



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre : SIOD05/M2/2023

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : Système d'Information Optimisation et Décision (SIOD)

Composition automatique des sites Web pour réaliser Des applications flexibles et adaptables au sein de L'université de Biskra

Par :

GUEDOUARI NORA

Soutenu le 26/06/2023

Devant le jury composé de :

Zerarka	Mohamed Faouzi	Dr	Président
DJEROU	Leila	Pr	Rapporteur
Merizig	Abdelhak	Dr	Co-rapporteur
Houhou	Okba	Dr	Examineur

شكر وتقدير

أود أن أعبر عن امتناني العميق وشكري الجزيل لمشرفي البروفيسور جرو ليلي على التعاون القيم والمساعدة الثمينة التي قدمتها لي خلال مشروعي في مرحلة الماجستير. لقد كنتي رمزاً للتفاني والمهنية والمعرفة العميقة، ولم أكن أستطيع تحقيق هذا الإنجاز دون دعمي وإرشاداتكي القيمة، وأشكر الدكتور المساعد مريزيق عبد الحق.

إخلاص

أرغب في أن أعبر عن حبي وامتناني العميق لوالدي ووالدتي، وأختي (ضياء الحق، ولاء الدين، عبد الرحمان، محمد)، وجميع أهلي (جوامع، قدواري)، وأصدقائي الأعداء (بثينة، نعيمة، أميرة، شيبية). أنتم جميعاً جزء لا يتجزأ من حياتي وقلبي مليء بالامتنان لكل لحظة قضيتها معها إلى جانبي.

الحمد لله

Table de Matière

Introduction générale.....	8 -
----------------------------	-----

Chapitre 1 : Services Web

1.1 Introduction.....	11
1.2 Paradigme orienté Service.....	11
1.2.1 Évolution vers les approches orientées services	11
1.2.2 Les services	12
1.2.3 Architecture orientée services (SOA)	12
1.3 Concepts de base de services Web.....	15
1.3.1 Architecture des services web.....	15
1.4 Service web sémantique.....	20
1.4.1 Langages de description de services web sémantiques.....	21
1.5 Avantages des Services Web.....	25
1.6 Inconvénients des Services Web.....	25
1.7 Conclusion.....	26

Chapitre 2 : Composition Des Services Web

2.1 Introduction.....	28
2.2 Concepts et définitions.....	28
2.2.1 Services Web atomiques et services Web composites.....	28
2.2.2 Propriétés fonctionnelles et non fonctionnelles d'un service	28
2.2.3 Workflow	29
2.3 Processus de composition de services.....	30
2.4 Méthodes de composition de services.....	32
2.4.1 Composition semi-automatique	32
2.4.2 Composition automatique	33
2.4.3 Composition manuelle	33
2.4.4 Composition statique	34

2.4.6 Orchestration.....	35
Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs	
2.5 Conclusion.....	36
3.1 Introduction.....	38
3.2 Procédure d'inscription des NBs.....	38
3.3 Développement d'un service composite pour l'inscription des NB.....	42
3.3.1 Environnement de travail	42
3.3.2 Création des services web	44
3.3.3 Création d'un service composite.....	44
3.3.4 Architecture détaillée	46
3.4 Front end du système.....	49
3.5 Conclusion.....	55
Conclusion générale	57
References	59

Table de figure

Figure 1. 1: Interactions au sein d'une architecture orientée service [1].	14
Figure 1. 2: L'utilisation des technologies des services web par les acteurs fournisseur et demandeur [1].	17
Figure 1. 3: Structure générale d'un document WSDL 2.0 [8].	20
Figure 1. 4: Structure d'un message SOAP [9].	20
Figure 1. 5: Structure de l'ontologie de services OWL-S [1].	22
Figure 1. 6: Structure de WSMO [1].	24
Figure 2. 1: Exemple de flux de travail complexe [16].	30
Figure 2. 2: Composition des services [15].	31
Figure 2. 3: Processus de composition des services [17].	32
Figure 2. 4: Méthodes de composition de services.	36
Figure 3. 1: Scénario de processus d'inscription de nouveaux bacheliers.	41
Figure 3. 2: Workflow désignant la séquence ordonnée d'activités du processus d'inscription.	41
Figure 3. 3: NetBeans logo [26].	42
Figure 3. 4: MySQL Workbench logo [29].	43
Figure 3. 5: Visual studio logo [31].	43
Figure 3. 6: PHP logo [33].	44
Figure 3.7: Gestion des transactions entre les services.	46
Figure 3. 8: Architecture détaillée.	47
Figure 3. 9: Page login.	49
Figure 3. 10: Page inscription partie 1.	49
Figure 3. 11: Page inscription partie 2.	50
Figure 3. 12: Page inscription partie 3.	50
Figure 3. 13: Page home.	51
Figure 3. 14: Page update transport.	52
Figure 3. 15: Page verification.	53
Figure 3. 16: Page return.	53
Figure 3. 17: Fichier WSDL partie 1.	54
Figure 3. 18: Fichier WSDL partie 2.	54
Figure 3. 19: Fichier WSDL partie 3.	55

Résumé

Ce projet s'inscrit dans le cadre de la digitalisation des administrations et des services publics et vise à développer une application flexible pour l'inscription des nouveaux bacheliers dans un établissement d'enseignement supérieur. L'Université de Biskra est un exemple d'une telle institution. Le développement de cette application est basé sur la composition de services Web créés pour des tâches simples qui collaborent en partageant des informations et fournissent des services transparents aux nouveaux bacheliers, leur permettant de réduire le temps et les efforts nécessaires pour se déplacer entre les différents bureaux situés à l'intérieur et à l'extérieur de l'université, tout en réduisant les informations redondantes

Mots clés

Digitalisation des administrations et des services publics, composition de services Web, l'inscription des nouveaux bacheliers

Abstract

This project is part of the digitalization of administrations and public services and aims to develop a flexible application for the registration of new bachelors in a higher education institution. The University of Biskra is an example of such an institution. The development of this application is based on the composition of web services created for simple tasks that collaborate by sharing information and provide transparent services to new bachelors, allowing them to reduce the time and effort required to move between different offices located inside and outside the university, while reducing redundant information

Keywords

Digitization of administrations and public services, composition of Web services, registration of new bachelors

ملخص

هذا المشروع هو جزء من رقمنة الإدارات والخدمات العامة ويهدف إلى تطوير تطبيق مرن لتسجيل الطلاب الجدد في مؤسسة التعليم العالي. جامعة بسكرة مثال على هذه المؤسسة. يعتمد تطوير هذا التطبيق على تكوين خدمات الويب التي تم إنشاؤها لمهام بسيطة تتعاون من خلال مشاركة المعلومات وتقديم خدمات شفافة للطلاب الجدد، مما يسمح لهم بتقليل الوقت والجهد اللازمين للتنقل بين المكاتب المختلفة الموجودة داخل وخارج الجامعة. ، مع تقليل المعلومات الزائدة عن الحاجة.

الكلمات الدالة

رقمنة الإدارات والخدمات العامة ، وتكوين خدمات الويب ، وتسجيل الطلاب الجدد

Introduction générale

Introduction générale

L'accélération de la transition numérique à travers la généralisation de l'usage des technologies de l'information et de la communication, notamment dans les administrations et les services publics, est devenue une priorité pour le gouvernement algérien qui a mis des plans et des stratégies pour soutenir les procédures administratives décentralisées répondant aux exigences et intérêts fondamentaux des citoyens. Ce projet s'inscrit dans le contexte de numérisation et digitalisation des systèmes d'information, il s'agit de développer une application flexible pour l'inscription des nouveaux bacheliers (NBs) dans un établissement d'enseignement supérieur. L'université Biskra qui est un exemple de cet établissement.

Le développement de cette application se base sur une composition des services Web créées pour des tâches simples et qui collaborent par le partage des informations, en fournissant des services transparents aux NBs permettant de réduire le temps et les efforts de NBs de se déplacer entre les différents bureaux situés à l'intérieur et à l'extérieur de leurs universités accueillantes en réduisant aussi les informations redondantes. Dans ce cas, le contexte général dans lequel s'inscrit ce projet puise sa source dans un effort de recherche à la fois dans le domaine des services Web et la composition de Web services.

Les services web sont des modules logiciels qui fournissent un service et qui peuvent être consultés et appelés via un réseau. Le service Web est devenu la technologie de choix pour l'informatique orientée services pour répondre aux exigences d'interopérabilité dans le Web applications [4]. Cependant, dans la plupart des cas, un seul service ne peut répondre aux besoins des clients, et par conséquent, on est obligé de composer plusieurs services afin de satisfaire une requête [3].

La composition de services Web est définie comme le développement d'une coordination simplement optimale entre un certain nombre de services Web disponibles pour fournir un nouveau service Web composé, destiné à satisfaire les besoins de certains utilisateurs pour lesquels un seul service Web n'est pas suffisant. Un objectif important de la composition de services Web est d'atteindre une flexibilité maximale en s'adaptant dynamiquement à un environnement dans lequel les services disponibles changent constamment en termes de disponibilité, d'équilibrage de charge ou d'application [21].

La composition des services a reçu beaucoup d'attention de la part de la communauté de la recherche et de l'industrie, qui a proposé de nombreuses approches qui peuvent être classées selon différents critères [13], On trouve les approches automatiques, semi-automatiques ou manuelles, si on considère le critère de l'implication de l'utilisateur dans la définition du schéma de composition. Il peut également être classé selon que la spécification du workflow avant ou après l'exécution, dans ce cas, on parle des approches statiques ou dynamiques [16]. Enfin, la composition peut être abordée de deux

manières différentes en fonction du contrôle des enchaînements d'appels des services participant, dans ce cas, la composition sera appelée soit orchestration, soit chorégraphie.

Dans ce manuscrit, nous visons à fournir une compréhension approfondie des services web, de leur composition et de leur application dans le domaine de l'inscription des NBs, à travers les trois chapitres suivants :

Dans le premier chapitre 1, nous explorons le paradigme orienté service, qui constitue la base conceptuelle des services web puis nous examinons les concepts de services web et avec les différents aspects liés à leur conception, leur utilisation et leurs avantages.

Le deuxième chapitre, abordera la manière dont les services web peuvent être combinés pour créer des fonctionnalités plus complexes et étendues. Nous explorerons les concepts de base de la composition des services web et les différentes méthodes de composition.

Le troisième chapitre se concentrera sur le développement d'une application spécifique pour l'inscription des nouveaux bacheliers. Nous détaillerons la procédure d'inscription des NBs et décrirons le processus de développement d'un service composite en présentant l'architecture de l'application et les outils de son développement ainsi que ses différentes interfaces.

Enfin, le manuscrit se terminera par une conclusion qui résume le travail réalisé et les résultats obtenus en donnant les améliorations possibles et quelques perspectives.

Chapitre1 : Services Web

1.1 Introduction

Le service Web est devenu la technologie de choix pour l'informatique orientée services pour répondre aux exigences d'interopérabilité dans le Web applications. Il est la réalisation la plus importante d'une architecture Orientées Services (SOA) qui vise à mettre en place des processus métier performants ainsi que des systèmes d'information constitués de services applicatifs indépendants et interconnectés.

Dans ce chapitre, nous présentons, dans la première partie, le paradigme émergeant de l'Informatique Orientée Services qui utilise les concepts de service et d'Architecture Orientée Service pour la construction rapide et à faible coût de nouvelles applications logicielles. Ensuite, nous décrivons les concepts de base Web service.

1.2 Paradigme orienté Service

1.2.1 Évolution vers les approches orientées services

En génie logiciel, différents paradigmes ont été proposés pour le développement de logiciels. Chaque nouveau paradigme essaie de résoudre les limitations des paradigmes précédents en réutilisant certains principes et en ajoutant de nouveaux concepts. Les approches orientées objets, orientées composants et orientées services ont émergé successivement.

Trois architectures à base de composants ont été proposées pour dissimuler la complexité d'intégration d'applications autonomes et hétérogènes : EJB (Enterprise Java Beans), CORBA (Common Object Request Broker Architecture) et .NET. Cependant, le couplage entre les objets est relativement fort dans ces technologies à base de composants distribués. Par conséquent, pour pallier les limites de toutes ces architectures, les efforts de conception ont donné lieu au concept d'architecture orientée service (Service Oriented Architecture ou SOA en abrégé). Ainsi, les approches orientées services ont vu le jour.

Le paradigme orienté services (Service-Oriented Computing) est un paradigme interdisciplinaire destiné aux applications distribuées. Son apparition a introduit une nouvelle manière de concevoir, d'intégrer, de déployer et d'utiliser les logiciels. Les approches orientées services mettent en avant l'idée de composer des services indépendants pour réaliser des applications logicielles agiles faiblement couplées. Ces approches exploitent le concept de service comme élément fondamental autour duquel les applications complexes sont développées. L'idée de base est d'encapsuler les fonctionnalités offertes par les organisations sous forme de services.

Chapitre1 : Services Web

Les approches à base de services offrent la possibilité de réaliser une interopérabilité à grande échelle en garantissant la souplesse nécessaire pour s'adapter à l'évolution des technologies et des besoins des utilisateurs (individus ou entreprises). Un des principaux bénéfices des approches à base de services est le couplage faible entre le fournisseur et le consommateur de service d'une part, et entre les différents services réunis dans une même application d'autre part. Le consommateur d'un service n'a pas besoin d'avoir connaissance des détails techniques tels que la technologie d'implémentation et la plateforme d'exécution d'un service. Ce faible couplage permet le développement d'applications de façon parallèle et indépendante, ce qui entraîne une réduction des coûts de développement et d'intégration des applications [1].

1.2.2 Les services

Le service est la pierre angulaire d'une architecture orientée services, et bien qu'il soit utilisé dans plusieurs domaines, il peut être simplement défini comme une action effectuée par une entité au profit d'une autre. Cette entité peut être une donnée, un objet sur le web ou un système d'information adaptable d'une entreprise. Avant d'utiliser un service, il est important de connaître ses fonctionnalités et ses caractéristiques non-fonctionnelles, telles que sa performance et sa disponibilité. Pour un service logiciel, qui est une entité logicielle mise à disposition d'autres applications, il doit être décrit et publié pour que les utilisateurs potentiels puissent découvrir ses capacités et exécuter le service pour obtenir la fonctionnalité souhaitée [1].

1.2.3 Architecture orientée services (SOA)

L'architecture orientée services (SOA) est un paradigme qui a été largement adopté et qui fournit un ensemble de méthodes pour le développement et l'intégration de systèmes. Ces systèmes sont conçus avec des fonctionnalités développées sous forme de services interopérables et indépendants les uns des autres. En général, l'architecture orientée services repose sur un ensemble de principes qu'il est important de respecter pour garantir son bon fonctionnement [2], [3]:

Couplage faible : Le couplage se réfère au niveau d'interaction entre deux ou plusieurs composants logiciels. Deux composants sont considérés comme couplés s'ils échangent des informations. Le degré de couplage est qualifié de fort s'ils échangent beaucoup d'informations et faible dans le cas contraire. Dans une architecture SOA, il est recommandé que le couplage entre les applications clientes et les services soit faible. Les services web, qui utilisent des messages décrits par le standard XML, sont caractérisés par leur généricité et leur haut niveau d'abstraction,

Chapitre 1 : Services Web

permettant ainsi la coopération entre les applications tout en garantissant un faible taux de couplage. Par conséquent, il est possible de modifier un service sans compromettre sa compatibilité avec les autres services qui composent l'application.

Interopérabilité : L'interopérabilité permet à des applications écrites dans des langages de programmation différents et s'exécutant sur des plateformes différentes de communiquer entre elles. En manipulant différents standards que ce soit XML ou les protocoles d'Internet, les services web garantissent un haut niveau d'interopérabilité des applications et ceci indépendamment des plateformes sur lesquelles elles sont déployées et des langages de programmation dans lesquels elles sont écrites. Ainsi, en s'appuyant sur un format d'échange de messages standard et sur l'ubiquité de l'infrastructure d'Internet, l'interopérabilité est donc une caractéristique intrinsèque aux services Web.

Réutilisabilité : La réutilisation présente l'avantage de réduire les coûts de développement en utilisant des composants existants. Dans le cas de l'approche des services web, la séparation des opérations en services autonomes a pour objectif de promouvoir leur réutilisation. Ainsi, lorsqu'un client définit ses exigences, il est souvent possible d'utiliser des services déjà existants pour répondre à une partie de ces exigences. Cette approche facilite la maintenance de l'application et permet un gain de temps considérable. En réutilisant des services web, les développeurs peuvent se concentrer sur les fonctionnalités spécifiques de l'application plutôt que de réinventer la roue, ce qui peut entraîner des coûts élevés et une duplication de travail.

Découverte : La découverte de services est une étape essentielle pour permettre la réutilisation de ces services. Dans l'approche des services web, il est important de réduire autant que possible l'intervention humaine pour permettre une découverte automatique des services. Un développeur peut simplement interroger un moteur de recherche de services pour trouver le service approprié et l'utiliser dans son application. Cette approche de découverte automatique de services permet aux développeurs de gagner du temps et de trouver plus facilement les services dont ils ont besoin. En résumé, la découverte automatique de services est un élément clé de l'approche des services web pour favoriser la réutilisation des services existants.

Composition : Les collections de services peuvent être coordonnées et assemblées pour former une composition de services. C'est l'un des avantages de l'architecture orientée services (SOA) qui permet de construire de nouveaux systèmes à partir de services existants. Grâce à la composition, les services web peuvent être réutilisés, mais cela nécessite de pouvoir les découvrir et de les

Chapitre 1 : Services Web

composer en utilisant les technologies offertes par Internet et un ensemble de standards dédiés à la composition.

L'architecture orientée services repose sur un modèle qui décrit les interactions entre trois types d'acteurs : les fournisseurs de services, les clients de services et les registres de services. Ces interactions sont basées sur trois primitives de communication : la publication, la découverte et l'invocation. Les fournisseurs de services sont responsables de la création et du déploiement des services. Ils publient la description de leurs services dans un registre (étape 1 : publication). Les clients de services envoient des requêtes au registre pour découvrir un service qui répond à leurs besoins (étape 2 : découverte). Une fois qu'un service a été sélectionné, le client de service peut se connecter au fournisseur de service et utiliser le service en suivant sa description (étape 3 : invocation). Ce modèle d'interaction est illustré dans la Figure 1.1 et permet aux différents acteurs de collaborer efficacement en utilisant les services fournis par les fournisseurs. Grâce à ce modèle, les services peuvent être facilement découverts, utilisés et réutilisés par différents clients [4].

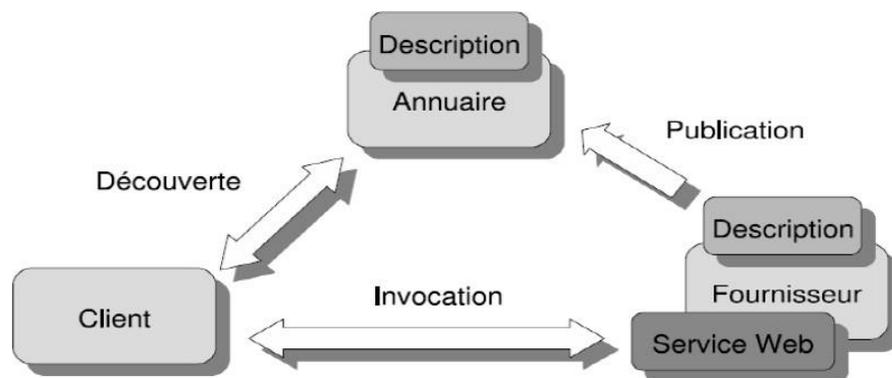


Figure 1. 1: Interactions au sein d'une architecture orientée service [1].

L'architecture orientée services vise à faciliter l'accès à des fonctionnalités proposées par des tiers distribués et hétérogènes de manière simple et rapide. Pour y parvenir, il est nécessaire de mettre en place un environnement qui spécifie les modalités d'interaction et d'utilisation des services, comprenant un ensemble de normes, protocoles et mécanismes qui peuvent être divisés en deux catégories : les mécanismes fonctionnels et les mécanismes non fonctionnels. Les mécanismes fonctionnels concernent les principales tâches liées à l'utilisation des services, à savoir

Chapitre1 : Services Web

la publication, la découverte, la composition et l'invocation. Quant aux mécanismes non fonctionnels, ils prennent en charge les besoins optionnels tels que la sécurité, les transactions et la qualité de service [1].

1.3 Concepts de base de services Web

Les services web sont des modules logiciels qui fournissent un service et qui peuvent être consultés et appelés via un réseau. Ils ont été proposés initialement par IBM et Microsoft, puis en partie standardisés par le W3C (le consortium du World Wide Web).

IBM décrit les services web comme une nouvelle vague d'applications web. Ils sont modulaires, auto-contenus et autodescriptifs, ce qui leur permet d'être publiés, localisés et appelés depuis le web. Les services web sont capables d'effectuer un large éventail d'actions, allant de requêtes simples à des processus complexes. Une fois qu'un service web est déployé, d'autres applications peuvent le découvrir et l'appeler [5].

D'après cette définition, un service web est une collection de fonctionnalités mises à disposition pour être utilisées, utilisant Internet comme canal pour accomplir une tâche. Cela peut être comparé à un processus métier virtuel qui établit des interactions au niveau de l'application [6].

Le groupe de travail du W3C sur les services Web a défini un service Web comme étant un système logiciel conçu pour permettre l'interaction entre ordinateurs via le réseau. Ce système possède une interface décrite dans un format pouvant être traité par des ordinateurs, comme WSDL. D'autres systèmes interagissent avec le service Web de manière prescrite par sa description en utilisant des messages SOAP, qui sont généralement transmis à l'aide du protocole HTTP et d'une sérialisation XML, en conjonction avec d'autres normes Web [7].

Les efforts de recherche et de développement récents autour des Web services ont conduit à un certain nombre de spécifications qui définissent aujourd'hui l'architecture de référence des Web Services.

1.3.1 Architecture des services web

L'architecture des services web consiste à acheminer une demande de service entre clients/serveur, qui sont, tous les deux, connectés au réseau. Le client est soit un être humain qui formule sa requête via un navigateur web, soit une demande de service web automatisée par une application.

Cette architecture vise trois objectifs importants :

- Identification des composants fonctionnels ;

Chapitre 1 : Services Web

- Définition des relations entre ces composants ;
- Établissement d'un ensemble de contraintes sur chaque composant de manière à garantir les propriétés globales de l'architecture.

L'architecture de référence des Web services (Figure 1.2) s'articule autour des trois rôles suivants [5] :

- Le fournisseur de service : correspond au propriétaire du service. Techniquement, il est constitué par la plate-forme d'accueil du service.
- Le client : correspond au demandeur de service. Techniquement, il est constitué par l'application qui va rechercher et invoquer un service. L'application client peut être elle-même un Web service.
- L'annuaire des services : correspond à un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs ainsi que des facilités de recherche de services à l'intention des clients.

Les interactions de base entre ces trois rôles incluent les opérations de publication, de recherche et de liens d'opérations (binding).

Nous décrivons ci-dessous un scénario type d'utilisation de cette architecture (Figure 1.2).

- Le fournisseur de services définit la description de son service (en WSDL) et la publie sur un annuaire à l'aide de UDDI.
- Le demandeur recherche un service web qui correspond à certaines caractéristiques X, Y et Z.
- L'annuaire trouve un service qui répond à ces caractéristiques et envoie les informations sur le fournisseur et le service au demandeur.
- Le demandeur demande le contrat du service au fournisseur.
- Le fournisseur envoie le contrat du service, qui est généralement décrit dans un document WSDL.
- Le demandeur appelle le service web en utilisant le contrat spécifié (par exemple, en envoyant une requête SOAP).

Chapitre 1 : Services Web

- Le service web renvoie le résultat de l'appel (également sous forme de message SOAP).

Le service web renvoie le résultat de l'appel sous forme de message SOAP, ce qui est l'un des éléments clés de l'infrastructure des services web. Ce qui rend cette infrastructure unique est qu'elle est mise en place en utilisant exclusivement les protocoles d'Internet tels que HTTP et les formats d'échange standardisés tels que XML. Cette infrastructure est basée sur trois spécifications devenues des standards : SOAP, UDDI et WSDL. Dans la suite de cette partie, nous allons présenter ces trois langages qui constituent les fondements des services web [8].

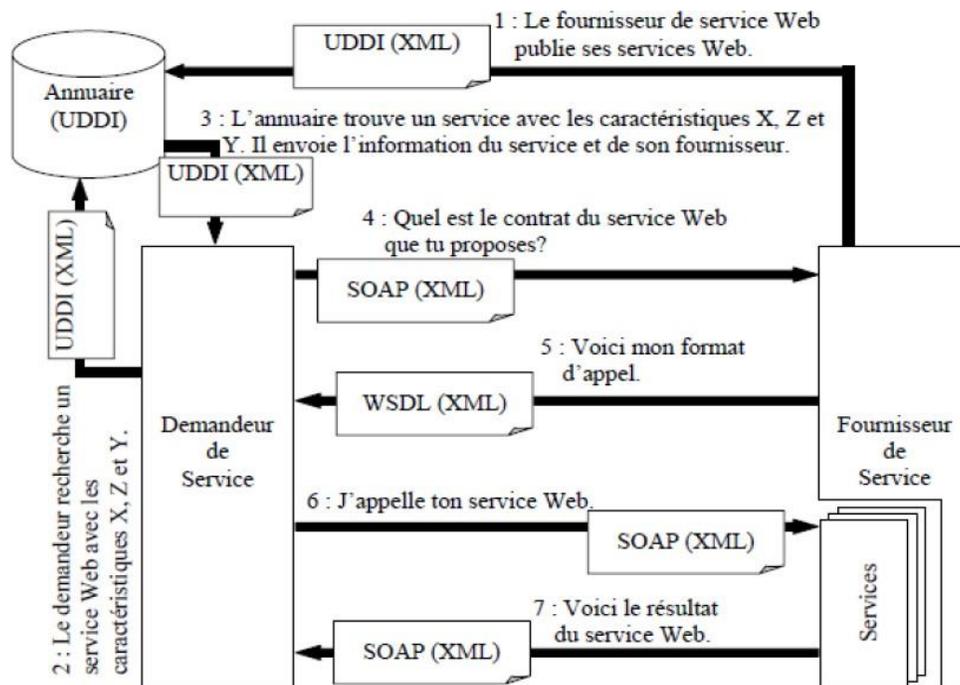


Figure 1. 2: L'utilisation des technologies des services web par les acteurs fournisseur et demandeur [1].

1.3.1.1 Découverte UDDI

UDDI (Universal Description, Discovery and Integration) est une spécification technique qui est soutenue par OASIS. Elle a été créée suite à un accord entre plusieurs entreprises telles que Microsoft, Ariba et IBM. UDDI permet la publication et la découverte de services web grâce à un mécanisme de registre distribué. Ce registre est organisé en trois parties distinctes : les pages blanches, les pages jaunes et les pages vertes. Les données stockées dans UDDI sont structurées en XML :

Chapitre 1 : Services Web

- Pages blanches : Les pages blanches d'UDDI contiennent des informations générales sur les fournisseurs de services, telles que le nom de l'entreprise, ses coordonnées et une description de l'entreprise. Elles contiennent également l'ensemble de leurs identifiants
- Pages jaunes : Les pages jaunes d'UDDI comprennent des descriptions détaillées sur les fournisseurs de services qui ont été catalogués dans les pages blanches. Cela permet de classer les entreprises et les services en fonction de leur secteur d'activité. En outre, les pages jaunes recensent les services web de chaque entreprise sous le standard WSDL.
- Pages vertes : Les pages vertes d'UDDI fournissent des informations techniques précises sur les services fournis. Elles indiquent généralement les adresses web des services et les moyens d'y accéder. Ces informations incluent la description détaillée du service, le processus d'utilisation et les protocoles utilisés pour son invocation.

1.3.1.2 Description WSDL

Le langage WSDL (Web Service Description Language) permet de décrire de manière détaillée les fournisseurs de services répertoriés dans les pages blanches, en classant les entreprises et les services selon leur secteur d'activité. Il recense également les services web de chaque entreprise sous le standard WSDL, qui est un langage de description de services web basé sur XML proposé par le W3C. Le W3C définit notamment les catégories d'informations à prendre en compte dans la description d'un service web, telles que les opérations proposées par le service web, les données et messages échangés lors de l'appel d'une opération, le protocole de communication et les ports d'accès au service. Le standard WSDL offre une description sur deux niveaux : abstrait et concret. Le niveau abstrait est principalement utilisé lors du processus de sélection, tandis que le niveau concret est plutôt utilisé lors de l'invocation des opérations du service web (voir figure 1.3) [8].

La partie abstraite : Décrit les messages et les opérations disponibles via les éléments suivants :

- **Les types de données :** Le rapport concerne les structures de données des paramètres qui sont utilisés par le service. Ce rapport est facultatif et l'utilisation d'une seule balise est autorisée.
- **Les messages :** La fonction de l'encapsulation consiste à protéger les données transmises pour chaque opération d'un service en deux parties distinctes : la requête et la réponse. Ces deux éléments sont délimités par une balise et plusieurs messages peuvent être inclus.

Chapitre1 : Services Web

- **Les opérations** : Le service est constitué d'une ou plusieurs opérations, qui sont décrites par des éléments spécifiques. Chaque opération est identifiée par un nom et peut comporter zéro ou plusieurs messages en entrée, ainsi que zéro ou plusieurs messages de sortie ou d'erreur. Il est possible d'inclure plusieurs messages pour chaque opération.

La partie concrète : La description d'un service Web particulier comprend le protocole de communication et le type d'encodage à utiliser pour les messages nécessaires à son invocation. Les informations clés à spécifier pour un service sont le protocole utilisé et le service en lui-même.

- **Le protocole de communication** : La définition des protocoles de communication utilisés pour appeler un service Web permet d'établir les liens entre le document, les messages SOAP et les opérations invoquées. Il existe plusieurs protocoles de communication autorisés, et la balise utilisée dépendra de celui choisi.
- **Service** : L'accès à un service est défini par une collection de ports d'accès regroupés dans un élément XML appelé "service". Ces ports représentent les adresses URI (Uniform Resource Identifier) du service, ce qui signifie qu'un même service Web peut être accessible à partir de plusieurs ports différents [1].

1.3.1.3 Transport SOAP

SOAP (Simple Object Access Protocol) est un protocole recommandé par le W3C pour faciliter l'échange d'informations structurées avec les services web. Ce protocole utilise principalement les protocoles HTTP (Hyper-Text Transfer Protocol) et SMTP (Simple Mail Transfer Protocol) pour transporter les messages. SOAP repose sur le standard XML pour l'encodage des données, ce qui lui permet de bénéficier des avantages de la généricité, de l'abstraction et de la portabilité offerts par ce standard pour normaliser et structurer les données.

Chapitre 1 : Services Web

En tant que couche inférieure de la pile des protocoles, SOAP fournit une base solide pour les échanges d'informations entre les applications web

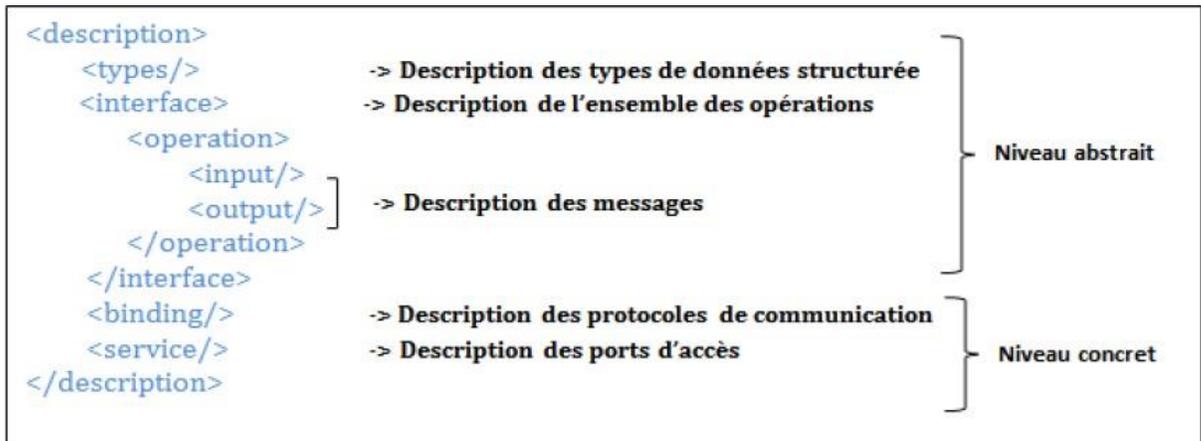


Figure 1. 3: Structure générale d'un document WSDL 2.0 [8].

Les services Web fournissent un framework permettant l'échange de messages sur lequel les services peuvent se baser. Un message SOAP est divisé en trois parties distinctes : une enveloppe contenant un entête et un corps (voir Figure 1.4). L'entête est facultatif et fournit des données supplémentaires au message SOAP. Le corps contient l'opération du service invoquée ainsi que les valeurs des paramètres nécessaires à cette invocation, du côté client. Du côté du service, le corps contient le résultat de l'exécution de l'opération invoquée [9] .



Figure 1. 4: Structure d'un message SOAP [9].

1.4 Service web sémantique

Tim Berners-Lee a défini le Web sémantique [9] comme suit : « Le Web sémantique est une extension du Web actuel (prolongation du Web actuel), dans laquelle l'information reçoit une signification bien définie, améliorant les possibilités de travail collaboratif entre les ordinateurs et les personnes ».

L'objectif principal du web sémantique est de faciliter l'utilisation, la découverte, l'intégration et la réutilisation de ressources du Web en les définissant et en les reliant. Pour y parvenir, le web

Chapitre 1 : Services Web

sémantique doit fournir un accès aux ressources via des descriptions sémantiques exploitables et compréhensibles par des machines. Ces descriptions sémantiques reposent sur des ontologies. Selon la définition, une ontologie est une spécification explicite d'une conceptualisation, c'est-à-dire un modèle abstrait représentant la manière dont les gens conçoivent les choses réelles dans le monde réel, avec des concepts et des relations dotés de noms et de définitions explicites [11].

Ainsi, les services web sémantiques combinent deux technologies : les services web et le web sémantique [1].

1.4.1 Langages de description de services web sémantiques

Les services web sémantiques doivent être décrits sémantiquement afin de faciliter leur utilisation par des machines. Les langages de description de service web tels que WSDL permettent de décrire les services de manière syntaxique. Cependant, pour faciliter la découverte, la sélection et la composition de services web sémantiques, il est nécessaire d'utiliser des langages de description de service web sémantiques tels que SAWSDL, WS-* ainsi que des approches basées sur des modèles sémantiques comme OWL-S et WSMO [1]. Ces langages permettent de décrire les services web sémantiques en utilisant des ontologies, ce qui rend les descriptions plus compréhensibles pour les machines et facilite leur automatisation.

1.4.1.1 OWL-S

OWL-S (Ontology Web Language for Service) [11] est une ontologie web destinée aux services web. Elle a été créée pour faciliter la découverte automatique, la sélection et la composition de services Web. OWL-S permet de décrire les services web de manière non-ambiguë et interprétable par les programmes [12]. L'ontologie OWL-S organise la description d'un service web en trois composants : le profil de service (Service Profile), le modèle de processus (Service Model) et les liaisons (Service Grounding). La Figure 1.5 illustre la structure de l'ontologie supérieure d'OWL-S.

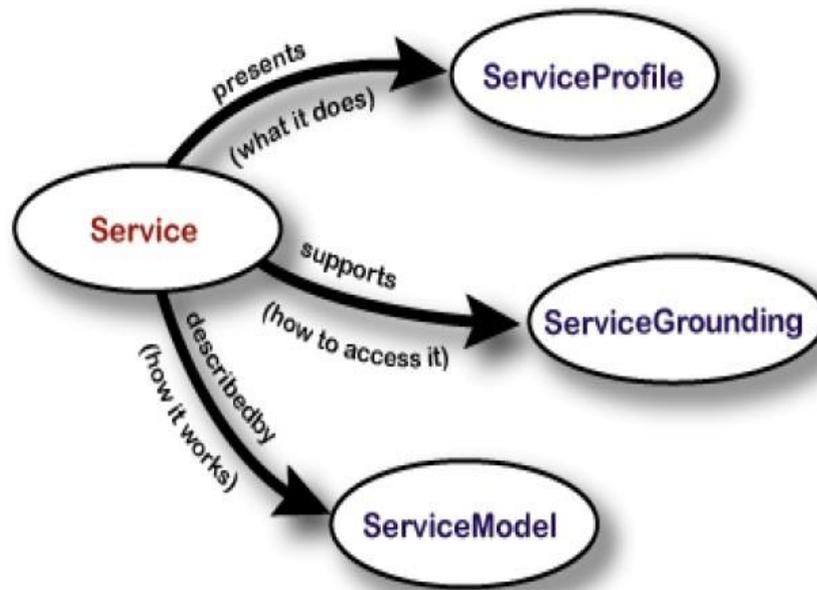


Figure 1. 5: Structure de l'ontologie de services OWL-S [1].

- **Service Profile** Le profil du service dans OWL-S indique ce que le service fait. Cette section est utilisée à la fois par les fournisseurs pour publier leurs services et par les clients pour spécifier leurs besoins afin de déterminer si le service répond à leurs exigences. Le Service Profile est utilisé pour décrire les propriétés fonctionnelles du service, telles que ses entrées, ses sorties, ses pré-conditions et ses effets, ainsi que ses propriétés non-fonctionnelles, telles que la confiance de service, la fiabilité, l'objet, le coût, etc.

- **Service Model** Le modèle de processus est une composante d'OWL-S qui décrit le fonctionnement interne d'un service web sémantique et comment l'utiliser. Il est utilisé pour décrire le modèle d'un service composite. En d'autres termes, le modèle de processus définit comment le service est exécuté et décrit les étapes nécessaires pour exécuter le service. OWL-S modélise les services en tant que processus avec des entrées et des sorties.

- **Service Grounding** Le Service Grounding, quant à lui, spécifie les détails techniques de la façon dont le service est exposé pour être invoqué. Il décrit comment interagir avec le service en fournissant des informations sur les protocoles de communication, les formats de message, les détails de transport, etc. Cette information est essentielle pour l'invocation et l'exécution automatiques des services, ainsi que pour leur intégration dans des architectures plus larges.

OWL-S définit trois types de processus : les processus atomiques (Atomic Process), simples (Simple Process) et composites (Composite Process). Chaque pr Les processus décrits dans OWL-S sont de trois types : atomiques, simples et composites. Chacun de ces processus possède des entrées, des sorties, des préconditions et des effets. Les processus atomiques sont simples et ne peuvent pas être décomposés en sous-processus, ils peuvent être exécutés en une seule étape. Les processus simples fournissent une vue abstraite d'un processus existant, mais contrairement aux processus atomiques, ils ne sont pas associés à un grounding. Les processus composites sont composés d'autres processus à l'aide de structures de contrôle telles que la séquence, la division, la division-jointe, l'exécution dans n'importe quel ordre, le choix, la condition si-alors, la répétition tant que, la répétition jusqu'à et l'itération. ocessus possède des entrées, des sorties, des préconditions et des effets. Les processus atomiques n'ont pas de sous-processus et peuvent être exécutés en une seule étape. Les processus simples fournissent une vue abstraite d'un processus existant. Cependant, à la différence des processus atomiques, un processus simple n'est pas associé à un grounding. Un processus composite est composé d'autres processus via les structures de contrôle telles que Sequence, Split, Split-Join, Any-Order, Choice, If-Then-Else, Repeat-While, Repeat-Until, et Iterate [1].

1.4.1.2 WSMO

WSMO (Web Service Modeling Ontology) est une ontologie conçue pour décrire les différentes facettes de la composition dynamique de services web, telles que la découverte, la sélection, la médiation et l'invocation. WSMO est basé sur WSMF (Web Service Modelling Framework), qui définit les principaux éléments de description sémantique des services web. Pour modéliser un service web, WSMO utilise le langage WSML (Web Service Modeling Language). WSMO propose quatre éléments clés pour modéliser les différents aspects des services web sémantiques : les ontologies, les médiateurs, les objectifs (goals), et les services (voir Figure 1.6).

- **WSMO Ontologies** Les ontologies dans WSMO jouent un rôle important dans la description sémantique des éléments appartenant à des domaines spécifiques en fournissant les concepts et les relations entre eux. Elles permettent de décrire les propriétés sémantiques des relations et des concepts en regroupant un ensemble d'attributs tels que les concepts, les relations, les fonctions, les instances et les axiomes. Les axiomes sont des règles logiques qui permettent de spécifier les contraintes et les règles qui régissent les concepts et les relations dans une ontologie WSMO.

Chapitre 1 : Services Web

- **WSMO Mediators** WSMO utilise un ensemble de médiateurs pour résoudre les incompatibilités détectées au niveau des données ou des processus. Ces incompatibilités peuvent être de nature structurelle, sémantique ou conceptuelle. Les médiateurs permettent ainsi de connecter les ressources WSMO hétérogènes en résolvant les différences entre les descriptions de services. Ils agissent comme des interfaces entre les différentes ressources en effectuant des transformations de données ou en fournissant des traductions entre les différentes ontologies utilisées.
- **WSMO goals** Dans WSMO, les buts sont utilisés pour décrire ce que l'utilisateur cherche à accomplir avec un service web. Les critères de recherche sont définis par l'utilisateur à travers la description du but, qui inclut les interfaces et les fonctionnalités souhaitées. Un médiateur est alors utilisé pour connecter le but demandé avec le service web correspondant.
- **WSMO Web services** WSMO utilise deux points de vue différents pour décrire un service web : la capacité (Capability) et l'interface (Interface), qui permettent de considérer différents aspects fonctionnels et comportementaux. La capacité décrit les propriétés fonctionnelles du service en décrivant les variables partagées (Shared Variables) entre les préconditions (Preconditions), les postconditions (Postconditions), les assumptions (Assumptions) et les effets (Effects). L'interface, quant à elle, décrit les opérations du service et leurs paramètres, ainsi que la manière dont le service peut être invoqué. Ces deux points de vue permettent une description complète et précise du service web [13].

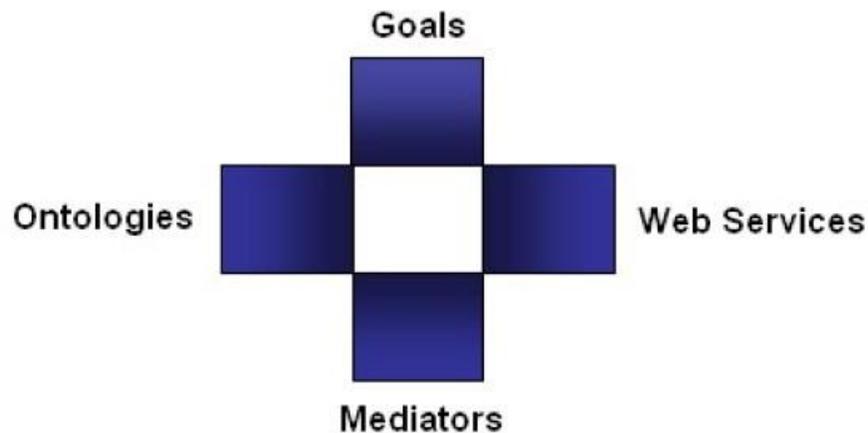


Figure 1. 6: Structure de WSMO [1].

1.5 Avantages des Services Web

La technologie des services web est populaire et couramment utilisée car elle offre des avantages intéressants pour les utilisateurs des systèmes distribués :

- Réduction du temps de mise en marché des services proposés par les entreprises.
- Amélioration de l'interopérabilité entre les logiciels écrits dans des langages différents et sur des plateformes différentes grâce aux normes utilisées.
- Utilisation de normes et protocoles ouverts pour assurer une meilleure intégration et une plus grande facilité d'utilisation.
- Possibilité de fonctionner avec des pare-feux sans nécessiter de modification des critères de filtrage grâce au protocole HTTP utilisé.
- Utilisation de protocoles et formats de données en format texte pour faciliter la compréhension et l'interprétation des échanges.
- Réduction des coûts grâce à l'automatisation interne et externe des processus commerciaux [15].

1.6 Inconvénients des Services Web

La technologie des services web comporte plusieurs inconvénients dont :

- Problèmes de sécurité : Les services web peuvent présenter des failles de sécurité et être vulnérables aux attaques de pirates informatiques. L'utilisation du protocole HTTP peut aussi poser des problèmes de sécurité, car les normes de sécurité des services web ne sont pas toujours suffisantes. Les technologies plus matures comme CORBA sont plus sécuritaires.
- Problèmes de performance : Les services web sont souvent moins performants que d'autres approches de l'informatique répartie, comme CORBA ou RMI. Cela est dû en partie à la surcharge de communication réseau et au traitement des données dans le service web.
- Confiance : Les relations de confiance entre différentes composantes d'un service web peuvent être difficiles à établir, car les composantes peuvent ne pas se connaître mutuellement. Cela peut poser des problèmes de sécurité et de fiabilité.
- Syntaxe et sémantique : Les services web se concentrent souvent sur la syntaxe pour invoquer des services, mais pas assez sur la sémantique pour décrire ce que les services

Chapitre1 : Services Web

web offrent. Cela peut entraîner des problèmes d'interopérabilité et de compréhension des services.

- **Fiabilité** : Il est difficile de garantir la fiabilité d'un service web, car cela dépend de la fiabilité de ses fournisseurs et des personnes qui l'invoquent. Des mécanismes de gestion d'erreur et de récupération doivent être mis en place pour minimiser les perturbations en cas de défaillance du service.
- **Disponibilité** : Les services web peuvent répondre à un ou plusieurs besoins du client, mais leur disponibilité et leur utilisation continue restent un défi pour les services web. Les mécanismes de gestion de la disponibilité doivent être mis en place pour minimiser les interruptions de service [5]. Dans ce chapitre, nous avons donné les caractéristiques des architectures orientées services et la technologie de services Web. Nous avons présenté les principales spécifications qui supportent la technologie des services web, et web sémantique

1.7 Conclusion

Dans ce chapitre, nous avons donné les caractéristiques des architectures orientées services et la technologie de services Web. Nous avons présenté les principales spécifications qui supportent la technologie des services web pour la construction rapide et à faible coût de nouvelles applications logicielles. Cependant, cette construction est rendue possible grâce au mécanisme de composition de services qui constitue actuellement un des challenges les plus importants de l'informatique orientée services. Le chapitre suivant sera consacré à la description du problème de la composition de services.

Chapitre 2 : Composition Des Services Web

Chapitre 2 : Composition Des Services Web

2.1 Introduction

La composition de services Web fait référence au processus de combinaison et de coordination d'un ensemble de services Web existants pour créer un service composite à valeur ajoutée qui peut effectuer des actions qui ne peuvent pas être remplies par un service autonome. Nous présentons dans ce chapitre, un survol sur le problème de composition des services Web, en décrivant ses concepts et ses approches

2.2 Concepts et définitions

La composition des services Web est le processus d'agrégation de plusieurs services en un seul service afin d'exécuter des fonctions plus complexes. Dans ce qui suit, nous introduisons quelques concepts et terminologies de base liés à la composition des services Web.

2.2.1 Services Web atomiques et services Web composites

Un service atomique (également appelé service élémentaire) est un point d'accès à une application qui ne s'appuie pas sur un autre service Web pour répondre aux requêtes des utilisateurs. Chaque service atomique fournit une interface de programmation basée sur SOAP et WSDL. Pour les applications héritées telles que celles écrites en CORBA, des adaptateurs appropriés peuvent être développés afin qu'ils puissent être invoqués en tant que services Web. Un service composite est une structure complète qui combine d'autres services composites et atomiques qui coopèrent pour effectuer un ensemble d'opérations. Les services regroupés par un service composite sont appelés services composants.

2.2.2 Propriétés fonctionnelles et non fonctionnelles d'un service

Les caractéristiques d'un service incluent à la fois ses propriétés fonctionnelles et non-fonctionnelles qui déterminent son identité. Comme pour tout autre logiciel, un service est caractérisé par les fonctionnalités qu'il peut offrir ainsi que la qualité avec laquelle il les exécute. Les propriétés fonctionnelles se rapportent au fonctionnement du service et sont généralement décrites en termes d'opérations. Ces opérations sont spécifiées par deux types d'informations : les entrées, appelées "Inputs", et les sorties, appelées "Outputs". Les inputs représentent les données requises pour l'exécution du service, tandis que les outputs représentent les données produites après son exécution. Pour compléter la description de l'aspect fonctionnel d'un service, d'autres extensions sémantiques peuvent être utilisées, telles que les préconditions et les postconditions.

Chapitre 2 : Composition Des Services Web

Les préconditions décrivent l'état initial nécessaire pour que le service puisse être exécuté, tandis que les postconditions décrivent les effets de l'exécution du service sur le monde [15].

Les propriétés non-fonctionnelles des services, également appelées "Qualités de Service" ou QoS, permettent d'évaluer la manière dont un service s'exécute du point de vue du fournisseur et de l'utilisateur. Elles sont définies par des paramètres mesurables tels que le temps de réponse, le débit, la disponibilité, la fiabilité, etc. Même si deux services offrent sémantiquement les mêmes fonctionnalités, ils peuvent être différenciés par leurs propriétés non-fonctionnelles.

2.2.3 Workflow

Un workflow est un ensemble d'étapes qui permet de combiner les fonctionnalités de différents services web (WS) pour répondre aux besoins de l'utilisateur. Dans ce processus, une activité regroupe plusieurs WS qui ont la même fonctionnalité, et un modèle représente la relation temporelle entre différentes activités. Dans cet article, nous examinons les trois modèles les plus courants : la séquence, le parallèle (AND) et le choix exclusif (XOR) [16].

Exemple 1 : Un exemple d'un tel flux de travail complexe est donné dans la figure 2.1, représentant un processus de planification de voyage. Pendant la première activité (A1), l'utilisateur souhaite réserver un billet d'avion. Ensuite, le processus est divisé, par un modèle XOR, en trois sous-processus possibles : soit l'utilisateur loue une voiture (A2) et réserve ensuite un hôtel (A3) avec un parking (A4) (avec un modèle AND), soit l'utilisateur demande à une agence de voyage d'organiser tout le voyage (A5), soit il réserve un taxi (A6) et un hôtel (A7).

Par conséquent, il existe trois itinéraires possibles de bout en bout pour ce processus de planification de voyage : [A1, A2, A3, A4], [A1, A5], [A1, A6, A7]. Dans l'itinéraire de bout en bout [A1, A2, A3, A4], A3 et A4 peuvent être effectuées en parallèle [16].

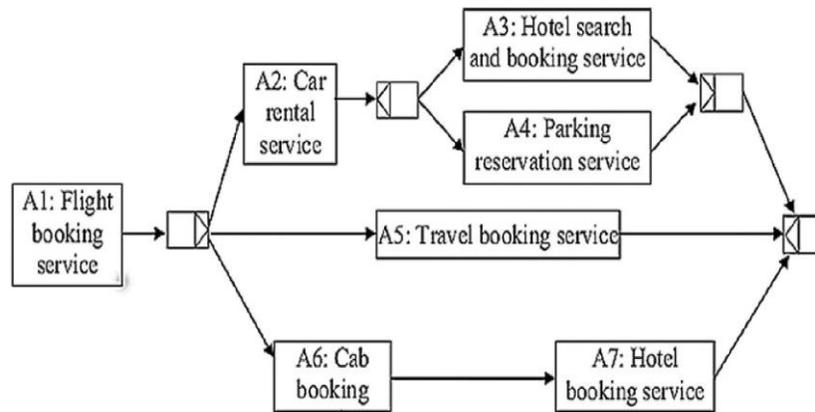


Figure 2. 1: Exemple de flux de travail complexe [16].

2.3 Processus de composition de services

La composition de services Web est un processus complexe qui implique l'intégration de plusieurs services Web existants pour fournir un service plus complexe et plus puissant. L'objectif de la composition de services est de réutiliser les services Web existants et de les combiner pour créer un processus plus complexe. Cependant, la composition de services Web est considérée comme une tâche très complexe pour trois raisons [17].

- La prolifération des services Web disponibles sur Internet entraîne une augmentation considérable du nombre de référentiels de services disponibles pour la recherche, ce qui rend la recherche de services appropriés plus difficile.
- Les services Web sont constamment mis à jour et évoluent, ce qui nécessite une composition dynamique au moment de l'exécution, ce qui peut entraîner des calculs intensifs.
- La mise en correspondance entre les services Web peut être difficile car différents fournisseurs développent des services Web dans des modèles différents.

La Figure 2.2 illustre le principe de la composition de services, qui consiste à partir d'un ensemble de services plus simples disponibles dans un registre, pour créer un service composite plus complexe et plus utile. Ce processus peut impliquer la coordination des services Web, la conception de flux de travail et la sélection des services appropriés pour répondre aux besoins spécifiques de l'utilisateur [15].

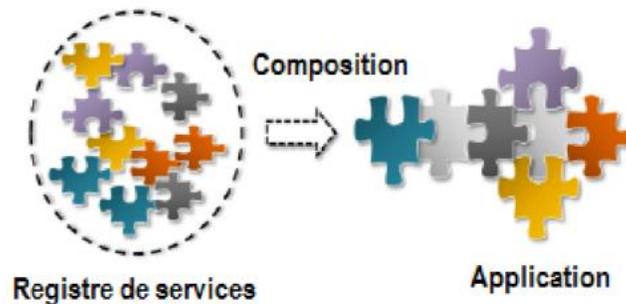


Figure 2. 2: Composition des services [15].

Le processus de composition de services comporte plusieurs phases, telles qu'illustrées dans la Figure 2.3 :

- La planification de la composition, qui consiste à spécifier le service requis et à le décomposer en un ensemble abstrait de tâches.
- La découverte des services, qui implique la recherche de services correspondant aux exigences de fonctionnalité et de non-fonctionnalité pour chaque tâche de la composition.
- La sélection des services ; parmi les multiples services découverts lors de la phase précédente, la sélection des services permet de choisir le service le plus approprié pour chaque tâche de la composition afin de satisfaire les besoins de l'utilisateur.
- L'exécution du service, où chaque tâche individuelle de la composition est appelée et exécutée pour proposer le service final.

Ce processus de composition peut être complexe et nécessite une coordination et une intégration efficaces de plusieurs services pour fournir un service composite complet et utile [17].

Chapitre 2 : Composition Des Services Web

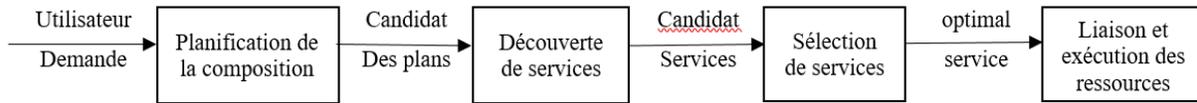


Figure 2. 3: Processus de composition des services [17].

2.4 Méthodes de composition de services

La composition de services a suscité un grand intérêt de la part de la communauté de recherche et industrielle, qui a proposé de nombreuses approches diverses qui peuvent être classées selon différents critères [15], tels que le degré de participation de l'utilisateur dans la définition du schéma de composition, avec des approches manuelles, semi-automatiques ou automatiques. On peut également les classer selon que la découverte des services et la spécification du workflow (le plan exécutable) soient réalisées avant ou après l'exécution de la requête de l'utilisateur, avec des approches statiques ou dynamiques (online ou offline). Enfin, la composition peut être abordée de deux manières différentes en fonction du contrôle des enchaînements d'appels des services participants : soit par un point central appelé orchestrateur, soit délégué aux services participants eux-mêmes, et dans ce cas, la composition sera appelée soit orchestration, soit chorégraphie [15].

2.4.1 Composition semi-automatique

La composition semi-automatique implique une combinaison d'intervention humaine et d'automatisation dans le processus de composition des services web. Dans ce cas, un utilisateur ou un concepteur spécifie les objectifs et les contraintes de la composition, puis utilise des outils ou des systèmes informatiques pour aider à la découverte, à la sélection et à l'assemblage des services web. L'utilisateur peut être assisté par des suggestions, des recommandations ou des techniques d'apprentissage automatique pour faciliter le processus de composition. Cependant, l'intervention humaine est nécessaire pour prendre des décisions finales et pour garantir que la composition répond aux besoins spécifiques [18].

Offre une vue d'ensemble des méthodes de composition semi-automatique des services web, en se concentrant principalement sur les systèmes de gestion de flux de travail. Il présente les différentes approches de composition, les techniques de modélisation des workflows, les algorithmes de composition automatique et les

Chapitre 2 : Composition Des Services Web

problèmes de recherche connexes. Il examine également les défis et les opportunités associés à la composition semi-automatique des services web.

2.4.2 Composition automatique

La composition automatique des services web implique une automatisation complète du processus de composition sans intervention humaine directe. Dans ce cas, des algorithmes et des systèmes informatiques sont utilisés pour découvrir, sélectionner et assembler automatiquement les services web en fonction des objectifs spécifiés. Ces algorithmes peuvent être basés sur des techniques d'intelligence artificielle, de raisonnement logique, d'optimisation ou de recherche heuristique. La composition automatique est généralement utilisée lorsque les contraintes et les objectifs de composition sont bien définis et que des connaissances préalables sur les services web sont disponibles.

[19] Propose une revue approfondie des méthodes de composition automatique des services web. Il examine les différentes approches et techniques utilisées pour automatiser le processus de composition, y compris les méthodes basées sur les recherches de planification, les approches basées sur les règles, les méthodes basées sur l'intelligence artificielle et les techniques basées sur l'apprentissage automatique. L'article présente également les principaux défis et les opportunités de recherche dans le domaine de la composition automatique des services web.

2.4.3 Composition manuelle

La composition manuelle des services web implique une intervention humaine complète dans le processus de composition. Dans ce cas, un utilisateur ou un concepteur examine manuellement les services web disponibles, identifie les services pertinents, spécifie les relations et les dépendances entre eux, et les assemble pour créer une composition fonctionnelle. La composition manuelle est souvent utilisée lorsque les besoins spécifiques de l'utilisateur sont complexes, ou lorsque les services web disponibles ne peuvent pas être facilement découverts ou sélectionnés automatiquement. Elle permet une plus grande flexibilité et un contrôle plus précis sur la composition finale des services.

[20] Présente une revue systématique de la littérature sur les approches de composition manuelle des services web. Il examine les différentes méthodes et

Chapitre 2 : Composition Des Services Web

techniques utilisées pour permettre aux développeurs de composer manuellement des services web en fonction de leurs besoins spécifiques. L'article analyse également les avantages, les défis et les bonnes pratiques associés à la composition manuelle des services web, en mettant l'accent sur la flexibilité et le contrôle offerts aux développeurs.

2.4.4 Composition statique

La composition statique des services web se réfère à une approche où la structure de composition est déterminée à l'avance et reste fixe pendant l'exécution. Les services sont préalablement sélectionnés et assemblés pour former une composition spécifique qui est ensuite déployée et utilisée. Cette approche convient lorsque les besoins et les exigences sont bien définis, et que la composition ne nécessite pas de changements fréquents. La composition statique peut être réalisée à l'aide de langages de modélisation spécifiques, tels que BPEL (Business Process Execution Language) ou BPMN (Business Process Model and Notation). [21] présente une enquête sur la composition statique des services web. Il examine les différentes approches et techniques utilisées pour réaliser la composition statique, où la structure du workflow est déterminée à l'avance et ne change pas dynamiquement. L'article discute des modèles de composition statique, des langages de description utilisés et des techniques d'optimisation associées à la composition statique des services web. Il met également en évidence les avantages, les limites et les défis de la composition statique des services web.

2.4.5 Composition dynamique

La composition dynamique des services web se réfère à une approche où la structure de composition peut être modifiée pendant l'exécution en fonction des besoins changeants. Les services peuvent être découverts, sélectionnés et assemblés dynamiquement en fonction des circonstances et des conditions en temps réel. Cette approche est souvent utilisée lorsque les besoins sont moins prédictibles ou lorsque de nouvelles opportunités ou contraintes apparaissent pendant l'exécution. La composition dynamique peut être réalisée à l'aide de techniques telles que la découverte de services, la planification automatique ou l'adaptation basée sur des

Chapitre 2 : Composition Des Services Web

règles.

[22] propose une enquête sur la composition dynamique des services web. Il examine les différentes approches et techniques utilisées pour réaliser la composition dynamique, où la structure du workflow peut changer dynamiquement en fonction des Conditions et des événements en cours d'exécution. L'article explore les modèles de composition dynamique, les langages de description utilisés et les mécanismes de coordination des services web. Il discute également des avantages, des défis et des opportunités de recherche dans le domaine de la composition dynamique des services web.

2.4.6 Orchestration

L'orchestration se concentre sur la coordination centralisée des services web dans un processus de composition. Dans ce cas, un service appelé orchestrateur (ou moteur d'orchestration) contrôle l'exécution des services en définissant l'ordre d'appel, les dépendances et les flux de données entre les services. L'orchestrateur agit comme un chef d'orchestre qui dirige l'exécution de la composition en contrôlant les messages et les données échangés entre les services. L'orchestration est généralement réalisée en utilisant un langage de modélisation tel que BPEL (Business Process Execution Language) pour spécifier le flux de travail et la coordination des services.

[23] Propose une enquête sur les techniques et les outils de composition par orchestration des services web. Il examine les différents aspects de la composition par orchestration, tels que la modélisation des workflows, les langages de composition, les moteurs d'exécution et les outils de développement. L'article discute également des défis et des opportunités de recherche dans le domaine de la composition par orchestration des services web.

2.4.7 Chorégraphie

La chorégraphie se concentre sur la coordination décentralisée des services web dans un processus de composition. Dans ce cas, chaque service est responsable de son propre comportement et communique directement avec d'autres services en utilisant des messages. Les services interagissent entre eux de manière

Chapitre 2 : Composition Des Services Web

collaborative et coopérative, sans avoir besoin d'un contrôleur centralisé. La chorégraphie définit les modèles d'interaction, les messages échangés et les séquences d'événements entre les services. Cette approche permet une plus grande flexibilité et autonomie des services, mais nécessite une coordination efficace pour garantir le bon déroulement de la composition.

[24] Cet article propose une enquête sur la composition par chorégraphie des services web en utilisant des chorales complètes. Il examine les différentes approches et techniques utilisées pour réaliser la composition par chorégraphie, où les services web collaborent de manière décentralisée pour atteindre un objectif commun. L'article discute des modèles de composition par chorégraphie, des langages de description utilisés et des mécanismes de coordination des services web. Il met également en évidence les avantages, les défis et les opportunités de recherche dans le domaine de la composition par chorégraphie des services web.

Approches de composition	Critère de classification
Manuelles	Degré de participation de l'utilisateur
Semi-automatiques	
Automatiques	
Statique	Spécification du workflow avant ou après l'exécution
Dynamique	
Orchestration	Contrôle des enchainements
Chorégraphie	

Figure 2. 4: Méthodes de composition de services.

2.5 Conclusion

Dans ce chapitre, nous avons donné une étude sur les principes de composition des services Web, les types de composition ainsi que quelques travaux présents dans la littérature. Cette étude nous permet de faciliter le développement d'un service Web composite qui sera l'objet de chapitre suivant.

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

3.1 Introduction

Ce projet s'inscrit dans le cadre de numérisation et digitalisation des systèmes d'information au sein de l'université Biskra qui est un exemple des universités algériennes. Nous nous intéressons au développement d'une application flexible pour l'inscription des nouveaux bacheliers (NBs) qui se base sur l'intégration dynamique des services concernant les inscriptions des NBs, et de les composer automatiquement pour répondre à des exigences qui ne sont pas réalisées par les services existants.

Dans ce chapitre, nous donnons un aperçu sur la procédure d'inscription de NBs afin de montrer la complexité du système d'inscription actuel. Nous décrivons ensuite notre proposition qui est basée sur une composition des services Web créées pour des tâches simples et qui collaborent par le partage des informations, en fournissant des services transparents aux NBs permettant de réduire le temps et les efforts de NBs de se déplacer entre les différents bureaux situés à l'intérieur et à l'extérieur de leurs universités accueillantes en réduisant les informations redondantes.

3.2 Procédure d'inscription des NBs

Les inscriptions des NBs à l'Université d'Algérie, sont organisées, chaque année, par une circulaire ministérielle qui définit :

- Les conditions et les règles générales qui président aux inscriptions des nouveaux bacheliers.
- Les domaines ou filières ouverts par établissement universitaire.
- Les séries de baccalauréat autorisées pour l'inscription dans chaque domaine ou filière.
- Les circonscriptions géographiques, ou wilayas rattachées à chacun des domaines (filière) ouvert par établissement.

Le processus d'inscription se déroule en plusieurs étapes selon un calendrier spécifié au niveau national. Les procédures comprennent l'inscription préalable, l'orientation et l'inscription finale.

Voici une explication détaillée de chacune d'entre elles :

- **Inscription préalable**

Le processus d'inscription préalable commence par la soumission d'une liste de vœux exclusivement en ligne sur le site dédié à l'inscription des nouveaux bacheliers qui peuvent choisir l'université et les spécialités de leur choix et les classer selon leurs préférences personnelles. Une période

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

est déterminée pour la soumission et la mise à jour de cette liste.

- **Orientation**

Le processus d'orientation est mis en œuvre au niveau national à l'aide d'un système informatique.

Les données fournies lors de l'inscription préalable sont utilisées pour orienter les bacheliers et déterminer les universités et les spécialités auxquelles ils peuvent être admis. Les critères d'orientation sont définis en fonction des résultats académiques et des capacités disponibles dans les universités.

- **Inscription finale**

Pour compléter l'inscription finale, le NB doit faire les activités suivantes :

A1 : Inscription définitive au niveau de scolarité centrale

Pendant ce processus, le NB doit faire déposer son dossier d'inscription, au niveau de scolarité centrale, qui est composé de :

- Original du relevé des notes du baccalauréat
- 02 copies, certifiées conformes, du relevé des notes du baccalauréat
- 06 photos d'identités récentes
- 02 extraits de naissance originaux
- L'attestation de confirmation de l'orientation
- Le payement des droits d'inscription

Après, l'inscription, il obtient les documents suivants :

- Sa carte d'étudiant
- Des certificats d'inscriptions destinés à lui permettre la confection des dossiers relatifs à la bourse, la bibliothèque et à la chambre universitaire, ainsi qu'à l'ouverture d'un compte auprès des Chèques Postaux

Avec les données fournies lors de l'inscription de l'inscription définitive au niveau de scolarité centrale, l'étudiant peut compléter les autres activités :

A2 : Inscription pour bénéficier d'une chambre universitaire.

A3 : Inscription pour bénéficier d'une carte bibliothèque

A4 : l'ouverture d'un compte auprès de la poste et bénéficier d'un chèque postal

A5 : Inscription au niveau de la CNAS pour bénéficier d'un numéro de sécurité sociale et la carte Chifa.

A6 : Inscription pour bénéficier d'une carte de transport universitaire

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

A7 : Inscription pour bénéficiaire de la bourse.

Ce processus est présenté à la Figure 3.1 et la séquence ordonnée d'activités ou les tâches nécessaires pour accomplir ce processus d'inscription sont illustrées par le workflow décrit dans Figure 3.2. Il est clair qu'il s'agit d'un processus qui prend du temps et des efforts car le NB doit visiter différents bureaux situés à l'intérieur et à l'extérieur de son université accueillante. En fait, la complexité de la procédure provient du manque d'échange d'informations entre les différents acteurs impliqués dans le processus de l'inscription. Cela signifie que le NB doit transférer les informations de chaque processus et fournir ces informations à chaque acteur. Par conséquent, la principale contribution d'un système informatique dans cette situation est de permettre de fournir des services transparents aux NBs en permettant une véritable coopération interservices qui permet le partage de services et d'informations, tout en réduisant la nécessité pour les étudiants de fournir des informations redondantes déjà détenues par les autorités. C'est exactement ce que notre proposition vise à faire.

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

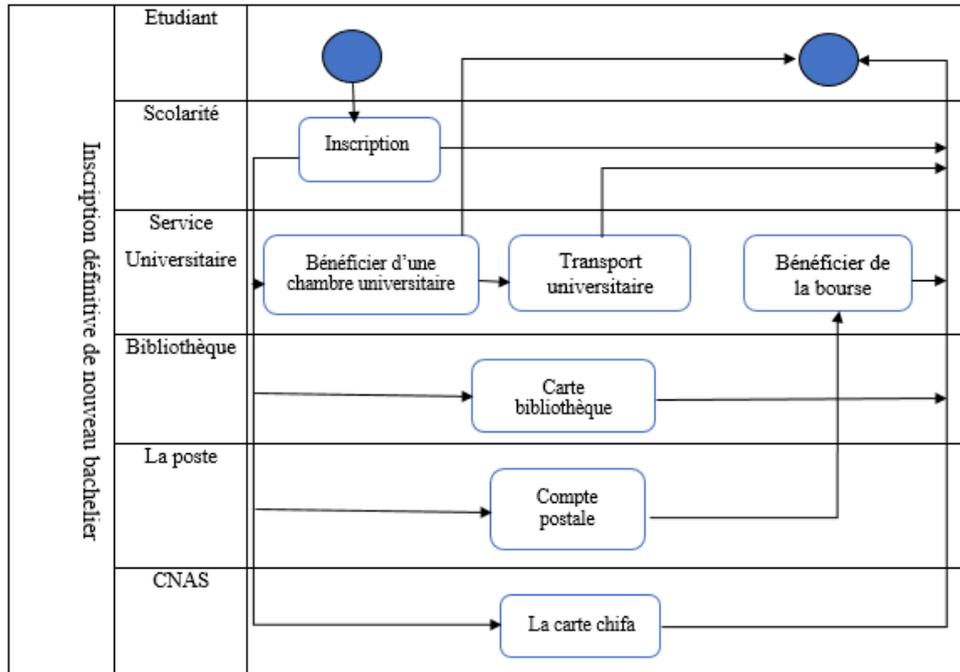


Figure 3. 1: Scénario de processus d'inscription de nouveaux bacheliers.

Le workflow suivant désigne la séquence ordonnée d'activités ou de tâches nécessaires pour accomplir ce processus d'inscription

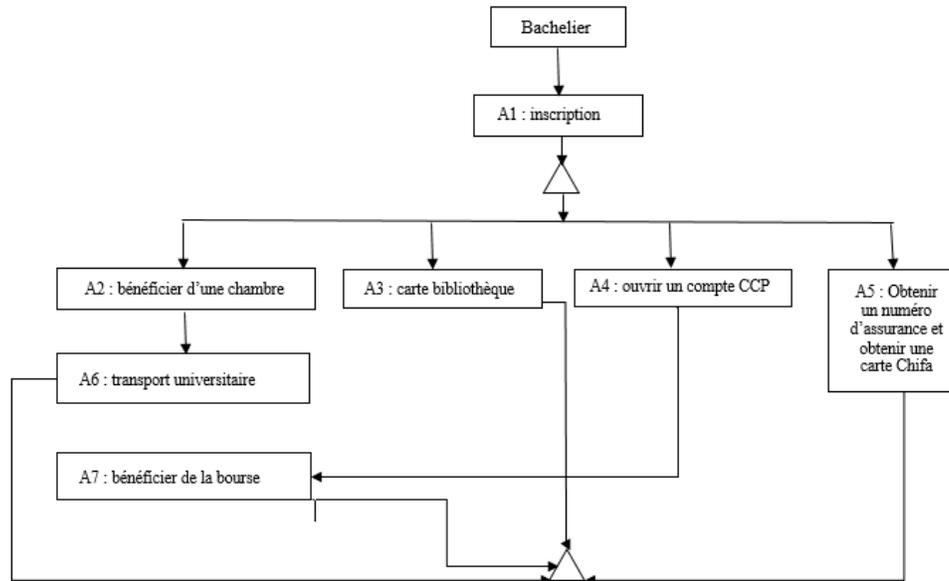


Figure 3. 2: Workflow désignant la séquence ordonnée d'activités du processus d'inscription.

3.3 Développement d'un service composite pour l'inscription des NB

La composition est un modèle de programmation dans lequel un ensemble de services web indépendants est fusionné et regroupé pour créer un seul service composite.

3.3.1 Environnement de travail

Nous avons opté d'utiliser les outils suivants pour la réalisation de l'application web : NetBeans IDE 8.2, GlassFish Server 4.1.1, Java EE 7 Web, jdk1.8, MySQL Workbench 8.0, Visual Studio Code, PHP.

- **NetBeans** est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web) [25].



Figure 3. 3: NetBeans logo [26].

- **GlassFish** est le nom du serveur d'applications Open Source Java EE 5 et d'aujourd'hui Java EE 7 avec la version 4.1 qui sert de socle au produit Oracle GlassFish Server4 (anciennement Sun Java System Application Server5 de Sun Microsystems). Sa partie Toplink persistence6 provient d'Oracle. C'est la réponse aux développeurs Java désireux d'accéder aux sources et de contribuer au développement des serveurs d'applications de nouvelle génération [26].
- **JDK** (Java Development Kit) est un logiciel édité par Sun pour le développement d'application en Java.

- **MySQL Workbench** est un logiciel de gestion de base de données MySQL. Disponible sous Windows, Mac et Linux, il permet de gérer des tables (ajout, modification, suppression) à travers une interface graphique simple d'usage [28].



Figure 3. 4: MySQL Workbench logo [29].

- **Visual Studio Code** est un éditeur de code open source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'autocomplétions, la coloration syntaxique, le débogage, et les commandes git [30].

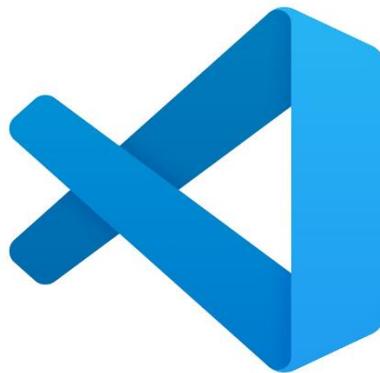


Figure 3. 5: Visual studio logo [31].

- **PHP** (officiellement, ce sigle est un acronyme récursif pour PHP Hypertext Preprocessor) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML [32].

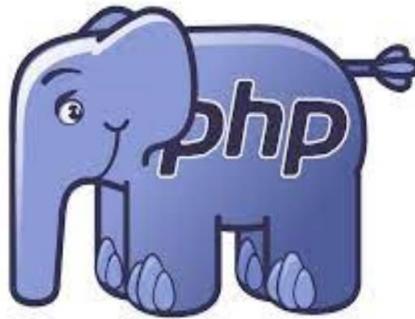


Figure 3. 6: PHP logo [33].

3.3.2 Création des services web

Nous avons utilisé Glassfish pour héberger notre code et exposer le service. Glassfish repose sur l'implémentation Metro qui prend en charge JAX-WS (Java API for XML Web Services) pour prendre en charge la traduction WSDL/Java et SOAP/Java. L'utilisation de l'API JAX-WS permet d'utiliser des métadonnées directement dans le code source pour transformer une classe et certaines de ses méthodes en web service. Ces métadonnées sont appelées Annotations.

Trois annotations sont utilisées :

- `@WebService` : définit la classe comme étant un web service, identifié par un nom.
- `@WebMethod` : expose une méthode de la classe en tant qu'opération proposée par le Web service, sous un nom donné.
- `@WebParam` : définit un paramètre de l'opération.

Dans les 6 services (inscription, bib, ccp, chifa, chambre, transport) il y'a 4 méthodes :

Register : Transférer les informations à la base de données et les conserver.

Update : Envoyer des informations à la base de données et les mettre à jour.

Delete : Envoyer des informations à la base de données et les supprimer.

Return : Renvoyer des informations contenant un code pour l'étudiant.

3.3.3 Création d'un service composite

La composition des services web se réfère au processus de création d'un service composite offrant une nouvelle fonctionnalité, à partir de services web existants plus simples en suivant le paradigme de l'orchestration des services Web qui consiste à combiner les services Web disponibles avec l'ajout d'un nouveau service 'central service' qui joue le rôle de coordinateur central chargé d'invoquer et de combiner la sous-activité unique.

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

Ce nouveau service web 'central service' agit en tant qu'intermédiaire entre l'utilisateur (le NB) et les autres services. Lorsque l'utilisateur soumet une demande à ce service, il la redirige vers tous les autres services et rassemble les données qui y sont répondues.

Le nouveau service (central service) enregistre également ces données dans une base de données. Cela signifie qu'il conserve une copie des données liées à tous les autres services dans sa propre base de données, ce qui facilite l'accès à ces données et leur manipulation ultérieure.

De plus, nous avons également créé une page pour le service de composition qui envoie toutes les données. Cette page facilite la distribution des données à tous les services concernés. Lors de la réception des données, la page les dirige vers chaque service web en fonction des besoins, ce qui permet une distribution efficace et organisée des données entre les différents services.

La Figure 3.7 illustre la gestion des transactions entre les services qui se fait selon le schéma suivant :

1. Dans chaque page Web écrite en HTML, il y a une partie du code qui gère la connexion et l'interaction (central service) avec un service spécifique (service targeted).
2. Le service cible (service cible) est appelé selon le besoin du site ou de l'application sur lesquels le développeur travaille, c'est-à-dire que nous utilisons un code ou une fonction côté serveur (Backend) qui est appelé par la page HTML, et le service appelé dépend des besoins du site ou l'application sur laquelle travaille le développeur.
3. Les informations sont envoyées à la base de données (Database) : après l'appel du service cible (Target Service), les informations (les données de l'utilisateur, les entrées ou le autres données) de la page HTML sont envoyées au service.
4. Effectuer des opérations d'enregistrement, de suppression, de modification ou de retour des informations ; cette étape dépend du type de service (service targeted) appelé et de l'opération à effectuer. Les opérations possibles peuvent être l'enregistrement (ajout) de nouvelles informations dans la base de données (database), la suppression de données existantes, la mise à jour des informations existantes avec de nouvelles valeurs ou la récupération d'une information depuis la base de données (database) et son affichage sur la page HTML.

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

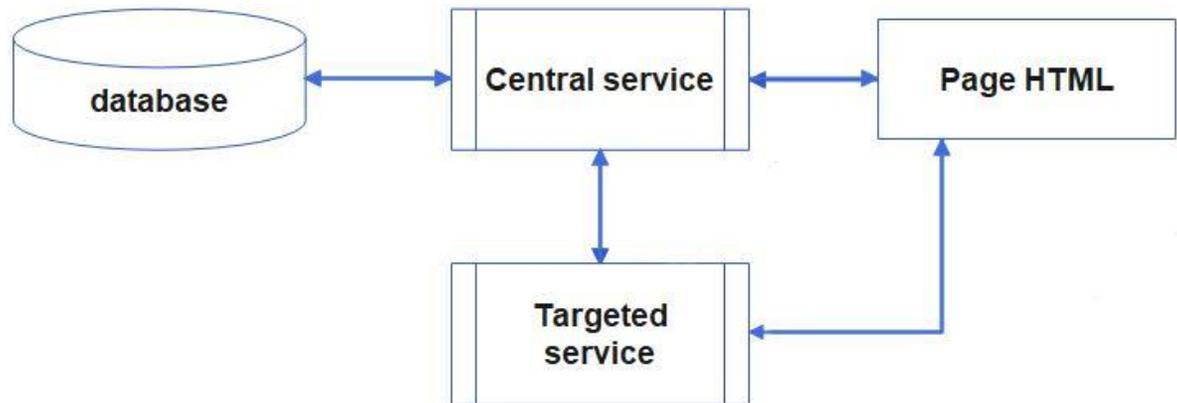


Figure 3.7: Gestion des transactions entre les services.

3.3.4 Architecture détaillée

L'architecture détaillée de notre système est décrite selon la Figure 3.8.

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

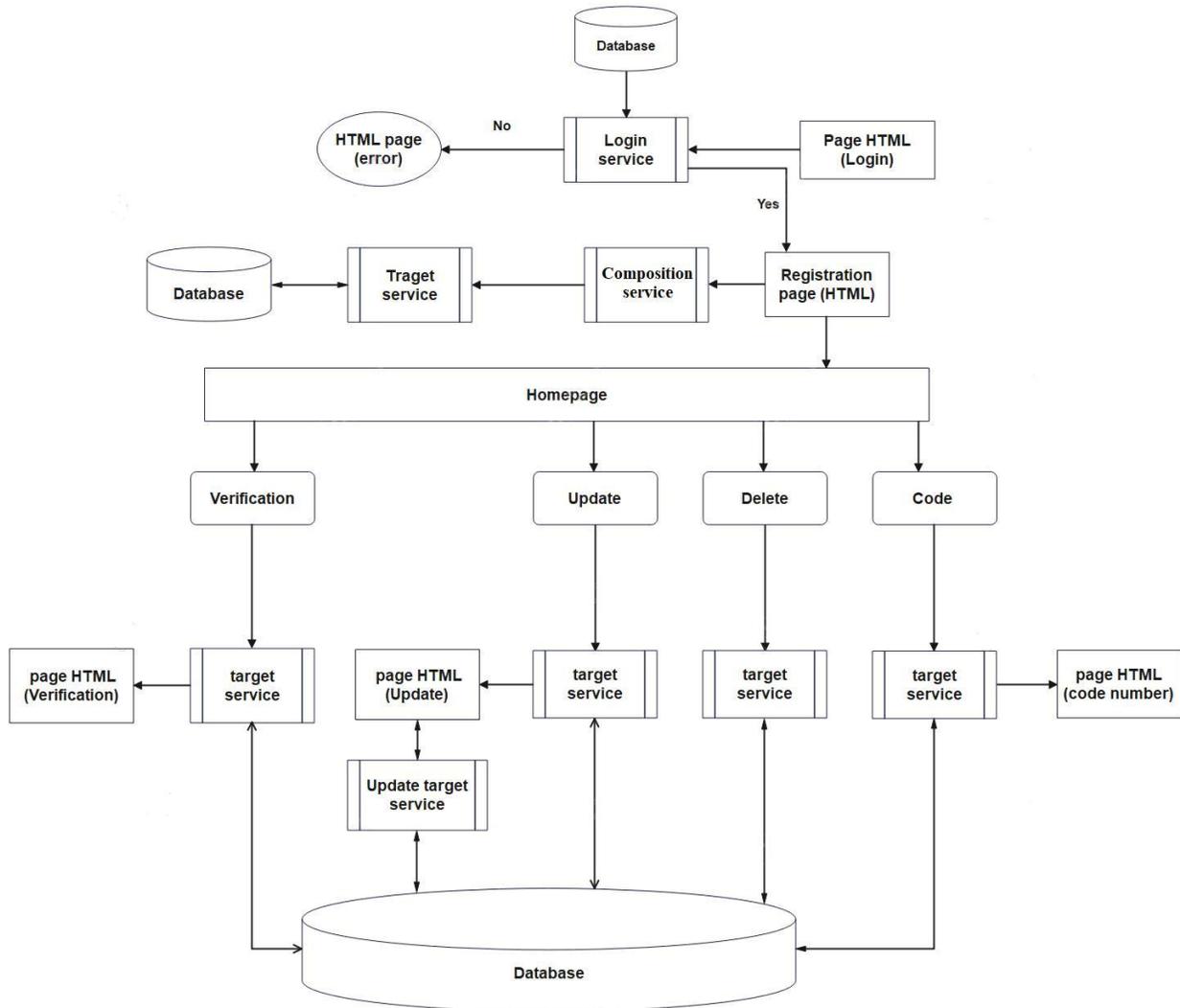


Figure 3. 8: Architecture détaillée.

Le principe de fonctionnement de notre système est présenté comme suit :

- Page de connexion (Page HTML : Login) : l'utilisateur (le nouveau bachelier) saisit son identifiant et son mot de passe qui lui ont été préalablement donné.
- Après s'être connecté, l'utilisateur se rend sur la page d'inscription (Registration page (HTML)), où il renseigne ses informations personnelles. Vous devez fournir des informations telles que votre carte d'assurance maladie, votre carte de séjour, vos informations de transport, vos informations sur les bourses d'études (le cas échéant) et une carte de bibliothèque (si demandée par la page). À ce stade, le service de configuration (Composition service) appelé par la page (Registration page (HTML)) s'exécute et envoie

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

les informations saisies par le l'utilisateur au service cible (service target).

- Le service cible (service target) reçoit les informations envoyées par le service d'intégration (Composition service) et les stocke dans sa propre base de données (database). De cette manière, on s'assure que les informations personnelles du l'utilisateur sont vérifiées et enregistrées de manière sécurisée et fiable dans la base de données (database). Cela permet de garantir que ces informations sont utilisées correctement et efficacement pour mener à bien le processus d'inscription, puis fournir les données nécessaires au l'utilisateur sur place.
- Vérification de la base de données (database) : Une fois que l'utilisateur clique sur le bouton de connexion, la page se connecte à un service chargé de vérifier les données de l'utilisateur. Le service vérifie la présence de données utilisateur dans la base de données (database).
- Notifier la page du résultat de la vérification : Si les données utilisateur n'existent pas dans la base de données (database), la page affiche un message d'erreur indiquant que la connexion a échoué, que le mot de passe est erroné ou que le nom d'utilisateur est incorrect. Si les informations sont correctes, l'utilisateur est redirigé vers la page d'accueil (Home page).
- Accueil : La page d'accueil (Home page) contient trois types de boutons.
- **Bouton vérification** : lorsque l'utilisateur clique sur le bouton de vérification, le système envoie une demande au service concerné pour récupérer ses informations personnelles. Ensuite, ces informations sont affichées sur la page de vérification pour le l'utilisateur.
- **Bouton update** : lorsque l'utilisateur clique sur ce bouton, la page appelle le service cible (service target). Pour récupérer les informations de l'utilisateur et les afficher sur la page d'actualisation (Page HTML (Update)). L'utilisateur peut modifier les informations, puis appuyer sur le bouton de mise à jour. Dans ce cas, la page contacte le service cible (service target). Pour mettre à jour les informations dans la base de données (database).
- **Bouton delete** : lorsque l'utilisateur clique sur ce bouton, la page se connecte au service cible (service target). Pour supprimer les données de l'utilisateur.
- **Bouton Code** : Lorsque l'utilisateur clique sur ce bouton, la page appelle le service cible (service target). Pour récupérer le code de la base de données (database) et l'afficher dans la page HTML (Page HTML (code number)).

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

Il s'agit d'une description détaillée du processus de connexion et de vérification et des types de boutons disponibles sur la page d'accueil (Home page).

3.4 Front end du système

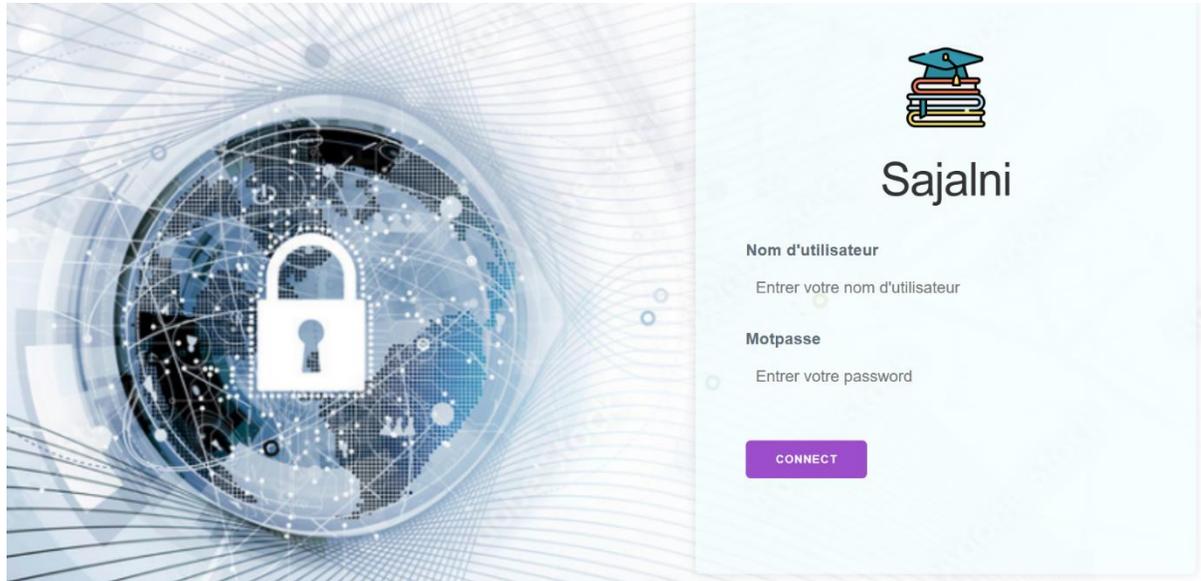


Figure 3. 9: Page login.

Figure 3.9 est une page de connexion où NB saisit son nom d'utilisateur et son mot de passe pour accéder à la page ou à la zone suivante qui lui permet d'accéder aux services disponibles dans le site web.

 Inscription

Prénom :

Nom :

Date naissance :

Lieu naissance :

Groupe sanguin :

Nationalité :

Figure 3. 10: Page inscription partie 1.

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

Adresse :

Wilaya :

Num telephone :

Sexe :

Numéro d'identification national :

Img :

 No file chosen

Moyenne de bac :

Figure 3. 11: Page inscription partie 2.

Mat bac :

Mat etudiant :

Specialite :

Code inscription:

Code niveau :

Code confirmation orientation :

Code paiement des droits inscription :

Figure 3. 12: Page inscription partie 3.

Cette page est considérée comme une interface interactive à travers laquelle l'utilisateur peut facilement saisir ses informations personnelles.

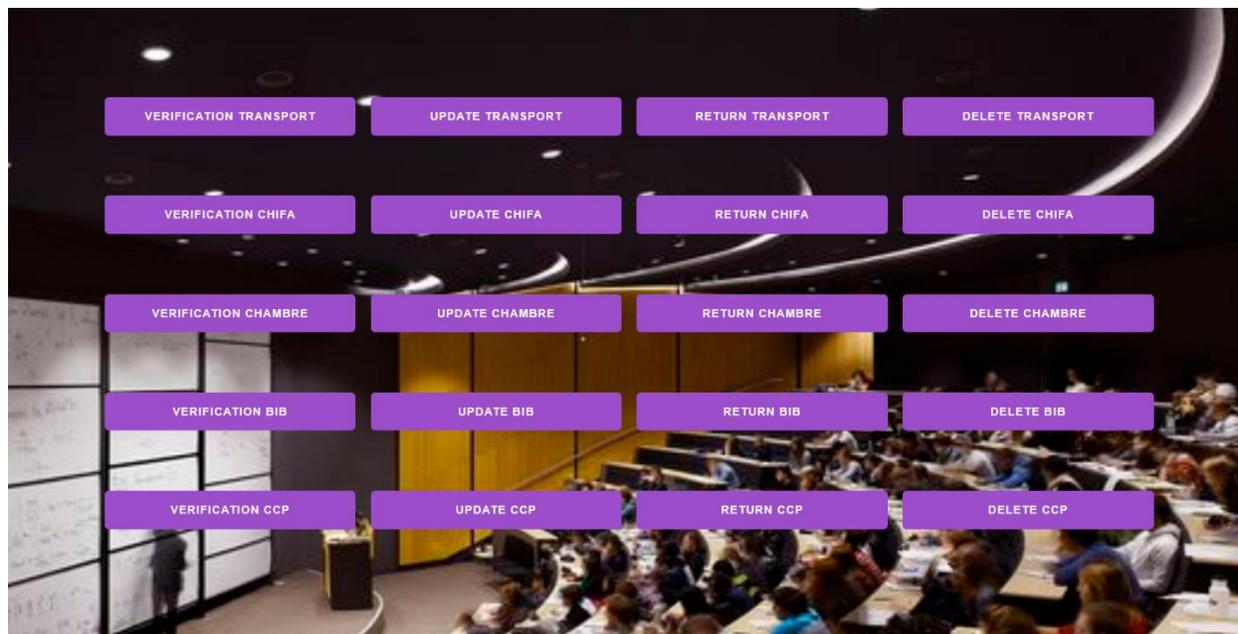


Figure 3. 13: Page home.

La Figure 3.13 représente la page principale du service ou de l'application en question. Permet à l'utilisateur de supprimer ses données du service cible en appuyant sur le bouton delete. Il peut également vérifier ses informations personnelles pour le service cible en appuyant sur le bouton vérification, ou les modifier en cas de saisie d'informations erronées en appuyant sur le bouton vérification. De plus, il peut demander le code de service cible en appuyant sur le bouton return.

- En appuyant sur le bouton "delete", les données de l'utilisateur sont supprimées définitivement du service ciblé, ce qui signifie que les informations personnelles et les autres données associées seront complètement supprimées.
- En ce qui concerne le bouton "vérification", il offre à l'utilisateur la possibilité de consulter et de valider ses informations personnelles, telles que son nom, son adresse ou toute autre donnée pertinente. Cela permet à l'utilisateur de s'assurer que les informations enregistrées sont exactes et à jour. De plus, grâce à cette fonctionnalité, l'utilisateur peut également apporter des modifications si nécessaire, garantissant ainsi l'exactitude et l'intégrité de ses informations personnelles. Le bouton "vérification" joue donc un rôle crucial dans l'autonomie de l'utilisateur en lui permettant de contrôler et de gérer activement ses propres données.

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

- En ce qui concerne le bouton "update", l'utilisateur peut mettre à jour ou modifier ses informations personnelles dans le service ciblé, telles que son nom, son adresse ou toute autre information.
- Par utilisation du bouton "return", on demande un code de service ciblé. Ce code peut être requis pour effectuer certaines actions dans le service, telles que la récupération du compte utilisateur ou la modification des paramètres de sécurité.

Update transport

Prénom :

Nom :

Numéro d'identification national :

Nationalité :

Adresse :

[S'INSCRIRE](#)

Figure 3. 14: Page update transport.

Après avoir cliqué sur le bouton "Update", l'utilisateur est redirigé vers une nouvelle page, comme indiqué ci-dessus, qui contient un formulaire ou une interface utilisateur capable d'afficher les données précédemment enregistrées. L'utilisateur peut modifier toute information présente sur la page, comme mettre à jour une valeur spécifique ou modifier une information particulière.

verification transport

Prénom :nora

Nom :guedouari

Numéro d'identification national :123456789012345

Nationalité :Algérienne

Adresse :57 rue saleh bey biskra

Figure 3. 15: Page verification.

Après avoir cliqué sur le bouton "Verification", l'utilisateur est redirigé vers une nouvelle page, comme indiqué ci-dessus, qui contient un formulaire ou une interface utilisateur capable d'afficher les données précédemment enregistrées. L'utilisateur vérifie la validité de ses informations.

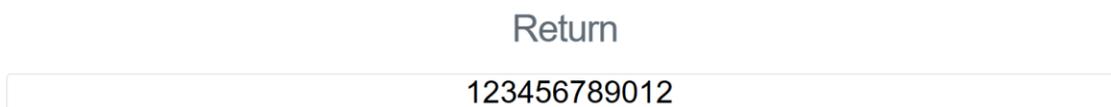


Figure 3. 16: Page return.

Lorsque vous cliquez sur le bouton "Return", la page de return s'affiche. Cette page contient un code ou une balise pour le service ciblé. À l'aide de ce code, nous pouvons obtenir des informations spécifiques sur le service ciblé, notamment sa carte d'identité.

En utilisant le code disponible sur la page de retour, nous pouvons accéder aux informations du service ciblé, telles que ses coordonnées de contact, ses données essentielles ou toute autre information pertinente. Ce code nous permet de reconnaître le service et de récupérer sa carte d'identité, qui est un ensemble d'informations décrivant le service de manière générale.

Voici quelque capture d'écran sur les fichiers WSDL de nos services web.

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

```
▼ <definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://web/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://web/" name="chifa">
  ▼ <types>
    ▼ <xsd:schema>
      <xsd:import namespace="http://web/" schemaLocation="http://localhost:34037/NOURA/chifa?xsd=1"/>
    </xsd:schema>
  </types>
  ▼ <message name="register">
    <part name="parameters" element="tns:register"/>
  </message>
  ▼ <message name="registerResponse">
    <part name="parameters" element="tns:registerResponse"/>
  </message>
  ▼ <message name="update">
    <part name="parameters" element="tns:update"/>
  </message>
  ▼ <message name="updateResponse">
    <part name="parameters" element="tns:updateResponse"/>
  </message>
  ▼ <message name="delete">
    <part name="parameters" element="tns:delete"/>
  </message>
  ▼ <message name="deleteResponse">
    <part name="parameters" element="tns:deleteResponse"/>
  </message>
  ▼ <message name="returne">
    <part name="parameters" element="tns:returne"/>
  </message>
  ▼ <message name="returneResponse">
    <part name="parameters" element="tns:returneResponse"/>
  </message>

```

Figure 3. 17: Fichier WSDL partie 1.

```
▼ <portType name="chifa">
  ▼ <operation name="register">
    <input wsam:Action="http://web/chifa/registerRequest" message="tns:register"/>
    <output wsam:Action="http://web/chifa/registerResponse" message="tns:registerResponse"/>
  </operation>
  ▼ <operation name="update">
    <input wsam:Action="http://web/chifa/updateRequest" message="tns:update"/>
    <output wsam:Action="http://web/chifa/updateResponse" message="tns:updateResponse"/>
  </operation>
  ▼ <operation name="delete">
    <input wsam:Action="http://web/chifa/deleteRequest" message="tns:delete"/>
    <output wsam:Action="http://web/chifa/deleteResponse" message="tns:deleteResponse"/>
  </operation>
  ▼ <operation name="returne">
    <input wsam:Action="http://web/chifa/returneRequest" message="tns:returne"/>
    <output wsam:Action="http://web/chifa/returneResponse" message="tns:returneResponse"/>
  </operation>
</portType>
▼ <binding name="chifaPortBinding" type="tns:chifa">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  ▼ <operation name="register">
    <soap:operation soapAction=""/>
    ▼ <input>
      <soap:body use="literal"/>
    </input>
    ▼ <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  ▼ <operation name="update">
    <soap:operation soapAction=""/>
  </operation>

```

Figure 3. 18: Fichier WSDL partie 2.

Chapitre 3 : Développement d'une application flexible pour l'inscription des NBs

```
▼<input>
  <soap:body use="literal"/>
</input>
▼<output>
  <soap:body use="literal"/>
</output>
</operation>
▼<operation name="delete">
  <soap:operation soapAction=""/>
  ▼<input>
    <soap:body use="literal"/>
  </input>
  ▼<output>
    <soap:body use="literal"/>
  </output>
</operation>
▼<operation name="returne">
  <soap:operation soapAction=""/>
  ▼<input>
    <soap:body use="literal"/>
  </input>
  ▼<output>
    <soap:body use="literal"/>
  </output>
</operation>
</binding>
▼<service name="chifa">
  ▼<port name="chifaPort" binding="tns:chifaPortBinding">
    <soap:address location="http://localhost:34037/NOURA/chifa"/>
  </port>
</service>
</definitions>
```

Figure 3. 19: Fichier WSDL partie 3.

3.5 Conclusion

Dans ce chapitre, nous avons donné une description détaillée de développement de notre système, en présentant les outils de développement utilisés et les étapes de réalisation d'une application permettant de fournir des services transparents aux NBsn en se basant sur une véritable coopération et le partage des informations, tout en réduisant la complexité du système d'inscription actuel.

Conclusion générale

Conclusion générale

La composition des services web offre une approche puissante pour créer des fonctionnalités plus complexes à partir de services individuels. Les différentes méthodes de composition et les techniques associées offrent des options adaptées aux besoins spécifiques de chaque application.

Dans ce projet, nous avons exploité le domaine de composition des services web pour développer une application flexible pour l'inscription des nouveaux bacheliers (NBs) qui se base sur l'intégration dynamique des services concernant les inscriptions des NBs, et de les composer automatiquement pour répondre à des exigences qui ne sont pas réalisées par les services existants. Pour réaliser ce travail, nous avons fait une étude détaillée sur les concepts de base des services web, et de composition des services web, en explorant les outils de développement d'un service composite. Nous avons également exploré la procédure courante de l'inscription de NBs, afin de montrer la complexité du système d'inscription actuel et d'extraire les informations nécessaires pour développer notre application.

Pour réaliser la composition, nous avons utilisé le paradigme de l'orchestration des services Web qui représente un processus opérationnel centralisé unique (l'orchestrateur) qui coordonne l'interaction entre les différents services et chargé d'invoquer et de combiner les services.

Bien que la composition des services fournisse à l'application de l'inscription des NB des services complexes permettant de réduire le temps et les efforts de NBs, de se déplacer entre les différents bureaux situés à l'intérieur et à l'extérieur de leurs universités accueillantes, en réduisant les informations redondantes, cette technique présente certaines préoccupations telles que la sécurité et la qualité de service. En effet, en impliquant plusieurs services venant interagir avec plusieurs fournisseurs dans le processus de composition, le problème de sécurité augmente avec sa gestion. De plus, la performance et la qualité de service restent un défi dans la composition automatisée, ce qui ouvre la voie à de nombreuses applications potentielles dans ce contexte.

References

References

- [1] A. MERZOUG, Vers une composition automatique Thèse de DOCTORAT, UNIVERSITÉ ABOU BEKR BELKAID TLEMCEN, 03 Juillet 2018.
- [2] M. Huhns, Service-Oriented Computing: Key Concepts and Principles, University of South Carolina, North Carolina State University, IEEE INTERNET COMPUTING, 2005.
- [3] Thomas Erl, SOA: Principles of Service Design, Prentice Hall Upper Saddle River., 2008.
- [4] J. O'Sullivan, What's in a Service? Towards Accurate Description of Non-Functional Service Properties, Innovation, Queensland University of Technology, GPO Box 2434, Brisbane QLD 4001, Australia, 2002.
- [5] T. Andrews, Business Process Execution Language for Web Services, 2003.
- [6] J. O'Sullivan, What's in a service Distrib. Parallel Databases ?, 2002.
- [7] D. Booth, Web Services Architecture, February 2004.
- [8] Chinnici, Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, 2007.
- [9] T. BERNERS-LEE, Artist, *Scientific American: Feature Article: The Semantic Web: May 2001 The Semantic Web A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities.* [Art]. 2001.
- [10] D. MEKHZOUNI, Reformulation de requêtes pour la découverte des services web sémantiques, Mémoire magistère en Informatique option Réseaux et Systèmes Distribués, Université Abderrahmane Mira de Bejaia, 2009.
- [11] B. Martin.D, Bringing semantics to web services with owl-s. *World Wide Web*, 10(3) :243–277., 2007.
- [12] Claro.D, Approaches of web services composition. In *International Conference on Enterprise Information Systems*, pages 208–213., 2005.
- [13] K. U. Roman. D, Web service modeling ontology. *Applied ontology*, 1(1) :77–106., 2005.
- [14] e. M. Z. C. B. ALILI Sofian, "La sélection des services Web À base d' "algorithme de la recherche Harmonique" Mémoire MASTER., Université Abou Bekr Belkaid Tlemcen, 2015.
- [15] T. Nacéra, La composition dynamique des Services Web -Thèse de Doctorat, Université des

Sciences et de la Technologie Houari Boumediene, 19-10-2019.

- [16] M. M. V. Gabriel, Contents lists available at ScienceDirect Discrete Applied Mathematics Web services composition: Complexity and models, Université Paris-Dauphine, 75775 Paris Cedex 16, France, 15-11-2014.
- [17] A. A. Touri, WEB SERVICE COMPOSITION PROCESSES, Department of Computer Science, King Saud University, Riyadh, KSA, March 2016.
- [18] H. W. Xumin Liu, A Survey on Automated Service Composition Methods: From the Perspective of Workflow Management Systems", IEEE Transactions on Services Computing, 2015.
- [19] Z. Xiaohui Zhao, Automated Web Service Composition Survey and Perspective, IEEE Transactions on Services Computing, 2013.
- [20] S. A. M. Fahmidehkhoshnoud, Manual Service Composition Approaches: A Systematic Literature Review, IEEE Access, 2019.
- [21] A. K. Bouchaib Falah, Static Web Service Composition: A Survey, Procedia Computer Science, 2017.
- [22] K. S. Simon Dobson, Dynamic Web Service Composition: A Survey, International Journal of Web Services Research, 2005.
- [23] B. Liangzhao Zeng, Web Service Composition: A Survey of Techniques and Tools, ACM SIGMOD Record, 2004.
- [24] D. Crawl, Web Service Composition by Full-Fledged Choruses: A Survey, ACM SIGMOD Record, 2006.
- [25] "NetBeans," wikipedia, 18 6 2023. [Online]. Available: <https://fr.wikipedia.org/wiki/NetBeans>.
- [26] "Integration with NetBeans IDE," virtuozzo, 18 6 2023. [Online]. Available: <https://www.virtuozzo.com/company/blog/netbeans-ide-integration/>.
- [27] "GlassFish," wikipedia, 18 6 2023. [Online]. Available: <https://fr.wikipedia.org/wiki/GlassFish>.
- [28] "Cours et tutoriels sur le langage SQL," sql, 18 6 2023. [Online]. Available: <https://sql.sh/logiciel/mysqlworkbench#:~:text=MySQL%20Workbench%20est%20un%20logiciel,interface%20graphique%20simple%20d'usage>.
- [29] "MySQL Workbench," macupdate, 18 6 2023. [Online]. Available: <https://www.macupdate.com/app/mac/31829/mysql-workbench>.

- [30] “Visual Studio Code,” framalibre, 18 6 2023. [Online]. Available: <https://framalibre.org/content/visual-studio-code>.
- [31] “Visual Studio Code,” wikipedia, 18 6 2023. [Online]. Available: https://fr.wikipedia.org/wiki/Visual_Studio_Code.
- [32] “Qu'est ce que PHP?,” php, 18 6 2023. [Online]. Available: [https://www.php.net/manual/fr/intro-whatis.php#:~:text=PHP%20\(officiellement%2C%20ce%20sigle%20est,%C3%AAtre%20int%C3%A9gr%C3%A9%20facilement%20au%20HTML](https://www.php.net/manual/fr/intro-whatis.php#:~:text=PHP%20(officiellement%2C%20ce%20sigle%20est,%C3%AAtre%20int%C3%A9gr%C3%A9%20facilement%20au%20HTML).
- [33] «PHP, conformité et validation,» alsacreations, 18 6 2023. [En ligne]. Available: <https://www.alsacreations.com/article/lire/559-php-conformite-validation.html>.
- [34] N. L. Y. Mitra, “SOAP Version 1.2 Part 0: Primer (Second Edition) W3C Recommendation,” 27 April 2007. [Online]. Available: <https://www.w3.org/TR/soap12-part0/>. [Accessed 8 6 2023].