

CaaS - Conversion as a Service

Il s'agit de développer une application cloud native de transcodage de vidéo asynchrone proposant plusieurs types de transcodage et plusieurs niveaux de qualité de service

Fonctionnement général

- Un client se rend sur la page d'accueil de l'application. Il y dépose sa vidéo, choisit les conversions à réaliser parmi celles disponibles, la qualité de service attendu et un e-mail de contact.
- Une fois la demande enregistrée, l'application envoie un mail confirmant que l'opération est en cours et un lien vers une page résumant le statut de chaque conversion : en attente, en cours, terminé. Pour chaque statut, il importe de connaître le temps passé dans l'état courant.
- Lorsqu'une vidéo est convertie, un lien est mis à disposition depuis la page de statut permettant de la télécharger. L'utilisateur dispose alors 5 minutes pour la télécharger après quoi elle sera supprimée.
- Quand la vidéo a été transcodée dans tous les formats demandés, un e-mail est envoyé à l'utilisateur.

Particularités pour le TP

- L'utilisateur ne dépose pas une vraie vidéo, il indique une durée et un nom. Du côté serveur, un fichier aléatoire de 1MB par seconde de vidéo sera stocké.
- On ne développe pas l'opération de transcoding à proprement parlé. Elle sera simulée par une phase d'attente.
- On considère 5 types de conversion. Hormis un identifiant, elles se différencient par la durée de l'opération (une durée de conversion par seconde de vidéo)

Qualité de service

L'application fournit 3 types de qualité de service

- niveau `bronze`: la vidéo doit faire 1 minute maximum. Toutes les vidéos à convertir avec ce contrat (quel que soit l'utilisateur) sont mises dans un pot commun et transcodées une à une dans l'ordre de soumission.
- niveau `silver`: Il n'y a plus de restrictions sur la durée de la vidéo. Il est possible de faire 3 conversion à la fois *par utilisateur*.
- niveau `gold`: Il est possible de réaliser 5 conversions simultanément. Les vidéos sont conservées 10 minutes.

Implanter certains SLA précisément peut s'avérer techniquement difficile. À vous de faire des choix et de relaxer certains règles théoriques dans la pratique si cela est nécessaire. À vous de vous justifier également.

Travail à rendre

Un rapport sur Google doc décrivant:

- l'architecture de l'application (et sa justification): avant le 12/10, 23h59
- L'architecture de la couche donnée (et sa justification): avant le 26/10, 23h59
- l'url où l'application est disponible et un scénario de test
- un modèle de coût pour l'application
- Un accès en lecture¹ au dépôt git (privé) de l'application et le nécessaire pour compiler et déployer

Deadline pour le rendu final: 11 novembre, 23h59

¹ Pseudos:
github/bitbucket: fhermeni
Atlas.i3s.unice.fr: hermenier