**DOCUMENT DE DÉFINITION D’ARCHITECTURE**



**David EVAN**

**05/04/2022**

**Version 1.0**

**Projet de streaming vidéo interactif - GIBBERISH.NET**

**Historique des révisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Numéro de version** | **Auteur** | **Description** | **Date de modification** |
| 1.0 | EVAN David  *(Architecte logiciel)* | Livraison initiale | 05/04/2022 |
|  |  |  |  |
|  |  |  |  |

Tableau - Historique des révisions

**Objectif du document**

Ce document présente l'analyse de l’architecture actuelle (dite : de référence), les objectifs et les lacunes du projet de streaming vidéo interactif Gibberish.net.

Il 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 du projet et des principes d’architecture**

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

**Table des matières**

[ARCHITECTURE DE REFERENCE 5](#_Toc100000136)

[APPROCHE ARCHITECTURALE 6](#_Toc100000137)

[Justification de l’approche architecturale 6](#_Toc100000138)

[Principes d’architectures 6](#_Toc100000139)

[Cloud Based Architecture 6](#_Toc100000140)

[Briques d’architecture de référence (ABB) 7](#_Toc100000141)

[Composants logiciels et techniques 7](#_Toc100000142)

[Composants « middleware » 7](#_Toc100000143)

[Schéma des briques d’architecture de référence 8](#_Toc100000144)

[Étude exploratoire de la stack technologique 9](#_Toc100000145)

[Choix préférés pour les outils et technologies 9](#_Toc100000146)

[Technologies pour la diffusion des médias vidéo sur le web 9](#_Toc100000147)

[Protocoles HLS et DASH 9](#_Toc100000148)

[Technologies pour l’authentification / l’autorisation 11](#_Toc100000149)

[Briques de solution de référence (SBB) 12](#_Toc100000150)

[Solution : Production des médias interactifs (SBB-1) 12](#_Toc100000151)

[Solution : Transcodage / Convertisseur media (SBB-2) 12](#_Toc100000152)

[Solution : Stockage des médias (SBB-3) 13](#_Toc100000153)

[Solution : Diffusion de média sur le web (SBB-4) 13](#_Toc100000154)

[Solution : Authentification / Autorisation (SBB-5) 15](#_Toc100000155)

[Solution : Visualisation des médias (SBB-6) 15](#_Toc100000156)

[Synthèse de l’étude de la stack technologique 16](#_Toc100000157)

[Catalogue des briques de solution de référence 16](#_Toc100000158)

[Schéma d’implémentation 16](#_Toc100000159)

[Risques d’architecture 17](#_Toc100000160)

[ALIGNEMENT SUR LE REFERENTIEL D'ARCHITECTURE 18](#_Toc100000161)

[Alignement sur le paysage architectural 18](#_Toc100000162)

[Alignement sur les modèles de référence 18](#_Toc100000163)

[Alignement sur les normes 18](#_Toc100000164)

[Évaluation de la réutilisation 18](#_Toc100000165)

[ANALYSE DES ÉCARTS 19](#_Toc100000166)

[Écarts d’architecture 19](#_Toc100000167)

[Business 19](#_Toc100000168)

[Data 19](#_Toc100000169)

[Application 19](#_Toc100000170)

[Technique 19](#_Toc100000171)

[ANALYSE DES IMPACTS 20](#_Toc100000172)

[TABLES DES RÉFÉRENCES 21](#_Toc100000173)

[Figures 21](#_Toc100000174)

[Tableaux 21](#_Toc100000175)

# ARCHITECTURE DE REFERENCE

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-après 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), et les départements clés (en orange, au bas du schéma) :

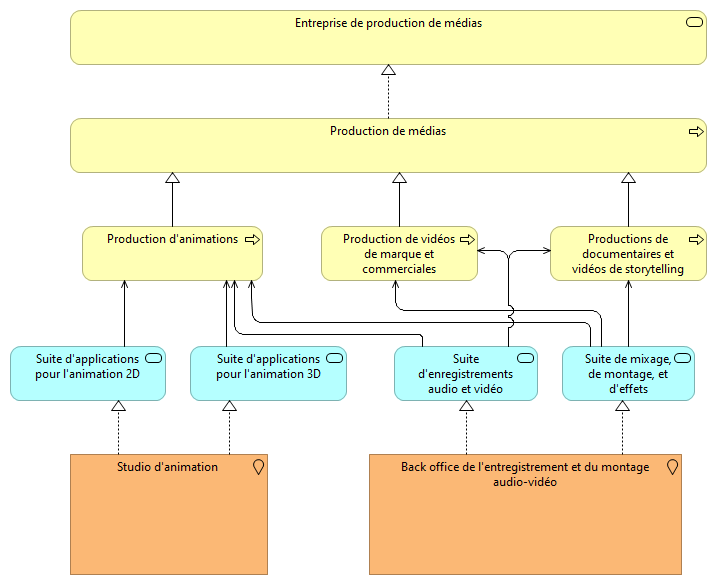


Figure : Vue synthétique de l'architecture de référence

# APPROCHE ARCHITECTURALE

## Justification de l’approche architecturale

L’architecture cible définie pour la réalisation du projet de « *streaming vidéo interactif* » doit permettre de répondre aux principales exigences des parties prenantes telles que décrites dans le cahier des charges d’architecture (Voir les sections : Exigences fonctionnelles et non fonctionnelles).

La section ci-après présente et justifie l’approche architecturale retenue pour ce projet.

### Principes d’architectures

Au-delà des spécifications de ce projet décrite dans le cahier des charges, l’architecture doit répondre aux principes d’architecture définis tel que défini dans le référentiel de l’entreprise et pouvant être considérés plus globalement comme des bonnes pratiques de conception, à savoir :

* L’architecture doit être au service de l’expérience utilisateur. (**Performance**)
* L’architecture doit faciliter l’innovation rapide et s’adapter aux besoins business. (**Évolutivité**)
* L’architecture doit faciliter les interactions entre composants logiciels. (**Interopérabilité**)
* L’architecture doit être capable de s’adapter aux besoins des utilisateurs. (**Scalabilité**)
* L’architecture doit être orienté sécurité et garantir la protection des données et l’accès aux ressources en fonction des habilitations. (**Sécurité**)

### Cloud Based Architecture

Une approche Cloud permet de complémenter l’offre de service déjà existante de l’entreprise par un service supplémentaire accessible dans le **Cloud** sous forme de **SaaS**.

Une approche Cloud sera préférée à une approche On-Premise pour satisfaire aux besoins de scalabilité et d’évolutivité tout en maximisant le ROI et en s’adaptant aux besoins réels du projet. À ce titre, la plateforme cloud retenu devra obligatoirement implémenter **des mécanismes de mise à l’échelle automatique** (*autoscaling*).

### Briques d’architecture de référence (ABB)

#### Composants logiciels et techniques

Le projet de « *streaming vidéo interactif* » est composé de plusieurs briques d’architecture logicielles ou techniques de référence (*Architecture Building Blocks).* Notons que ces briques peuvent mettre en œuvre plusieurs composants logiciels pour remplir leur fonction.

|  |  |  |  |
| --- | --- | --- | --- |
| **Id.** | **ABB** | **Description** | **Type** |
| **ABB-1** | Production médias interactif | Brique logicielle permettant pour les producteurs d’assurer la création de contenu interactifs. | Logiciel |
| **ABB-2** | Transcodage / Convertisseur media | Brique logicielle visant à transcoder (convertir) des fichiers multimédias d'un format source vers un format lisible par des appareils tels que les smartphones, les tablettes et les ordinateurs. | Logiciel |
| **ABB-3** | Stockage des médias | Brique technique visant à permettre le stockage massif des médias produits et / ou des différents artefacts (versionning, assets, ressources …) | Technique |
| **ABB-4** | Diffusion de média sur le web | Brique logicielle et technique assurant la distribution (diffusion) des flux de médias selon les protocoles adaptés (HLS, DASH …). | Logiciel / Technique |
| **ABB-5** | Authentification / Autorisation | Brique logicielle permettant d’assurer l’authentification et l’autorisation d’accès aux différentes ressources (ex : contrôle des droits d’accès ou de la présence d’un contexte d’authentification) | Logiciel |
| **ABB-6** | Visualisation des médias | Brique logicielle visant à permettre à l’utilisateur final d’accéder aux médias interactifs. Est composé à la fois d’une plateforme d’accès (ex : site web) ainsi que de composants logiciels spécifiques (video player) permettant la diffusion des médias produits. | Logiciel |

Tableau : Catalogue des briques d'architecture de référence (ABB)

#### Composants « middleware »

Bien que non décrit comme brique d’architecture de référence (et non présenté dans le schéma d’architecture), des composants middleware peuvent être envisagé afin d’améliorer l’efficacité et les performances de l’architecture :

* **CDN** (*Content Delivery Network*), pour la distribution de contenu au plus proche de l’utilisateur.
* **API Gateway**, pour la fédération des API, **Load balancer** pour la répartition de charge.
* **WAF** (*Web Application Firewall*), pour la sécurisation des applications exposés sur le web,
* **Système de cache à accès rapide** (type *Redis*) pour l’amélioration des performances.

Ces composants et leur mise en œuvre seront décrits dans les *spécifications techniques* de la plateforme.

#### Schéma des briques d’architecture de référence

Le schéma ci-après présente les différentes briques d’architecture développée dans la section précédente et défini les relations entre les différents composants.

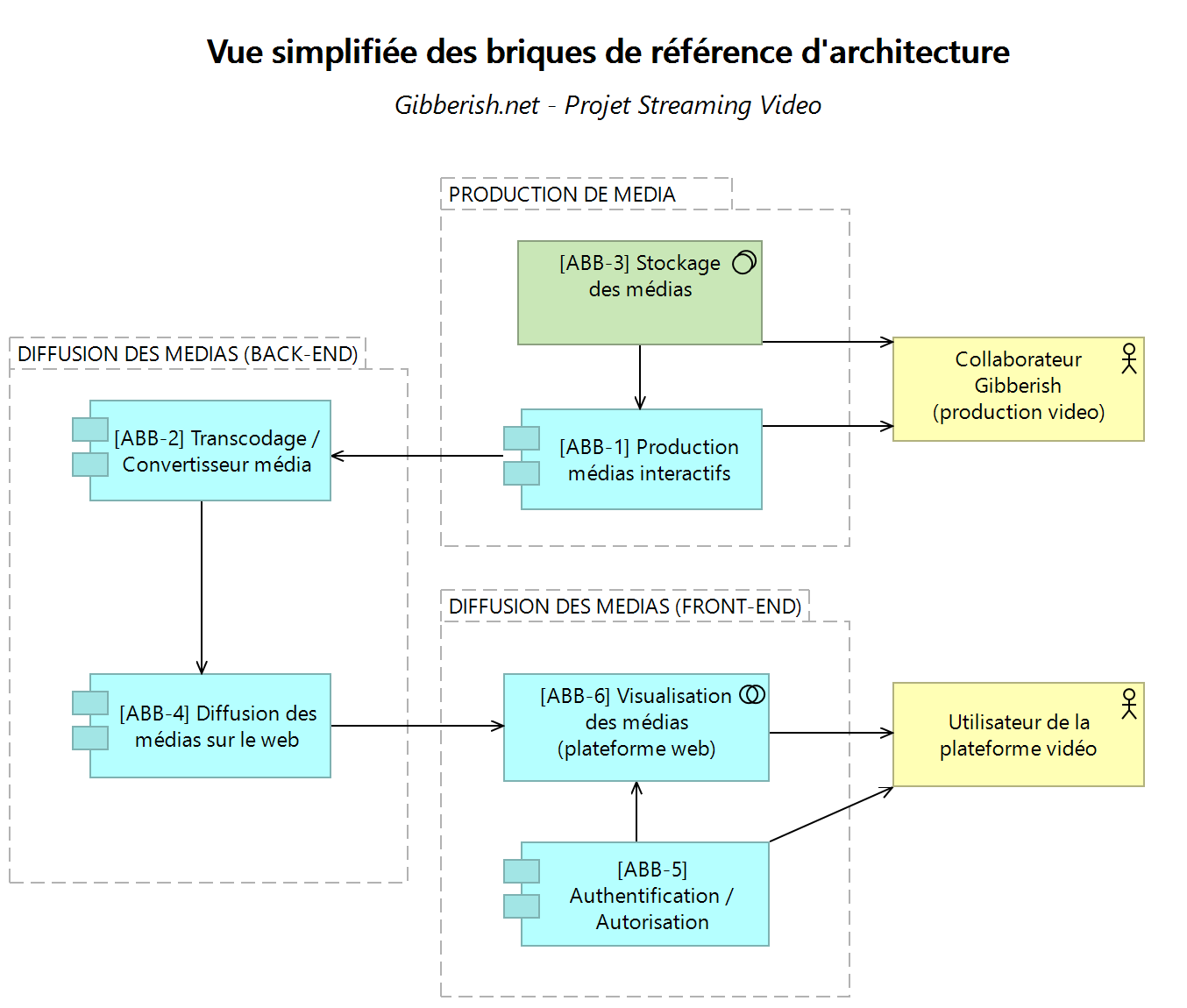


Figure : Vue simplifiée des briques de référence d'architecture (ABB)

## Étude exploratoire de la stack technologique

L’étude exploratoire de la stack technologique vise à présenter et / ou à confronter les briques de solution de référence (*Solution Building Block*) retenues pour la conception de l’architecture cible.

### Choix préférés pour les outils et technologies

Bien que les solutions puissent être des conceptions « from scratch », **les solutions préexistantes seront favorisées** dès lorsqu’ils répondent aux besoins définis dans le cahier des charges et que la tarification est adaptée. Cette approche vise à permettre d’assurer une livraison rapide du projet, une réduction des coûts de mise en œuvre et favorise une approche modulaire.

Les choix d’outils et de technologies retenues devront répondre à des critères de cohérence d’ensemble. Les solutions compatibles les unes par rapport aux autres et/ou facilement interopérables seront préférés. Cette approche vise à favoriser l’évolutivité de l'architecture retenue.

Notons que les critères de popularités des outils, de facilité à trouver des ressources et des profils de collaborateurs expérimentés seront aussi analysée pour le choix des solutions.

Le coût des licences et l’adéquation au budget du projet (non défini au moment de la rédaction de ce document) seront pris en compte pour le choix final des solutions.

### Technologies pour la diffusion des médias vidéo sur le web

Le streaming de médias vidéo (interactifs ou non) sur le web nécessite l’utilisation de format adaptés afin de garantir la meilleur qualité selon les performances de la connexion internet de l’utilisateur final.

Les protocoles disposant d’un bitrate adaptatifs seront préférés afin de satisfaire à cette exigence.

#### Protocoles HLS et DASH

Les protocoles HLS et MPEG-DASH seront retenues comme technologie de diffusion pour le projet de média interactif.

De manière simplifiée, HLS (*HTTP Live Steaming*) et DASH (*Dynamic Adaptive Streaming over HTTP*) sont deux protocoles de streaming audio / vidéo basé sur HTTP et visant à délivrer des médias à l’utilisateur en « fragmentant » les fichiers d’origines en plusieurs « sous-fichiers » de qualités différentes afin de fournir le plus adapté à l’utilisateur en fonction de l’avancement de sa visualisation.

HLS et DASH supportent tous deux les principaux codec audio / vidéo de l’industrie : *VP9, AAC-LC, FLAC, H.265, H.264 … (non-exhaustif)*.

Le protocole DASH embarque plusieurs fonctionnalités supplémentaire par rapport au HLS, notamment la possibilité d’utiliser des DRM sur les médias transportés.

Bien que l’implémentation de DASH soit relativement plus complexe, le fonctionnement général est relativement similaire. *(DASH ne sera pas approfondi dans cette section. Si nécessaire, des ressources documentaires sont disponible en fin de section pour approfondir la compréhension de ces technologies)*

Le schéma ci-après présente de façon simplifiée le fonctionnement d’un flux HLS :

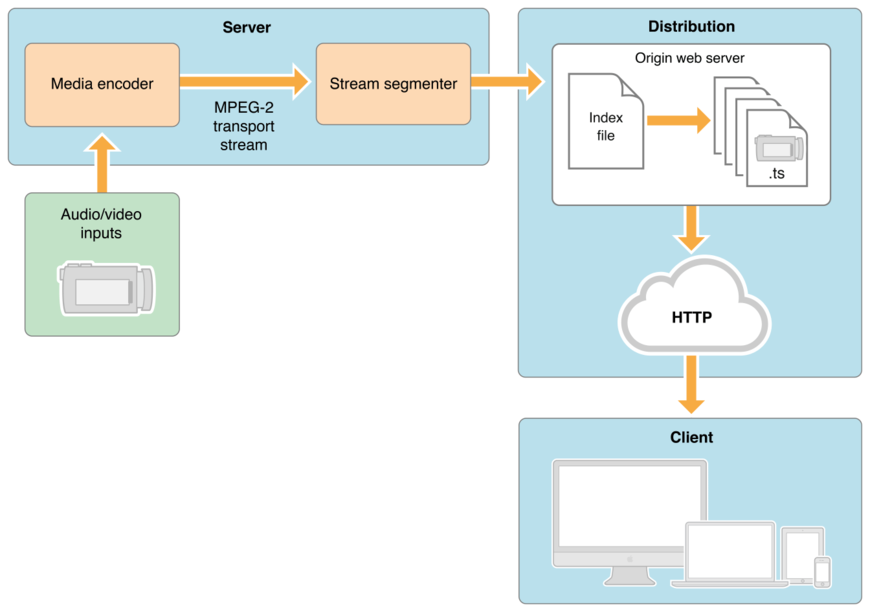


Figure : Vue synthétique d'un envoi réalisé avec HLS (source : Eleven Labs Blog)

Comme indiqué précédemment, la découpe des fichiers source en segments de qualité différentes permet l’adaptation du bitrate de la diffusion au fur et à mesure de l’avancé afin de garantir à l’utilisateur une lecture fluide en sacrifiant la qualité sur certains passage (lorsque la connexion de l’utilisateur n’arrive plus à supporter le flux).

Un manifeste est fourni en complément et permet de décrire au client (navigateur web) comment construire la vidéo.

La figure ci-après (*Figure 4*) montre le fonctionnement du principe de segmentation du fichier source en section différentes.

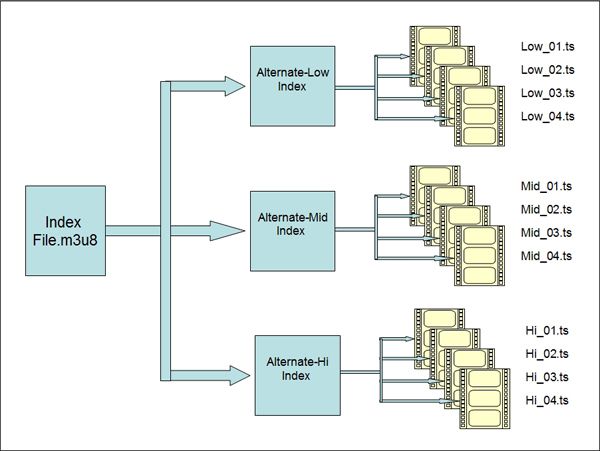


Figure : Segmentation d’un média en différentes qualités pour diffusion via HLS (source : Eleven Labs Blog)

##### Documentation complémentaire :

* <https://www.wowza.com/blog/mpeg-dash-dynamic-adaptive-streaming-over-http>
* <https://blog.eleven-labs.com/fr/video-live-dash-hls/>

### Technologies pour l’authentification / l’autorisation

Afin de disposer d’une solution évolutive et pour permettre de simplifier l’ensemble des mécanismes d’authentification et d’autorisation, les technologies OAuth2 et sa couche d’identité OIDC (*Open ID Connect*) seront utilisées pour la gestion des accès aux ressources via un mécanisme de jeton d’accès (*access\_tokens*).

Les niveaux d’autorisation seront gérés à l’aide des *scopes* embarqués dans les jetons OAuth2 et les profils utilisateurs à l’aide des jetons d’identité (*id\_tokens). Étant hors du scope de ce document de définition d’architecture, les mécanismes de fonctionnement de l’authentification / autorisation seront abordés dans les spécifications techniques.*

##### Documentation complémentaire :

* <https://datatracker.ietf.org/doc/html/rfc6749>
* <https://openid.net/connect/>

### Briques de solution de référence (SBB)

#### Solution : Production des médias interactifs (SBB-1)

La production de médias interactifs nécessite l’utilisation d’outils complets et performants afin de couvrir l’ensemble des besoins décrits dans le cahier des charge d’architecture (vidéo 360, vidéo interactives, multi-view …).

Notons par ailleurs que la production de vidéo 360° nécessite par ailleurs des outils complémentaires pour la *capture*, le *stitching* et le *montage* des séquences.

Le logiciel **Unity** (et sa plateforme) sera retenu comme outil de production de médias interactif. Ce logiciel offre une gamme très large d’outils pour la création de média et couvre 100% des besoins définis. Notons qu’un système de « plug-ins » téléchargeable via le **Unity Asset Store** permet d’enrichir les fonctionnalités offertes par le logiciel de base.

##### Documentation complémentaire :

* <https://unity.com/fr/solutions/film-animation-cinematics>
* <https://unity.com/fr/solutions/360video>
* <https://assetstore.unity.com/>

#### Solution : Transcodage / Convertisseur media (SBB-2)

L’un des besoins décrit dans les briques d’architecture de référence consiste dans le transcodage des médias produits (format de sortie de l’outil de production) au format compatible avec la diffusion sur le web (par exemple, HLS / DASH comme décrit dans les technologies retenues).

La solution **AWS Elemental Media Converter** sera utilisée pour la conversion des médias source. Cet outil couvre l’intégralité des besoins nécessaire au projet et s’adapte au volume nécessaire (tarification à la minute convertie). Le workflow peut être automatisé et couplé avec AWS S3 pour automatiser l’ensemble de chaine et libérer les créateurs de contenu de ce travail.

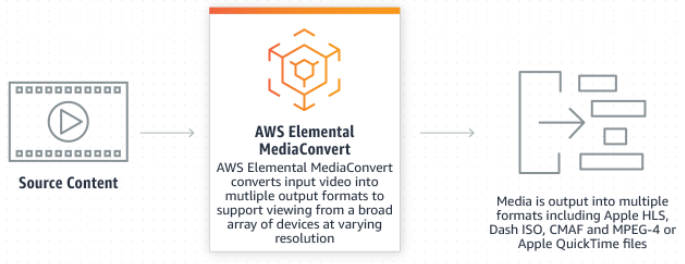


Figure : Fonctionnement d'AWS Elemental Media Converter (Source : AWS)

##### Documentation complémentaire :

* <https://aws.amazon.com/fr/mediaconvert/>

#### Solution : Stockage des médias (SBB-3)

Afin de permettre le stockage de l’ensemble des médias produit et en attente de diffusion (version drafts, médias non publiés …) ainsi que les ressources nécessaires à la production (audio, vidéos brutes …), un espace de stockage de haute capacité est nécessaire.

La solution **AWS S3** (*Simple Storage Service*) sera retenue. Cette solution répond parfaitement aux besoins de la société tout en permettant une intégration simplifiée au système d’autorisation basé sur les rôles (identique pour les clients et les collaborateurs) afin de garantir la sécurité du contenu.

Cette solution est totalement évolutive et garanti une adaptation parfaite aux besoins grandissant de l’entreprise. Le SLA extrêmement important garanti la disponibilité des données.

##### Documentation complémentaire :

* <https://aws.amazon.com/fr/s3/>

#### Solution : Diffusion de média sur le web (SBB-4)

La diffusion sur de média sur le web à grande échelle nécessite de prendre en compte les besoins de haute disponibilité, de régularité et de faible latence pour la distribution du contenu. Le service doit par ailleurs disposer de fonctionnalité de scalabilité automatique afin de s’adapter aux pics de diffusion.

Plusieurs options peuvent être envisagée pour le choix de la brique de solution retenu pour l’architecture.

##### Option 1 : Solution AWS Elemental MediaStore

**AWS Elemental Media Store** est un service fourni par la plateforme AWS et est spécialisé dans le stockage et la distribution de vidéo. Il fourni un point de stockage et assure la diffusion des médias tout en s’adaptant automatiquement à la demande.

L’intégration avec les autres services AWS (*IAM* et *CDN* par exemple) en fait une solution de choix pour l’architecture du projet. Notons que la tarification pour la plateforme peut toutefois devenir un frein à la croissance de l’entreprise notamment pour le contenu non premium et faiblement monétisable.

###### Documentation complémentaire :

* <https://aws.amazon.com/fr/mediastore/>

##### Option 2 : Solution Unity Multiplay

La plateforme Unity embarque une solution alternative, **Unity Multiplay**. Cette solution fournit une plateforme auto-scalable initialement prévue pour la diffusion de jeux-vidéo mais pouvant parfaitement couvrir nos besoins de diffusion de média à haute échelle.

Unity Multiplay offre l’avantage d’intégrer l’ensemble des frontaux (*CDN*) de manière native et peut être totalement personnalisée pour être adapté à nos besoins.

Par ailleurs, l’absence de tarification sur les flux de données entrant / sortant peut représenter une source importante d’économie.

Notons toutefois que cette plateforme impose plusieurs choix technologique, notamment sur l’OS embarqué (Windows Server 2012, Windows Server 2019 ou Ubuntu 18.04) et dispose d’une communauté et d’une documentation moins importante que la solution AWS.

###### Documentation complémentaire :

* <https://unity.com/fr/products/multiplay>
* <https://docs.unity.com/multiplay/shared/welcome-to-multiplay.html>

##### Avantages et inconvénients des deux options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Option** | **Solution** | **Avantages** | | **Inconvénients** |
| **Option 1** | AWS Elemental Media Store | * Totalement intégré à l’écosystème AWS (Ressources et autres services) * Auto-scalable * Très hautes performances * Disponible sous forme de service * Faible configuration (*SaaS*) | * Pas de personnalisation possible des services * Peut nécessiter l’ajout de frontaux (CDN) * Tarification sur le stockage et sur les flux de données entrant / sortant. | |
| **Option 2** | Unity Multiplay | * Totalement personnalisable * Tarification avantageuse (uniquement sur les performance de la plateforme) * Auto-scalable * Intégration des frontaux de base. | | * Configuration et intégration pouvant être complexe (*IaaS*) * Peu de ressources documentaires * Peut être intégré au workflow automatisé mais nécessite un travail important. |

Tableau : Avantages et inconvénients des différentes solutions pour la diffusion de média

##### Choix de la solution

Les deux solutions présentées pour la diffusion de média peuvent être retenue pour la conception de l’architecture.

Au vu des enjeux du projet, **il semble nécessaire de réaliser un PoC** sur la solution Unity afin de garantir sa cohérence vis-à-vis du projet et permettre ainsi un ROI substantiel par rapport à la solution AWS.

#### Solution : Authentification / Autorisation (SBB-5)

L’autorisation et l’authentification nécessite de faire appel à une solution IAM (Identity Access Manager) compatible avec les standards OAuth2 et OIDC afin de disposer d’une couche « universelle » d’authentification / d’identification / d’autorisation.

Bien que de nombreuses solution soient envisageable (Okta, Gravitee, Azure AD …), la solution **AWS Cognito** sera retenue. Ce choix se justifie par sa parfaite intégration à l’environnement AWS permettant de facilement configurer les autres services (*S3, Media Converter, Media Store, API Gateway* …) pour prendre en compte les règles à appliquer.

##### Documentation complémentaire :

* <https://aws.amazon.com/fr/cognito/>

#### Solution : Visualisation des médias (SBB-6)

La visualisation des médias doit répondre à deux caractéristiques : Fournir une plateforme web pour la présentation des médias (ex : SPA WebApp) et fournir les outils, notamment le player video, compatibles avec l’ensemble des exigences fonctionnelles : lecture de média disposant d’une interaction avec l’utilisateur, vidéo « 360 » (l’utilisateur peut « déplacer » la caméra), support HLS / DASH ...

La contrainte du multi-plateforme doit être respectée afin que la WebApp puisse être facilement accessible depuis les supports désignés dans les exigences non fonctionnelles du cahier des charges d’architecture.

**La WebApp sera développée en interne (« from strach »)** en utilisant les technologies adaptés (React, Vue.Js, Angular pour le front-end, Java Spring, PHP Laravel / Symfony pour le back-end) en fonction de l’écosystème de l’entreprise et de l’expertise des développeurs.

La solution **NexPlayer** sera retenue comme player vidéo. Cette solution se présente sous la forme d’un SDK permettant de créer un player video totalement personnalisable à partir d’une solution couvrant la totalité des besoins exprimés (vidéo 360, multi-view, Dynamic Streaming …)

Ce choix se justifie par plusieurs critères :

* Le player vidéo couvre l’ensemble des besoins décrit dans le cahier des charges.
* La solution est totalement personnalisable, permettant à l’entreprise de pouvoir créer sa propre UI adaptée à l’image de marque, voire de développer de nouvelles fonctionnalités.
* L’outil est open source et dispose d’une implémentation sur les principales plateforme (Web HTML5, Application mobiles, consoles de jeux, smart TV …)
* Le player dispose d’une excellente implémentation dans les outils Unity.

##### Documentation complémentaire :

* <https://nexplayersdk.com/>
* <https://github.com/NexPlayer/NexPlayer_Unity_Plugin>

### Synthèse de l’étude de la stack technologique

#### Catalogue des briques de solution de référence

|  |  |  |  |
| --- | --- | --- | --- |
| **Id. SBB** | **Référence  aux ABB** | **Solution** | **Rôle** |
| **SBB-1** | ABB-1 | Unity | Production médias interactif |
| **SBB-2** | ABB-2 | AWS Elemental Media Converter | Transcodage / Convertisseur media |
| **SBB-3** | ABB-3 | AWS S3 (*Simple Storage Service*) | Stockage des médias |
| **SBB-4-A** | ABB-4 | AWS Elemental MediaStore | Diffusion de média sur le web |
| **SBB-4-B** | ABB-4 | Unity Multiplay | Diffusion de média sur le web |
| **SBB-5** | ABB-5 | AWS Cognito | Authentification / Autorisation |
| **SBB-6** | ABB-6 | * Solution custom (Web App) * NexPlayer (Player vidéo) | Visualisation des médias |

Tableau : Catalogue des briques de solution de référence (SBB)

#### Implémentation de la stack technologique

L’implémentation de la stack technologique proposée vise à exploiter au maximum des services déjà disponibles pour faciliter l’implémentation et accélérer les livraisons. L’ensemble des composants déployés devront être redonder afin de garantir la continuité de service en cas de défaillance ou de maintenance sur tout ou partie des composants.

Des composants additionnels (base de données SQL / NoSQL, MoM, Back-end API) seront probablement nécessaires pour compléter la solution (notamment la web app).

Notons toutefois, ces aspects sortent de la définition de l’architecture et seront précisés lors de la rédaction des spécifications techniques que chaque composants.

# ARCHITECTURE CIBLE

# ALIGNEMENT SUR REFERENTIEL D'ARCHITECTURE

* Présentation de l’intégration de la solution dans l’architecture de l’entreprise

## Alignement sur le paysage architectural

## Alignement sur les modèles de référence

## Alignement sur les normes

## Évaluation de la réutilisation

# ANALYSE DES ÉCARTS

-> Analyse des écarts business, logiciels, technique et data …

## Écarts d’architecture

### Business

### Data

### Application

### Technique

# ANALYSE DES IMPACTS

-> Identification des impacts organisationnels, opérationnels, techniques, financiers, juridiques etc …

# TABLES DES RÉFÉRENCES

## Figures

[Figure 1 : Vue synthétique de l'architecture de référence 4](#_Toc99949956)

[Figure 2 : Vue simplifiée des briques de référence d'architecture (ABB) 7](#_Toc99949957)

[Figure 3 : Vue synthétique d'un envoi réalisé avec HLS (source : Eleven Labs Blog) 9](#_Toc99949958)

[Figure 4 : Segmentation d’un média en différentes qualités pour diffusion via HLS (source : Eleven Labs Blog) 10](#_Toc99949959)

[Figure 5 : Fonctionnement d'AWS Elemental Media Converter (Source : AWS) 11](#_Toc99949960)

## Tableaux

[Tableau 1 - Historique des révisions 2](#_Toc100000653)

[Tableau 2 : Catalogue des briques d'architecture de référence (ABB) 7](#_Toc100000654)

[Tableau 3 : Avantages et inconvénients des différentes solutions pour la diffusion de média 14](#_Toc100000655)

[Tableau 4 : Catalogue des briques de solution de référence (SBB) 16](#_Toc100000656)