

# CaaS - Conversion as a Service

---

## *Storage Architecture*

**Mercredi 01 novembre 2017**

**SACC**

**Étudiants :**

BELHASSEN Issam M2IFI AL

THIERNO Balde M2IFI AL

NAAMANE Ilias M2IFI WEB

EL MRIHY Mohamed M2IFI WEB

### **Plan**

<b>Plan</b>	<b>1</b>
<b>I.Introduction</b>	<b>2</b>
<b>II.Architecture de la couche donnée</b>	<b>2</b>
1.Schéma	2
2.Cloud Storage	3
3.Data Store	5
<b>III.Diagramme de séquence</b>	<b>7</b>

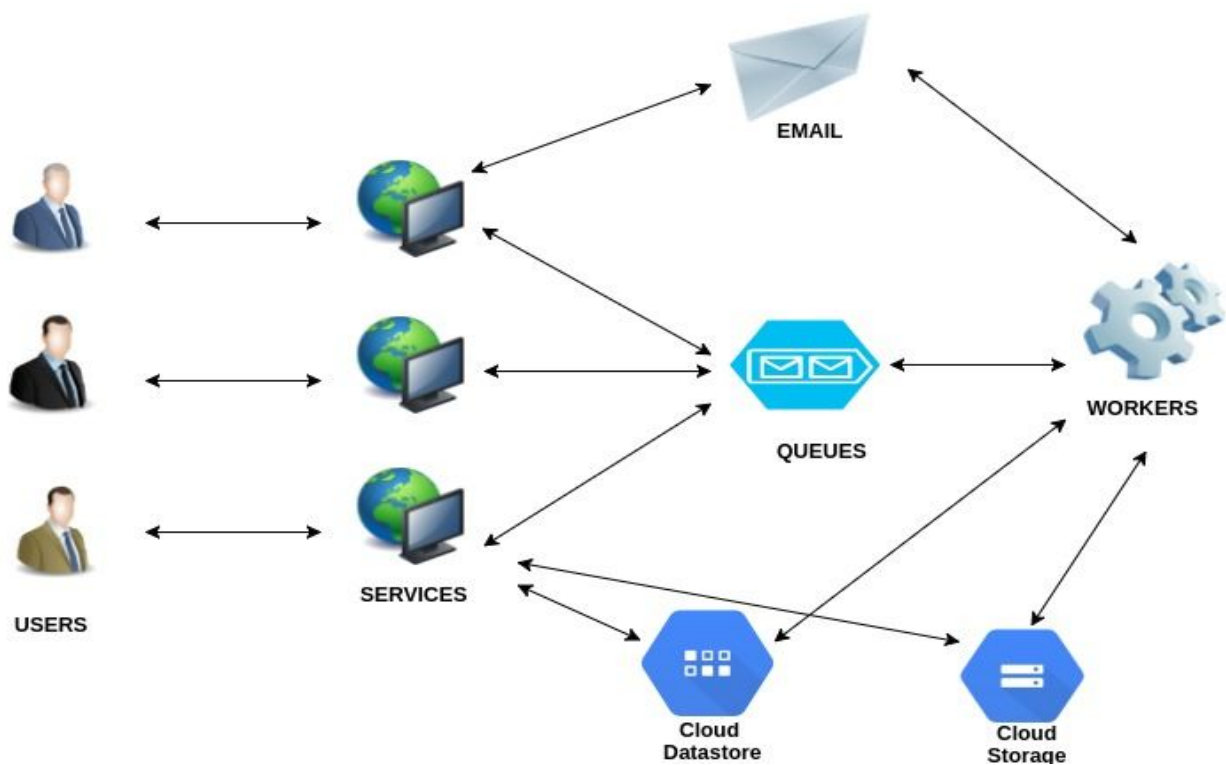
## I.Introduction

Ce rapport décrit l'architecture de la couche donnée pour le projet CaaS (Conversion as a Service). La description de composants et technologies de stockage de Google Cloud PLaform et les justifications de ces choix et l'estimation de leurs coûts et des cas d'utilisations.

## II.Architecture de la couche donnée

### 1.Schéma

Dans ce schéma nous présentons les différents composants de l'architecture tels que les services de stockage de Google Cloud Platform(Cloud Datastore et Cloud Storage) et leurs interactions avec les autres composant de notre système.



*figure 1 : Architecture cloud de projet (data and components)*

Nous avons deux principaux composants de stockage qui permettent de :

- Stocker les informations utilisateurs tel type mail, identifiant de l'utilisateur, la qualité de service (bronze, silver, gold) liées à la demande de conversion, les données des vidéos c'est à dire le nom et la durée en seconde et l'état des tâches sur le service.

=> Ce type de stockage nécessite peu d'espace par entité mais il est nécessaire de pouvoir accéder et modifier rapidement les données. Nous n'avons pas besoin d'une consistance forte concernant ces données.

- Stocker les vidéos avant la conversion, pendant la conversion et durant un certain temps jusqu'à soit la récupération de cette dernière ou expiration.

=> Ce type de stockage nécessite le stockage d'un volume conséquent de manière fiable pour un intervalle de temps (max un temps de stockage de 30mn)

Google offre plusieurs technologies de stockage, on peut citer Google Cloud Storage, Google Cloud Datastore, Cloud SQL, Cloud BigTable. Donc c'est à nous de faire le choix entre ces technologies. nous avons fait le choix d'utiliser la solution Cloud Datastore pour le stockage des entités semi-structuré et la solution Cloud Storage pour le stockage des vidéos.

## **2.Cloud Storage**

Google Cloud Storage est une solution unifiée de stockage d'objets de manière scalable tels que les objets de type binary (les vidéos dans notre projet). Il propose quatre classes de stockage: le stockage multirégional, le stockage régional, le stockage nearline et le stockage Coldline.

Notre choix doit prendre en considération l'accès fréquenté à nos vidéos et ce qui poursuit en terme de coût et de débit et surtout de disponibilité.

Toutes les classes de stockage offrent le même débit, une faible latence (délai de premier octet généralement de quelques dizaines de millisecondes) et une grande durabilité. Les classes diffèrent par leur disponibilité, leur durée de stockage minimale et leur prix de stockage et d'accès.

Nous avons choisie l'offre regional storage qui permet de stocker nos données à moindre coût par rapport aux autres, le compromis des données étant stocké dans un emplacement régional spécifique, au lieu d'avoir une redondance distribuée sur une grande zone géographique. Cet emplacement régional est le même que les instances Google App Engine qui utilisent les données. Cela nous permet d'obtenir de meilleures performances pour l'utilisation intensive de données, par opposition au stockage de nos données dans un emplacement multirégional. En outre, le stockage de nos données en tant que stockage régional dans ce scénario peut réduire les frais de réseau et permet d'avoir une haute disponibilité (99,9% de disponibilité SLA) et du coût inférieur par GB stocké et des données stockées dans une région géographique étroite.

En terme de pricing l'offre Région se situe autour 0.02\$/GB (le prix pouvant différer suivant la région choisie).

## Estimation des coûts pour le Cloud Storage :

Pour le calcul des coût des vidéos nous allons les déterminer par qualité de service et en fonction de la durée de la vidéo.

Notre choix étant le régional storage, nous prenons comme exemple la région d'Europe de l'ouest ( Belgique) et le mensuel comme repère de temps.

1 seconde de durée (s) = 1 megabytes (MB) .

Pour chaque demande de conversion nous effectuerons trois opérations sur le cloud storage , une pour stocker la vidéo(POST), une pour le téléchargement(GET) une fois que la conversion est terminée et une pour supprimer la vidéo après l'expiration du temps de disponibilité de vidéo pour le téléchargement (la taille de la vidéo est calculé par rapport à la durée de la vidéo 1s = 1MB)

- **Service Bronze :**

Nous prenons le maximum que peut offrir ce service qui est de 60 MB (s)

Hypothèses:

- 800 conversions de type bronze par jour équivalent à 24000 conversions par mois
- Durée de transcodage est de 108s
- taille de la vidéo: 60MB
- On considère que la taille de la vidéo après transcodage est la même

Ceci dit qu'on doit réserver 60MB de stockage pour la vidéo à convertir et 60 MB de stockage pour la vidéo converti. La durée de transcodage est de 108s , donc une fois le transcodage finit, la vidéo original laisse la place pour une nouvelle vidéo pour être transcodé tout en conservant la vidéo précédente converti pendant 5minute. Donc on doit prévoir un espace de stockage de plus qui dans ce cas sera suffisant pour pouvoir stocker la nouvelle vidéo converti.

### Conclusion:

- 60 MB stockage de vidéo à convertir
- 60 MB stockage de la vidéo converti pendant 5 minutes
- 120 MB stockage reserve

Total espace stockage nécessaire pour les bronzes: 240 MB.

Nombre d'opérations GET; PUT; POST:  $24000 \times 4 = 96000$  ( 24000 conversions avec 4 opérations pour chaque conversion  $\Rightarrow$  une POST pour stocker la vidéo à convertir, GET pour prendre la vidéo à convertir de la part du Worker, POST pour stocker la vidéo après conversion et GET pour prendre la vidéo converti )

Cout\_total ( mensuel ) = \$0.48/month

- **Service Silver :**

Pour ce service nous avons plus la restriction sur la durée des vidéos, alors calculer le coût de ce service nous considérons les hypothèses suivantes:

Hypotheses:

- Nombre de vidéos: **4500**
- Taille de vidéo: 300MB(s)
- Durée de transcodage: **500s**
- Taille de la vidéo après transcodage est la même

On a 3 conversions en parallèle, donc :

- ❖ 900 MB pour le stockage des vidéos à convertir
- ❖ 900 MB pour le stockage des vidéos converti ( conservé 5 minutes )
- ❖ 4 opération pour chaque 600MB (Explication auparavant pour les bronzes )

donc il est à prévoir 1800 MB avec 12 opérations en total pour chaque pack de 3 conversion

Nb\_operations (mensuel ) =  $12 * 1500 = 18000$

Taille total: 1800 MB

Cout\_total ( mensuel ) = **\$0.13/month**

- **Service Gold:**

Pour ce service nous avons plus la restriction sur la durée des vidéos, alors calculer le coût de ce service nous considérons les hypothèses suivantes:

Hypotheses:

- Nombre de vidéos: **3000**
- Taille de vidéo: 500MB(s)
- Durée de transcodage: **500s**
- Taille de la vidéo après transcodage est la même

On a 5 conversions en parallèle, donc :

- ❖ 2500 MB pour le stockage des vidéos à convertir
- ❖ 2500 MB pour le stockage des vidéos converti ( conservé 5 minutes )
- ❖ 4 opération pour chaque 1000MB (Explication auparavant pour les bronzes )

donc il est à prévoir 2500 MB avec 20 opérations en total

Nb\_operations (mensuel ) =  $20 * 3000 = 60000$

Taille total: 5000 MB

Cout\_total ( mensuel ) = **\$0.40/month**

### 3.Data Store

Google Cloud Datastore est une base de données de documents NoSQL conçue pour la mise à l'échelle automatique, la haute performance et la facilité de développement d'applications, il est idéal pour les applications qui reposent sur des entités semi-structurés de taille variable hautement disponibles à grande échelle.

Nous allons utiliser Cloud Datastore pour stocker et interroger nos différents types de données, tels que les informations de type mail, identifiant de l'utilisateur, la qualité de service (bronze, silver, gold) liées à la demande de conversion, et l'état des conversions associé à chaque utilisateur (Pending, Queued, Processing, Terminated).

Il est important que le résultat d'une lecture à un instant  $t(i)$  (ex: l'utilisateur x a une conversion en cours), soit visible pour n'importe quelle lecture à un instant  $t(i+1)$ , cela bien sûr pour garantir les SLA imposés pour les différents types de compte, car dans le cas contraire, un utilisateur peut avoir 2 conversions en parallèle parce que pendant la deuxième conversion, on a pas l'information qu'il y en a une en cours, l'Eventual Consistency n'est pas suffisante dans ce cas. Nous avons besoin de mettre une parenté entre les entités qui permet de faire des requêtes cohérentes, cette solution est de relier chaque utilisateur avec sa tâche en créant une Entity, donc les lectures sur ses entités sont des strongly-consistent reads à travers des ancestry queries.

Strongly-consistent reads renvoie toujours les données actuelles et, si elles sont exécutées dans une transaction, semble provenir d'un instantané unique et cohérent, par conséquent, toute opération d'update, de suppression ou de création aura lieu au sein d'une transaction. Chaque transaction est garantie atomique, ce qui signifie que les transactions ne sont jamais partiellement appliquées. Soit toutes les opérations de la transaction sont appliquées, soit aucune d'entre elles n'est appliquée.

#### Liste d'entités :

- **User :**
  - Long id : l'id de l'utilisateur
  - String identifiant : l'identifiant de l'utilisateur
  - String mail : l'email de l'utilisateur
  - Offer offer : l'offre de l'utilisateur (BRONZE, SILVER or GOLD)
- **Task :**
  - Long id : l'id de la tâche
  - User user : l'utilisateur associé //lien de parenté
  - String video\_original : la vidéo original
  - String video\_converted : la vidéo convertie
  - String format : le format de la vidéo
  - State state : l'état de la tâche (PENDING, RUNNING or DONE)

#### Estimation des coûts :

On estime que notre application permet de gérer 1 million d'utilisateurs et on suppose que notre application a déjà effectué 1 million de conversions. Donc l'estimation du coût mensuel dépend de la taille de stockage des utilisateurs et les informations concernant les conversions et le nombre de READ, WRITE et DELETE concernant ce stockage.

#### Hypothèses:

- Les informations des utilisateurs sont stockées sur environ 1 KB/utilisateur avec un entity read de 10 million/mois et entity write de 1000/mois et entity delete de 500/mois
- Les informations concernant l'état de conversion (Pending, Queued, Processing, Terminated) sont stockées sur environ 1 KB/conversion avec un entity read de 1 million/mois et entity write de 100000/mois et entity delete de 500/mois

Donc en total on aura:

- 2GB de stockage pour les utilisateurs, et états de conversion
- 11million/mois entity read
- 101000/mois entity write
- 1000/mois entity delete

Ceci donnera comme résultat un coût total mensuel de: **\$5.69**

### III. Diagramme de séquence

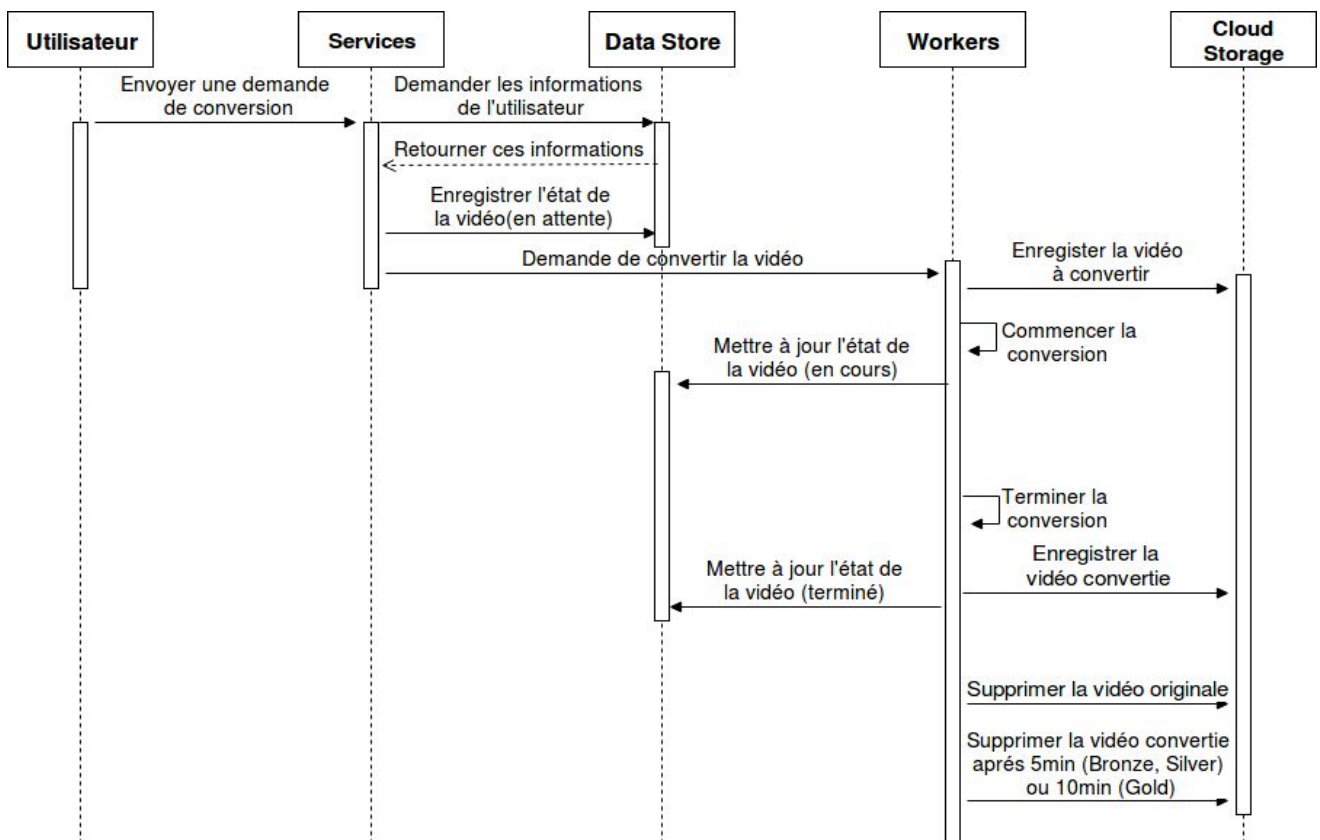


figure 2 : Diagramme de séquences

Ce diagramme met en évidence les différentes interactions qui se produisent au niveau du Storage ( DataStore, Cloud Storage ) pour assurer une conversion, quand l'utilisateur upload sa vidéo s'entraîne 2 accès sur le Cloud Store(vérifier les informations de l'utilisateur et d'initialiser une tâche de conversion ), et un accès en écriture sur le cloud storage à fin d'enregistrer la vidéo avant la conversion. Pour les prochaines étapes de conversion, on a des accès multiples sur le cloud Data à fin de mettre à jour l'état de la vidéo(pour contrôler le nombre de conversion pour

un utilisateur donné) et l'accès sur le Cloud Store à fin d'enregistrer la vidéo convertie et assure sa suppression après telles minutes.