

# **Système de Vidéo Surveillance**

« Réalisation d'un système de vidéo surveillance avec un Script Shell Windows et l'outil VLC. »

Par Vasco Valadares Semana – En 2023 – Projet en Entreprise

# Table des Matières

1. Introduction .....	3
2. Objectifs du Projet .....	3
3. Contexte et Enjeux Techniques.....	3
4. Architecture et Structure du Système.....	4
4.1. Fichiers Essentiels et Leurs Rôles.....	4
4.2. Technologies et Outils Utilisés .....	4
5. Gestion et Structure des Données de Configuration .....	5
5.1. Configuration des Caméras (Flux RTSP).....	5
5.2. Configuration des Mosaïques Vidéo.....	5
5.3. Flux de Données Vidéo.....	6
6. Fonctionnalités Principales et Mon Rôle dans leur Implémentation .....	6
6.1. Agrégation Multi-Flux RTSP et Configuration des Canaux.....	6
6.2. Ma Conception des Mosaïques Vidéo avec VLM.....	6
6.3. Mon Développement des Scripts de Lancement Automatisé (.CMD) .....	7
6.4. Adaptation et Configuration Multi-Sites (BelleVille, Yuna).....	7
6.5. Mon Travail sur l'Optimisation des Ressources Système.....	7
7. Interface Utilisateur (Affichage Mosaïque).....	7
8. Perspectives d'Évolution et Améliorations Futures.....	8
9. Conclusion.....	8

## 1. Introduction

Dans le cadre de mes activités chez SabSystem, j'ai été amené à travailler sur la mise en place et la maintenance d'un système de surveillance vidéo multi-caméras. Ce système s'appuie sur VLC Media Player pour agréger et afficher en temps réel des flux RTSP (Real Time Streaming Protocol) provenant de caméras IP installées dans divers établissements hôteliers. Mon intervention a notamment porté sur la création de scripts et de configurations permettant d'afficher simultanément jusqu'à 20 flux vidéo sur un seul écran, sous forme de mosaïques personnalisées.

## 2. Objectifs du Projet

Mon travail sur ce projet visait à atteindre plusieurs objectifs clés pour nos clients hôteliers :

- **Centralisation de la surveillance** : Permettre au personnel des hôtels d'afficher simultanément tous les flux vidéo pertinents sur un seul écran, facilitant ainsi la supervision ;
- **Optimisation des ressources système** : Concevoir des configurations VLC qui minimisent la charge sur les machines clientes, souvent des postes de réception ou de sécurité aux performances modestes. J'ai particulièrement veillé à cet aspect ;
- **Facilitation de l'administration à distance** : Mettre en place un accès via AnyDesk pour permettre à notre équipe (et à moi-même) d'intervenir rapidement pour la maintenance et le support ;
- **Déploiement simplifié et standardisé** : Développer une méthode d'installation et de configuration qui puisse être reproduite facilement dans différents établissements avec des besoins variés ;
- **Solution temporaire** : Cette solution devait agir en tant que solution temporaire, en attente d'une solution plus robuste.

## 3. Contexte et Enjeux Techniques

Ce système a été développé pour répondre aux besoins de surveillance vidéo de deux chaînes hôtelières partenaires, BelleVille et Yuna, chacune ayant des configurations spécifiques :

- **Hôtel BelleVille** : Nécessitait un système capable de gérer 20 caméras IP ;
- **Hôtels Yuna** : Demandait un système plus modulaire avec plusieurs configurations possibles selon les sites et le nombre de caméras par site (de 10 à 16).

Les enjeux techniques principaux sur lesquels j'ai dû me concentrer étaient :

- **Compatibilité et Stabilité de VLC** : Assurer un fonctionnement fiable de VLC avec un grand nombre de flux RTSP simultanés ;
- **Scripting sous Windows** : Développer des scripts de lancement .cmd efficaces et faciles à utiliser pour le personnel des hôtels ;
- **Gestion des Configurations VLM** : Maîtriser le langage VLM (VideoLAN Manager) de VLC pour définir précisément les mosaïques et les sources de flux ;
- **Performance Réseau et Vidéo** : Optimiser les paramètres des flux et du transcodage pour une visualisation fluide sans surcharger le réseau local ou la machine cliente.

## 4. Architecture et Structure du Système

### 4.1. Fichiers Essentiels et Leurs Rôles

- **Scripts .cmd (Batch Windows)** : Ce sont les points d'entrée que j'ai créés pour chaque configuration de mosaïque. Un simple double-clic sur ces fichiers lance VLC avec les paramètres et le fichier de configuration VLM appropriés.
  - Exemple de contenu d'un fichier .cmd que j'ai écrit :



```
@echo off
REM Script pour lancer la mosaïque Belleville
REM Auteur: Vasco @ SabSystem
start "" "C:\Program Files\VideoLAN\VLC\vlc.exe" --vlm-conf belleville_mosaic.vlm -I dummy --no-qt-privacy-ask --no-metadata-network-access
```

J'ai ajouté les options `--no-qt-privacy-ask` et `--no-metadata-network-access` pour éviter les pop-ups au démarrage de VLC et minimiser le trafic réseau inutile. L'option `-I dummy` est cruciale pour lancer VLC sans son interface graphique standard, affichant directement la mosaïque et économisant des ressources.

- **Fichiers .vlm (VideoLAN Manager)** : Ces fichiers contiennent la configuration détaillée que j'ai écrite pour VLC. Ils définissent chaque flux de caméra comme un canal, spécifient les URLs RTSP (avec authentification), et décrivent comment ces canaux doivent être assemblés dans la mosaïque (nombre de colonnes, de lignes, résolution, etc.) ;
- **Images de Fond (bg.jpg)** : J'ai inclus la possibilité d'utiliser une image de fond pour les mosaïques afin de personnaliser l'affichage et combler les espaces vides si le ratio des caméras ne correspond pas parfaitement à la grille. L'image est entièrement noire.

### 4.2. Technologies et Outils Utilisés

Les principaux outils et technologies sur lesquels je me suis appuyé sont :

- **VLC Media Player** : Le cœur du système, utilisé pour sa capacité à gérer de multiples flux et à créer des mosaïques via VLM ;
- **Scripts Batch Windows (.cmd)** : Pour automatiser le lancement de VLC avec les configurations spécifiques ;
- **VLM (VideoLAN Manager)** : Le langage de script interne à VLC que j'ai utilisé pour définir les flux et les mosaïques ;
- **RTSP (Real Time Streaming Protocol)** : Le protocole standard utilisé par les caméras IP pour diffuser leurs flux vidéo ;
- **Codecs Vidéo (H.264/MP4V)** : Les codecs généralement utilisés par les caméras et supportés par VLC ;
- **Système d'Exploitation** : Windows (les postes clients dans les hôtels fonctionnent sous Windows) ;
- **AnyDesk** : L'outil que nous utilisons pour l'accès et la maintenance à distance.


## 5. Gestion et Structure des Données de Configuration

Dans ce projet, les "données" principales sont les configurations des flux et des mosaïques.

### 5.1. Configuration des Caméras (Flux RTSP)

Pour chaque caméra, j'ai défini un canal dans le fichier .vlm. Cela inclut :

- La déclaration du canal (ex: new ch1 broadcast enabled) ;
- L'URL d'entrée du flux RTSP, incluant l'adresse IP de la caméra, le port, le chemin du flux, et les identifiants. J'ai dû m'assurer que ces informations étaient correctes pour chaque caméra de chaque site.



```
// Ma configuration pour un canal caméra dans un fichier .vlm
new ch1 broadcast enabled
setup ch1 input "rtsp://USER:PASSWORD@172.16.0.22:554/sub1"
setup ch1 output #mosaic-bridge{id=ch1,width=374,height=206}
```

La partie output #mosaic-bridge définit comment ce flux sera intégré dans la mosaïque globale, avec un identifiant unique et les dimensions souhaitées pour cette vignette dans la grille.

### 5.2. Configuration des Mosaïques Vidéo

La configuration de la mosaïque elle-même est également définie dans le fichier .vlm, au sein d'un bloc setup bg output #transcode{...sfilter=mosaic{...}}. J'y ai spécifié :

- Les paramètres de transcodage (vcodec, fps, vb, acodec=none pour désactiver l'audio et économiser des ressources) ;
- Les dimensions de la mosaïque finale (width, height) ;
- Le nombre de colonnes et de lignes (cols, rows) ;
- L'ordre d'affichage des canaux (order="ch1,ch2,...") ;
- Des options pour la gestion du ratio d'aspect et le maintien de l'image en cas de perte de flux (keep-aspect-ratio=enabled, keep-picture=1).



```
// Ma configuration typique pour une mosaïque 20 caméras (5x4)
setup bg output #transcode{
  vcodec=mp4v, fps=25, vb=0, channels=20, acodec=none,
  sfilter=mosaic{
    alpha=255, width=1920, height=1080, cols=5, rows=4, position=1,

    order="ch1,ch2,ch3,ch4,ch5,ch6,ch7,ch8,ch9,ch10,ch11,ch12,ch13,ch14,ch15,ch16,ch17,ch18,ch19,ch20"
  },
  keep-aspect-ratio=enabled, mosaic-align=5, keep-picture=1
}:bridge-in{offset=100}:display
```

### 5.3. Flux de Données Vidéo

Le flux est assez direct :

1. Les **Caméras IP** génèrent les flux RTSP sur le réseau local de l'hôtel (principalement des plages IP en 172.16.0.x) ;
2. **VLC** (sur le PC de supervision) se connecte à chaque URL RTSP configurée, en utilisant les identifiants fournis ;
3. Le module **Mosaic Bridge** de VLC agrège ces flux ;
4. Le module **Transcodeur** (avec le filtre sfilter=mosaic) formate la sortie finale en une seule image mosaïque ;
5. Cette mosaïque est ensuite **affichée** sur l'écran de supervision. Il n'y a pas de sauvegarde des flux vidéo dans cette configuration ; c'est un affichage en temps réel.

## 6. Fonctionnalités Principales et Mon Rôle dans leur Implémentation

### 6.1. Agrégation Multi-Flux RTSP et Configuration des Canaux

- **Mon Rôle** : J'ai été responsable de la collecte des adresses RTSP de toutes les caméras pour chaque site hôtelier. J'ai ensuite écrit les sections correspondantes dans les fichiers .vlm pour définir chaque caméra comme un canal VLC distinct, en m'assurant que les URLs, les identifiants et les paramètres de sortie vers le mosaic-bridge étaient corrects ;
- **Détails d'Implémentation** : Pour chaque canal, j'ai utilisé new chX broadcast enabled et setup chX input "rtsp://...". La partie output #mosaic-bridge{id=chX,width=W,height=H} a nécessité des calculs pour déterminer les dimensions appropriées de chaque vignette en fonction de la résolution finale et du nombre de caméras.

### 6.2. Ma Conception des Mosaïques Vidéo avec VLM

- **Mon Rôle** : La création de la mosaïque elle-même était une tâche centrale. J'ai défini la grille (nombre de colonnes et de lignes) pour chaque configuration (ex: 5x4 pour 20 caméras, 4x3 pour 10 ou 12 caméras). J'ai également configuré l'ordre d'affichage des

caméras dans la mosaïque et les options d'optimisation, plusieurs tests ainsi que plusieurs manipulations du système d'exploitation hôte ont été réalisées pour optimiser le programme et assurer une stabilité raisonnable ;

- **Détails d'Implémentation** : J'ai utilisé le filtre `sfilter=mosaic` dans la configuration `bg` output des fichiers `.vlm`. J'ai ajusté les paramètres `width`, `height`, `cols`, `rows`, `order`, et `keep-aspect-ratio` pour obtenir le meilleur affichage possible tout en respectant les performances.

### 6.3. Mon Développement des Scripts de Lancement Automatisé (.CMD)

- **Mon Rôle** : Pour simplifier le lancement pour le personnel des hôtels, j'ai créé des scripts Batch Windows (`.cmd`) pour chaque configuration de mosaïque. Ces scripts lancent VLC avec les bons paramètres et le fichier `.vlm` correspondant ;
- **Détails d'Implémentation** : J'ai utilisé la commande `start "" vlc.exe --vlm-conf <fichier.vlm> -I dummy [autres options]`. L'option `-I dummy` était cruciale pour un fonctionnement en "kiosque", sans l'interface standard de VLC. J'ai également inclus des commentaires dans les scripts pour en faciliter la compréhension.

### 6.4. Adaptation et Configuration Multi-Sites (BelleVille, Yuna)

- **Mon Rôle** : J'ai adapté le système pour les besoins spécifiques de chaque chaîne hôtelière et de chaque site. Cela impliquait de créer des ensembles de fichiers `.cmd` et `.vlm` distincts pour BelleVille (20 caméras) et pour les différentes configurations des hôtels Yuna (Porte Maillot avec 10 ou 12 caméras, Saint-Germain avec 12 ou 16 caméras) ;
- **Détails d'Implémentation** : Pour chaque configuration, j'ai dû lister les bonnes adresses IP des caméras, définir la disposition de la grille appropriée, et ajuster les dimensions des vignettes dans les fichiers `.vlm`. J'ai également géré les images de fond (`bg.jpg`) spécifiques ou communes.

### 6.5. Mon Travail sur l'Optimisation des Ressources Système

- **Mon Rôle** : La performance était un enjeu majeur, surtout sur des machines moins puissantes. J'ai donc travaillé sur l'optimisation des paramètres VLC pour minimiser la charge CPU et réseau ;
- **Détails d'Implémentation** : Dans les fichiers `.vlm`, j'ai spécifié des paramètres de transcodage légers : `vcodec=mp4v`, un fps réduit à 25, et surtout `acodec=none` pour désactiver le traitement audio des flux, qui n'était pas nécessaire pour la surveillance visuelle et consommait inutilement des ressources. J'ai aussi recommandé des optimisations système comme la réduction de la fréquence de rafraîchissement de l'écran du poste de supervision, et forcer l'utilisation du GPU pour l'exécution du programme.

## 7. Interface Utilisateur (Affichage Mosaïque)

L'interface utilisateur de ce système est directement la fenêtre VLC affichant la mosaïque des flux vidéo.

- **Vue Mosaïque Principale** : C'est l'affichage en grille des différents flux, que j'ai configuré pour être en plein écran ;
- **Pas d'Interface de Contrôle Directe** : Grâce à l'option -I dummy, l'interface standard de VLC est cachée. Le système est conçu pour une supervision passive. La fermeture se fait via Ctrl+Q ou le gestionnaire des tâches, comme je l'ai documenté dans le README.txt ;
- **Composants Visuels** : Les grilles vidéo que j'ai définies s'adaptent au nombre de caméras. J'ai utilisé une image de fond (bg.jpg) pour l'esthétique et pour combler les espaces. L'option keep-picture=1 que j'ai activée permet de maintenir la dernière image affichée en cas de perte temporaire d'un flux, évitant les écrans noirs.

## 8. Perspectives d'Évolution et Améliorations Futures

Bien que le système actuel soit fonctionnel, je vois plusieurs pistes d'amélioration que nous pourrions explorer à l'avenir :

- **Interface de Configuration Graphique Simplifiée** : Plutôt que de modifier directement les fichiers .vlm ou .cmd, je pourrais développer une petite interface graphique qui permettrait de gérer plus facilement les adresses des caméras et la disposition des mosaïques ;
- **Fonctionnalités Avancées pour VLC** : Intégrer des options pour l'enregistrement programmé ou l'envoi d'alertes en cas de défaillance d'un flux ;
- **Sécurisation Renforcée** : Gérer une authentification plus dynamique et sécurisée aux flux RTSP ;
- **Monitoring et Reporting** : Mettre en place un système simple pour vérifier l'état des flux et notifier un administrateur en cas de problème prolongé avec une caméra ;
- **Alternative à VLM pour plus de Flexibilité** : Pour des besoins plus complexes, explorer l'utilisation de libVLC avec un langage de programmation comme Python ou C# pourrait offrir plus de contrôle et de possibilités d'intégration.

## 9. Conclusion

En conclusion, le système de mosaïques de surveillance vidéo que j'ai contribué à mettre en place pour les hôtels partenaires de SabSystem est une solution innovante et efficace, qui tire parti de la puissance de VLC et de la simplicité des scripts Shell Windows. Mon travail sur la configuration des flux, la conception des mosaïques VLM, et le développement des scripts de lancement a permis d'offrir une solution automatisée, modulable et facile à déployer pour nos clients. Cette réalisation a non seulement renforcé mes compétences techniques en scripting et en intégration d'outils, mais aussi ma capacité à m'adapter à des environnements et des contraintes spécifiques pour fournir des solutions viables et performantes.