

Livrable

Architecture technique

Version	1.0
Date	18/09/2016
Auteur(s)	M. Magnier
N° du Lot	4.4, 4.5, 4.6, 4.7



Le lecteur media accessible à tous

Livrable 4.4 : Bibliothèque d'API et outils nécessaires

Livrable 4.5 : spécifications techniques détaillées

Livrable 4.6 : Développement Html5 – architecture

Livrable 4.7 : Tests et intégration

Titre du projet	media4Dplayer
Abréviation	M4DP
Désignation	media4Dplayer, le lecteur media accessible à tous.
Durée du projet	De Janvier 2015 à Juin 2016 – 18 mois
Coordinateur projet	France Télévisions
Partenaires du lot	DOTSCREEN (DTS)
Organisme labellisateur	CAP DIGITAL
Financeurs	La Région Ile-de-France La BPIfrance
Titre de subvention	Fonds Unique Interministériel – FUI18

Le projet media4Dplayer, lecteur media accessible à tous.

Architecture

Date de soumission : 18/06/16

Version : 1.00

Objectif(s) du livrable

Expliquer l'architecture technique de l'application media4Dplayer.

Présenter les différents tests qui ont été mis en œuvre, ainsi que les choix technologiques qui en ont découlés.

Historique	Date	Modification(s)
V 0.9	18/06/2016	Création du document
V 1.0	19/09/2016	Finalisation du document

Le projet media4Dplayer

Media4Dplayer est un projet collaboratif labellisé par le pôle de compétitivité Cap Digital et subventionné au titre du Fonds Unique Interministériel (FUI) par la région Île de France et BPI France. Ce projet de recherche et de développement s'inscrit dans la stratégie de Cap Digital, autour des thématiques d'accessibilité des contenus, de développement numérique et de Silver économie.

Durée de projet 18 mois : Janvier 2015 – Juin 2016

Avertissement

Les informations contenues dans ce document peuvent être sujet à modification sans préavis. Société ou noms de produits mentionnés dans ce document peuvent être des marques ou des marques déposées de leurs sociétés respectives.

Tous les droits sont réservés

Le document est la propriété des membres du consortium media4Dplayer. Aucune copie ou distribution, sous quelque forme ou par tout moyen, n'est autorisée sans l'accord écrit et préalable du (des) propriétaire(s) des droits.

Ce document ne reflète que le point de vue de ses auteurs. Le consortium media4Dplayer et les financeurs ne peuvent être tenus responsables de l'usage qui pourrait être fait des informations contenues dans ce document.

©2016 media4Dplayer

Table Des Matières

1. PRINCIPES D'ARCHITECTURES	7
1.1. L'INTEROPERABILITE DES DIFFERENTS COMPOSANTS	7
1.2. POTENTIEL D'INDUSTRIALISATION A COURT TERME.....	7
1.3. REVERSER LE CODE EN OPEN SOURCE	7
2. ARCHITECTURE.....	8
2.1. ARCHITECTURE GLOBALE	8
2.1.1. SERVEUR DES CONTENUS.....	9
2.1.2. SERVEUR DE RECHERCHE	9
2.1.3. SERVEUR D'HEBERGEMENT DES MEDIAS	9
2.2. ARCHITECTURE DE L'APPLICATION	9
2.2.1. NAVIGATION	11
2.2.2. GESTION DU CATALOGUE	11
2.2.3. REGLAGES.....	11
2.2.4. RECHERCHE.....	11
2.2.5. MODULE PLAYER.....	12
2.3. BIBLIOTHEQUE D'APIS.....	14
2.3.1. JQUERY	14
2.3.2. JQCLLOUD : MIT.....	14
2.3.3. GOOGLE FONT	14
2.3.4. POPCORN.JS :	14
2.3.5. DASH.JS.....	14
2.3.6. WEB AUDIO API (WAA).....	15
2.3.7. PEAK LIMITER	15
2.3.8. ENRICHISSEMENT WAA IRCAM	15
2.3.9. CHROMEVOX ET SHARK	15
2.4. OUTILS D'INTEGRATION.....	15
2.4.1. WAA SANDBOX.....	15
2.4.2. PLAYER DASH.JS DE REFERENCE.....	16
3. COMPOSANTS.....	17
3.1. LIBRAIRIE DE GESTION DES FLUX AUDIO	17
3.1.1. PRINCIPE DE FONCTIONNEMENT	18
3.1.2. SOURCES AUDIO	19
3.1.3. GRAPHE DE ROUTAGE.....	19
3.1.4. AUDIO STREAM DESCRIPTION COLLECTION	20
3.1.5. PROFILS SONORES.....	20
3.2. PLAYER VIDEO	20
3.3. SYNCHRONISATION DES PLAYERS VIDEO	21
3.4. ACCESSIBILITE DE L'APPLICATION.....	21
3.4.1. RGAA	22
3.4.2. INTERFACE RAPIDE ET SIMPLIFIEE.....	22
3.4.3. LECTURE D'ECRAN	22
3.5. PARCOURS DU CATALOGUE.....	23
3.5.1. API /MEDIAS.....	23
3.5.2. API : /MEDIAS/ROOT_ID:EBU100000000	23
3.6. RECHERCHE.....	24

3.6.1.	AUTOCOMPLETE.....	24
3.6.2.	RECHERCHE DE TYPE WORDCLOUD (NUAGE DE MOTS).....	24
3.6.3.	API // RECHERCHE POUR UN GROUPE DE MOTS.....	25
4.	TESTS ET INTEGRATIONS	26
4.1.	LANGAGE.....	26
4.2.	PLAYER.....	26
4.2.1.	HAS PLAYER	26
4.2.2.	JW	26
4.2.3.	DASH.JS VERSION 1.6.0.....	26
4.3.	SYNCHRONISATION DES PLAYERS.....	27
4.4.	PORTABILITE.....	27
	CONCLUSION.....	29
5.	ANNEXE	30

Introduction

L'objectif du media4DPlayer est d'adresser l'ensemble des problématiques d'accessibilités d'une application de lecture de vidéo :

- Accessibilité de l'interface d'une application
- Gestion de la vidéo principale
- Gestion de la vidéo signée (type Langue des Signes Française - LSF) en option et en surimpression de la vidéo principale.
- Intégration d'une librairie de gestion des flux audio pour traiter l'accessibilité correspondante.
- Intégration d'outils d'accessibilité (loupe, lecteur d'écran, ...).
- Parcours de catalogue de contenus

D'un point de vue technologique, la principale difficulté du projet est l'interopérabilité de composants Open Source. Pour chacune des fonctionnalités, il existe une ou plusieurs briques logicielles permettant de répondre, au besoin dans le domaine de l'Open Source. Il s'agit donc d'intégrer un sous-ensemble de solutions compatibles au sein d'une application démontrant la faisabilité du contexte.

1. Principes d'architectures

Les tests et choix d'architecture pour l'application Media4D Player se sont basés sur les principes suivants :

1.1. **l'interopérabilité des différents composants**

Avant toute chose, l'objet du projet est réaliser une application démontrant la faisabilité technique de l'intégration de l'ensemble des problématiques d'accessibilité au sein d'une seule et même application. Les composants techniques intégrés doivent donc pouvoir fonctionner ensemble de manière opérationnelle. Il ne s'agit pas de réaliser une succession de démonstration indépendante.

Dans ce contexte, il a fallu rechercher les différentes alternatives possibles pour chacun des composants techniques.

1.2. **potentiel d'industrialisation à court terme**

L'objectif du projet media4Dplayer est certes une démonstration de faisabilité. Néanmoins on privilégie les solutions ayant un potentielle d'industrialisation à court terme, permettant une exploitation sur le maximum de plateformes possibles (Web, tablette, smartphone), ainsi que sur le maximum d'OS (IOS, Android, Windows, Mac).

1.3. **Reverser le code en Open Source**

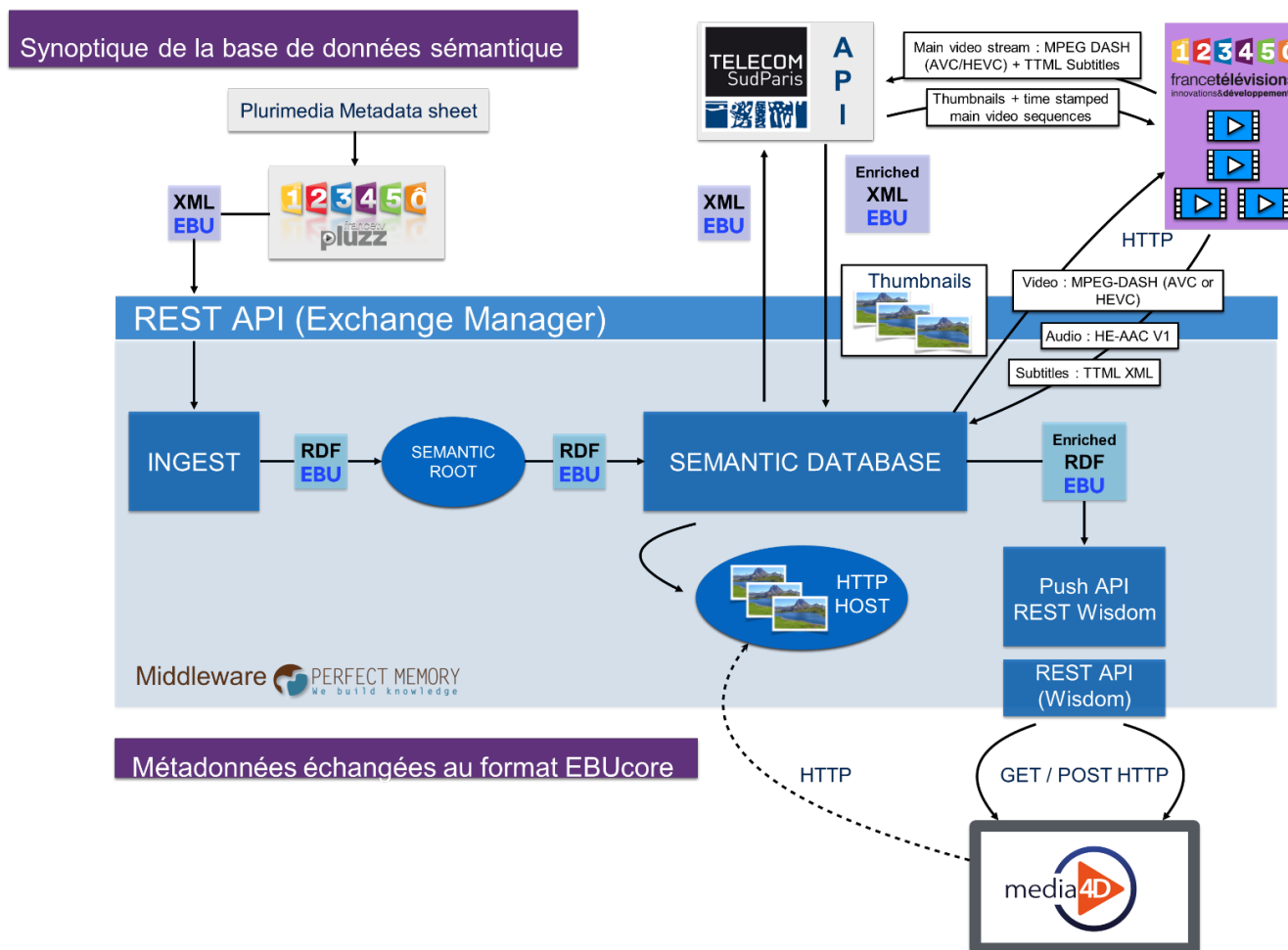
Enfin, il a été convenu au début du projet collaboratif, que l'application sera reversée dans le monde de l'Open Source, afin de promouvoir sa pérennité. Les composants logiciels choisis doivent donc satisfaire à cette contrainte.

2. Architecture

2.1. Architecture globale

Le projet Media4D Player est composé :

- D'une application Media4D.
- D'un serveur d'hébergement des métadonnées des contenus
- D'un serveur d'hébergement des médias.
- D'un serveur pour gestion de la recherche



L'application media4D Player est un site web.

Le code de l'application est donc hébergé sur un serveur d'hébergement Web.

L'application s'exécute dans un navigateur.

2.1.1. Serveur des contenus

Ce serveur héberge l'ensemble des métadonnées du catalogue de contenus disponibles pour l'application Media4D.

Par l'intermédiaire des APIs de Perfect-Memory (<http://www.perfect-memory.com/fr/>) l'application :

- parcourt le catalogue de contenus,
- récupère les métadonnées d'un contenu :
 - Titre
 - Description du contenu
 - Description technique du média (au format EBUCore)
 - Vignette du contenu.

Ce serveur gère également toute la partie ingestion de données sur la base de fichier EBUcore. Cette ingestion permet la gestion du catalogue proposé à l'application Media4D Player. La partie ingestion n'est pas traitée dans ce document.

2.1.2. Serveur de recherche

Ce serveur Telecom SudParis expose les APIs de recherche de contenus utilisés par l'application Media4D Player.

- Auto-complétion
- Regroupement sémantique
- Liste de contenus

2.1.3. Serveur d'hébergement des médias

Ce serveur héberge l'ensemble des médias, au format DASH des différents contenus accessibles par l'application Media4D Player.

2.2. Architecture de l'application

L'application Media4D Player est composée des différents Modules :

- Accueil
- Gestion du catalogue
- Réglage
- Navigation
- Player

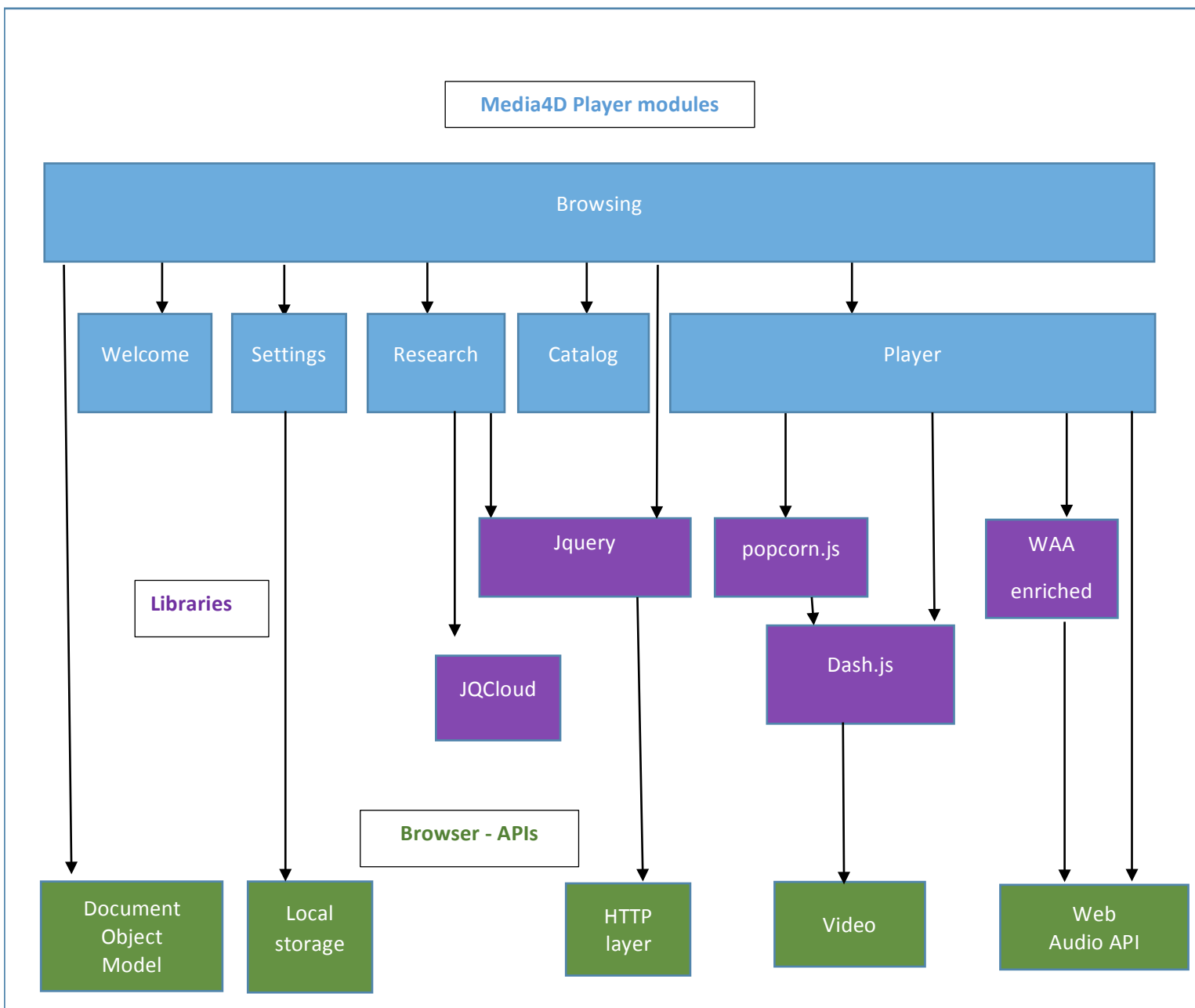
Ces modules sont composés de codes JavaScript, Html et CSS.

Ces différents modules utilisent différentes bibliothèques JavaScript :

- Popcorn.js

- Dash.js
- WAA enrichie
- JQuery
- JQcloud

Enfin, l'ensemble des modules et des librairies JavaScript repose sur les APIs du navigateur qui exécute l'application.



2.2.1. Navigation

Module gérant la navigation au sein de l'application et permettant de faire le lien entre les différents modules applicatifs.

L'application est conçue sur le principe d'application web monopage (Single-page application). Il n'y a donc qu'une seule page web pour l'ensemble de l'application.

Le but est d'éviter le chargement d'une nouvelle page à chaque action demandée, et de fluidifier ainsi l'expérience utilisateur.

La structure HTML de l'application (DOM) est ainsi modifiée dynamiquement par l'application en fonction du module affiché à l'écran. L'application se base sur l'ensemble des APIs du navigateur permettant la gestion dynamique du DOM.

L'application Media4D Player gère 2 types de navigations :

- une navigation standard
- une navigation rapide et simplifiée.

Cette dernière navigation est une version de navigation adaptée à l'accessibilité. Elle dispose de menu épuré et de fonctionnalité d'accessibilité permettant la lecture d'écran.

2.2.2. Gestion du catalogue

Module gérant la navigation dans le catalogue de contenus, ainsi que la récupération de l'ensemble des métadonnées d'un contenu.

Ces données sont récupérées depuis le serveur d'hébergement des contenus au travers d'APIs Rest.

Ce module s'appuie sur la librairie JQuery pour la gestion des requêtes http permettant l'appel des APIs Rest.

2.2.3. Réglages

Le module de réglages traite l'ensemble des fonctionnalités de préférences utilisateurs.

L'ensemble des réglages de chaque utilisateur est stocké dynamiquement au sein du navigateur, permettant de retrouver les derniers réglages à chaque session de l'application.

Le module de réglages se base sur les APIs de local Storage du navigateur.

2.2.4. Recherche

Module de gestion des fonctionnalités de recherche de l'application Media4D Player.

La recherche se base sur les APIs Rest du serveur de recherche. Ces Apis retournent des identifiants de contenus qui font références aux identifiants des contenus du serveur d'hébergement des contenus. Pour afficher les résultats de recherche, le module de recherche utilise les fonctionnalités du module de catalogue.

Le module de recherche se base sur JQuery pour la gestion des requêtes http permettant l'appel des APIs Rest.

2.2.5. Module Player

Module central de l'application Media4D Player. Ce module gère la lecture des contenus.

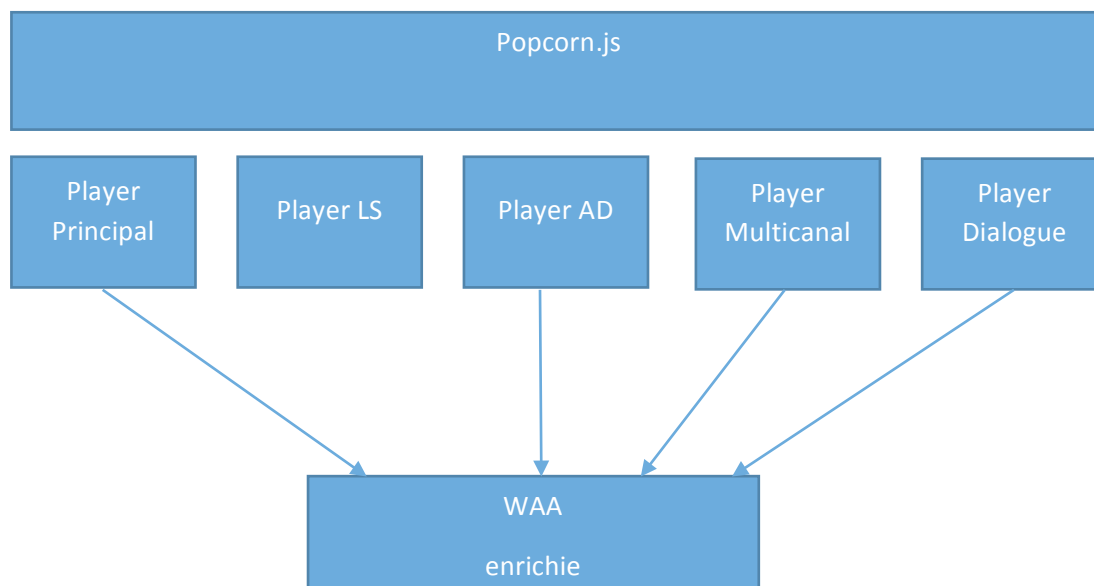
Les contenus de l'application Media4DPlayer sont composés jusqu'à 5 flux :

- Flux principal : La Vidéo principal, avec un son principal en stéréo et sous-titre
- Flux vidéo signée : Une Vidéo signée, sans son.
- Flux audio description : Un flux audio d'Audio description mono, sans vidéo.
- Flux Ambiance : un flux Audio d'ambiance 5.1 (sur 6 canaux), sans vidéo. même contenu audio que le son principal, mais sur 6 canaux. Un canal contient les dialogues, les 5 autres canaux contenant uniquement l'ambiance sonore spatialisée sur 5 canaux.
- Flux dialogue : Un flux dialogue mono, sans vidéo uniquement les dialogues du son principal, sur un seul canal.

Suivant les choix d'activation de l'utilisateur, les différents flux Audio et Vidéo doivent être joués simultanément. Quelques exemples de cas d'usage :

- 1) **L'utilisateur ne demande que la lecture du flux principal** sans activation de sous-titres. Il s'agit du cas standard lorsque l'on regarde la télévision dans son salon. Seul le flux principal est joué par le player, les sous-titres étant désactivés.
- 2) **L'utilisateur demande en plus l'activation des de la vidéo signée** : en plus du flux principal, la vidéo signée doit s'afficher en surimpression de la vidéo principale. L'utilisateur aura le choix de déterminer la taille et la position de la vidéo signée.
- 3) **Ajout de l'audiodescription** : en plus de la vidéo principale et du son principal, de la vidéo signée, le flux audio d'audiodescription doit être mixé avec le son principal. L'audiodescription est conçue pour émettre des commentaires sonores en dehors des plages de dialogue du flux principal.
- 4) **Ambiance 5.1** : l'utilisateur peut choisir de jouer le flux 5.1. Il s'agit du flux Audio correspondant à la vidéo, mais sur 6 canaux, et non sur 2 canaux (stéréo) tel qu'embarqué dans le flux principal. Le player doit alors jouer la vidéo principale (sans le flux audio principal) et jouer le flux audio 5.1, qui doit éventuellement être mixé avec le flux d'audio description si elle est activée.
- 5) **Spatialisation des sons** : l'utilisateur peut choisir de spatialiser l'audiodescription. Concrètement, l'utilisateur peut choisir la direction d'où provient l'audio description. Par exemple, l'utilisateur peut choisir (via les réglages) que l'audiodescription proviendra de sa droite, ou derrière lui, ou enfin au-dessus de lui. Le player doit alors effectuer un traitement au niveau du rendu des sons, pour que l'utilisateur ait la sensation auditive que l'audiodescription lui parvient depuis un angle (0-360) et une hauteur qu'il a définie dans les réglages.
- 6) **Renforcement des dialogues** : dans le cadre de l'accessibilité, il peut être utile de pouvoir rehausser le volume sonore des dialogues (afin de mieux les comprendre), sans toutefois rehausser l'ensemble des autres flux audio (audio description et ambiance). Il s'agit là d'un rehaussement sonore sélectif. Dans ce cas de figure, le player doit jouer la vidéo principal sans le son principal, jouer le flux dialogue et l'ensemble des autres flux sonore sélectionné par l'utilisateur (ambiance sans dialogue et audio description). L'ensemble des flux sonores doit être mixé, et les dialogues doivent être rehaussés par rapport aux autres flux sonores.

Chacun des contenus disponibles pour l'application Media4D Player ne dispose pas nécessairement l'ensemble des 5 flux. Le player est donc en charge de gérer la combinatoire des choix d'activation de l'utilisateur, en fonction des flux disponibles pour le contenu.



Pour réaliser l'ensemble de ces fonctionnalités, le module player instancie 5 players audio/vidéo. Chaque player audio/vidéo étant en charge d'un flux. Nous avons ainsi :

- Le player principal qui joue le flux principal (video principale, son principal, sous-titre)
- Le player LS qui joue la vidéo signée.
- Le player Audio description qui joue le flux audio d'audio description.
- Le player multicanal, qui joue le flux audio d'ambiance.
- Le player dialogue, qui joue le flux audio de dialogue

Seuls les deux premiers players jouent un flux vidéo. Les 3 derniers players ne jouent que des flux audio.

Chacun de ces players est une instance du player Dash.js, qui est une librairie incluse au projet. (Voir définition des librairies)

L'ensemble des fonctionnalités audio du module player est assuré par la Web Audio API :

- Mixage des différents flux audio
- Spatialisation de certains flux
- Traitement sonore (rehaussement des dialogues, gestion du volume total, ...)

Cette Web Audio API est enrichie pour les besoins du projet.(voir chapitre composant WAA)

Les sorties de chacun des players (sauf le player LS qui ne joue aucun flux audio) sont redirigés vers la WAA enrichie afin qu'elle puisse effectuée l'ensemble des traitements sonore, et produire la sortie sonore finale.

Enfin, jouer 5 flux Audio/vidéo simultanée pose le problème de la synchronisation des flux. La librairie popcorn.js permet de piloter l'ensemble des players

2.3. Bibliothèque d'APIs

2.3.1. jQuery

Bibliothèque JavaScript Open source, permettant entre autres de gérer les requêtes http, de gérer certaines animations graphiques, de gérer dynamiquement le DOM pour une application the type SinglePage.

Licence de type MIT.

<https://jquery.com/>

2.3.2. jQCloud : MIT

Plugin JQuery permettant d'afficher élégamment des nuages de mots. Ce Plugin est utilisé essentiellement pour la recherche.

Licence de type MIT.

<http://mistic100.github.io/jQCloud/>

2.3.3. Google Font

Font Open source.

<https://fonts.google.com/about>

2.3.4. Popcorn.js :

Framework de gestion de Media HTML5. Principalement utilisé ici pour la synchronisation des différents Player de l'application.

Licence de type MIT.

<http://popcornjs.org/>

2.3.5. Dash.js

Librairie JavaScript open source permettant la lecture de flux DASH, implémentant l'extension Media Source Extension (MSE) définit par le W3C.

Licence de type BSD.

<http://dashif.org/software/>

2.3.6. Web Audio API (WAA)

La Web Audio API permet manipuler le contenu audio dans une application web. Elle permet de manipuler mixages audio, effets, balance, etc.

BSD-3-Clause license

https://developer.mozilla.org/fr/docs/Web/API/Web_Audio_API

2.3.7. Peak limiter

Nœuds de la Web Audio API permettant de fournir un gain supplémentaire tout en empêchant le signal de sortie d'être coupé.

<https://github.com/tcarpent/PeakLimiter>

2.3.8. Enrichissement WAA IRCAM

L'IRCAM a développé des nœuds spécifiques pour la WAA, afin de gérer des cas d'usage pour l'application Media4D Player :

smart fader

Dialog enhancement

receiver mix

<https://github.com/Ircam-RnD/m4dp-audio-modules.git>

2.3.9. ChromeVox et Shark

Chromevox et shark sont des applications de lecture d'écran. Elles permettent de vocaliser le contenu de l'écran, rendant accessible l'application aux non-voyants.

A proprement parler, ces applications ne sont pas des bibliothèques, et ne sont donc pas intégrées à l'application media4D Player.

ChromeVox est une extension de chrome. Elle doit être installée par l'utilisateur dans son navigateur Chrome grâce au menu dédié de Chrome.

Shark est une application indépendante. Elle doit être téléchargée et installée sur la machine de l'utilisateur, au niveau de son système d'exploitation.

2.4. outils d'intégration

2.4.1. WAA Sandbox.

La WAA Sandbox est un outil de tests de la WAA enrichie.

C'est un outil d'intégration des nouveaux nœuds de la Web Audio API.

La Sandbox propose une interface simplifiée, permettant de tester les différents cas d'usage de la WAA sur un contenu dont l'URL de la description est fournie en paramètre.

Cette Sandbox est hébergé <https://github.com/Ircam-RnD/m4dp-audio-modules.git>

Pour tester la Sandbox :

<http://staging.dotscreen.com/media4d/waa/examples/index.html>

2.4.2. Player dash.js de référence

Afin de tester les différents flux Dash

<http://dashif.org/reference/players/javascript/v1.6.0/samples/dash-if-reference-player/index.html>

3. Composants

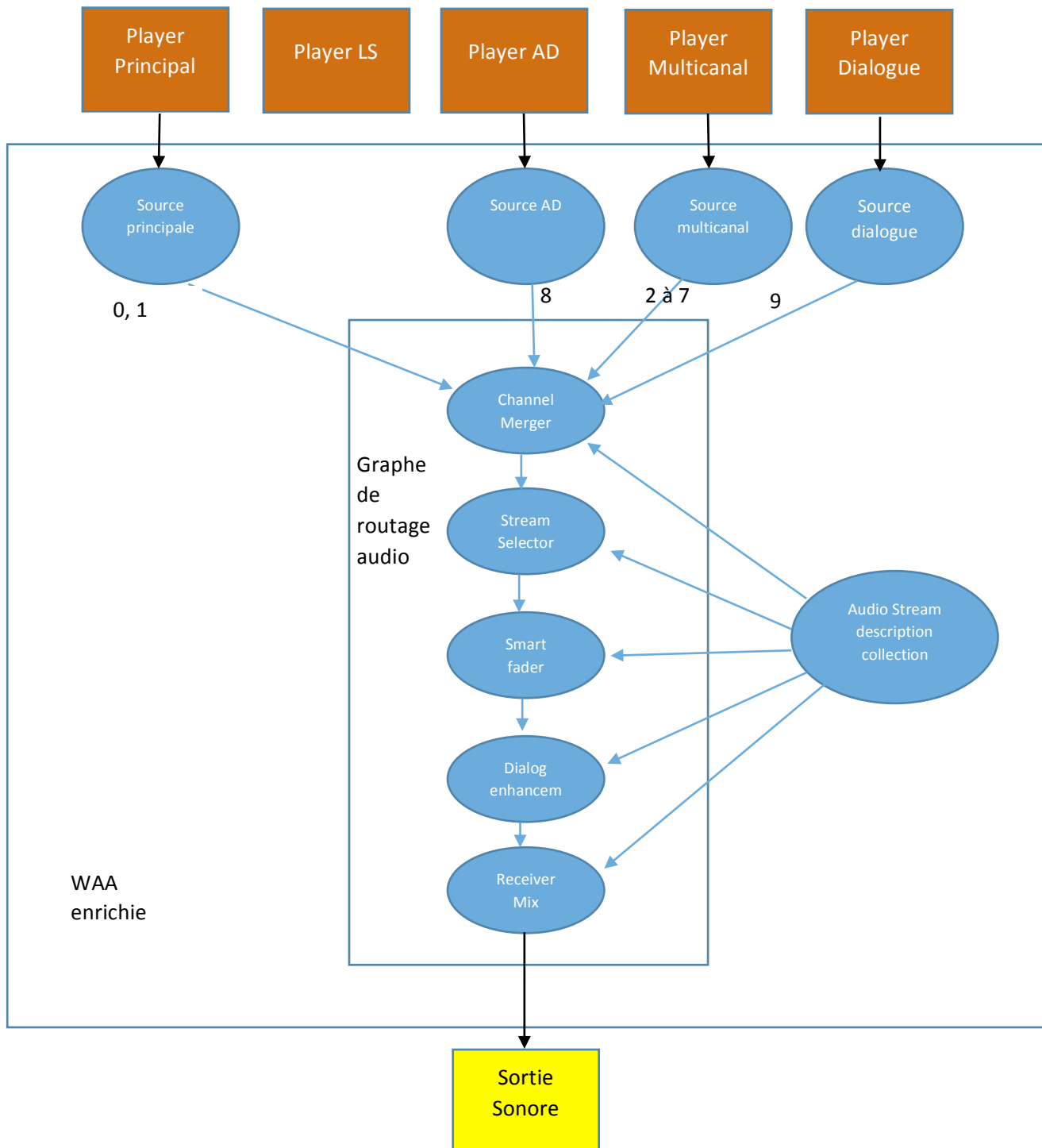
3.1. Librairie de gestion des flux audio

Une recherche de l'existant à montrer que la librairie de gestion des flux audio satisfaisant les principes d'architecture logicielle retenus était le facteur le plus structurant. Une seule solution était disponible : La Web Audio API. <https://webaudio.github.io/web-audio-api/>

La Web Audio API (WAA dans la suite du document) est une librairie JavaScript permettant la synthèse et le traitement des flux Audio au sein d'une application JavaScript. Cette librairie est spécifiée par l'organisme W3C. Elle est intégrée aux différents navigateurs du marché, en fonction de leur niveau de compatibilité.

A la date du démarrage du projet, la WAA standard ne permettait pas de répondre à l'ensemble des fonctionnalités du projet media4Dplayer. Pour les besoins du projet, un développement spécifique, a donc été nécessaire, afin d'enrichir la WAA Api. Ce développement a été pris en charge par l'IRCAM.

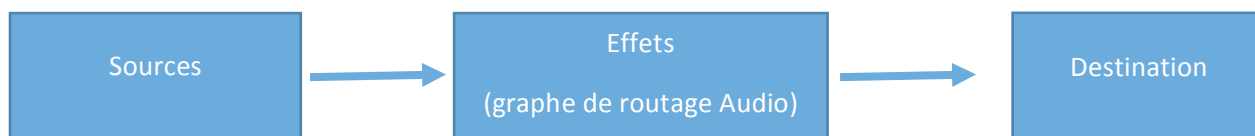
Pour permettre une meilleure interopérabilité de la solution, l'enrichissement de la librairie se situe au niveau de la couche JavaScript. Un enrichissement de la couche native (C/C++) aurait nécessité l'ajout de « plugin » au sein du navigateur, ou alors de recompiler un navigateur indépendant.



3.1.1. Principe de fonctionnement

Les opérations audio sont réalisées via des **nœuds audio** reliés entre eux pour former un **graphe de routage audio**. Plusieurs sources, avec différents types d'agencements de canaux, peuvent être supportées, même

dans un seul contexte. Ce design modulaire et flexible permet de créer des fonctions audio complexes avec des effets dynamiques.



3.1.2. Sources Audio

Chacun des players (sauf le player LS) est associé à une source de la WAA. Cette association permet de mapper les canaux correspondant aux flux du player.

- Source principal : canaux 0, 1 (car le player principal dispose d'un flux principal stéréo, donc 2 canaux)
- Source Multicanal : canaux 2 à 7 (car le player d'ambiance dispose des 6 canaux du flux audio 5.1)
- Source AD : Canal 8 (car le flux d'audiodescription est mono, donc 1 canal)
- Source Dialogue : canal 9 (car le flux audio dialogue est mono)

Le nombre de canal associé à chaque flux audio n'est pas dynamique. Cela nécessite que l'encodage des différents flux du média (principale, ambiance, ...) soit fait en conséquence. Pour le projet Media 4D player, les flux de chacun des contenus respectent la définition du nombre de canaux ci-dessus.

3.1.3. Graphe de routage

Le graphe de routage est composé de l'ensemble des nœuds audio qui permettent de réaliser les traitements sonores.

Les nœuds du graphe de routage de l'application Media4D player proviennent à la fois des nœuds standards de la WAA, ainsi que des nœuds qui ont été ajoutés spécifiquement pour ce projet et qui sont contenus dans la librairie WAA enrichie ((receiver Mix, Peak Limiter, smart fader, ...). Pour plus de détails sur les nœuds spécifiques ajouter par la WAA enrichie, se reporter à la documentation technique correspondante.

Chacun des nœuds audio disposent d'un ensemble d'API qui permet à l'application de spécifier le comportement des nœuds.

C'est grâce à ces Apis que l'application :

- Fournit le réglage du volume sonore final (défini par l'utilisateur grâce à l'interface du module player) au nœud « smart fader »
- Active ou désactive le renforcement des dialogues (défini par l'utilisateur dans le module réglage de l'application) au nœud Dialog Enhancement.
- Fournit les paramètres de spatialisation, angle et hauteur (défini par l'utilisateur dans le module réglage de l'application) au nœud the spatialisation.

- Affiche dans le module player le fait que la compression sonore ait atteint le seuil au-delà duquel toute hausse de volume implique une perte de qualité.
- ...

3.1.4. Audio stream description collection

L'audio stream description collection permet de passer à l'ensemble du graphe de routage audio, les paramètres métiers concernant chaque flux audio, ainsi que l'activation de ces flux.

- Loudness : paramètre loudness du flux audio
- Max True Peak : paramètre max True Peak du flux audio
- ...

Ces paramètres proviennent des métadonnées du contenu (au format EBUcore), fournies par le serveur des contenus et récupérées par le module de catalogue.

3.1.5. Profils sonores

Grâce à une API proposée par l'IRCAM, l'application Media4D player permet de télécharger des profils sonores. Ces profils sonores améliorent la qualité de la spatialisation au casque (mode binaural).

3.2. Player Vidéo

Le projet Media4D gère des médias au format MPEG-DASH.

Le player choisi pour lire ces contenus est le player Dash.js. À l'initiative du DASH industry forum, le player Dash.js est une bibliothèque JavaScript open source, permettant de lire des contenus MPEG-DASH sur différents navigateurs Web. Ce player repose sur les Media Source Extension API, définies par le W3C.

<https://github.com/Dash-Industry-Forum/dash.js/wiki>

Pour le projet Media 4D player, nous avons utilisé la version 1.6.0 du player Dash.js. Ce player continue d'évoluer. La dernière version connue à date étant la version 2.3.0

Ce player dispose d'un ensemble d'APIs permettant de contrôler le player :

- Définition de l'URL du flux à jouer. Dans le cadre du media4D player, les URL des flux proviennent des métadonnées du contenu (au format EBUcore), fournies par le serveur des contenus et récupérées par le module de catalogue
- Affichage des sous-titres
- Positionnement du démarrage de la lecture du contenu (SEEK)
- Événement de bufferisation du flux
- ...

Ces 2 dernières APIs étant particulièrement utiles pour la synchronisation des flux joués en parallèles. Voir chapitre sur la synchronisation des flux.

3.3. Synchronisation des Players vidéo

L'application Media4D player instancie 5 players, afin de jouer jusqu'à 5 flux audio/vidéo en parallèle. Voir chapitre Module player.

Pour que l'expérience utilisateur soit acceptable, il faut que ces flux soient finement synchronisés. Il faut en effet éviter :

- Que les flux audio soient désynchronisés des flux vidéo. Pour éviter que les dialogues ne soient pas calés avec la vidéo.
- Que les différents flux audio soient désynchronisés entre eux. Pour éviter par exemple que les sons d'ambiance ne soient pas calés avec les dialogues
- Que certains flux ne soient pas encore (ou plus) bufferisés en cours de lecture.

L'application Media4D player utilise les APIs de la librairie Popcorn.js pour synchroniser l'ensemble des player Vidéo.

Le principe de synchronisation repose sur les principes suivants :

- Le player principal est déclaré comme player « maître », les autres players étant déclarés comme esclaves
- Au lancement de la lecture, on vérifie l'état de bufferization de chacun des player. Quand tous les players sont prêts, on déclenche la demande de lecture à l'ensemble des player de manière synchrone.
- A intervalle régulier, on vérifie que chacun des player en ait au même niveau de lecture (même position d'avancement dans le flux). En cas de désynchronisation, les player sont recalés sur le player « maître »

3.4. Accessibilité de l'application

L'objectif de l'application Media4D Player, est de rendre accessible à tous. L'application Media 4D player implémente un ensemble de solutions techniques visant la réalisation de cet objectif.

Une partie de ces solutions mises en œuvre concernent la lecture des médias au niveau du module player :

- Affichage des sous-titres pour l'accessibilité auditive
- Renforcement des dialogues pour l'accessibilité auditive.
- L'affichage d'une vidéo en langue des signes pour l'accessibilité auditive
- Gestion de la taille des polices dans l'application et dans les sous-titres pour l'accessibilité visuelle.
- Gestion de la couleur des sous-titres pour l'accessibilité visuelle
- ...

Ce chapitre traite de la gestion de l'accessibilité de l'application elle-même, en dehors du module player. Elle se concentre principalement sur la gestion de la navigation, et de la possibilité d'interfacer les principaux outils d'accessibilité.

3.4.1. RGAA

Le (RGAA) est un ensemble de documents administratifs disponible sur le site du secrétariat général pour la modernisation de l'action publique, SGMAP, indiquant les dispositions à suivre pour rendre les pages web des sites publics accessibles à tous.

Le RGAA ne constitue pas une norme en lui-même. Il s'agit d'un référentiel opérationnel pour vérifier l'application des règles internationales pour l'accessibilité des contenus web, les [WCAG 2.0](#).

Le WCAG définit un ensemble de tests permettant de vérifier les conformités d'une application aux normes RGAA. Il existe 3 niveaux de qualités : A, AA, AAA. Le niveau le plus bas, A, définit un ensemble de 128 tests.

<http://www.rgaa.net/+mot4+.html>

Suite aux travaux réalisés en collaboration avec le laboratoire Les Lutins, et à l'intervention d'un ergonomiste non voyant, nous avons optés pour la mise en place d'une interface rapide et simplifiée.

3.4.2. Interface rapide et simplifiée

Au début de l'application, il est proposé à l'utilisateur de choisir entre une interface standard ou une interface rapide et simplifiée.

Cette interface rapide et simplifiée propose une navigation simplifiée, basée sur les principes suivants :

- Chaque page ne contient que les informations nécessaires
- Les choix de navigation se présente sous la forme liste textuelle uniquement
- L'ensemble de la navigation est réalisable au clavier, en utilisant les touches standards (tabulation, espace, entrée, echap, ...)
- Pas d'utilisation d'images

Ces choix d'interface permettent de fait la compatibilité avec la plupart des règles du WCAG2.0 qui concernent la gestion des images, les gestions des formulaires et autres composants HTML qui ne sont pas utilisés

Dans l'interface standard, toutes ces règles ne sont pas respectées. Elles auraient représentés un énorme travail de compatibilités qui ne rentrerait pas dans le budget alloué. Par ailleurs, le projet Media4D player a plus une vocation de démonstrateur technologique et de potentiel d'industrialisation.

Enfin, et c'est là le plus important, la présence de l'interface rapide et simplifiée respecte l'ensemble des tests du WCAG2.0. L'application Media4D Player être considéré comme satisfaisant le RGAA à part entière.

3.4.3. Lecture d'écran

La compatibilité de l'application Media4D Player avec les principales applications de lecture d'écran du marché, Jaws et chromevox par exemple, est rendu grâce l'emploi de propriété des différentes balises HTML5 qui composent l'application dans l'interface rapide et simplifiée.

- ajout de l'attribut "title" sur certain élément quand le texte de celui-ci n'est pas pertinent
- ajout d'un sélecteur (bordure noir) quand l'élément à le focus (pour être compatible avec JAWS)

- utilisation de l'attribut "tabindex" pour naviguer avec la touche TAB.
- utilisation des bonnes balises selon les normes RGAA
- utilisation d'attributs ARIA sur certains éléments : role, aria-disabled, aria-valuemin, aria-valuemax, aria-valuenow, aria-valuetext, aria-label, aria-labelledby, aria-hidden, aria-checked
-
- role = radio/radiogroup/slider/button/tooltip - précise le rôle de la balise
- aria-disabled = false - indique au lecteur d'écran que l'élément est désactivé
- aria-valuemin = {number} - valeur mini d'un slider
- aria-valuemax = {number} - valeur max d'un slider
- aria-valuenow = {number} - valeur actuelle d'un slider
- aria-valuetext = la valeur qui devra lire le lecteur d'écran
- aria-label = le texte que devra lire le lecteur d'écran quand un titre à le focus
- aria-labelledby = contient l'élément HTML dont le lecteur d'écran devra lire le contenu à la place.
- aria-hidden = true - indique au lecteur d'écran que l'élément est invisible
- aria-checked = true/false - permet au lecteur d'écran d'utiliser les mots appropriés pour des éléments checkbox.

3.5. Parcours du catalogue

Le parcours de catalogue se base sur les APIs exposés par la plateforme Perfect Memory qui constitue le serveur des contenus.

La plateforme Perfect Memory est bien plus riche que l'utilisation qui en est faite dans l'application Media4D Player. En particulier, elle permet de faire de la recherche sémantique qui n'a pas été intégré au projet.

Voir la documentation Perfect Memory pour plus de détails.

Voir également les exemples d'APIS en annexe.

On utilise en particulier 2 APIs

3.5.1. API /medias

Cette API permet de retourner l'ensemble des contenus du catalogue. Un certain nombre de critères de filtres permette de sélectionner une sous-partie du catalogue.

Cette API renvoie une liste de rootID

3.5.2. API : /medias/root_id:EBU100000000

Permet de récupérer l'ensemble des métadonnées d'un contenu, à partir de son rootID. Parmi les données renvoyées :

- Titre
- Description

- Description longue
- Type de programme
- Vignette
- Ensemble des paramètres métiers des médias

Si la structure retournée est un JSON, les paramètres métiers sont un extrait du fichier EBUcore qui a permis l'ingestion des données dans la plateforme Perfect Memory. On y trouve pour chacun des flux disponibles du média :

- L'url du manifest Dash du flux. Utilisé par le player dash.js
- La langue des flux audio. Pour interface du module player et catalogue
- Type de flux (principal, LSF, Audiodescription, Ambiance,...) pour les modules player, catalogue , recherche (filtrage de la recherche sur la présence d'audio description par exemple)
- Paramètres métier des flux audio. Pour initialisation des nœuds audio de la WAA

3.6. Recherche

La recherche dans le catalogue se base sur les APIs exposées par le serveur de recherche développé par Telecom Sud Paris.

Le principe de cette recherche :

- Se base sur les sous-titres des contenus
- Permet de regrouper les mots sous-titre en nuages de mots ayant un rapport sémantique.

Voir en annexe pour des exemples de requêtes.

3.6.1. Autocomplete

Permet de fournir une liste de mots sur la base des premières lettres saisies par l'utilisateur.

Ces mots peuvent être ensuite utilisés dans les autres APIs de recherche.

3.6.2. Recherche de type wordCloud (nuage de mots)

Cette APIs permet de distinguer la sémantique de 2 mots. Par exemple, en fournissant le mot Europe, l'API retourne une liste de groupes de mots qui permet de distinguer les différents usages du mot Europe dans les sous-titres des contenus.

On pourra ensuite distinguer « Europe » dans le cadre d'une affaire politique, ou Europe dans le cadre de la coupe d'Europe de football.

Dans le premier cas, le nuage de mots contiendrait « Europe », « politique », « CEE », « Bruxelles », ...

Dans le second cas, le nuage contiendrait « Europe », « football », « victoire », « Madrid », ...

3.6.3. API // Recherche pour un groupe de mots

En utilisant le groupeID de l'API précédente, cette API renvoie les identifiants des contenus dont les sous-titre contiennent le mot recherché (ex Europe) dans le contexte sémantique choisi par l'utilisateur (ex : Europe, politique, Cee, Bruxelles,...)

Cette APIs renvoie également la liste des séquences (position de début dans le flux) correspondant à l'emploi de ce mot. Ainsi, le module player pourrait directement lancer la lecture du contenu à la position qui correspond à l'emploi de ce mot.

Pour afficher récupérer les métadonnées des contenus renvoyer par cette APIs, l'application Media4D player utilise l'API de catalogue « /media/rootid » avec l'identifiant du contenu comme paramètre.

4. Tests et intégrations

4.1. Langage

Pour le projet media4Dplayer, deux grands choix de langages se sont dégagés :

- Une solution de type site Web
- Une solution de type application, avec développement natif

Bien qu'offrant l'avantage de permettre une diffusion via les « stores » des différentes plateformes, la seconde solution offrait un certain nombre d'inconvénients.

- Multiplier les nombres de développement par plateforme qui n'aurait pas pu être réellement adressé dans le planning imposé
- Par ailleurs, l'utilisation de la WAA a été très structurant sur le projet, imposants une solution de type site Web.

4.2. Player

Plusieurs players ont été testés, voir intégrer au sein de l'application Media4D player

4.2.1. Has Player

Has player est une branche de Dash.js. Ce player offre la possibilité des flux au format HLS, en plus des flux MPEG DASH.

Has player a été intégré au sein de l'application Media4D Player. Néanmoins, à la date de l'intégration, nous avons rencontré un certains de difficultés pour les fonctionnalités liées aux sous-titres.

Par ailleurs, le layer Dash.js est en avance sur le projet Has Player. Les nouvelles fonctionnalités de Dash.js sont progressivement intégrées dans les version successives de Has Player.

4.2.2. JW

JW est un player natif permettant de jouer des flux MPEG DASH. Il offre la possibilité de développer relativement une couche de synchronisation Multi-player.

Néanmoins, après étude, son intégration au sein d'une architecture Web s'est avérée trop complexe. JW est développé en C/C++

4.2.3. Dash.js version 1.6.0

Différentes version de Dash.JS ont été délivrée au cours du développement du projet Media4D player.

Néanmoins, la mise à jour de la version 2.0 du player avait engendré un certain nombre de régressions, notamment avec le mode de synchronisation « Media Controller » qui était employé à ce moment.

Pour éviter tout type de problème de régression, la version de Dash.js a été fixé à 1.6 dans l'application Media4D player.

Cependant, suite à l'abandon de la méthode de synchronisation Media Controller, il est probable que les dernières versions du player Dash.js soient tout à fait compatibles avec l'application.

4.3. Synchronisation des players

Les premières versions de synchronisation des players étaient basées sur le composant Media Controller. Ce composant est défini au niveau du W3C, et promettait donc de devenir un standard, implémenté dans l'ensemble des navigateurs du marché.

Ce composant offrait l'avantage de gérer la synchronisation des player dans les couches basses du navigateur. De ce fait, il anticipait beaucoup mieux les problématiques de pénurie de bufferization en cours de lecture, et permettait une synchronisation très fines (inférieur à 5 ms) des différents players.

A date d'intégration, ce composant n'était développé que sur les navigateurs Chrome.

Etant le seul à avoir implémenté ce composant après plusieurs mois, Chrome a décidé de supprimer le Media Controller à partir de la version 48.

Il est relativement complexe d'empêcher la mise à jour du browser Chrome. De ce fait, l'utilisation de l'application Media4D player est devenue rapidement inutilisable au fur et à mesure de la mise à jour du navigateur Chrome dans le parc des appareils du marché.

Après quelques vaines tentative de recompiler par nous même une version de navigateur Chrome embarquant le Media Controller, qui nous aurait permis de fournir une application (et non un site Web) nous avons donc abandonné ce mode de synchronisation.

Cette piste reste toutefois ouverte pour de futures évolutions.

Une branche du code contenant le Media Controller est toutefois présente dans le Git hébergeant l'application Media4D Player, et une version est toujours hébergée sur les serveurs de Dotscreen.

4.4. Portabilité

A date, la portabilité de la solution Media 4D player a été évaluée sur les différents navigateurs et différents OS du marché.

Pour les navigateurs, le principal problème provient de l'implémentation de la WAA sur l'ensemble des navigateurs. A date, bien que défini au niveau du W3C, seul Chrome semble supporter l'ensemble des fonctionnalités de la WAA nécessaire à l'application Media 4D.

Sur le navigateur Chrome, l'application Media4D player a été testée avec succès sur différents OS :

- Windows 10 (tablette et PC)
- Android (tablette et PC)

Néanmoins, la WAA semble être devenu un standard établi au niveau du W3C. La plupart des navigateurs l'implémentent partiellement. Nous pouvons raisonnablement espérer que la compatibilité de la WAA sur les différents navigateurs va aller en augmentant. C'est du moins la tendance actuelle.

Conclusion

L'objectif de l'application Media4D player nous semble avoir été atteint, à savoir adresser l'ensemble des problématiques d'accessibilités d'une application de lecture de vidéo :

- Accessibilité de l'interface d'une application
- Gestion de la vidéo principale
- Gestion de la vidéo signée (type LSF) en option et en surimpression de la vidéo principale.
- Intégration d'une librairie de gestion des flux audio pour traiter l'accessibilité correspondante.
- Intégration d'outils d'accessibilité (Loupe, lecteur d'écran, ...).
- Parcours de catalogue de contenus

Le stade d'industrialisation nécessite cependant encore quelques approfondissements :

- La technologie de synchronisation des players manquent encore d'efficacité
- La portabilité sur l'ensemble des navigateurs du marché n'est pas encore complète, même si nous pouvons espérer des améliorations rapides
- L'utilisation de 5 players en parallèle sur la même machine induit des problèmes de performance sur les appareils les moins performants

L'ensemble des problématiques d'accessibilité d'une application de lecture de Vidéos de ont été implémentés dans le démonstrateur que constitue l'application Media4D Player, sur la base de technologie Open source et pérennes.

5. Annexe

API : `/medias/root_id:EBU100000000?auth_token=__ -`

```
{
  • id: "5771605372ff340025fe8948",
  • label: "Des volcans et des hommes",
  • description: "Le Monde de Jamy est une collection de documentaires inédits sur le thème du savoir et de la découverte qui montre le spectacle de la terre dans toute sa diversité sous le regard curieux et complice de Jamy Gourmaud que les téléspectateurs de France 3 connaissent bien.",
  • created_at: "2016-06-27T17:20:19.709Z",
  • captured_at: "2015-02-16T00:00:00.000+00:00",
  • types:
    [
      o "audiovisual_programme",
      o "main_editorial_object",
      o "movie",
      o "documentary"
    ],
  • status: "ready",
  • scope: "private",
  • source: "upload",
  • language: null,
  • root_id: "EBU100000000",
  • owner_id: null,
  • referenced_in_ids: [ ],
  • rate_count: 0,
  • rating: 0,
  • duration: 589.96,
  • thumbnails:
    [
      o {
        ▪ width: 960,
        ▪ height: 540,
        ▪ label: "default",
        ▪ url: "http://cloud.ftv.perfect-memory.com/ftv/9/9fdf545b-58bb-45f1-8940-ad313e33f5d9.png"
      }
    ],
}
```

- **uri**: "<http://www.perfect-memory.com/profile/ftv/resource/46343dfe-4b39-488c-aacc-88f53e86fa93>",
- **user_rating**: null,
- **proxies**:
 - [
 - o {
 - **url**: "http://videos-pmd.francetv.fr/innovation/media4D/Le_Monde_de_Jamy/EBU100000000/manifest-ad.mpd",
 - **format**: "XML",
 - **label**: "AD"
 - },
 - o {
 - **url**: "http://videos-pmd.francetv.fr/innovation/media4D/Le_Monde_de_Jamy/EBU100000000/manifest-ea3.mpd",
 - **format**: "XML",
 - **label**: "EA3"
 - },
 - o {
 - **url**: "http://videos-pmd.francetv.fr/innovation/media4D/Le_Monde_de_Jamy/EBU100000000/manifest.mpd",
 - **format**: "XML",
 - **label**: "Main"
 - },
 - o {
 - **url**: "http://videos-pmd.francetv.fr/innovation/media4D/Le_Monde_de_Jamy/EBU100000000/manifest-lsf.mpd",
 - **format**: "XML",
 - **label**: "SL"
 - },
 - o {
 - **tcin**: 0,
 - **tcout**: 589.96,
 - **url**: "http://videos-pmd.francetv.fr/innovation/media4D/Le_Monde_de_Jamy/EBU100000000/JAMY_CUT-output_video2_1.2Mbps.mp4",
 - **format**: "MP4",
 - **quality**: "LQ",
 - **label**: "video"

```

    }

  ],

  • properties:

  [

    ○ {
      ▪ id: "57ba9f9dab92210026d1b6a4",
      ▪ value: true,
      ▪ predicate:

      {

        ▪ id: "57ba9ec572ff340022f5ea7e",
        ▪ label: "A une audio description",
        ▪ multiline: false

      },

      ▪ type: "boolean"

    },

    ○ {
      ▪ id: "57ba9f9dab92210026d1b6a5",
      ▪ value: "EBU100000000",
      ▪ predicate:

      {

        ▪ id: "5730ab8372ff3400250e6c54",
        ▪ label: "UID FTV",
        ▪ multiline: false

      },

      ▪ type: "string"

    },

    ○ {
      ▪ id: "57ba9f9dab92210026d1b6a6",
      ▪ value: "100000000",
      ▪ predicate:

      {

        ▪ id: "5730ab8372ff3400250e6c6c",
        ▪ label: "ID Plurimedia",

```



```

    ▪ multiline: false
  },

  ▪ type: "string"
},

o {
  ▪ id: "57ba9f9dab92210026d1b6a7",
  ▪ value: 0,
  ▪ predicate:

  {

    ▪ id: "574ee16672ff3400250e7133",
    ▪ label: "Rating",
    ▪ multiline: false

  },

  ▪ type: "int"
},

o {
  ▪ id: "57ba9f9dab92210026d1b6a8",
  ▪ value: "[{"formatName":"Main","trackLanguage":"FRA","type":"Stereo","dialog":true,"ambiance":true,"commentary":false,"loudness":-22.1,"maxTruePeak":-5.1,"locator":"http://videos-pmd.francetv.fr/innovation/media4D/Le_Monde_de_Jamy/EBU100000000/manifest.mpd","subtitle":true},{"formatName":"SL","locator":"http://videos-pmd.francetv.fr/innovation/media4D/Le_Monde_de_Jamy/EBU100000000/manifest-lsf.mpd","subtitle":true},{"formatName":"AD","trackLanguage":"FRA","type":"Mono","dialog":false,"ambiance":false,"commentary":true,"loudness":-23.2,"maxTruePeak":-6.0,"locator":"http://videos-pmd.francetv.fr/innovation/media4D/Le_Monde_de_Jamy/EBU100000000/manifest-ad.mpd","subtitle":false},{"formatName":"DI","trackLanguage":"FRA","type":"Mono","dialog":true,"ambiance":false,"commentary":false,"loudness":-22.7,"maxTruePeak":-6.1,"locator":"http://videos-pmd.francetv.fr/innovation/media4D/Le_Monde_de_Jamy/EBU100000000/manifest-di.mpd","subtitle":false},{"formatName":"EA3","trackLanguage":"UND","type":"MultiWithLFE","dialog":false,"ambiance":true,"commentary":false,"loudness":-19.6,"maxTruePeak":-5.9,"locator":"http://videos-pmd.francetv.fr/innovation/media4D/Le_Monde_de_Jamy/EBU100000000/manifest-ea3.mpd","subtitle":false}]",
  ▪ predicate:

  {

    ▪ id: "57695f8b72ff340026f181f1",

```

```

    ▪ label: "Metadata",
    ▪ multiline: true
  },

  ▪ type: "string"
},

o {
  ▪ id: "57ba9f9dab92210026d1b6a9",
  ▪ value: 1,
  ▪ predicate:
    {
      ▪ id: "5730ab8372ff3400250e6c71",
      ▪ label: "Numéro d'épisode",
      ▪ multiline: false
    },

  ▪ type: "int"
},

o {
  ▪ id: "57ba9f9dab92210026d1b6aa",
  ▪ value: 2015,
  ▪ predicate:
    {
      ▪ id: "5730ab8372ff3400250e6c9a",
      ▪ label: "Année de production",
      ▪ multiline: false
    },

  ▪ type: "int"
},

o {
  ▪ id: "57ba9f9dab92210026d1b6ab",
  ▪ value: 1,
  ▪ predicate:
    {
      ▪ id: "5730ab8372ff3400250e6c9c",

```

```

    ▪ label: "Numéro de saison",
    ▪ multiline: false
  },

  ▪ type: "int"
},

o {
  ▪ id: "57ba9f9dab92210026d1b6ac",
  ▪ value: "-Les volcans nous fascinent. Surgis des entrailles de la
    Terre, ils réinventent chaque jour la planète. Ils construisent,
    détruisent, donnent lieu à un fantastique spectacle de son et lumière
    qui tantôt nous attire, tantôt nous repousse. Pour l'homme, l'enjeu
    est de taille : comprendre ce chantier colossal et mieux anticiper les
    éruptions..."
  ▪ predicate:
    {
      ▪ id: "5730ab8372ff3400250e6cb2",
      ▪ label: "contenu texte du document",
      ▪ multiline: true
    },
  ▪ type: "literal"
}
],

• publications: [ ]
}

```

API : [/medias/5771605372ff340025fe8948/annotations?media_predicate_id=5730ab8372ff3400250e6c85&auth_token=__](http://medias/5771605372ff340025fe8948/annotations?media_predicate_id=5730ab8372ff3400250e6c85&auth_token=__) -

Response :

```

[
  • {
    o id: "57ba9fa2ab92210026d1b768",
    o media_id: "5771605372ff340025fe8948",
    o subject_id: "575a602c72ff340025773807",
    o media_predicate:
      {

```

```

    ▪ id: "5730ab8372ff3400250e6c85",
    ▪ label: "émission"
  },
  ○ subject_predicate:
  {
    ▪ id: "5730ab8372ff3400250e6c84",
    ▪ label: "a pour programme"
  },
  ○ subject:
  {
    ▪ id: "575a602c72ff340025773807",
    ▪ types:
    [
      ▪ "broadcast_show",
      ▪ "subject"
    ],
    ▪ label: "Le Monde de Jamy",
    ▪ description: null,
    ▪ thumbnail: null,
    ▪ source: "local",
    ▪ lod_uri: null,
    ▪ scope: "public",
    ▪ root_id: null,
    ▪ owner_id: null,
    ▪ metadata: { },
    ▪ info_sources:
    [
      ▪ "User"
    ],
    ▪ accessible: true,
    ▪ enrichment_status: "finished"
  }
}
]

```

WS TSP (<http://157.159.160.210:8080/artemisSearch/ArtemisMoteur>)

```
// Autocomplete
```

```
POST : type=autocomplete&phrase=jam
```

```
Response :
```

```
1. [
  1. 0:"jamais"
  2. 1:"jambes lourdes"
  3. 2:"jambes"
  4. 3:"jambe"
  5. 4:"james"
  6. 5:"jambon"
  7. 6:"jamenais"
  8. 7:"jamy"
  9. 8:"jame1"
```

```
// Recherche de type wordCloud
```

```
POST : type=wordCloud&phrase=volcan
```

```
Response :
```

```
1. {
  1. Groups:[
    1. 0:{
      1. GroupID_0:[
        1. 0:{
          1. sc:50
          2. sug:"Teide"
        2. 1:{sc: 40, sug: "éruptions"}
        3. 2:{sc: 30, sug: "histoire"}
        4. 3:{sc: 30, sug: "Pedro Hernandez"}
        5. 4:{sc: 20, sug: "archipel"}
        6. 5:{sc: 20, sug: "sommet"}
        7. 6:{sc: 20, sug: "Canaries"}
        8. 7:{sc: 20, sug: "Pedro"}
        9. 8:{sc: 20, sug: "faille"}
        10. 9:{sc: 15, sug: "terre"}
        11. 10:{sc: 15, sug: "Tenerife"}
        12. 11:{sc: 15, sug: "Cumbre Vieja"}
        13. 12:{sc: 10, sug: "océan"}
        14. 13:{sc: 10, sug: "Atlantique"}
        15. 14:{sc: 10, sug: "coeur"}
        16. 15:{sc: 10, sug: "environnement"}
        17. 16:{sc: 10, sug: "passion"}
        18. 17:{sc: 10, sug: "restes"}
        19. 18:{sc: 10, sug: "cendres"}
        20. 19:{sc: 10, sug: "surface"}
      2. 1:{
        1. GroupID_1:[
          1. 0:{sc: 50, sug: "soufre"}
          2. 1:{sc: 40, sug: "mineurs"}
          3. 2:{sc: 30, sug: "cratère"}
        ]
      }
    }
  ]
}
```

```

4. 3:{sc: 30, sug: "montagne"}
5. 4:{sc: 20, sug: "coeur"}
6. 5:{sc: 20, sug: "milieu"}
7. 6:{sc: 10, sug: "France"}
8. 7:{sc: 10, sug: "Europe"}
9. 8:{sc: 10, sug: "cause"}

3. 2:{,...}

```

// Recherche de type query

POST : type=query&phrase=jamy

Response :

```

1. 0:{
  1. FTV_UID:"EBU100000000"
  2. episodeSynopsis:"Pour ce premier numéro consacré aux volcans, <span style='background-color:#3B9C9C'>Jamy</span> nous convie à un voyage extraordinaire à la découverte d'une planète qui vit, gronde et palpite. Des lieux et des territoires en fusion dont les secrets, les mystères et les légendes nous sont révélés par ceux qui les connaissent le mieux.Ces hommes et ces femmes avec lesquels <span style='background-color:#3B9C9C'>Jamy</span> a rendez-vous in situ sont chercheurs, scientifiques, historiens ou simples témoins d'un phénomène redouté mais qui fascine pourtant depuis la nuit des temps.Toujours attentif pour rendre simple et intelligible les phénomènes les plus complexes, c'est un <span style='background-color:#3B9C9C'>Jamy</span> impliqué que nous découvrons au cours de ce voyage plein de surprises et de rebondissements, un <span style='background-color:#3B9C9C'>Jamy</span> aventurier aussi qui ne craint pas de s'approcher au plus près de la lave, d'un spectaculaire volcan d'Éthiopie ou de se poser sur un cratère instable de l'île de Montserrat.L'Éthiopie, les Antilles, l'Italie du Sud et Hawaï, autant d'étapes sur un parcours exceptionnel qui conjugue passion, émotion et envie d'apprendre et de transmettre. Un véritable tour du monde filmé comme un grand spectacle de la nature."
  3. episodeTitle:"Des volcans et des hommes"
  4. morceaux_soustitre:[

    1. 0:{
      1. capBegin:"00:03:33.280"
      2. capEnd:"00:04:25.400"
      3. capText:"J'ai rendez-vous avec Thomas, un jeune chercheur français en poste à l'université d'Honolulu. Bonjour, Thomas. -Salut, <span style='background-color:#3B9C9C'>Jamy</span>. -C'est la caméra ? -C'est la caméra. Je te montre l'objet. -C'est une caméra normale, hein ? - Oui, ça a l'air d'être une caméra normale, sauf qu'elle a une lentille spéciale. Elle détecte la chaleur. -Vous l'utilisez pour quoi ? -Pour regarder les dégagements de chaleur. On suit la lave sous terre et en surface. -Ca permet de voir des choses que l'oeil ne voit pas. Si on me filme, ça donne quoi ? -On va le faire. -A quoi je ressemble dans l'infrarouge ? -Ton blouson, tes lunettes, tes cheveux apparaissent froids."
      4. score:0.9411331415176392
    }
  ]
  5. programTitle:"Le Monde de <span style='background-color:#3B9C9C'>Jamy</span>"
  6. programType_typeLabel:"Documentaire"
  7. programType_typeLink:"documentaire"
  8. scoreEBU:20.010545253753662
  9. scoreGlobal:13.654074549674988

```

10. synopsis:"Le Monde de Jamy est une collection de documentaires inédits sur le thème du savoir et de la découverte qui montre le spectacle de la terre dans toute sa diversité sous le regard curieux et complice de Jamy Gourmaud que les téléspectateurs de France 3 connaissent bien."

// Recherche pour un groupe de mots

POST : type=queryANDGroup&phrase=volcan&groupID=GroupID_0

Response :

- ```
1. 0:{
 1. FTV_UID:"EBU138989517"
 2. episodeSynopsis:"Jason, plasticien et canarien d'adoption, est en train de créer le premier musée sous-marin d'Europe sur l'île de Lanzarote. Volcanologue à Tenerife, Pedro surveille le Teide, le plus grand des volcans de l'Atlantique. Mingo, un pêcheur natif de l'île de la Graciosa n'a pas à se soucier de la concurrence. Pour Delphine et Jean Camille, l'archipel des Canaries est l'escale incontournable sur la route des Alizés. En famille, ils ont quitté la France à bord de leur catamaran flambant neuf. Ali a quitté l'Afrique il y a neuf ans. En 2015, il a souhaité faire venir son jeune fils Adou resté en Côte d'Ivoire : un voyage clandestin qui défraye l'actualité lorsque l'enfant est découvert par les douaniers dans une valise."
 3. episodeTitle:"On ira tous aux Canaries !"
 4. morceaux_soustitre:[
 1. 0:{
 1. capBegin:"00:26:40:20"
 2. capEnd:"00:28:15:18"
 3. capText:"J'aime beaucoup les animaux, les chameaux. J'ai l'habitude. Il y a beaucoup de gros. Tu vas voir, le 4e chameau, le poids qu'il a transporté. Mais c'est normal pour eux. Ce sont des chameaux très solides. C'est un animal costaud. ... -A l'ombre du volcan, Pedro Hernandez raconte chaque soir à ses enfants des histoires de feu, des histoires de lave. Pedro Hernandez est né et a grandi à Tenerife. Par passion, il est devenu volcanologue. Pedro vit juste au-dessous du plus grand volcan de l'Atlantique, le Teide. -S'il n'y avait pas de volcans aux Canaries, il n'y aurait pas d'îles et je n'existerais pas."
 4. score:1.9498779773712158
 }
 2. 1:{score: 1.3207123279571533, capEnd: "00:30:43:13", capBegin: "00:29:32:00",...}
 3. 2:{score: 0.9797223806381226, capEnd: "00:29:29:07", capBegin: "00:28:16:01",...}
 4. 3:{score: 0.5780099034309387, capEnd: "01:33:39:24", capBegin: "01:32:17:19",...}
 5. 4:{score: 0.5162803530693054, capEnd: "01:34:51:04", capBegin: "01:33:40:08",...}
```

6. 5:{score: 0.41834622621536255, capEnd: "00:20:28:05", capBegin: "00:19:30:04",...}
7. 6:{score: 0.41834622621536255, capEnd: "00:31:52:09", capBegin: "00:30:47:02",...}
8. 7:{score: 0.4174635410308838, capEnd: "01:36:04:12", capBegin: "01:34:53:01",...}
9. 8:{score: 0.3725324869155884, capEnd: "00:03:33:05", capBegin: "00:01:58:21",...}
10. 9:{score: 0.33618468046188354, capEnd: "01:37:22:21", capBegin: "01:36:07:12",...}
11. 10:{score: 0.3033701777458191, capEnd: "01:32:17:13", capBegin: "01:30:49:11",...}
12. 11:{score: 0.27067211270332336, capEnd: "00:01:58:14", capBegin: "00:00:03:00",...}
13. 12:{score: 0.25294029712677, capEnd: "00:41:39:06", capBegin: "00:40:25:13",...}
5. programTitle:"Thalassa"
6. programType\_typeLabel:"Magazine de découvertes"
7. programType\_typeLink:"magazine"
8. scoreEBU:0.28523707762360573
9. scoreGlobal:2.90164428204298
10. synopsis:"Thalassa est un magazine consacré à la mer et son environnement, d'un point de vue écologique, sportif, humain et historique. Un programme placé sous le signe de l'aventure, du grand large, des vagues et des embruns..."
2. 1:{programTitle: "Le Monde de Jamy",...}
3. 2:{programTitle: "JT 13h", episodeTitle: "",...}
4. 3:{programTitle: "JT 13h", episodeTitle: "",...}
5. 4:{programTitle: "JT 13h", episodeTitle: "",...}
6. 5:{programTitle: "Echappées belles", episodeTitle: "Hawaii, l'autre rêve américain",...}
7. 6:{programTitle: "La quotidienne", episodeTitle: "Le grand ménage du printemps, c'est maintenant !",...}

## WS Ircam (<http://bili2.ircam.fr>)

// Récupère la liste des profils sonores

API : [/catalog.xml](#)

Response :

```
<thredds:catalog xmlns:thredds="http://www.unidata.ucar.edu/namespaces/thredds/InvCatalog/v1.0" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:bes="http://xml.opendap.org/ns/bes/1.0#"><thredds:service name="dap" serviceType="OPeNDAP" base="/hyrax" /><thredds:service name="file" serviceType="HTTPServer" base="/hyrax" /><thredds:service name="wms" serviceType="WMS" base="/ncWMS/wms" /><thredds:dataset name="/" ID="/hyrax /"><thredds:catalogRef name="OLPS" xlink:href="/OLPS/catalog.xml" xlink:title="OLPS" xlink:type="simple" ID="/hyrax/OLPS/" /><thredds:catalogRef name="SimpleFreeFieldHRIR" xlink:href="/SimpleFreeFieldHRIR/catalog.xml" xlink:title="SimpleFreeFieldHRIR" xlink:type="simple" ID="/hyrax/SimpleFreeFieldHRIR/" /><thredds:catalogRef name="SimpleFreeFieldSOS" xlink:href="/SimpleFreeFieldSOS/catalog.xml" xlink:title="SimpleFreeFieldSOS" xlink:type="simple" ID="/hyrax/SimpleFreeFieldSOS/" /></thredds:dataset></thredds:catalog>
```