Bonjour à vous et bienvenue dans cette expérience inédite. Si cette expérience est un succès, cela pourrait un apport majeur au monde de l’architecture informatique. Nous allons essayer de faire un truc de génial, mais compliqué à mettre en œuvre.

Notre but ici est de créer une architecture permettant la synchronisation des données entre les différents utilisateurs d’une application. Nous allons permettre à tous les utilisateurs d’une application d’avoir sur leurs appareils toutes les données nécessaires à leurs usages, et permettra l’usage de l’application même lorsqu’elle est hors ligne.

## Problème à résoudre

Aujourd’hui, en ce qui concerne la disponibilité des données d’une application, il existe deux types d’applications, l’une qui est en ligne, avec un serveur central, et d’autre qui ne fonctionne qu’en local sans connexion internet.

L’inconvénient d’une application en ligne c’est qu’il faut constamment être connecté pour pouvoir l’utiliser, une instabilité de réseau rend son utilisation extrêmement laborieuse.

En ce concerne une application hors-ligne, ses données ne sont conservées sur l’appareil et ne sont nullement partagée sur l’ensemble du réseau. Si l’appareil est endommagé, les données sont perdues.

## Solution

La solution qui est proposé ici est de fusionner ces deux types d’applications.

Nous proposons ici une architecture qui permettra de stocker les opérations faites par un utilisateur sur son appareil, et une fois la connexion internet disponible, ces modifications seront envoyées sur le serveur et répliquées chez les autres utilisateurs.

Contrairement aux solutions actuelles de synchronisation de base de données, et qui nécessite donc une base de données par utilisateur, nous proposons ici une solution qui ne nécessite par cette charge de travail.

## Description

Le fonctionnement de l’architecture est la suivante :

1. L’utilisateur effectue une action (Par exemple : Il ajoute un produit),
2. L’action est normalement effectuée sur l’appareil,
3. L’action ensuite sauvegardé sous forme d’évènement,
4. L’évènement en envoyée sur le serveur,
5. Le serveur exécute l’action décrit par l’évènement (A savoir ajouter un produit)
6. Le serveur informe les autres périphériques concernés par l’action.
7. Les autres appareils récupèrent l’action et l’exécutent chez eux.

Voici les propriétés d’une action :

* Son nom (Ex : PRODUCT\_ADD) ce nom permet d’indiquer le type de l’action effectuée ;
* Ses données, ici les informations du produit,
* Sa date
* Son identifiant unique.

# Description générale

Voici les entités dont nous aurons besoin pour notre architecture :

### Actor

C’est l’utilisateur qui effectue une action. Lors de la création d’une utilisateur, on lui assignera également un actor.