



Rapport de Projet (8INF951-TA)

UNIVERSITÉ DU QUÉBEC À RIMOUSKI

Développement d'une plateforme web à base de microservices pour le domaine de la santé

Maîtrise en informatique (profil professionnel)
Département de mathématiques, informatique et génie

SESSION Automne 2023

Réalisé par :
Zaid TIDJET

Encadré par :
Pr Mehdi ADDA

Contents

1	Introduction et Contexte	2
1.1	Contexte	2
1.2	Problématique	2
1.3	Solution	3
2	Développement Web	3
2.1	Quelques définitions	3
2.1.1	Qu'est ce que le Web	3
2.1.2	Pages Web	3
2.1.3	Lien hypertextes	3
2.1.4	Protocole HTTP (HyperText Transfer Protocol)	3
2.1.5	C'est quoi un site Web	3
2.1.6	C'est quoi un navigateur web	3
2.1.7	C'est quoi un serveur web	3
2.1.8	Qu'est ce que c'est les Micro-services	4
2.2	Architecture d'application web	4
3	Langages, bibliothèques et Framework	5
4	ASP.NET Core Models-Views-Controllers(MVC).	5
4.1	Qu'est ce que ASP.Net	5
4.2	Le Paradigme MVC	5
5	Front-End	5
5.1	Page d'accueil :	6
5.2	A propos	6
5.3	Professionnel de la santé	7
5.4	Nous joindre et aides	8
5.5	Chercher un médecin	8
5.6	Page de Connexion	8
5.7	Créer un compte et prendre un Rendez-vous	9
6	Backend	10
6.1	Utilisation ASP.Net Core MVC	10
6.1.1	Ajouter un Controllers	10
6.1.2	Ajouter un models	10
6.2	Persistance des données avec Entity Framework (ORM).	10
6.2.1	Introduction et Contexte	10
6.2.2	Persistance des données	10
6.2.3	SGBD Ou DBMS	11
6.2.4	Bases de données relationnelles	11
6.2.5	Qu'est-ce qu'un ORM ?	12

7	Authentification et autorisation	13
7.1	Qu'est ce que l'authentification ?	13
7.2	Qu'est ce que l'autorisation ?	13
7.3	Authentification et autorisation	13
7.3.1	Authentication	13
7.3.2	Authorization	13
8	Utilisation de Dapr. Net	14
8.1	Dapr HTTP et Dapr gRPC	14
8.2	Utilisation de .Net Dapr SDK	14
9	automatisation des tests fonctionnels	15
9.1	Selenium Web Driver	15
9.1.1	Definition:	15
9.1.2	Option de navigateur	16
9.1.3	Pilote Web	16

List of Figures

1	Architecture des Applications Web.	4
2	Aperçu de la page d'accueil	6
3	Aperçu de la page A propos	7
4	Aperçu de la page professionnel de santé	7
5	Aperçu de la page Nous rejoindre et aide	8
6	Aperçu de la page Chercher un médecin	8
7	Aperçu de la page de Connexion	9
8	Aperçu des pages du création du compte et de rendez-vous	9
9	Dapr et le problème qu'il résout	14

Abstract

tout au long du travail j'ai développé une application web pour une clinique médicale, en utilisant ASP.NET Core MVC pour le framework de l'application. Pour la gestion des données, j'ai utilisé Entity Framework. Dapr est intégré pour faciliter la création et la gestion de microservices. Enfin, j'ai automatisé les tests fonctionnels avec Selenium WebDriver, garantissant ainsi une meilleure qualité et fiabilité de l'application. Ce projet combine des technologies modernes pour créer une solution robuste et efficace dans le domaine médical.

1 Introduction et Contexte

1.1 Contexte

L'avènement d'Internet et l'utilisation croissante des dispositifs de communication ont eu un impact profond sur la société, la culture, l'économie et la façon dont nous interagissons les uns avec les autres. De plus en plus de sites et d'applications Web sont déployés chaque jour.

1.2 Problématique

Dans les développement des applications web avec de fonctionnalités plus complexes, Soucis sur la sécurité des données et développement rapide sans partir from scratch avec les bonnes pratiques.

1.3 Solution

Faire appel aux technologies du Web en termes de bibliothèques et de frameworks est une pratique courante dans le développement web. Ces outils sont utilisés pour simplifier et accélérer le processus de développement, améliorer la performance, la sécurité et la convivialité des sites web et des applications web. Nous allons utiliser des bibliothèques et des frameworks les plus couramment utilisés dans le développement web dans notre application comme ASP.Net core MVC, La persistance des données avec Entity Framework(ORM), Dapr docker et enfin on procède aux développements des tests automatisés fonctionnels avec Selenium WebDriver.

2 Développement Web

2.1 Quelques définitions

2.1.1 Qu'est ce que le Web

ou World Wide Web (années 1980) est un système permettant l'organisation visuelle et la publication de documents, leur consultation via navigateur et leur interconnexion à l'aide de liens. Il a été développé par Tim Berners Lee.

2.1.2 Pages Web

Qui est le support du contenu textuel publié sur le Web.

2.1.3 Lien hypertextes

Où **URL (Uniforme Resource Locator)** permet la liaison entre différentes pages Web.

2.1.4 Protocole HTTP (HyperText Transfer Protocol)

Mécanisme permettant aux navigateurs Web d'accéder aux pages Web.

2.1.5 C'est quoi un site Web

c'est l'ensemble de pages web et de ressources reliées par des hyperliens accessible via une adresse Web.

2.1.6 C'est quoi un navigateur web

c'est un logiciel, programme ou application côté client permettant à l'internaute de communiquer avec des logiciels serveurs afin d'obtenir de l'information (pages Web).

2.1.7 C'est quoi un serveur web

c'est un ordinateur robuste et sécurisé sur lequel un logiciel, également appelé serveur web est chargé de répondre aux requêtes des internautes en leur servant les pages voulues ou en renvoyant un message d'erreur.

2.1.8 Qu'est que c'est les Micro-services

Les microservices sont une architecture de développement d'applications dans laquelle une application monolithique est décomposée en un ensemble de petits services autonomes et indépendants, chacun responsable d'une tâche ou d'une fonction spécifique. Ces services communiquent entre eux via des API (interfaces de programmation d'applications) bien définies et généralement exposées via HTTP ou d'autres protocoles

2.2 Architecture d'application web

- **Fonctionnement d'un Site Web Simple :**

Saisie d'URL ou Clic sur Lien : L'utilisateur entre l'URL dans son navigateur ou clique sur un lien. après Le navigateur trouve l'adresse IP de l'application via les serveurs DNS. Le navigateur envoie une requête HTTP à l'application. Ensuite L'application renvoie le contenu, généralement une page web. Et enfin Le navigateur affiche la page web à l'écran.

- **Traitement par le Serveur :**

L'équilibreur de charge envoie la requête à une instance de serveur disponible, qui traite la requête et appelle les API externes si nécessaire. Le serveur compile les résultats et les renvoie à l'équilibreur de charge. L'équilibreur de charge envoie la réponse à la passerelle API, qui la transmet à l'utilisateur via le client frontend, marquant la fin de la requête.

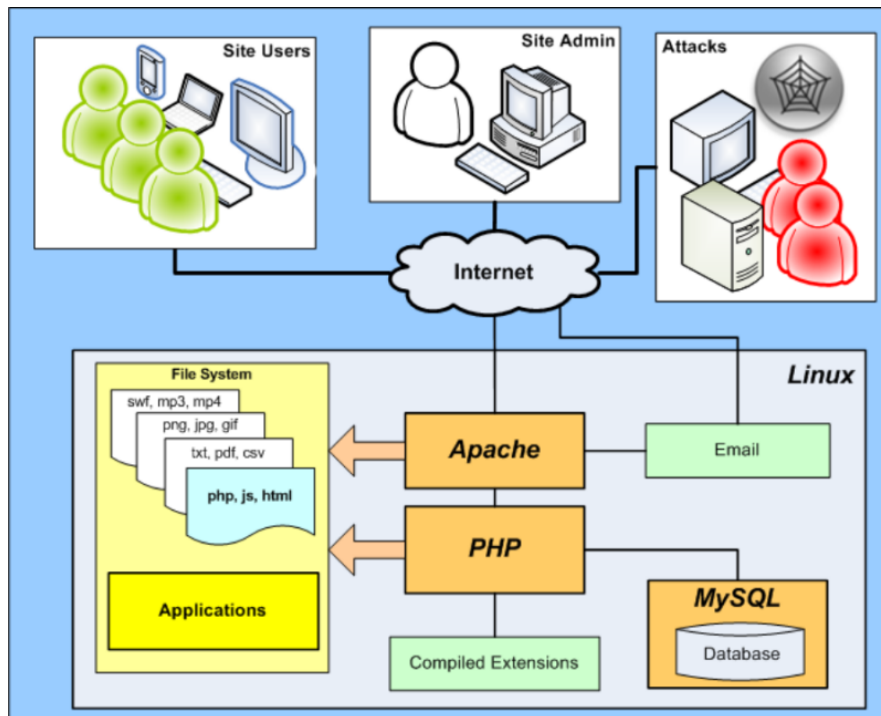


Figure 1: Architecture des Applications Web.

3 Langages, bibliothèques et Framework

Le développement Web se divise en deux catégories principales **front-end et Backend**, le premier est celui du Côté client construit à l'aide d'une combinaison de technologies telles que le langage Hypertext Markup Language (HTML), JavaScript et les feuilles de style en cascade (CSS). Les développeurs front-end conçoivent et construisent les éléments d'expérience utilisateur sur la page Web ou l'application, notamment les boutons, les menus, les pages, les liens, les graphiques et bien plus encore.

Le back-end, également appelé côté serveur, se compose du serveur qui fournit des données sur demande, de l'application qui les canalise et de la base de données qui organise les informations. Il utilise plusieurs langages de programmation comme **python** (Django et flask), **Java** (Java EE), **C** (ASP.Net), **PHP** (Laravel), **Ruby** (ruby on Rails)...

4 ASP.NET Core Models-Views-Controllers(MVC).

4.1 Qu'est ce que ASP.Net

ASP.net est un framework de serveur Open Source qui est utilisé pour gérer la logique côté serveur, la gestion des bases de données, la sécurité, et d'autres processus côté serveur. Il est destiné à la programmation web développée et maintenu par Microsoft, initialement créé en 2007 par Scott Guthrie, le paradigme de programmation MVC lui a été attribué en 2009, il peut générer des pages web dynamiques se fait à travers la syntaxe Razor.

4.2 Le Paradigme MVC

- **Définition** : c'est un motif d'architecture logicielle ou design pattern destiné à la conception d'interface graphique lancé en 1978. Il est souvent utilisé pour la conception de sites et applications Web.

Objectif : Le paradigme MVC (Modèle-Vue-Contrôleur) dans ASP.NET Core vise à séparer les différentes parties de l'application pour faciliter le développement, la maintenance, et le test de l'application. Voici les objectifs clés de l'utilisation de MVC dans ASP.NET

Il est composé de trois principaux éléments :

- **Modèle** : Représente la structure des données et la logique métier. Il interagit avec la base de données et traite les données.
- **Vue** : responsable du rendu graphique ou partie apparence,
- **Contrôleur** : Agit comme intermédiaire entre le modèle et la vue. Il reçoit les entrées de l'utilisateur, les transmet au modèle pour traitement, puis sélectionne une vue pour afficher la réponse.

5 Front-End

Vu qu'on j'avais utilisé ASP.Net core dans l'application la partie du Front-end est les Razor Pages /MVC Views qui permet de créer des vues Razor Pages ou le modèle MVC (Modèle-Vue-Contrôleur), qui sont utilisées pour générer le HTML envoyé au navigateur.

5.1 Page d'accueil :

Notre application web est une plateforme en ligne pour une clinique médicale. Qui est conçue pour offrir une expérience utilisateur fluide et informative. La page d'accueil, avec son agencement clair et ses couleurs apaisantes, invite les visiteurs à en apprendre davantage sur les services médicaux disponibles. Les utilisateurs sont accueillis par un grand titre qui annonce la clinique, suivi d'un sous-titre qui promet des soins de qualité pour toute la famille.

La navigation est intuitive avec une barre de menu en haut, comprenant des onglets pour l'accueil, des informations sur la clinique, les professionnels de la santé, ainsi que les moyens de contacter la clinique ou de trouver de l'aide. Un bouton distinct invite également les utilisateurs à se connecter à leur tableau de bord personnel, ce qui suggère une fonctionnalité de suivi des soins pour les patients.

Dans la partie principale de la page, trois sections clairement définies présentent les services principaux de la clinique : les consultations, les urgences et la pharmacie. Chacune de ces sections est accompagnée d'une brève description qui met en avant les atouts de la clinique, comme l'accès à des services de consultations médicales sans rendez-vous, des services d'urgence de qualité et un accès fiable à des médicaments et conseils de santé.

En bas de la page, on trouve le copyright de la Clinique Médicale, Inc., datant de l'année en cours, ce qui indique que le site est bien maintenu et à jour.

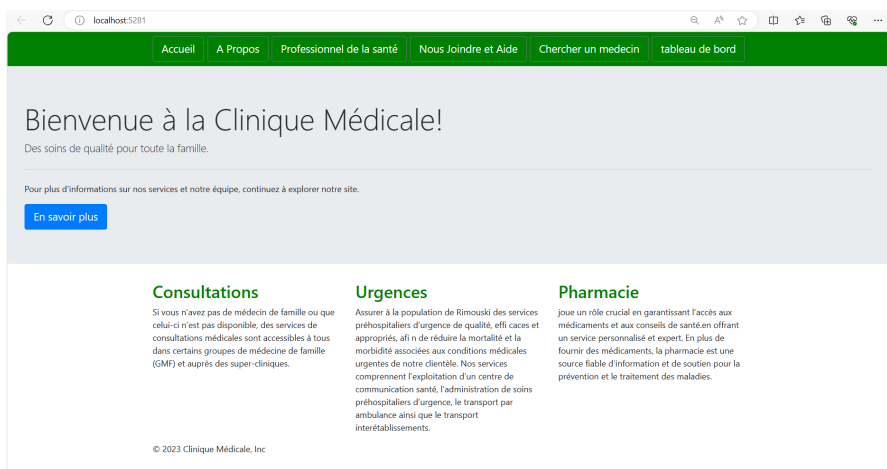


Figure 2: Aperçu de la page d'accueil

5.2 A propos

La section "À propos" du site de la Clinique Médicale fournit des informations détaillées sur l'engagement de la clinique à faciliter l'accès aux soins de santé pour les résidents du Québec. La page présente le service "Rendez-vous santé Québec" comme un système harmonisé et convivial qui permet la prise de rendez-vous en ligne avec des professionnels de la santé.

L'interface utilisateur épurée met en exergue un bouton important et bien visible : "Prendre un rendez-vous", qui suggère que l'action la plus importante pour l'utilisateur est de planifier facilement

une rencontre avec un professionnel de santé. Cette fonctionnalité semble être le cœur du service offert, soulignant la facilité d'accès à des services de médecine de famille.

La page décrit les trois options de prise de rendez-vous : avec son professionnel de la santé attitré, avec un autre professionnel du même groupe de médecine de famille, ou avec un professionnel de la santé d'une clinique à proximité. Cela démontre la flexibilité et l'adaptabilité du système en fonction des besoins des patients, tout en soulignant l'objectif de fournir un accès équitable et personnalisé aux soins médicaux.

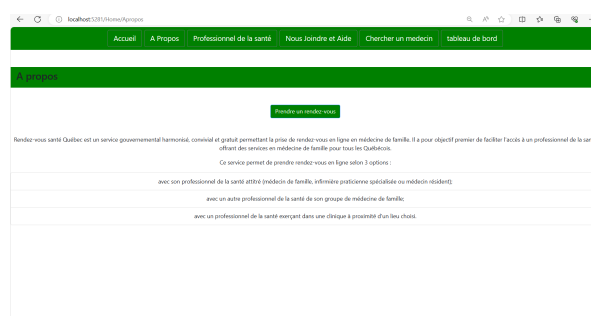


Figure 3: Aperçu de la page A propos

5.3 Professionnel de la santé

La page "Professionnel de la santé" de la Clinique Médicale sert de portail pour les professionnels de santé qui utilisent la plate-forme. En haut de la page, les onglets de navigation restent cohérents avec le reste du site, assurant une expérience utilisateur uniforme.

Un bouton "Connexion" est mis en évidence, indiquant que les professionnels de santé peuvent accéder à leurs comptes pour gérer les rendez-vous.

La page se termine par une liste des circonstances sous lesquelles un patient peut être réorienté, soulignant la flexibilité et la réponse rapide du système dans différentes situations d'urgence.

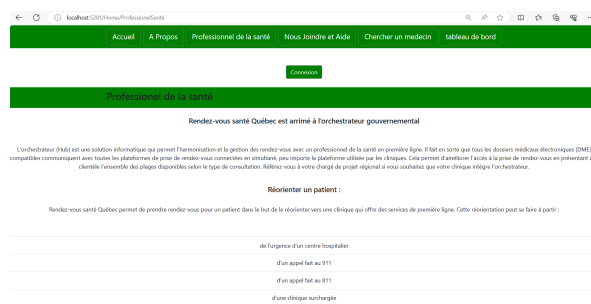


Figure 4: Aperçu de la page professionnel de santé

5.4 Nous joindre et aides

La page "Nous joindre et aide" de l'application web est une ressource pour les utilisateurs qui cherchent de l'aide ou qui souhaitent contacter la clinique. Elle est conçue pour fournir une assistance et des informations de contact aux utilisateurs du site de la Clinique Médicale. Avec une navigation simple et directe en haut de la page, les utilisateurs peuvent facilement naviguer entre les différentes sections du site, y compris l'accueil, à propos, et le profil professionnel de la santé.

Au centre de la page, il y a deux onglets : "Citoyen" et "Professionnel de la santé", indiquant que le site propose des informations et des services d'assistance personnalisés selon que l'utilisateur est un patient (citoyen) ou un fournisseur de soins de santé. Cette distinction aide à orienter les utilisateurs vers les ressources les plus pertinentes pour leurs besoins spécifiques.

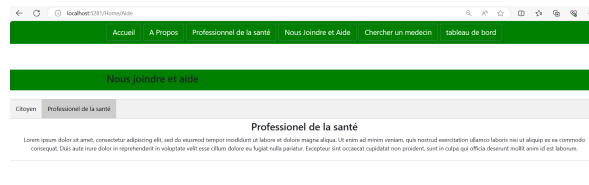


Figure 5: Aperçu de la page Nous joindre et aide

5.5 Chercher un médecin

La page "Chercher un médecin" offre une interface simple et efficace pour la recherche de professionnels de santé au sein de la Clinique Médicale. Avec son design épuré et son schéma de couleurs cohérent avec le reste du site, cette page permet aux utilisateurs de trouver rapidement et facilement un médecin par son prénom, son nom ou sa spécialité.

Au centre de la page, trois champs de saisie sont clairement indiqués pour le prénom ("FirstName"), le nom ("LastName") et la spécialité ("Spécialité"), permettant une recherche personnalisée et ciblée. Le bouton "Chercher" en vert vif attire l'attention et semble inviter à une action immédiate, suggérant une réponse rapide et des résultats pertinents à la suite de la recherche.

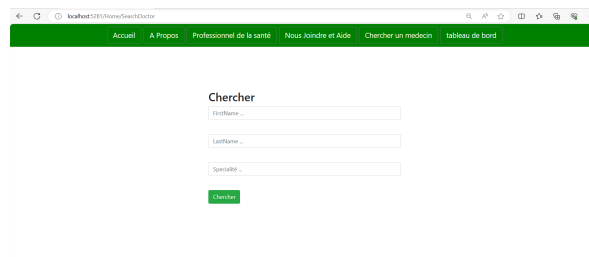


Figure 6: Aperçu de la page Chercher un médecin

5.6 Page de Connexion

La fenêtre "Connexion" est un composant essentiel de l'interface utilisateur de votre application web, permettant aux utilisateurs de s'identifier pour accéder à leur espace personnel. Le formulaire est simple et direct, demandant le "Nom d'utilisateur" et le "Mot de passe" dans deux champs distincts.

Les utilisateurs sont invités à saisir leurs informations d'identification, avec des indications claires en gris dans chaque champ pour guider la saisie.

En dessous des champs de saisie, deux boutons offrent des options pour soit "Se connecter" pour les utilisateurs existants, soit "Créer un compte" pour les nouveaux visiteurs. Le bouton "Se connecter" est probablement celui qui mène à la vérification des identifiants et à l'accès au compte, tandis que le bouton "Créer un compte" invite les nouveaux utilisateurs à s'inscrire sur le site.

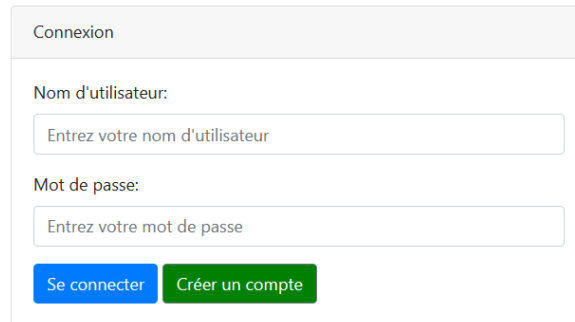


Figure 7: Aperçu de la page de Connexion

5.7 Créer un compte et prendre un Rendez-vous

Les deux sections "Créer un Compte" et "Prendre un Rendez-vous" sont conçues pour recueillir des informations essentielles des utilisateurs qui souhaitent s'inscrire et prendre un rendez-vous sur le site de la Clinique Médicale. Les formulaires sont intuitifs et invitent l'utilisateur à entrer son prénom et son nom, avec un message "Champs requis !" qui souligne l'importance de ces informations pour la création du compte.

Le champ "Sexe" propose une liste déroulante, probablement offrant plusieurs options pour que l'utilisateur puisse identifier son genre. La "Ville" est aussi une liste déroulante, suggérant que le site peut fournir des informations personnalisées basées sur la localisation géographique de l'utilisateur.

La "Date de naissance" est demandée avec un placeholder au format "aaaa-mm-DD" pour assurer que l'utilisateur entre sa date de naissance dans le format approprié, et le champ "Email" est prévu pour que l'utilisateur puisse fournir son adresse électronique, probablement utilisée pour la communication et la récupération du compte.

En bas des formulaires, des boutons sont présents : "Créer" pour soumettre le formulaire et "Retour" pour annuler l'inscription et revenir à la page précédente et "Confirmer" pour la prise des rendez-vous.

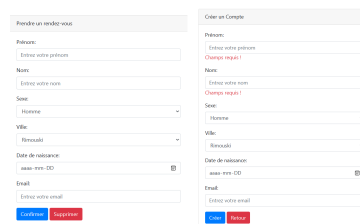


Figure 8: Aperçu des pages du création du compte et de rendez-vous

6 Backend

6.1 Utilisation ASP.Net Core MVC

Pendant la création de l'application web on va ajouter et échafauder un modèle, travailler sur une base de données et ajouter la recherche a la validation. Le modèle d'architecture MVC (Model-View-Controller) sépare une application en trois composants principaux : Model, View et Controller. Le modèle MVC vous aide à créer des applications plus testables et plus faciles à mettre à jour que les applications monolithiques traditionnelles.

6.1.1 Ajouter un Controllers

Le controller contient des classes qui permettent de gérer les requêtes du navigateur, de récupérer les modèles de données.

6.1.2 Ajouter un models

6.2 Persistance des données avec Entity Framework (ORM).

6.2.1 Introduction et Contexte

Dans le développement de logiciels, la gestion des données est un aspect crucial. Ces données doivent être gérées de manière efficace et être facilement accessibles pour assurer le bon fonctionnement du programme ou de l'application. Habituellement, les données sont stockées en mémoire volatile. Cela signifie que lorsqu'un programme ou une application est fermé, toutes les données stockées dans cette mémoire sont perdues. La question importante est donc de savoir comment conserver ces données de manière permanente.

La solution à ce problème est la persistance des données. La persistance des données fait référence à la conservation des données même après la fermeture du programme ou de l'application. Elle peut prendre différentes formes, telles que la sauvegarde des données dans des fichiers, des bases de données ou d'autres systèmes de stockage persistants.

Cette approche permet de garantir que les données importantes ne sont pas perdues et peuvent être récupérées ou réutilisées chaque fois que le programme ou l'application est relancé. La persistance des données est essentielle pour de nombreuses applications, notamment celles qui nécessitent le stockage de grandes quantités de données, comme les systèmes de gestion de bases de données, les applications web, etc.

6.2.2 Persistance des données

c'est l'ensemble des mécanismes mis en œuvre afin de permettre la sauvegarde des données de façon permanente. Son objectif est de Conserver les données même après fermeture de l'application ou du programme. Il est connu avec ces différents formes car il est possible d'utiliser de simples fichiers sans structure, mais Ça engendre des problèmes et de difficultés d'accès et d'utilisation des données. Il est également possible d'utiliser des fichiers structurés ou formatés comme par exemple : JSON, XML, CSV, ...

Il y a amélioration, mais il y a aussi des limitations liées à la quantité de données:

- Il est nécessaire de lire tout le fichier afin de récupérer une certaine donnée,
- Il n'est pas possible de lancer des requêtes personnalisées,

- La gestion et l'accès aux données n'est ni sécurisé ni optimisé,
- Risque accrue d'erreur et de perte de données,
- Temps de réponse important dans le cas de grande masse de données,
- ...

6.2.3 SGBD Ou DBMS

1. **Définition :** Un SGBD ou DBMS est un logiciel ou un ensemble de programmes conçu pour stocker, manipuler et gérer des données. Il sert d'interface entre les données et les utilisateurs ou les applications, permettant un accès et une manipulation efficaces des données.
2. **Types de SGBD :** Les SGBD peuvent être classés en deux grandes catégories en fonction du langage de requête qu'ils utilisent :
 - (a) **SQL (Structured Query Language) :** Il s'agit d'un langage de requête structurée, utilisé pour gérer et manipuler des données organisées sous forme de tables. Les SGBD SQL sont basés sur le modèle relationnel, où les données sont structurées en lignes et colonnes. Ils sont idéaux pour les situations où l'intégrité et la consistance des données sont primordiales. Les exemples incluent MySQL, PostgreSQL, et Microsoft SQL Server.
 - (b) **NoSQL (No Structured Query Language):** Ces SGBD ne suivent pas le modèle relationnel et ne nécessitent pas forcément l'utilisation de SQL pour la manipulation des données. Les SGBD NoSQL sont conçus pour gérer de grandes quantités de données distribuées (Big Data) et sont plus flexibles en termes de schémas de données. Ils prennent en charge divers modèles de données, comme les documents, les graphes, les clés-valeurs, et les colonnes larges. Des exemples de SGBD NoSQL incluent MongoDB, Cassandra, et Redis.
3. **Comparaison entre SQL et NoSQL :**
 - (a) **SQL:** est plus ancien et est largement utilisé dans les applications traditionnelles. Il est hautement structuré, ce qui facilite la garantie de l'intégrité des données et des transactions.
 - (b) **NoSQL :** est relativement plus récent et offre plus de flexibilité pour gérer différentes formes de données, en particulier pour les applications traitant de Big Data, où les schémas de données peuvent évoluer rapidement.

Chacun de ces systèmes a ses propres avantages et est mieux adapté à certains types d'applications. Le choix entre SQL et NoSQL dépend des besoins spécifiques en matière de gestion des données, de l'évolutivité requise, de la structure des données et d'autres facteurs liés à l'application en question.

6.2.4 Bases de données relationnelles

Ce modèle de base de données est l'un des plus utilisés dans le domaine de la gestion des données. Il a plusieurs caractéristiques comme :

1. **Représentation sous forme de tableaux à deux dimensions :** Dans une base de données relationnelle, les données sont organisées en tables, également appelées relations. Chaque table est structurée comme un tableau à deux dimensions.

2. **Colonnes (Champs ou Attributs) :** Les colonnes d'une table représentent les différents champs ou attributs. Chaque colonne a un type de données spécifique et décrit une caractéristique particulière des données stockées dans la table.
3. **Lignes (Données ou Enregistrements) :** Les lignes dans une table représentent les enregistrements ou les données individuelles. Chaque ligne est un enregistrement distinct et comprend des valeurs pour chaque colonne/attribut de la table.
4. **Indépendance des Lignes :** Les lignes dans une base de données relationnelle sont indépendantes les unes des autres. Cela signifie que chaque ligne (ou enregistrement) est distincte et peut être manipulée indépendamment des autres.
5. **Introduction par Edgar F. Codd :** Le modèle de base de données relationnelle a été introduit par Edgar F. Codd en 1970. Cette introduction a marqué un tournant dans la façon dont les données étaient organisées et manipulées dans les systèmes informatiques.
6. **Utilisation du Langage SQL :** Le SQL (Structured Query Language) est le langage standard utilisé pour interroger et manipuler les bases de données relationnelles. Il permet de créer, modifier, interroger et gérer les données dans les tables.
7. **Création de Liaisons, Associations ou Jointures :** Une caractéristique clé des bases de données relationnelles est la capacité de créer des relations entre les tables. Cela est souvent réalisé grâce à des jointures, qui permettent de combiner des données de différentes tables en fonction de conditions relationnelles. Ces jointures facilitent la manipulation des données de manière significative, permettant des requêtes complexes et des analyses de données interconnectées.

Le modèle relationnel est largement répandu et utilisé dans de nombreux systèmes et applications, allant des systèmes de gestion d'entreprise aux applications web, en raison de sa robustesse, de sa facilité d'utilisation et de sa capacité à maintenir l'intégrité des données.

6.2.5 Qu'est-ce qu'un ORM ?

L'Object-Relational Mapping (ORM) est un programme informatique conçu pour créer une abstraction ou une interface entre une application et sa base de données. L'ORM joue un rôle crucial en rendant la manipulation des données plus facile, moins sujette aux erreurs, et plus efficace. Les principales caractéristiques de l'ORM sont :

1. **Abstraction de la Base de Données :** L'ORM permet aux développeurs de travailler avec des objets de base de données comme s'ils faisaient partie du langage de programmation utilisé, au lieu d'écrire du code SQL brut.
2. **Indépendance du Langage de Programmation :** Différents langages de programmation ont leurs propres ORM, par exemple, Python utilise SQLAlchemy, Java utilise Hibernate, et C utilise Entity Framework. Cela signifie que les développeurs peuvent utiliser le langage avec lequel ils sont le plus à l'aise tout en bénéficiant des avantages de l'ORM.
3. **Amélioration de la Productivité :** En automatisant les tâches répétitives liées à la gestion des bases de données, les ORM aident à accélérer le processus de développement.
4. **Optimisation des Performances :** Les ORM peuvent optimiser les requêtes et améliorer les performances grâce à des techniques comme le caching et le lazy loading.

5. **Gestion des Erreurs et Sécurité** : Les ORM aident à gérer les erreurs de manière plus efficace et à améliorer la sécurité en évitant les injections SQL, une faille de sécurité courante dans les applications web.

6. **Deux Formes d'Abstraction** :

- **Partie Application** : Côté application, les données sont gérées comme des objets (base de données orientée objet).
- **Partie SGBD (Système de Gestion de Base de Données)** : Côté base de données, les données sont stockées dans un format relationnel.

7 Authentification et autorisation

7.1 Qu'est ce que l'authentification ?

L'authentification est un processus crucial permettant aux individus de démontrer leur véritable identité. Selon une étude menée en 2020 par la Digital Shadows Photon Research Team, environ 15 milliards d'identifiants volés, y compris des noms d'utilisateurs et mots de passe pour des comptes bancaires, réseaux sociaux et services de streaming, étaient accessibles sur le dark web.

Dans notre vie quotidienne, nous sommes souvent invités à prouver notre identité. Par exemple, il peut nous être demandé de présenter un permis de conduire ou une autre pièce d'identité pour effectuer certaines activités, comme déposer un chèque, acheter de l'alcool ou entrer dans des zones réglementées. Cette vérification d'identité sert à confirmer que la personne correspond bien à celle sur la photo et que le document n'est ni volé ni falsifié. En cas de pièce d'identité volée ou fausse, l'individu peut être refusé, signalé aux autorités, et le document peut être confisqué.

7.2 Qu'est ce que l'autorisation ?

Les entreprises mettent en place des systèmes d'autorisation pour contrôler l'accès aux ressources. Ces autorisations sont définies ou personnalisées en fonction du rôle de l'utilisateur, de ses attributs d'identité ou de certains critères de risque. Après l'authentification de l'utilisateur, l'autorisation détermine quelles données, applications et autres ressources il est autorisé à utiliser. De plus, ce système empêche les utilisateurs non autorisés d'accéder à des éléments pour lesquels ils n'ont pas les droits nécessaires, y compris les applications locales et celles hébergées dans le cloud.

7.3 Authentification et autorisation

7.3.1 Authentication

pour accéder à votre compte, vous devez prouver votre identité avec au moins un facteur d'authentification.

7.3.2 Authorization

une fois que vous êtes authentifié, l'entreprise vous accorde certaines autorisations d'accès en tant que client. Vous pouvez accéder à votre compte, consulter la description des produits, utiliser les fonctionnalités du panier d'achats, etc. Vous n'êtes pas autorisé à utiliser des ressources ou des fonctionnalités strictement réservées aux employés et qui nécessitent un certain niveau d'autorisation de sécurité ou une autorisation spéciale.

8 Utilisation de Dapr. Net

Les développeurs en entreprise se concentrent sur la logique métier, s'appuyant sur des plateformes pour intégrer échelle, résilience et autres caractéristiques des architectures cloud-native. Cela a conduit à un intérêt croissant pour les plateformes sans serveur qui masquent l'infrastructure sous-jacente. Il est souligné que les développeurs ne devraient pas nécessairement devenir experts en systèmes distribués.

Dapr, une plateforme de programmation de microservices axée sur les développeurs et les entreprises, soutient cette évolution. Avec son approche "tout langage, tout cadre, fonctionne partout", Dapr facilite la création d'applications distribuées sur diverses infrastructures, y compris le cloud public et les dispositifs IoT. Inspiré par les expériences d'Azure et les besoins des clients, Dapr offre une "bibliothèque" de meilleures pratiques pour les microservices, utilisable dans les nouvelles applications et la modernisation des existantes. Il utilise un modèle de sidecar pour offrir des fonctionnalités de systèmes distribués facilement accessibles aux développeurs.

Dapr propose également des fonctionnalités comme la gestion d'état, l'invocation de service à service, la publication/souscription, et l'intégration avec des systèmes externes, en tirant parti de son modèle de composant. Ce modèle facilite la portabilité et la flexibilité du code, et permet aux développeurs d'éviter d'intégrer des SDK de services spécifiques ou de gérer des éléments comme l'authentification ou la gestion des secrets.

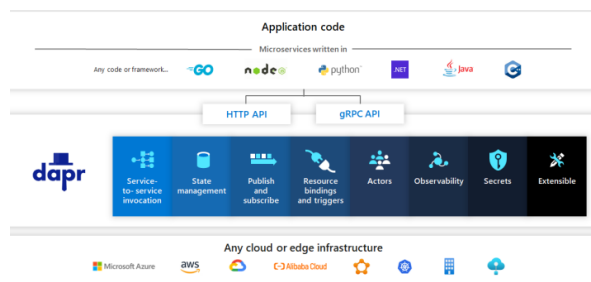


Figure 9: Dapr et le problème qu'il résout

8.1 Dapr HTTP et Dapr gRPC

Les API Dapr permettent à toute pile d'applications prenant en charge HTTP ou gRPC d'utiliser les blocs de construction Dapr. Par conséquent, le module d'invocation de service peut servir de pont entre les protocoles. Les services peuvent communiquer entre eux en utilisant HTTP, gRPC ou une combinaison des deux. Le SDK .NET de Dapr offre aux développeurs .NET un moyen intuitif et spécifique au langage d'interagir avec Dapr.

8.2 Utilisation de .Net Dapr SDK

Le .Net Dapr SDK offre aux développeurs trois façons d'effectuer des appels d'invocation de services à distance :

1. **Invoquer des services HTTP à l'aide de HttpClient** : L'utilisation de la classe HttpClient pour l'invocation de services Dapr présente de nombreux avantages :

- HttpClient est une classe bien connue que de nombreux développeurs utilisent déjà dans leur code. L'utilisation de HttpClient pour l'invocation de services Dapr permet aux développeurs de réutiliser leurs compétences existantes.
 - HttpClient prend en charge des scénarios avancés, tels que les en-têtes personnalisés et le contrôle total des messages de demande et de réponse et les messages de réponse.
 - Dans .NET 5, HttpClient prend en charge la sérialisation et la désérialisation automatiques à l'aide de System.Text.Json.
2. **Invocation de services HTTP à l'aide de DaprClient :** L'intégration de HttpClient avec Dapr pour les appels d'invocation de service offre plusieurs avantages, en particulier dans des environnements où Dapr est utilisé pour faciliter le développement de microservices. Elle est connue aussi par leur plusieurs avantages comme il est largement utilisé et bien intégré dans de nombreux frameworks et bibliothèques, en particulier dans l'écosystème .NET. L'utilisation de 'HttpClient' avec Dapr permet de bénéficier de cette intégration facile et familière. L'utilisation de 'HttpClient' avec Dapr permet de profiter de ses optimisations en termes de gestion des connexions réseau et de performance.
3. **Invoquer des services gRPC à l'aide de DaprClient :** Pour l'utilisation de "DaprClient" pour invoquer des méthodes gRPC, nous devons spécifier l'ID de l'application Dapr du service cible, le nom de la méthode et toutes les données de la requête. DaprClient abstrait les appels gRPC réels, Nous permettant d'invoquer des méthodes comme si vous faisiez des appels de fonction locaux.

9 automatisation des tests fonctionnels

9.1 Selenium Web Driver

9.1.1 Definition:

Selenium WebDriver est un outil d'automatisation de navigateur web très populaire. Il permet de piloter un navigateur de manière native, comme le ferait un utilisateur, que ce soit en local ou sur une machine distante à travers le serveur Selenium. Cette capacité marque une évolution significative dans le domaine de l'automatisation des navigateurs.

Selenium est connu avec ces points importants :

- **Contrôle Natif du Navigateur:** WebDriver interagit avec les navigateurs de manière native, offrant un contrôle plus précis et des interactions plus naturelles comparées à d'autres approches d'automatisation.
- **Support Multi-langue:** Il offre des liaisons pour plusieurs langages de programmation, ce qui permet aux développeurs d'utiliser le langage avec lequel ils sont le plus à l'aise.
- **Code de Contrôle Individuel du Navigateur:** Chaque navigateur a sa propre implémentation du code de contrôle, ce qui assure une compatibilité et une efficacité optimales lors de l'exécution des scripts sur différents navigateurs.
- **WebDriver comme Terme Commun:** Dans le domaine de l'automatisation des navigateurs, le terme "WebDriver" est couramment utilisé pour désigner l'ensemble de ces fonctionnalités et capacités.

9.1.2 Option de navigateur

Les fonctionnalités du Selenium WebDriver sont partagées pour tous les navigateurs. Chaque navigateur dispose d'options personnalisées qui peuvent être définies en plus de celles définies dans la spécification

9.1.3 Pilote Web

Selenium supporte l'automatisation de tous les principaux navigateurs grâce à WebDriver, une API et un protocole qui offre une interface de contrôle des navigateurs web indépendante du langage de programmation. Chaque navigateur est pris en charge par un pilote WebDriver spécifique, responsable de la communication entre Selenium et le navigateur.

Cette architecture reflète une stratégie délibérée visant à impliquer les fournisseurs de navigateurs dans le développement de leurs propres implémentations. Bien que Selenium privilégie l'utilisation de ces pilotes tiers, le projet maintient aussi ses propres pilotes pour les situations où cela s'avère nécessaire.

Le framework Selenium intègre ces composants via une interface unifiée, facilitant l'automatisation transparente sur différents navigateurs et plateformes.

La configuration de Selenium se distingue des autres outils commerciaux. Pour débiter avec Selenium, il faut installer les bibliothèques de liaisons pour votre langage de programmation, le navigateur cible et son pilote correspondant.

References

1. <https://kinsta.com/fr/blog/architecture-application-web/>.
2. <https://www.pingidentity.com/fr/resources/blog/post/authentication-vs-authorization.html> Authentification-et-autorisation
3. <https://www.conceptatech.com/blog/difference-front-end-back-end-development>: :text=The
4. https://www.selenium.dev/documentation/webdriver/getting_started/https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-8.0&tabs=visual-studio
5. <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-controller?view=aspnetcore-8.0&tabs=visual-studio>
6. <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-view?view=aspnetcore-8.0&tabs=visual-studio>
7. <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-view?view=aspnetcore-8.0&tabs=visual-studio>
8. <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-model?view=aspnetcore-8.0&tabs=visual-studio>
9. <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/working-with-sql?view=aspnetcore-8.0&tabs=visual-studio>
10. <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/controller-methods-views?view=aspnetcore-8.0>
11. <https://learn.microsoft.com/en-us/azure/container-apps/dapr-overview>
12. <https://dapr.io/>