The background of the slide is a complex network diagram. It consists of numerous nodes of varying sizes and colors (dark blue, light blue, and grey) connected by thin, light grey lines. Some nodes are highlighted with larger, concentric circles. The overall aesthetic is modern and technological, suggesting a network or data structure.

MY CONTENT : APPLICATION MOBILE DE RECOMMANDATION DE CONTENU

Vochangy Joan ALEONARD – 23/07/2021

AGENDA DU JOUR



CONTEXTE PROJET
ET DONNEES



SYSTÈMES DE
RECOMMANDATION



ARCHITECTURE ET
DÉPLOIEMENT



INTÉGRATION / DEV
CONTINUS & DEMO



CONTEXTE PROJET ET DONNEES

MY CONTENT



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

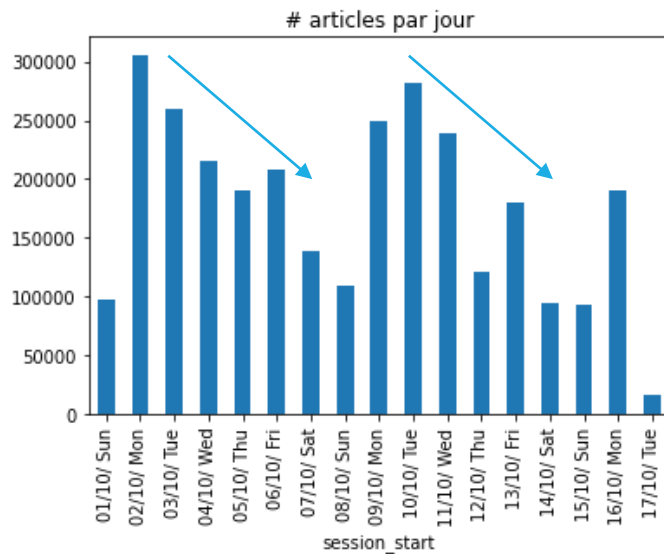


Informations essentielles :

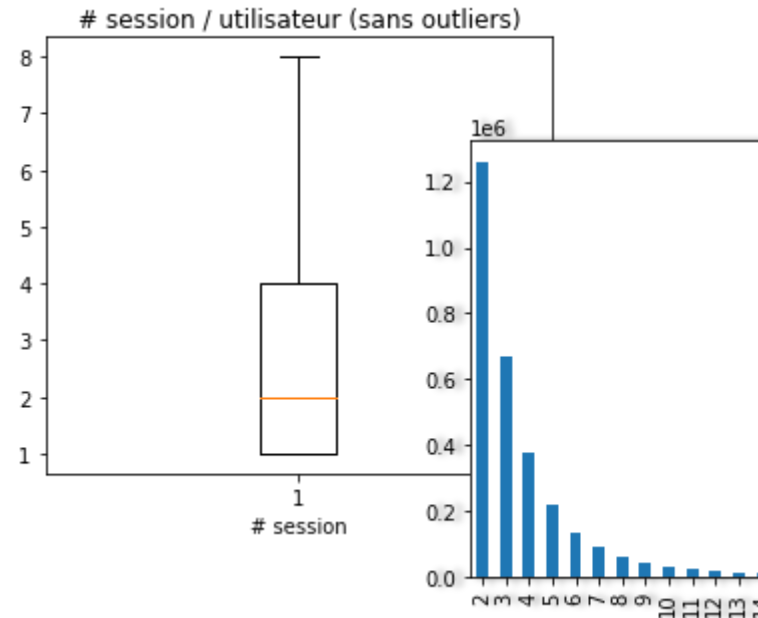
- Nb utilisateurs uniques : ~323k
- Nb articles uniques : ~46k
- Nb interactions (clicks) : ~3 millions

EXPLORATION DES DONNÉES : LES UTILISATEURS

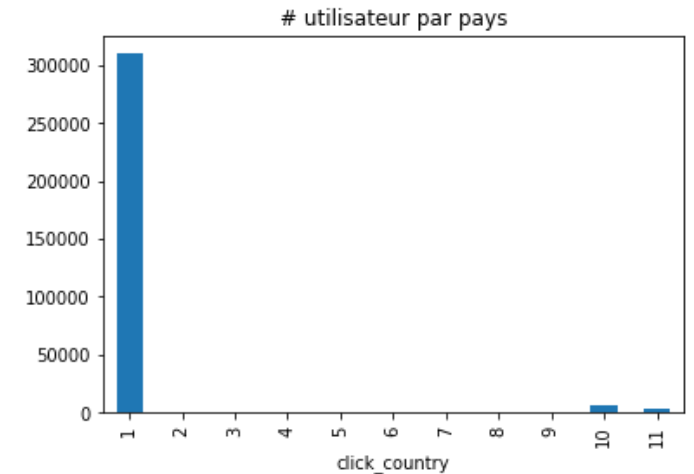
(1/3)



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



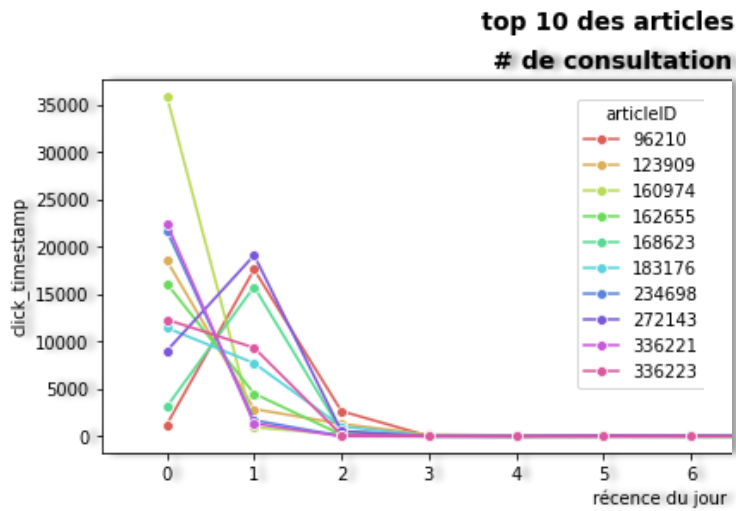
Sur les 15 jours d'observations, les utilisateurs ont eu en moyenne 3 sessions : la majorité consulte 2 articles par session



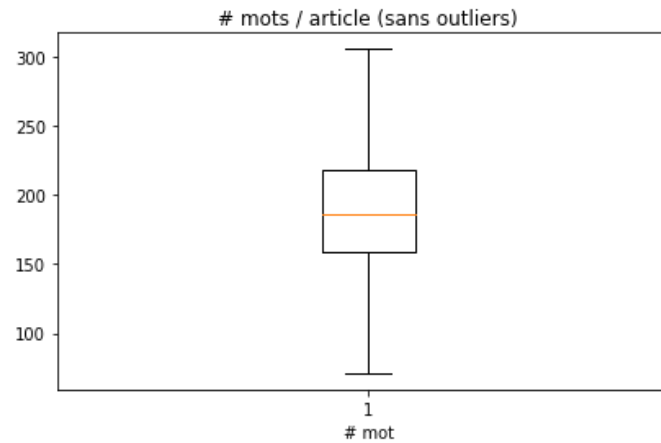
La majorité d'utilisateur sont dans le pays 1 (principalement sur les régions 25, 21 et 13)

EXPLORATION DES DONNÉES : LES ARTICLES

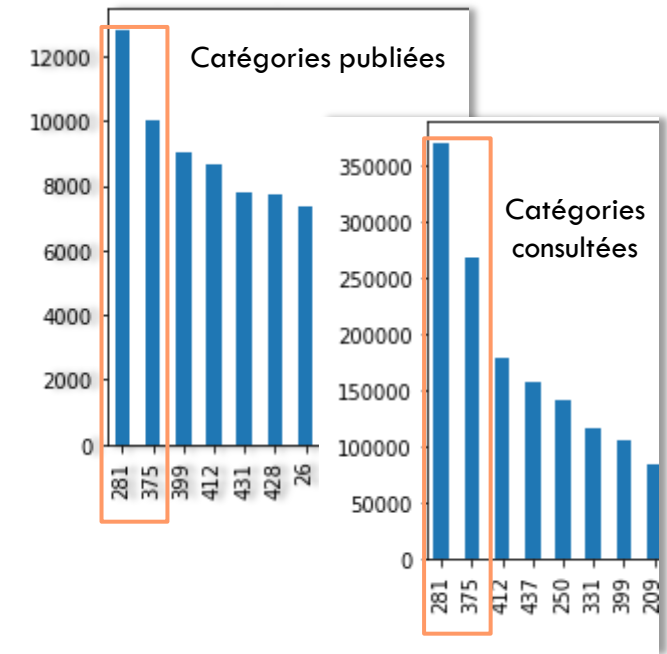
(1/3)



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

(1 / 3)

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



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



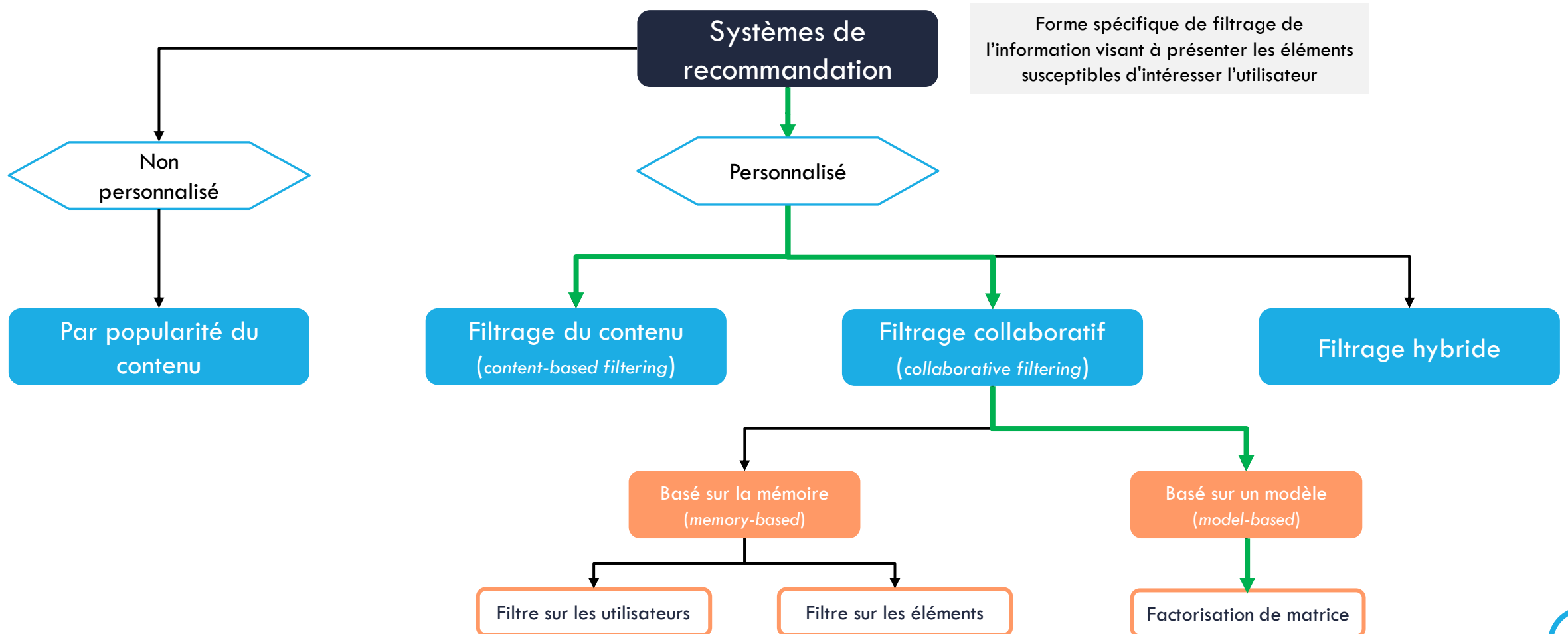
Evolution dans le temps des préférences utilisateurs : elles changent en fonction de l'actualité (*élection, événement sportif, catastrophe naturelle, crise sanitaire, ...*)



SYSTÈMES DE RECOMMANDATION

CATÉGORIES DE SYSTÈMES DE RECOMMANDATION

(1/3)

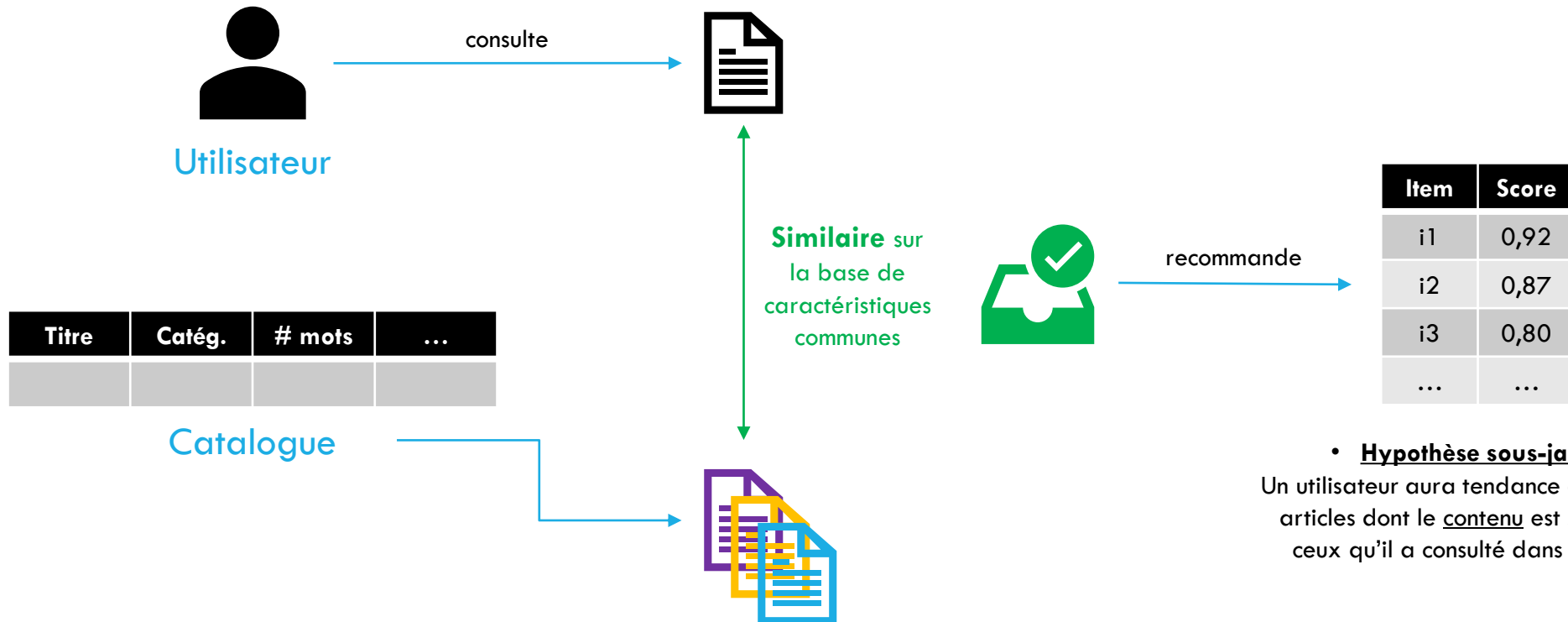


CATÉGORIES DE SYSTÈMES DE RECOMMANDATION

(2/3)

Basé sur le Contenu (content-based)

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



- **Hypothèse sous-jacente**

Un utilisateur aura tendance à aimer des articles dont le contenu est similaire à ceux qu'il a consulté dans le passé.

PROCESSUS DE LA RECO BASÉE SUR LE CONTENU

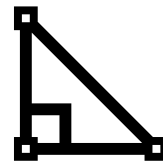


Données utilisées

- ✓ Données de clicks
- ✓ Matrice Embeddings

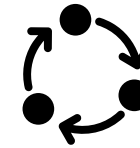
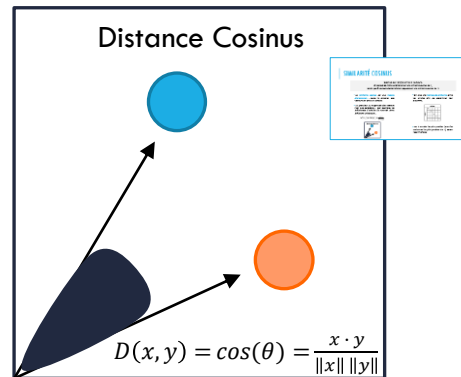
Données non utilisées

- ✗ 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

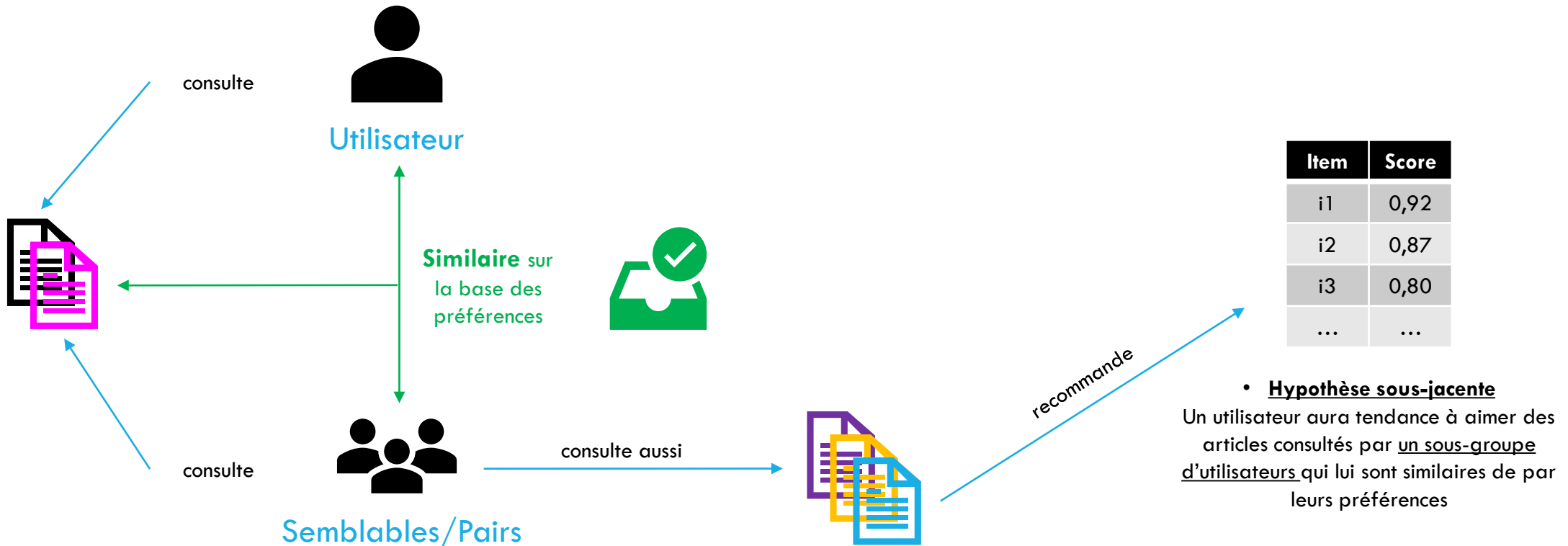
CATÉGORIES DE SYSTÈMES DE RECOMMANDATION

(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



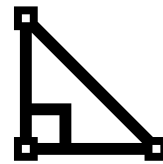
Données utilisées

✓ Données de clicks

Module additionnel

✓ Implicit

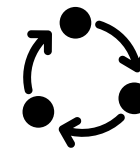
<https://implicit.readthedocs.io/en/latest/>



Algorithmes

✓ Moindres carrés alternés

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



Étapes

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

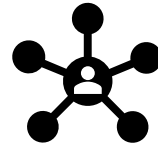
	Content-based	Collaborative-Filtering
Avantages	<ul style="list-style-type: none">• Peu d'impact du démarrage à froid : pas besoin de données relatives aux utilisateurs : recommandation possible de nouveaux articles dès leur apparition dans la base.	<ul style="list-style-type: none">• Pas besoin de métadonnées sur les articles ou les utilisateurs.
Inconvénients	<ul style="list-style-type: none">• Manque de personnalisation.• Risque 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 nouvel article sans consultation.

CHOIX DE NOTRE MODÈLE



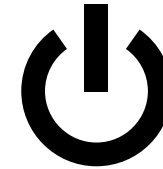
Absence de notation

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.

EVALUATION DE LA QUALITÉ DES MODÈLES



Hors ligne : Train/Test set

- Création d'un jeu d'apprentissage et de test (pas de convention existante)
- Utilisation de l'algorithme sur le jeu d'apprentissage
- Mise en place de métriques de mesure de qualité entre le réel et les prédictions



Ici on mesure des scores, ce qui n'est pas tout à fait de l'amélioration de la qualité du système de recommandation



En ligne : A/B Testing

- Test de modèles avec une unique différence sur un groupe A et un groupe B d'utilisateurs
- Validation de la méthode répondant à l'objectif défini (augmentation du nbre de consultation, satisfaction / fidélité utilisateur, etc.)

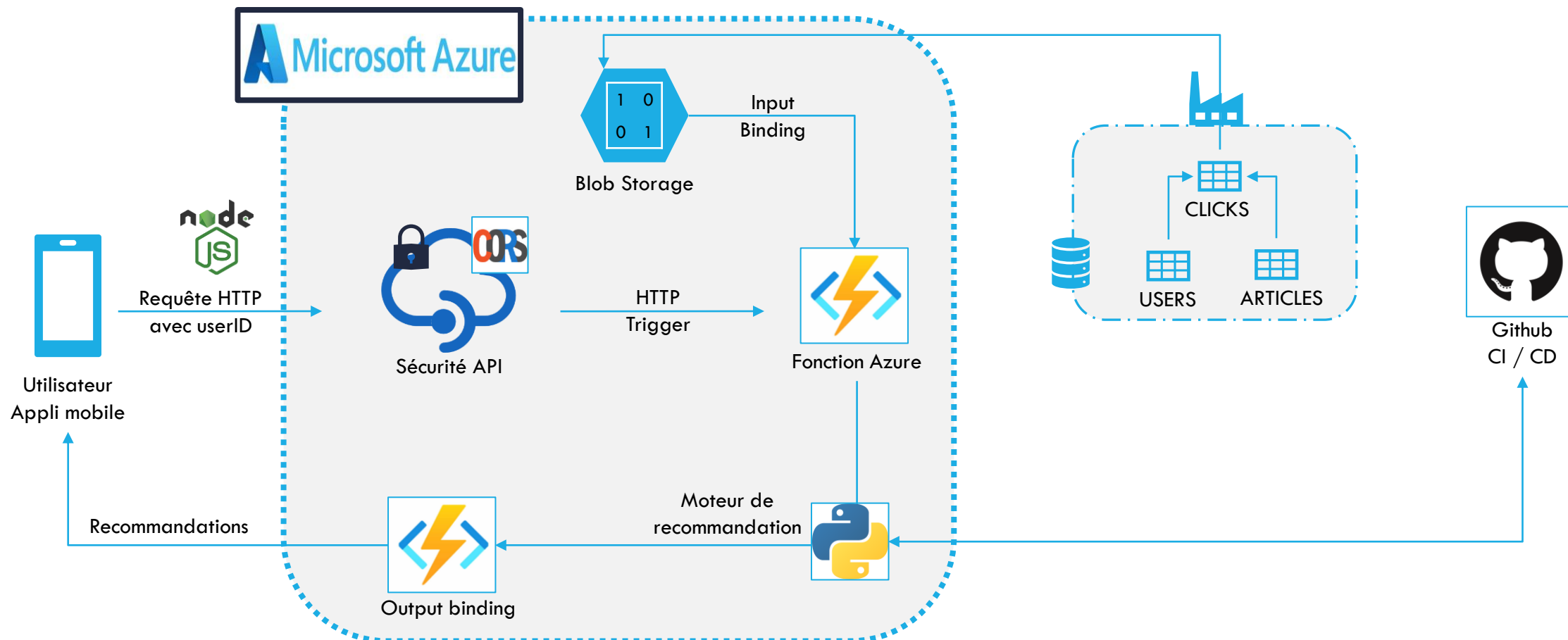


Bien que **approprié et pertinente**, cette méthode d'évaluation est très coûteuse



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**.

Microsoft Azure

Home > Resource groups >

Resource groups

Répertoire par défaut

+ Create Manage view

Filter for any field...

Name
DefaultResourceGroup-PAR
mycontentreco
ourcontentreco
p7-demo

ourcontentreco Resource group

Search (Ctrl+/)

+ Create Edit columns Delete resource group Refresh Export to CSV Open query Assign tags Move Delete Export template

Essentials

Subscription (change): [Microsoft Azure Sponsorship 2](#) Deployments: [No deployments](#)

Subscription ID: 707e32d3-9ddb-48cb-96e5-2f846d6bc387 Location: France Central

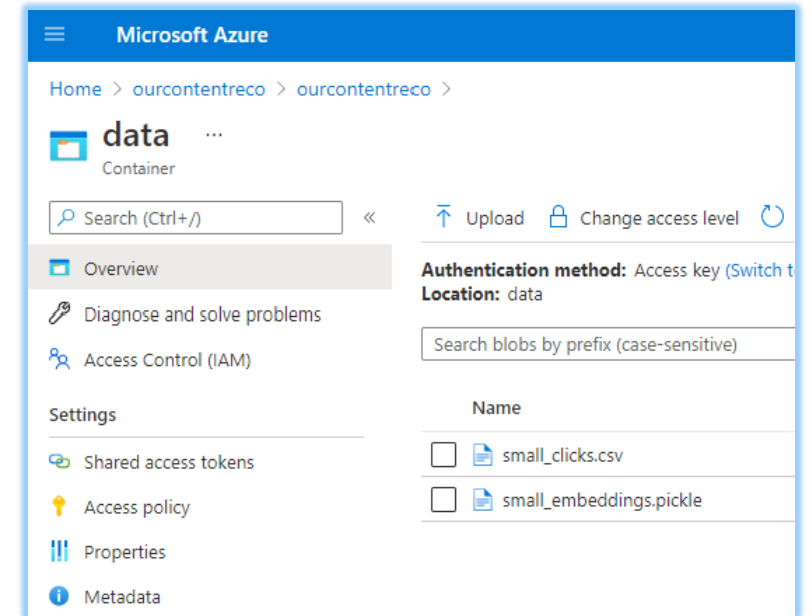
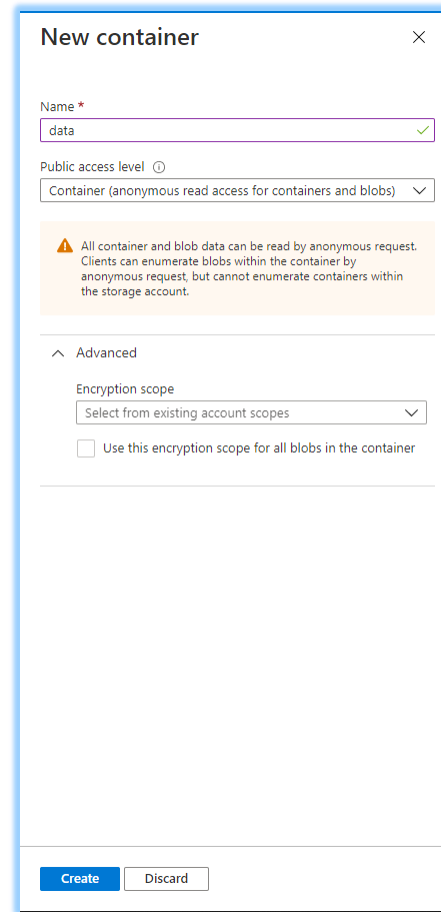
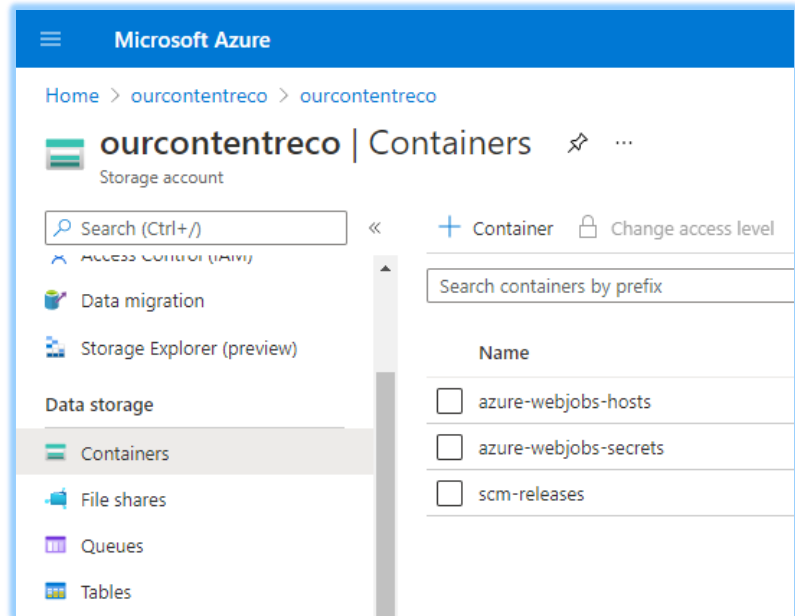
Tags (change): [Click here to add tags](#)

Filter for any field... Type == all Location == all Add filter

Showing 1 to 4 of 4 records. Show hidden types

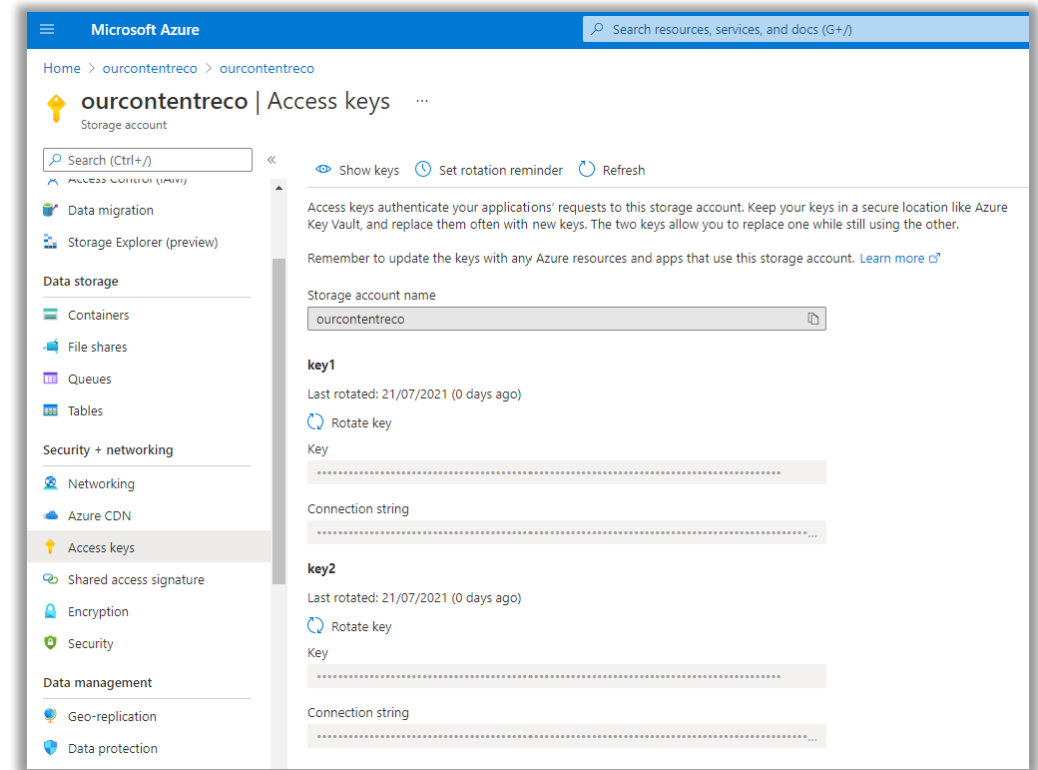
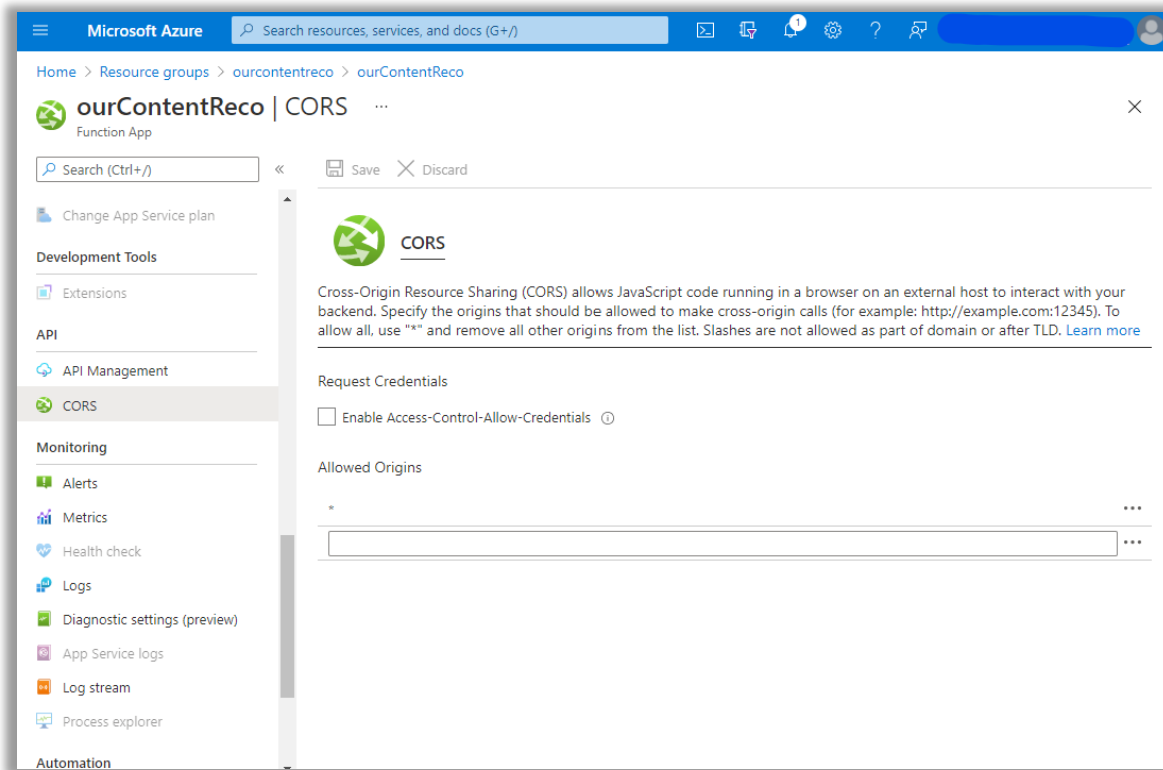
Name	Type	Location
ASP-ourContentReco-4fa5	App Service plan	France Central
ourcontentreco	Application Insights	France Central
ourcontentreco	Storage account	France Central
ourContentReco	Function App	France Central

CHARGEMENT DES FICHIERS (BLOB) DANS LE CONTAINER DU STORAGE ACCOUNT



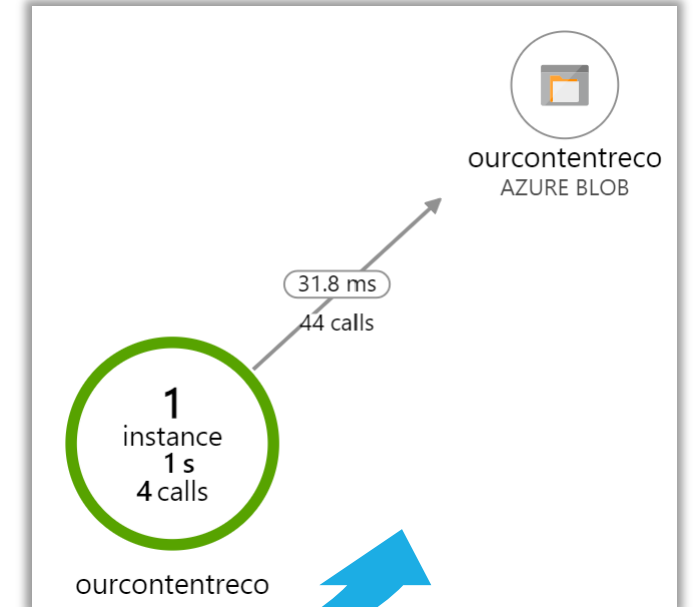
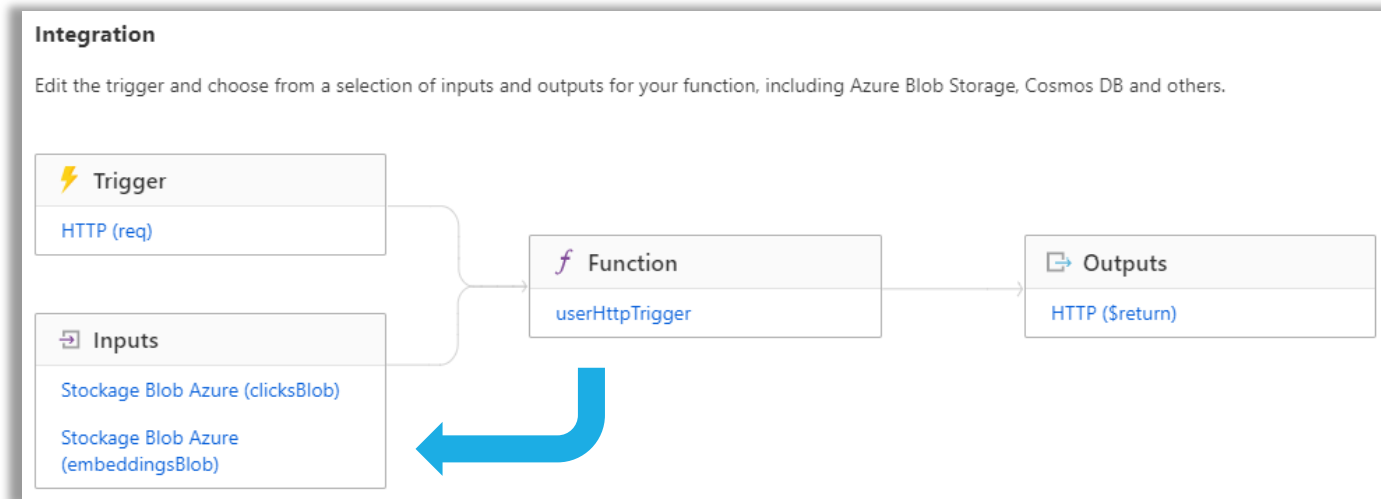
GESTION DES ACCÈS

(PARTAGE DE RESSOURCES INTRA-ORIGINES, AZURE STORAGE)



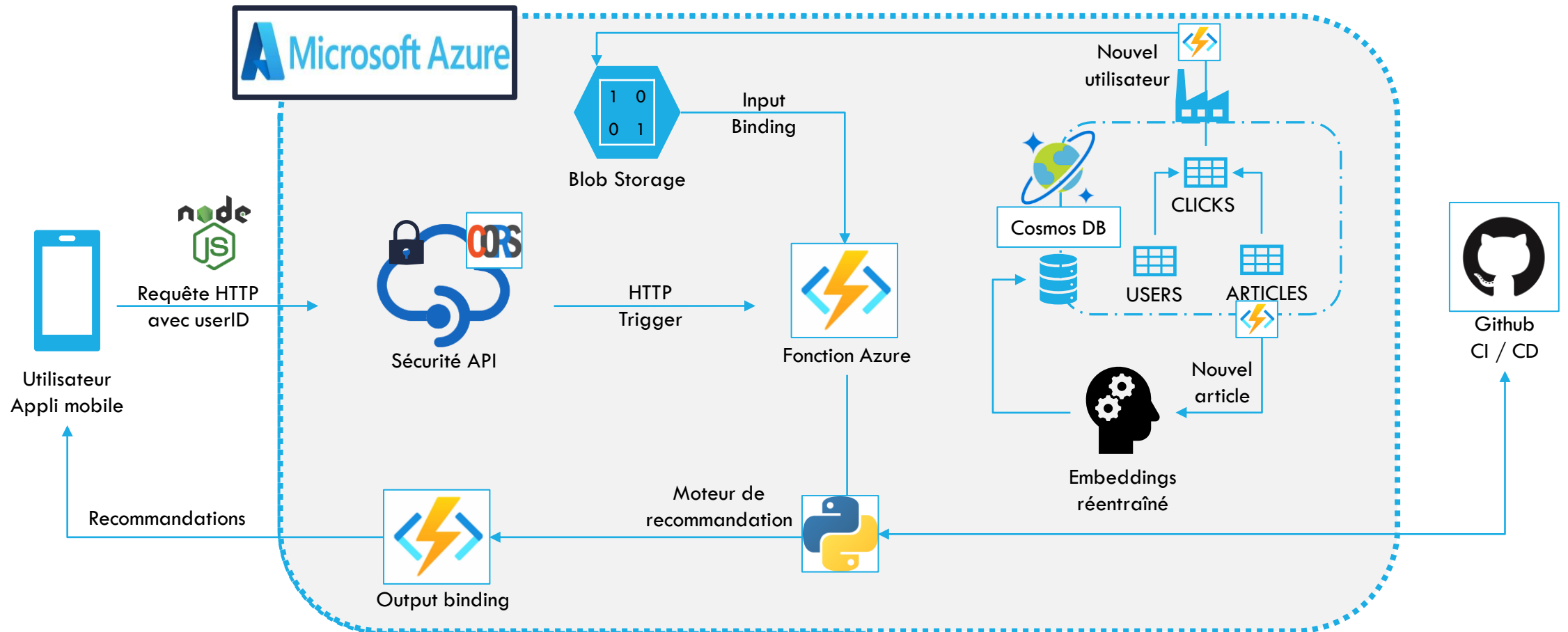
FUNCTION APP & APPLICATION INSIGHTS :

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/)



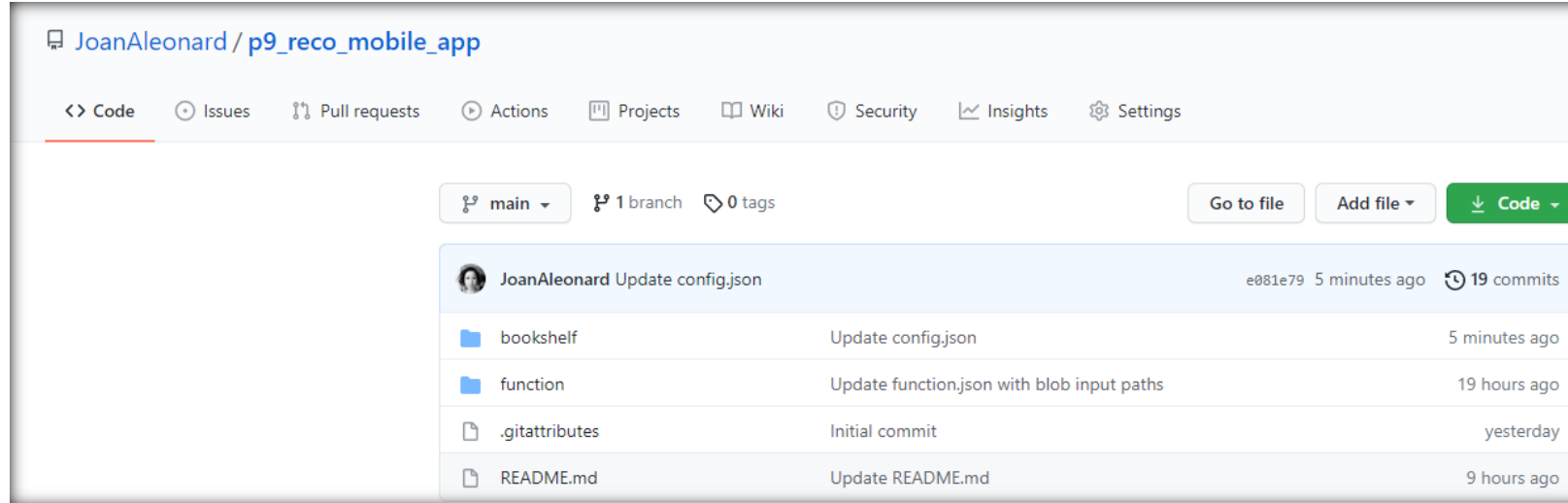
*CORS: Cross-Origin Resource Sharing
Partage de ressources inter-origines



INTÉGRATION / DEV
CONTINUS & DEMO

GESTION DES VERSIONS POUR UN SYSTÈME IA DE BOUT-EN-BOUT

https://github.com/JoanAleonard/p9_reco_mobile_app



OU



Azure
Repos



TEST/RUN SUR AZURE PORTAL :

USER 100

The screenshot displays the Azure Portal interface for testing a function named 'userHttpTrigger'. The left sidebar shows the 'Developer' section with 'Code + Test' selected. The main area shows the Python code for the function, which includes imports for logging, azure.functions, pandas, numpy, sklearn, and io. The function 'get_ContentBased_Reco' takes 'userID', 'small_clicks', and 'small_embeddings' as parameters and returns a list of recommended article IDs. The right sidebar shows the 'Input' tab for testing the function. The 'HTTP method' is set to 'POST', the 'Key' is 'master (Host key)', and the 'Body' is a JSON object with 'userId': '100'. The 'Output' tab shows the 'HTTP response code' as '200 OK' and the 'HTTP response content' as a list of article IDs: '237452, 233478, 237429, 234128, 233716'.

Home > ourContentReco > userHttpTrigger

userHttpTrigger | Code + Test

Function

Search (Ctrl+/) Save Discard Refresh Test/Run Upload Get function URL

Overview

Developer

Code + Test

Integration

Monitor

Function Keys

Editing functions in the portal is not supported for Linux Consumption Function Apps.

ourContentReco \ userHttpTrigger \ _init_.py

```
1 import logging
2 import azure.functions as func
3 import pandas as pd
4 import numpy as np
5 from sklearn.metrics.pairwise import cosine_similarity
6 from io import StringIO
7
8
9 def get_ContentBased_Reco(userID, small_clicks, small_embeddings, n_reco=5):
10     """Return 5 recommended articles ID to user"""
11
12     # Get the list of articles viewed by the user
13     var = small_clicks.loc[small_clicks.user_id == userID]['article_id'].to_list()
14
15     # Get the list of unique article ID in small_clicks
16     list_articleID = sorted(list(small_clicks.article_id.unique()))
17
18     # Retrieve the corresponding index of the articles viewed by userID in var
19     idx_var = []
20     for i in range(0, len(var)):
21         for idx, item in enumerate(list(list_articleID)):
22             if item == var[i]:
23                 idx_var.append(idx)
24
25     # Select the last element of the list
26     value = idx_var[-1]
27     # print(value)
28
29     # Compute the cosine similarity
30     emb = small_embeddings
31     distances = cosine_similarity([emb[value]], emb)[0]
```

Input Output

Provide parameters to test the HTTP request. Results can be found in the Output tab.

HTTP method

POST

Key

master (Host key)

Query

+ Add parameter

Headers

+ Add header

Body

```
1 {}
2 "userId": "100"
3 {}
```

HTTP response code

200 OK

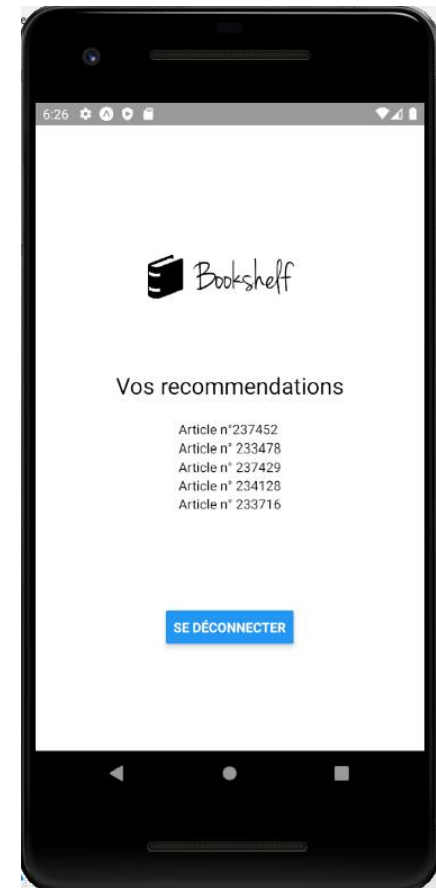
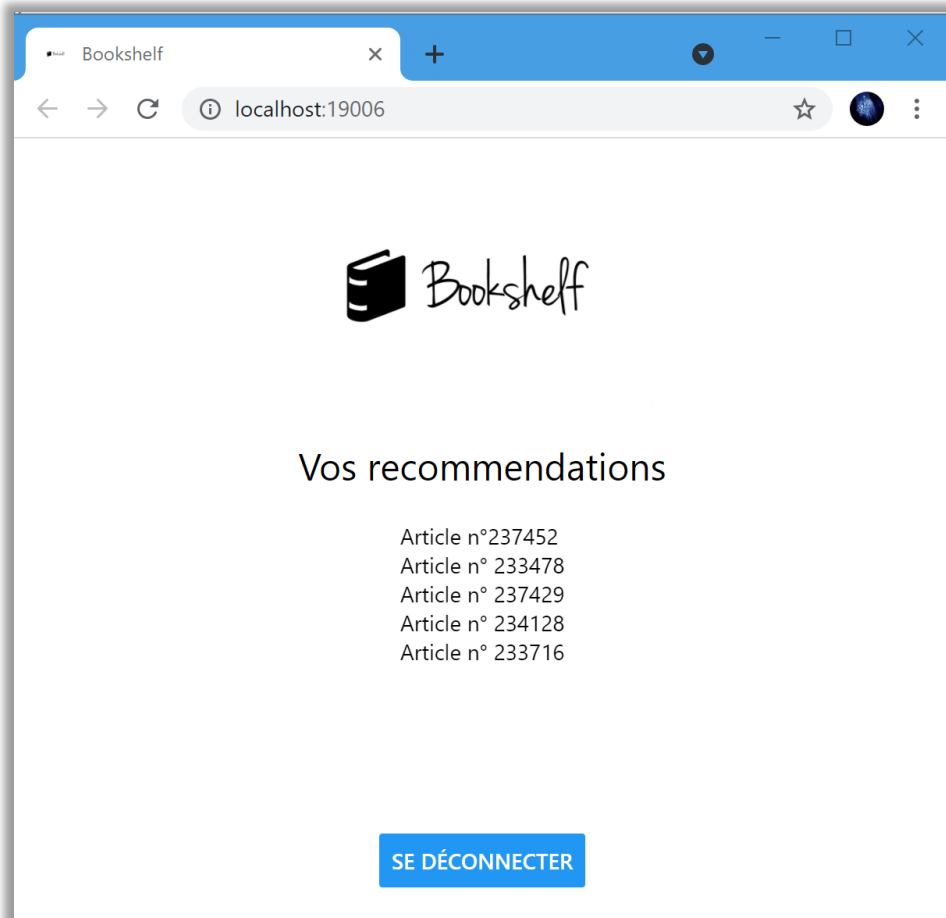
HTTP response content

237452, 233478, 237429, 234128, 233716

Run Close

TEST SUR NAVIGATEUR & APPLI MOBILE :

USER 100



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.

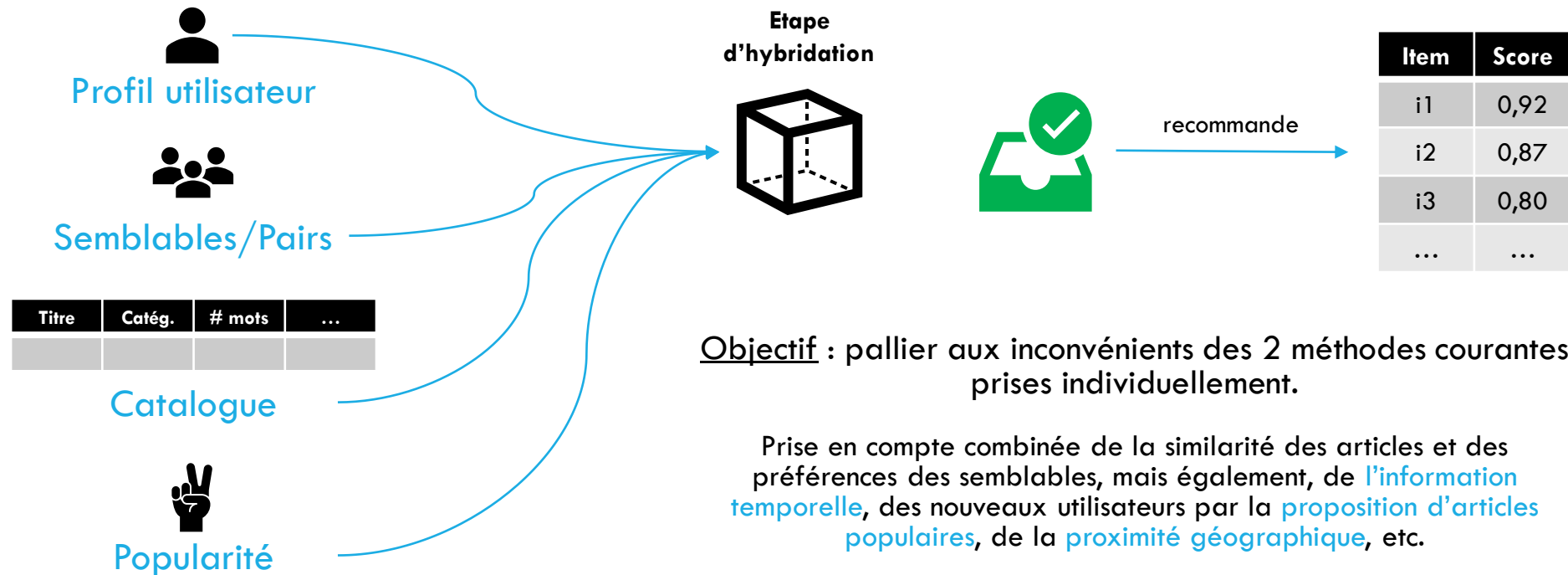
❑ Il n'est pas évident d'évaluer la pertinence des algorithmes tant les méthodes de mesure du *feedback* utilisateur ne sont pas explicites (**une question à réfléchir en interne ?**) : nous sommes intéressés de voir les méthodes d'évaluation pratiquées dans la vie réelle.

❑ Nous pourrions 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



The background of the slide is a complex network diagram. It consists of numerous small grey dots connected by thin grey lines, forming a web-like structure. Several larger circles are also present: a large white circle with a dark blue center at the top, a large blue circle at the bottom left, and a large grey circle at the bottom center. There are also smaller blue and dark blue circles scattered throughout the network.

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.

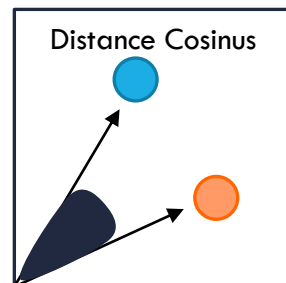
SIMILARITÉ COSINUS

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 considérée : 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					
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 = ALTERNATED LEAST 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		

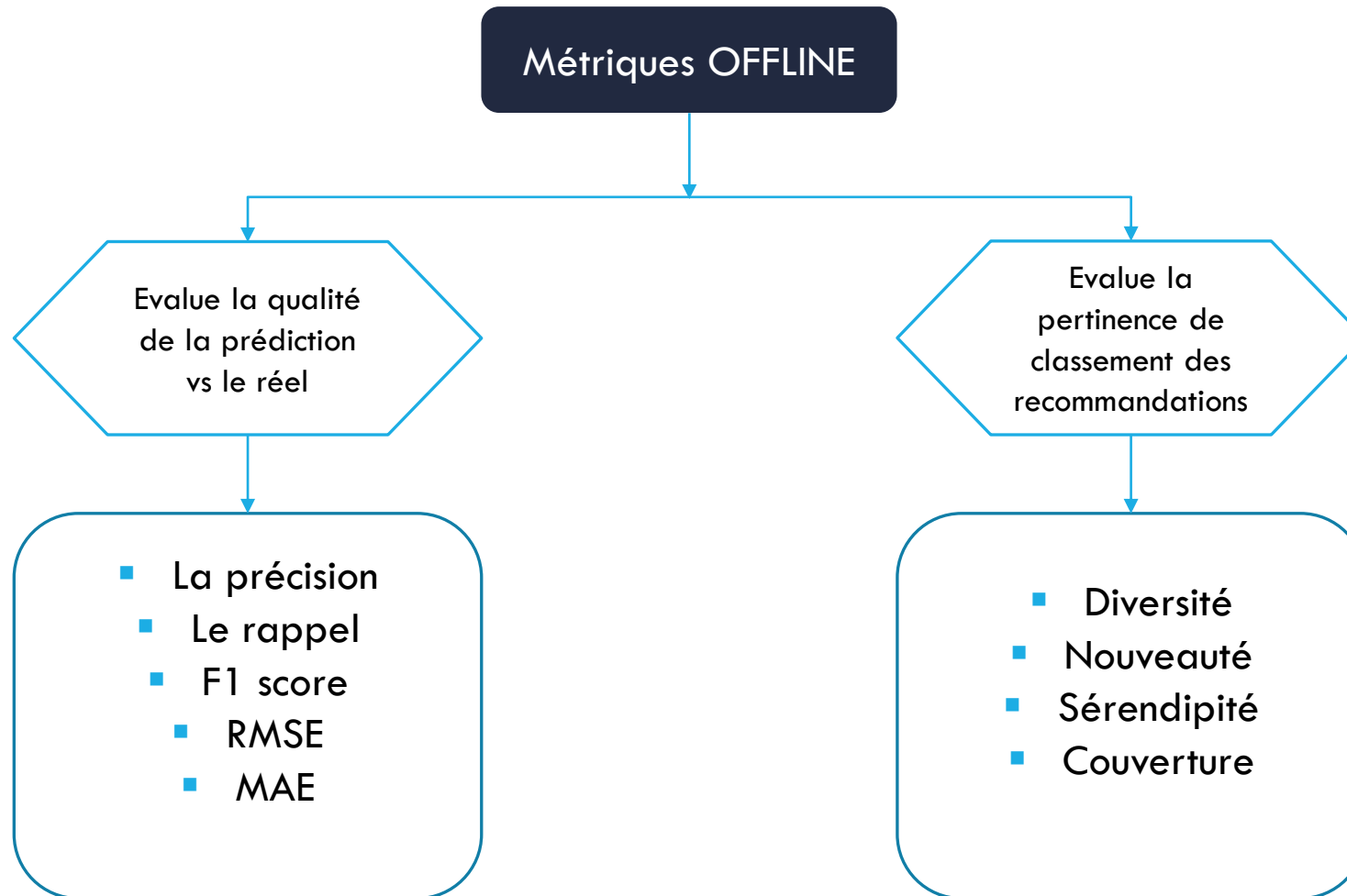


Matrice Item V - taille $f \times i$
(f étant des facteurs latents)

	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 événements de base de données, des alertes de surveillance, etc.

Why Build Serverless Application?



Benefit from a fully managed service



Scale flexibly



Only pay for resources you use



Enhance developer productivity



Seamless Connections



Develop Intelligent Apps

TRADITIONAL vs SERVERLESS

TRADITIONAL



SERVERLESS (using client-side logic and third-party services)



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)	
Exécutions	3 million d'exécutions
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 €

2

Calcul de la facturation des exécutions

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

3

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 »

Mentor : Thierno DIOP

Evaluateur : Yoann BOJ

