilité de la plateforme proposée, une étude de cas a été présentée. Elle montre le fonctionnement de la plate-forme de publication de services Web et de découverte et de composition dynamique en fonction d'une demande donnée.

Partager des informations personnellement identifiables entre institutions nécessite de réfléchir à la sécurité de ces informations supposées accessibles via le cloud. Cet aspect, ainsi que l'amélioration du fonctionnement des modules de découverte et de composition en ajoutant d'autres critères de recherche sémantique, constitueront les perspectives de cette recherche.

[1] Bouri, Sanhouidi et All. Rapport National Sur Le Developpement Humain Madagascar 2006.

[2] SARI, Abdelhaq. Le rôle de la gestion des connaissances dans un projet d'e-gouvernance : cas de la stratégie e-Maroc 2010. 2006.

[3] NDIAYE, Mamadou. E-gouvernance et démocratie en Afrique : le Sénégal dans la mondialisation des pratiques. 2006. Thèse de doctorat. Bordeaux 3.

[4]TAZDAIT Fatiha, L'e-gouvernance et collectivités locales : Etat des lieux dans la Daïra de Sidi-Aïch. Master en Sciences de Gestion, UNIVERSITE ABDERRAHMANE MIRA DE BEJAIA.

[5] 13ème Réunion du Comité Intergouvernemental d'Experts (CIE) Mahe, Seychelles, 27-29 avril 2009.

[6] https://www.idrc.ca/sites/default/files/openebooks/562-6/index.html.

[7] Georges, Labaki. Le Gouvernement Electronique : Visions et stratégies pour le cas libanais. 2011.

[8] AMADOU, Diabagate. E-Gouvernement : Modélisation d'un système d'information décisionnel intégrant les concepts de l'intelligence artificielle, de Big Data et de data science pour une gouvernance efficace des marchés publics. Thèse de doctorat. Université Abdelmalek Essaadi. 2016.

[9]https://www.algerie-eco.com/2018/07/22/indice-de-developpement-de-le-gouvernement-de-lonu-lalgerie-ameliore-son-classement-de-20-places-et-se-classe-130eme/

[10]https://www.01net.com/actualites/les-web-services-des-middle-ware-pour-internet-150665.html

[11] Chemaa-Sofiane, Une approche de composition de services Web à l'aide des Réseaux de Petri orientés objet, Thèse de Doctorat en informatique, UNIVERSITÉ ABDELHAMID MEHRI, CONSTANTINE 2 Année : 2014

[12] https://www.commentcamarche.net .

[13] FDHILA, Walid. Décentralisation optimisée et synchronisation des procédés métiers inter-organisationnels. 2011. Thèse de doctorat. Université Henri Poincaré-Nancy I.

[14] http:// www.dico dunet.com

[15] Djilali Bounaama, Composition des Services Web, Mémoires de Master, Khemis Miliana 2016.

[16] https://www.commentcamarche.net/contents/1332-xml-introduction-a-xml.

[17] KELLERT, Patrick et TOUMANI, Farouk. Les web services sémantiques. Web sémantique, Action spéci_que, 2003, vol. Laboratoire LIMOS - UMR (6158) du CNRS

[18] ALKAMARI, Aniss. Composition de services web par appariement de signatures.2008. UNIVERSITÉ DU QUÉBEC À MONTRÉAL.

[19] BEKKOUCHE, Amina. Composition des Services Web Sémantiques à base d'Algorithmes Génétiques.2012. Université Abou-bekr Belkaid Tlemcen.

[20] https://www.01net.com/actualites/uddi-lannuaire-des-services-web-180308.html

[21]https://www.guideinformatique.com/dossiers-actualites-informatiques-entreprise-ressources-planning-1/soap-uddi-wsdl-transport-decouverte-description-170.html

[22] BOULAHIA, NADIA. Conception et réalisation d'une application à base des services web d'une agence de voyage en ligne.2017. Université Abderrahmane Mira de Béjaia.

[23] Remadelia, Abdellah. Matallah, Imane. Composition des Services Web. 2016. Mémoires de Master Université de Djilali BOUNAÂMA Khemis Miliana.

[24] Remadelia Abdellah, Composition des Services Web, Master en Informatique , Université de Djilali BOUNAÂMA Khemis Miliana Année Universitaire 2015/2016.

[25] https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java

[26] https://netbeans.org/features/ide/index.html

[27] https://www.jmdoudoux.fr/java/dej/chap-jsp.htm

[28] https://www.oracle.com/technetwork/middleware/glassfish/overview/index.html

[29] https://dev.mysql.com/

[30] www.jdom.org

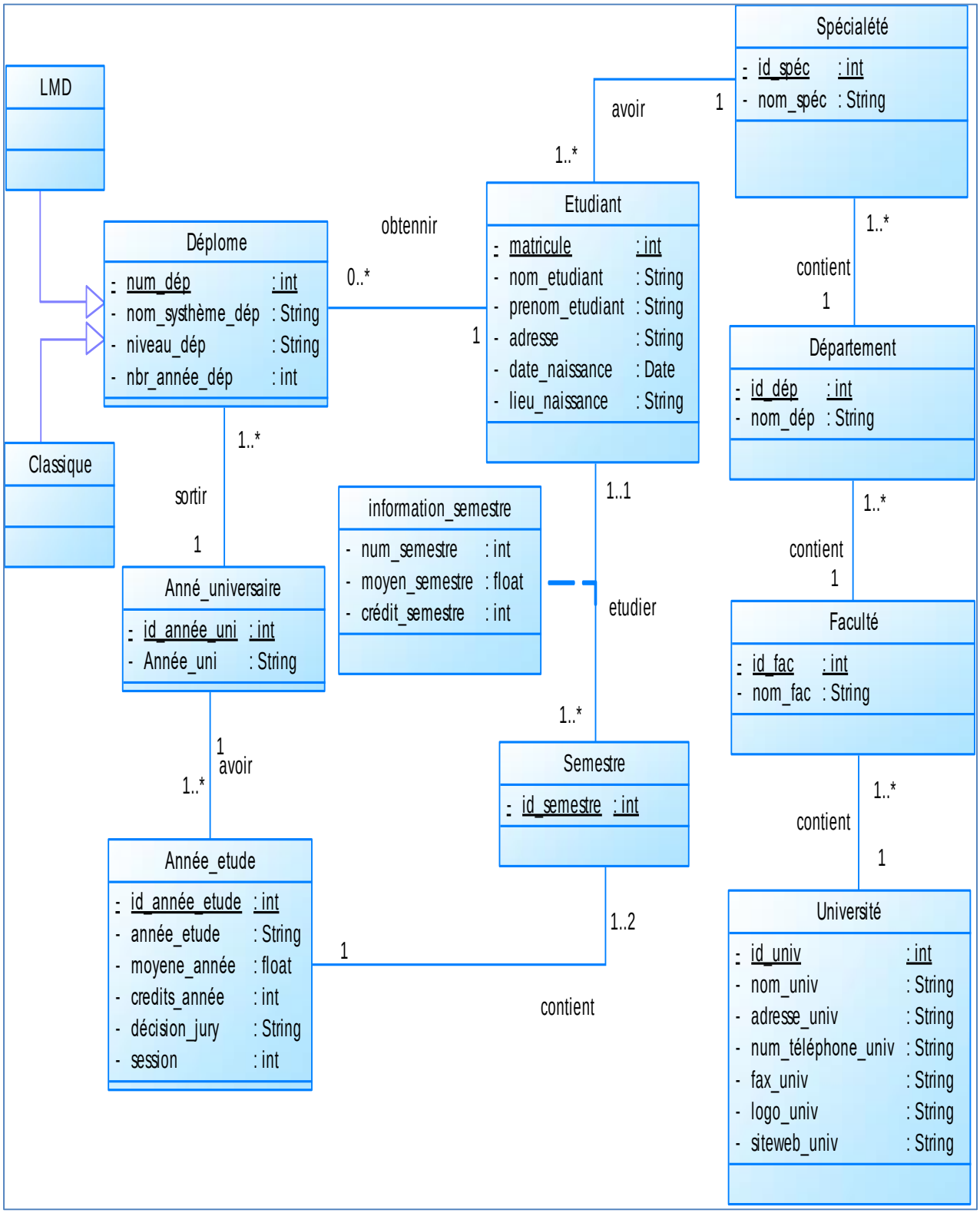
[31] https://developer.mozilla.org/fr/docs/Web/HTML

[32] http://www.standard-du-web.com/feuilles_de_style_en_cascade.php

[33] https://tonyarchambeau.com/blog /400-php-coefficient-de-dice/

❖ **Les diagrammes des bases de données :**

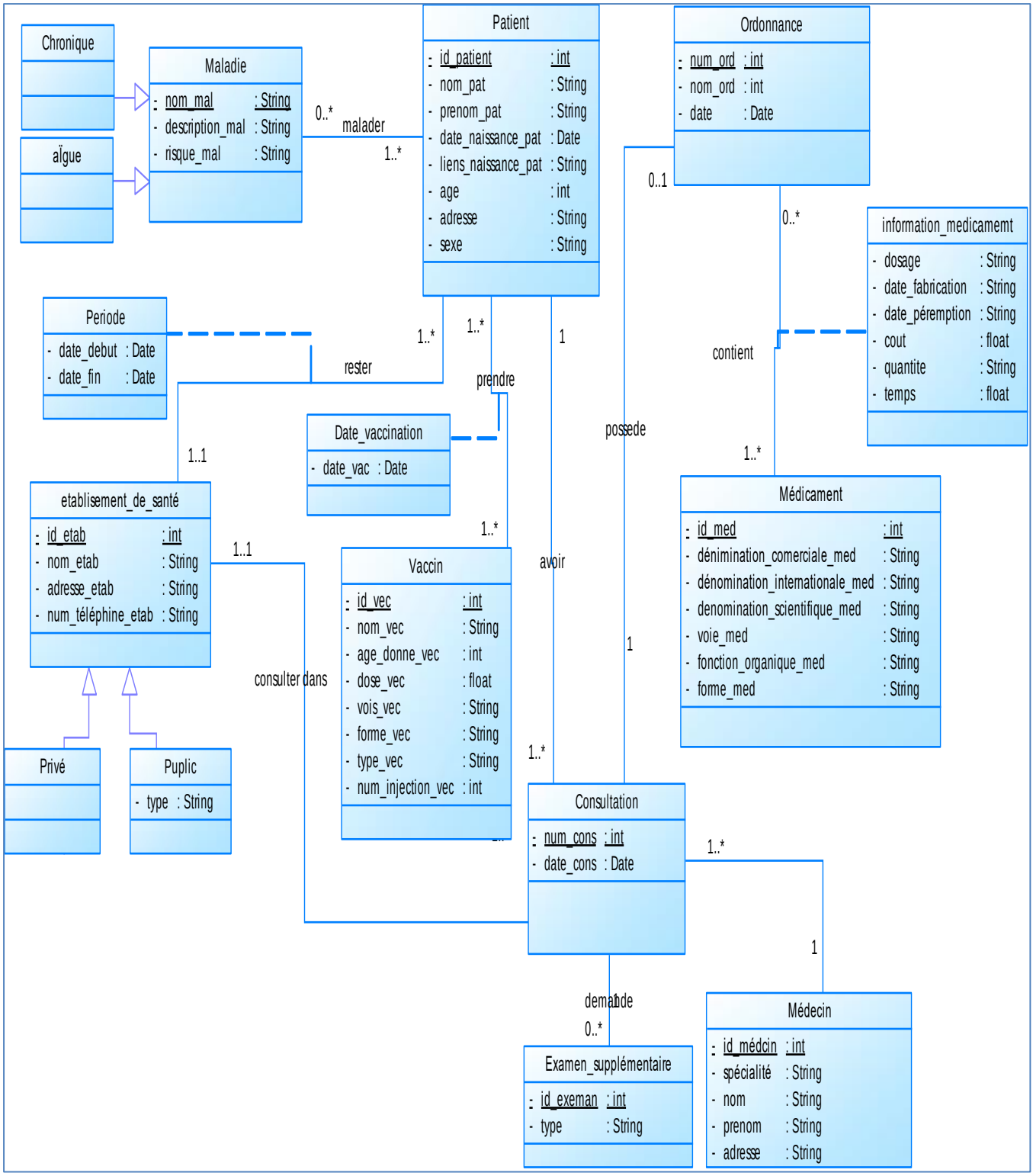
1. Diagramme de diplôme _universitaire :



❖ **DiplomeServiceWeb.wsdl :**

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="DiplomeServiceWeb" targetNamespace="http://diplomeuniv/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://diplomeuniv/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
<types>
<xsd:schema>
<xsd:import
schemaLocation="http://localhost:8080/Webdiplomeuniversity/DiplomeServiceWeb?xsd=1 "
namespace="http://diplomeuniv/"/>
</xsd:schema>
</types>
<message name="getDiplomeServiceWeb">
<part name="parameters" element="tns:getDiplomeServiceWeb"/>
</message>
<message name="getDiplomeServiceWebResponse">
<part name="parameters" element="tns:getDiplomeServiceWebResponse"/>
</message>
<portType name="DiplomeServiceWeb">
<operation name="getDiplomeServiceWeb">
<input message="tns:getDiplomeServiceWeb"
wsam:Action="http://diplomeuniv/DiplomeServiceWeb/getDiplomeServiceWebRequest "/>
<output message="tns:getDiplomeServiceWebResponse"
wsam:Action="http://diplomeuniv/DiplomeServiceWeb/getDiplomeServiceWebResponse "/>
</operation>
</portType>
<binding name="DiplomeServiceWebPortBinding" type="tns:DiplomeServiceWeb">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="getDiplomeServiceWeb">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="DiplomeServiceWeb">
<port name="DiplomeServiceWebPort" binding="tns:DiplomeServiceWebPortBinding">
<soap:address
location="http://localhost:8080/Webdiplomeuniversity/DiplomeServiceWeb "/>
</port>
</service>
</definitions>
```

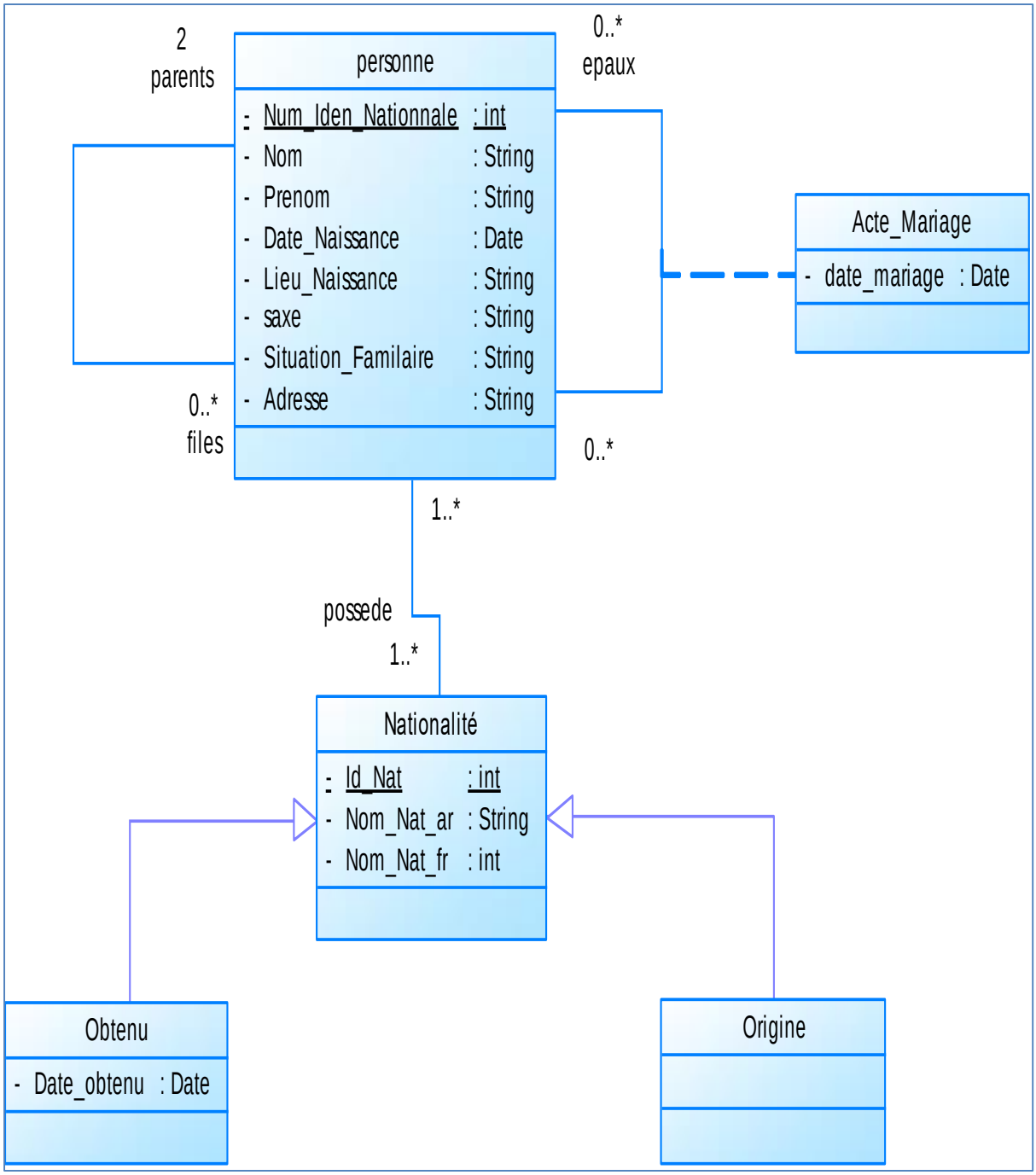
2. Diagramme de santé :



❖ **SanteServiceWeb.wsdl :**

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="SanteServiceWeb" targetNamespace="http://sante.com/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://sante.com/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
<types>
<xsd:schema>
<xsd:import
schemaLocation="http://localhost:8080/SanteServiceWeb/SanteServiceWeb?xsd=1 "
namespace="http://sante.com/"/>
</xsd:schema>
</types>
<message name="getSanteServiceWeb">
<part name="parameters" element="tns:getSanteServiceWeb"/>
</message>
<message name="getSanteServiceWebResponse">
<part name="parameters" element="tns:getSanteServiceWebResponse"/>
</message>
<portType name="SanteServiceWeb">
<operation name="getSanteServiceWeb">
<input message="tns:getSanteServiceWeb"
wsam:Action="http://sante.com/SanteServiceWeb/getSanteServiceWebRequest "/>
<output message="tns:getSanteServiceWebResponse"
wsam:Action="http://sante.com/SanteServiceWeb/getSanteServiceWebResponse "/>
</operation>
</portType>
<binding name="SanteServiceWebPortBinding" type="tns:SanteServiceWeb">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="getSanteServiceWeb">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="SanteServiceWeb">
<port name="SanteServiceWebPort" binding="tns:SanteServiceWebPortBinding">
<soap:address location="http://localhost:8080/SanteServiceWeb/SanteServiceWeb"/>
</port>
</service>
</definitions>
```

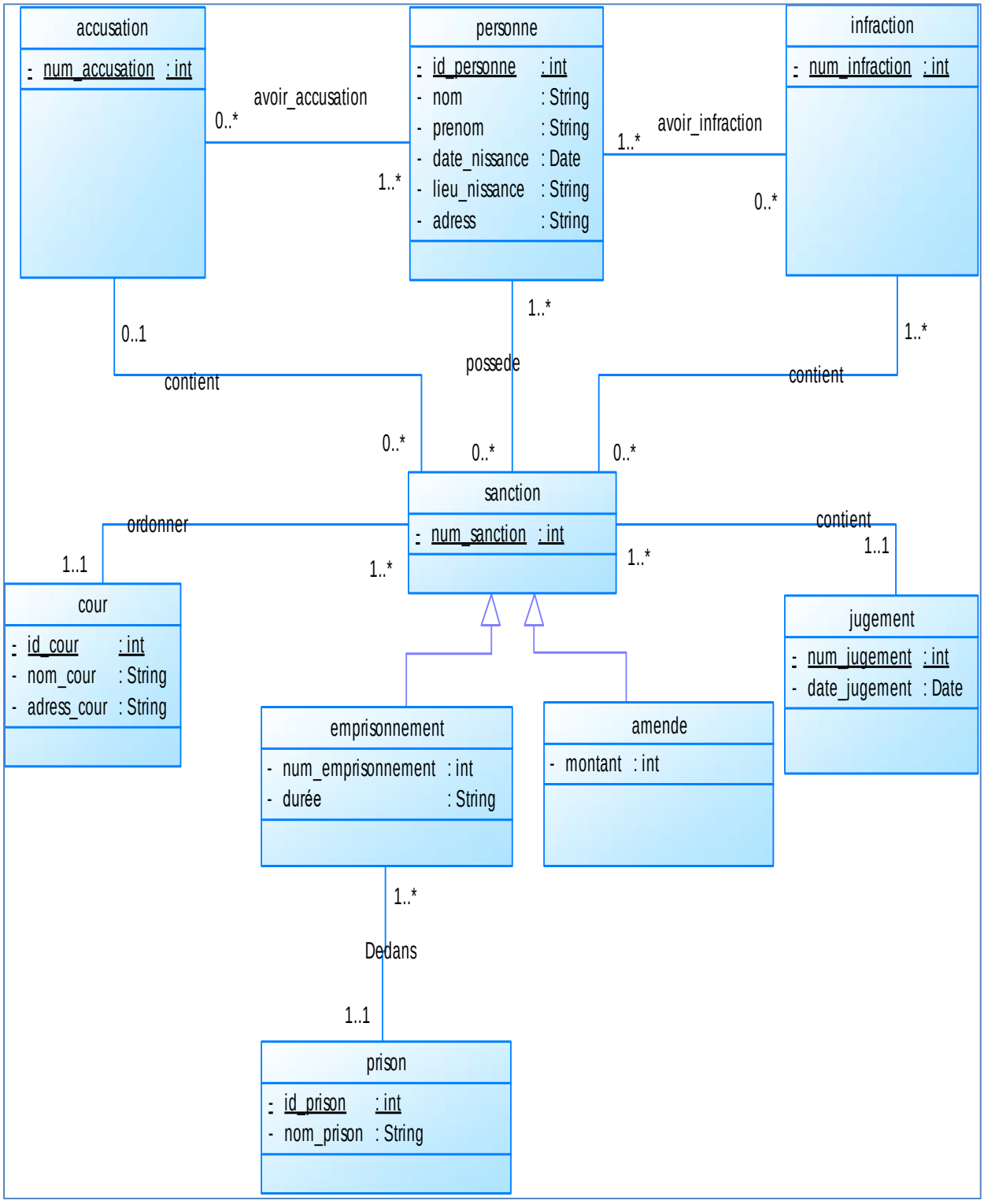
3. Diagramme d'état civil :



❖ **EtatCivilServiceWeb.wsdl :**

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="EtatCivilServiceWeb" targetNamespace="http://com.etatcivil/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://com.etatcivil/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
<types>
<xsd:schema>
<xsd:import
schemaLocation="http://localhost:8080/EtatCivilServiceWeb/EtatCivilServiceWeb?xsd=1 "
namespace="http://com.etatcivil/"/>
</xsd:schema>
</types>
<message name="getEtatCivilServiceWeb">
<part name="parameters" element="tns:getEtatCivilServiceWeb"/>
</message>
<message name="getEtatCivilServiceWebResponse">
<part name="parameters" element="tns:getEtatCivilServiceWebResponse"/>
</message>
<portType name="EtatCivilServiceWeb">
<operation name="getEtatCivilServiceWeb">
<input message="tns:getEtatCivilServiceWeb"
wsam:Action="http://com.etatcivil/EtatCivilServiceWeb/getEtatCivilServiceWebRequest "/>
<output message="tns:getEtatCivilServiceWebResponse"
wsam:Action="http://com.etatcivil/EtatCivilServiceWeb/getEtatCivilServiceWebResponse "/
>
</operation>
</portType>
<binding name="EtatCivilServiceWebPortBinding" type="tns:EtatCivilServiceWeb">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="getEtatCivilServiceWeb">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="EtatCivilServiceWeb">
<port name="EtatCivilServiceWebPort" binding="tns:EtatCivilServiceWebPortBinding">
<soap:address
location="http://localhost:8080/EtatCivilServiceWeb/EtatCivilServiceWeb "/>
</port>
</service>
</definitions>
```

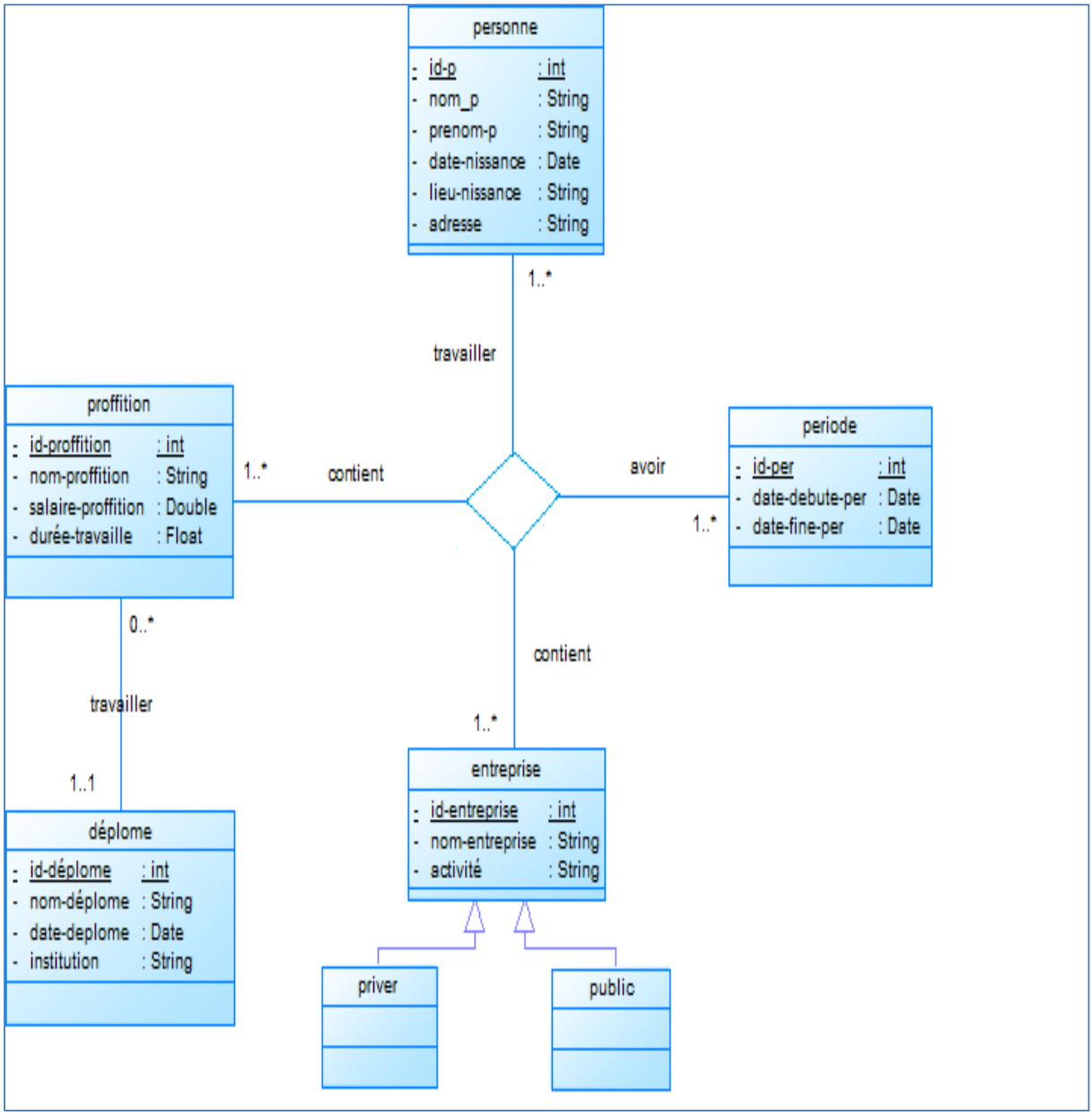
4. Diagramme de casier judiciaire :



❖ **CasierServiceWeb.wsdl :**

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="CasierServiceWeb" targetNamespace="http://com.justice/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://com.justice/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
<types>
<xsd:schema>
<xsd:import
schemaLocation="http://localhost:8080/CasierServiceWeb/CasierServiceWeb?xsd=1 "
namespace="http://com.justice/"/>
</xsd:schema>
</types>
<message name="getCasierServiceWeb">
<part name="parameters" element="tns:getCasierServiceWeb"/>
</message>
<message name="getCasierServiceWebResponse">
<part name="parameters" element="tns:getCasierServiceWebResponse"/>
</message>
<portType name="CasierServiceWeb">
<operation name="getCasierServiceWeb">
<input message="tns:getCasierServiceWeb"
wsam:Action="http://com.justice/CasierServiceWeb/getCasierServiceWebRequest "/>
<output message="tns:getCasierServiceWebResponse"
wsam:Action="http://com.justice/CasierServiceWeb/getCasierServiceWebResponse "/>
</operation>
</portType>
<binding name="CasierServiceWebPortBinding" type="tns:CasierServiceWeb">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="getCasierServiceWeb">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="CasierServiceWeb">
<port name="CasierServiceWebPort" binding="tns:CasierServiceWebPortBinding">
<soap:address location="http://localhost:8080/CasierServiceWeb/CasierServiceWeb "/>
</port>
</service>
</definitions>
```

5. Diagramme de travail :



❖ **TravailServiceWeb.wsdl :**

```
<?xml version="1.0" encoding="UTF-8"?>
2 <definitions name="TravailServiceWeb" targetNamespace="http://com.travaile/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://com.travaile/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
3 <types>
4 <xsd:schema>
5 <xsd:import
schemaLocation="http://localhost:8080/TravailServiceWeb/TravailServiceWeb?xsd=1 "
namespace="http://com.travaile/"/>
6 </xsd:schema>
7 </types>
8 <message name="getTravailTravailServiceWeb">
9 <part name="parameters" element="tns:getTravailTravailServiceWeb"/>
10 </message>
11 <message name="getTravailTravailServiceWebResponse">
12 <part name="parameters" element="tns:getTravailTravailServiceWebResponse"/>
13 </message>
14 <portType name="TravailServiceWeb">
15 <operation name="getTravailTravailServiceWeb">
16 <input message="tns:getTravailTravailServiceWeb"
wsam:Action="http://com.travaile/TravailServiceWeb/getTravailTravailServiceWebRequest "
/>
17 <output message="tns:getTravailTravailServiceWebResponse"
wsam:Action="http://com.travaile/TravailServiceWeb/getTravailTravailServiceWebResponse
"/>
18 </operation>
19 </portType>
20 <binding name="TravailServiceWebPortBinding" type="tns:TravailServiceWeb">
21 <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
22 <operation name="getTravailTravailServiceWeb">
23 <soap:operation soapAction=""/>
24 <input>
25 <soap:body use="literal"/>
26 </input>
27 <output>
28 <soap:body use="literal"/>
29 </output>
30 </operation>
31 </binding>
32 <service name="TravailServiceWeb">
33 <port name="TravailServiceWebPort" binding="tns:TravailServiceWebPortBinding">
34 <soap:address location="http://localhost:8080/TravailServiceWeb/TravailServiceWeb "/>
35 </port>
36 </service>
37 </definitions>
```

✓ **Page Authentification :**

Est une interface où on gère l'authentification des utilisateurs :



Figure N°01 : Page Authentification

✓ **Page Fenêtre d'accueil :**

Dans la suit on prend l'exemple d'authentification de l'administrateur de la plateforme



Figure N° 02: Fenêtre d'accueil

✓ **Fenêtre recherche :**

Cette fenêtre permet à l'utilisateur de faire une recherche à un service, l'utilisateur saisir la requête voulu et les data pour récupérer les informations.

Ces informations peuvent être transmises par d'autres institutions chaque institution est un moyen de Service Web

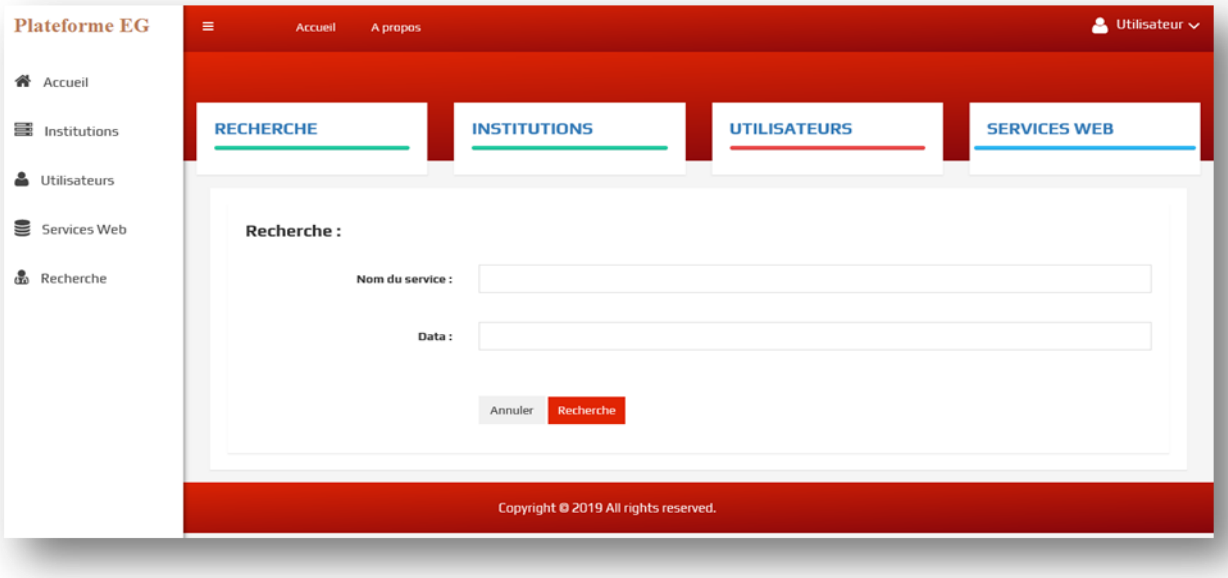


Figure N°03 : Fenêtre recherche