



MY CONTENT : APPLICATION MOBILE DE RECOMMANDATION DE CONTENU

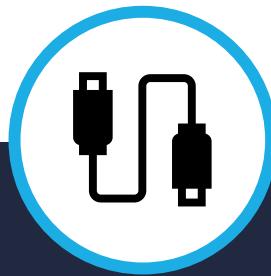
AGENDA DU JOUR



CONTEXTE PROJET
ET DONNEES



SYSTÈMES DE
RECOMMANDATION



ARCHITECTURE ET
DÉPLOIEMENT



INTÉGRATION /
DEMO



CONTEXTE PROJET ET DONNEES

MYCONTENT



Encourager la lecture



Recommander des contenus pertinents



Créer une MVP* (appli mobile) qui :
1) Identifie un utilisateur
2) Affiche les recommandations

*Minimum Viable Product



Jeux de données :
News provenant du portail d'articles
d'information de Globo.com

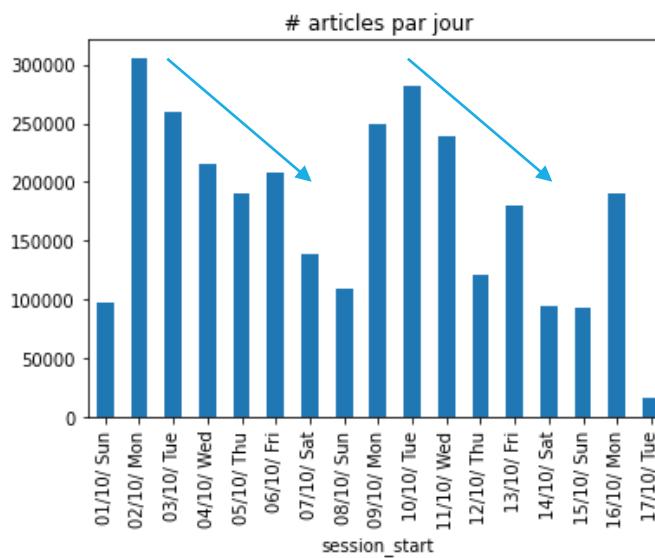


3 fichiers :
- clicks.zip
- articles_metadata.csv
- articles_embeddings.pickle

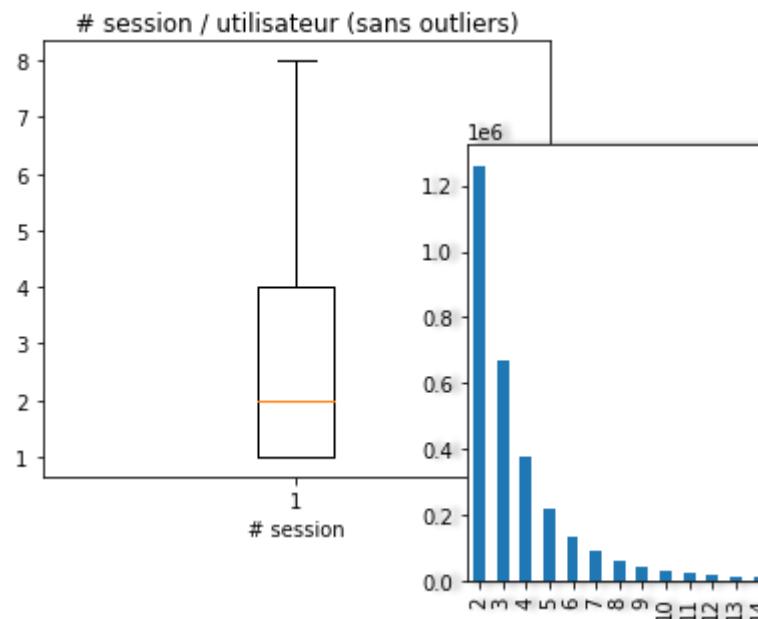


Informations essentielles :
- Nb utilisateurs uniques : ~323k
- Nb articles uniques : ~46k
- Nb interactions (clicks) : ~3 millions

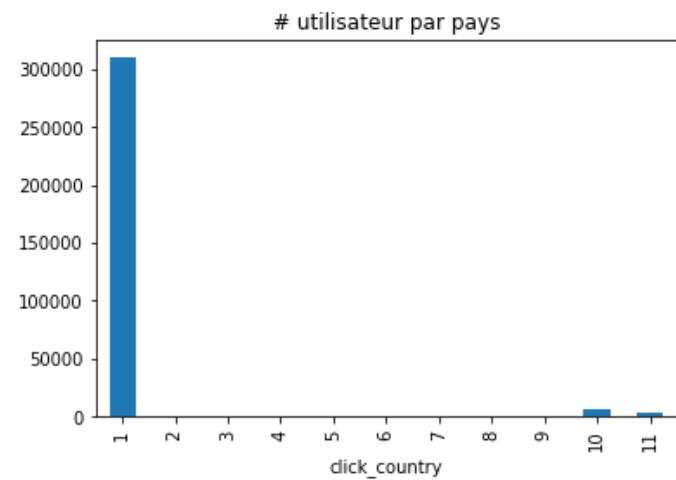
EXPLORATION DES DONNÉES : LES UTILISATEURS



Les utilisateurs consultent les articles en début de semaine plutôt qu'en fin de semaine

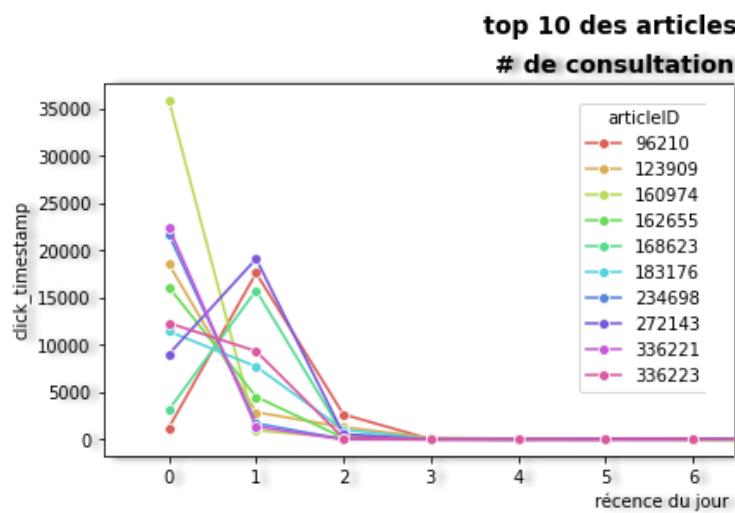


Sur les 15 jours d'observations, les utilisateurs ont eu en moyenne 3 sessions : la majorité consulte 2 articles par session

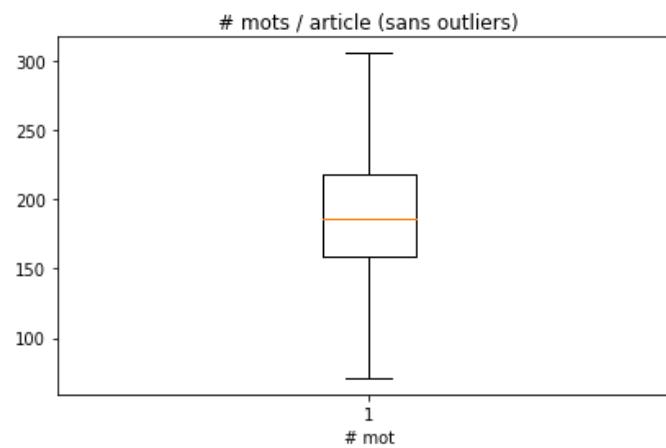


La majorité d'utilisateur sont dans le pays 1 (principalement sur les régions 25, 21 et 13)

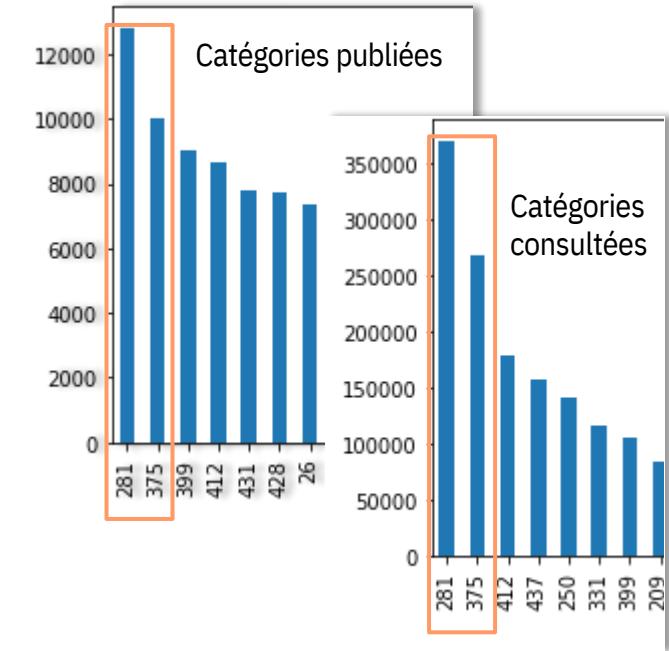
EXPLORATION DES DONNÉES : LES ARTICLES



Les articles sont consultés le jour même, ou le lendemain –l'intérêt diminuant très vite



La moitié des articles contiennent entre 159 et 218 mots, pour une moyenne à 191



Les 2 catégories les plus publiées font partie des 2 catégories les plus consultées : probablement des sujets d'actualité

EXPLORATION DES DONNÉES : LES INTERACTIONS

Présence d'historique de navigation

ID utilisateur associé à l'ID article consulté, sur toute la durée de la collecte des données

Absence de durée de consultation
Elle est systématiquement limitée à 30s pour chaque dernier article consulté sur une session

Absence de notation

Ceci nous complexifie la mesure des préférences utilisateurs : on dit ici que nous avons des [données implicites](#)

	user_id	session_id	view_duration
0	0	1506825423271737	1405.0
1	0	1506825423271737	30.0
2	1	1506825426267738	1592.0
3	1	1506825426267738	30.0
4	2	1506825435299739	1656.0

SPÉCIFICITÉS DE LA RECOMMANDATION D'ARTICLES D'INFORMATION (NEWS)

Le domaine de la recommandation d'articles d'informations accentue quelques challenges déjà existants et met en exergue ceux spécifiques au domaine.



Sparsité(\neq densité) des données d'interaction : les utilisateurs n'auront été en contact qu'avec un nombre très faible d'articles comparé au volume total de contenus disponibles



Augmentation rapide du nombre d'articles : problème de **scalabilité**(extension de capacité limitée) et de **démarrage à froid** (*cold-start*) en cas d'apparition d'un nouvel article dans la base



Dégénération accélérée de la valeur des articles (*value decay*) : l'intérêt pour une information diminue avec le temps, l'information fraîche ayant plus de valeur

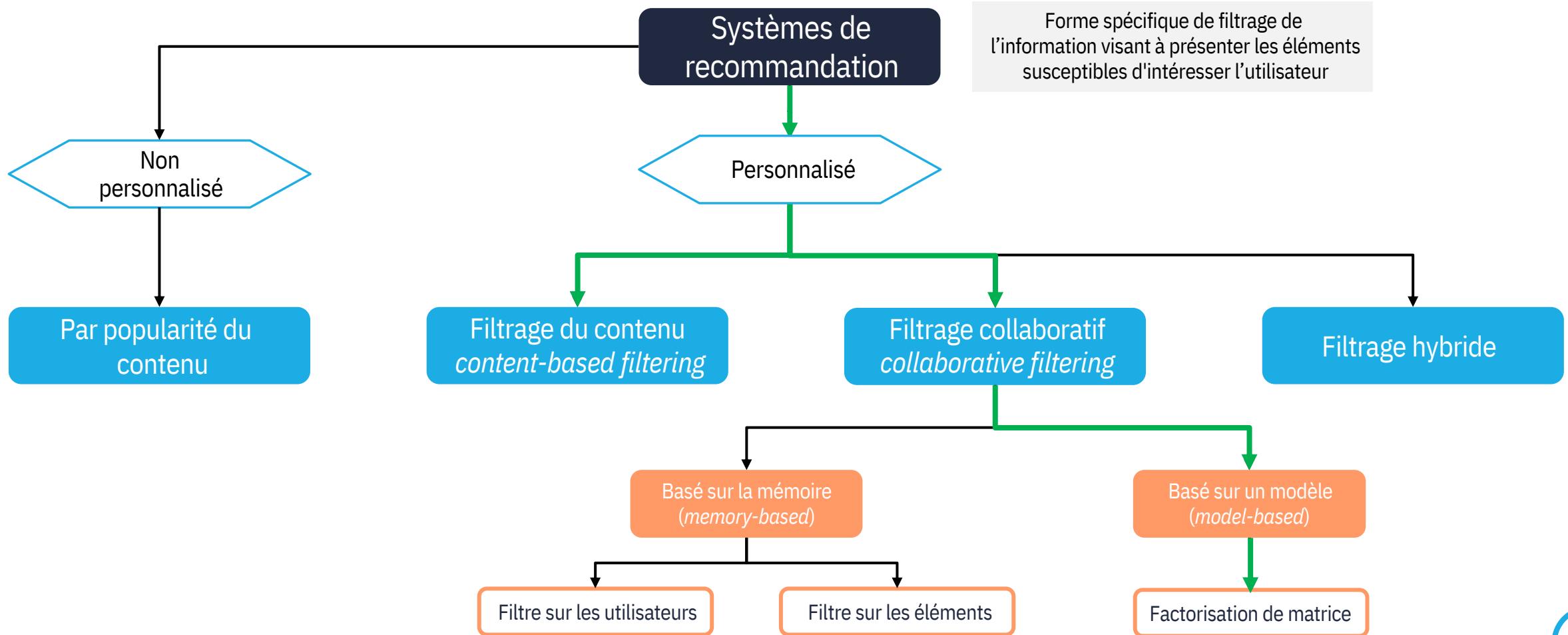


Evolution dans le temps des préférences utilisateurs : elles changent en fonction de l'actualité (*élection, évènement sportif, catastrophe naturelle, crise sanitaire, ...*)



SYSTÈMES DE RECOMMANDATION

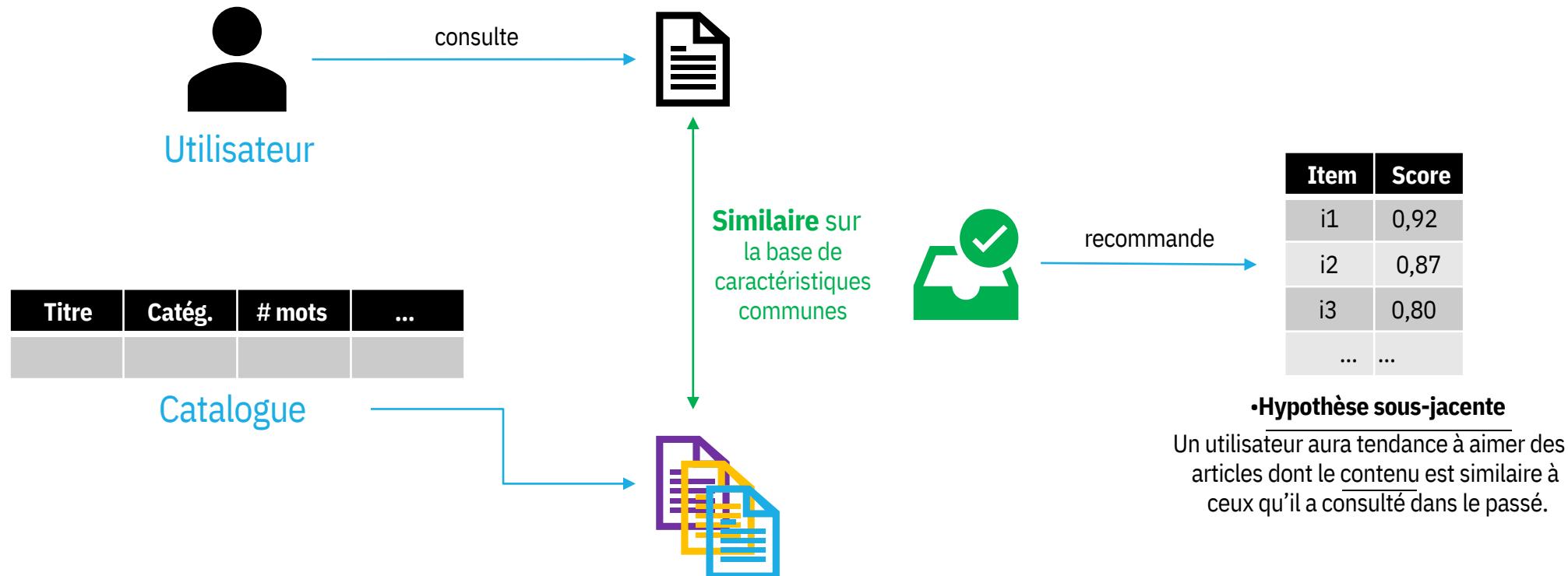
CATÉGORIES DE SYSTÈMES DE RECOMMANDATION



CATÉGORIES DE SYSTÈMES DE RECOMMANDATION

Basé sur le Contenu (content-based)

«Montrez-moi ce qui ressemble le plus à ce que j'ai aimé»



PROCESSUS DE LA RECO BASÉE SUR LE CONTENU

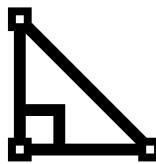


Données utilisées

- **Données de clicks**
- **Matrice Embeddings**

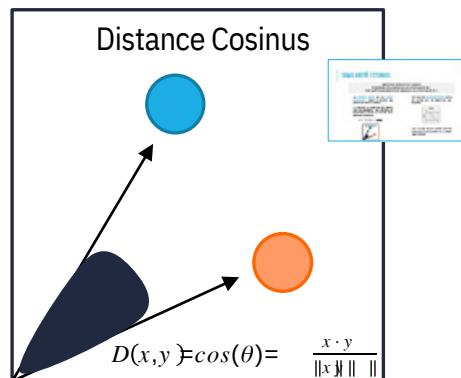
Données non utilisées

- **Catégories**
- **Date de publication**
- **Nombre de mots**



Algorithmes

- **Similarité cosinus**



Etapes

1. Choix du dernier article lu par l'utilisateur
2. Calcul de la distance cosinus entre les articles (création de la *matrice de similarité item-item*)
3. Exclusion des articles déjà lus
4. Recommandation des 5 articles les plus proches

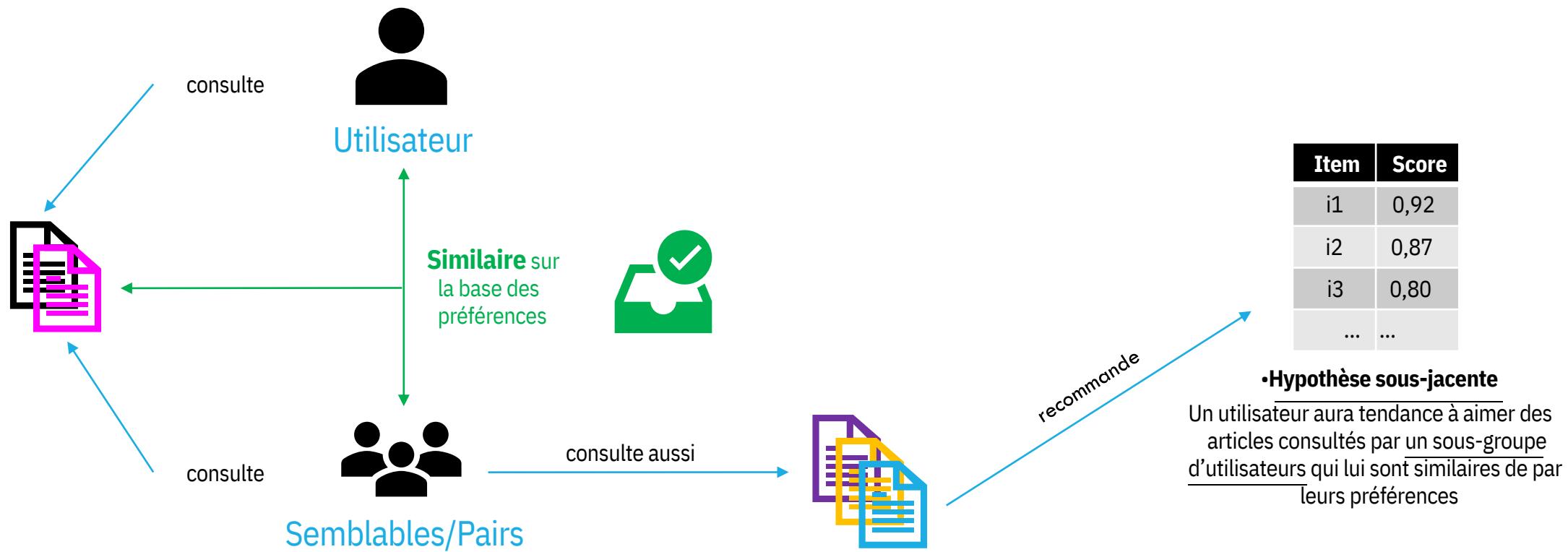
CATÉGORIES DE SYSTÈMES DE RECOMMANDATION

(3/3)

Basé sur le filtrage collaboratif (collaborative filtering)

«Les utilisateurs qui sont similaires à vous, ont AUSSI aimé ça.»

«Les utilisateurs qui ont aimé ça, ont AUSSI aimé ça.»



PROCESSUS DE LA RECO COLLABORATIVE



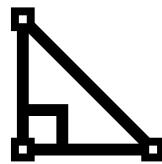
Données utilisées

Données de clicks

Module additionnel

Implicit

<https://implicit.readthedocs.io/en/latest/>



Algorithmes

Moindres carrés alternés

	Item 1	Item 2	Item 3	...	Item n
User 1	Blue		Blue		Blue
User 2		Blue		Blue	
User 3	Blue			Blue	
...		Blue	Blue		
User n		Blue		Blue	

MINIMISATION ALTERNÉE
des erreurs quadratiques

$\text{Minimiser } \|Ax - b\|_2^2$



Etapes

1. Création des matrices: item-user et user-item
2. Apprentissage du modèle sur la matrice item/user
3. Recommandation des 5 articles sur la base de la matrice user/item

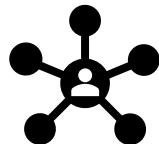
AVANTAGES / INCONVÉNIENTS DES SYSTÈMES

	Content-based	Collaborative-Filtering
Avantages	<ul style="list-style-type: none">Peu d'impact du démarrage à froid : pas besoin de données relatives aux utilisateurs: recommandation possible de nouveaux articles dès leur apparition dans la base.	<ul style="list-style-type: none">Pas besoin de métadonnées sur les articles ou les utilisateurs.
Inconvénients	<ul style="list-style-type: none">Manque de personnalisationRisque de surspécialisation (enfermement) sur des articles similaires (réponses trop homogènes).	<ul style="list-style-type: none">Problème de démarrage à froid: nouvel utilisateur sans préférence connue, ou article sans consultation.

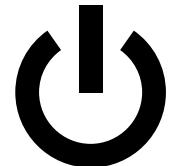
CHOIX DE NOTRE MODÈLE



Absence de notation
La factorisation
matricielle est moins
performante

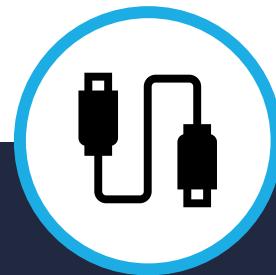


Absence de données utilisateurs élargies
La factorisation
matricielle est moins
performante



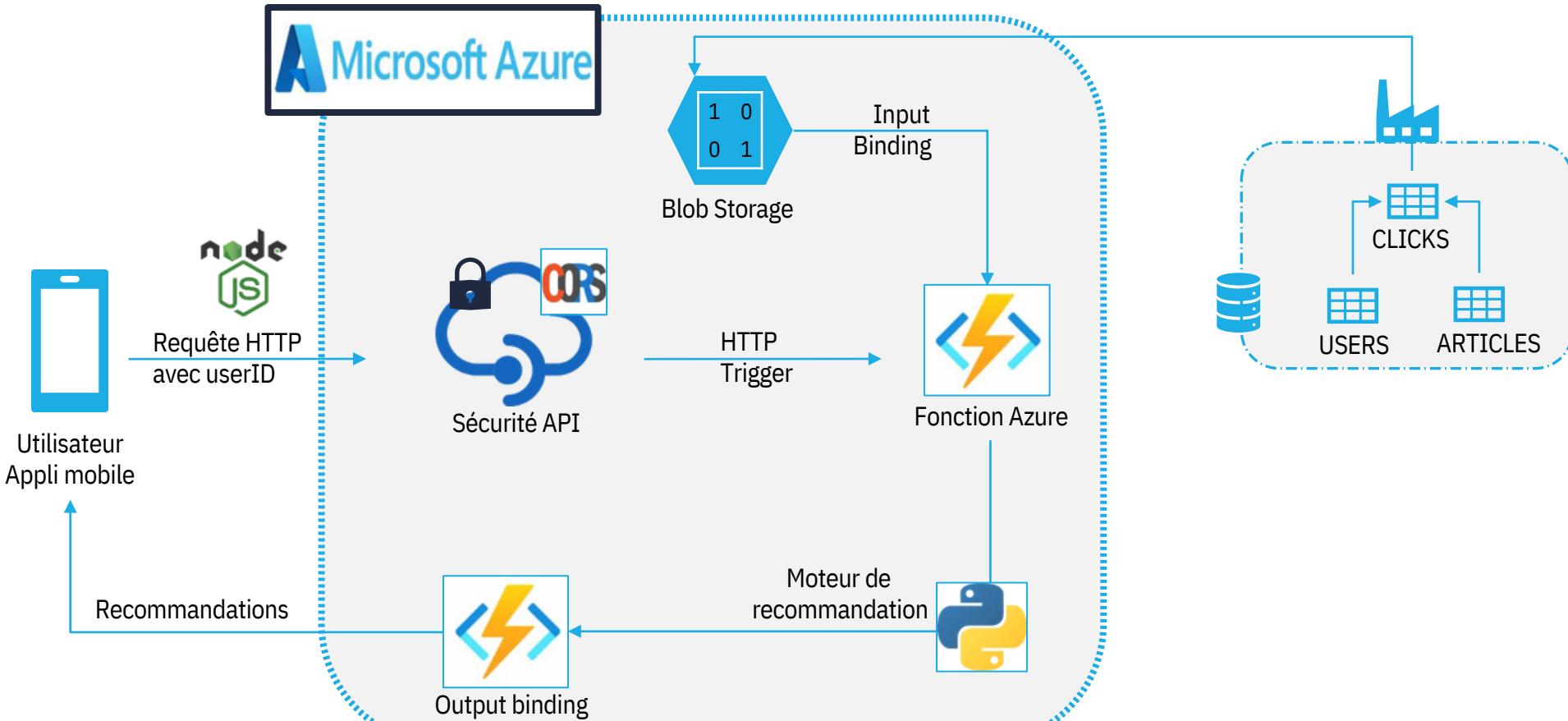
Problème de cold-start
Le filtrage collaboratif
ne peut prendre en
compte les nouveaux
articles et utilisateurs

Au regard de nos contraintes, le **filtrage basé sur le contenu** semble
être la solution la plus pertinente.



ARCHITECTURE ET DÉPLOIEMENT

ARCHITECTURE 'SERVERLESS' ACTUELLE



*CORS: Cross-Origin Resource Sharing
Partage de ressources inter-origines

DÉPLOIEMENT SUR AZURE : CRÉATION DES RESSOURCES

Les ressources suivantes sont créées, et reposent sur le nom du projet: le Groupe de ressources, le compte de Stockage Azure, le plan de service (par défaut: Consumption Plan), l'application Azure Functions, une instance Application Insights.

The screenshot shows the Microsoft Azure portal interface for managing resource groups. The main view displays the 'ourcontentreco' resource group details, including its subscription information (Microsoft Azure Sponsorship 2), location (France Central), and deployment status (No deployments). The left sidebar lists other resource groups: DefaultResourceGroup-PAR, mycontentreco, ourcontentreco (selected), and p7-demo. The right pane shows a table of resources created under this group, including an App Service plan, Application Insights instance, Storage account, and Function App, all located in France Central.

Type	Name	Location
App Service plan	ASP-ourContentReco-4fa5	France Central
Application Insights	ourcontentreco	France Central
Storage account	ourcontentreco	France Central
Function App	ourContentReco	France Central

CHARGEMENT DES FICHIERS (BLOB) DANS LE CONTAINER DU STORAGEACCOUNT

Microsoft Azure

Home > ourcontentreco > ourcontentreco

ourcontentreco | Containers Storage account

Search (Ctrl+ /) Container Change access level

Data migration

Storage Explorer (preview)

Data storage

Containers

File shares

Queues

Tables

Search containers by prefix

Name

- azure-webjobs-hosts
- azure-webjobs-secrets
- scm-releases

New container

Name *

data

Public access level

Container (anonymous read access for containers and blobs)

All container and blob data can be read by anonymous request. Clients can enumerate blobs within the container by anonymous request, but cannot enumerate containers within the storage account.

Advanced

Encryption scope

Select from existing account scopes

Use this encryption scope for all blobs in the container

Create Discard

Microsoft Azure

Home > ourcontentreco > ourcontentreco >

data Container

Search (Ctrl+ /) Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Access policy

Properties

Metadata

Authentication method: Access key (Switch t

Location: data

Search blobs by prefix (case-sensitive)

Name

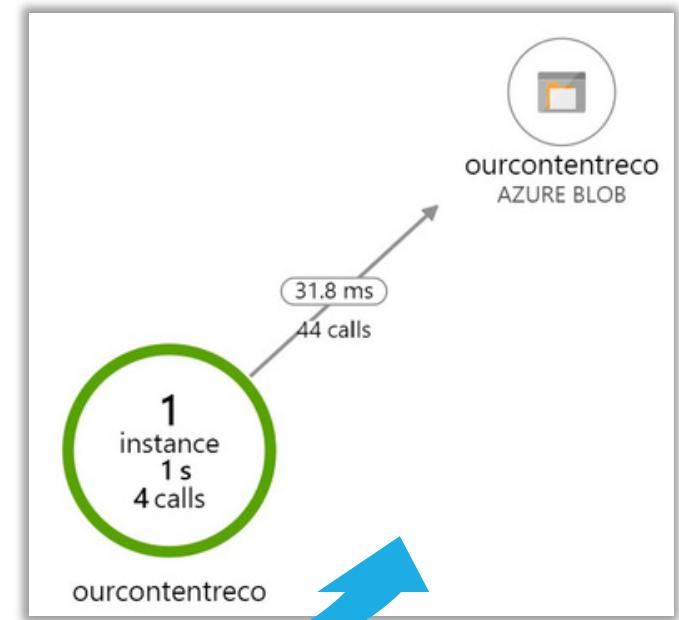
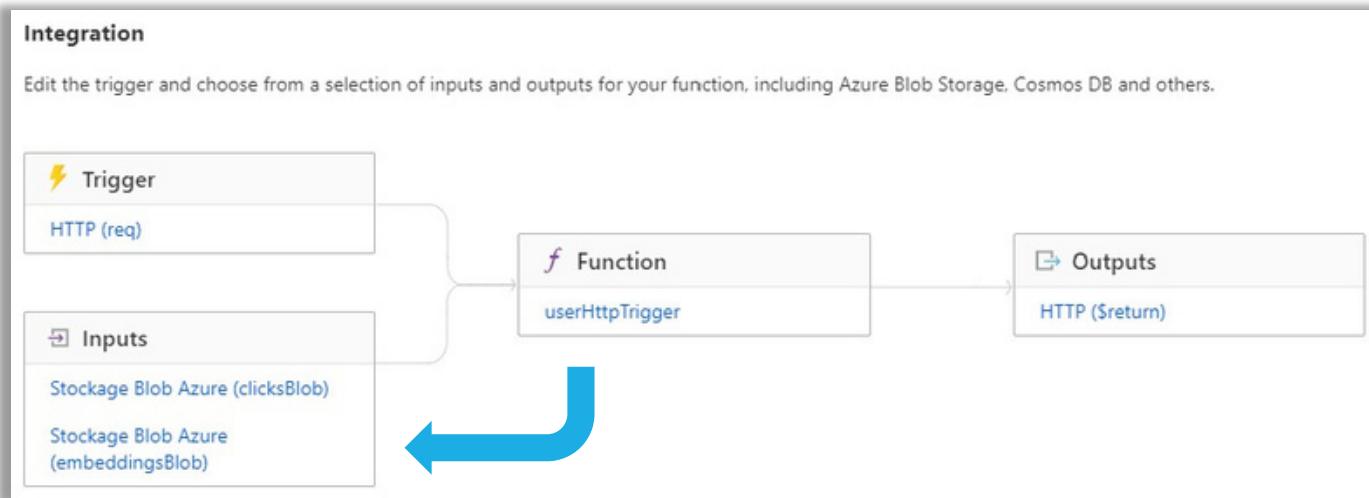
- small_clicks.csv
- small_embeddings.pickle

GESTION DES ACCÈS PARTAGE DE RESSOURCES INTRA-ORIGINES, AZURE STORAGE)

The screenshot shows the Microsoft Azure portal interface for managing CORS settings. The left sidebar lists various monitoring and development tools. The main content area is titled "ourContentReco | CORS" and contains sections for "Request Credentials" and "Allowed Origins". A note explains that CORS allows JavaScript code running in a browser on an external host to interact with your backend. It includes a checkbox for "Enable Access-Control-Allow-Credentials" and a list of allowed origins.

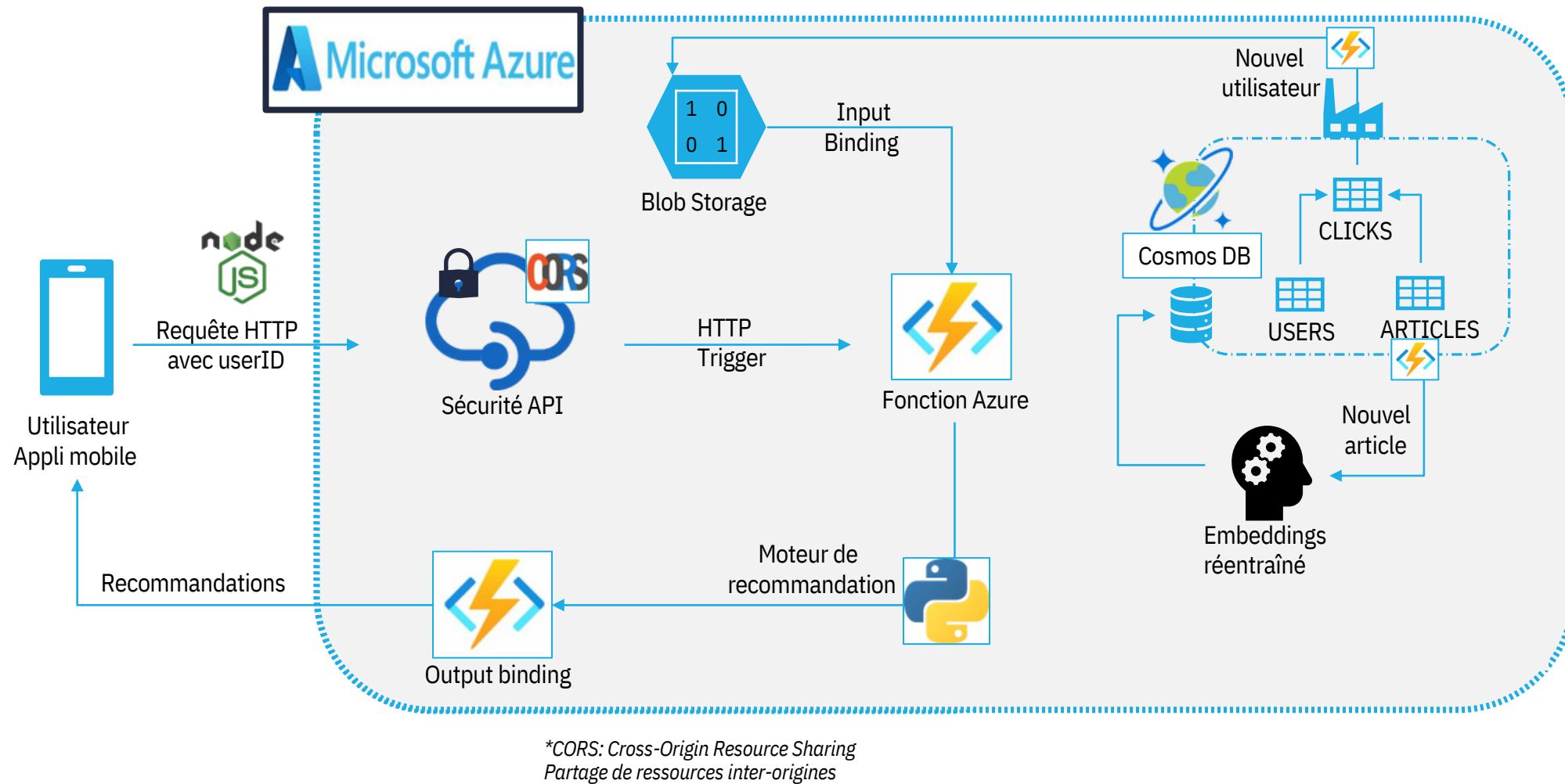
The screenshot shows the Microsoft Azure portal interface for managing access keys for a storage account named "ourcontentreco". The left sidebar lists various storage-related services. The main content area is titled "ourcontentreco | Access keys" and displays two sets of keys: "key1" and "key2". Both keys were last rotated on 21/07/2021. The "key1" section includes a "Rotate key" button and a "Connection string" input field. The "key2" section also includes a "Rotate key" button and a "Connection string" input field.

FUNCTION APP & APPLICATION INSIGHTS : VISUALISATION GRAPHIQUE DES FLUX



ARCHITECTURE 'SERVERLESS' CIBLE

[HTTPS://AZURE.MICROSOFT.COM/FR-FR/FEATURES/DEVOPS-PROJECTS/](https://azure.microsoft.com/fr-fr/features/devops-projects/)





INTÉGRATION
DEMO

TEST/RUN SUR AZURE PORTAL : USER 100

The screenshot shows the Azure Functions portal interface. On the left, the code editor displays Python code for a function named `userHttpTrigger`. The code implements a recommendation system using cosine similarity based on user click history. On the right, the test interface is open, showing the results of a POST request with a JSON body containing `{"userId": "100"}`. The response code is `200 OK` and the response content is a list of article IDs: `237452, 233478, 237429, 234128, 233716`.

```
import logging
import azure.functions as func
import pandas as pd
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
from io import BytesIO

def get_ContentBased_Reco(userID, small_clicks, small_embeddings, n_reco=5):
    """Return 5 recommended articles ID to user"""
    # Get the list of articles viewed by the user
    var = small_clicks.loc[small_clicks.user_id == userID]['article_id'].to_list()

    # Get the list of unique article_ID in small_clicks
    list_articleID = sorted(list(small_clicks.article_id.unique()))

    # Retrieve the corresponding index of the articles viewed by userID in var
    idx_var = []
    for i in range(0, len(var)):
        for idx, item in enumerate(list(list_articleID)):
            if item == var[i]:
                idx_var.append(idx)

    # Select the last element of the list
    value = idx_var[-1]
    # print(value)

    # Compute the cosine similarity
    emb = small_embeddings
    distances = cosine_similarity([emb[value]], emb)[0]
```

HTTP response code
200 OK

HTTP response content
237452, 233478, 237429, 234128, 233716

TEST SUR NAVIGATEUR & APPLI MOBILE : USER 100

The image shows three screens illustrating the user interface for User 100 across different platforms.

- Left Screen (Web Browser):** A screenshot of a web browser window titled "Bookshelf" at "localhost:19006". It features a logo of a book on a shelf and the heading "Vos recommandations". Below this, a list of article titles is displayed:
 - Article n°237452
 - Article n° 233478
 - Article n° 237429
 - Article n° 234128
 - Article n° 233716A blue "SE DÉCONNEXTER" button is located at the bottom left.
- Middle Screen (Mobile App - Login Screen):** A smartphone screen showing the "Bookshelf" logo and the text "Choisissez votre profil afin de recevoir des recommandations de lecture personnalisées". Below this, "User 100" is displayed, followed by a "SE CONNECTER" button. A large blue arrow points from the right side of this screen towards the rightmost screen.
- Right Screen (Mobile App - Home Screen):** A smartphone screen showing the "Bookshelf" logo and the heading "Vos recommandations". Below this, a list of article titles is displayed:
 - Article n°237452
 - Article n° 233478
 - Article n° 237429
 - Article n° 234128
 - Article n° 233716A blue "SE DÉCONNEXTER" button is located at the bottom left.

DÉMONSTRATION





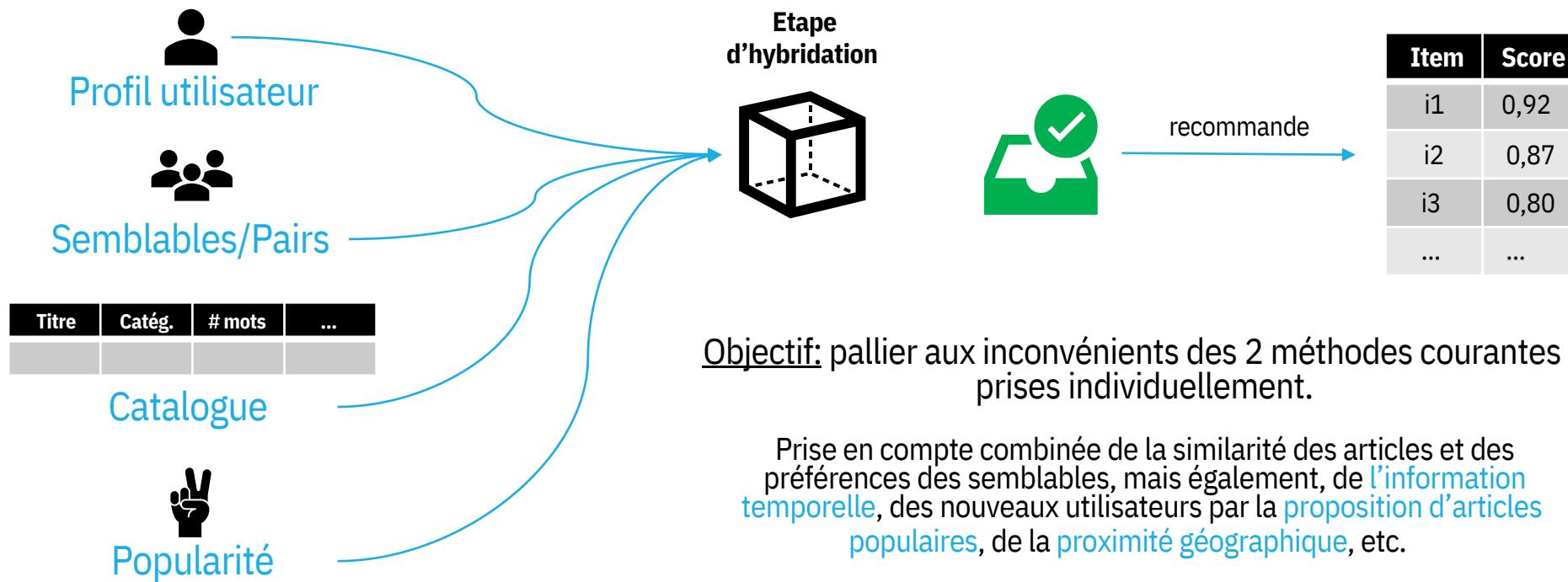
SYNTHÈSE

CONCLUSION

- Dans ce projet, nous avons pu:
 - Tester des modèles de recommandations
 - Déployer un modèle avec Azure Functions en serverless;
 - Intégrer un modèle dans l'application mobile Bookshelf.
- Il n'est pas évident d'évaluer la pertinence des algorithmes tant les méthodes de mesure du feedback utilisateur nes ont pas explicites (une question à réfléchir en interne?): nous sommes intéressés de voir les méthodes d'évaluation pratiquées dans la vie réelle.
- Nous pourrons petit à petit améliorer le modèle et notre architecture, en prenant en compte:
 - Les nouveaux articles et utilisateurs;
 - La date de publication des articles comme paramètre de filtrage;
 - La proximité géographique, etc.

NEXT STEPS : OPTIMISATION POSSIBLE

Création d'un modèle hybride
= combinaison de plusieurs approches





QUESTIONS / RÉPONSES





ANNEXES

OBJECTIFS DES SYSTÈMES DE RECOMMANDATION

- Pour l'utilisateur:
 - Réduire l'effort de l'utilisateur quant à la recherche d'articles qui peuvent l'intéresser (même ceux auxquels il n'aurait pas pensé);
 - Promouvoir des articles non populaires; découvrir des articles difficiles à trouver;
- Pour le propriétaire du système:
 - Accroître la satisfaction de l'utilisateur pour les fidéliser;
 - Augmenter le temps passé par l'utilisateur sur le système et potentiellement, les ventes;

Note: certains objectifs doivent être priorisés selon les besoins métier auxquels le système doit répondre.

COLLECTE DE DONNÉES UTILISATEUR

Il existe 2 façons de collecter les données permettant de modéliser l'intérêt de l'utilisateur pour des articles:

- La collecte **explicite** des données, qui consiste à impliquer l'utilisateur en lui demandant par exemple de donner une notation à un article, de mettre un 'like' à une publication, etc.; ces données sont difficiles à récolter car la démarche peut être fastidieuse pour l'utilisateur ou elle est perçue comme une atteinte à la vie privée;
- La collecte **implicite** des données, qui consiste à capturer des interactions de l'utilisateur avec le système, tels que l'historique de navigation ou de recherche, les clics, la durée de consultation, les téléchargements, la plateforme utilisée (ordi, mobile), etc.; ces données sont faciles à récolter—dans la mesure où la méthode respecte la vie privée (consentement explicite), mais elles ne disent pas si un article déplaît à un utilisateur ou si l'article consulté concerne l'utilisateur lui-même ou une tierce personne.



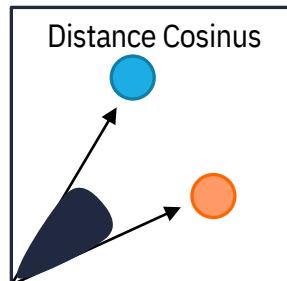
SIMILARITÉ COSINUS

Cosinus de l'angle entre 2 vecteurs.

*2 vecteurs de même orientation ont une similarité cosinus de 1,
tandis que 2 vecteurs diamétralement opposés ont une similarité cosinus de -1*

- La similarité cosinus est une mesure d'orientation : seule la direction des vecteurs est prise en compte
- La grandeur ou magnitude des vecteurs n'est pas considéré : par exemple, les différences d'échelle de notation entre différents utilisateurs;

$$D(x,y) = \cos(\theta) = \frac{x \cdot y}{\|x\| \|y\|}$$



- On crée une matrice de similarité entre les articles afin de déterminer leur proximité.

		articles				
		1				
			1			
				1		
					1	
						1

- Les N articles les plus proches (avec les similarités les plus proches de 1) seront recommandés.



MOINDRES CARRÉS ALTERNÉS (ALS = ALTERNATEDLEAST SQUARES)

- Avec la **factorisation matricielle**, on décompose une matrice de grande taille pour la transformer en 2 matrices de dimension inférieure, et dont le produit est égal à la matrice d'origine : c'est une *réduction de dimension*, où nous révélons **chaque utilisateur comme un vecteur de ses préférences** et, en même temps, **chaque article comme un vecteur de ce qu'il représente**.
- Les dimensions sont appelées **caractéristiques latentes ou cachées**, et nous les apprenons à partir des données à notre disposition. La méthode **des moindres carrés alternés** nous permet de manière itérative d'arriver à la meilleure approximation de R , en alternant entre l'optimisation de U et la fixation de V .

Matrice originale R de taille $u \times i$

(creuse ou clairsemée)

	Item 1	Item 2	Item 3	...	Item n
User 1	■		■		■
User 2		■		■	■
User 3	■			■	
...		■	■		
User n		■		■	



Matrice User U -taille $u \times f$

(f étant des facteurs latents)

	factor 1	factor 2
User 1	■	■
User 2	■	■
User 3	■	■
User 4	■	■
User 5	■	■
User 6	■	■
User 7	■	■
...	■	■
User n	■	■



Matrice Item V -taille $f \times i$

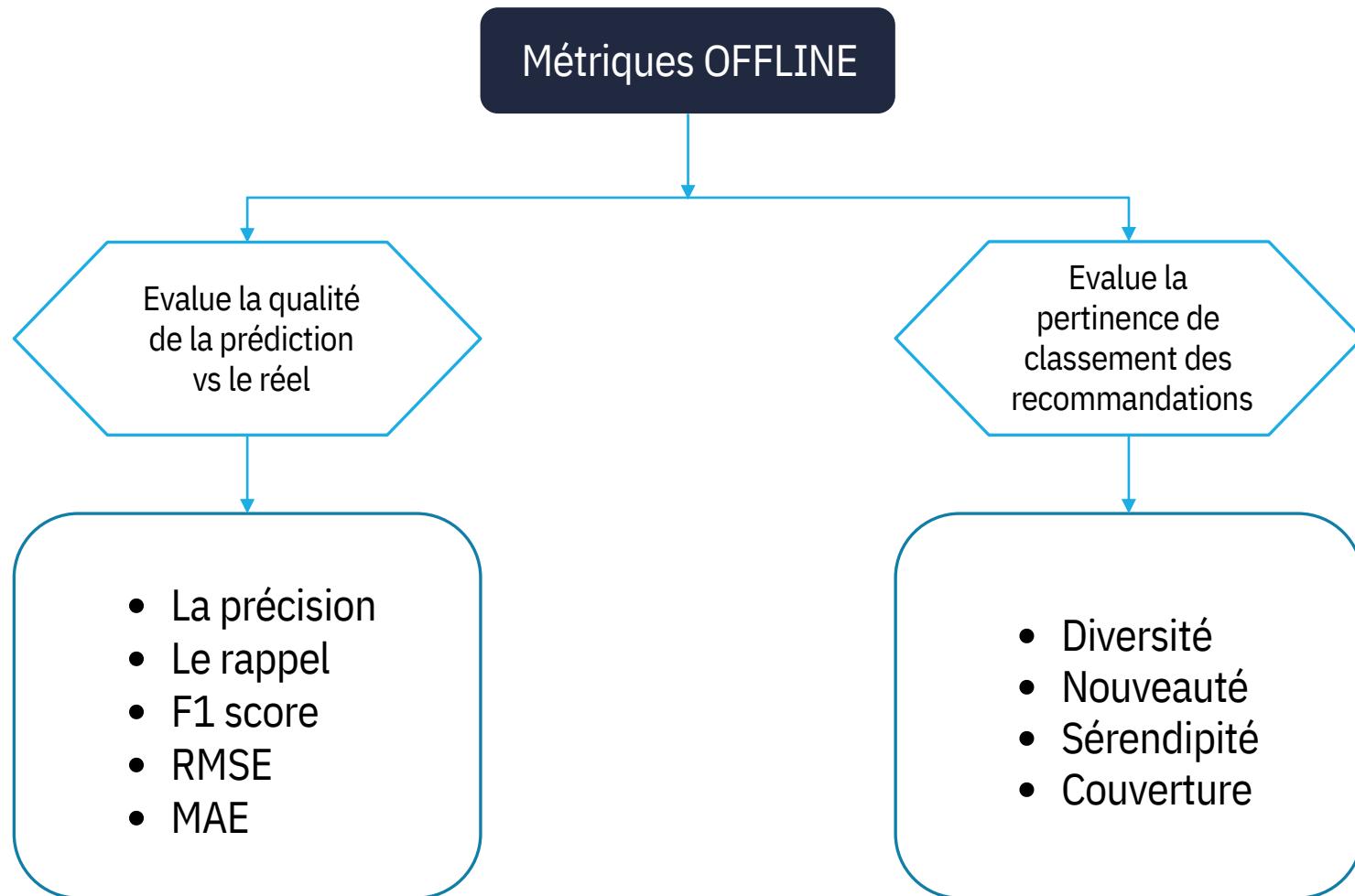
(f étant des facteurs latents)

	Item 1	Item 2	Item 3	...	Item n
factor 1	■	■	■		■
factor 2	■	■	■		■

- Le produit scalaire (vecteur utilisateur \times transposition des vecteurs item) donne un **score de recommandation**.

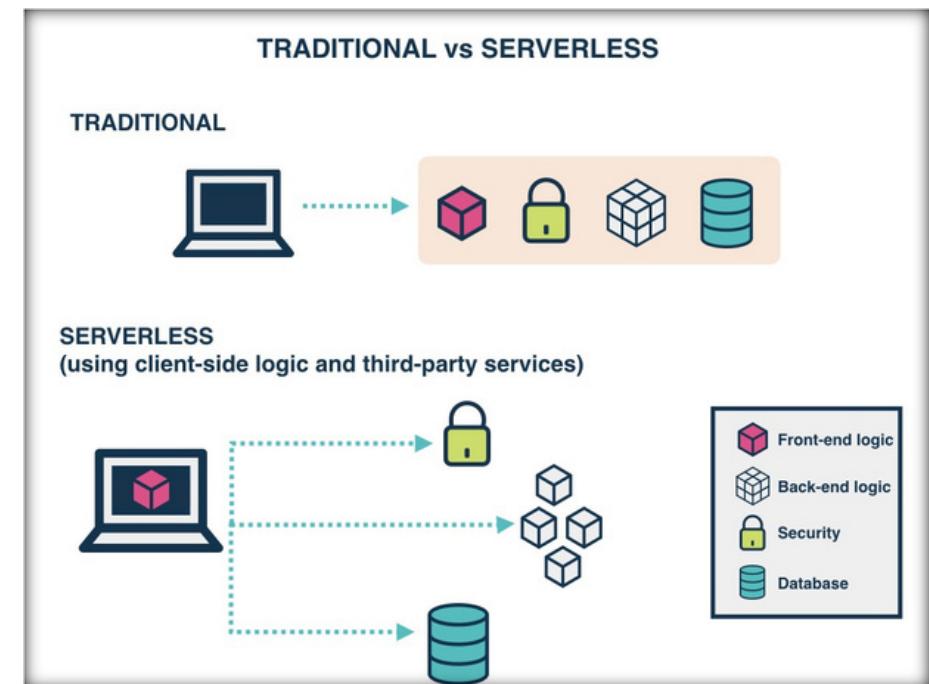
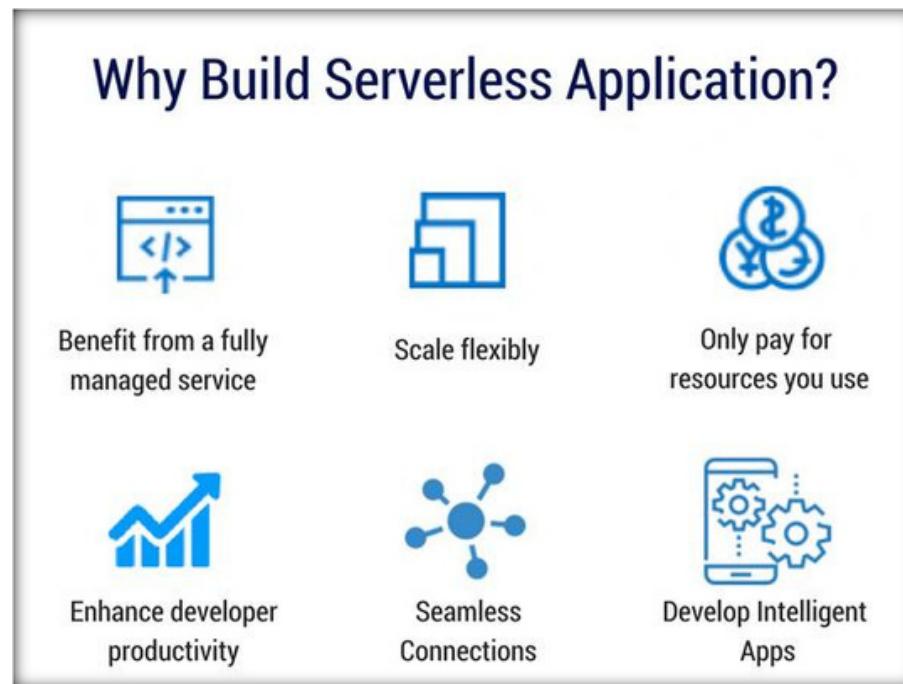


EVALUATION DES MODÈLES



QUELQUES NOTIONS SUR LE SERVERLESS

- Allocation dynamique des ressources nécessaires à l'exécution d'un code sans l'obligation de maintenir en interne la disponibilité, la scalabilité et la sécurité des serveurs.
- Le code est généralement sous la forme d'une fonction, dont le déclencheur peut être une requête http, des évènements de base de données, des alertes de surveillance, etc.



FACTURATION AZURE FUNCTIONS : CONSOMMATION

<https://azure.microsoft.com/fr-fr/pricing/details/functions/>

Consommation

L'offre de consommation Azure Functions est facturée en fonction des exécutions et de la consommation de ressources par seconde. La tarification du plan Consommation inclut une allocation mensuelle gratuite de 1 millions de requêtes et 400 000 Go de consommation de ressources par mois par abonnement avec tarification à l'utilisation pour toutes les applications de fonction de l'abonnement en question. L'offre Premium Azure Functions fournit de meilleures performances. De plus, elle est facturée à la seconde en fonction du nombre de vCPU-s et de Go-s consommés par vos fonctions Premium. Les clients peuvent également exécuter Functions dans le cadre de leur plan App Service au [tarif normal](#).

Mesure	Tarif	Attribution gratuite (par mois)
Délai d'exécution*	0,000014 €/secondes par Go	400 000 secondes par Go
Nombre total d'exécutions*	0,169 € par million d'exécutions	1 million d'exécutions

*Les octrois gratuits ne sont valables que pour les abonnements payants basés sur la consommation.

Remarque : un compte de stockage est créé par défaut avec chaque application Functions. Le compte de stockage n'est pas inclus dans l'octroi gratuit. Des [tarifs de stockage](#) et des [frais de mise en réseau](#) Standard sont facturés séparément selon le cas.

EXEMPLES DE FACTURATION: CONSOMMATION

Pour une fonction avec une consommation de mémoire constatée de 512 Mo, qui s'exécute 3 000 000 fois par mois et dont la durée d'exécution est de 1 seconde. La facturation mensuelle est calculée comme suit:

1

Calcul de la facturation de la consommation de ressources

Consommation de ressources (en secondes)	
Exécutons	3 million d'exécutons
Durée d'exécution (en secondes)	× 1 seconde
Consommation totale de ressources	3 million de secondes
Consommation des ressources (Go-s)	
Consommation de ressources convertie en Go-s	512 Mo / 1 024 Mo
Délai d'exécution (en secondes)	× 3 million de secondes
Total de Go-s	1,5 million de Go-s
Consommation des ressources facturable	
Consommation des ressources	1,5 million de Go-s
Octroi gratuit mensuel	- 400 000 secondes par Go
Consommation totale facturable	1,1 million de Go-s
Coût de la consommation mensuelle des ressources	
Consommation des ressources facturable	1,1 million de Go-s
Tarif de la consommation des ressources	× 0,000014 €/secondes par Go
Coût total	14,843 €

3

Calcul de la facturation des exécutions

Exécutions facturables	
Nombre total d'exécutions mensuelles	3 million d'exécutons
Exécutions mensuelles gratuites	- 1 million d'exécutons
Exécutions facturables par mois	2 million d'exécutons
Coût des exécutions mensuelles	
Exécutions facturables par mois	2 million d'exécutons
Prix par million d'exécutons	× 0,169 €
Coût d'exécution mensuel	0,338 €

Calcul de la facturation de la consommation totale

Coût mensuel total	
Coût de la consommation mensuelle des ressources	14,843 €
Coût des exécutions mensuelles	+ 0,338 €
Coût mensuel total	15,18 €

RÉFÉRENCES

- <https://interstices.info/les-systemes-de-recommandation-categorisation/>
- <https://ichi.pro/fr/9-mesures-de-distance-en-science-des-donnees-159983401462266>
- <https://serverless-stack.com/chapters/fr/what-is-serverless.html>
- <https://docs.microsoft.com/fr-fr/azure/azure-functions/create-first-function-vs-code-python>



Ce document a été produit dans le cadre de la soutenance du projet n°9 du parcours Ingénieur IA d'OpenClassrooms:
«Réalisez une application mobile de recommandation de contenu»

