

EQUIPE F

Hadil AYARI

Chahan MOVSESSIAN

Floriane PARIS

Nicolas GUIBLIN

Rapport PolySnap

Rendu Final

Rendu du 05 novembre 2023

Tables des matières

Tables des matières	2
Analyse du besoin	3
Fonctionnalités du sujet	3
Définition du périmètre MVP	3
Diagramme d'architecture logicielle	5
Diagramme de déploiement cloud	6
Modèle de données	7
Collection conversation	7
Collection story	7
Collection user	7
Coûts prévisionnels	8
Google Cloud Pub/Sub	8
Google Cloud Storage (Bucket)	9
Google Cloud Function	10
Google Scheduler	10
Clever Cloud (Application)	11
MongoDB Atlas (Dedicated : Google Cloud Platform)	12
Présentation de la supervision envisagée du système	13
Faiblesses détectées	14

Analyse du besoin

Fonctionnalités du sujet

Fonctionnalité : Messagerie instantanée PolySmoke :

En tant qu'utilisateur, je souhaite pouvoir envoyer un message texte à un autre utilisateur afin de communiquer avec lui.

Priorité : Haute

Critères d'acceptation : Le message doit être livré.

En tant qu'utilisateur, je veux envoyer des photos dans mes messages afin de partager des moments visuels.

Priorité : Haute

Critères d'acceptation : Les photos doivent être uploadés et téléchargés.

En tant qu'utilisateur, je veux que certains de mes messages disparaissent après avoir été lus pour garantir la confidentialité.

Priorité : Haute

Critères d'acceptation : Les messages doivent disparaître du chat après le temps prédéfini.

Fonctionnalité : Conversation de groupe :

En tant qu'utilisateur, je souhaite créer des groupes de discussion afin de communiquer avec plusieurs personnes en même temps.

Priorité : Moyenne

Critères d'acceptation : Je dois pouvoir créer une conversation et communiquer avec les utilisateurs de celle-ci.

En tant que membre d'un groupe, je veux que mes messages disparaissent pour chaque lecteur indépendamment des autres.

Priorité : Moyenne

Critères d'acceptation : Les messages disparaissent pour chaque utilisateur après leur consultation individuelle.

Fonctionnalité : PolyStories :

En tant qu'utilisateur, je veux créer une séquence chronologique de médias (PolyStories) pour partager mon quotidien.

Priorité : Moyenne

Critères d'acceptation : Les stories doivent être visibles pendant X heures.

Définition du périmètre MVP

Après avoir défini les différentes fonctionnalités du sujet, nous avons décidé que notre Scope comprendrait “PolySmoke” et “PolyStories”, qui nous ont semblé être les fonctionnalités les plus intéressantes.

Notre MVP serait donc de pouvoir envoyer des textes et photos qui disparaissent après un temps défini.

Fonctionnalités minimales :

- On peut envoyer un message texte à un autre utilisateur afin de communiquer avec lui.
- On peut envoyer des photos afin de partager des moments visuels.
- Les messages disparaissent après avoir été lus pour garantir la confidentialité.

Diagramme d'architecture logicielle

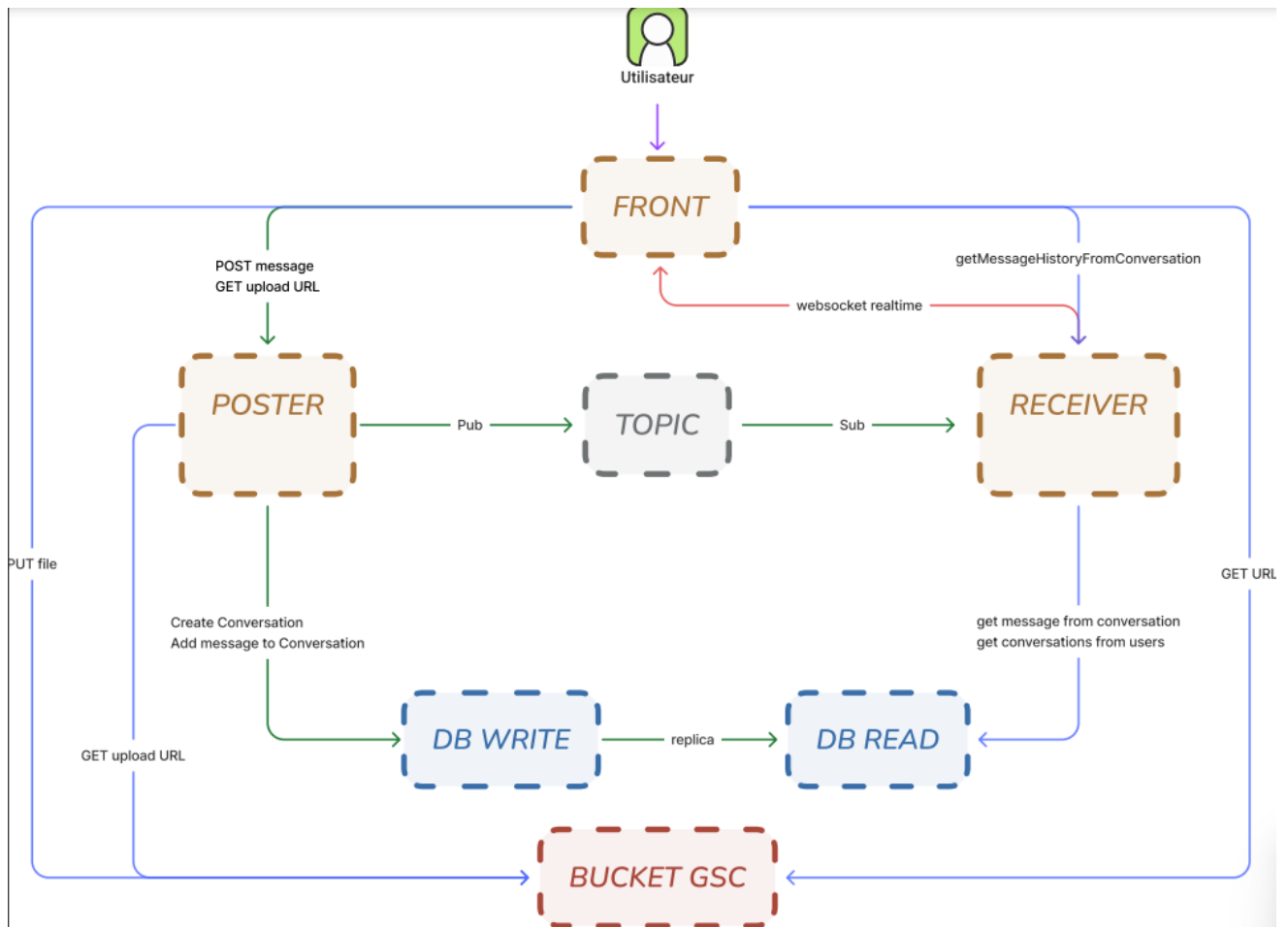
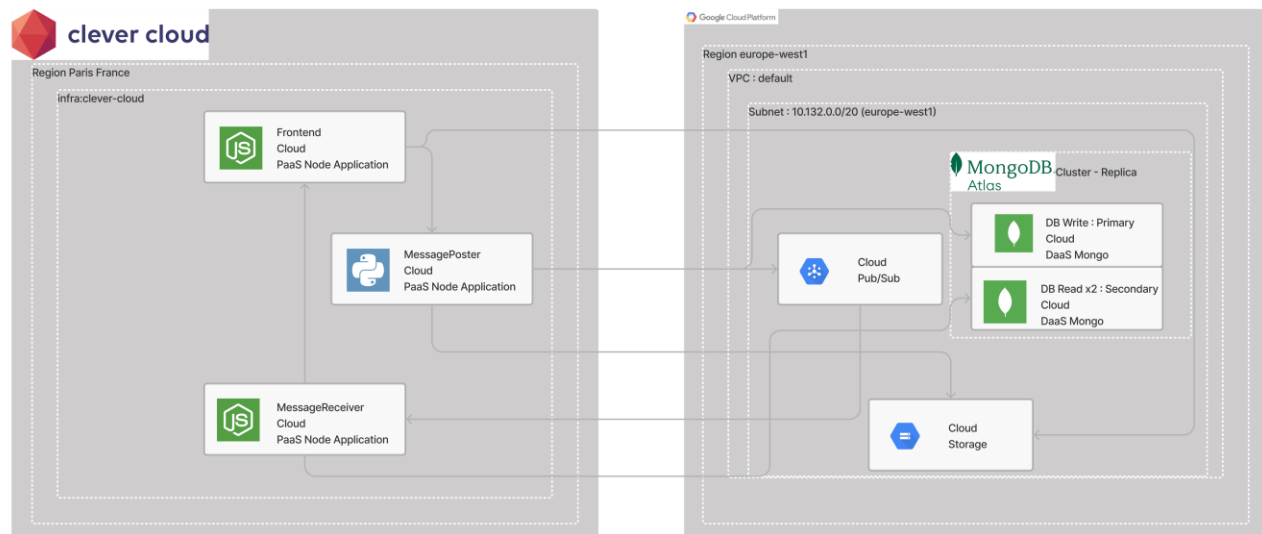


Diagramme de déploiement cloud



([Lien image diagramme déploiement cloud](#))

Modèle de données

Base de données MongoDB :

Collection conversation

- **_id** : Identifiant unique de la conversation (ObjectId).
- **name** : Nom ou titre de la conversation (String).
- **participants** : Liste des identifiants des utilisateurs participant à la conversation (Array de Strings).
- **messages** : Liste des messages dans la conversation (Array d'Objects), où chaque message a :
 - **_id** : Identifiant unique du message (ObjectId).
 - **conversation_id** : Identifiant de la conversation associée (ObjectId).
 - **sender** : Identifiant de l'utilisateur qui a envoyé le message (String).
 - **timestamp** : Horodatage de la création du message (String au format ISO 8601).
 - **text** : Contenu textuel du message (String).
 - **image** : URL de l'image associée au message, le cas échéant (String).
 - **read_by** : Liste des identifiants des utilisateurs qui ont lu le message (Array de Strings).
 - **smoke** : Indicateur si le message est éphémère (Boolean).
- **read_duration** : Durée après laquelle le message est supprimé après lecture, applicable si **smoke** est **true** (Integer).

Collection story

- **_id** : Identifiant unique de l'histoire (ObjectId).
- **user_id** : Identifiant de l'utilisateur qui a posté l'histoire (String).
- **image_url** : URL de l'image de l'histoire (String).
- **timestamp** : Horodatage de la création de l'histoire (String au format ISO 8601).
- **viewers** : Liste des identifiants des utilisateurs qui ont vu l'histoire (Array de Strings).
- **duration** : Durée de vie de l'histoire en minutes (Integer).

Collection user

- **_id** : Identifiant unique de l'utilisateur (ObjectId).
- **user_id** : Nom d'utilisateur ou identifiant (String).
- **conversations** : Liste des identifiants des conversations auxquelles l'utilisateur participe (Array d'ObjectIds).
- **my_stories** : Liste des identifiants des histoires postées par l'utilisateur (Array d'ObjectIds).
- **viewable_stories** : Liste des identifiants des histoires visibles pour l'utilisateur, y compris celles des autres utilisateurs (Array d'ObjectIds).

Coûts prévisionnels

Google Cloud Pub/Sub

Pour estimer les coûts associés à l'utilisation de Google Cloud Pub/Sub dans le cadre de l'application PolySnap avec 1 000 000 d'utilisateurs connectés, certaines hypothèses ont été formulées pour fournir une estimation basique. Les coûts de Pub/Sub peuvent être catégorisés en trois composants principaux : coûts de débit, coûts de stockage, et coûts de sortie. Voici l'approche adoptée pour cette estimation :

Hypothèses :

Nombre de messages par utilisateur par jour : On suppose que chaque utilisateur envoie en moyenne 10 messages par jour.

Taille moyenne d'un message : La taille moyenne d'un message est estimée à 200 octets (ce qui équivaut à 0.2 Ko), en prenant en compte le user_id, le texte du message, la date, et l'URL de l'image si elle est présente.

Calcul du débit :

Débit de publication quotidien = Nombre d'utilisateurs × Nombre de messages par utilisateur par jour
= 1 000 000 utilisateurs × 10 messages = 10 000 000 messages par jour.

Volume de données publiées quotidiennement = Débit de publication quotidien × Taille moyenne du message
= 10 000 000 messages × 0.2 Ko = 2 000 000 Ko = 2 Go.

Volume de données publiées mensuellement = Volume de données publiées quotidiennement × 30 jours
= 2 Go × 30 = 60 Go.

Coûts du débit :

Selon les tarifs fournis, les 10 premiers Go de débit par mois sont gratuits, puis le tarif est de 40 \$ par To.

Coût du débit mensuel = $(60 \text{ Go} - 10 \text{ Go}) \times (40 \text{ $/To}) = (50 \text{ Go}) \times (40 \text{ $/1024 Go}) \approx 1.95 \text{ $}$.

Coûts de stockage :

Supposons que vous configuriez un sujet pour conserver tous les messages, et que ces messages soient conservés pendant 7 jours avant d'être supprimés.

Stockage mensuel moyen = Volume de données publiées quotidiennement × 7 jours = 2 Go × 7 = 14 Go.

Coût de stockage mensuel = Stockage mensuel moyen × 0.27 \$/Go-mois = 14 Go × 0.27 \$ ≈ 3.78 \$.

Coûts de sortie :

Pour cette estimation, nous supposons que tous les utilisateurs et services consommateurs de messages résident dans la même région que votre service Pub/Sub, de sorte qu'il n'y ait pas de frais de sortie inter-régionaux.

Total des coûts mensuels :

Coût total mensuel ≈ Coût du débit mensuel + Coût de stockage mensuel = 1.95 \$ + 3.78 \$ ≈ 5.73 \$.

Google Cloud Storage (Bucket)

Pour estimer les coûts associés à l'utilisation de Google Cloud Storage dans le cadre de l'application PolySnap, des hypothèses ont été formulées concernant l'utilisation des images et des stories par les 1 000 000 d'utilisateurs en France. Voici l'approche adoptée pour cette estimation :

Hypothèses :

Nombre de messages par utilisateur par jour : On suppose que chaque utilisateur envoie en moyenne 10 messages par jour.

Fréquence des images : On suppose qu'il y a une image tous les 10 messages, soit 10% des messages contenant une image.

Taille moyenne d'une image : La taille moyenne d'une image est estimée à 2 Mo.

Nombre de stories par utilisateur par jour : On suppose que chaque utilisateur publie en moyenne 2 stories par jour, chaque story contenant une image.

Calcul du stockage pour les images dans les messages :

Nombre total de messages contenant une image : $1,000,000 \times 10 \times 0.1 = 1,000,000$ messages

Stockage total nécessaire pour ces images : $1,000,000 \times 2\text{Mo} = 2,000,000\text{Mo}$ soit 2,000 Go

Calcul du stockage pour les stories :

Stories par jour : $1,000,000 \times 2 = 2,000,000$ stories

Stockage nécessaire par jour pour les stories : $2,000,000 \times 2\text{Mo} = 4,000,000\text{Mo}$ soit 4,000 Go

Stockage nécessaire par mois pour les stories : $4,000 \times 30 = 120,000\text{ Go}$ $4,000 \times 30 = 120,000\text{Go}$

Stockage total par mois :

Stockage total mensuel = Stockage pour les images dans les messages + Stockage pour les stories = $2,000\text{ Go} + 120,000\text{ Go} = 122,000\text{ Go}$

Coûts du stockage :

Selon les tarifs fournis pour la Belgique (Europe-west1) de 0.020\$ par Go et par mois, le coût mensuel de stockage sera : $122,000 \times 0.020 = 2,440\$$

Total des coûts mensuels :

Coût total mensuel pour le stockage \approx 2,440\$

Google Cloud Function

Pour estimer les coûts associés aux fonctions cloud utilisées dans l'application PolySnap pour la suppression des messages et des stories expirés, une fonction nommée `delete_stories_expired` et une autre appelée `delete_messages_expired` sont exécutées quotidiennement. Ces fonctions sont hébergées dans la région de la Belgique (europe-west1) sur Google Cloud. Voici l'approche adoptée pour cette estimation :

Hypothèses :

Invocations : Chaque fonction est invoquée une fois par jour, soit 31 fois par mois.

Durée d'exécution : Nous avons alloué 10 minutes (600,000 ms) pour la durée d'exécution de chaque fonction, bien que cette estimation puisse nécessiter des ajustements basés sur les performances réelles.

Ressources allouées : Pour chaque fonction, 128MB de RAM et 200MHz de CPU sont alloués.

Coûts :

CPU : Le coût du CPU pour la fonction `delete_stories_expired` est de 1.12 USD par mois.

Mémoire : Il n'y a pas de coût associé à l'utilisation de la mémoire dans l'estimation fournie.

Coût de la bande passante réseau :

Réseau : Le coût de la bande passante réseau pour la fonction `delete_stories_expired` est négligeable et est estimé à 0.00 USD.

Coût des instances minimales :

Instances Minimales : Un coût de 20.81 USD par mois est associé au nombre minimum d'instances pour la fonction `delete_stories_expired`.

Le coût total estimé pour la fonction `delete_stories_expired` est de 21.93 USD par mois. Comme les configurations et les exigences de la fonction `delete_messages_expired` sont similaires, nous estimons également son coût à 21.93 USD par mois.

Total des coûts mensuels :

Coût total mensuel pour les deux fonctions cloud $\approx 21.93 \text{ USD} + 21.93 \text{ USD} = \mathbf{43.86 \text{ USD}}$.

Google Scheduler

Google Cloud Scheduler est utilisé pour déclencher les deux fonctions cloud mentionnées précédemment sur une base quotidienne. Voici les détails relatifs à l'utilisation de Google Cloud Scheduler dans le cadre du projet PolySnap :

Configuration :

Nombre de jobs : 2 (un pour déclencher la fonction `delete_stories_expired` et un autre pour déclencher la fonction `delete_messages_expired`)

Fréquence : Chaque job est programmé pour s'exécuter une fois par jour.

Coûts : Le coût associé à l'utilisation de Google Cloud Scheduler pour ces deux jobs est de 0.00 USD, car cette utilisation entre dans la tranche gratuite offerte par Google Cloud.

Clever Cloud (Application)

Dans cette section, nous estimons les coûts associés à l'hébergement de trois applications sur Clever Cloud, en tenant compte de l'auto-scalabilité entre les tailles d'instances XS à L, et avec un maximum de 40 machines pour la scalabilité horizontale.

Hypothèses :

Heures pleines (9h - 21h) : Nous supposons que l'usage de l'application est au plus haut durant ces heures, nécessitant le nombre maximal de machines.

Heures creuses (21h - 9h) : Pendant ces heures, nous supposons que l'usage de l'application est réduit, nécessitant moins de machines.

Nous avons calculé les coûts quotidiens et mensuels pour chaque application, en utilisant les coûts quotidiens estimés pour chaque plan (XS, S, M et L), obtenus en divisant les coûts mensuels par 30.

Les coûts quotidiens estimés sont les suivants :

XS : 0,625 \$

S : 1,25 \$

M : 2,97 \$

L : 5,94 \$

Calcul du coût pour chaque application :

Heures Pleines :

Nombre de machines : 40

Durée : 12 heures par jour

Heures Creuses :

Supposons que seulement 10 machines sont nécessaires pendant les heures creuses.

Durée : 12 heures par jour

Nous avons ensuite calculé le coût total par jour pour chaque application, en additionnant les coûts des heures pleines et des heures creuses. Ce total a été multiplié par 30 pour obtenir le coût total par mois pour chaque application.

Calcul du coût total pour les trois applications :

Les coûts totaux mensuels estimés pour les trois applications, pour chaque plan, sont les suivants :

Plan XS :

Coût total par mois pour les trois applications : 2,812,50 \$

Plan S :

Coût total par mois pour les trois applications : 5,625 \$

Plan M :

Coût total par mois pour les trois applications : 13,365 \$

Plan L :

Coût total par mois pour les trois applications : 26,730 \$

Résumé des Coûts pour Clever Cloud (Application) :

En tenant compte des différents plans et des scénarios d'usage en heures pleines et creuses, les **coûts totaux mensuels estimés** pour les trois applications varient entre **≈ 2,812,50 \$ (Plan XS) et ≈ 26,730 \$ (Plan L)**.

Configuration Actuelle sur le Cloud :

Dans la configuration actuelle sur le cloud pour le projet, nous avons opté pour un scaling vertical et horizontal plus restreint afin de contrôler et minimiser les coûts. Cette décision permet d'aligner les ressources disponibles avec les exigences actuelles du projet tout en conservant une marge pour la gestion des pics d'activité, sans engendrer les coûts associés à une scalabilité plus élevée comme illustré dans les estimations ci-dessus.

MongoDB Atlas (Dedicated : Google Cloud Platform)

Pour la partie stockage des messages en base de données de l'application PolySnap sur MongoDB Atlas (en cluster dédié sur GCP), nous avons estimé la configuration nécessaire pour stocker les messages pendant 30 jours. En se basant sur le volume de données généré comme indiqué dans la section Google Cloud Pub/Sub, le volume total des données à stocker est de 60 Go par mois.

La configuration minimale qui répond à notre besoin en termes de stockage est le M40 avec 80 GB de stockage, 16 GB de RAM et 4 vCPUs, avec un coût de base de \$0.86/hr, soit environ \$619.20 par mois.

Cependant, pour assurer une meilleure sécurité et anticiper une éventuelle croissance du nombre d'utilisateurs et donc une augmentation du volume de messages, nous optons pour la configuration M50 qui offre 160 GB de stockage, 32 GB de RAM et 8 vCPUs, avec un coût de base de \$1.66/hr, soit environ \$1195.20 par mois.

Coût mensuel = 1.66 USD/hr × 24 heures/jour × 30 jours/mois = 1195.20 USD/mois

Ce choix nous permet non seulement de couvrir nos besoins actuels en termes de stockage et de performance, mais aussi de nous prémunir contre d'éventuelles augmentations de la charge de travail sur notre base de données.

Présentation de la supervision envisagée du système

Nous avons conçu l'architecture du projet afin de pouvoir gérer le plus d'accidents possible, car on sait qu'ils sont inévitables. Par exemple, le réplica de la base de données permet non seulement de ne pas la surcharger, mais également de retrouver les informations à plusieurs endroits en cas de problèmes. De même, si les bases de données sont hors services, on ne peut pas retrouver l'historique des messages cependant les messages envoyés sont quand même reçus grâce à un topic.

Il est essentiel de bien superviser l'application Polysnap afin de détecter et résoudre rapidement les problèmes de performance, assurer la sécurité des données, et optimiser les coûts d'exploitation.

Avec Google Cloud et Clever Cloud, on peut mettre en place des outils de surveillance des performances, et des tableaux de bord de suivi pour surveiller la disponibilité du service, les temps de réponse, l'utilisation des ressources, la consommation de bande passante et mettre en place des alertes afin de pouvoir réagir le plus rapidement possible.

Il est important de mettre en place des processus, des outils et des métriques pour maintenir la qualité du service offert aux utilisateurs.

Des tests de charges simulant une centaine d'utilisateur simultanés qui échangent un total d'environ 25000 messages envoyés ont été réalisés avec grafana k6.

Faiblesses détectées

Connexion WebSocket

Nous avons identifié une faiblesse concernant la gestion des états de la connexion WebSocket dans notre application. La connexion tend à être instable, ce qui pourrait affecter l'expérience utilisateur en temps réel lors de l'utilisation des fonctionnalités de messagerie instantanée. Nous suspectons que cette instabilité est due à une mauvaise gestion des états de la connexion.

Accès à l'app

-- **Pour tester l'application, accédez au frontend ici (<https://app-061c7eb9-4e4d-4bff-a3ba-ac5f184e2f25.cleverapps.io/>)**. Entrez un nom d'utilisateur (un existant parmi Sylcantor, Chahan, Testeur ...) ou un nouveau nom.