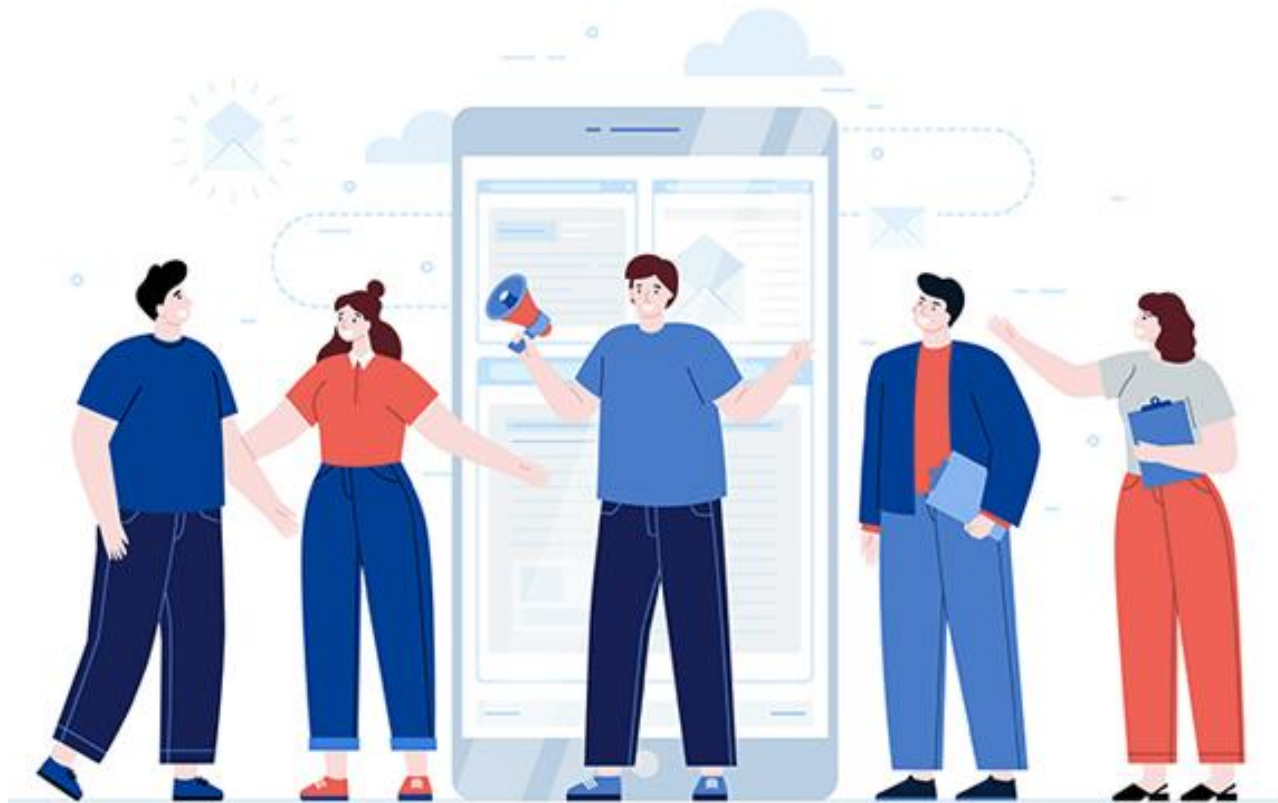




RECOMMENDATION SYSTEM

Cédric Dietzi
11/06/2021



08/06/2021

Contexte

La société 'My Content' souhaite réaliser son premier MVP de recommandation de contenus d'actualités et de livres sous la forme d'une application mobile.

Système de recommandations – Cadre d'analyse

Domaine

Actualités et livres => Dynamiques différentes

Objectif

Améliorer le taux de clics

Contexte de recommandation

Mobilité => Latence faible

Niveau de personnalisation

News: + de popularité Livres: + de personnalisation

Interfaces

Type de sortie: recommandations (top k)

Type d'entrée: implicite

Recommendation algorithms

Content based filtering: cold start

Collaborative filtering

Système de recommandations – Cadre d'analyse

Domaine

Actualités et livres => Dynamiques différentes

Objectif

Améliorer le taux de clics

Contexte de recommandation

Mobilité => Latence faible

Niveau de personnalisation

News: + de popularité Livres: + de personnalisation

Interfaces

Type de sortie: recommandations (top k)

Type d'entrée: implicite

Recommendation algorithms

Content based filtering: cold start

Collaborative filtering

MVP

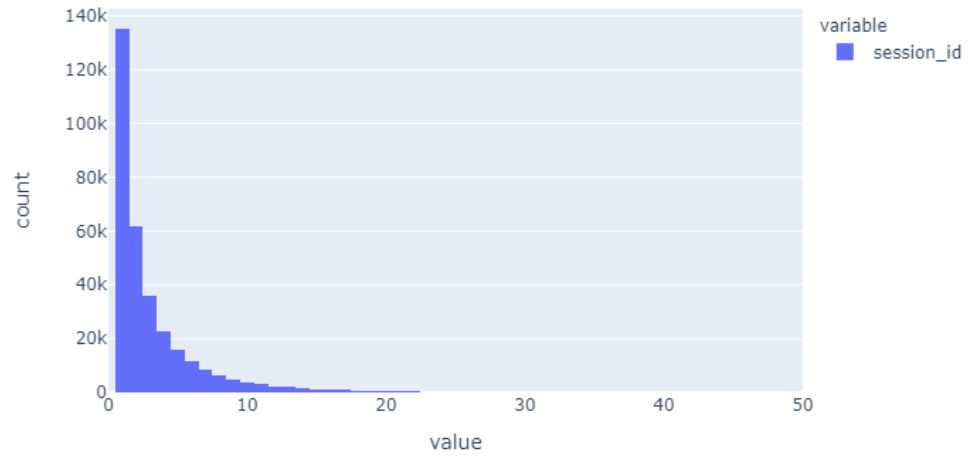


Analyse exploratoire de données

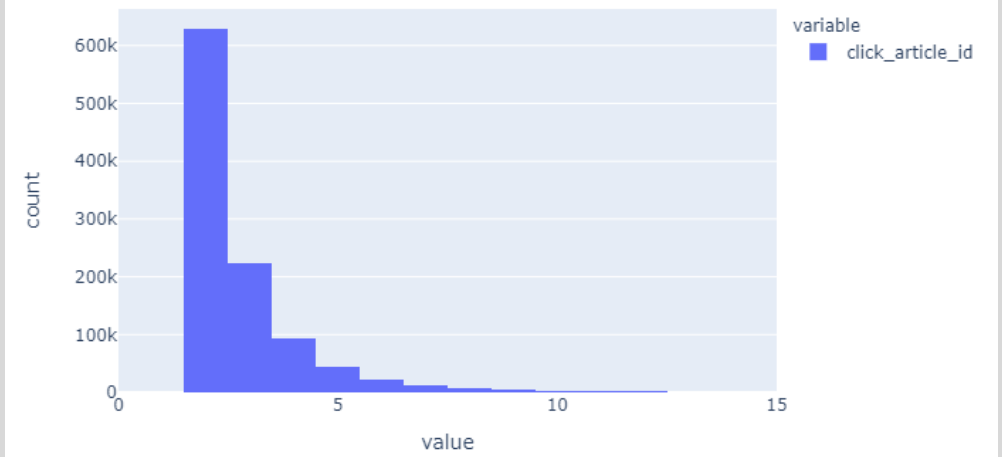
# utilisateurs	322 897
# sessions	1 048 594
# articles lus	46 033
# articles	364 047

	Moyenne	Min	Max
# sessions par utilisateur	3.25	1	207
# articles par session	2.85	2	124
# utilisateurs par article lu	64,10	1	34145
# articles par utilisateurs	9.14	2	1048

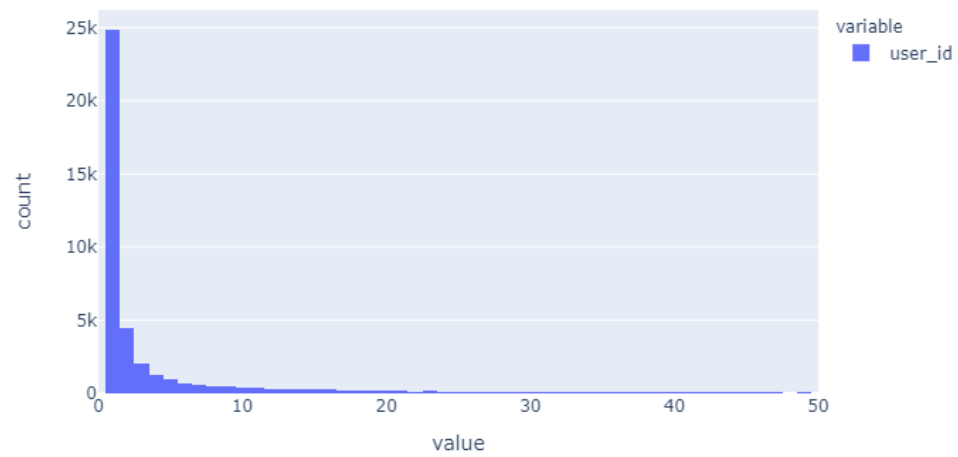
Number of sessions per user



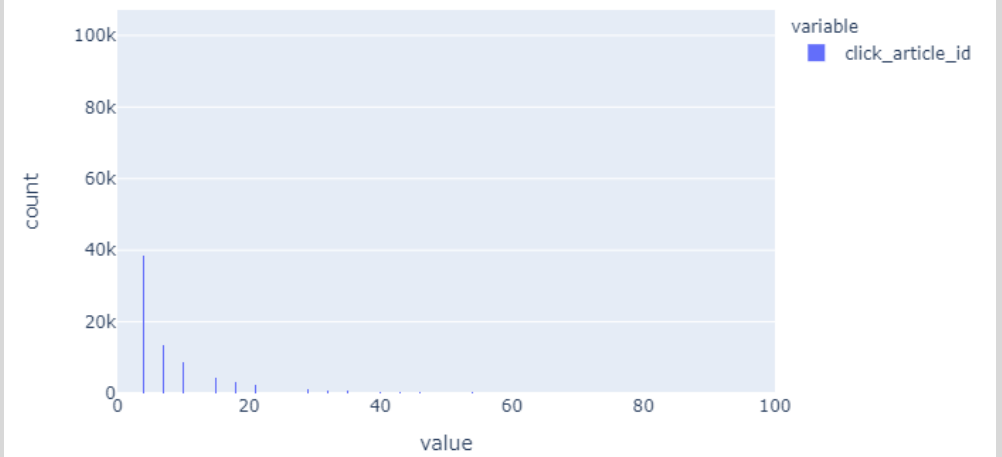
Number of articles per session



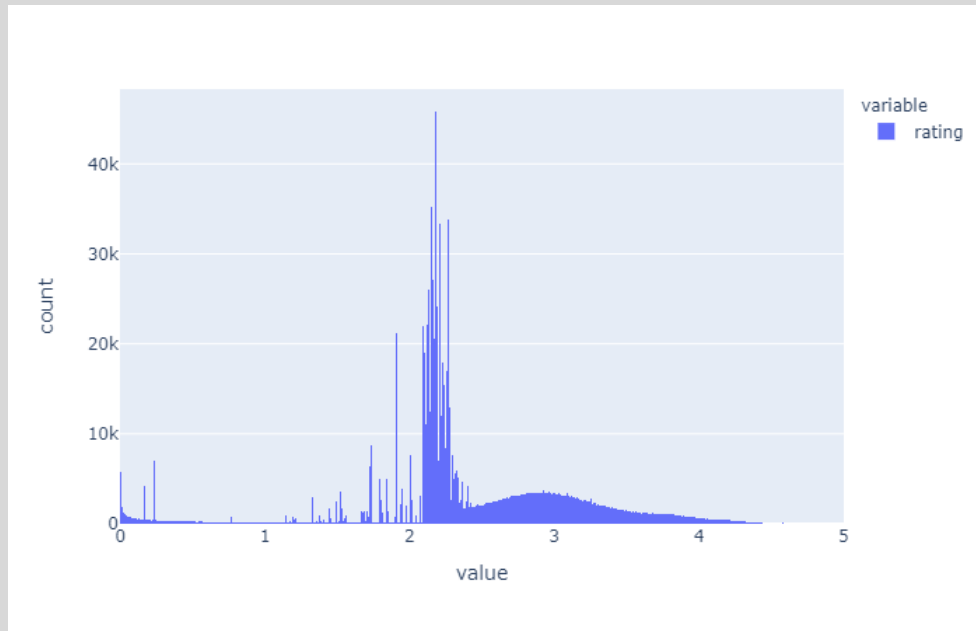
Number of users per article



Number of articles per user

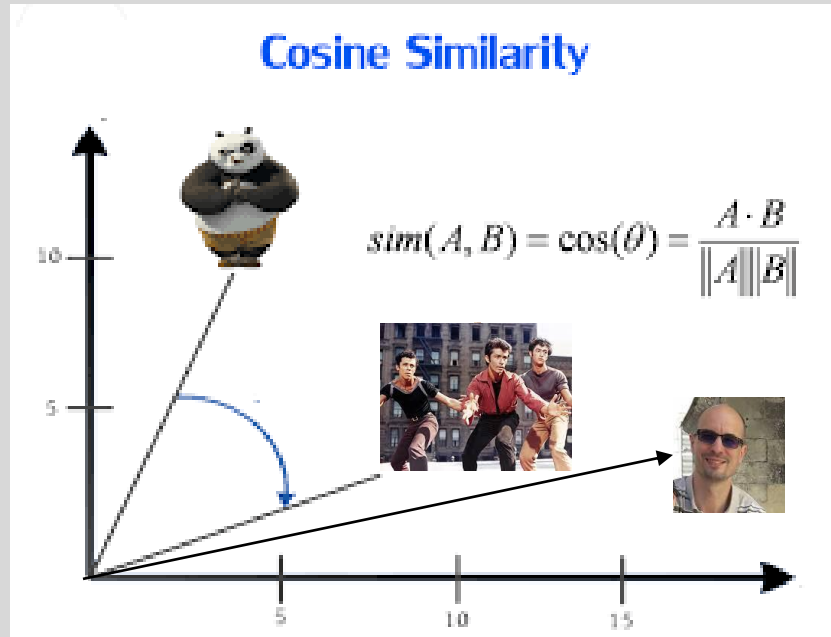


Rating implicite

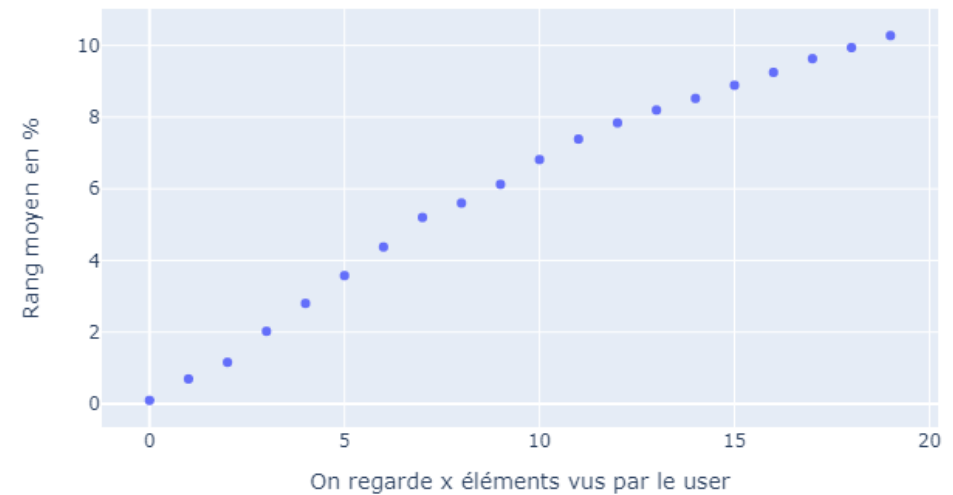


- Temps passé sur un article
- Normalisé par la longueur de l'article
- Les queues de la distribution sont resserrées autour de la moyenne
- Plage ramenée entre 0 et 5

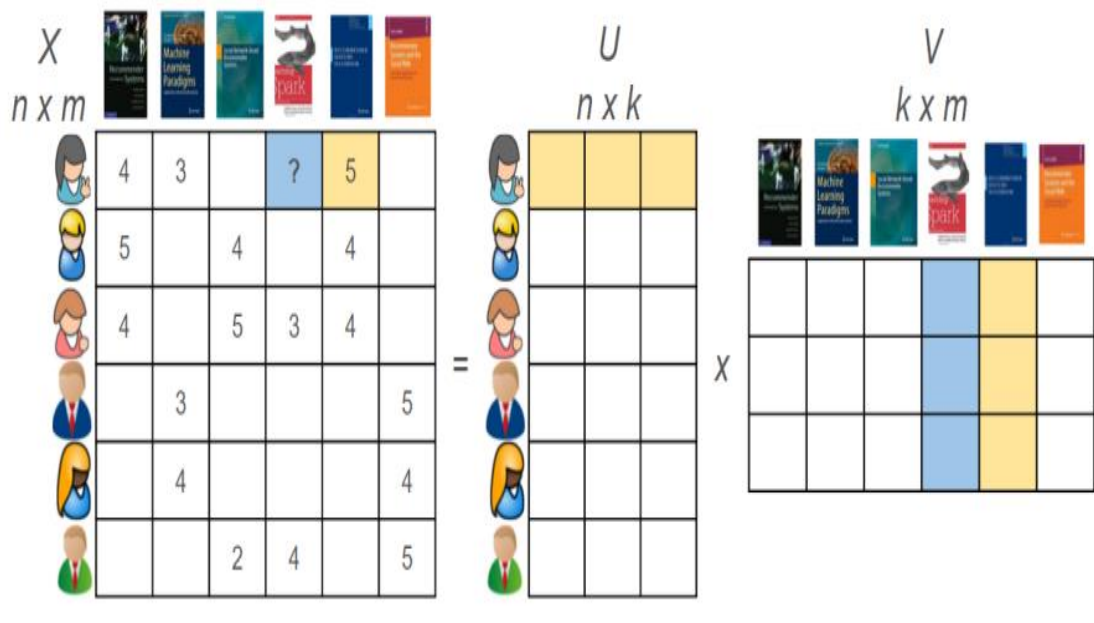
Content based filtering



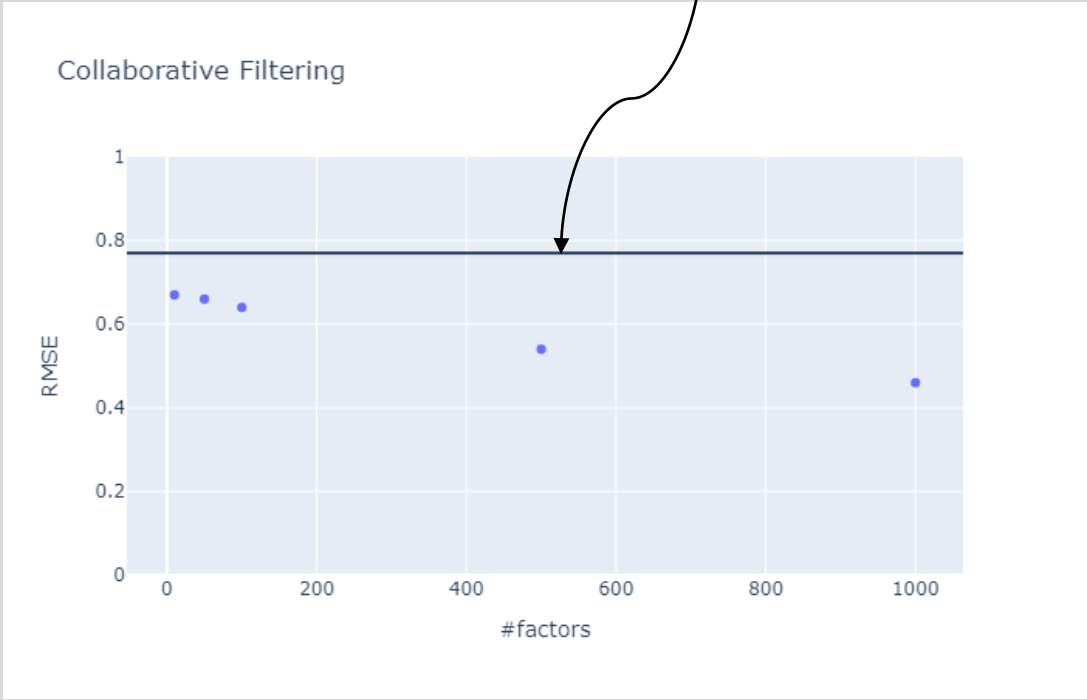
Content Based Filtering



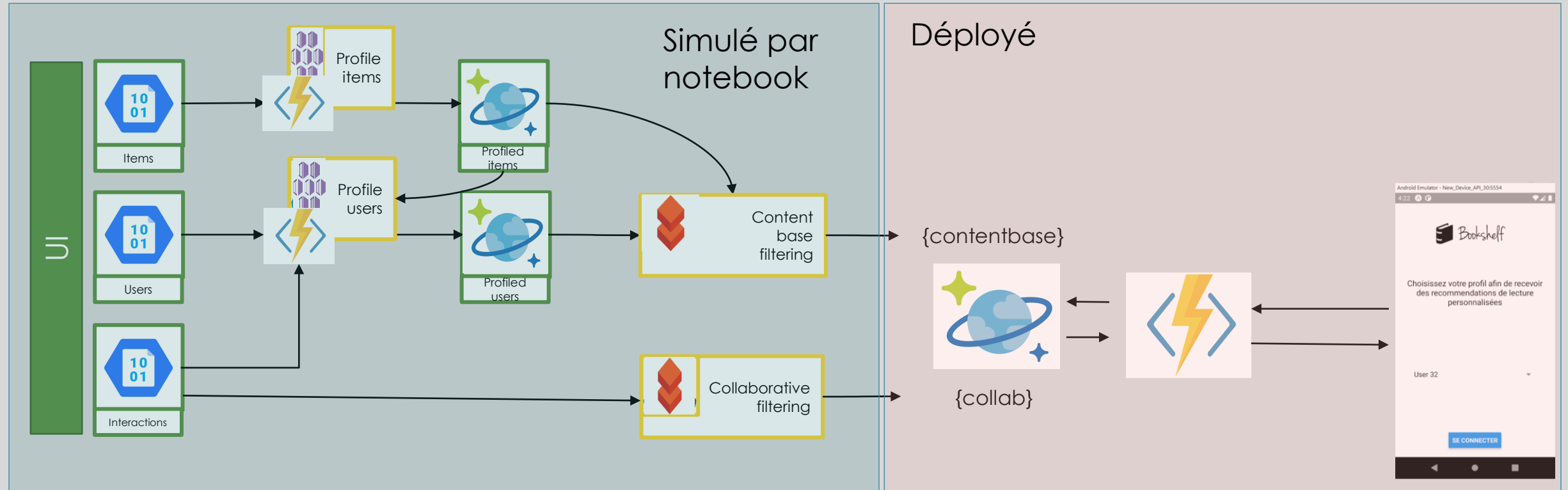
Collaborative filtering SVD








Baseline: erreur / score moyen

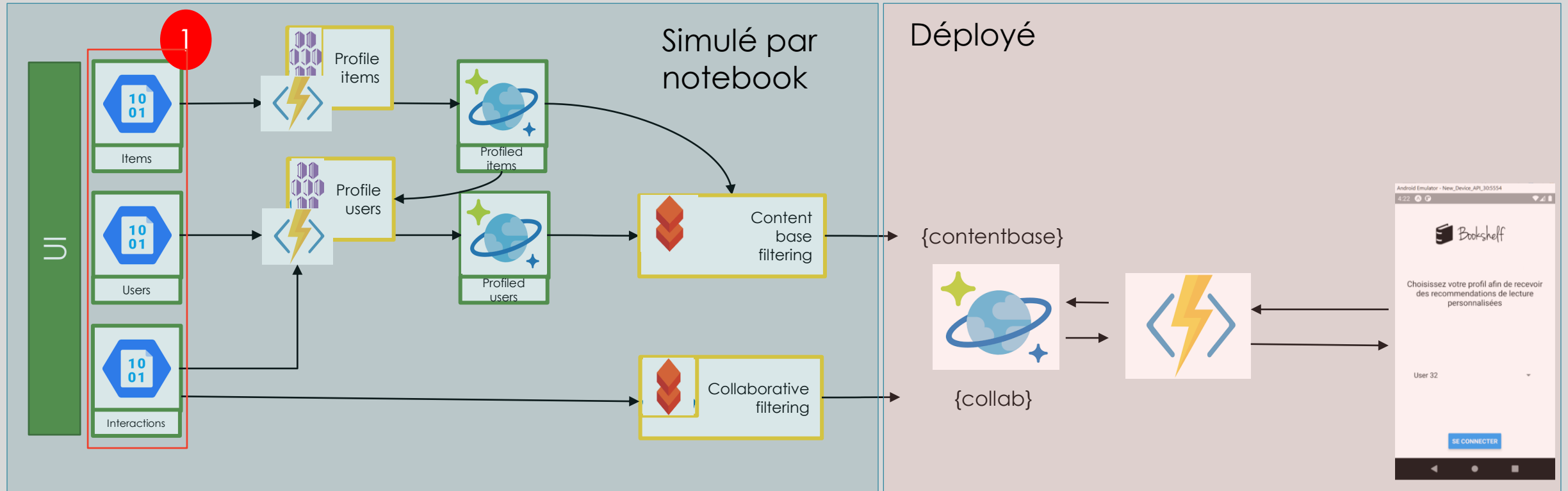


Architecture cible



-  Azure storage
-  Azure Kubernetes
-  Azure Databricks
-  Azure Function
-  Azure Cosmos DB

Architecture cible



Azure storage



Azure Kubernetes



Azure Databricks



Azure Function

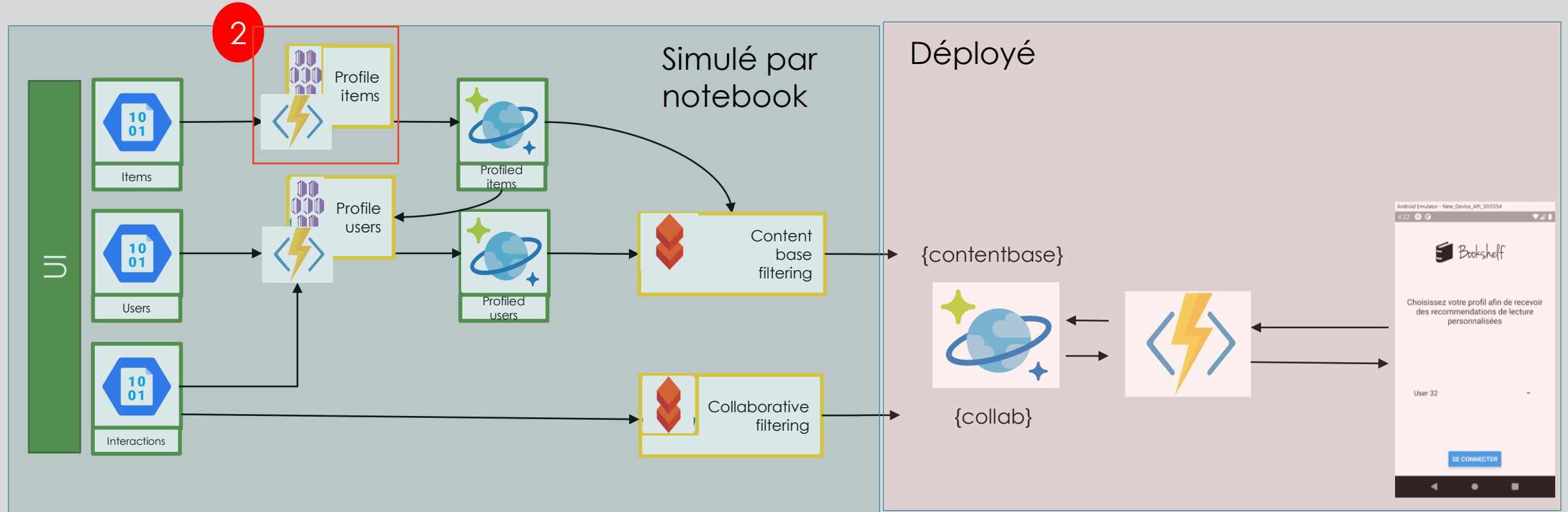







Azure Cosmos DB

1

Les 'items', 'users' et 'interactions' sont enregistrés dans un stockage type blob via des interfaces utilisateurs dédiées

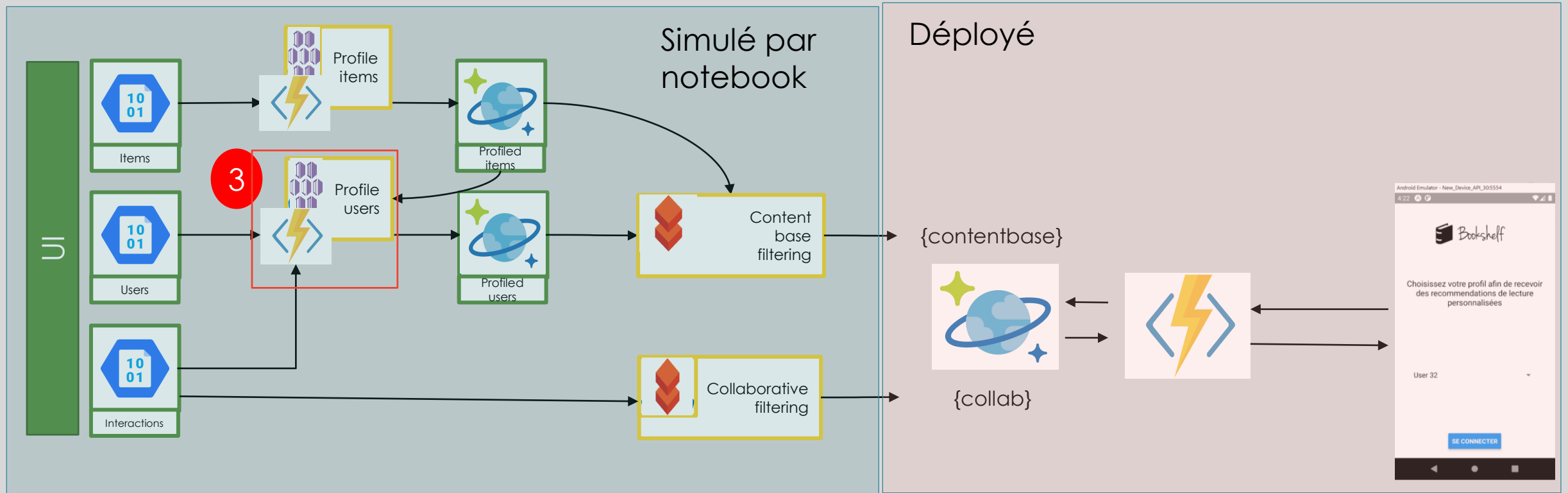
Architecture cible








-  Azure storage
-  Azure Kubernetes
-  Azure Databricks
-  Azure Function
-  Azure Cosmos DB

2 Une fonction serverless pilote un modèle de calcul de l'embedding des items. La fonction est déclenchée par l'enregistrement de nouveaux items dans le blob. Le modèle peut construire l'embedding à partir de techniques comme TF-IDF, Word2Vec, LDA.

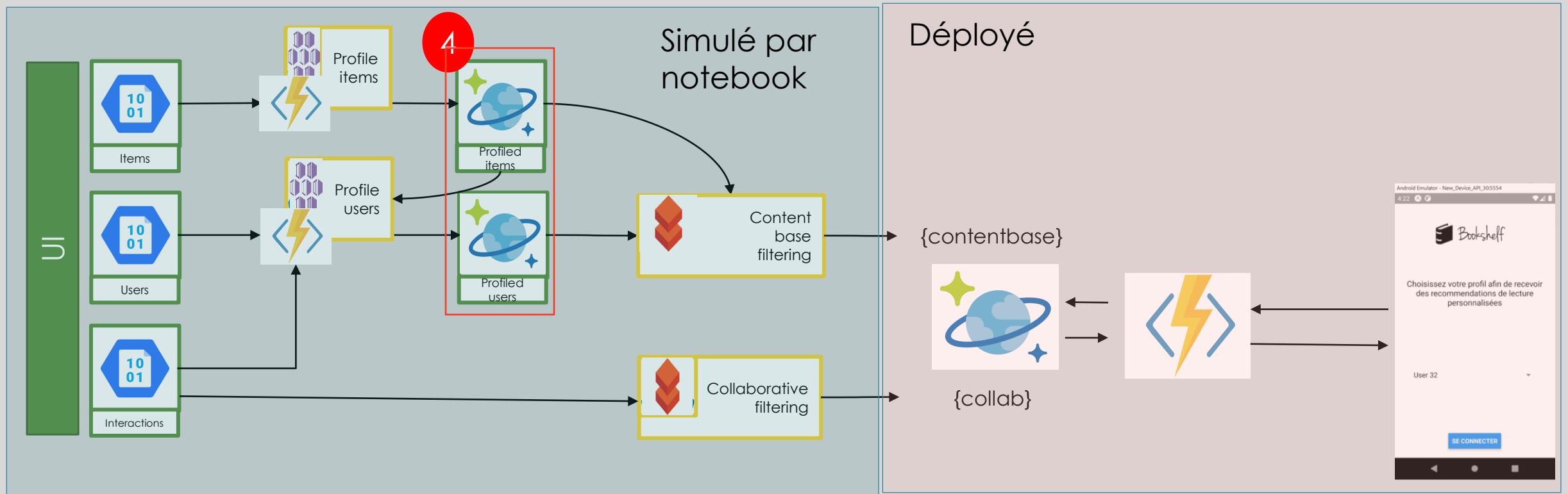
Architecture cible



-  Azure storage
-  Azure Kubernetes
-  Azure Databricks
-  Azure Function
-  Azure Cosmos DB

3 Une fonction serverless pilote un modèle de calcul du profil des utilisateurs. La fonction est déclenchée par l'enregistrement de nouveaux utilisateurs ou de nouvelles interactions dans le blob. Le modèle construit le profil comme la moyenne pondérée des items vus par l'utilisateur.

Architecture cible



Azure storage



Azure Kubernetes



Azure Databricks



Azure Function

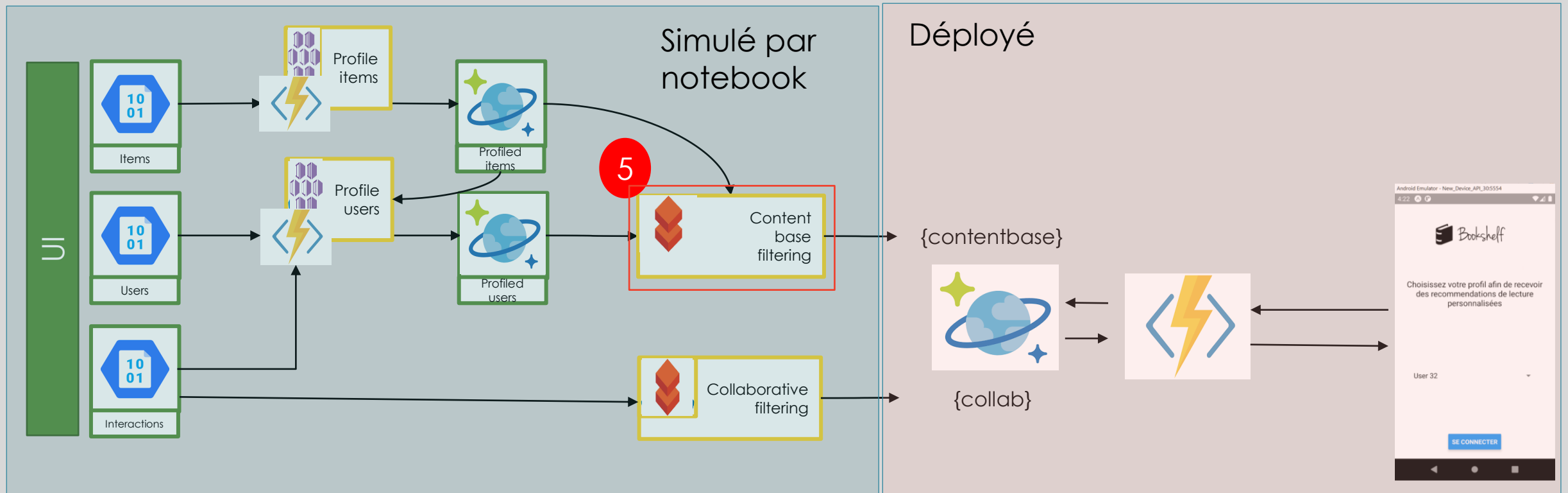







Azure Cosmos DB

4

Les embeddings des items et les profils utilisateurs sont stockés dans des bases de données CosmosDB.

Architecture cible

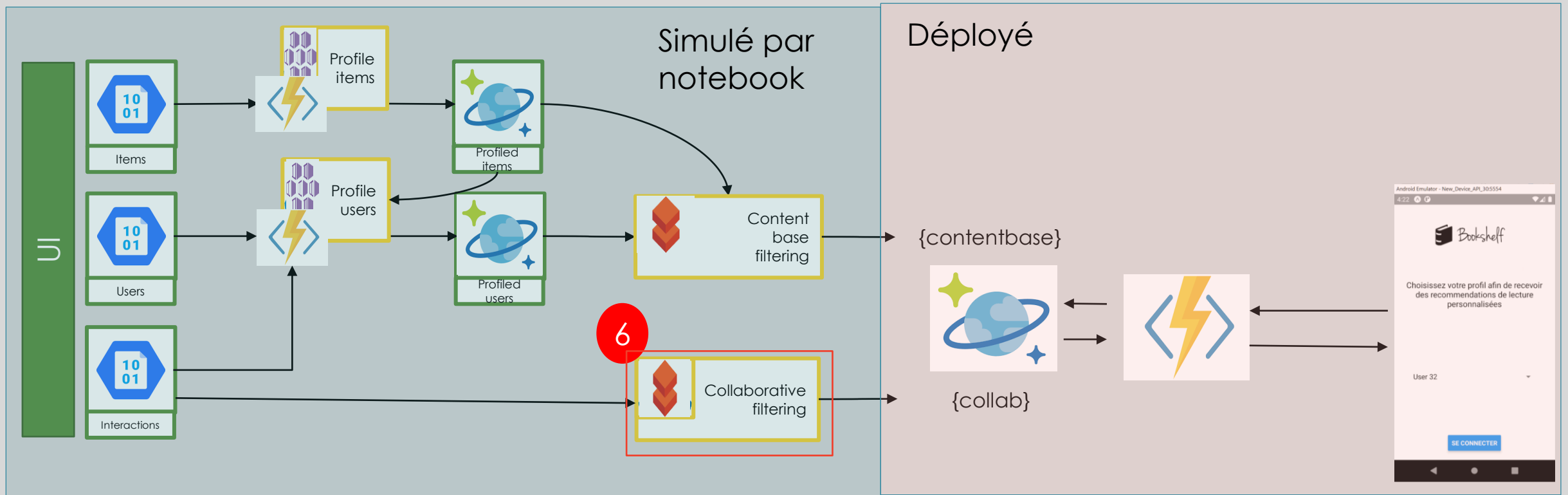







-  Azure storage
-  Azure Kubernetes
-  Azure Databricks
-  Azure Function
-  Azure Cosmos DB

5 Les recommandations basées sur le collaborative filtering sont calculées par batch. Les résultats sont stockés dans une base CosmosDB. Le calcul peut être distribué selon les utilisateurs.

NB: si un utilisateur n'a pas encore d'interactions avec le système, on peut retourner les préférences d'un utilisateur fictif 'proche'.

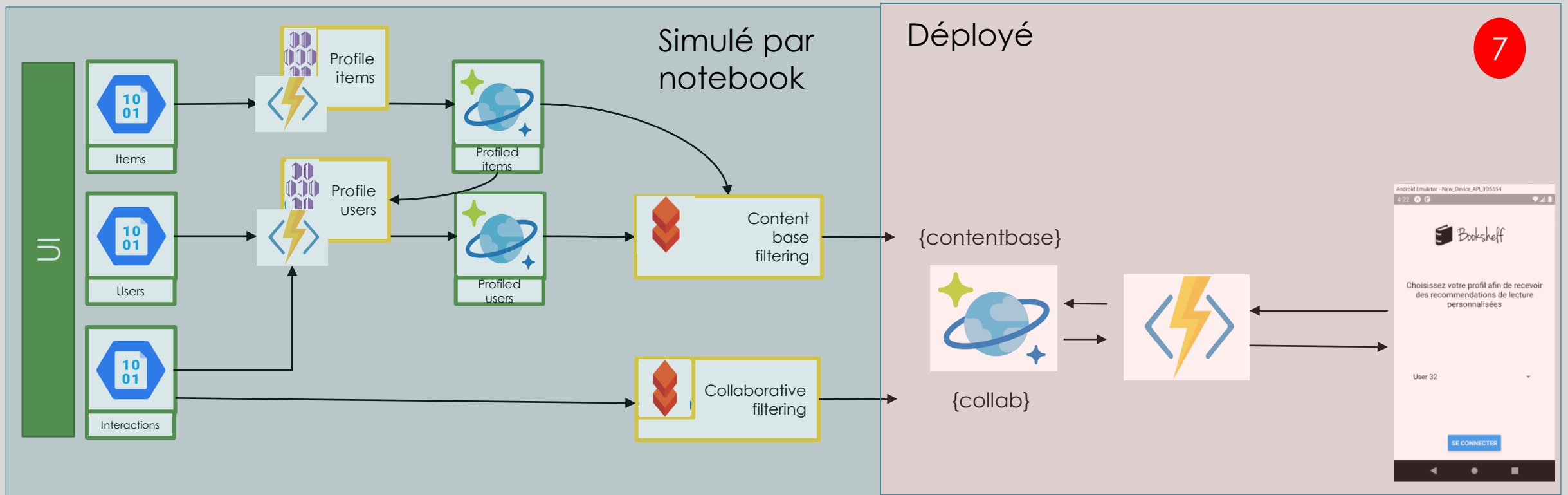
Architecture cible








-  Azure storage
-  Azure Kubernetes
-  Azure Databricks
-  Azure Function
-  Azure Cosmos DB

6 Les recommandations basées sur le collaborative filtering sont calculées par batch. Les résultats sont stockés dans une base CosmosDB. Le calcul peut être distribué selon des groupes d'utilisateurs.

Architecture cible



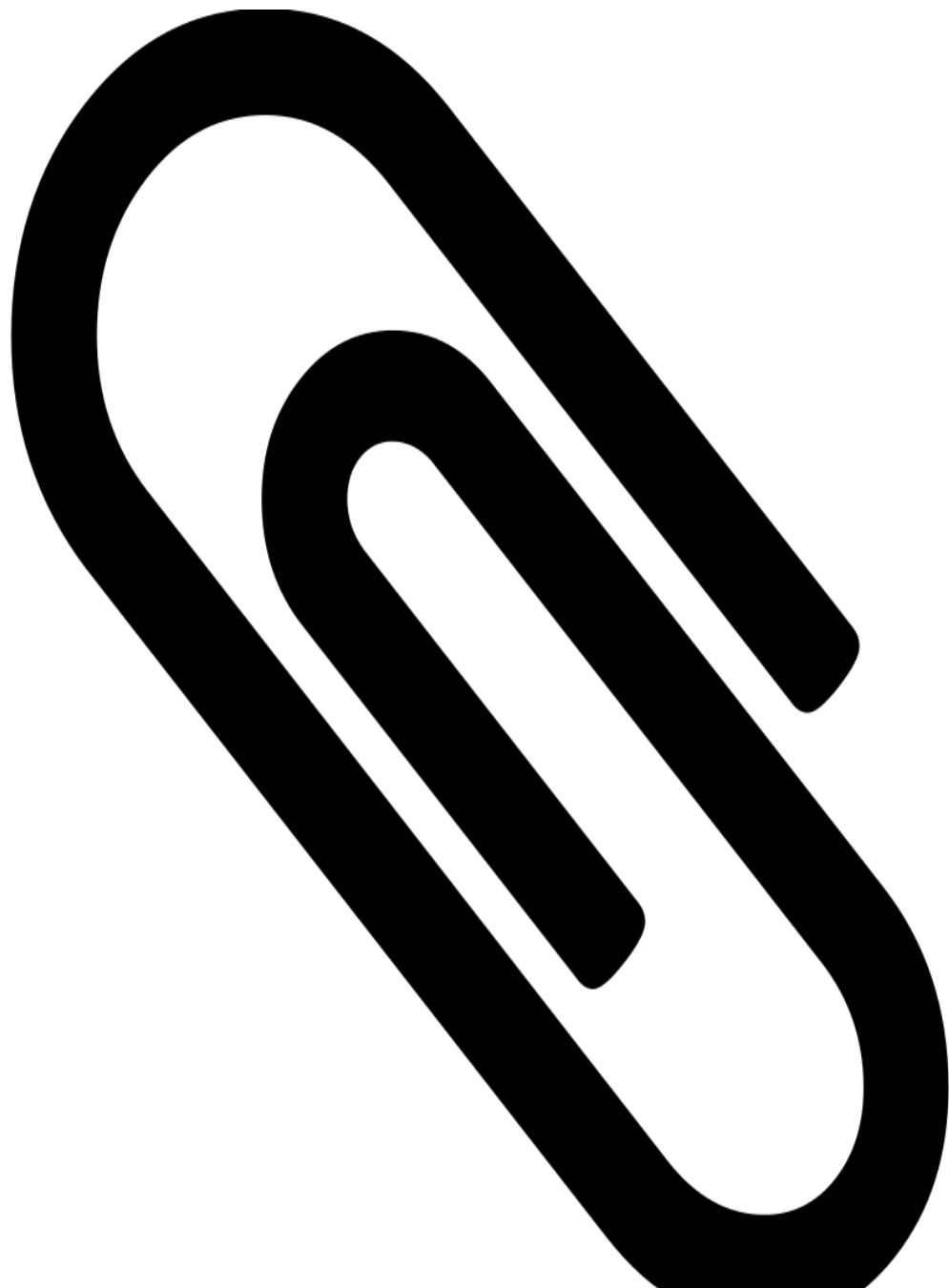
-  Azure storage
-  Azure Kubernetes
-  Azure Databricks
-  Azure Function
-  Azure Cosmos DB

7

Lorsqu'un utilisateur s'identifie sur son smartphone, une fonction serverless cherche les recommandations dans la base cosmosDB. S'il a plus de 10 interactions avec le système, la fonction cherche parmi les recommandations provenant du collaborative filtering, sinon du content based filtering.

A hand is shown in the upper right corner, holding a red marker and drawing a thick red oval on a white surface. The oval is nearly complete, with the marker tip just finishing the right side. In the center of the oval, the word "Démo !" is written in a black, handwritten-style font.

Démo !



Annexes