INSA Lyon – Département Télécommunications
Année universitaire : 2025 – 2026

Appareil de diffusion vidéo non intrusif
Etat de l'Art

Encadrant :
Stéphane Frenot - Damien Reimert

Réalisé par :

- Orhon Gabriel
- Chkoundali Yasmine
- Mohammi Islam
- Abidi Jean
- Adjami Axel

Sommaire

Abstract

Ce document présente l'état de l'art des solutions existantes de diffusion d'écran, selon une démarche comparative fondée sur cinq critères : non intrusion logicielle, non intrusion matérielle, sécurité, universalité et faisabilité technique.

Les premières éliminations portent sur le critère de non-intrusion logicielle : les protocoles de bureau distant RDP, VNC et DisplayLink sont exclus car ils nécessitent l'installation d'un logiciel et demandent parfois des droits administrateur, ce qui contrevient aux contraintes.

Sur le critère matériel, les solutions spécialisées (ClickShare, InstaShow, boîtiers HDMI sans fil) sont rejetées car elles imposent l'ajout d'un périphérique externe sur la machine de l'utilisateur, introduisant une dépendance matérielle.

Le critère de sécurité et celui d'universalité conduisent à écarter AirPlay qui est rejeté pour son manque d'universalité, Chromecast reste limité par sa dépendance à l'écosystème Google, Miracast reste utilisable mais repose sur un chiffrement WPA2-PSK et présente une implémentation complexe sur Raspberry Pi.

L'évaluation de faisabilité nous pousse à choisir WebRTC plutôt que Miracast qui repose sur un protocole fermé basé sur Wi-Fi Direct/RTSP, difficile à déployer sur un dispositif embarqué sans licences spécifiques. À l'inverse, WebRTC répond ainsi à l'ensemble des critères : absence d'installation logicielle et de périphérique additionnel , chiffrement de bout en bout du flux via DTLS, compatibilité avec tous les navigateurs modernes, et faisabilité élevée sur Raspberry Pi grâce aux implémentations open source du protocole WebRTC, supportées sur les systèmes Linux embarqués.

Le choix de WebRTC résulte ainsi d'une démarche raisonnée d'élimination successives.

Une description du protocole WebRTC est fournie dans ce document, ainsi qu'une description équivalente des protocoles éliminés en annexe.

1. Introduction

1.1. Contexte et Problématique

Dans les environnements professionnels et académiques, la connexion d'un ordinateur à un dispositif d'affichage (vidéoprojecteur, écran) est traditionnellement réalisée par câble (HDMI, USB-C). Cependant, ces connexions physiques présentent des risques de sécurité significatifs, incluant l'accès non autorisé, l'injection de données via les ports USB (BadUSB), ou la compromission par l'installation de pilotes non vérifiés.

Le projet a pour objectif de concevoir et de réaliser un appareil de déport d'affichage sécurisé et non-intrusif. Cet État de l'Art vise à analyser les solutions existantes (protocoles, logiciels et matériels) afin de déterminer la technologie la plus appropriée pour répondre aux contraintes strictes de sécurité et d'ergonomie du projet.

1.2. Objectifs de l'Analyse

Notre benchmark des solutions existantes s'articule autour des exigences fondamentales du projet, qui nous servent de critères de sélection et de rejet.

2. Définition des Critères de Sélection

Pour valider une solution, celle-ci doit satisfaire les critères prioritaires suivants :

Critère	Description	Niveau d'Exigence
C1. Non-Intrusion Logicielle	L'ordinateur source ne doit nécessiter aucune installation de driver, de logiciel lourd ou de droits administrateur pour fonctionner.	Obligatoire (Éliminatoire)
C2. Non-Intrusion Matérielle	La solution ne doit pas exiger le branchement d'un périphérique tiers inconnu (dongle USB, bouton, adaptateur spécifique) sur l'ordinateur source.	Prioritaire
C3. Sécurité et Confiance	Le flux vidéo doit être chiffré (isolation des données) et l'utilisateur doit garantir explicitement le partage (principe de moindre privilège).	Obligatoire
C4. Universalité et Compatibilité	La solution doit être compatible avec les principaux systèmes d'exploitation (Windows, macOS, Linux) sans dépendre d'un écosystème propriétaire unique (ex: Apple).	Prioritaire
C5. Faisabilité Technique et Coût	La technologie doit être accessible et réalisable dans le cadre d'un projet étudiant (budget limité, compétences maîtrisables, temps de développement court).	Opérationnel

3. État de l'Art et Comparaison des Solutions

Nous allons évaluer les solutions du marché (professionnelles, grand public et protocolaires) par rapport à nos critères, en commençant par les plus éliminatoires.

3.1. Tableau de Comparaison (Critère C1 : Non-Intrusion Logicielle)

Solution	Type de Solution	Installation Logicielle Requise ?	Note C1
RDP / VNC / TeamViewer	Logiciel "Bureau à distance"	OUI (Installation complète, droits admin)	**Éliminée**
DisplayLink (USB Graphics)	Pilote Vidéo USB	OUI (Installation d'un driver lourd)	**Éliminée**
ClickShare (Bouton)	Propriétaire (via dongle)	OUI (Application légère ou driver inclus dans le dongle)	Faible
BenQ InstaShow	Matérielle (HDMI/USB)	NON (Vu comme un moniteur standard)	Très Fort
AirPlay / Miracast	Protocole natif (OS)	NON (Intégré à l'OS)	Très Fort
WebRTC	API de Navigateur Web	NON (Intégré au navigateur)	Très Fort

Conclusion C1 : Les solutions nécessitant des droits d'administration ou une installation de pilotes (RDP, VNC, DisplayLink) sont immédiatement rejetées, car elles contredisent directement le principe de non-intrusion et d'isolation des systèmes.

3.2. Tableau de Comparaison (Critère C2 : Non-Intrusion Matérielle)

Nous évaluons ici les solutions qui ont réussi le test C1.

Solution	Type de Périphérique Physique Requis	Note C2
ClickShare	Dongle USB à brancher et presser.	**Éliminée**
BenQ InstaShow	Dongle HDMI + USB (pour alimentation) à brancher.	**Éliminée**
Chromecast	NON (Seul le réseau Wi-Fi est requis).	Très Fort
AirPlay / Miracast	NON (Utilise les cartes réseau existantes).	Très Fort
Wireless HDMI (RF)	Émetteur HDMI/USB-C (Très Intrusionnel).	**Éliminée**
WebRTC	NON (Utilise les cartes réseau existantes).	Très Fort

Conclusion C2 : Les solutions matérielles propriétaires (ClickShare, InstaShow, Wireless HDMI) sont écartées. Bien qu'elles puissent être non-intrusives au niveau logiciel, le branchement d'un périphérique tiers sur un port USB (vecteur d'attaque potentiel, cf. BadUSB) est contraire à notre objectif de sécurité physique.

3.3. Tableau de Comparaison (Critères C3 & C4 : Sécurité et Universalité)

Seules les solutions basées sur des protocoles (Chromecast, AirPlay, Miracast, WebRTC) restent en lice.

Solution	Chiffrement (C3 Sécurité)	Universalité (C4 Compatibilité OS)	Note C3 & C4
Chromecast	Chiffrement TLS (HTTPS)	Faible (Dépend du navigateur Chrome/App Google).	Faible
AirPlay	Chiffrement propriétaire Apple.	Très Faible (Écosystème Apple uniquement).	**Éliminée**
Miracast	Chiffrement WPA2-PSK (Wifi Direct).	Faible (Support variable sous Windows/Android).	Moyenne
WebRTC	**DTLS** (Datagram TLS) pour le transport.	Très Forte (Tout navigateur standard).	Très Fort

Conclusion C3 & C4 : AirPlay est rejeté pour son manque d'universalité. Chromecast est limité par sa dépendance à l'écosystème Google. Miracast est techniquement utilisable, mais son implémentation sur un dispositif embarqué est complexe (Critère C5) et son chiffrement est basé sur WPA2.

4. Comparaison des Solutions Restantes et Choix

Après les phases d'élimination, **WebRTC** apparaît comme la solution la plus prometteuse. Nous comparons ici les derniers candidats possibles (Miracast et WebRTC) sur le critère de faisabilité.

Solution	Protocole	Chiffrement / Isolation	Faisabilité (C5)
Miracast	Wi-Fi Direct (RTSP)	WPA2-PSK	**Faible**. Protocole fermé par la Wi-Fi Alliance. Implémentation complexe sur Raspberry Pi sans licences spécifiques.
WebRTC	Open Source (DTLS/ICE/STUN)	DTLS (Très Fort)	**Très Forte**. API ouverte, excellent support sur Linux/systèmes embarqués (comme Raspberry Pi).

4.1. Arbre de Décision (Synthèse)

Le processus de sélection mène sans ambiguïté au choix de WebRTC.

4.2. Choix de la Solution

La solution retenue est l'implémentation du protocole **WebRTC (Web Real-Time Communication)** pour les raisons suivantes :

1. **Conformité Totale aux Critères C1 et C2 :** L'utilisation d'une API de navigateur garantit l'absence d'installation de logiciel ou de connexion physique intrusive.
2. **Sécurité Intrinsèque (C3) :** WebRTC utilise le protocole DTLS (Datagram Transport Layer Security) pour chiffrer la connexion de bout en bout, assurant l'isolation du flux vidéo.
3. **Universalité (C4) :** Le protocole est un standard web supporté par tous les navigateurs modernes, garantissant la compatibilité avec toutes les plateformes.
4. **Faisabilité (C5) :** C'est une technologie Open Source, parfaitement adaptée à une implémentation sur un système embarqué économique tel qu'un Raspberry Pi.

5. Solution Retenue : WebRTC

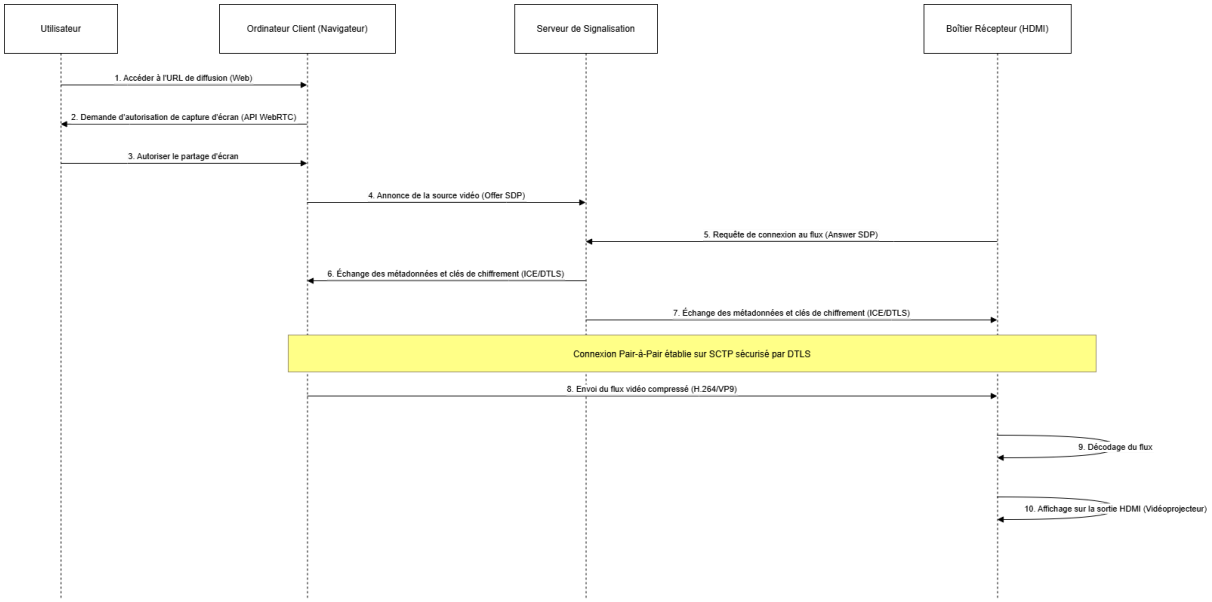
WebRTC

Fabricant/Développeur	Prix	Type de licence	Compatibilité
W3C	0€	BSD	Tout navigateur internet à partir d'une certaine version

Détails du protocole :

WebRTC est une API des navigateurs internet pour proposer ce service. Le protocole se base sur SCTP au niveau de la couche de transport. SCTP reprend les concepts d'UDP tout en garantissant que les données reçues sont les bonnes, comme dans TCP. Le transport des données est sécurisé par DTLS.
Pour compresser les flux de données, WebRTC utilise les codecs vidéos supportés par les navigateurs internet (H.264, VP9, AV1).
L'architecture de WebRTC est Peer-to-Peer : l'échange des données se fait de client à client. Néanmoins, il y a un serveur central qui va gérer les annonces de création de room et la qualité vidéo à utiliser.

Scénario d'utilisation :



Limitations :

Simplement une API des navigateurs internet, pas une application en elle-même. On partage un écran, un onglet ou une fenêtre, on ne peut donc pas profiter du mode présentateur offert par certains logiciels comme PowerPoint ou Google Slides

Sources :

https://webrtc.org/
https://en.wikipedia.org/wiki/WebRTC

Annexe : Solutions Rejetées

Les solutions suivantes ont été étudiées mais ont été écartées du corps principal de l'analyse car elles contreviennent aux critères fondamentaux du projet .

A. Solutions Intrusives Logicielles (Rejet C1)

- **Logiciels "Bureau à distance" (VNC, RDP, TeamViewer) :** Ces solutions sont conçues pour la prise de contrôle à distance. Elles exigent l'installation d'un service d'écoute avec des droits administrateur et ouvrent une faille de sécurité majeure sur le système source. **Rejeté : Contraire à la sécurité.**
- **DisplayLink (USB Graphics) :** Ce protocole transforme le flux vidéo en données compressées sur USB. Il nécessite l'installation de pilotes propriétaires pour fonctionner. **Rejeté : Non-Intrusion Logicielle violée.**

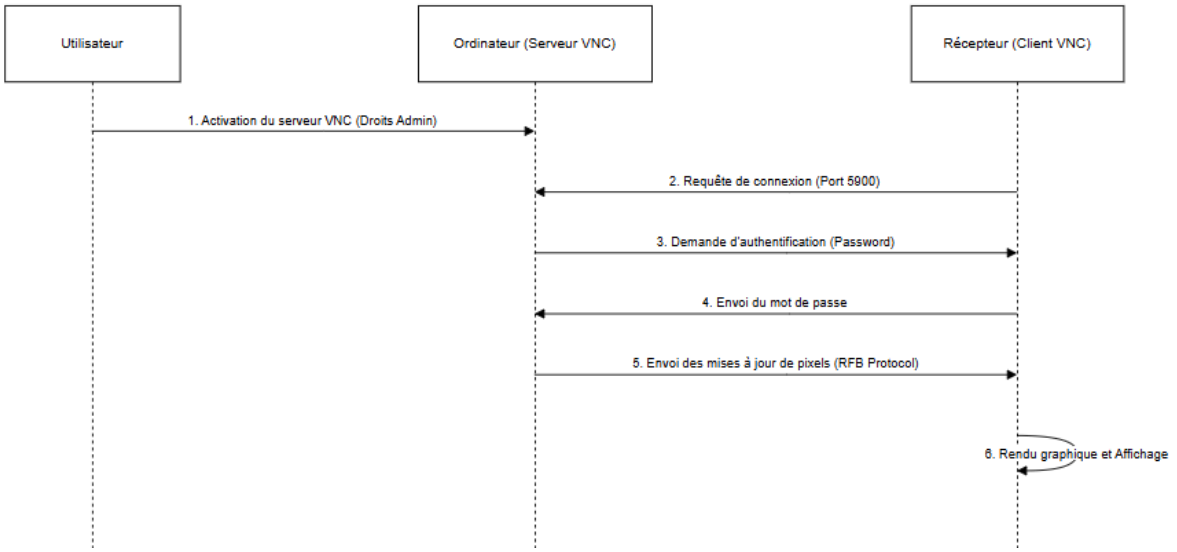
Virtual Network Computing

Fabricant/Développeur	Prix	Type de licence	Compatibilité
Open source (TigerVNC,realVNC…)	0€ (versions de base)	GPL/propriétaire (selon version)	Windows, Linux, Mac

Détails du protocole

VNC utilise le protocole RFB (Remote Frame Buffer). Contrairement à RDP ou Miracast, VNC travaille au niveau du pixel. Le serveur (l'ordinateur source) découpe son écran en rectangles, détecte les changements de pixels (Framebuffering), les compresse (souvent en utilisant des algorithmes comme Hextile ou ZRLE) et les envoie au client via TCP/IP.
C'est un protocole "dumb" : il ne sait pas qu'il diffuse une vidéo ou du texte, il ne voit que des pixels qui changent.

Scénario d'utilisation :



Limitations :

Performance vidéo : Comme le protocole traite des images brutes/compressées sans accélération matérielle spécifique pour le flux vidéo, le framerate est souvent faible et la latence élevée pour la lecture de vidéo.
Pas de son : Le protocole RFB standard ne transporte pas l'audio (bien que certaines variantes propriétaires le fassent).
Intrusif : Nécessite l'installation d'un serveur logiciel sur la machine source.

Sources :

https://fr.wikipedia.org/wiki/Virtual_Network_Computing

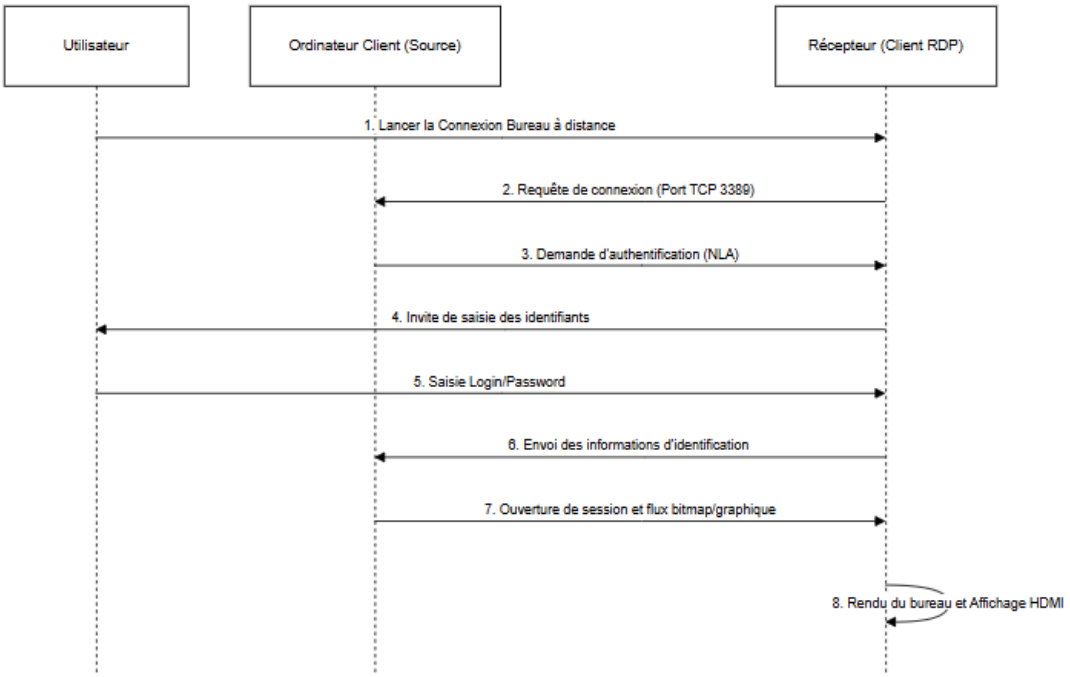
RDP (Remote Desktop Protocol)

Fabricant/Développeur	Prix	Type de licence	Compatibilité
Microsoft	0€ (inclut dans windows)	Propriétaire (Spécification ouverte)	Natif Windows, Clients sur Linux/Mac/Android

Détails du protocole

Contrairement à VNC qui envoie des images, RDP est un protocole sémantique (à l'origine). Il envoie des instructions de dessin ("dessine une fenêtre grise ici", "écrit tel texte là"). Dans les versions modernes (RDP 10 avec RemoteFX), il utilise aussi du streaming vidéo H.264 pour les contenus multimédias. Il gère très bien la redirection de périphériques (USB, Imprimantes, Son) à travers le réseau.

Scénario d'utilisation :



Limitations :

Sécurité : Nécessite de donner ses identifiants Windows à la machine qui projette (risque énorme si le projecteur est compromis).

Intrusif : Nécessite une configuration réseau et l'ouverture de ports (3389) souvent bloqués par les pare-feux d'entreprise.

Sources :

https://fr.wikipedia.org/wiki/Remote_Desktop_Protocol

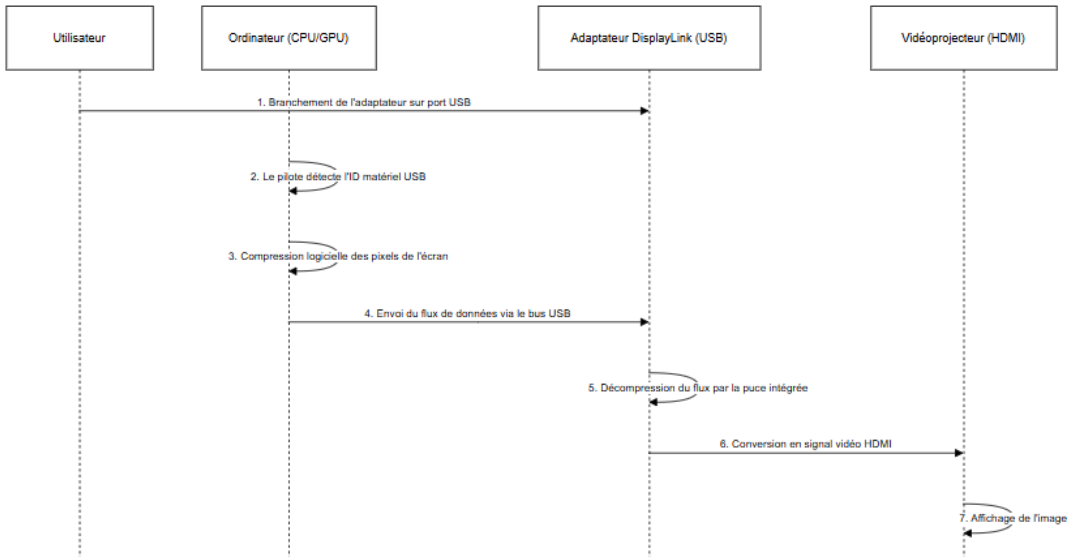
DisplayLink (USB Graphics)

Fabricant/Développeur	Prix	Type de licence	Compatibilité
Synaptics (Display Link)	100€ - 300€ (Docking stations)	Propriétaire + Drivers	Windows, Ubuntu(driver requis) , Mac

Détails du protocole

C'est la technologie standard utilisée par la plupart des stations d'accueil USB universelles. Le CPU de l'ordinateur source compresse l'affichage en temps réel via un driver spécifique (carte graphique virtuelle). Les données compressées sont envoyées via USB (ou Wifi pour d'anciennes versions) vers une puce dédiée dans la station d'accueil qui décode et sort le signal HDMI/DP.

Scénario d'utilisation :



Limitations :
Très intrusif (Drivers) :. DisplayLink ne fonctionne pas sans l'installation d'un driver lourd (souvent impossible sans droits administrateur).
Consommation CPU : La compression est faite par le processeur de l'ordinateur, ce qui peut ralentir les petites machines lors de diffusion vidéo.
Sécurité : Le driver a un accès bas niveau au système d'exploitation.

Sources :

https://en.wikipedia.org/wiki/DisplayLink

B. Solutions Intrusives Matérielles (Rejet C2)

- **Barco ClickShare / BenQ InstaShow :** Ces systèmes reposent sur un "bouton" ou "dongle" à brancher en USB/HDMI. Bien qu'ergonomiques, le branchement d'un périphérique tiers sur USB représente un risque de sécurité physique important (injection de code malveillant) et viole le critère de non-intrusion matérielle. **Rejeté : Risque USB et Intrusion Matérielle.**
- **Wireless HDMI (Transmetteurs RF 60GHz) :** Solution purement matérielle qui imite un câble HDMI sans fil. Bien que non-intrusive au niveau logiciel, elle exige une technologie radio (RF) très coûteuse et complexe à maîtriser et à implémenter dans le cadre d'un projet étudiant (Rejet C5). **Rejeté : Faisabilité Technique.**

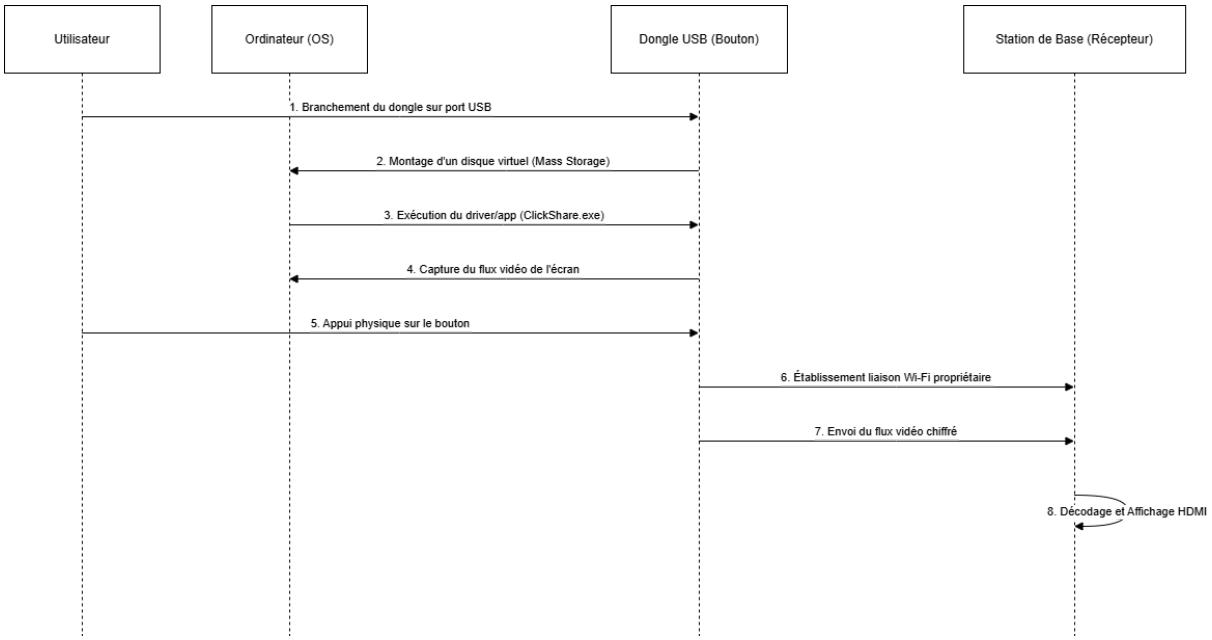
Clickshare

Fabricant/Développeur	Prix	Type de licence	Compatibilité
Barco	~2000€ + 200€ par dongle supplémentaire	Propriétaire	Mac/Windows

Détails du protocole :

Les boutons sont appariés à la station de base qui émet un réseau Wi-Fi privé avec un SSID caché. Une fois le premier appairage fait, les boutons agissent comme une seconde carte Wi-Fi pour le PC : quand ils sont branchés, le dongle se connecte automatiquement au réseau Wi-Fi privé de la station de base. L'application/Driver disponible sur le dongle va ensuite lancer la capture de l'écran et la transmettre sur le Wi-Fi privé avec du chiffrement (AES).

Scénario d'utilisation :



Limitations :

Pas de support pour Linux.
Coût d'installation très élevé, solution propriétaire.
Toujours un Dongle à brancher en USB, ce n'est pas transparent pour l'utilisateur

Sources :

https://www.barco.com/fr/support/knowledge-base/2898-clickshare-button-sharing-usage
https://en.wikipedia.org/wiki/Barco_(manufacturer)

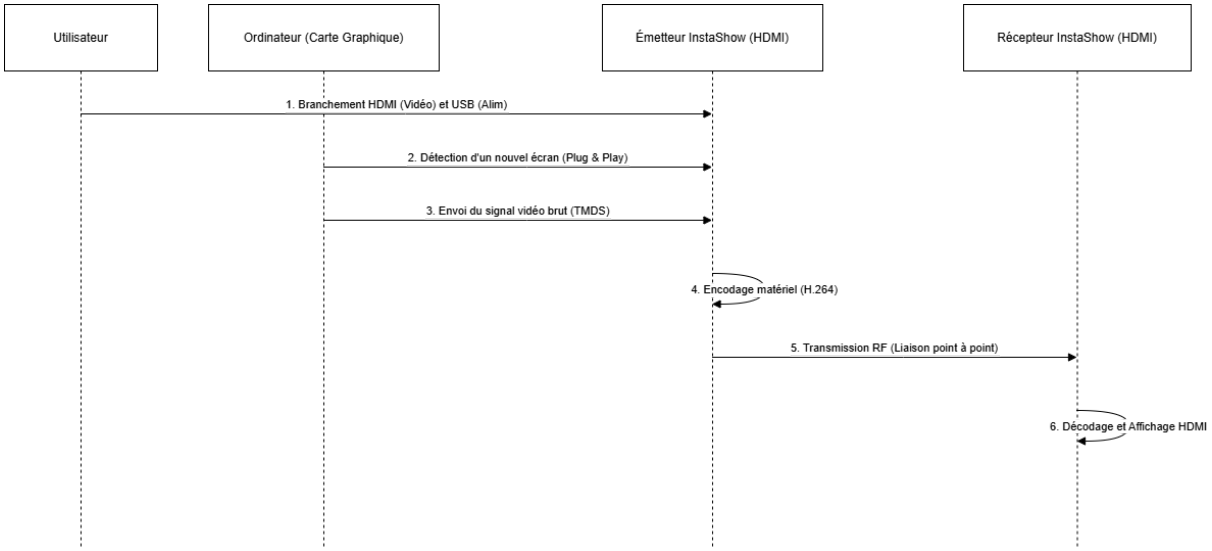
BenQ InstaShow (Solution Hardware)

Fabricant/Développeur	Prix	Type de licence	Compatibilité
BenQ	~1000€ - 1500€	Propriétaire (Matériel)	Universel (Tout port HDMI + USB)

Détails du protocole

Contrairement au ClickShare de Barco, la solution de BenQ est purement matérielle. Le "bouton" (émetteur) qu'on branche sur le PC encode le signal HDMI en H.264 et l'envoie via un Wi-Fi 5Ghz propriétaire et sécurisé (WPA2-Enterprise) vers le récepteur.
Le bouton nécessite un port HDMI (pour l'image) et un port USB (uniquement pour l'alimentation électrique).

Scénario d'utilisation :



Limitations :

Connectique : Nécessite d'avoir un port HDMI disponible sur le PC portable (de plus en plus rare sur les ultrabooks qui n'ont que de l'USB-C), ce qui oblige souvent à avoir des adaptateurs.
Coût : Solution très coûteuse destinée aux entreprises.
Surface d'attaque physique : Comme pour ClickShare, cela nécessite de brancher un périphérique tiers inconnu (USB pour l'alim) sur la machine.

Sources :

https://www.benq.eu/fr-fr/campaign/instashow.html

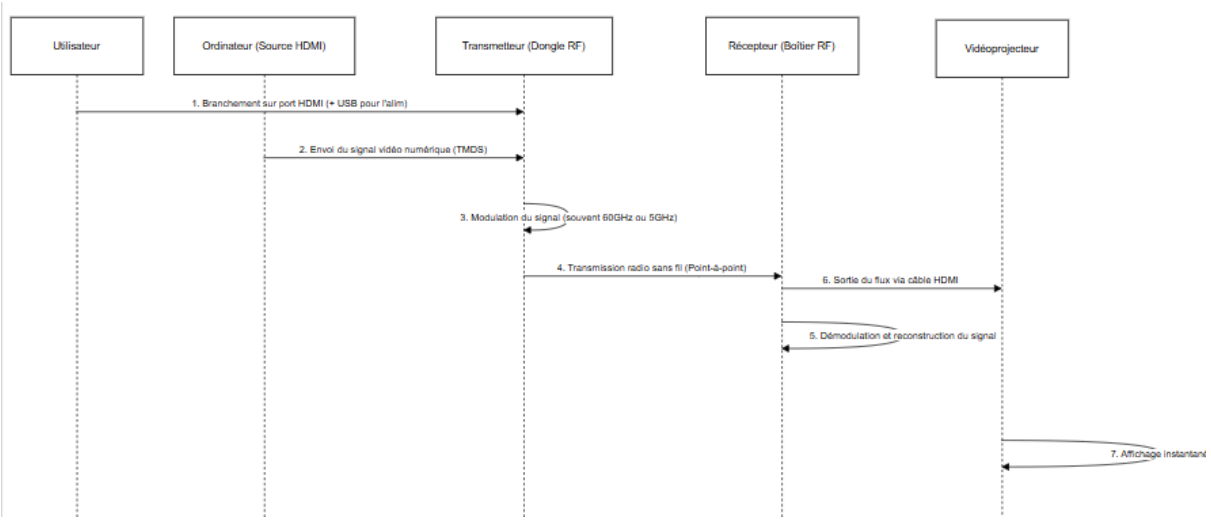
Wireless HDMI (Transmetteurs RF)

Fabricant/Développeur	Prix	Type de licence	Compatibilité
Divers (Ugreen, Nyrius)	100 - 300€	matériel universelle	Tout port HDMI

Détails du protocole :

Pas de Wi-Fi, pas d'adresse IP, pas de logiciel. C'est full transmission radio.. C'est littéralement un "câble HDMI invisible". L'émetteur et le récepteur communiquent via des ondes radio (souvent 60GHz ou 5GHz) pour envoyer l'image pixel par pixel. L'ordinateur ne se rend compte de rien, pour lui, il est juste branché à un écran externe classique.

Scénario d'utilisation :



Limitations :

La portée est courte (10-15 mètres max), pour de la projection de cours en salle de TD ca va mais en amphi c'est un peu short. Ça chauffe et ça prend de la place (il faut souvent une alim externe car le HDMI ne fournit pas assez). C'est du "point-à-point" : impossible de partager l'écran à plusieurs ou de switcher facilement de présentateur sans se passer physiquement le dongle.
Ca oblige a avoir un dongle physique donc on ne résout pas le problème de la surface d'attaque physique

Sources :

https://en.wikipedia.org/wiki/Wireless_Home_Digital_Interface

C. Solutions non-Universelles (Rejet C4)

- **AirPlay (Apple) :** Système performant mais strictement limité à l'écosystème Apple (macOS, iOS). Son caractère propriétaire le rend infaisable pour un système universel basé sur Raspberry Pi. **Rejeté : Non-Universalité.**

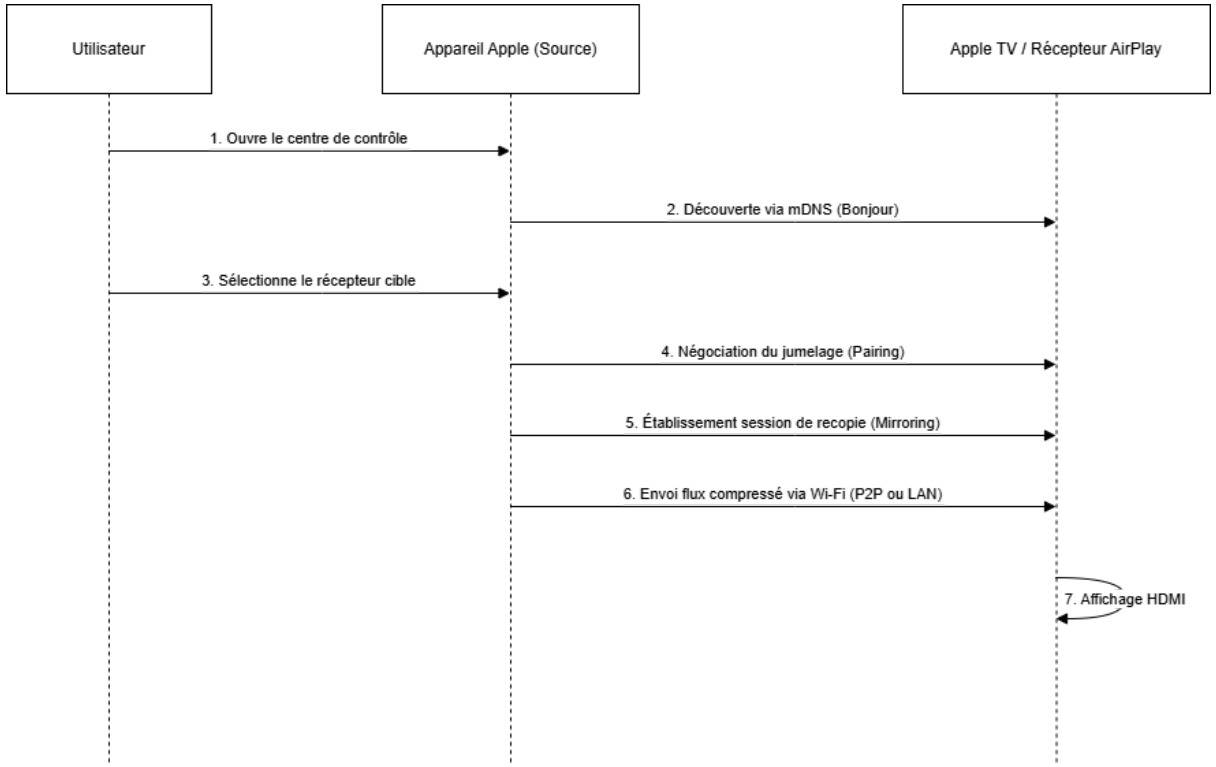
AirPlay

Fabricant/Développeur	Prix	Type de licence	Compatibilité
Apple	170€	Propriétaire	Ecosystème Apple

Détails du protocole :

Le protocol AirPlay peut fonctionner de deux manières : connexion pair à pair entre les deux devices (WiFi Direct) ou en passant par le réseau LAN.
Dans le cas où les deux méthodes sont disponibles, le protocole privilégiera la méthode la plus stable et avec le moins de latence.
Pour le streaming vidéo, la stack protocolaire utilisé par Apple n'a pas encore été dévoilé/rétro-ingénieuré totalement. Il existe cependant une implémentation Open Source du mirroring server (serveur qui affiche).

Scénario d'utilisation :



Limitations :

Ce protocole n'est disponible que dans l'écosystème Apple. A moins de réussir le reverse-engineering de la partie client du protocole, le déploiement de cette solution nécessiterait de passer l'ensemble du parc machine sur Apple.

Sources :

https://en.wikipedia.org/wiki/AirPlay

D. Solutions Complexes à implémenter (Rejet C5)

- **Miracast :** Protocole fermé par la Wi-Fi Alliance. Implémentation complexe sur Raspberry Pi sans licences spécifiques.

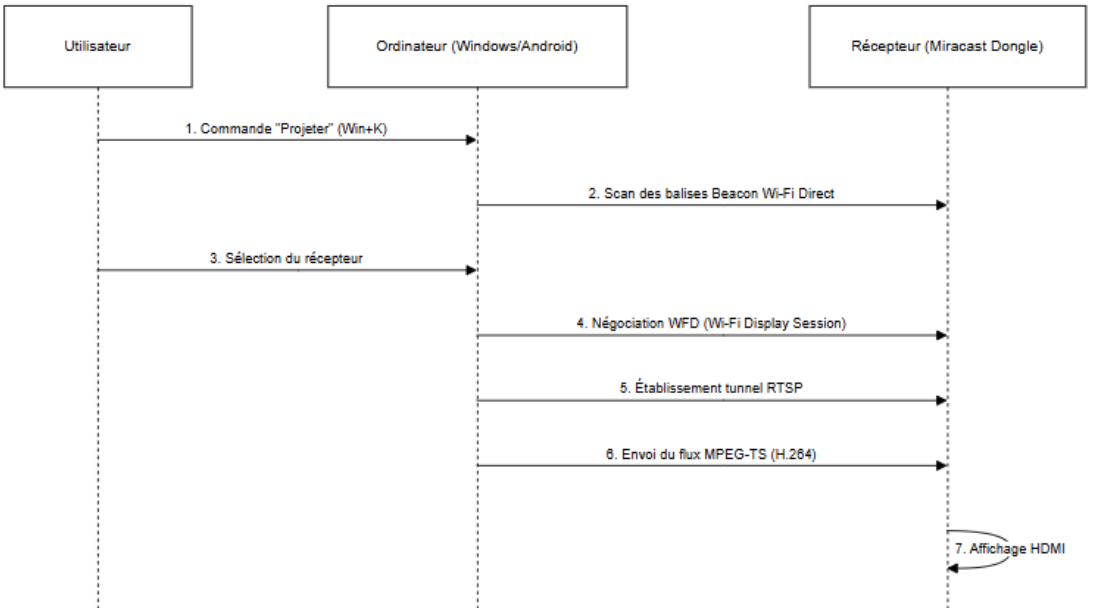
Miracast

Fabricant/Développeur	Prix	Type de licence	Compatibilité
Wi-Fi Allicance	0€ ?	Non concerné (Spécification)	Dépend de l'implémentation

Détails du protocole :

Le fonctionnement est très proche de la solution d'Apple : les deux devices établissent une connexion en Wi-Fi Direct (pair à pair, sans passer par le routeur). Une fois la connexion établie et sécurisée par WPA2, on utilise le protocole RTSP (comme Apple pour l'Audio) pour transmettre les informations. Avec le codec H264, on peut envoyer du 1080p.

Scénario d'utilisation :



Limitations :

Le protocole MiraCast n'est pas implémenté par défaut sur les machines Linux et Apple. Cette solution nécessite donc l'installation d'une application sur le PC de l'utilisateur, ce qui ne correspond pas vraiment au critère de transparence que l'on recherche.

Sources :

https://en.wikipedia.org/wiki/Miracast
https://github.com/albfan/miraclecast/issues/194