

ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

[polytech.univ-tours.fr](http://polytech.univ-tours.fr)

**Ajout PRD**

**2017-2018**

# **Plateforme d'acquisition et de formatage temps réel multiflux TV Web**

**Tuteur académique**

**Mathieu DELALANDRE**

**Étudiant**

**Romain ROUSSEAU (DI5)**

4 avril 2018



# Liste des intervenants

Nom	Email	Qualité
Romain ROUSSEAU	<a href="mailto:romain.rousseau@etu.univ-tours.fr">romain.rousseau@etu.univ-tours.fr</a>	Étudiant DI5
Mathieu DELALANDRE	<a href="mailto:mathieu.delalandre@univ-tours.fr">mathieu.delalandre@univ-tours.fr</a>	Tuteur académique, Département Informatique



# Avertissement

Ce document a été rédigé par Romain ROUSSEAU susnommé l'auteur.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Mathieu DELALANDRE susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



## Pour citer ce document

Romain ROUSSEAU, *Plateforme d'acquisition et de formatage temps réel multiflux TV Web*, Ajout PRD, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2017-2018.

```
@mastersthesis{
  author={ROUSSEAU, Romain},
  title={Plateforme d'acquisition et de formatage temps réel multiflux TV Web},
  type={Ajout PRD},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2017-2018}
}
```

# Table des matières

Liste des intervenants	a
Avertissement	b
Pour citer ce document	c
Table des matières	i
Table des figures	iv
Introduction	1
<b>I Contexte de la réalisation et cadre générale du projet</b>	<b>2</b>
1 Contexte, enjeux et acteurs .....	2
2 Objectifs .....	2
3 Hypothèses .....	2
4 Base méthodologique .....	2
<b>II État de l'art et Veille technologique</b>	<b>3</b>
<b>1 État de l'art sur la télévision</b>	<b>4</b>
1 Diffusion d'une chaîne de télévision .....	4
2 Réception de la télévision.....	4
2.1 Réception Hertzienne .....	4
2.2 Réception par satellite .....	4
2.3 Réception par câble .....	4
2.4 Service par contournement ou Over the Top (OTT) .....	4

<b>2</b>	<b>La diffusion de flux en continu sur Internet</b>	<b>5</b>
1	Principe générale du <i>streaming</i> .....	5
2	Genèse du <i>streaming</i> vidéo et évolution au fil du temps .....	5
3	Diffusion en flux adaptatif ou Adaptive Streaming .....	5
4	Les protocoles .....	5
4.1	HTTP.....	5
4.2	RTSP et RTSP 2.0 .....	5
4.3	RTMP .....	5
4.4	MPEG-DASH, HLS, HDS et les nouveaux protocoles basés sur HTTP .....	5
5	Les formats de fichiers utilisés.....	5
<b>3</b>	<b>Les chaînes de TV sur Internet en France</b>	<b>6</b>
1	Critères de recherche.....	6
2	Résultats de l'étude .....	6
3	Tendances observées.....	6
4	Un mot sur le copyright .....	6
<b>4</b>	<b>Les bibliothèques et logiciels d'acquisition</b>	<b>7</b>
1	Les frameworks Windows (VfW, DirectShow, Media Foundation).....	7
2	FFMPEG.....	7
3	libVLC et VLCJ .....	7
4	Streamlink .....	7
<b>III</b>	<b>Description générale et analyse</b>	<b>8</b>
<b>5</b>	<b>Description générale</b>	<b>9</b>
1	Environnement du projet .....	10
2	Caractéristiques des utilisateurs .....	10
3	Fonctionnalités du système .....	10
3.1	Sélection des chaînes via le fichier texte .....	10
3.2	Lancement en ligne de commande.....	10
3.3	Affichage des streams.....	10
4	Structure générale du système .....	10
4.1	La classe TVChannel.....	10
4.2	La classe MainLaunch.....	10
4.3	La classe MainController .....	10
4.4	La classe StreamController .....	10
4.5	La classe TextInputStreamController .....	10
4.6	La classe StreamView .....	10
4.7	Lien entre les classes.....	10

<b>6 Analyse des méthodes</b>	<b>11</b>
<b>IV Qualité de code et Tests de charges</b>	<b>12</b>
<b>7 Documentation et contrôle de qualité du code</b>	<b>13</b>
1 Javadoc .....	13
2 Utilisation de SonarQube et SonarLint .....	13
<b>8 Tests de charges</b>	<b>15</b>
<b>Bilan et conclusion</b>	<b>16</b>
<b>Annexes</b>	<b>17</b>
<b>A Interface Humain/Machine</b>	<b>18</b>
<b>B Gestion de projet, diagramme de Gantt</b>	<b>19</b>
<b>C Diagramme de classes</b>	<b>20</b>
<b>D Diagramme de séquences</b>	<b>21</b>
<b>E Guide d'installation et d'utilisation</b>	<b>22</b>
1 Pré-requis.....	22
2 Installations .....	22
2.1 Streamlink.....	22
2.2 VLC .....	22
2.3 Java Runtime Environment .....	23
3 Récupération du projet et fonctionnement.....	23
4 Fichier texte et syntaxe.....	23
5 Problèmes pouvant survenir .....	24
5.1 Pare-feu de Windows.....	24
5.2 Problèmes d'affichage.....	24



# Table des figures

## 7 Documentation et contrôle de qualité du code

1	Lancement de l'instance SonarQube.....	13
2	Résultats de l'analyse Sonar .....	14

## E Guide d'installation et d'utilisation

1	Commande de lancement du programme .....	23
2	Fichier texte exemple <i>testStream</i> .....	24
3	Exemple de problèmes d'affichage .....	24





# Introduction

Alors qu'approche la fin du cursus à l'école Polytech Tours, se dresse le projet le plus important du parcours, le projet Recherche et Développement. Il s'étale sur l'ensemble de la cinquième année et représente une forme de synthèse des connaissances acquises lors des années d'études précédentes. Il permet de développer et d'approfondir son savoir sur un ou plusieurs champs de compétences spécifiques, afin de devenir un spécialiste du domaine choisi dans le cadre du projet. Ce projet est mené seul, sous la supervision du tuteur académique et avec l'aide d'un intervenant extérieur qui fait partie des initiateurs du sujet. Il permet ainsi de développer son autonomie en menant à bien un travail des prémices jusque, dans le meilleur des cas, à la production.

J'ai décidé de me consacrer à un sujet lié à un domaine en plein essor : le streaming. Ce sujet faisait partie de mes choix privilégiés lorsque j'ai vu la liste des sujets proposés. En effet, je regarde moi-même beaucoup de contenu audiovisuel par ce biais. J'ai pris pour habitude de regarder mes programmes télévisuelles préférés via mon ordinateur plutôt que par les usages traditionnelles avec une télévision depuis plusieurs années déjà, et il m'a paru intéressant de me pencher sur l'envers du décor et le fonctionnement de ces applications utilisées au quotidien par des milliers de personnes aujourd'hui.

Le but du projet est de réaliser une plateforme pour l'acquisition de flux de TV Web. Le contexte, les acteurs et les objectifs seront détaillés dans la première partie de ce rapport. En deuxième partie, nous retrouverons l'état de l'art et la veille technologique pour la plateforme, à savoir des informations sur les protocoles utilisés, les différentes librairies d'acquisition, les serveurs distribuant les flux, etc. La description générale du projet, avec son environnement, ses fonctionnalités et ses caractéristiques, sera dans la troisième partie de ce rapport. Le choix de présenter l'état de l'art et la veille technologique avant la description de la plateforme découle de la présentation des librairies et des protocoles utilisés, détaillés dans la deuxième partie. Il s'agit d'un choix qui permet d'accroître sa compréhension lorsque les descriptions fonctionnelles seront présentées. Enfin nous évoquerons les prémices d'analyse et de conception de la plateforme dans la dernière partie.

Des éléments complémentaires concernant notamment la gestion de projet via un diagramme de Gantt sont disponibles en annexes.

## Première partie

# Contexte de la réalisation et cadre générale du projet

1 Contexte, enjeux et acteurs

2 Objectifs

3 Hypothèses

4 Base méthodologique

## Deuxième partie

# État de l'art et Veille technologique

# 1

## État de l'art sur la télévision

### 1 Diffusion d'une chaîne de télévision

### 2 Réception de la télévision

#### 2.1 Réception Hertzienne

#### 2.2 Réception par satellite

#### 2.3 Réception par câble

#### 2.4 Service par contournement ou Over the Top (OTT)

# 2

## La diffusion de flux en continu sur Internet

### 1 Principe générale du *streaming*

### 2 Genèse du *streaming* vidéo et évolution au fil du temps

### 3 Diffusion en flux adaptatif ou Adaptive Streaming

### 4 Les protocoles

#### 4.1 HTTP

#### 4.2 RTSP et RTSP 2.0

#### 4.3 RTMP

#### 4.4 MPEG-DASH, HLS, HDS et les nouveaux protocoles basés sur HTTP

### 5 Les formats de fichiers utilisés

# 3

## Les chaînes de TV sur Internet en France

1 Critères de recherche

2 Résultats de l'étude

3 Tendances observées

4 Un mot sur le copyright

# 4

## Les bibliothèques et logiciels d'acquisition

1 Les frameworks Windows (VfW, DirectShow, Media Foundation)

2 FFMPEG

3 libVLC et VLCJ

4 Streamlink

## Troisième partie

# Description générale et analyse





## 5

## Description générale

## 1 Environnement du projet

## 2 Caractéristiques des utilisateurs

## 3 Fonctionnalités du système

## 3.1 Sélection des chaînes via le fichier texte

## 3.2 Lancement en ligne de commande

## 3.3 Affichage des streams

## 4 Structure générale du système

## 4.1 La classe TVChannel

## 4.2 La classe MainLaunch

## 4.3 La classe MainControler

## 4.4 La classe StreamControler

## 4.5 La classe TextInputStreamControler

## 4.6 La classe StreamView

## 4.7 Lien entre les classes

# 6

## Analyse des méthodes

Quatrième partie

**Qualité de code et Tests de charges**

# 7

## Documentation et contrôle de qualité du code

### 1 Javadoc

### 2 Utilisation de SonarQube et SonarLint

Pour le développement du site, nous nous sommes servis de l'outil de qualité de code *SonarQube*. C'est un logiciel libre permettant de mesurer la qualité du code produit en se basant sur divers règles mises en place au préalable. Il peut détecter les bugs potentiels, les duplications de code, la couverture de code par les tests unitaires, et bien d'autres. Il supporte plus de 25 langages et s'adapte aux outils de build traditionnels (Ant, Maven, Gradle) ainsi qu'à certains IDE comme Eclipse. Il est aussi adaptable et personnalisable selon les besoins avec de nombreux plug-ins à disposition.

L'installation de SonarQube se fait très simplement. Il suffit de télécharger le programme, puis de lancer le script `StartSonar.bat`. Une instance de SonarQube sera lancée, et une fois que le message "*SonarQube is up*" s'inscrit sur la console comme on peut le voir sur la [Figure 1](#), nous pouvons ouvrir la page du programme sur un navigateur. Par défaut, SonarQube utilise le port 9000 sur l'hôte local.

```
jvm 1 | 2018.04.01 04:23:18 INFO app[[o.e.p.PluginsService] no modules loaded
jvm 1 | 2018.04.01 04:23:18 INFO app[[o.e.p.PluginsService] loaded plugin [org.elasticsearch.transport.Ne
tty4Plugin]
jvm 1 | 2018.04.01 04:24:27 INFO app[[o.s.a.SchedulerImpl] Process[es] is up
jvm 1 | 2018.04.01 04:24:27 INFO app[[o.s.a.p.ProcessLauncherImpl] Launch process[[key='web', ipcIndex=2,
logFilenamePrefix=web]] from [C:\Users\Romain\Documents\sonarqube-7.0]: C:\Program Files\Java\jre1.8.0_161\bi
n\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Romain\Documents\sonarqube-7.0\
\temp -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -cp ./lib/common/*;./lib/server/*;C:\Users\Romain\Docu
ments\sonarqube-7.0\lib\jdbc\h2\h2-1.3.176.jar org.sonar.server.app.WebServer C:\Users\Romain\Documents\sonarq
ube-7.0\temp\sq-process8686175319692195026properties
jvm 1 | 2018.04.01 04:25:42 INFO app[[o.s.a.SchedulerImpl] Process[web] is up
jvm 1 | 2018.04.01 04:25:42 INFO app[[o.s.a.p.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3,
logFilenamePrefix=ce]] from [C:\Users\Romain\Documents\sonarqube-7.0]: C:\Program Files\Java\jre1.8.0_161\bin\
java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Romain\Documents\sonarqube-7.0\
temp -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -cp ./lib/common/*;./lib/server/*;./lib/ce/*;C:\Users\Ro
main\Documents\sonarqube-7.0\lib\jdbc\h2\h2-1.3.176.jar org.sonar.ce.app.CeServer C:\Users\Romain\Documents\so
narqube-7.0\temp\sq-process4006912520959163448properties
jvm 1 | 2018.04.01 04:26:16 INFO app[[o.s.a.SchedulerImpl] Process[ce] is up
jvm 1 | 2018.04.01 04:26:16 INFO app[[o.s.a.SchedulerImpl] SonarQube is up
```

Figure 1 – Lancement de l'instance SonarQube

Maintenant que SonarQube est opérationnel, nous allons analyser notre projet. Dans un premier temps, nous regarderons l'application Angular. Dans notre cas, on souhaite que Sonar examine

les fichiers *typescript*. Étant donné que nous n'avons pas utilisé d'outils de build pour cette partie, nous avons installé un scanner sonar pour tout type de projet. Pour effectuer une analyse, il suffit de lancer depuis un terminal le script *sonar-scanner* et de créer un fichier *sonar-project.properties* comportant les détails concernant le projet, l'analyse s'effectuera par la suite. Le script va scanner tous les fichiers du dossier depuis lequel nous exécutons le programme et déterminera les éléments à modifier dans le code. Les résultats seront affichés dans l'instance SonarQube sur le navigateur internet.

Lors de notre première analyse sur l'application, nous avons eu les résultats suivants :

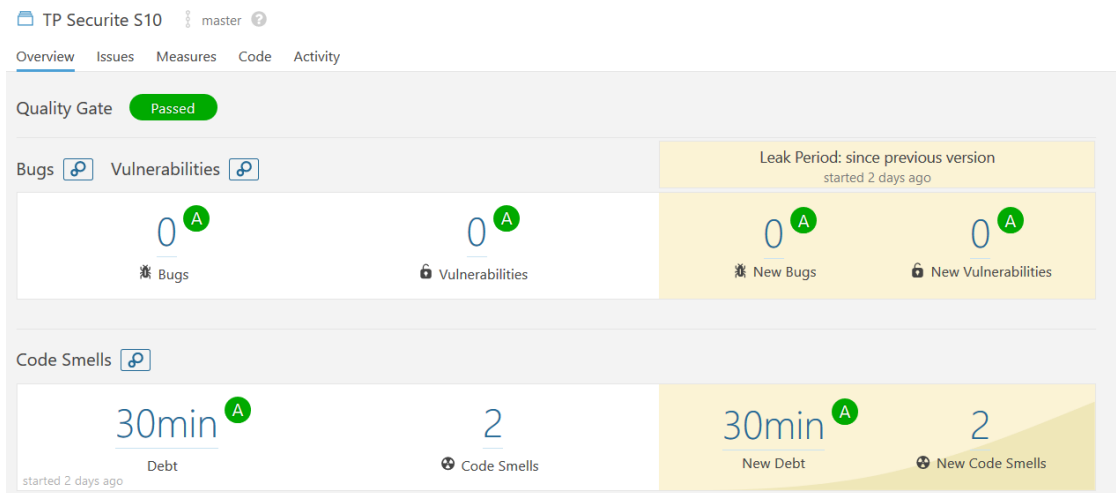


Figure 2 – Résultats de l'analyse Sonar

Nous pouvons remarquer que nous avons produit un code sans bug et sans vulnérabilités tout de suite, ce qui est déjà une très bonne chose. Il ne reste ainsi seulement qu'à corriger les deux erreurs détectées par Sonar pour avoir un code qui respecte les règles de Sonar.

# 8

## Tests de charges



## Bilan et conclusion



## Annexes

A

## Interface Humain/Machine

**B**

## Gestion de projet, diagramme de Gantt



C

## Diagramme de classes

D

## Diagramme de séquences

# E

# Guide d'installation et d'utilisation

## 1 Pré-requis

La plateforme est destinée à fonctionner sur les machines de l'école. Il est nécessaire d'avoir une configuration réseau et matérielle solide si l'on souhaite faire fonctionner un maximum de chaînes en simultané sur l'application. La configuration minimale requise est la suivante :

- Windows 7 Pro 64 bits
- Intel Xen CPU 3.06GHz
- 2 processeurs
- 4 Go de RAM

Pour la configuration réseau, la plateforme peut fonctionner sur un réseau Wi-Fi domicile mais la charge plafonne autour de 7 streams de qualité moyenne.

## 2 Installations

L'application nécessite plusieurs installations :

### 2.1 Streamlink

Streamlink est le logiciel en Python qui va prendre en charge les streams et qui va activer les flux à lire. Il existe plusieurs moyens de se procurer *streamlink*, le plus simple est l'exécutable disponible à l'adresse : <https://github.com/streamlink/streamlink/releases> .

L'application fonctionne pour les versions 0.10.0 et supérieurs.

### 2.2 VLC

VLC est le logiciel permettant à l'application de visualiser les chaînes. Le programme fonctionne avec la version 3.0.1 de VLC disponible à l'adresse suivante : <https://get.videolan.org/vlc/3.0.1/win64/vlc-3.0.1-win64.exe>

## 2.3 Java Runtime Environment

Le programme est un fichier jar qui fonctionne sur la version 8 de Java. Il est nécessaire pour le faire fonctionner de disposer du JRE 8 disponible à l'adresse : [www.oracle.com/technetwork/java/javase/downloads-downloads-2133155.html](http://www.oracle.com/technetwork/java/javase/downloads-downloads-2133155.html)

## 3 Récupération du projet et fonctionnement

Le projet se trouve sur le dépôt GitHub disponible à l'adresse suivante : <https://github.com/RomainR37/stream>

Une fois sur cette page, vous pouvez soit cloner le dépôt Git sur votre ordinateur, soit télécharger le dossier compressé en cliquant sur le bouton *Clone or download* puis en sélectionnant son choix.

Après avoir récupéré le projet, vous pouvez l'exécuter en lançant une invite de commande Windows (en tapant Win+R au clavier puis *cmd* dans le champ disponible). Une fois sur l'invite, dirigez vous vers le dossier du projet récupéré puis dans le dossier, allez dans le dossier *target*. Il contient le fichier jar exécutable et un fichier texte exemple *testStream.txt*. Pour lancer l'application avec le fichier *testStream.txt* à disposition, la commande est la suivante : `java -jar streamPlatform-1.0.0-jar-with-dependencies.jar testStream.txt`. L'application se lancera.

```
C:\Users\Romain\Documents\Projet R&D\Plateforme_Streaming\plateforme\target>java -jar streamPlatform-1.0.0-jar-with-dependencies.jar testStream.txt
```

Figure 1 – Commande de lancement du programme

## 4 Fichier texte et syntaxe

Il est possible de lancer l'application avec le fichier texte que l'on souhaite. Celui-ci pour fonctionner devra se situer dans le dossier *target* également. Cependant le fichier doit respecter une convention de nommage.

Une chaîne nécessite deux informations pour pouvoir fonctionner : son nom et l'adresse du site diffusant son stream. Dans le fichier texte que l'on veut lire, ces deux informations doivent apparaître séparées d'une tabulation. Vous pouvez voir l'exemple du fichier *testStream.txt* ci-après :

Ici sur la **Figure 2**, nous pouvons voir un # avant une ligne. Il s'agit de la notation pour mettre une ligne en commentaire. Ainsi, le fichier *testStream.txt* contient déjà une quinzaine de chaînes de télévision fonctionnelles, il suffit donc de supprimer les # devant la chaîne que l'on veut visualiser. Néanmoins, il est possible de rajouter de nouvelles chaînes tant qu'elle respecte la convention.

Certaines chaînes ne fonctionnent pas avec streamlink, et ne sont donc pas disponibles sur l'application. Par exemple, la version 0.11.0 de streamlink ne prend pas en charge les chaînes diffusées sur Dailymotion. Le problème sera sans doute réglé dans les prochaines versions ce qui rajoutera de nouvelles chaînes disponibles au visionnage.

```
#TF1 https://www.tf1.fr/tf1/direct
#France 2 https://www.france.tv/france-2/direct.html
#France 3 https://www.france.tv/france-3/direct.html
#France 5 https://www.france.tv/france-5/direct.html
#Arte https://www.arte.tv/fr/direct/
#TMC https://www.tf1.fr/tf1/direct
#TFX https://www.tf1.fr/tf1/direct
#LCP http://playtv.fr/television/lcp-ps/
#France 4 https://www.france.tv/france-o/direct.html
#BFMTV http://www.bfmtv.com/mediaplayer/live-video/
#CNews http://playtv.fr/television/cnews/
#Gulli http://replay.gulli.fr/Direct
#France O https://www.france.tv/france-o/direct.html
#TF1-Séries-Films https://www.tf1.fr/tf1-series-films/direct
#L'Equipe http://playtv.fr/television/lequipe/
#LCI https://www.tf1.fr/lci/direct
```

Figure 2 – Fichier texte exemple testStream

## 5 Problèmes pouvant survenir

### 5.1 Pare-feu de Windows

Un message du firewall peut apparaître au lancement de l'application. Cela est dû au script Python lancé par Streamlink. Il suffit que le firewall accepte les scripts pour que le fonctionnement se passe correctement.

### 5.2 Problèmes d'affichage

Sur certaines machines, et en particulier sur les machines virtuelles de l'école, il est possible de constater des problèmes d'affichage des chaînes. On peut voir en effet sur des streams de bonne qualité, des problèmes au niveau des contours sur l'image (comme sur la Figure 3, en particulier à droite de l'image).

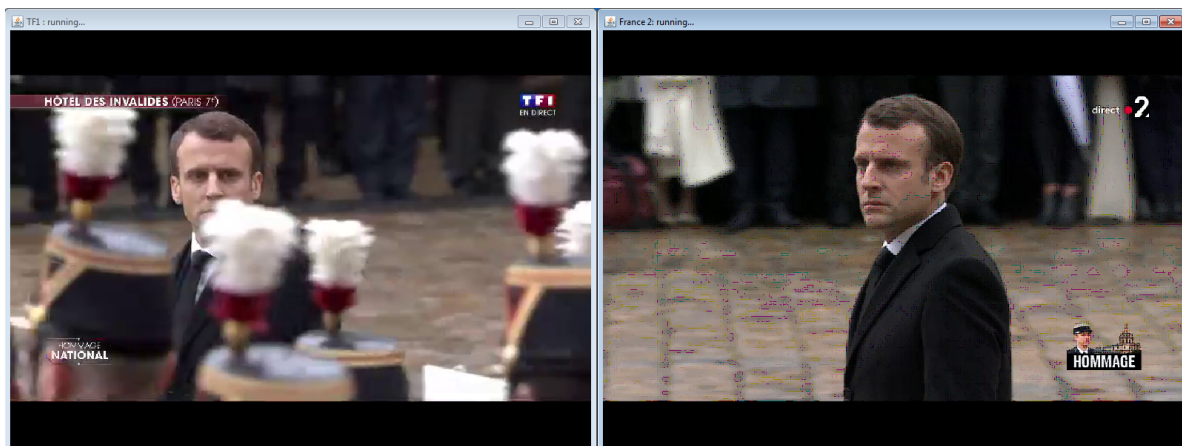


Figure 3 – Exemple de problèmes d'affichage

Ces problèmes peuvent venir des configurations graphiques de la machine, notamment de DirectX. Néanmoins, lors des tests effectués sur machines Windows 10, il n'y avait aucun problème d'affichage, donc cela pourrait être des anciens systèmes Windows également.



# Plateforme d'acquisition et de formatage temps réel multiflux TV Web

## Résumé

Ce projet consiste en l'élaboration d'une plateforme d'acquisition de flux TV Web en temps réel. L'objectif dans un premier temps consistera en l'affichage de plusieurs flux en simultané avant d'éventuellement faire évoluer l'application pour y faire figurer des éléments d'analyse d'images. Au travers de ce rapport, nous verrons la démarche, les objectifs liés à la mise en place d'une telle plateforme, l'architecture logicielle avec une structure en multi-threads, ainsi qu'un état de l'art sur ce domaine constante expansion et qui représente près de 80% du trafic de données sur Internet.

## Mots-clés

Plateforme d'acquisition, Flux de données, Temps réel, Protocole de communication, Télévision en ligne

## Abstract

This project consists in the making of acquisition platform of Live TV streams in real time. The first aim is to display multiple streams at the same time before evolving the platform to add elements of frame analysis. Through this document, we will see the approach, the goals link in the making of such a platform, the software architecture with a multi-threaded structure, as a state-of-the-art in this constantly evolving domain which represents almost 80% of data traffic on the Internet.

## Keywords

Streaming Platform, Live Stream, Real Time, Communication Protocol, Online TV