

**République Algérienne Démocratique et Populaire**  
**Ministère de l'enseignement supérieur et de la recherche scientifique**  
**Université de 8 Mai 1945 – Guelma -**  
**Faculté des Mathématiques, d'Informatique et des Sciences de la matière**  
**Département d'Informatique**



***Mémoire de Fin d'études Master***

**Filière : Informatique**

**Option : Sciences et Technologie de l'Information et de la Communication**

(STIC)

**Thème :**

---

**Conception et développement d'une plateforme d'apprentissage  
adaptative basée sur les microservices**

---

**Présenté par : *Soudani Aya***

**Membres du jury :**

- **Président** : Dr. Gouasmi Noureddine
- **Encadrant** : Dr. Séridi Ali
- **Examineur** : Dr. Bendjebar Safia

**Juin 2024**

# **Remerciements**

*Alhamdoulillah, je suis finalement arrivé ici, et j'ai pu finaliser ce travail et réaliser l'un de mes objectifs dans la vie.*

*Tout d'abord, je voudrais remercier Allah qui m'a guidé jusqu'ici malgré les obstacles que seul le Tout-Puissant connaît. Il m'a donné la force, les connaissances, la capacité et la possibilité de réaliser cette étude de recherche, de la poursuivre et de la terminer de manière satisfaisante.*

*Je tiens à remercier monsieur « **Séridi Ali** », mon encadrant, pour m'avoir bien encadré et suivi durant mon travail, ses précieux conseils, sa disponibilité et ses remarques pertinentes qui m'ont permis d'améliorer et de réaliser ce travail.*

*Je remercie également tous les membres du jury pour avoir accepté d'examiner ce travail. Leurs commentaires et suggestions précieux ont considérablement enrichi cette recherche.*

*Je suis reconnaissant envers notre chef de département, Dr Zineeddine Kouahla, ainsi que tous les professeurs du département d'informatique de l'Université du 8 mai 1945 de Guelma, pour leur soutien et leurs enseignements précieux. J'ai eu la chance d'avoir des enseignants respectés et dévoués.*

*Je tiens également à remercier mes amies, mes collègues de l'université 08 mai 1945 Guelma, toute la promotion 2024 du département d'informatique et tous ceux qui m'ont aidé et contribué de près ou de loin à la réalisation de ce travail.*

*Enfin, un grand merci à mes parents et toute ma famille pour leur amour, leur patience, leurs sacrifices et leur soutien pendant les moments les plus difficiles de ma vie, et tout au long de mon parcours éducatif. Que Dieu vous donne la santé, le bonheur, et veille à ce que je ne vous laisse jamais tomber.*

# Dédicaces

Avant tout, je voudrais remercier **Dieu** de m'avoir donné la santé, le courage pour pouvoir réaliser ce modeste travail.

**J'ai le grand plaisir de dédier ce modeste travail.**

En premier lieu ; à la lumière de mes yeux, toute ma gratitude ; que Dieu les gardes ;

Mes parents :

**Mon très cher père**, ma force, mon soutien moral celui qui s'est toujours sacrifié pour me voir réussir, j'espère que vous êtes fier de moi.

**Ma chère mère**, la source de mes efforts, ma vie et mon bonheur. Que dieu vous bénisse chère maman.

A mes plus chers frères, **Ibrahim** et **Hamza**, et A mes plus chères sœurs, **Zahira**, **Linda**, **Wafa**, **Samira** qui me donnent tendresse et amour et qui veulent me voir toujours heureuse et réussie, et qui m'ont toujours encouragé pour être la meilleure. Que Dieu vous bénisse. À mes belles-sœurs **Fatima** et **Amani** qui m'ont toujours encouragé. Que Dieu leur accorde le succès.

A mes neveux **Ahmed Zakaria**, **Siradje eddine** et à mes nièces **Nesrine**, **Chaima**, **Hadil**, **Soundes**, **Rahaf**, **Iline** qui sont le bonheur de ma vie.

A tous les membres de ma famille **Soudani**.

A mes chères amies **Salsabil Khelaifia**, **Touahri Sérine** les personnes qui m'a fait toujours rire est toujours à mes cotés et m'a encouragé dans les moments difficiles.

A mes meilleures amies et à tous mes collègues de Master 2 STIC PROMO 2019-2024. Merci d'avoir rendu cette année si mémorable et pleine de rire, vous avez transformé même les jours les plus difficiles en quelque chose d'amusant. Nous avons créé des souvenirs qui dureront toute une vie. En leur souhaitant un avenir très brillant.

A tous ceux qui m'ont aidé de loin ou de près durant mes études.

*Aya Soudani*

# ***Résumé***

L'évolution du domaine de l'apprentissage en ligne au cours des dernières années a été significativement renforcée par l'adoption des nouvelles technologies de développement .parmi ces technologies nous trouvons les microservices. ces derniers permettent de décomposer les applications complexes en modules autonomes et interconnectés,

Ce projet vise à la réingénierie d'une plateforme d'apprentissage traditionnelle en une architecture de microservices, pour améliorer la flexibilité, la scalabilité et la maintenabilité. En utilisant des microservices, chaque fonctionnalité est décomposée en modules autonomes, permettant une évolution rapide et des adaptations précises. L'objectif est de développer une solution e-learning modulaire, réutilisable et adaptable, favorisant la réutilisation et la distribution. Ce projet démontre la viabilité des microservices pour créer des solutions éducatives performantes et évolutives.

**Mots clés :** microservices, e-learning, apprentissage en ligne, web service.

# *Abstract*

The evolution of the field of online learning in recent years has been significantly reinforced by the adoption of new development technologies. Among these technologies we find microservices. the latter make it possible to break down complex applications into autonomous and interconnected modules,

This project aims to reengineer a traditional learning platform into a microservices architecture, to improve flexibility, scalability and maintainability. Using microservices, each functionality is broken down into autonomous modules, allowing rapid evolution and precise adaptations. The objective is to develop a modular, reusable and adaptable e-learning solution, promoting reuse and distribution. This project demonstrates the viability of microservices to create efficient and scalable educational solutions.

**Keywords:** microservices, e-learning, online learning, web service.

## ملخص

لقد تم تعزيز تطور مجال التعلم عبر الإنترنت في السنوات الأخيرة بشكل كبير من خلال اعتماد تقنيات التطوير الجديدة ومن بين هذه التقنيات نجد الخدمات الصغيرة. هذا الأخير يجعل من الممكن تقسيم التطبيقات المعقدة إلى وحدات مستقلة ومتراصة،

يهدف هذا المشروع إلى إعادة هندسة منصة التعلم التقليدية إلى بنية الخدمات الصغيرة، لتحسين المرونة وقابلية التوسع وقابلية الصيانة. باستخدام الخدمات الصغيرة، يتم تقسيم كل وظيفة إلى وحدات مستقلة، مما يسمح بالتطور السريع والتكيف الدقيق. والهدف هو تطوير حل للتعلم الإلكتروني معياري وقابل لإعادة الاستخدام وقابل للتكيف، وتشجيع إعادة الاستخدام والتوزيع. يوضح هذا المشروع جدوى الخدمات المصغرة لإنشاء حلول تعليمية فعالة وقابلة للتطوير

**الكلمات المفتاحية:** الخدمات المصغرة، التعلم الإلكتروني، التعلم عبر الإنترنت، خدمة الويب

# Table des matières

Liste des tableaux .....	vi
Liste des figures .....	vii
Introduction générale.....	1
Chapitre 1.....	3
Architecture orientée service (SOA) et architecture micro-service.....	3
1.1 Introduction.....	4
1.2 Concepts et notions de base .....	4
1.2.1 L'architecture orientée service.....	4
1.2.2 Le service.....	5
1.2.3 Les micro-services.....	7
1.2.4 La différence entre l'architecture orientée services et micro-services .....	13
1.2.5 Les approches "Top down " et "Button up " .....	14
1.3 Conclusion .....	14
Chapitre 2.....	15
Plateforme e-learning et les micro-services.....	15
2.1 Introduction.....	16
2.2 E-learning.....	16
2.2.1 Définition.....	16
2.2.2 Les avantages du e-learning.....	17
2.2.3 Les inconvénients du e-learning.....	17
2.2.4 Les principaux acteurs d'une plateforme d'e-learning .....	17
2.3 Analyse des travaux sur utilisation des micro-services dans le e-learning .....	18
2.4 Conclusion .....	23
Chapitre 3.....	24
Conception du système .....	24
3.1 Introduction.....	25
3.2 Objectifs.....	25
3.3 Processus d'identification des micro-services .....	26
3.4 Architecture générale d'application.....	27

3.5	Les schémas fonctionnels .....	28
3.5.1	Espace apprenant .....	28
3.5.2	Espace enseignant .....	32
3.5.3	Espace administrateur .....	33
3.5.4	Espace tuteur .....	33
3.6	Duplication de la BDD.....	36
3.7	Le MCD (Modèle Conceptuel de Donnée) .....	37
3.8	Le MLD (Modèle Logique de Donnée) .....	39
3.9	Conclusion .....	41

## **Chapitre 4.....42**

### **Implémentation du système.....42**

4.1	Introduction.....	43
4.2	Environnements logiciels de développement.....	43
4.2.1	HTML.....	43
4.2.2	CSS.....	44
4.2.3	JavaScript.....	44
4.2.4	PHP.....	44
4.2.5	La bibliothèque NuSOAP.....	45
4.2.6	Visual Studio Code.....	45
4.2.7	MySQL.....	45
4.2.8	XAMPP .....	46
4.3	Description du fonctionnement de notre projet.....	46
4.3.1	La configuration .....	47
4.4	Présentation du système.....	47
4.4.1	La page d'accueil .....	47
4.4.2	L'espace apprenant .....	48
4.4.3	L'espace tuteur .....	52
4.5	Conclusion .....	58

### **Conclusion générale .....60**

### **Bibliographie .....62**

### **Webographie.....64**

# Liste des tableaux

Tableau 1.1: <i>Tableau de comparaison de l'architecture orientée services et des microservices.</i> .....	13
Tableau 2.1 : <i>Analyse des travaux sur les micro-services dans les plateformes e-learning.</i> .....	22
Tableau 3.1 <i>tableau du MLD de MS test de niveau.</i> .....	39
Tableau 3.2 : <i>tableau du MLD de MS du cours.</i> .....	39
Tableau 3.3 : <i>tableau du MLD de MS du test final.</i> .....	40
Tableau 3.5 : <i>tableau du MLD de MS de collaboration.</i> .....	40
Tableau 3.6 : <i>tableau du MLD de MS de gestion des réunions.</i> .....	40

# Liste des figures

Figure 1.1 : <i>Cohésion interne d'un micro-service.</i> .....	9
Figure 1.2 : <i>Découplage entre les micro-services</i> .....	10
Figure 1.3 : <i>Distribution des micro-services</i> .....	10
Figure 3.1 : <i>Architecture générale du système e-learning basé sur les micro-services.</i> .....	27
Figure 3.2 : <i>Schéma des fonctionnalités de l'apprenant.</i> .....	28
Figure 3.3 : <i>Schéma de MS du test de niveau.</i> .....	29
Figure 3.4 : <i>Schéma de MS du cours.</i> .....	30
Figure 3.5 : <i>Schéma de MS du test final.</i> .....	30
Figure 3.6 : <i>Schéma de Ms de la collaboration.</i> .....	31
Figure 3.7 : <i>Schéma de MS de gestion des réunions.</i> .....	32
Figure 3.8 : <i>Schéma des fonctionnalités de l'enseignant.</i> .....	32
Figure 3.9 : <i>Schéma des fonctionnalités de l'administrateur.</i> .....	33
Figure 3.10 : <i>Schéma des fonctionnalités de tuteur.</i> .....	34
Figure 3.11 : <i>Schéma de MS de suivi d'accès.</i> .....	35
Figure 3.12 : <i>Schéma de MS d'avancement.</i> .....	36
Figure 3.13 : <i>Schéma de MS de communication.</i> .....	36
Figure 3.14 : <i>Le MCD de la plateforme d'apprentissage utilisé.</i> .....	37
Figure 3.15 : <i>table de micro-service test de niveau.</i> .....	38
Figure 3.16 : <i>les tables de micro-service du cours.</i> .....	38
Figure 3.17 : <i>les tables de micro-service du test final.</i> .....	38
Figure 3.18 : <i>les tables de micro-service collaboration.</i> .....	38
Figure 3.19 : <i>table de micro-service gestion des réunions.</i> .....	39
Figure 4.1 : <i>Interface de gestion de BDD XAMPP.</i> .....	46
Figure 4.2 : <i>Interface de site d'hébergement.</i> .....	47
Figure 4.3 : <i>Interface principale de la plateforme.</i> .....	48
Figure 4.4 : <i>Interface de l'espace apprenant.</i> .....	49
Figure 4.5 : <i>Interface de test du niveau.</i> .....	49
Figure 4.6 : <i>Consultation des cours et évaluation des exercices.</i> .....	50
Figure 4.7 : <i>Evaluation test final du module.</i> .....	51
Figure 4.8 : <i>Accueil de la collaboration.</i> .....	51
Figure 4.9 : <i>Programmation et accès à la réunion</i> .....	52
Figure 4.10 : <i>Accueil espace tuteur.</i> .....	53
Figure 4.11 : <i>Graphe suivi d'accès.</i> .....	54
Figure 4.12 : <i>Liste des cours, l'avancement d'étude et progression dans les cours.</i> .....	55
Figure 4.13 : <i>Espace de communication.</i> .....	55
Figure 4.14 : <i>Statistique sur les emails.</i> .....	56
Figure 4.15 : <i>Statistique sur les collaborations.</i> .....	56
Figure 4.16 : <i>Statistique sur réunion.</i> .....	56

Figure 4.17 : <i>Programmé et accès à la collaboration</i> .....	57
Figure 4.18 : <i>Programmation et accès à la réunion.</i> .....	58

# **Introduction générale**

# Introduction générale

L'intégration des technologies de l'information et de la communication dans le domaine de l'éducation a remplacé l'apprentissage traditionnel en présentiel en ce qu'on appelle aujourd'hui l'apprentissage en ligne ou la formation à distance. Depuis, ces systèmes n'ont cessé d'évoluer.

Dans le contexte de l'évolution rapide des technologies éducatives, les plateformes e-learning sont essentielles pour offrir des solutions d'enseignement et d'apprentissage flexibles et accessibles.

Cependant, malgré ces progrès, de nombreux systèmes restent encore basés sur des architectures monolithiques, ce qui limite leur scalabilité, leur flexibilité et leur capacité à s'adapter rapidement aux nouvelles exigences.

Pour surmonter ces défis, il est impératif de concevoir des plateformes basées sur une architecture modulaire, évolutive, maintenable et sécurisée. Parmi ces architectures nous trouvons l'architecture à base de microservices. Cette architecture permet de décomposer les fonctionnalités de la plateforme en petites unités indépendantes mais interconnectées, chaque micro-service peut être optimisé et évolué indépendamment. Cela permet de faciliter le développement, le déploiement et la maintenance des applications, offrant une meilleure scalabilité et une plus grande flexibilité.

L'objectif de notre travail est de faire la réingénierie d'une plateforme d'apprentissage classique vers une plateforme à base de microservice scalable et évolutive qui s'adapte aux besoins individuels des apprenants. Pour ce faire, une analyse détaillée des différentes fonctionnalités d'une plateforme e-learning sera réalisée, afin d'identifier les microservices nécessaires que nous pourrons utiliser et intégrer, puis ces microservices seront développés puis déployés et testés. L'avantage d'une telle approche c'est la possibilité de réutiliser et d'intégrer facilement les microservices dans d'autres plateformes qui en sont dépourvues.

Notre mémoire est organisé en trois principaux chapitres, organisés de la manière suivante :

## **Chapitre 1 : Architecture orienté service (SOA) et architecture micro-service**

Nous avons présenté les concepts liés à l'architecture orientée services (SOA). Ensuite, nous avons exploré les microservices, en fournissant des définitions et en mettant en évidence leurs avantages et inconvénients. Nous avons ensuite comparé ces architectures pour en montrer les différences, avant de conclure ce chapitre.

## **Chapitre 2 : Plateforme e-learning et les micro-services**

Nous commençons par une présentation du domaine du l'E-learning, en définissant ce qu'il est, en mettant en évidence ses avantages et ses inconvénients. Nous explorons également les différents acteurs des plateformes E-learning qui existent. Enfin, nous exposons un état de l'art sur les travaux liés à l'utilisation les micro-services dans le domaine du l'E-learning.

## **Chapitre 3 : Conception du système**

Ce chapitre s'intéresse à l'étude conceptuelle de notre système. Nous commençons par les objectifs principaux, puis détaillons l'architecture globale du notre système. Ensuite, nous expliquons les schémas fonctionnels des espaces de notre système puis nous détaillons les micro-services et les fonctionnalités développées.

## **Chapitre 4 : Implémentation du système**

Ce dernier chapitre concerne l'implémentation de notre système. Nous présentons les environnements, les outils et langages utilisés pour le développement, puis nous présentons les principales interfaces de notre plateforme.

A la fin, nous terminons le mémoire par une conclusion générale et perspective.

# **Chapitre 1**

## **Architecture orientée service (SOA) et architecture micro-service**

## 1.1 Introduction

À l'ère numérique actuelle, la capacité d'innover rapidement et de s'adapter aux évolutions du marché est devenue une nécessité vitale pour les entreprises. Les architectures orientées service (SOA) et les microservices ont émergé comme des réponses novatrices à cette exigence, offrant des solutions modulaires et flexibles pour la conception des systèmes informatiques. Dans ces architectures chaque service est une unité autonome, capable de fonctionner indépendamment tout en contribuant à un objectif commun. Cette approche permet de réduire les dépendances entre les différentes parties d'un système, rendant les processus de développement et de déploiement beaucoup plus agiles et réactifs.

Bien que dans notre projet nous nous intéressons à l'utilisation des microservices, nous avons préféré de revenir aussi sur les notions de base de la SOA afin de bien comprendre l'avantage de chacune des architectures et de maîtriser les aspects qui les différencient.

## 1.2 Concepts et notions de base

### 1.2.1 L'architecture orientée service

#### 1.2.1.1 Définition

Plusieurs définitions ont été proposées pour définir le terme «SOA», tels que :

*« Service Oriented Architecture (SOA) est un paradigme pour l'organisation et l'utilisation des capacités distribuées qui peuvent être sous le contrôle de différents domaines de propriété » [SW1].*

*« Les SOA (Service Oriented Architecture - ou Architecture Orientée Services) sont souvent assimilés à des technologies mais ce sont en réalité des principes d'architectures. En effet, la notion de SOA renvoie à une nouvelle manière d'intégrer et de manipuler les différentes briques et composants applicatifs d'un système informatique (comptabilité, gestion de la relation client, production, etc.) et de gérer les liens qu'ils entretiennent » [SW2].*

La SOA, ou L'architecture orientée services est un modèle de développement logiciel qui utilise des composants logiciels nommée « services ». Chaque service offre une

fonctionnalité spécifique, et ils peuvent échanger les données entre eux à travers des plateformes et des langages différents. SOA utilise les services pour construire des applications métier. Les développeurs utilisent la SOA. Il réutilise les services sur différents systèmes. Il se combine pour effectuer des tâches complexes [SW3].

### **1. 2. 1. 2      *Les avantages de L'architecture orientée services***

L'implémentation d'une méthode SOA dans une infrastructure informatique offre d'importants bénéfices, parmi lesquels [SW4] :

- Une modularité permettant de remplacer facilement un composant ou service par un autre.
- Une facilité de réutilisation des composants (par opposition à un système sur mesure conçu spécifiquement pour une organisation.
- Amélioration des possibilités d'évolution, en mettant simplement à jour un service ou en en ajoutant un nouveau.
- Une plus grande tolérance aux pannes.
- Une maintenance simplifiée.
- Facilité d'implémentation à partir d'une application objet déjà existante.
- Réduction des coûts en phase de la maintenance et d'évolution.
- Amélioration facile des performances pour les applications critiques (possibilité de répartition des traitements simplifiée).

### **1. 2. 1. 3      *Les inconvénients de L'architecture orientée services***

Bien que les avantages soient nombreux, il subsiste quelques inconvénients que nous pouvons mentionner [SW5] :

- Les normes de services Web dans certains domaines sont actuellement récentes.
- Les services Web souffrent de performances faibles.
- Nécessité d'appréhender de nouvelles technologies.
- Performances réduites pour des traitements simples (couche supplémentaire).

## **1. 2. 2      Le service**

Nous pouvons remarquer que la SOA est basé sur la notion de service d'où la nécessité de bien le définir et le cerner. Nous n'avons pas de définition standard pour le

service, mais il existe de nombreuses définitions, qui varient selon le domaine et le contexte.

### **1. 2. 2. 1 Définition**

Le service permet aux divers systèmes du système d'information de se communiquer et de partager des données : bref, Une partie intégrante du concept SOA concerne les protocoles de communication qui sont ensuite utilisés pour organiser le système d'information [SW6]. Le service serait mieux défini en expliquant ses caractéristiques, et c'est ce que nous allons voir.

### **1. 2. 2. 2 Les caractéristiques d'un service**

Cependant Thomas Erl [1] a donné dans son livre « SOA Principles of Service Design », en spécifiant huit caractéristiques principales pour les services.

Ces aspects sont [2] :

- **Contrat standardisé** : Les services utilisent un contrat formel pour interagir, lequel spécifie chaque service et établit les termes de l'échange d'informations.
- **Couplage lâche** : Les services doivent être conçus pour interagir sans devoir établir des dépendances étroites et croisées.
- **Abstraction** : signifie offrir les services du SI sous forme de **boîte noire**, la seule partie qui visible par le monde extérieur est exposée via le contrat de service.
- **Réutilisabilité** : es services doivent être conçus pour permettre une réutilisation potentielle, en encapsulant une logique de traitement suffisamment générique pour être utilisée comme ressource réutilisable.
- **Composabilité** : Les services peuvent être combinés avec d'autres services. Ceci permet de représenter la logique à divers niveaux de granularité et encourager la réutilisation et la création de couches d'abstraction.
- **Autonomie** : L'autonomie du service implique qu'il n'a pas besoin d'autres services pour fonctionner (il est indépendant des autres), il doit être en contrôle de son environnement et de ses ressources, et ne doit pas dépendre d'autres services pour gérer ses activités. Il doit être possible de le remplacer ou de le déplacer sans que cela affecte d'autres services.

- **Sans état (Stateless)** : Les services ne devraient pas être obligés de gérer les données d'état, car cela peut limiter leur capacité à rester faiblement couplés.
- **Découvrabilité** : Il est important que les services soient faciles à trouver et à comprendre pour les demandeurs de services afin qu'ils puissent les utiliser de manière efficace.

### **1. 2. 3 Les micro-services**

#### **1. 2. 3. 1 Définition 1**

La conception des microservices, introduite par Martin Fowler, vise à décomposer les fonctionnalités complexes d'un système en plusieurs microservices autonomes. Chaque microservice se concentre sur une fonction métier spécifique, favorisant ainsi une communication entre eux généralement via des protocoles légers tels que HTTP/API [3].

#### **1. 2. 3. 2 Définition 2**

Un microservice est un processus, une fonction ou un ensemble de fonctions qui accepte les demandes entrantes et effectue certaines actions. En communiquant avec d'autres parties du système à l'aide d'interfaces appelées API (Application Programming Interfaces) et de réseaux informatiques. Les microservices sont comme des équipes disposées en couches, chacune avec son propre ensemble de responsabilités et de fonctions [4].

#### **1. 2. 3. 3 Historique**

Le concept de "Micro-Web-Services" a été proposé par le Dr. Peter Rodgers lors d'un événement sur le cloud computing en 2005, tandis que le terme "microservices" a été utilisé lors d'une conférence d'architectes de logiciels au printemps 2011. Leur popularité augmente car ils sont utiles pour gérer les changements fréquents dans le domaine de l'informatique actuelle, tels que [SW7] :

- Appareils mobil
- Applications Web
- Conteneurisation des systèmes d'exploitation
- Utilisation des serveurs
- Serveurs multicœurs

- 10 Gigabit Ethernet

L'idée des microservices n'est pas récente. Google, Facebook, and Amazon utilisent cette approche à des degrés divers depuis plus de dix ans. Avant d'afficher les résultats de la page, plus de 70 microservices sont nécessaires pour réaliser une recherche sur Google. De plus, d'autres architectures ont été créées pour relever quelques-uns des défis similaires aux microservices. Parmi celles-ci, L'architecture orientée services (SOA) permet aux composants d'accéder aux services via un réseau, favorisant ainsi l'échange d'informations entre ces services. Malgré cela, elle a pour inconvénient de ne pas être capable de gérer les communications asynchrones.

#### **1. 2. 3. 4      *Les objectifs des architectures micro-services***

Les principaux objectifs des architectures micro-services sont [SW8] :

- **Implémente une seule tâche** (ou un ensemble de tâches étroitement associées) dans un contexte spécifique. Ceci est une caractéristique essentielle d'un microservice. Plus un microservice individuel essaie de faire des choses différentes, ou plus il franchit les frontières de propriété, moins il a de chances d'atteindre les autres objectifs des microservices.
- Est **faiblement couplé** et ne dépend que peu ou pas de la connaissance des définitions des autres microservices. De manière idéale, il communique avec d'autres services en utilisant des messages au format standard, comme JSON.
- Est **autonome** et peut être modifié sans nécessiter une coordination avec d'autres équipes de développement. C'est un élément essentiel pour favoriser l'agilité dans le développement de microservices.
- Est **déployable de manière indépendante** et être évalué, déployé et restauré séparément sans impact sur les autres microservices. C'est un élément essentiel pour favoriser la flexibilité dans le déploiement et l'évolutivité des microservices.

#### **1. 2. 3. 5      *Les caractéristiques des architectures micro-services***

Cependant, ses principales caractéristiques ont été bien définies [SW9] :

##### **1. Cohésion interne d'un micro-service :**

Un micro-service doit posséder une forte cohésion interne. Il faut réduire la portée fonctionnelle des fonctionnalités qu'il implémente. Principalement dédié à une fonctionnalité élémentaire. Le champ d'action d'un micro-service peut être associé au concept de *contexte délimité* (ou *bounded context*), provenant de l'approche du Domain Driven Design (DDD).

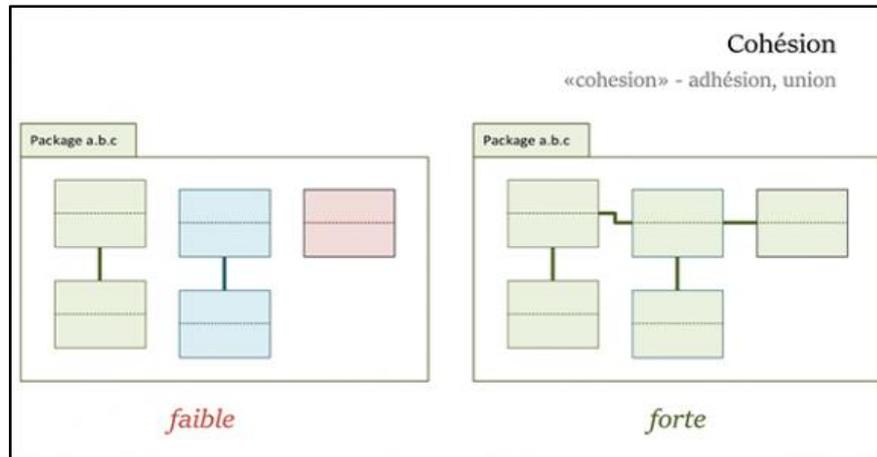


Figure 1.1 : Cohésion interne d'un micro-service[SW9].

## 2. Découplage entre les micro-services :

Dans un cadre de microservices, le logiciel est divisé en plusieurs services extrêmement **indépendants**. Chaque micro-service peut être :

- Déployé indépendamment.
- Conçu et développé indépendamment.
- Testé indépendamment.

En conséquence, chaque micro-service a la possibilité de se développer de façon autonome par rapport à une approche "monolithique". Cela donne à ce type d'architecture une capacité d'adaptation considérablement améliorée. Les architectures de micro-services (MSA) sont efficaces pour répondre de manière optimale aux objectifs de réactivité et d'adaptabilité des méthodes Agiles.

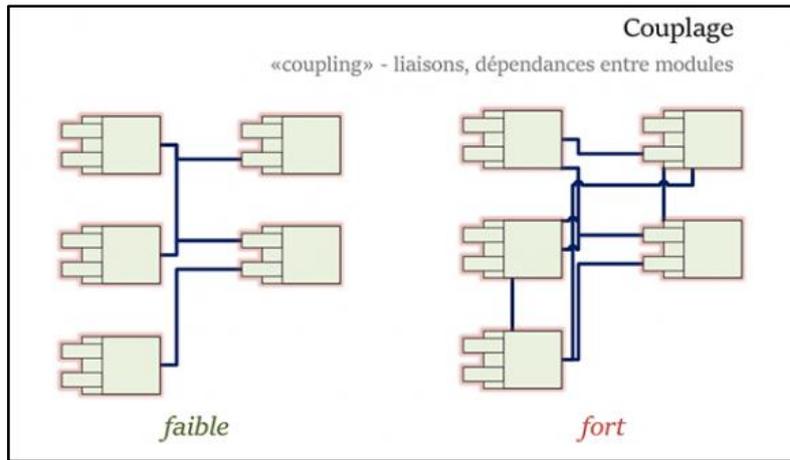


Figure 1.2 : Découplage entre les micro-services [SW9].

### 3. Distribution des micro-services :

Afin de réaliser cette séparation, chaque micro-service doit opérer de manière **autonome en tant que processus système**, que ce soit sur une machine dédiée ou à déployé d'un conteneur. Les échanges entre les clients et les microservices, ou entre les microservices eux-mêmes, sont réalisé via des services web ou un protocole de communication qui repose sur des échanges de messages, avec ou sans utilisation à un modèle d'acteurs.

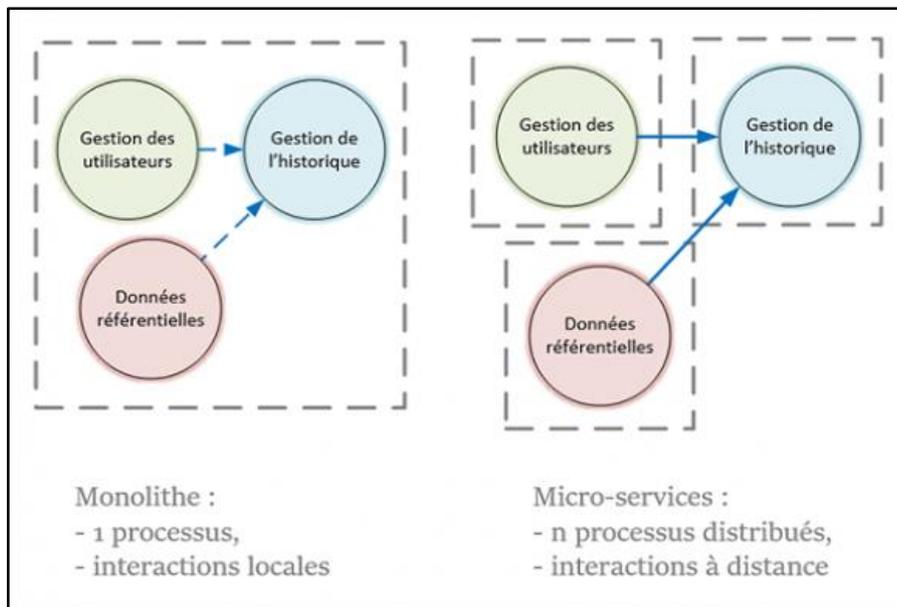


Figure 1.3 : Distribution des micro-services [SW9].

### **1. 2. 3. 6      *Les avantages des architectures micro-services***

L'architecture en micro-services présente de multiples bénéfices, notamment [4] :

- La facilité à mettre en place les microservices. Leurs programmes sont caractérisés par un code relativement court, ce qui les rend simples à maintenir et à gérer.
- Scalabilité et adaptabilité. Différents langages de programmation peuvent être utilisés pour la mise en œuvre des microservices, ce qui améliore leur exécution.
- La capacité à fonctionner de manière indépendante permet d'améliorer la fiabilité globale du MSA.
- Flexibilité dans le chargement du système.

### **1. 2. 3. 7      *Les inconvénients des architectures micro-services***

Malgré la multitude d'avantages, certains inconvénients persistent et peuvent être cités [SW10] :

- Utilisation élevée de la mémoire.
- Temps nécessaire pour fragmenter différents microservices.
- Complexité de la gestion d'un grand nombre de services.
- Les développeurs doivent résoudre des problèmes tels que la latence du réseau ou l'équilibrage de charge.
- Tests complexes sur le déploiement distribué.

### **1. 2. 3. 8      *Différence des microservices selon les plateformes***

Les microservices sont une méthode conceptuelle qui est interprétée de manière unique dans chaque langage. Cela constitue un avantage de cette structure, car les programmeurs peuvent utiliser le langage dans lequel ils excellent. Les langages les plus anciens peuvent se servir des microservices en se conformant à une architecture particulière adaptée à leur plateforme. Voici quelques attributs des microservices sur diverses plateformes.

Voici certaines caractéristiques des microservices sur différentes plateformes [SW7] :

- **Java**
  - Ne pas utiliser les fichiers Web Archive ou Enterprise Archive est recommandé.

- Les composants ne se déploient pas automatiquement. Des Amazon Machine Images ou des conteneurs Docker peuvent être utilisés en remplacement.

- **PHP**

Les microservices PHP de style REST sont en place depuis plusieurs années en raison de leurs avantages.

- Très adaptables au sein de l'organisation.
- Testés de manière rapide et simple.

- **Python**

- Il est facile de mettre en place un service Python qui fonctionne comme une interface Web pour les microservices écrits dans d'autres langages comme ASP ou PHP.
- Il y a une sélection variée de frameworks performants, tels que Flask ou Django, disponibles.
- Comprendre correctement l'API est crucial pour accélérer le processus de prototypage.
- On peut opter pour Pypy, Cython, C++ ou Golang en cas de besoin de davantage de rapidité ou d'efficacité.

- **Node.js**

Node.js est idéal pour les microservices car il correspond au langage utilisé dans les applications Web actuelles. Certains de ses points positifs sont :

- Utilise JavaScript et exploite le moteur open-source V8 puissant de Google.
- Machine code optimized dynamically during execution.
- Serveur HTTP avec des processus légers.
- Commutateurs non bloquants, contrôlés par des événements.
- Qualité élevée dans la gestion des paquets.
- Facile création de paquets pour les développeurs.
- Grande évolutivité grâce à une gestion asynchrone des entrées/sorties sur toute la chaîne.

- **.NET**

Dans les années 2000, .NET a été précurseur dans la création d'applications en tant que services à l'aide du protocole SOAP, dans un objectif similaire à celui des microservices actuels. De nos jours, la grande présence de .NET en milieu professionnel est l'un de ses

avantages.

Voici deux cas d'utilisation des microservices avec .NET :

- Création d'un service Web auto-hébergé en .NET en utilisant Open Web Interface for .NET (OWIN). On peut alors l'utiliser pour incorporer des microservices.

### 1. 2. 4 La différence entre l'architecture orientée services et micro-services

SOA et Microservices partagent plusieurs similitudes et différences décrites ci-dessous [5] [SW11] :

Architecture orientée services	Microservices
Services autonomes Faiblement couplé, cohérent et monolithiques	Petit, décomposé, isolé et Services découplés fonctionnant de façon indépendante
Communications entre services se produire via un service d'entreprise bus (ESB)	Communications entre services se produire grâce à des produits légers et standard protocoles de communication et interfaces (API)
Avec état (Stateful) et nécessite un mappage de dépendances du service lors des changements sont présenté	Apatride (Stateless) et moins fragile quand des changements sont introduits
Homogène	Hétérogène
Centralisé	Distribué
<b>Granularité</b> : Relativement importante, services modulaires	<b>Granularité</b> : Moindre, services plus flexibles soutenant un objectif ou une fonction spécifique de l'entreprise
<b>Interopérabilité</b> :Prise en charge de nombreux protocoles de messagerie, notamment SOAP, AMQP et MMQ	<b>Interopérabilité</b> : Utilisation de protocoles de messagerie légers et compatibles avec tous les langages, notamment HTTP, REST ou JMS
<b>Stockage</b> :Une seule couche de stockage des données partagée par l'ensemble des services d'une application donnée	<b>Stockage</b> : Serveur ou base de données indépendant(e) pour chaque service, en fonction des besoins

**Tableau 1.1:** Tableau de comparaison de l'architecture orientée services et des microservices.

## 1. 2. 5 Les approches “Top down “et “Button up “

Les termes "**de haut en bas**" et "**de bas en haut**" font référence respectivement à une approche "**descendante**" ou "**ascendante**" en technologie. On peut aussi choisir une approche hybride qui mélange les deux approches.

### 1. 2. 5. 1 “Top down “

La méthode **descendante** consiste à diviser fonctionnellement l'application en partant du niveau métier. L'architecte distingue les fonctionnalités en se basant soit sur le domaine (DDD - conception axée sur le domaine), soit sur la compétence métier [SW12]. Les décideurs non techniques trouvent plus facilement compréhensible cette approche.

### 1. 2. 5. 2 “Bottom up “

La méthode **ascendante** est strictement technique, se basant sur la disponibilité des ressources de développement pour la sélection des microservices. L'architecte a la capacité d'analyser les caractéristiques de service et de les combiner afin de former un microservice individuel. Il est également capable de subdiviser des applications plus volumineuses par technologie et de développer des microservices basés sur ces regroupements [SW12]. La méthode ascendante améliore l'efficacité de l'utilisation des ressources de développement disponibles.

## 1. 3 Conclusion

Dans ce chapitre, nous avons présenté les notions relatives à l'architecture orientée services (SOA), Puis passé au micro-service qui est une architecture logicielle où une application est construite comme une suite de petits services indépendants, chacun exécutant un processus unique et communiquant via des interfaces légères.

Notre choix s'est fixé sur les microservices qui offrent une meilleure modularité. SOA offre une vision plus globale de l'intégration des services à l'échelle de l'entreprise, Alors que les microservices se concentrent sur la décomposition des applications en unités autonomes et indépendantes.

# **Chapitre 2**

## **Plateforme e-learning et les micro-services**

## 2.1 Introduction

Dans le domaine de l'apprentissage en ligne, les plateformes éducatives jouent un rôle crucial pour rendre l'éducation accessible à tous. Ces plateformes nécessitent une architecture flexible et efficace pour satisfaire les divers besoins des apprenants et des éducateurs. En ce sens, les microservices se révèlent être une solution prometteuse, offrant agilité, scalabilité et personnalisation dans le développement et la gestion de ces plateformes.

Ce chapitre a pour objectif de rappeler les notions relatives aux plateformes e-learning, puis de faire un survol des différentes approches pour intégrer les microservices dans les plateformes e-learning.

## 2.2 E-learning

### 2.2.1 Définition

Il existe plusieurs définitions ont été proposées dans la littérature. La définition se diffère selon plusieurs critères tels que : formation en ligne, site web éducatif, etc.

#### 2.2.1.1 Définition 1

*« L'e-learning est un terme anglais qui veut dire « apprentissage par des moyens électroniques ». Il se réfère à l'utilisation du web et des nouvelles applications technologiques d'apprentissage distribué pour améliorer le processus d'acquisition d'un nouveau savoir ou la mise à jour de nouvelles connaissances. L'elearning est utilisé dans les programmes d'éducation nationale, les programmes de l'enseignement supérieur, les programmes de formation de l'entreprise, et les programmes de formation continue. » [6].*

#### 2.2.1.2 Définition 2

L'e-learning, également connu sous le nom d'apprentissage en ligne ou d'e-apprentissage.

Selon l'Union européenne (U.E) le 6 Janvier 2003 est : *« l'e-learning est l'utilisation des nouvelles technologies multimédias de l'Internet pour améliorer la qualité de*

*l'apprentissage en facilitant d'une part l'accès à des ressources et à des services, d'autre part les échanges et la collaboration à distance» [7].*

### **2.2.2 Les avantages du e-learning**

L'e-learning est une méthode d'apprentissage populaire et efficace grâce à ses nombreux avantages [8] :

- Participation active et interactive de l'apprenant et d'un large public.
- L'apprentissage en ligne offre un accès libre à des informations et nouvelles compétences qui sont plus importantes que jamais pour répondre aux exigences de la vie moderne. Tout le monde a la possibilité d'explorer les nouvelles technologies telles que, les ordinateurs, les systèmes multimédias et l'internet.
- La formation est ouverte à tout le monde, quel que soit leur âge, leur niveau d'éducation, leur statut socioprofessionnel, etc.
- L'autonomie de l'apprenant à tout moment.
- Il permet aux apprenants de communiquer, de s'exprimer et d'échanger.

### **2.2.3 Les inconvénients du e-learning**

Malgré ses avantages, l'apprentissage en ligne comporte des inconvénients. Voici quelques-uns de ces éléments nécessitant l'attention [8] :

- Problèmes liés à l'absence d'accès à l'Internet ou difficultés techniques (comme des perturbations du réseau de communications, des pannes d'ordinateurs, de terminaux ou de serveurs, des attaques de virus ou de pirates sur les documents électroniques de cours, etc.).
- L'absence physique de l'enseignant.
- L'e-learning nécessite une maîtrise suffisante des outils informatiques et Internet pour pouvoir suivre la formation.

### **2.2.4 Les principaux acteurs d'une plateforme d'e-learning**

Une plateforme d' e-learning est un espace d'éducation comprend divers acteurs, parmi lesquels on trouve [9] :

- ✚ **L'apprenant en 1er degré** : L'apprenant est le principal acteur, une personne active ayant la possibilité de consulter en ligne, de télécharger des ressources pédagogiques, d'échanger des informations et de réaliser des exercices et des évaluations qui seront ensuite corrigés par son enseignant, son tuteur, ou automatiquement par la plateforme.
- ✚ **L'enseignant** : est un professeur sa tâche consiste à partager des cours et des exercices pour évaluer la compréhension de l'apprenant et suivre sa progression dans l'apprentissage. Il est reconnu comme un expert dans ce domaine.
- ✚ **L'administrateur** : gère la plateforme, Il a la responsabilité de s'assurer que le système fonctionne correctement. Il assure l'installation, la maintenance, la gestion des droits d'accès, la validation des comptes des divers acteurs, la mise à jour des différents parcours pédagogiques, et d'autres tâches.
- ✚ **Le tuteur** : peut être un enseignant ou non, c'est un acteur qui aide les apprenants dans leur cursus pour atteindre le but de l'apprentissage. Il possède des fonctionnalités d'évaluation, de suivi (tracking) et d'accompagnement avec une communication avec son groupe d'apprenants.

### 2.3 Analyse des travaux sur utilisation des micro-services dans le e-learning

Le micro-service a été utilisé dans différents domaines. Ces dernières années, il y a eu un effort considérable pour développer des micro-services qui répondent à différents besoins éducatifs.

Dans la littérature nous pouvons trouver plusieurs travaux de recherche qui ont proposé d'utiliser les micro-services.

Le travail de **Yevhen et al** [10] examine l'utilisation de l'architecture microservices dans les systèmes d'apprentissage en ligne (e-learning) pour offrir un accès multi-variant aux matériaux éducatifs. En transition d'une architecture monolithique à une architecture microservices, l'étude surveille les indicateurs de performance du matériel serveur et le temps de réponse moyen aux requêtes des utilisateurs. Les résultats montrent que l'architecture microservices réduit significativement les exigences matérielles et les temps de réponse sous des charges élevées, tout en simplifiant le développement et la modification des systèmes en ligne. Cette méthode permet également de gérer les

ressources du système et d'appliquer de nouveaux paramètres sans redémarrage, assurant ainsi la continuité de l'opération du système.

L'étude de **Kurniawan et Dzikri** [11] présente le développement d'un prototype de plateforme d'apprentissage en ligne basé sur une architecture microservices et une passerelle API. Cette approche vise à améliorer l'accès à l'éducation en ligne, en particulier pour les régions rurales et urbaines. En utilisant une méthodologie de prototypage, les auteurs ont démontré que cette intégration permet d'améliorer la flexibilité, la scalabilité et l'efficacité de la plateforme, facilitant ainsi un accès plus équitable à l'éducation.

Dans cette étude réalisée par **Ana Milovanović** [12] cette dernière explore l'application de l'architecture microservices dans les systèmes d'apprentissage en ligne, en se concentrant sur l'intégration avec des systèmes de gestion de l'apprentissage (LMS) existants comme Moodle. Utilisant le Domain Driven Design pour identifier les microservices, l'article propose une architecture personnalisée intégrant des plateformes de données en nuage. L'accent est mis sur la flexibilité et la réutilisabilité des microservices, facilitant l'ajout de nouvelles fonctionnalités indépendantes. L'étude conclut que l'architecture microservices offre des avantages significatifs en termes de déploiement, évolutivité et gestion de données par rapport aux architectures monolithiques et orientées services traditionnelles.

Le travail présente un système de gestion de l'apprentissage (LMS) basé sur une architecture de microservices utilisant Moodle. Les auteurs **Rishabh et Surendra** [13] ont conçu une plateforme pour améliorer le système d'information départemental (DIS) en intégrant Moodle pour une gestion dynamique des données académiques. Le système vise à gérer efficacement les notes, les devoirs, les quiz et les présences. La conception modulaire utilise Angular pour l'interface utilisateur (le front-end), MySQL pour la base de données et Spring Boot pour le back-end, avec une passerelle API pour faciliter l'accès et l'authentification des utilisateurs. Le projet améliore la flexibilité, l'efficacité et la maintenabilité du DIS, avec des plans futurs pour étendre la génération de rapports à d'autres modules.

Nous avons proposé de résumer les articles en fonction de critères que nous avons sélectionnés. Les données sont présentées dans le tableau ci-dessous.

Article	Outils et technologies	Micro-services	Fonctions
[10]	<p><b>Architecture</b> : Monolithique et Microservices</p> <p><b>Language</b> : PHP</p> <p><b>BDD</b> : MYSQL</p> <p><b>Matériel</b> : Intel Xeon Servers (E3-1230 v5/6, 2x E5-2690 v2)</p> <p><b>Approche de Développement</b> : Agile et DevOps</p> <p><b>Surveillance et analyse</b> : Suivi des indicateurs clés comme l'utilisation du CPU, de la RAM, les temps de réponse des serveurs et des bases de données.</p>	<ul style="list-style-type: none"> <li>- Serveur de Cours de Formation</li> <li>- Serveur de Laboratoires Virtuels</li> <li>- Serveur de Comptes Personnels</li> <li>- Serveur de Système d'Évaluation des Connaissances</li> <li>- Serveur d'Évaluation des Performances Utilisateurs</li> <li>- Serveur d'Analytique</li> <li>- Serveur de Centre de Traitement des Données</li> <li>- Serveurs d'Importation de Données</li> </ul>	<ul style="list-style-type: none"> <li>- Gère les contenus des cours de formation.</li> <li>- Gère les laboratoires virtuels et les expériences interactives.</li> <li>- Gère les informations et les configurations des comptes utilisateurs.</li> <li>- Gère les tests et évaluations pour mesurer les connaissances des utilisateurs.</li> <li>- Gère l'évaluation et le suivi des performances des utilisateurs.</li> <li>- Centralise et traite les données provenant des différents microservices.</li> <li>- Traite et analyse les données pour fournir des rapports et des insights.</li> <li>- Gèrent l'importation de données et l'accès aux services externes.</li> </ul>
[11]	<p>Approche de prototypage</p> <p>API Gateway</p> <p>Architecture microservices</p> <p>Backend pour Frontend (BFF)</p> <p>Protocoles HTTP/HTTPS</p> <p>Protocoles de messages (MQTT, Apache Kafka)</p>	<ul style="list-style-type: none"> <li>- Gestion des cours</li> <li>- Gestion des utilisateurs</li> <li>- Gestion du contenu pédagogique.</li> <li>- Outils de communication</li> <li>- Suivi des progrès des étudiants</li> </ul>	<ul style="list-style-type: none"> <li>- Création, édition et configuration des cours.</li> <li>- Inscription, authentification, autorisation et gestion des profils utilisateur.</li> <li>- Téléchargement, stockage et gestion des matériaux pédagogiques (vidéos, documents, images).</li> <li>- Gestion des outils de communication entre les étudiants et les instructeurs (chat en direct, forums de discussion, notifications par email).</li> <li>- Gestion des données relatives aux devoirs terminés,</li> </ul>

	<p>RESTful API</p> <p>Network Profiler</p> <p>Technologie Docker</p>	<p>- Gestion des notifications</p>	<p>résultats d'examens et autres statistiques.</p> <p>- Livraison de notifications et alertes aux utilisateurs concernant les mises à jour des cours ou les devoirs imminents.</p>
[12]	<p><b>Protocole de Communication :</b> HTTP Protocol</p> <p><b>Gestion des API :</b> API Gateway</p> <p><b>Messagerie Asynchrone :</b> Message Broker (Kafka)</p> <p><b>Cloud Data Platform :</b> Comprend plusieurs couches : Ingest , Storage, Processing , Serving</p> <p><b>BDD :</b> NoSQL Database</p> <p><b>Gestion des Événements :</b> GitHub Webhook</p> <p><b>Plateforme LMS :</b> Moodle LMS</p> <p><b>Interopérabilité :</b> Protocole LTI (Learning Tools Interoperability)</p> <p><b>Technologies de Stockage Rapide :</b> Apache Kafka et Amazon Kinesis</p>	<p>- Service d'authentification</p> <p>- Service de recherche de ressources pédagogiques</p> <p>- Service de jeux éducatifs</p> <p>- Service de communication</p> <p>- Service d'analyse et de reporting</p>	<p>- Gère l'authentification des utilisateurs.</p> <p>- Permet la recherche de matériels pédagogiques selon certains critères.</p> <p>- Implémente des quiz éducatifs.</p> <p>- Fournit des fonctionnalités de chat pour la communication entre utilisateurs.</p> <p>- Génère des rapports sur les activités des étudiants.</p>
[13]	<p><b>Angular Framework :</b> Développement du front-end pour une interface utilisateur réactive et interactive.</p> <p><b>MySQL :</b> pour la base de données</p> <p><b>Spring Boot Framework :</b> Développement du back-end</p> <p><b>API Gateway :</b> Point d'entrée et de routage</p>	<p>- Microservice de gestion des notes</p> <p>- Microservice de gestion des devoirs</p> <p>- Microservice de gestion des quizz</p> <p>- Microservice de gestion de l'assiduité</p>	<p>- Récupération et gestion des notes des étudiants pour les cours et les évaluations.</p> <p>- Récupération et gestion des devoirs assignés aux étudiants.</p> <p>- Récupération et gestion des quizz administrés aux étudiants.</p> <p>- Gestion des présences des étudiants pour chaque cours.</p>

	<p><b>ZUUL Routing</b> : Utilisé dans l'API Gateway</p> <p><b>Service Discovery</b> : Enregistre indépendamment tous les microservices et permet leur interaction.</p> <p><b>Moodle</b> : Plateforme de gestion de l'apprentissage intégrée aux microservices pour gérer les données académiques dynamiques.</p>		
--	--	--	--

**Tableau 2.1** : Analyse des travaux sur les micro-services dans les plateformes e-learning.

## 2.4 Conclusion

En adoptant une architecture basée sur les microservices, les plateformes d'e-learning peuvent considérablement améliorer leur flexibilité, leur scalabilité et leur personnalisation. Cette approche permet de créer des systèmes plus robustes et plus faciles à maintenir, répondant ainsi aux besoins diversifiés des utilisateurs. Les microservices offrent une solution efficace pour surmonter les défis technologiques actuels, ouvrant la voie à une éducation en ligne plus dynamique et accessible à tous.

A travers ce chapitre nous avons fait le tour des plateformes d'apprentissage. De plus, nous avons analysé plusieurs études et constaté qu'il existe peu de recherches sur l'intégration des microservices dans l'e-learning, d'où l'importance de notre étude.

Dans le chapitre suivant, nous allons détailler les choix de conception pour la réalisation d'une plateforme d'apprentissage base de microservices..

# **Chapitre 3**

## **Conception du système**

### 3.1 Introduction

Dans le contexte de l'évolution rapide des technologies éducatives, les plateformes e-learning sont essentielles pour offrir des solutions d'enseignement et d'apprentissage flexibles et accessibles. Concevoir ces plateformes nécessite une architecture modulaire, évolutive, maintenable et sécurisée.

Dans le cadre de notre projet, nous avons choisi de concevoir et de réaliser une plateforme e-learning reposant sur une architecture des microservices. Deux approches peuvent être utilisées pour développer cette architecture : Bottom-Up et Top-Down.

L'approche Bottom-Up consiste à intégrer et combiner des services existants pour construire un nouveau système. Elle inclut des étapes telles que la recherche, l'intégration et la composition de services.

Cependant, nous avons choisi de suivre l'approche Top-Down. Cette méthode commence par la description du système cible, en affinant progressivement les détails jusqu'au niveau physique pour l'implémentation. Elle nécessite des tâches d'analyse, de modélisation et d'implémentation. En conséquence, nous adopterons une démarche de conception alignée avec cette approche et utiliserons un langage de modélisation approprié pour garantir une mise en œuvre efficace.

Dans ce chapitre, nous présenterons les objectifs de notre système, l'architecture globale et détaillée que nous avons proposée ainsi que les fonctionnalités offertes pour les différents acteurs de notre système.

### 3.2 Objectifs

Notre objectif est de développer une plateforme e-learning très modulaire qui utilise les micro-services. La plateforme sera construite en suivant un processus Top-Down.

Le développement des technologies de microservices améliore la performance, la scalabilité et la flexibilité des plateformes d'apprentissage en ligne.

Nous commencerons par identifier les différentes fonctionnalités de la plateforme, puis nous décomposerons l'application en petits modules autonomes. Chaque module aura une responsabilité bien définie et fonctionnera de manière indépendante.

La granularité de chaque module sera aussi fine que possible, garantissant idéalement une seule fonctionnalité ou un minimum de fonctionnalités cohérentes, réutilisables.

Enfin, nous développerons ces modules sous forme de micro-services et nous testerons leur bon fonctionnement.

### 3.3 Processus d'identification des micro-services

Afin d'identifier les différents micro-services qui composeront la plateforme e-learning, nous avons suivi les étapes suivantes :

- 1- Explorer toutes les fonctionnalités d'une plateforme elearning
- 2- Dresser une liste exhaustive des fonctionnalités pouvant être exécutés indépendamment du reste de la plateforme
- 3- Si possible proposer chaque fonctionnalité comme microservice
- 4- Sinon rassembler le minimum de fonctionnalité cohérente formant un module indépendant.
- 5- Vérifier la cohérence des fonctionnalités regroupées et le couplage faible avec le reste de l'application
- 6- Vérifier la prédisposition à la réutilisation et à l'accès distant.
- 7- Valider la liste des microservices.

Suite au processus d'analyse des différentes fonctionnalités assurées par une plateforme e-learning, nous avons pu identifier la liste des microservices suivante :

1. **Micro-service du test de niveau** : Il contient des questions d'intelligence, d'études, de langues, etc. Après récupérer le résultat de l'évaluation, qui détermine le niveau de l'apprenant.
2. **Micro-service des cours** : Partager des contenus pédagogiques Ils facilitent également la récupération et l'évaluation des exercices liés à un chapitre spécifique.
3. **Micro-service du test final d'un module** : consiste en un examen comprenant des questions à choix multiples pour chaque module. Ensuite, il récupère le score de l'examen et détermine si l'étudiant est acquis ou non acquis dans le module.

4. **Micro-service de collaboration** : Fournir une plateforme collaborative pour les apprenants du même groupe et le tuteur peut de capturer des statistiques de collaboration.
5. **Micro-service de suivi d'accès** : fournit des statistiques sur l'accès des apprenants par semaine, 10 jours, et semestre.
6. **Micro-service d'avancement** : offre des statistiques sur l'avancement des apprenants dans chaque module avec le nombre d'essais pour chaque test de chapitre, et la progression des apprenants dans les cours.
7. **Micro-service de communication**: les statistiques de l'interaction entre l'apprenant et ses pairs (forums, mail, chat).
8. **Micro-service gestion des réunions** : Offrir un espace où les apprenants peuvent communiquer avec le tuteur, tout en permettant à ce dernier de collecter des statistiques sur leur collaboration.

### 3.4 Architecture générale d'application

L'architecture générale de la plateforme d'apprentissage est schématisée comme suit :

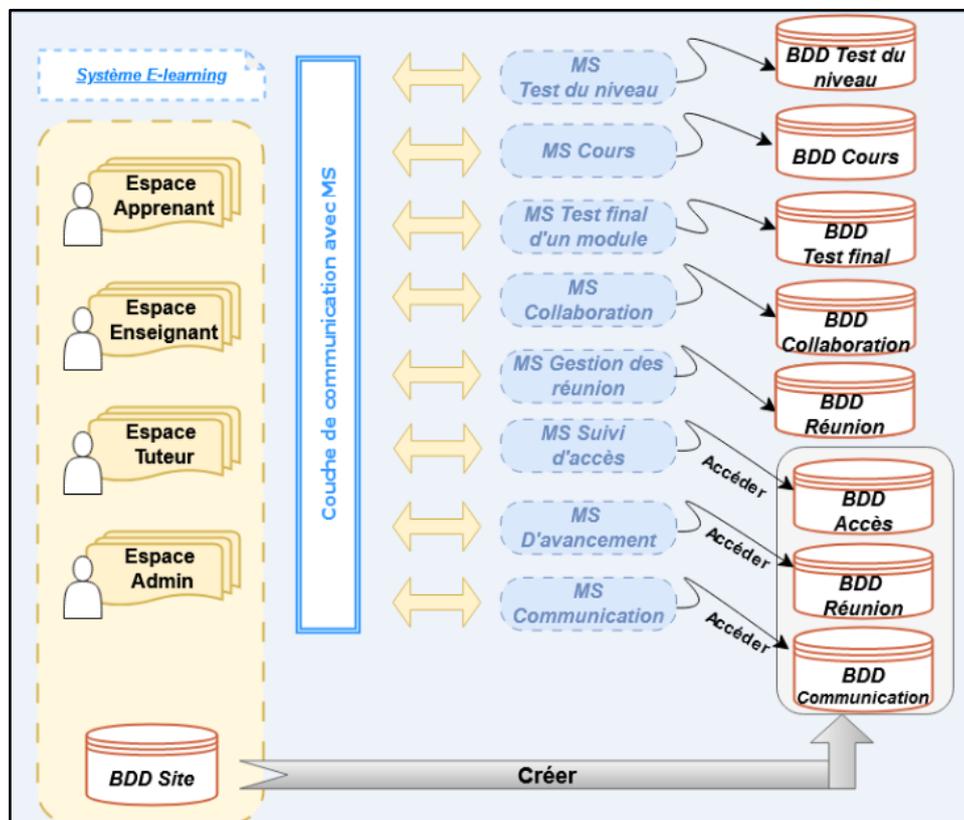


Figure 3.1 : Architecture générale du système e-learning basé sur les micro-services.

## 3.5 Les schémas fonctionnels

### 3.5.1 Espace apprenant

Quand un apprenant accède au site web pour la première fois :

- D'abord, il doit s'inscrire dans notre plateforme.
- Ensuite, il doit remplir le formulaire avec toutes ses coordonnées.
- Puis, il passe un test du niveau cognitif sous forme d'un QCM développé en tant que micro-service. Selon les résultats de ce test, on établit son niveau et on lui attribue des cours correspondants.
- Après la validation de l'inscription par l'administrateur, l'apprenant commence ses études en parcourant les cours locaux ou distants sous forme de microservice. Les cours affichés sont adaptés à son niveau.
- A la fin de chaque chapitre il passe un test d'évaluation pour pouvoir passer au chapitre suivant de chaque module.
- Enfin, il peut passer un test final pour chaque module au lieu de devoir passer plusieurs tests de chapitre (où existent dans le micro-service) comprenant des questions à choix multiples.
- Il peut aussi programmer une réunion de discussion avec le tuteur pour discuter des problèmes.

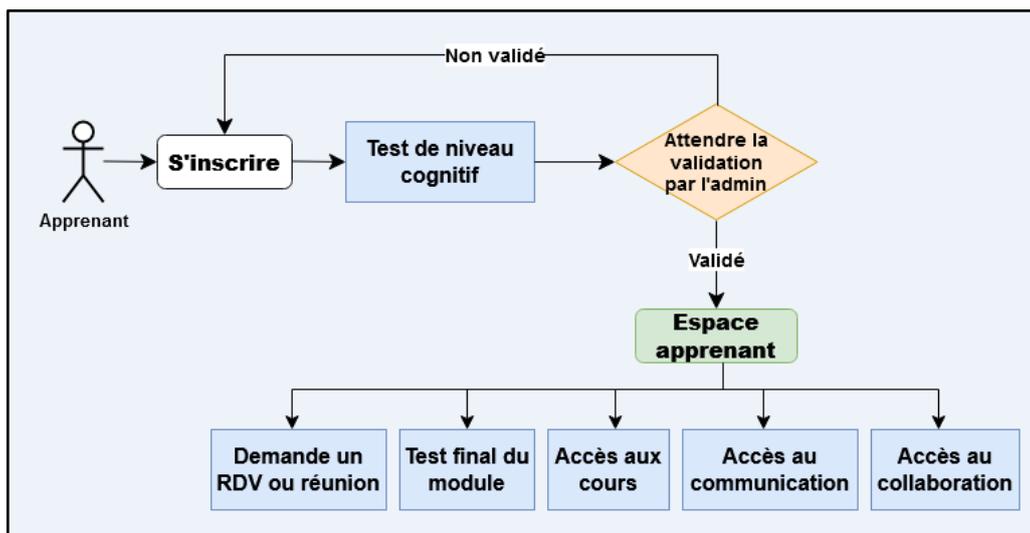


Figure 3.2 : Schéma des fonctionnalités de l'apprenant.

### 3. 5. 1. 1 Identifier les micro-services de l'apprenant

- Micro-service de test du niveau** : ce micro- service a été créé par des pédagogues et des chercheurs dans le domaine des sciences humaines. Le micro-service contient des questions sur l'intelligence, les études, les langues, etc. Après que l'apprenant a passé le test, le résultat de l'évaluation est envoyé, correspondant au niveau attribué à l'apprenant.

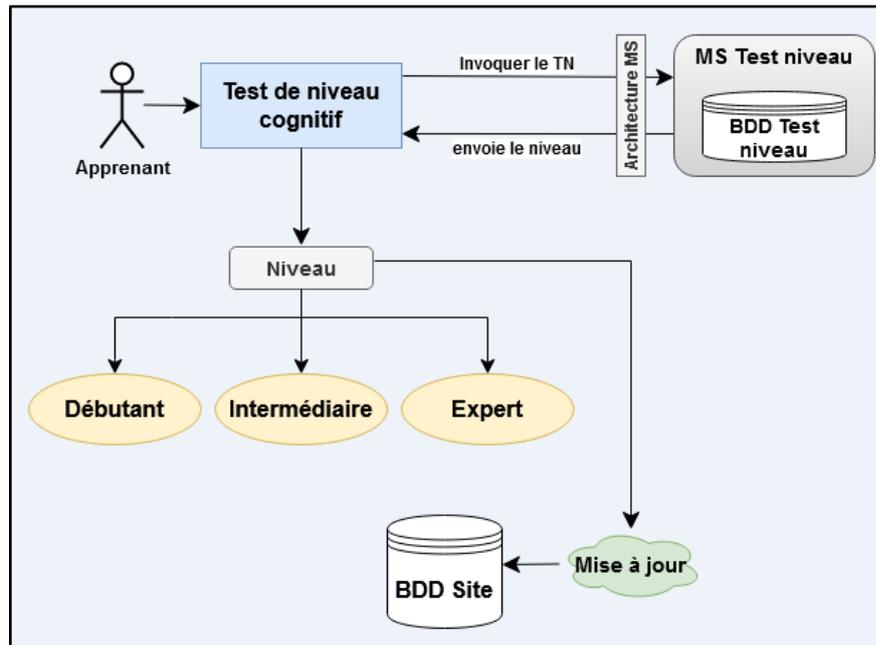


Figure 3.3 : Schéma de MS du test de niveau.

- Micro-service de cours** : Ce micro-service permet de mettre à disposition des cours distants qui ne sont pas disponibles localement, il permet ainsi de partager des contenus pédagogiques accessibles via diverses plateformes de formation à distance. Il offre la possibilité d'accéder aux exercices associés à un chapitre donné pour l'évaluation, ainsi que les résultats de cette évaluation, afin de les sauvegarder localement dans la base de données du système. Pour accéder à un chapitre, l'identifiant de l'apprenant et son niveau cognitif sont nécessaires. De même, pour obtenir les exercices, le code du chapitre concerné est requis.

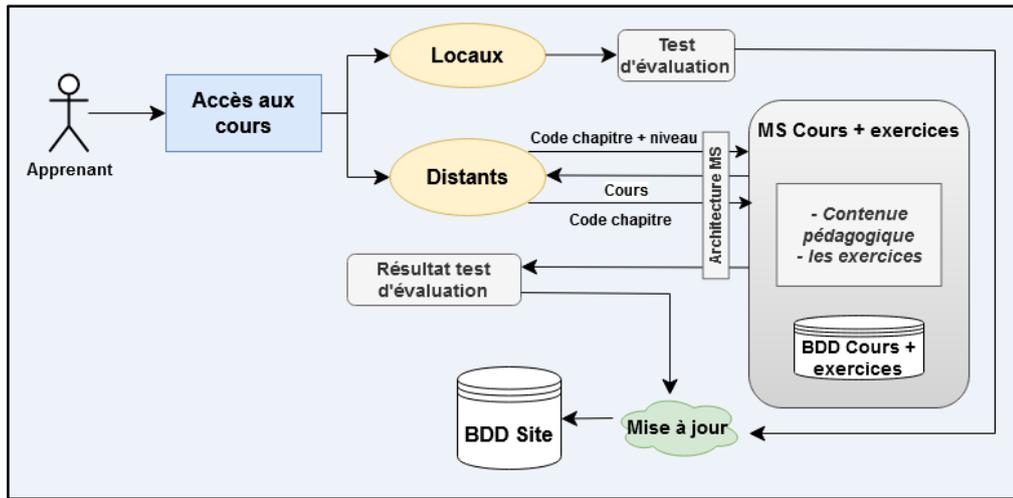


Figure 3.4 : Schéma de MS du cours.

- ✚ **Micro-service de test final d'un module** : Si l'apprenant a suivi le cours du module ou possède déjà une connaissance préalable du module, il est autorisé à passer le test final. Les questions du test final correspondant au module sont récupérées. Après que l'apprenant ait passé le test, le système communique avec le micro-service pour récupérer le score, qui détermine l'état de l'apprenant :
  - Si le score  $< 50\%$ , alors l'apprenant n'a pas acquis le module.
  - Si le score  $\geq 50\%$ , alors l'apprenant a acquis le module.

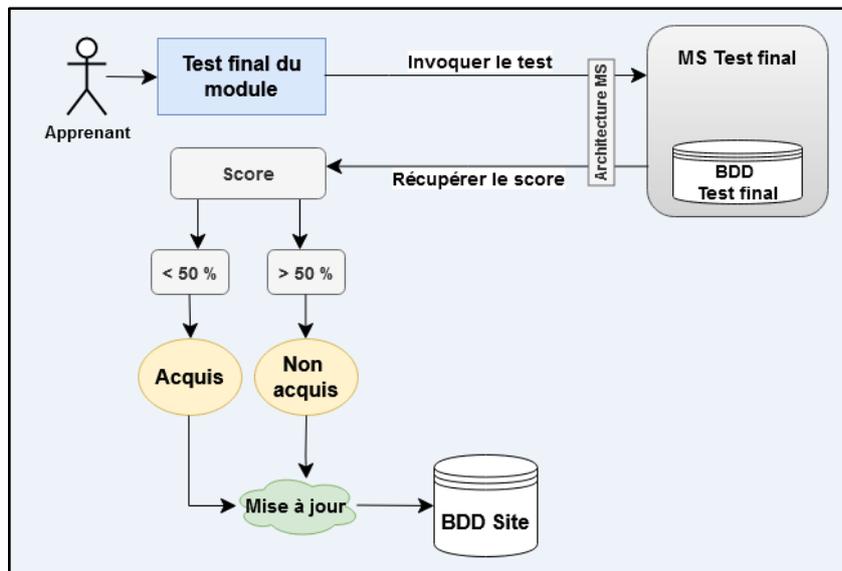


Figure 3.5 : Schéma de MS du test final.

- ✚ **Micro-service de la collaboration** : ce micro-service est une plateforme collaborative qui permet aux apprenants d'un groupe de travailler ensemble pour

discuter et implémenter la solution en ligne pour un projet donné. Ce MS permet aussi de recueillir des statistiques et des traces sur la collaboration de chaque apprenant et qui seront ultérieurement accessible par le tuteur pour de prendre en compte ces statistiques et se faire une idée sur l'avancement des études et des connaissances de cet apprenant .

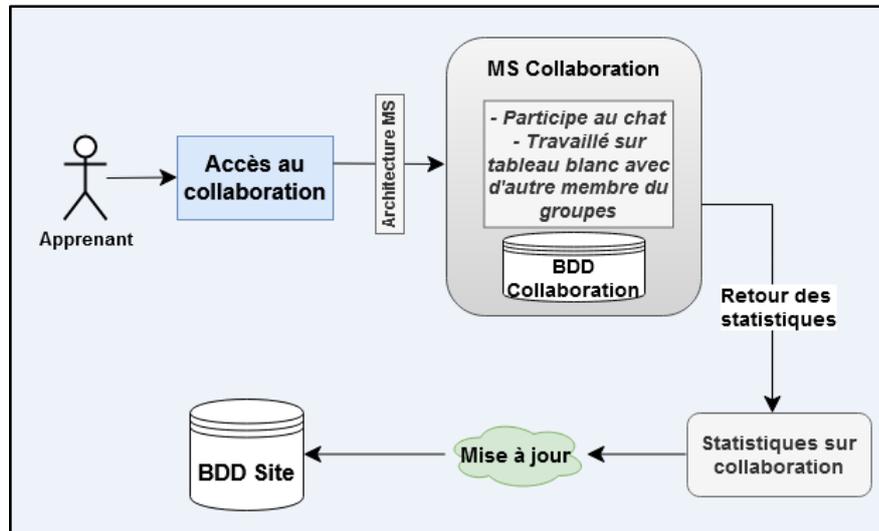


Figure 3.6 : Schéma de Ms de la collaboration.

- + **Micro-service de gestion des réunions :** Ce micro-service est conçu pour faciliter la planification et la gestion des réunions entre tuteurs et apprenants dans un cadre éducatif. Il permet aux apprenants et aux tuteurs de demander et d'organiser des réunions. L'apprenant peut envoyer une demande pour créer une réunion en précisant la date, l'heure, l'objet et le type. Une fois la demande soumise, le système notifie automatiquement le tuteur concerné, qui reçoit tous les détails de la demande. Le tuteur peut alors confirmer la réunion. Après confirmation, le système enregistre la réunion dans la base de données. L'apprenant accède à la réunion. Une fois la réunion terminée, l'apprenant peut accéder à l'évaluation de la réunion et la noter comme étant moyen, bénéfique, ou non bénéfique. Ce MS permet aussi de capturer des statistiques sur la réunion de chaque apprenant et tuteur et qui seront accessible par le tuteur ultérieurement afin de prendre en compte ces statistiques.

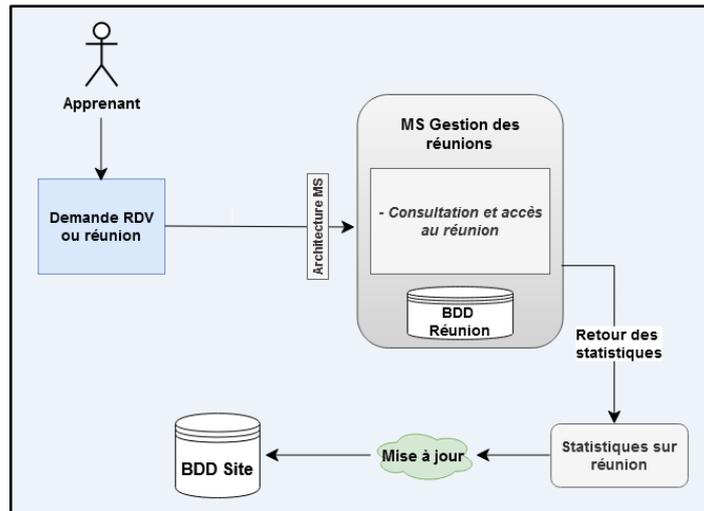


Figure 3.7 : Schéma de MS de gestion des réunions.

### 3.5.2 Espace enseignant

Lors de l'accès au site :

- D'abord, l'enseignant commence par une inscription validée par l'administrateur.
- Ensuite, il commence par ajouter des cours, chapitre par chapitre sous forme de PDF (en proposant plusieurs versions pour chaque niveau.).
- Il peut ajouter des modules pour chaque spécialité.
- Il a la possibilité d'ajouter, modifier ou supprimer des exercices avec des choix de réponses multiples.
- L'enseignant peut également modifier les informations de son profil.
- Il a la capacité de communiquer avec les autres participants de la plateforme.

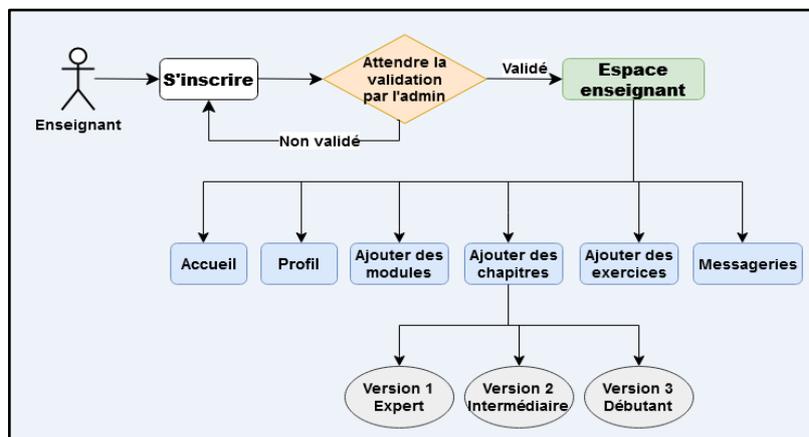


Figure 3.8 : Schéma des fonctionnalités de l'enseignant.

### 3.5.3 Espace administrateur

L'administrateur a la possibilité d'exécuter les actions suivantes :

- Validation des demandes d'inscriptions des différents acteurs de la plateforme.
- Ajouter, modifier, supprimer module ou chapitre.
- Ajouter, modifier, supprimer parcours.
- Ajouter, modifier, supprimer enseignant.
- Ajouter, modifier, supprimer apprenant.
- Ajouter, modifier, supprimer tuteur.
- Affectation tuteur / apprenant.

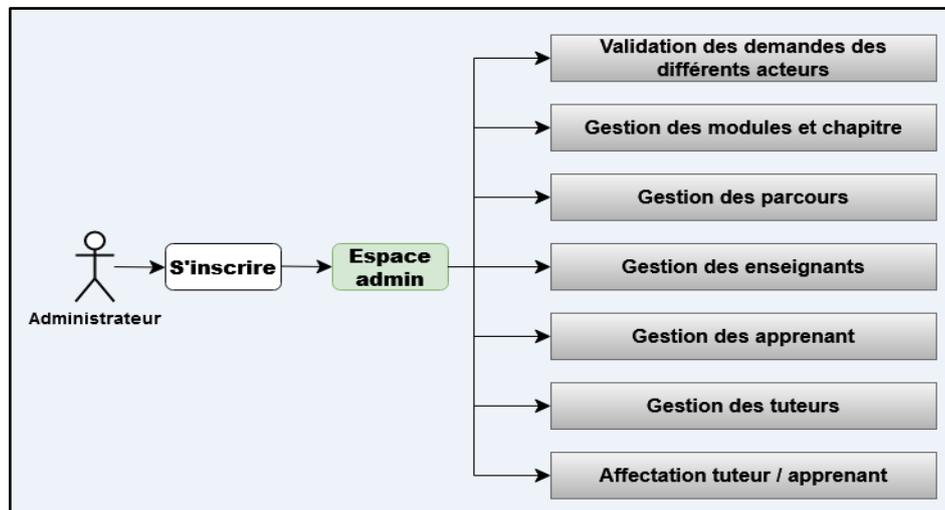


Figure 3.9 : Schéma des fonctionnalités de l'administrateur.

### 3.5.4 Espace tuteur

Le tuteur, à travers l'espace qui lui a été dédié peut accomplir les activités suivantes :

- Commence par une inscription qui sera validée par l'administrateur
- Il est affecté à un groupe automatiquement par l'administrateur.
- Après l'accès à son espace il peut parcourir son groupe d'apprenant dans la page d'accueil où il peut accéder aux détails de chaque apprenant.
- Consulter les statistiques de l'apprenant :
  - Le nombre d'accès (par 10 jours, mois et trimestre).
  - L'avancement dans le cours.
  - les communications effectuées avec les différents acteurs.

- la collaboration avec les pairs.
- Modifier le niveau actuel de l'apprenant après l'étude de ses statistiques.
- Programmer une réunion et RDV de discussion avec les apprenants et évaluer le RDV.

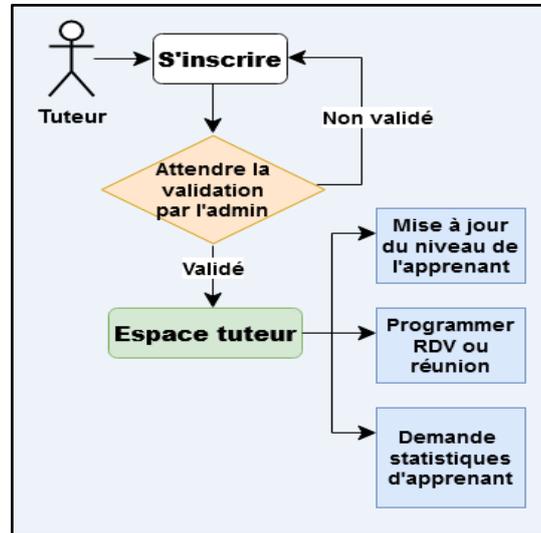


Figure 3.10 : Schéma des fonctionnalités de tuteur.

#### 3. 5. 4. 1 Identifier les micro-services de Tuteur

- ✚ **Micro-service de gestion des réunions** : ce micro-service est conçu pour faciliter la planification et la gestion des réunions entre tuteurs et apprenants. C'est la même MS accessible par l'apprenant.
- ✚ **Micro-service de suivi d'accès** : permet d'intégrer à distance une page web dans l'espace du tuteur, à travers une iframe. Son objectif principal est de fournir de manière claire et accessible les statistiques d'accès des apprenants à la plateforme. Il donne un aperçu détaillé du nombre d'accès par apprenant, segmenté par trimestre, mois et sur une période de 10 jours.

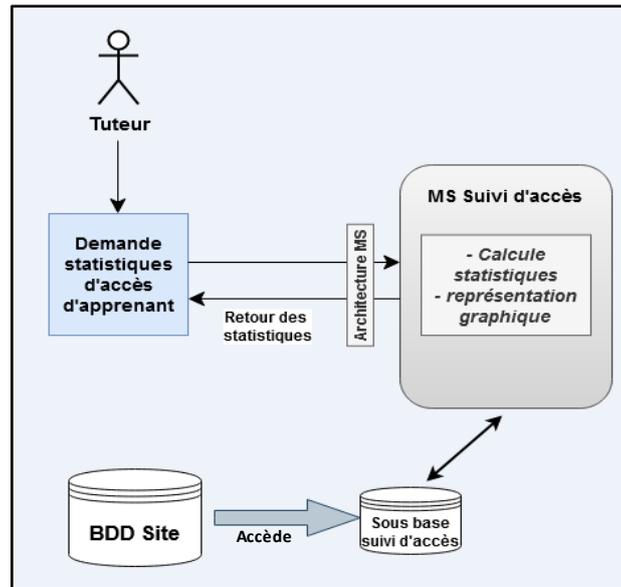


Figure 3.11 : Schéma de MS de suivi d'accès.

✚ **Micro-service d'avancement** : Ce microservice est conçu pour aider les tuteurs à suivre la progression des apprenants qu'ils supervisent dans les cours, en leur fournissant des outils pour examiner les statistiques d'avancement, et suivre la progression individuelle des étudiants.

*1- Avancement dans les Cours* : Expose en détail l'avancement dans chaque module, incluant le nombre d'essais requis pour réussir les tests dans chaque chapitre.

*2- La progression dans les cours* : La formule est déterminée en fonction du nombre de chapitres étudiés et du nombre de chapitres à étudier (avancement). Les résultats sont représentés graphiquement à l'aide d'un histogramme, qu'il soit par semaine, mois ou trimestre. Ces résultats serviront de critère essentiel pour évaluer le risque d'abandon.

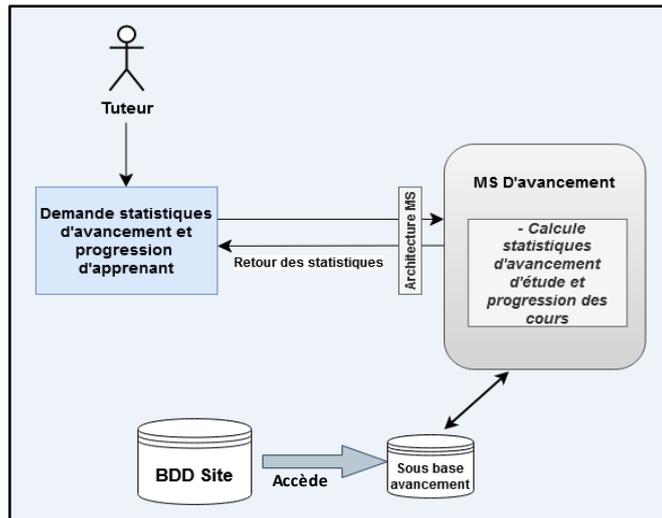


Figure 3.12 : Schéma de MS d'avancement.

- + **Micro-service de communication** : Ce microservice vise à fournir des statistiques détaillées sur les interactions des apprenants avec différents acteurs (email, forum). Cela permet aux tuteurs de suivre et d'évaluer l'engagement et la participation des apprenants dans un environnement éducatif. Permet de consulter les Statistiques des apprenants avec différents acteurs (les nombres d'emails envoyés et reçus, nombre de forums répondus).

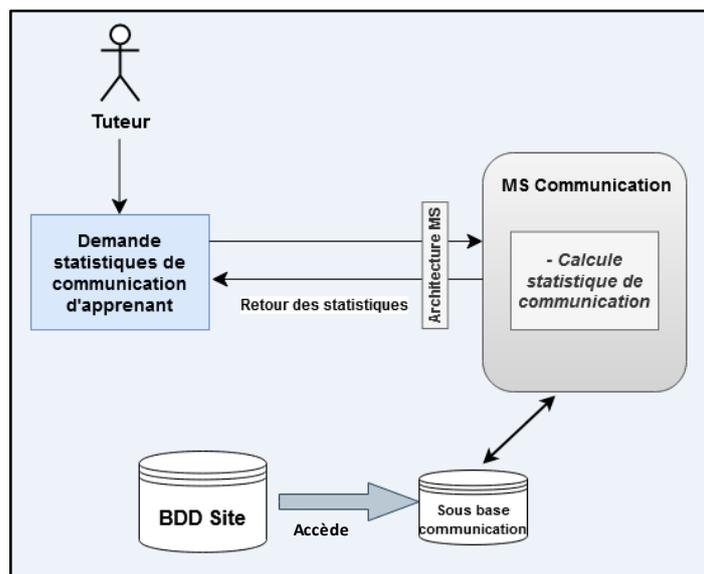


Figure 3.13 : Schéma de MS de communication.

### 3.6 Duplication de la BDD

L'autonomie est l'une des caractéristiques les plus importantes des micro-services. Les micro-services ne sont pas autorisés à accéder à la base de données de la plateforme



Nous avons étendu le MCD par les tables suivantes de chaque micro-service :

✚ **Micro-service test de niveau :**

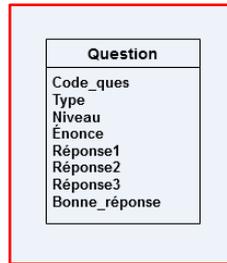


Figure 3.15 : table de micro-service test de niveau.

✚ **Micro-service du cours :**

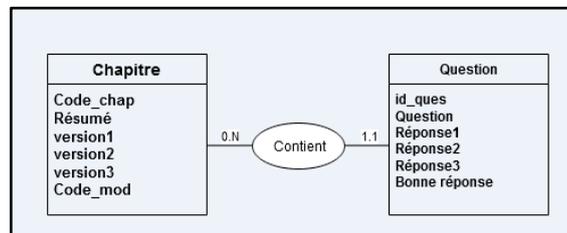


Figure 3.16 : les tables de micro-service du cours.

✚ **Micro-service du test final :**

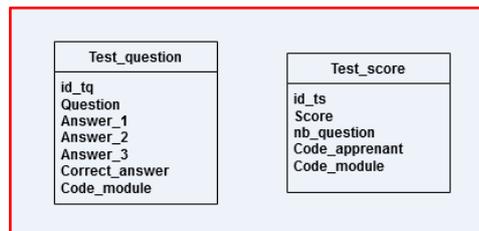


Figure 3.17 : les tables de micro-service du test final.

✚ **Micro-service collaboration :**

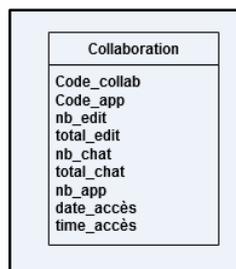


Figure 3.18 : les tables de micro-service collaboration.

✚ **Micro-service gestion des réunions :**

Agenda_reunion
Id
code_group_tut
date
temps begin
temps end
event
code_app
initiateur
type_reunion
valide
avis_tut
avis_app

Figure 3.19 : table de micro-service gestion des réunions.

### 3.8 Le MLD (Modèle Logique de Donnée)

✚ **Micro-service test de niveau :**

Table	Attributs
Question	Code_ques Type Niveau Enonce Réponse1 Réponse2 Réponse3 Bonne_réponse

Tableau 3.1 tableau du MLD de MS test de niveau.

✚ **Micro-service du cours :**

Table	Attributs
Chapitre	Code_chap Résumé Version1 Version2 Version3 Code_mod
Question	Id_ques Question Réponse1 Réponse2 Réponse3 Bonne_réponse

Tableau 3.2 : tableau du MLD de MS du cours.

✚ **Micro-service du test final :**

Table	Attributs
Test_question	Id_tq Answer1 Answer2 Answer3 Corresct_answer
Test_score	Id_ts Score nb_question Code_apprenant Code_module

**Tableau 3.3 :** *tableau du MLD de MS du test final.*

✚ **Micro-service collaboration :**

Table	Attributs
Collaboration	Code_collab Code_app Nb_edit Totl_edit Nb_chat Total_chat Nb_app Date_accès Ime_accès

**Tableau 3.4 :** *tableau du MLD de MS de collaboration.*

✚ **Micro-service gestion des réunions :**

Table	Attributs
Agenda_réunion	Id Code_group_tut Date Temps_begin Temps_end Event Code_app Initiateur Type_réunion Valide Avis_tut Avis_app

**Tableau 3.5 :** *tableau du MLD de MS de gestion des réunions.*

## 3.9 Conclusion

Dans ce chapitre nous avons présenté en détails les fonctionnalités de notre système sur la plateforme d'apprentissage à base de micro-service. Nous avons commencé par présenter les objectifs de notre système. Ensuite nous avons décrit l'architecture globale que nous avons proposée. Puis nous avons détaillé les schémas fonctionnels des espaces des différents acteurs dans la plateforme d'apprentissage de notre système.

Dans le chapitre suivant, nous allons présenter les principes de l'implémentation et les interfaces du système implémenté ainsi que ses différentes fonctionnalités.

# **Chapitre 4**

## **Implémentation du système**

## 4.1 Introduction

Après avoir présenté les principes de base de la conception de notre système. Dans ce chapitre nous présentons les détails d'implémentation de notre plateforme qui va être réalisée en utilisant des outils de développement de logiciels, en utilisant des langages de programmation. Nous présentons aussi les différentes interfaces offertes par le système.

## 4.2 Environnements logiciels de développement

Pour réaliser notre système, nous utilisons le XAMPP comme environnement de développement et phpMyAdmin qui permet la gestion, l'ajout, la modification et la suppression des contenus des bases de données de façon automatique.

Pour la création du code source et le design des pages web PHP nous avons utilisé un éditeur de texte open source Visual Studio Code .Cet éditeur qui prend en charge plusieurs langages de programmation : HTML, CSS, PHP, JavaScript.

Les services web sont développés à l'aide du langage PHP qui sera interprété au sein des serveurs distants. Il faut intégrer la bibliothèque NuSOAP pour le service web car elle permet d'envoyer et de recevoir des messages SOAP en utilisant le protocole http. Cette bibliothèque est développée sous forme de classes dans le langage PHP. Elle permet aussi traiter tous les types d'erreurs qui peuvent suspendre la communication entre le serveur et le client du service [9].

Dans ce qui suit nous présentons une brève explication des outils et langages utilisés :

### 4.2.1 HTML

HTML signifie « HyperText Markup Language ». Il est utilisé pour concevoir des pages web à l'aide d'un langage de balisage. C'est une combinaison d'hypertexte, qui définit le lien entre les pages Web, et de langage de balisage, qui est utilisé pour définir le document texte au sein de balises pour structurer les pages Web. Ce langage est utilisé pour annoter du texte afin que les machines puissent le comprendre et le manipuler en conséquence [SW13].

## 4.2.2 CSS

CSS signifie «Cascading Style Sheets» est un langage de conception simple destiné à simplifier le processus de création de pages Web présentables. Il gère l'apparence et la convivialité d'une page Web. À l'aide de CSS, vous pouvez contrôler la couleur du texte, le style des polices, l'espacement entre les paragraphes, la taille et la disposition des colonnes, les images ou couleurs d'arrière-plan utilisées, les conceptions de mise en page, les variations d'affichage pour différents appareils et tailles d'écran ainsi qu'une variété d'autres effets [SW14].

## 4.2.3 JavaScript

JavaScript est un langage de script qui vous permet de créer du contenu mis à jour de manière dynamique, de contrôler le multimédia, d'animer des images, des applications, des serveurs et même des jeux, et à peu près tout le reste [SW15].

## 4.2.4 PHP

PHP est un langage de script open source, interprété et orienté objet qui peut être exécuté côté serveur. PHP est bien adapté au développement Web. Par conséquent, il est utilisé pour développer des applications Web (une application qui s'exécute sur le serveur et génère la page dynamique) [SW16].

Certains points importants à noter à propos de PHP sont les suivants :

- PHP signifie Préprocesseur Hypertexte.
- PHP est un langage interprété, c'est-à-dire qu'il n'y a pas besoin de compilation.
- PHP est plus rapide que les autres langages de script, par exemple ASP et JSP.
- PHP est un langage de script côté serveur, utilisé pour gérer le contenu dynamique du site Web.
- PHP peut être intégré au HTML.
- PHP est un langage orienté objet.
- PHP est un langage de script open source.
- PHP est un langage simple et facile à apprendre.

### 4.2.5 La bibliothèque NuSOAP

Les services web sont développés à l'aide du langage PHP qui sera interprété au sein des serveurs distants. Il faut intégrer la bibliothèque NuSOAP pour le service web car elle permet d'envoyer et de recevoir des messages SOAP en utilisant le protocole http. Cette bibliothèque est développée sous forme de classes dans le langage PHP. Elle permet aussi de traiter tous les types d'erreurs qui peuvent suspendre la communication entre le serveur et le client du service [9].

### 4.2.6 Visual Studio Code

Visual Studio Code est un éditeur de code source gratuit, léger mais puissant qui s'exécute sur votre bureau et sur le Web et est disponible pour Windows, macOS, Linux et Raspberry Pi OS. Il est livré avec une prise en charge intégrée de JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages de programmation (tels que C++, C#, Java, Python, PHP et Go), des environnements d'exécution (tels que .NET et Unity), des environnements (tels que Docker et Kubernetes) et des cloud (tels qu'Amazon Web Services, Microsoft Azure et Google Cloud Platform). [SW17]

### 4.2.7 MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR) développé par Oracle et basé sur un langage de requête structuré (SQL).

Une base de données est une collection structurée de données. Il peut s'agir d'une simple liste de courses, d'une galerie de photos ou d'un endroit où stocker de grandes quantités d'informations dans un réseau d'entreprise. En particulier, une base de données relationnelle est un magasin numérique collectant des données et les organisant selon le modèle relationnel. Dans ce modèle, les tableaux sont constitués de lignes et de colonnes, et les relations entre les éléments de données suivent toutes une structure logique stricte. Un SGBDR est simplement l'ensemble d'outils logiciels utilisés pour réellement implémenter, gérer et interroger une telle base de données [SW18].

## 4.2.8 XAMPP

XAMPP est une abréviation où X signifie Cross-Platform, A signifie Apache, M signifie MYSQL et Ps signifie PHP et Perl, respectivement. Il s'agit d'un package open source de solutions Web qui inclut la distribution Apache pour de nombreux serveurs et exécutable de ligne de commande ainsi que des modules tels que le serveur Apache, MariaDB, PHP et Perl. [SW19]

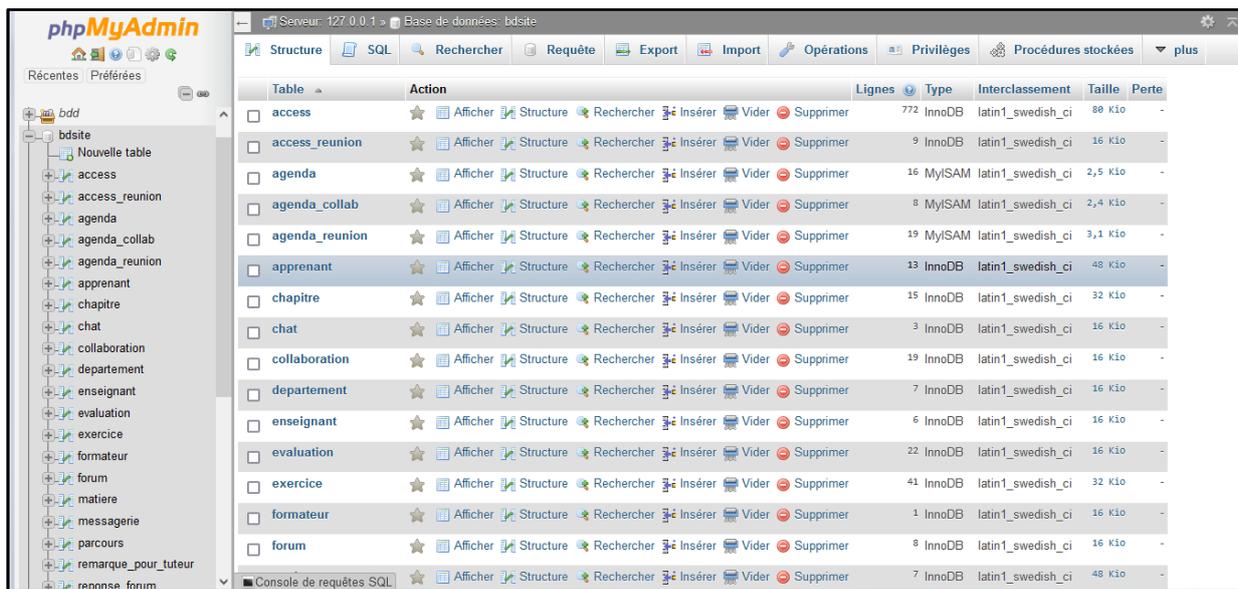


Figure 4.1 : Interface de gestion de BDD XAMPP.

## 4.3 Description du fonctionnement de notre projet

Notre plateforme d'apprentissage est composée de deux parties.

- 1- La partie principale qui représente la plateforme où les différents acteurs sont inscrits permettant d'accéder aux différentes fonctionnalités représentant ainsi le service client.
- 2- L'es microservices distants qui sont hébergés dans d'autres serveurs et qui fournissent un ensemble de fonctionnalités indépendantes qui peuvent être utilisées par n'importe quelle autre plateforme.

### 4.3.1 La configuration

Pour configurer et faire fonctionner les microservices nous avons commencé à les héberger sur 000WebHost, il faut commencez par s'inscrire et créer un nouveau site web dans le tableau de bord. Téléchargez ensuite les fichiers des microservices dans le dossier « public\_html » en utilisant le gestionnaire de fichiers intégrés ou un client FTP. Si les microservices nécessite une base de données, créez-en une via l'option "Manage Database" et configurez votre application pour utiliser les informations de connexion fournies. Enfin, accédez aux microservices via l'URL fournie par 000WebHost. Pour vérifier que tout fonctionne correctement et corriger les éventuelles erreurs en consultant les fichiers de logs.

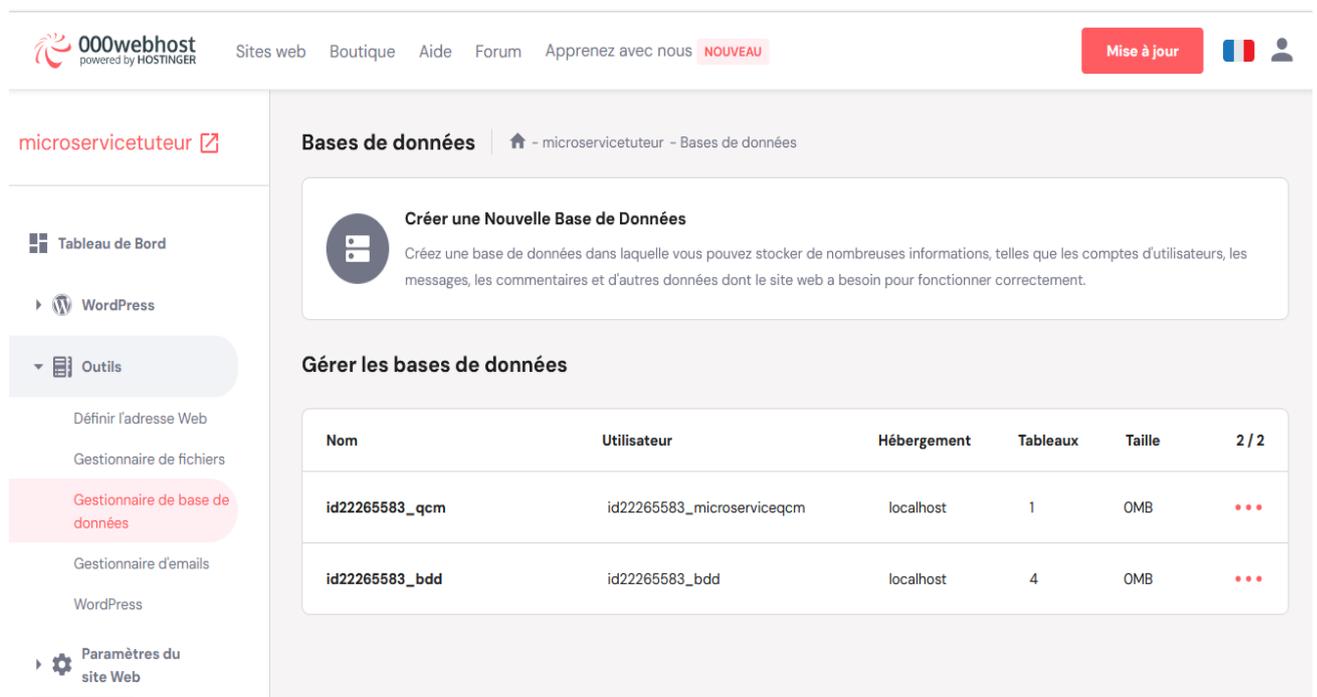


Figure 4.2 : Interface de site d'hébergement.

## 4.4 Présentation du système

### 4.4.1 La page d'accueil

Notre interface est une présentation de plusieurs pages enchainés entre elles par des liens hypertextes pour accéder aux pages ainsi qu'au micro-service.

La page d'accueil de notre système est illustrée par la **figure 4.3**. Il se compose de quatre parties: apprenant, tuteur, enseignant et administrateur. Chaque acteur a la possibilité d'atteindre son espace en passant par le menu principal du système.



**Figure 4.3** : Interface principale de la plateforme.

## **4. 4. 2** L'espace apprenant

### **4. 4. 2. 1** Inscription au système

Cela concerne tous les acteurs du système (apprenant, enseignant et tuteur) ils doivent s'inscrire et attendre la validation de l'administrateur pour se connecter au système.

L'apprenant doit s'inscrire dans le système pour devenir membre. Accéder à la page d'inscription et remplir un formulaire.

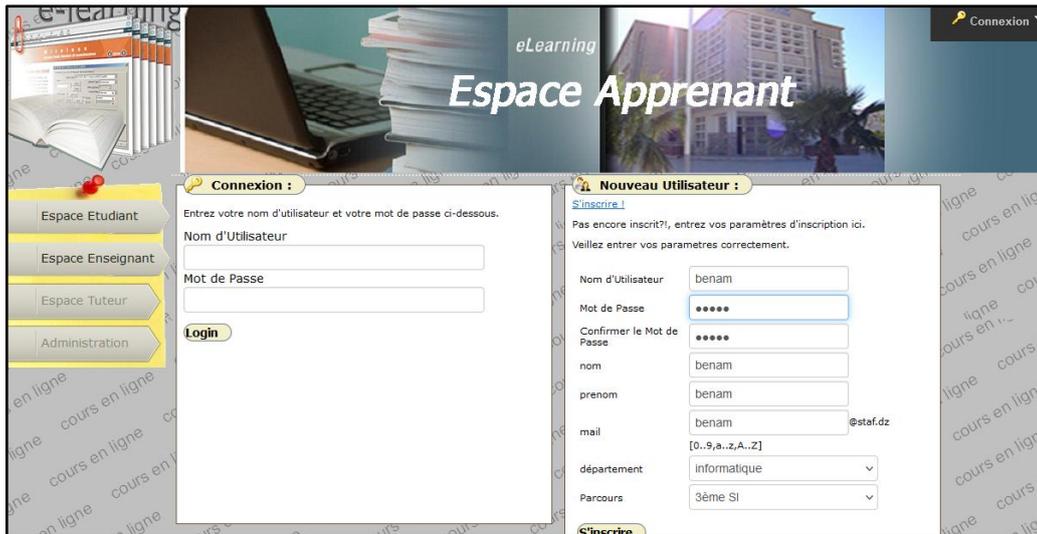


Figure 4.4 : Interface de l'espace apprenant.

#### 4. 4. 2. 2 Test du niveau cognitif

Après l'inscription, l'apprenant passe un test de niveau cognitif sous forme de QCM, développé en tant que micro-service.

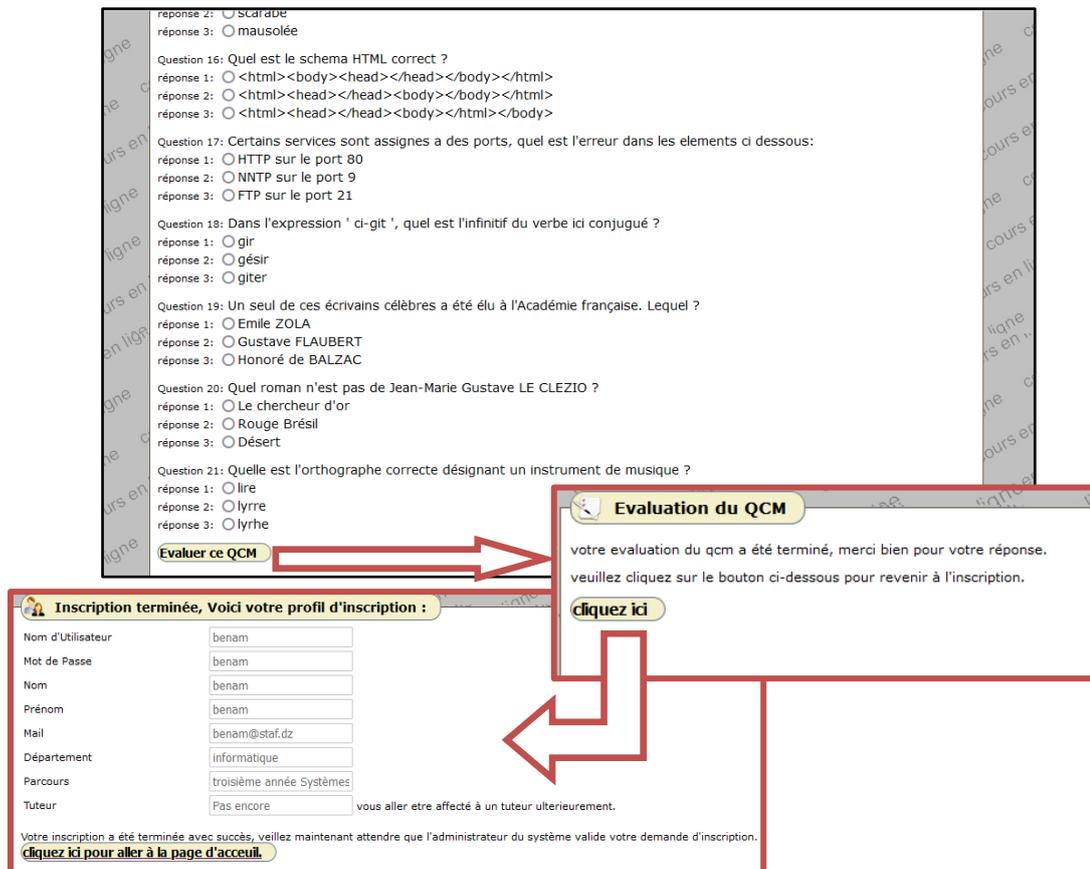


Figure 4.5 : Interface de test du niveau.

### 4. 4. 2. 3 Accès au cours et exercices

Une fois inscrit et validé par l'administrateur, l'apprenant a accès aux cours et peut évaluer les exercices.

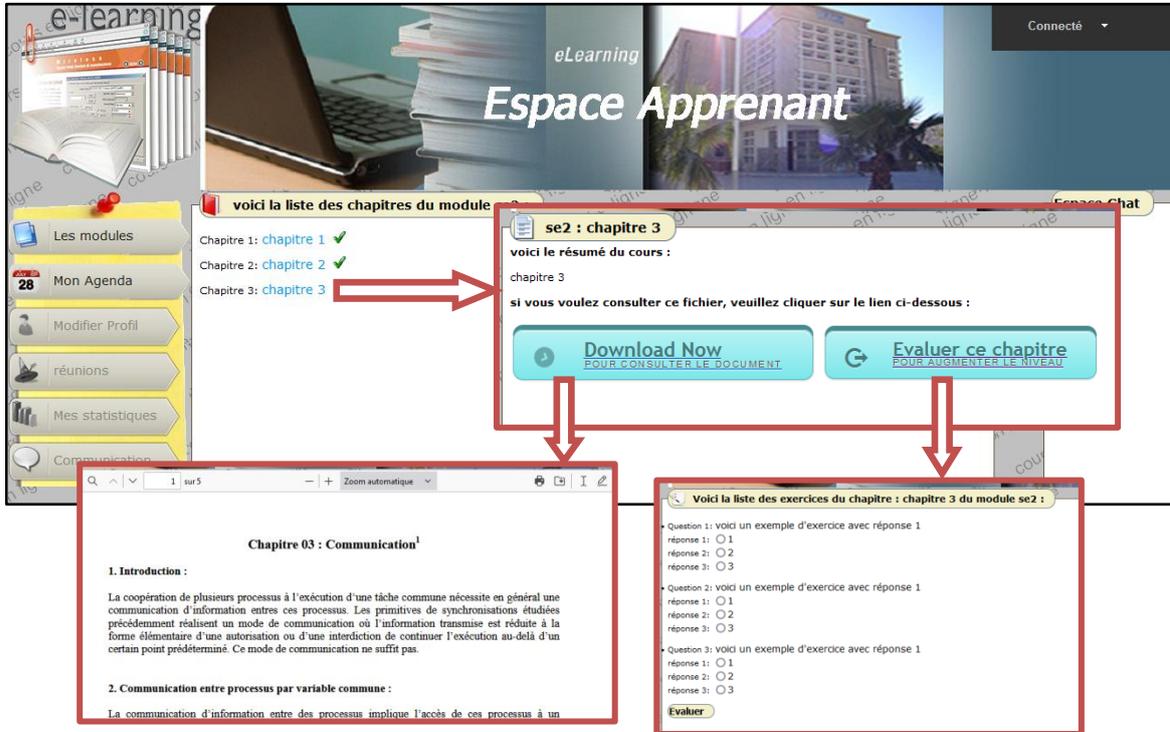


Figure 4.6 : Consultation des cours et évaluation des exercices.

### 4. 4. 2. 4 Test final du module

Si l'apprenant a déjà suivi ce module ou possède des connaissances préalables, il est autorisé à passer le test final qui est un microservice externe. Le score obtenu déterminera si l'apprenant a acquis ou non acquis le module.

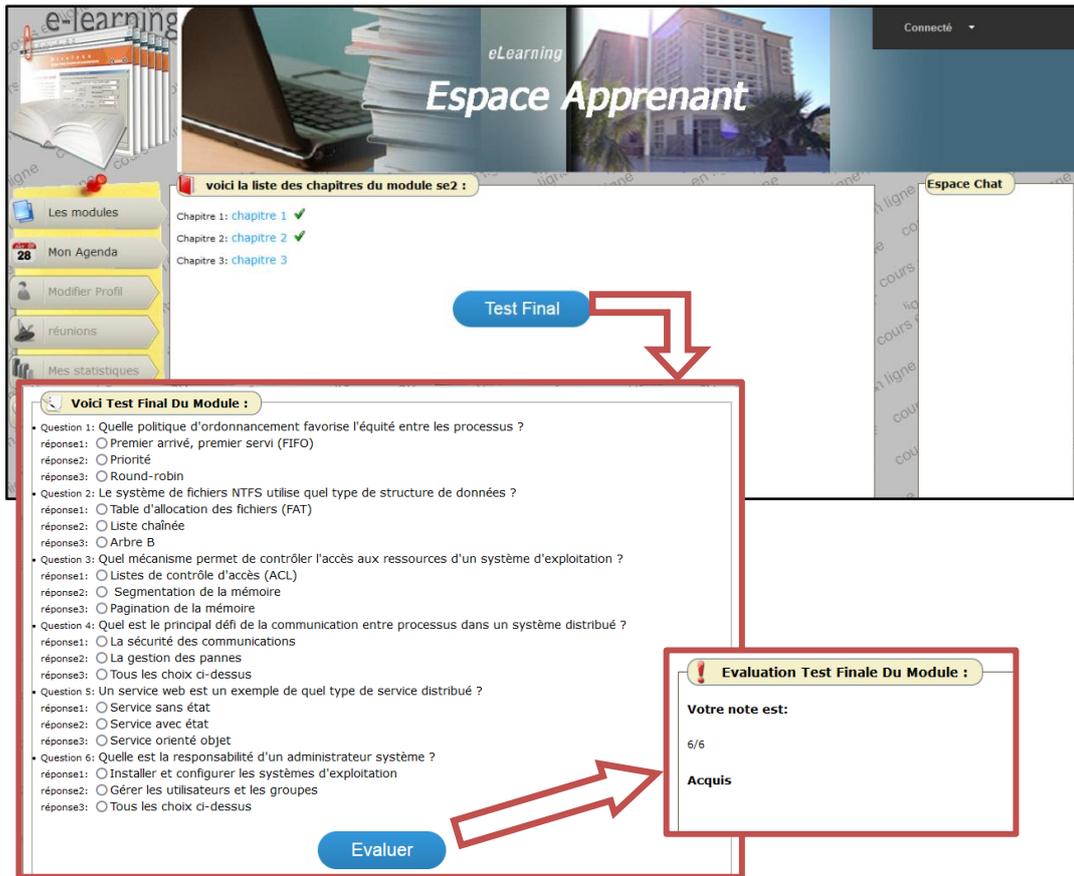


Figure 4.7 : Evaluation test final du module.

#### 4. 4. 2. 5 Collaboration

Ce microservice permet d'effectuer un travail collaboratif entre les apprenants du même groupe et fournir des informations au tuteur sur chaque session de collaboration.

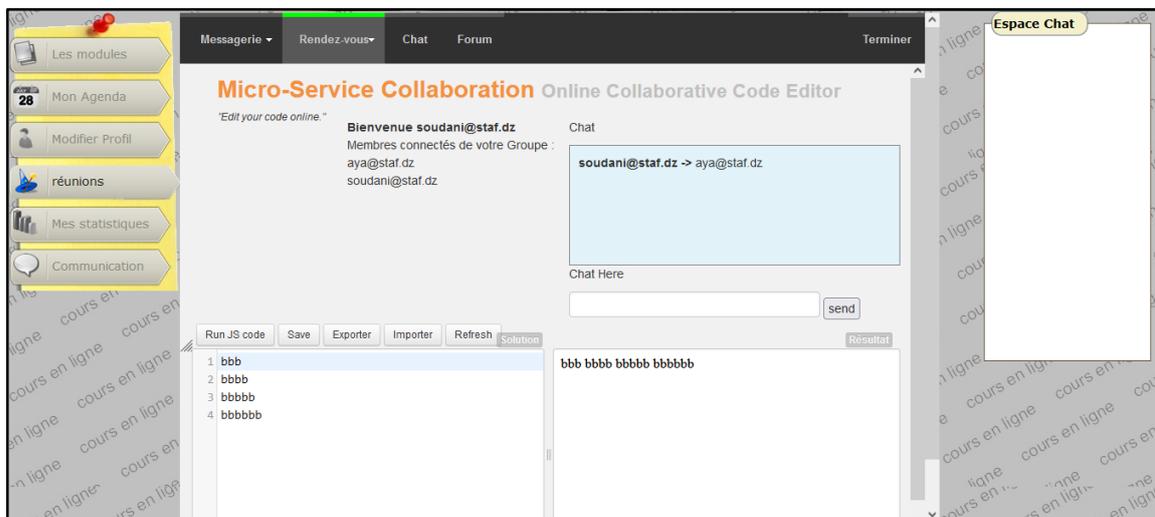


Figure 4.8 : Accueil de la collaboration.

#### 4. 4. 2. 6 Gestion réunion

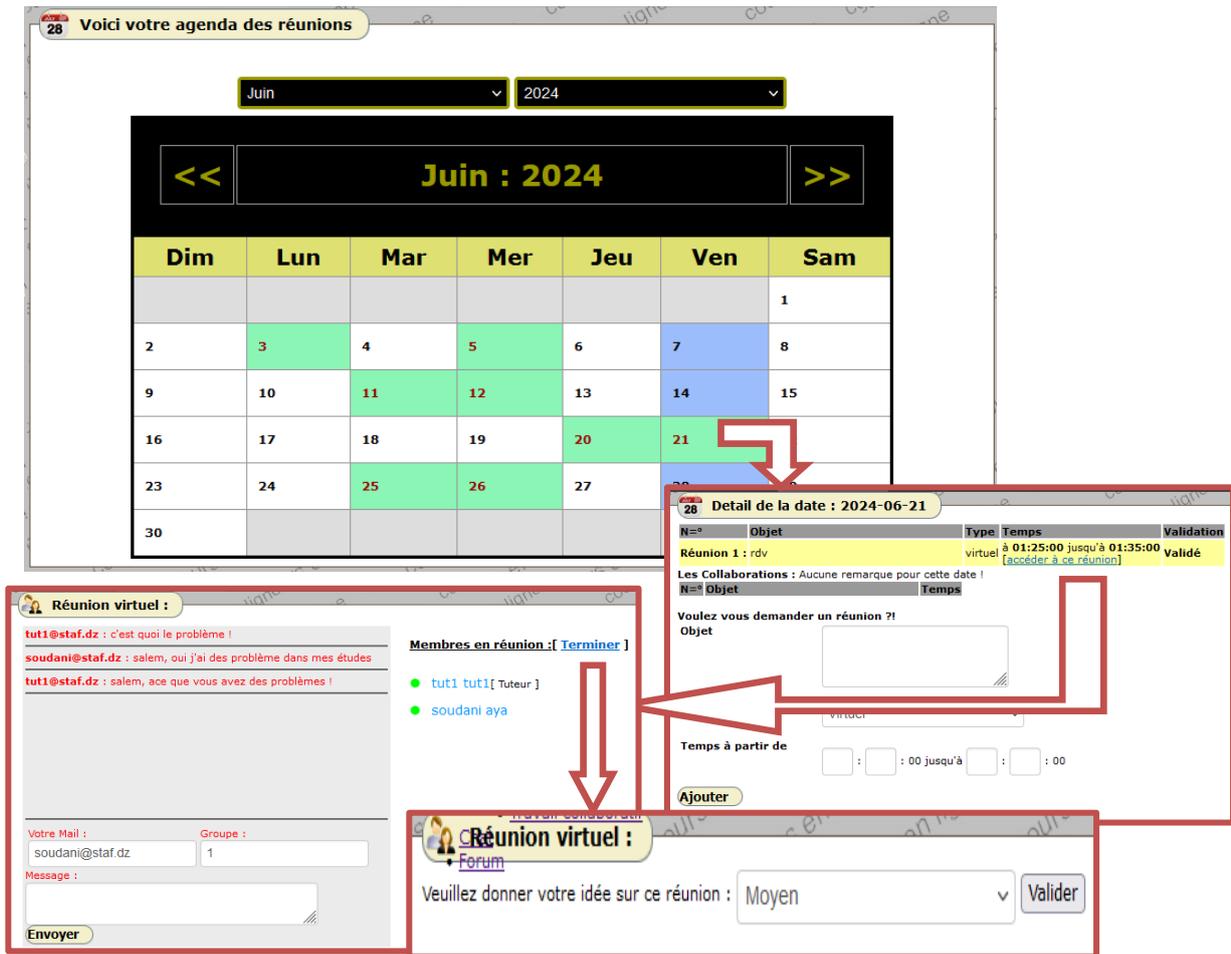


Figure 4.9 : Programmation et accès à la réunion

#### 4. 4. 3 L'espace tuteur

L'espace de tuteur sera affiché comme suit :

- L'espace de tuteur affiche la liste de son groupe avec un menu et espace chat.

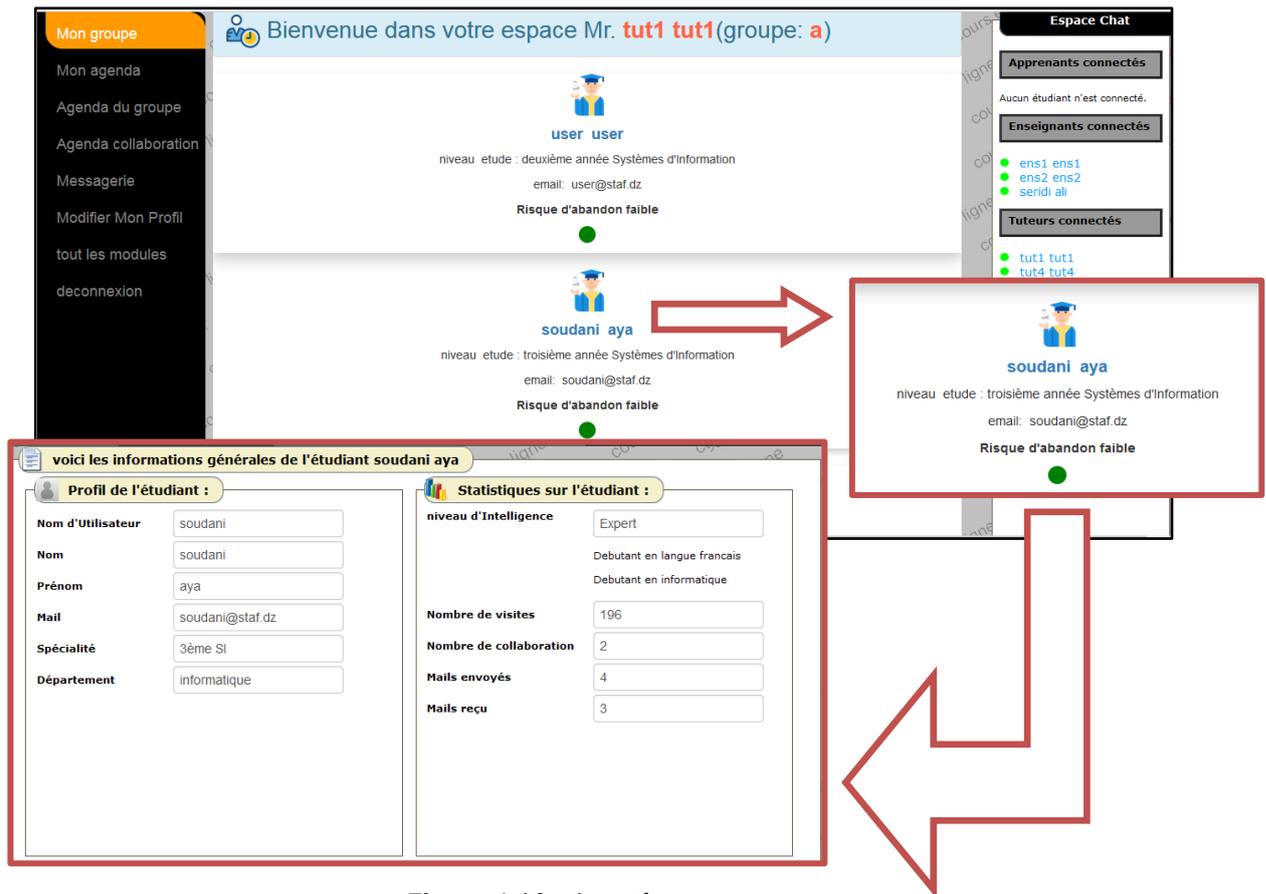


Figure 4.10 : Accueil espace tuteur.

- La fonction suivie d'accès est offerte par un micro-service qui affiche le graphe d'accès de chaque apprenant par mois, par 10 jours ou par trimestre.

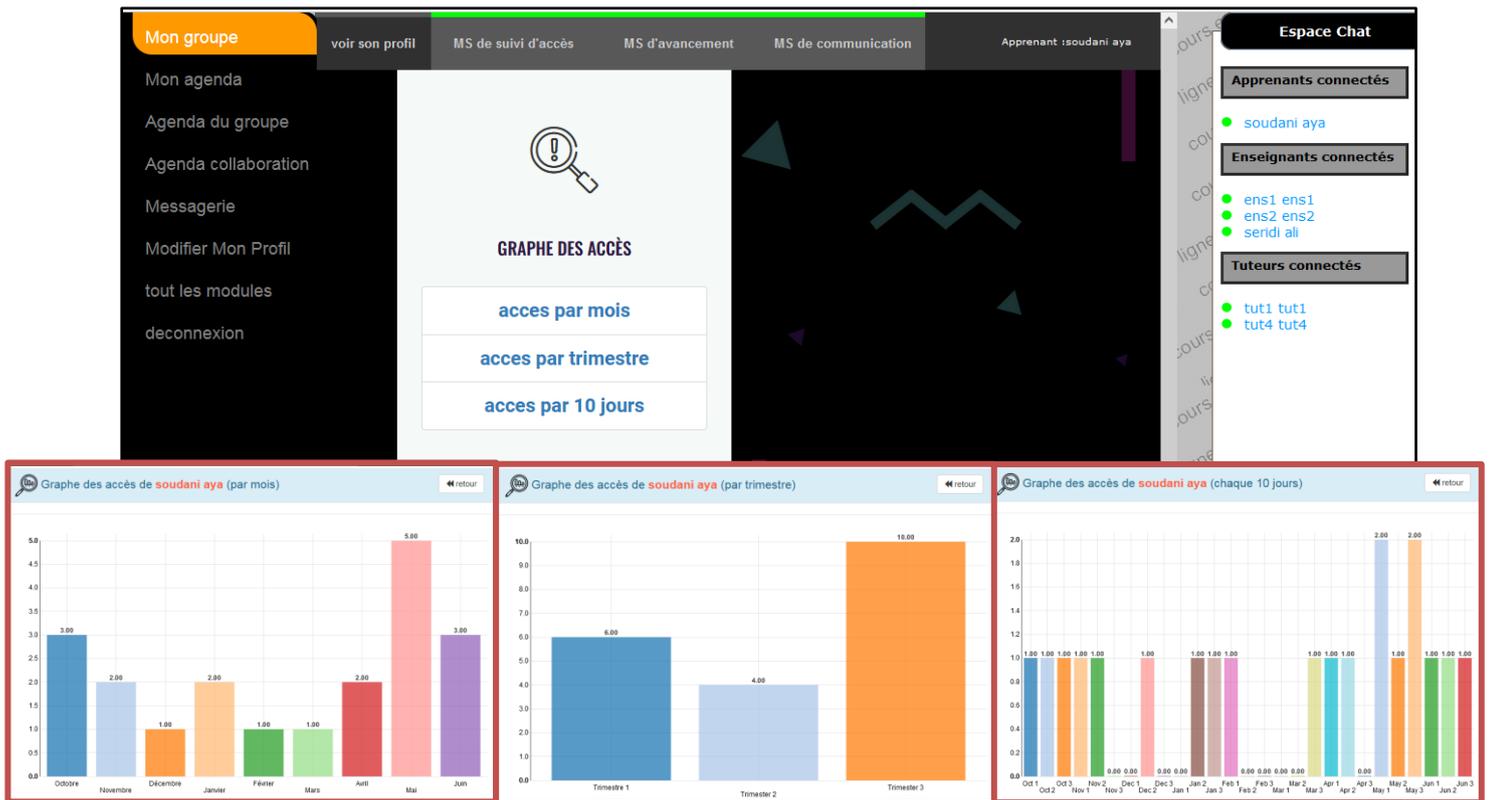


Figure 4.11 : Graphe suivi d'accès.

- Dans l'espace d'avancement, Chaque module est présenté avec le nom de l'enseignant en charge, ce qui permet au tuteur de le contacter en cas de souci. Il est aussi donné un aperçu de l'avancement total ainsi que du nombre de tentatives d'évaluation pour chaque module. En outre, il y a une représentation graphique qui indique l'évolution des étudiants dans les cours.

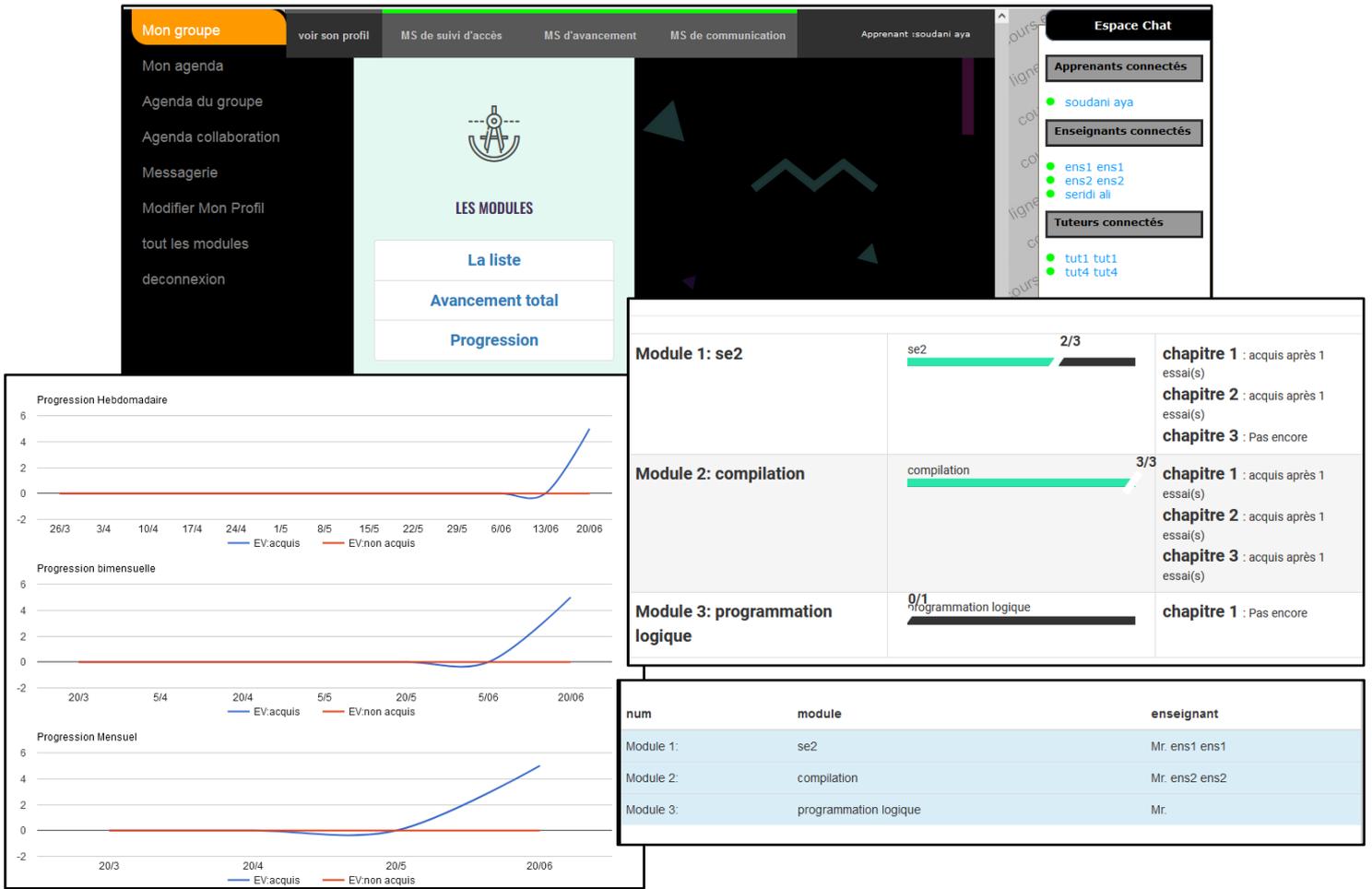


Figure 4.12 : Liste des cours, l'avancement d'étude et progression dans les cours

- Dans l'espace de communication, le tuteur trouve toutes les statistiques sur les collaborations et les chats de l'apprenant, ainsi que les détails sur les réunions et les participations aux forums.

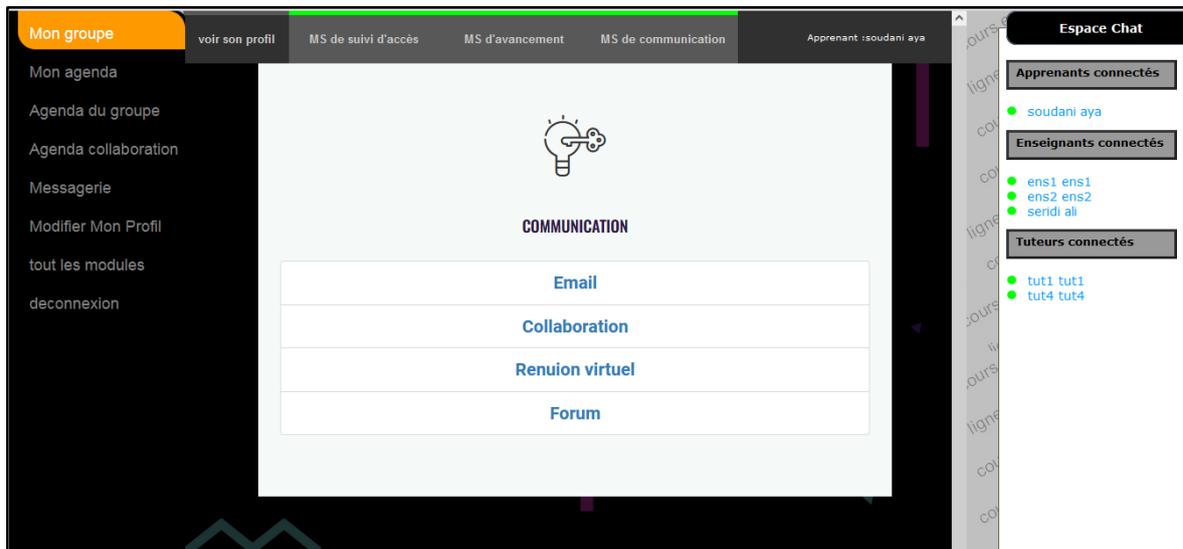


Figure 4.13 : Espace de communication.



Figure 4.14 : Statistique sur les emails.

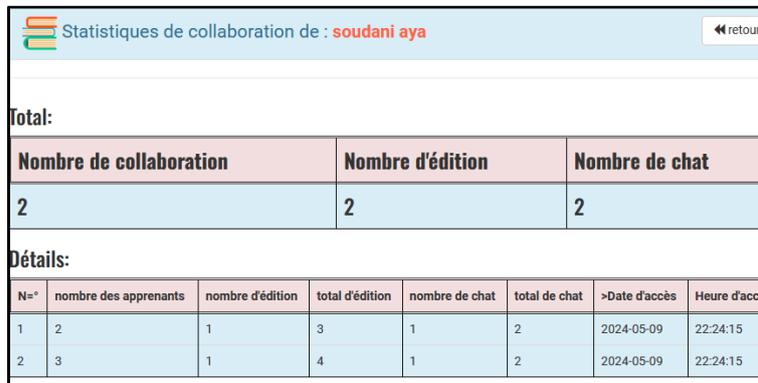


Figure 4.15 : Statistique sur les collaborations.

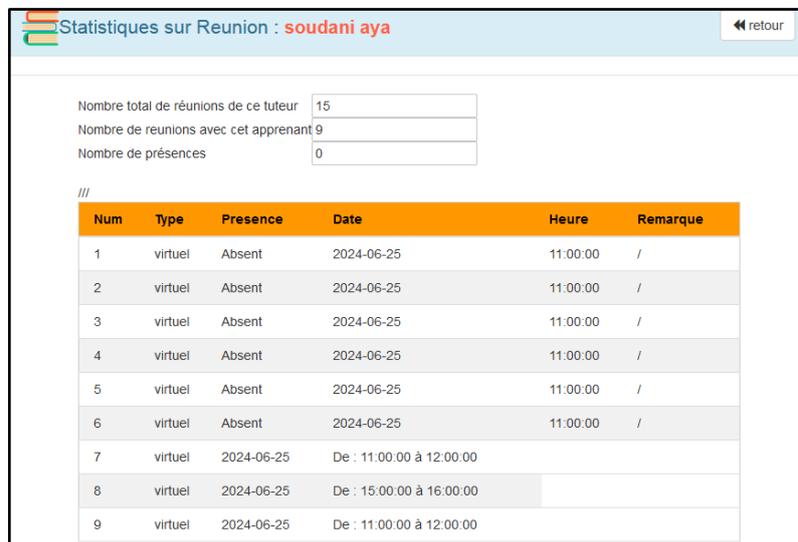


Figure 4.16 : Statistique sur réunion.

- Dans la Collaboration :

The screenshot displays a web application interface for collaboration. On the left is a navigation sidebar with options like 'Mon groupe', 'Mon agenda', and 'Agenda collaboration'. The main area shows a calendar for June 2024. A red arrow points from the date '21' in the calendar to a 'Detail de la date : 2024-06-21' panel. This panel lists a meeting 'Réunion 1 : collab' from 03:15:00 to 03:20:00. Below the meeting details is a chat area with a 'Terminer' button. A second red arrow points from the chat area to a 'RDV de collaboration' form, which asks for an idea on the meeting. A third red arrow points from the form to a 'Micro-Service Collaboration Online Collaborative Code Editor' window, which includes a chat area and a code editor with JS code.

Figure 4.17 : Programmé et accès à la collaboration

- Dans la gestion de réunion :

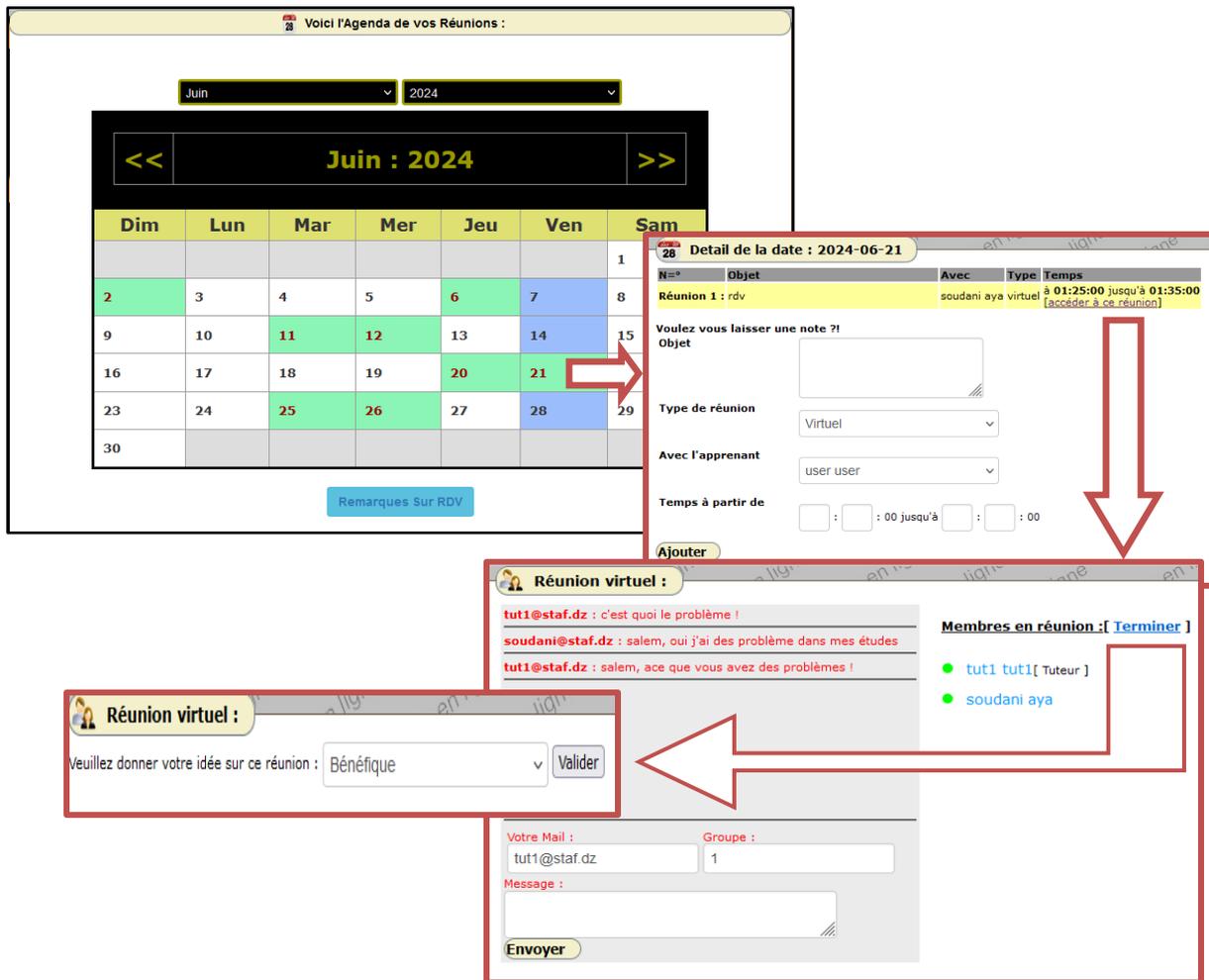


Figure 4.18 : Programmation et accès à la réunion.

## 4.5 Conclusion

Dans le dernier chapitre, nous avons abordé la phase de réalisation de notre application. Nous avons présenté les outils logiciels essentiels qui ont facilité notre travail, et les langages de programmation utilisés. Par la suite, nous avons exposé notre système basé sur les microservices en détaillant ses différentes fonctionnalités et en mettant en avant plusieurs interfaces que nous avons développées.

# **Conclusion générale**

# Conclusion générale

Notre objectif principal est de concevoir et de développer une plateforme d'apprentissage adaptative basée sur les microservices. En adoptant cette architecture modulaire, nous répondons aux besoins croissants de flexibilité, de scalabilité et de maintenabilité des systèmes éducatifs modernes.

Contrairement aux architectures monolithiques traditionnelles, les architectures microservices permettent de décomposer les différentes fonctionnalités de la plateforme en modules autonomes, chacun ayant une responsabilité bien définie et fonctionnant indépendamment, facilitant ainsi l'évolution rapide aux nouvelles exigences et l'optimisation continue de chaque composant. Cette approche de granularité fine, où chaque module assure une seule fonctionnalité ou un ensemble cohérent de fonctionnalités réutilisables, améliore considérablement la performance, la scalabilité et la flexibilité de la plateforme.

À travers une analyse détaillée des fonctionnalités essentielles d'une plateforme e-learning, nous avons identifié les microservices nécessaires à cette intégration. Cela nous a permis non seulement de doter notre système de fonctionnalités externes, mais aussi de mettre à disposition ces fonctionnalités à d'autres plateformes qui en sont dépourvues.

Nous estimons avoir atteint les objectifs que nous nous étions fixés en développant une plateforme e-learning très modulaire en utilisant des microservices.

Comme perspectives à notre travail, nous proposons de :

- Utiliser les technologies d'IA pour assurer des adaptations encore plus précises et en temps réel des parcours d'apprentissage, basées sur les performances et les préférences des apprenants et cela à travers des microservices afin de pouvoir réutiliser facilement ses fonctionnalités.

# **Bibliographie**

# Bibliographie

- [1] Erl, T. (2008). *SOA: principles of service design*. Prentice Hall.
- [2] Sériidi, A. (2021). Réingénierie et restructuration d'applications existantes vers des architectures à base de services. Thèse de doctorat, Université Badji Mokhtar Annaba, Faculté des Sciences de l'Ingéniorat, Département d'informatique.
- [3] Gu, H., Yang, S., Gu, M., & Yuan, M. (2022). Research on online teaching platform system based on microservice architecture. In *MATEC Web of Conferences* (Vol. 355, p. 03058). EDP Sciences. <https://doi.org/10.1051/mateconf/202235503058>
- [4] Palamarchuk, Y. . (2022). METHODS OF BUILDING MICROSERVICE ARCHITECTURE OF E-LEARNING SYSTEMS. *Information Technology and Computer Engineering*, 53(1), 43–54. <https://doi.org/10.31649/1999-9941-2022-53-1-43-54>
- [5] Karmel, A., Chandramouli, R., & Iorga, M. (2016). *Nist definition of microservices, application containers and system virtual machines* (No. NIST Special Publication (SP) 800-180 (Draft)). National Institute of Standards and Technology.
- [6] Benraouane, S. A. (2011). *Guide pratique du e-Learning*. Paris.
- [7] Kerrouzi, R. L'enseignement en ligne à l'université algérienne à l'heure du covid-19: entre apport et difficultés d'appropriation.
- [8] Belbachir, F. (2016). Le e-learning comme méthode d'apprentissage. Mémoire de Master Académique, Université Abou Bekr Belkaid de Tlemcen, Faculté des Langues Etrangères.
- [9] Mabrek, Z., & Kahli, L. (2018). Développement d'un service web d'aide aux tuteurs dans une plateforme E-learning (Mémoire de licence). Département d'informatique, Université de 8 mai 1945, Guelma.
- [10] Artamonov, Y., Golovach, I., & Zymovchenko, V. (2021). Use analysis of microservices in e-learning system with multivariant access to educational materials. *Technology Audit and Production Reserves*, 4(2(60)), 45-50.

- [11] Kurniawan, D., & Dzikri, A. (2024, January). Development of a Prototype for Microservices Architecture and API Gateway Integration in an eLearning Platform. In *Proceedings of the 6th International Conference on Applied Engineering, ICAE 2023, 7 November 2023, Batam, Riau islands, Indonesia*.
- [12] Milovanović, A. (2021). Microservice architecture in E-learning. *International Conference on E-business Technologies (EBT)*, 1(1), 105–108.
- [13] Rishabh, R., & Surendra, G. (2022). A novel learning management system based on microservice architecture using Moodle. *International Journal of Creative Research Thoughts (IJCRT)*, 10(12).

# Webographie

- [SW1] Reference Model for Service Oriented Architecture 1.0. <https://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf> . Dernier accès au site : 01/04/2024.
- [SW2] <http://adslbox.free.fr/rapports/rapport-gl-service-oriented-architecture.pdf> . Dernier accès au site : 01/04/2024.
- [SW3] <https://aws.amazon.com/fr/what-is/service-oriented-architecture/>. Dernier accès au site : 17/04/2024.
- [SW4] <https://web.maths.unsw.edu.au/~lafaye/CCM/web-services/soa-architecture-orientee-services.htm> . Dernier accès au site : 20/04/2024.
- [SW5] <https://www.zdnet.fr/actualites/soa-comprendre-l-approche-orientee-service-39206712.htm> . Dernier accès au site : 20/04/2024.
- [SW6] <https://www.redsen.com/architecture-entreprise/concepts-fondamentaux-soa/>. Dernier accès au site : 22/04/2024.
- [SW7] *How to Build and Scale with Microservices.*  
<https://cloud.kapostcontent.net/pub/93f35b9e-ca51-4fc9-8958-ad7cc5ccc289/how-to-build-and-scale-with-microservices-french.pdf>. Dernier accès au site: 01/05/2024.
- [SW8] <https://medium.com/nerd-for-tech/microservice-architecture-622e4148f1>. Dernier accès au site : 05/05/2024.
- [SW9] Architectures-micro-services-objectifs-benefices-et-defis-partie-1.  
<https://www.sqli.com/fr-fr/insights-news/blog/architectures-micro-services-objectifs-benefices-et-defis-partie-1> . Dernier accès au site : 05/05/2024.
- [SW10] <https://www.chakray.com/fr/que-sont-les-microservices-definition-caracteristiques-avantages-et-inconvenients/>. Dernier accès au site : 07/05/2024.
- [SW11] <https://www.crowdstrike.fr/cybersecurity-101/cloud-security/soa-vs-microservices/>. Dernier accès au site : 10/05/2024.
- [SW12] <https://logic2020.com/insight/microservices-architecture-choosing-your-approach/>. Dernier accès au site : 10/05/2024.
- [SW13] <https://www.geeksforgeeks.org/html-introduction/>. Dernier accès au site : 16/06/2024.
- [SW14] <https://www.tutorialspoint.com/css/what-is-css.htm>. Dernier accès au site : 16/06/2024.
- [SW15] [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript). Dernier accès au site : 16/06/2024.

[SW16] <https://www.javatpoint.com/php-tutorial>. Dernier accès au site : 16/06/2024.

[SW17] <https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsoft.html>. Dernier accès au site : 16/06/2024.

[SW18] <https://www.talend.com/resources/what-is-mysql/>. Dernier accès au site : 16/06/2024.

[SW19] <https://www.javatpoint.com/xampp>. Dernier accès au site : 16/06/2024.