



**PROJET DE
PLATEFORME DE
STREAMING INTERACTIF**

-

**DOCUMENT DE
DÉFINITION DE
L'ARCHITECTURE**

DÉFINITION DE L'ARCHITECTURE

TABLE DES MATIÈRES

SOMMAIRE	3
HISTORIQUE	3
OBJECTIF	3
DESCRIPTION ET CONTRAINTES DU PROJET	3
PRINCIPES D'ARCHITECTURE	3
ARCHITECTURE DE BASE	4
RAISON D'ÊTRE ET JUSTIFICATION DE L'APPROCHE ARCHITECTURALE	6
Standards de l'industrie	6
Sécurité	7
Scalabilité / Évolutivité	7
Portabilité	7
Couverture fonctionnelle	8
ARCHITECTURE CIBLE	9
Modèles d'architecture d'entreprise	12
Modèles d'architecture de données	12
Modèles d'architecture d'application	13
Modèles d'architecture technologique	14
ANALYSE DES LACUNES	16
Plateforme publique	16
Sécurisation des accès	16
Fonctionnalités intégrées	16
Scalabilité	16
Évolutivité	16
Portabilité	17
ANALYSE DE L'IMPACT	18

DÉFINITION DE L'ARCHITECTURE

SOMMAIRE

Ce document présente l'analyse de base, l'objectif et les lacunes du projet de streaming vidéo interactif Gibberish.net.

HISTORIQUE

Date	Version	Commentaires
8 décembre 20--	0.01	Document de vision préliminaire.
Dec 2022	1.0	

OBJECTIF

Ce document de définition de l'architecture définit l'architecture de base, l'architecture cible et l'analyse des lacunes pour la prochaine génération de l'offre de produits Gibberish.net : Projet de streaming vidéo interactif.

DESCRIPTION ET CONTRAINTES DU PROJET

Voir le document du **Cahier des charges d'architecture**.

PRINCIPES D'ARCHITECTURE

Voir le document du **Cahier des charges d'architecture**.

DÉFINITION DE L'ARCHITECTURE

ARCHITECTURE DE BASE

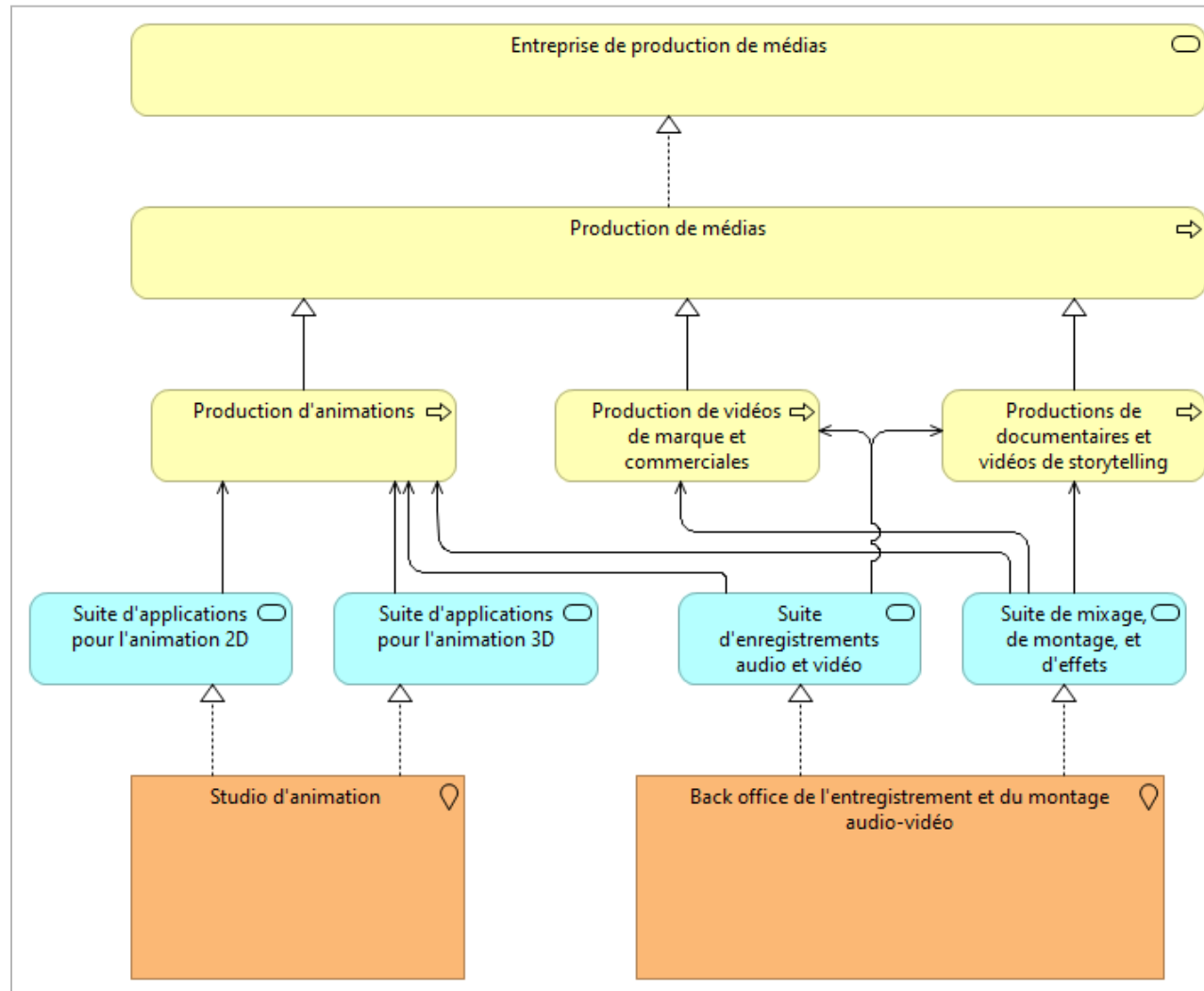
Gibberish.net propose actuellement trois types de services de production de médias :

- Vidéos de marque et commerciales
- Production de vidéos d'animation
- Documentaires et récits numériques

Le schéma ci-dessous montre l'architecture de base pour la production de médias, avec :

- les trois types de service (en jaune, à la troisième ligne),
- les suites (en bleu, à la quatrième ligne),
- les départements clés (en orange, au bas du schéma) :

DÉFINITION DE L'ARCHITECTURE



DÉFINITION DE L'ARCHITECTURE

RAISON D'ÊTRE ET JUSTIFICATION DE L'APPROCHE ARCHITECTURALE

L'architecture est de type microservice hébergée et couplée à un CDN (Content Delivery/Distribution Network) :

- Permet de réduire les temps de latence pour le service de streaming
- Permet de proposer des fonctionnalités métiers à nos utilisateurs externes (interactivité, vidéos personnalisées)
- Permet de changer de technologie rapidement afin de pouvoir toujours utiliser les meilleurs services

L'approche architecturale est justifiée par les éléments suivants :

- Respect des standards de l'industrie
- Sécurité
- Scalabilité
- Portabilité
- Couverture fonctionnelle

Standards de l'industrie

L'approche architecturale de type microservices respecte les standards suivants de l'industrie :

- Les principes SOLID (Responsabilité unique, Ouvert/fermé, Substitution de Liskov, Ségrégation des Interfaces, Inversion des Dépendances). L'acronyme SOLID est un acronyme mnémonique qui regroupe cinq principes de conception destinés à produire des architectures logicielles plus compréhensibles, flexibles et maintenables et qui permettent d'éviter la régression suite à l'ajout de nouvelles fonctionnalités.
- L'architecture REST (REpresentational State Transfer) qui, avec l'utilisation d'un

DÉFINITION DE L'ARCHITECTURE

protocole sans état et d'opérations standards, vise la réactivité, la fiabilité et l'extensibilité, par la réutilisation de composants pouvant être gérés et mis à jour sans affecter le système global, même pendant son fonctionnement.

Sécurité

L'architecture est sécurisée grâce à un système d'authentification centralisé ainsi qu'à un stockage organisé en fonction de la sensibilité des données/services.

Un firewall et un cryptage des données permet également de sécuriser les échanges de données.

Scalabilité / Évolutivité

L'utilisation de conteneurs cloud permet de s'adapter facilement à la montée en charge et permet donc à l'architecture d'être parfaitement scalable. De plus, l'association de conteneurs et de microservices permet de rendre l'application évolutive en permettant de remplacer un service indépendamment des autres.

Portabilité

Les applications front PWA (Progressive Web App) composant l'architecture s'adaptent à tous types de matériels, de navigateurs.

La conteneurisation des microservices permet de s'adapter à tous les systèmes d'exploitation.

DÉFINITION DE L'ARCHITECTURE

Couverture fonctionnelle

Toutes les exigences fonctionnelles requises pour ce projet de plateforme de streaming interactif sont couvertes :

- Le streaming est possible
- La création de médias composites est possible
- Les utilisateurs externes ont la possibilité de monter leur propre vidéo de type media composite ou non



DÉFINITION DE L'ARCHITECTURE

ARCHITECTURE CIBLE

L'apparence de l'IHM de chaque utilisateur se fait en fonction de son profil (géré par un service tel que KeyCloak). Leurs IHM et droits d'accès fonctionnels seront donc spécifiques :

- Pour les utilisateurs externes non-authentifiés avec accès public : accès au portail de diffusion des médias
- Pour les utilisateurs externes authentifiés : accès identique aux utilisateurs précédents avec en plus un accès au portail de production de médias dédié aux utilisateurs authentifiés
- Pour les utilisateurs internes : accès accès identique aux utilisateurs externes authentifiés avec en plus un accès au portail de production de médias dédié aux utilisateurs internes

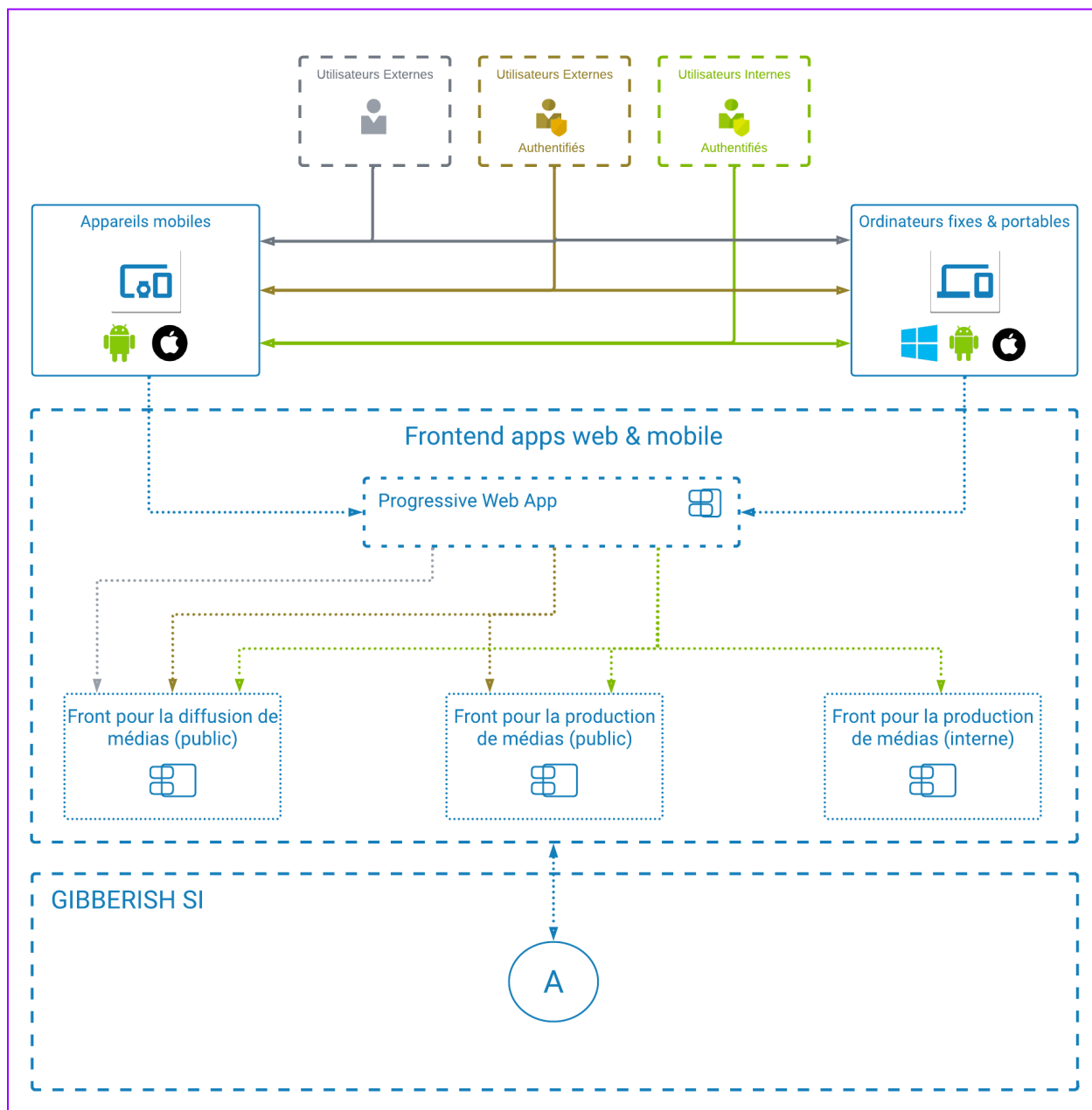
Les applications front seront de type PWA (Progressive Web App) afin de pouvoir être utilisées à la fois sur ordinateur, tablette et mobile.

L'architecture est de type microservices. Elle est organisée autour d'une passerelle d'API (API Gateway) avec d'un côté les différents portails d'accès et de l'autre les services permettant d'assurer les fonctions métiers.

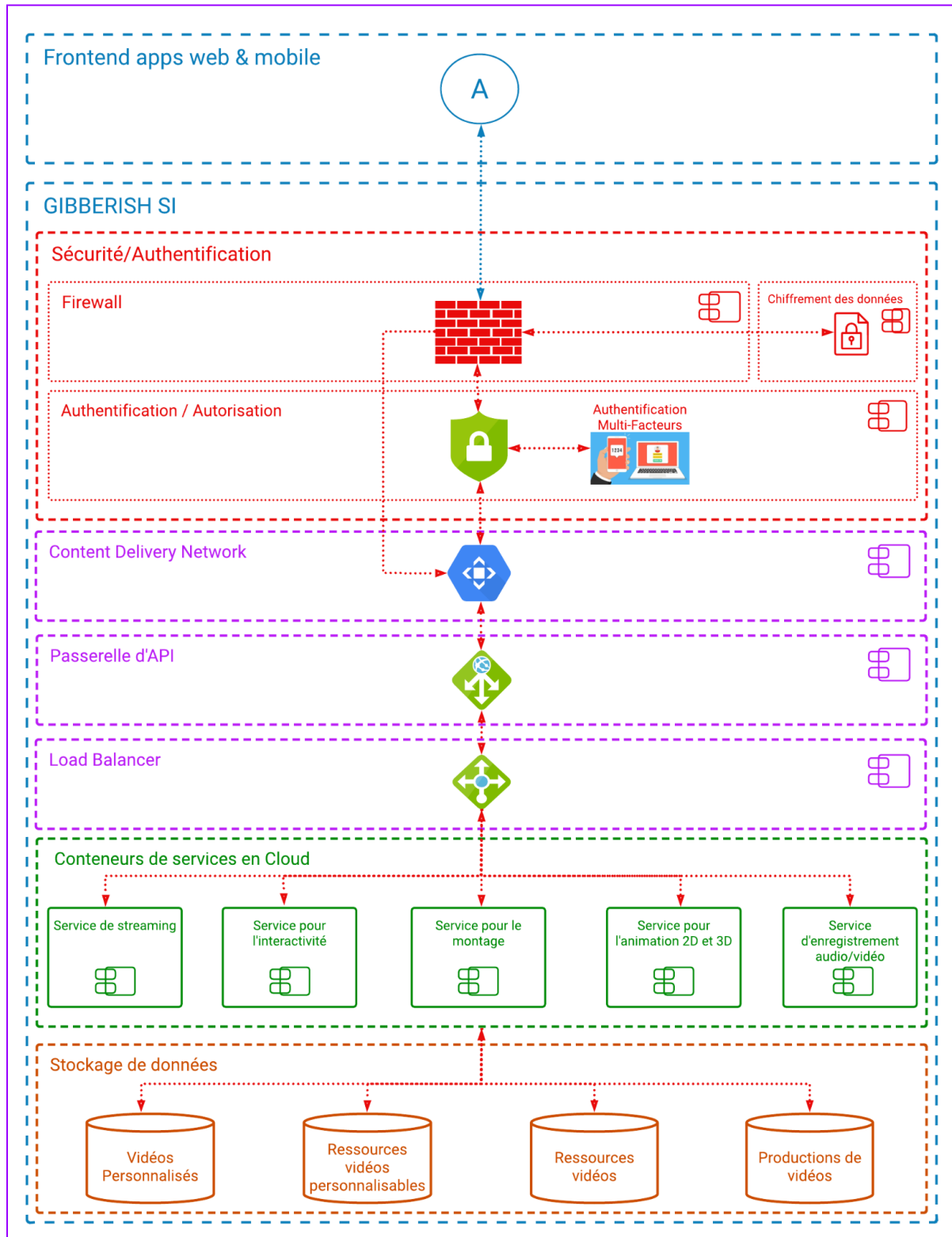
Cette architecture est couplée avec un CDN qui permet d'assurer la diffusion des vidéos en réduisant les temps de latence.

Un Load Balancer est également présent afin de s'assurer de la disponibilité de nos services en cas de forte charge d'utilisation.

DÉFINITION DE L'ARCHITECTURE



DÉFINITION DE L'ARCHITECTURE



DÉFINITION DE L'ARCHITECTURE

Modèles d'architecture d'entreprise

Les fonctions business sont couvertes grâce à :

- des fronts dédiés à chaque profil utilisateur
 - portail public permettant le streaming vidéo
 - portail public avec authentification permettant la production de vidéos composites personnalisées
 - portail interne avec authentification permettant la production de vidéos composites
- une gestion des rôles et de l'authentification assurée par un service dédié intégré à l'architecture
- un ensemble de services métier servant les portails applicatifs

Modèles d'architecture de données

Les données sont séparées en 4 grands types et accessibles selon les profils utilisateurs :

- Les données vidéos sources enregistrées par nos utilisateurs internes et seulement disponibles pour eux
- Les données vidéos produites par nos utilisateurs externes et disponibles pour nos utilisateurs internes
- Les données vidéos sources disponibles pour nos clients utilisateurs authentifiés
- Les données vidéos personnalisées réalisées par nos clients utilisateurs authentifiés et disponibles pour nos clients utilisateurs

Les accès à ces différents types de données sont sécurisés par le service d'authentification.

DÉFINITION DE L'ARCHITECTURE

Modèles d'architecture d'application

L'architecture est organisée autour de services métiers que les fronts applicatifs vont exploiter en fonction des droits d'accès de chaque utilisateur.

DÉFINITION DE L'ARCHITECTURE

Modèles d'architecture technologique

L'architecture est de type microservices et est organisée de la manière suivante en 3 parties :

- La partie frontend qui contient les interfaces destinées aux utilisateurs
- La partie backend qui contient les services et bases de données exploités par les interfaces
- La partie middleware qui est en charge d'organiser les échanges entre le frontend et le backend et d'assurer la disponibilité des services

Le frontend est composé de :

- Portail public de Gibberish (Gibberish.net) permettant de se connecter aux portails avec authentification et donnant accès au streaming public
- Accès sécurisés aux portails des utilisateurs internes et externes
- Portail interne :
 - Accès aux fonctionnalités de montage, animation 2D/3D, exploitation de vidéo 360
 - Accès aux fonctionnalités d'interactivité
 - Accès aux bases de données des ressources vidéos contenues dans les matériels d'acquisition (matériels de captation/enregistrement)
- Portail externe :
 - Accès aux fonctionnalités de montage, animation 2D/3D, exploitation de vidéo 360
 - Accès aux fonctionnalités de streaming
 - Accès aux fonctionnalités de montage et d'interactivité avec des droits d'accès spécifiques aux utilisateurs externes
 - Accès aux bases de données des vidéos personnalisées en fonction des droits spécifiques d'accès de chaque utilisateur

DÉFINITION DE L'ARCHITECTURE

Le backend est composé(*) de :

- Service d'authentification/autorisation assuré par [Keycloak](#) ou [Okta](#)
- Firewall sur lequel est exploité un protocole d'encryption
- Service de streaming et d'hébergement assuré par [Muvi](#) ou [Datacast](#)
- Service pour l'interactivité assuré par [Evantagesoft Interactive Media Platform](#) ou [Datacast](#)
- Service pour le montage et la vidéo 360 assuré par [Adobe Premiere Pro](#) ou [Final Cut Pro X](#)
- Service pour l'animation 2D assuré par [OpenToonz](#) ou [PowToon](#)
- Service pour l'animation 3D assuré par [Blender](#) ou [DAZ Studio](#)
- Bases de données hébergées sur [Vimeo](#), [Datacast](#) ou [AWS](#)
- Hébergement et conteneurisation des services sur [AWS](#) couplé à [Docker](#)

(*) Les outils et services sont donnés ici à titre d'exemple

L'articulation entre le frontend et le backend est assuré par :

- CDN (Content Delivery/Distribution Network) pour assurer un accès rapide aux données vidéos (notamment pour le streaming)
- API Gateway pour assurer un accès des différents utilisateurs aux services et entre les services
- Load Balancer pour assurer la disponibilité des microservices en cas de montée en charge

DÉFINITION DE L'ARCHITECTURE

ANALYSE DES LACUNES

Plateforme publique

L'architecture n'est pas adaptée pour être ouverte aux utilisateurs externes.

Sécurisation des accès

L'architecture de base ne permet pas l'accès sécurisé pour les utilisateurs externes.

L'architecture de base ne permet pas à des utilisateurs externes d'exploiter les ressources logicielles nécessaires pour la production de vidéos personnalisées.

Fonctionnalités intégrées

L'architecture de base ne permet pas le mix de sources vidéos : par exemple, elle ne permet pas de produire une vidéo commerciale composée d'animations.

L'architecture de base s'appuie sur des outils indépendants les uns des autres et ne permet donc pas d'avoir une utilisation unifiée et fluide de ceux-ci.

Scalabilité

L'architecture n'est pas scalable pour être utilisée en tant que plateforme de streaming.

Évolutivité

L'approche monolithique (*a priori*) de l'architecture la rend difficilement évolutive.

DÉFINITION DE L'ARCHITECTURE

Portabilité

L'architecture ne s'appuie pas sur des applications web.

L'architecture ne permet pas l'utilisation de tous les types de terminaux.



DÉFINITION DE L'ARCHITECTURE

ANALYSE DE L'IMPACT

Cette nouvelle architecture va nous permettre de proposer facilement de nouveaux services à nos clients et utilisateurs.

Elle pourra s'adapter à une forte augmentation de trafic (phénomène inhérent aux plateformes de streaming).

L'exploitation de microservices permet de remplacer aisément une technologie assurant une fonctionnalité sans empêcher la continuité de service.

Une monétisation des services est également envisageable en s'appuyant sur le service d'authentification/autorisation.