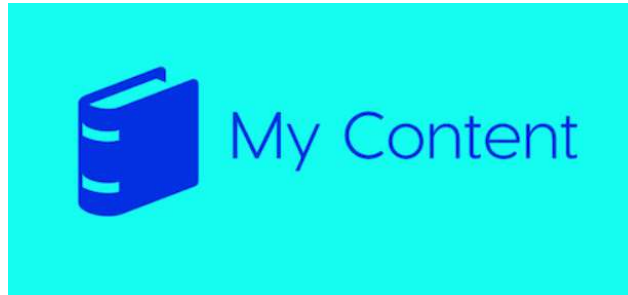


Projet 9 – My Content / Bookshelf

Sélection de l'architecture logicielle



Objectif métier :

- tester une solution de recommandation d'articles et de livres à des particuliers
- dans le but de mettre en place un premier MVP qui prendra la forme d'une application mobile
- la fonctionnalité à mettre en place pour lancer l'application est la suivante : *"En tant qu'utilisateur de l'application, je vais recevoir une sélection de cinq articles."*

Identification des différentes briques d'architecture nécessaires pour la réalisation de la solution répondant au besoin métier :

- Brique '**Application mobile**' :
 - Application Android codée en JavaScript avec le Framework ReactNative
 - Backend de l'application mobile : Node.js
- Brique '**Sélection du système de recommandation**' :
 - Scripts Python
- Brique '**Implémentation du système de recommandation**' :
 - Fonctionnalité Azure Function Serverless
- Brique '**Sauvegarde des données**' :
 - Pour la solution actuelle : fichiers CSV
 - Pour la solution cible : bases de données Utilisateurs et Articles

Description fonctionnelle de chaque brique d'architecture :

- La brique '**Application mobile**' à pour rôle d'effectuer les opérations suivantes :
 - ✓ Lorsque l'utilisateur se connecte à l'application Android, elle récupère son identifiant.
 - ✓ Elle envoie l'identifiant à l'API exposée par le service **Azure Function Serverless** qui lui renvoie le résultat, c'est-à-dire les recommandations pour l'utilisateur. Les échanges se font par des requêtes HTTP.
 - ✓ Elle affiche le résultat des recommandations à l'utilisateur.
- Cette brique peut être considérée comme la partie **FrontEnd** de l'application puisqu'elle interagit avec l'utilisateur.

- La brique '**Sélection du système de recommandation**' a pour rôle de préparer les données et sélectionner le système de recommandation qui sera déployé. Pour cela elle effectue les opérations suivantes :
 - ✓ Analyse et préparation des données
 - ✓ Modélisation, évaluation et sélection du meilleur modèle qui sera utilisé en production
 - ✓ Préparation des données pour le déploiement
- Cette brique fait partie du **BackEnd** de l'application.
- Elle permet d'exécuter la chaîne de traitements IA de bout-en-bout : analyse et préparation des données, puis modélisation et évaluation et enfin préparation au déploiement.
- Afin de permettre l'exécution de la chaîne de traitements IA de bout-en-bout, les scripts ont été déposés dans un dossier GitHub.

- La brique '**Implémentation du système de recommandation**' a pour rôle d'implémenter le système de recommandation.
 - Cette brique est implémentée en utilisant la fonctionnalité **Azure Function Serverless**. Celle-ci permet d'exécuter 'à la demande' une fonction hébergée dans Azure.
 - Les avantages sont les suivants :
 - ✓ L'utilisateur n'a pas besoin de s'occuper de la partie serveur, c'est Azure que s'en charge.
 - ✓ L'utilisateur paie uniquement quand la fonction est lancée.
 - Cette brique expose une **API** qui permet de communiquer avec la brique '**Application mobile**' grâce à des **requêtes HTTP**.
 - La fonction hébergée dans Azure est exécutée à la réception d'un **Trigger** qui est une requête HTTP venant de la brique '**Application mobile**' dans notre cas.
 - La fonction récupère les données nécessaires grâce à une fonctionnalité pré implémentée appelée **Input Binding**.
 - De façon symétrique, la fonction envoie la réponse grâce à une fonctionnalité pré implémentée appelée **Output Binding**.

Cette brique effectue les opérations suivantes :

- ✓ Elle récupère l'identifiant utilisateur envoyé par la brique '**Application mobile**'.
- ✓ Elle récupère les données nécessaires soit dans un dataset au format CSV enregistré dans un blob dans un compartiment Datastore Azure Storage (solution actuelle), soit en bases de données (solution cible).
- ✓ Elle fait tourner le modèle de recommandation (implémenté en langage Python).

- ✓ Elle renvoie la réponse à la brique '**Application mobile**'.
- Cette brique a un rôle central dans l'architecture logicielle mise en place, puisqu'elle fait le lien entre les briques '**Application mobile**' et '**Sélection du système de recommandation**' et elle récupère les données auprès de la brique '**Sauvegarde des données**'.
- La brique '**Sauvegarde des données**' sert à sauvegarder les données nécessaires au bon fonctionnement du projet. Celle-ci est mise en place en 2 étapes :
 - Etape actuelle (MVP) :
 - ✓ Les données sont enregistrées au format CSV dans un blob dans un compartiment Datastore Azure Storage.
 - ✓ Le lien entre la fonction Azure Serverless et le Datastore se fait grâce à la fonctionnalité Input Binding de Azure.
 - Etape cible (production) :
 - ✓ Les données seront enregistrées dans des bases de données Utilisateurs et Articles.
 - ✓ On pourra choisir une base de données Azure CosmosDB, ce qui permettra de lier directement la base de données à la fonction Azure Serverless grâce à la fonctionnalité Input Binding de Azure.
- Cette brique fait partie du **BackEnd** de l'application.

Mise à jour des bases de données Utilisateurs et Articles en cas de nouvel utilisateur et nouvel article :

- Nouvel utilisateur :
 - ✓ Il faut enregistrer le nouvel utilisateur dans la base de données Utilisateurs.
 - ✓ Au niveau du modèle il n'y a rien à faire car le meilleur modèle déployé est de type Content Based.
- Nouvel article :
 - ✓ Il faut enregistrer le nouvel article dans la base de données Articles avec notamment la catégorie de l'article qui sert pour la recommandation.
 - ✓ Au niveau du modèle il n'y a rien à faire car le modèle déployé utilise automatiquement la catégorie de l'article.

Diagramme Architecture Logicielle actuelle (MVP) :

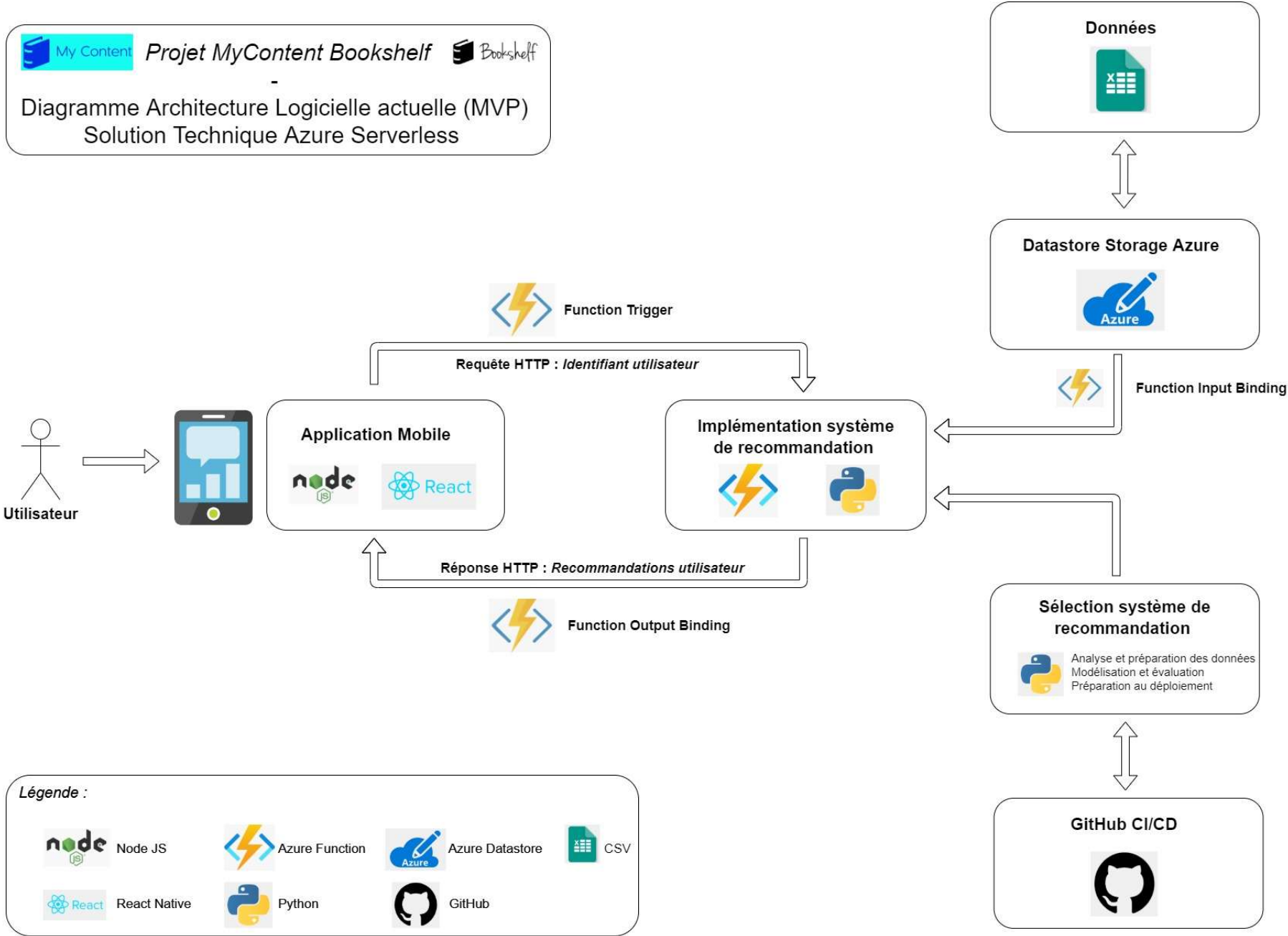


Diagramme Architecture Logicielle cible (PRODUCTION) :

 **Projet MyContent Bookshelf** 

Diagramme Architecture Logicielle cible (PROD)
Solution Technique Azure Serverless

