

# Architecture : server-side

*Dernière modification : 12 fev 2020*

## Technologies

Nous avons choisi de stocker nos données avec **Firebase**, notamment avec les services suivants :

- **Firebase Authentication:** gestion des utilisateurs
- **Cloud Storage:** stockage des fichiers (photos par exemple)
- **Cloud Firestore:** stockage des données (base de données)

Voici les raisons qui ont menées à choisir **Firebase** :

- **Très bien intégré à Flutter via la suite de plugin [flutterfire](#).** Ces plugins qui permettent d'accéder aux services de Firebase sont maintenus par l'équipe de Firebase. Cela permet d'avoir des plugins mis à jour rapidement lors de changements de Firebase.
- **La simplicité d'utilisation :** nous avons une équipe réduite (2 personnes), aucun d'entre nous a déjà eu de l'expérience dans la mise en place / configuration de serveurs. Utiliser un système *serverless* permet de nous abstraire beaucoup de contraintes
- **La gratuité du service pour commencer :** pas besoin d'investir de l'argent, le plan gratuit permet de subvenir à nos besoins durant le développement
- **Familiarité avec le service :** nous avons déjà quelques connaissances de base de certains services Firebase
- **Mesures de qualités :** Firebase ne permet pas uniquement le stockage de données mais fourni aussi des services permettant d'avoir des mesures de qualités de notre applications (performances, crash, app distribution, etc)
- **Évolution :** Firebase fourni aussi d'autres services qui peuvent être utiles pour la suite : Google Analytics pour analyser le comportement de nos utilisateurs, A/B testing, etc.

## Données / structure

### Utilisateur

Voir `archi_models.md > #Users` pour davantage information à propos des utilisateurs.

### Connexion / inscription

*Firebase Authentication* est utilisé pour gérer la connexion des utilisateurs via les façons suivantes :

- email / mot de passe
- Facebook
- Google

## Données

Voici la structure noSQL des données des utilisateurs :

```
col:users
  doc:uid#1
    - name: text
    - age: integer
    - description: text
    - country: text
    - languages: text
    - photo: text
  doc:uid#2
  ...
  ...
```

Note: l'attribut `photo` des utilisateurs est une URL vers la photo de profile de l'utilisateur. Cette URL correspond à la photo stocké sur *Cloud Storage* voir ci-dessous.

## Photo

Les photos des utilisateurs sont stockées dans *Cloud Storage* dans le dossier `users_photos` .

TBD: Définir les limites de tailles d'image / compression / format / etc.

## Activité

*In progress with the sprint#4*