

Journal : Visio-Display : retour d'expérience sur une plateforme libre d'affichage dynamique

Posté par woofix le 28 août 2026 à 07:07. Licence CC By-SA.
Étiquettes : logiciel_libre, auto-hébergement, docker, python, affichage_dynamique, vibe_coding



Sommaire

- Architecture
- Gestion des écrans et des contenus
- Déploiement et fonctionnement des clients
- Administration et exploitation
- Installation en une commande
- Pourquoi publier ce retour ici ?

Bonjour,

Je développe depuis quelque temps **Visio-Display**, une plateforme libre d'affichage dynamique distribuée sous licence **GPL-3.0**. Le projet est né d'un besoin simple : administrer plusieurs écrans depuis une interface centralisée, tout en conservant la maîtrise de l'hébergement, des données et des clients d'affichage.

Au fil du développement, le projet a dépassé la simple diffusion d'images et de vidéos. Il couvre maintenant une partie importante du cycle de vie d'une infrastructure d'affichage : création et planification des contenus, déploiement des clients, supervision, maintenance, sauvegardes et mises à jour.

Le dépôt est disponible sur GitHub :

https://github.com/woofix/visio_display

Architecture

Le serveur repose principalement sur :

- Python, Flask et Gunicorn pour l'application web ;
- PostgreSQL pour les données persistantes ;
- Redis et RQ pour le cache et les tâches en arrière-plan ;
- Docker Compose pour le déploiement et l'exploitation des services.

L'ensemble est auto-hébergé. Je l'exploite et le valide sur ma propre infrastructure, avec de véritables clients Linux raccordés au serveur.

Gestion des écrans et des contenus

Depuis l'interface web, il est possible de créer plusieurs écrans nommés et de gérer leur contenu indépendamment.

La bibliothèque de médias accepte notamment les images, les vidéos et les documents PDF. Les contenus peuvent être organisés dans des listes de lecture avec un ordre et une durée d'affichage propres à chaque élément.

La planification permet de définir des périodes de diffusion selon les dates et les horaires. Le projet dispose également de groupes de médias et de campagnes prioritaires temporaires permettant de remplacer momentanément la programmation normale avant de revenir automatiquement à la liste de lecture habituelle.

Un éditeur d'annonces au format 16:9 est intégré. Il permet de préparer rapidement un visuel depuis l'interface puis de l'exporter directement vers la bibliothèque de médias.

Un générateur de QR codes permet aussi de créer des contenus pour différents usages : réseau Wi-Fi, URL, contact, emplacement cartographique, messagerie ou texte libre.

Déploiement et fonctionnement des clients

Visio-Display peut préparer et déployer à distance des clients Linux configurés en mode kiosque, notamment via SSH.

Chaque client conserve localement sa configuration et son jeton de heartbeat. Un service systemd assure ensuite la communication avec le serveur. Le premier heartbeat est envoyé environ quinze secondes après le démarrage, puis toutes les trente secondes.

Si un client perd son alimentation ou sa connexion réseau, le serveur le considère simplement comme hors ligne après cinq minutes sans heartbeat. Aucun nouvel appairage manuel n'est nécessaire : lorsque la machine redémarre ou retrouve le réseau, le service reprend ses communications et le serveur reconnaît le même client.

L'interface permet de consulter l'état des clients, leurs informations système et leurs captures d'écran. Des opérations de maintenance à distance sont également présentes : redémarrage, arrêt, réinstallation et mise à jour.

Administration et exploitation

La plateforme intègre notamment :

- la gestion des utilisateurs, rôles et permissions ;
- des journaux d'activité ;
- des jetons distincts selon les usages ;
- des sauvegardes manuelles ou planifiées ;
- une restauration depuis l'interface web ;
- des copies SMB optionnelles ;
- un mécanisme contrôlé de mise à jour du serveur ;
- une API utilisée pour les communications et certaines opérations d'administration.

L'objectif n'est pas seulement d'afficher un média, mais de disposer d'un ensemble exploitable au quotidien : savoir quels écrans sont disponibles, intervenir à distance et restaurer le service sans devoir se connecter manuellement à chaque machine.

Installation en une commande

Pour simplifier les essais et les nouveaux déploiements, un installateur guidé est disponible. Sur un serveur Linux disposant de Docker, Docker Compose et Git, l'installation peut être lancée avec :

```
bash <(curl -fsSL https://raw.githubusercontent.com/woofix/visio_display/main/server-install.sh)
```

L'installateur est proposé en français et en anglais. Il vérifie les prérequis, clone le dépôt, génère les secrets, prépare l'environnement, crée le compte administrateur, construit les conteneurs et démarre les services.

Une installation manuelle avec Docker Compose reste documentée pour celles et ceux qui préfèrent contrôler chaque étape.

Pourquoi publier ce retour ici ?

Visio-Display est pour moi un projet concret à la croisée du développement web, de Linux, de Docker, du réseau, de l'automatisation et de l'exploitation système.

J'ai utilisé **ChatGPT/Codex comme assistant** pour certaines suggestions de code, revues, tests et tâches de documentation. Je reste le mainteneur du projet : je définis l'architecture, réalise les déploiements et valide les fonctionnalités sur l'instance en fonctionnement. Ce journal a également été préparé avec cette assistance, puis vérifié par rapport au code et à l'application déployée.

Je serais intéressé par les retours de personnes exploitant déjà des écrans d'information ou des bornes interactives : quels matériels et environnements vous sembleraient les plus pertinents à tester ? Les rapports de bogues, essais et contributions sont naturellement bienvenus.

(1 commentaire).

Markdown

EPUB

C'est lassant à la longue...

Posté par Tonton Th (site web personnel, Mastodon) le 28 août 2026 à 08:24.

Évalué à 2 (+0/-0).

Le dépôt est disponible sur GitHub :

Pourquoi tous les *slopwares* soit-disant "libre" qui débarquent ici depuis quelques mois sont-ils tous hébergés chez Microsoft ?

Répondre

Envoyer un commentaire

Suivre le flux des commentaires

Note : les commentaires appartiennent à celles et ceux qui les ont postés. Nous n'en sommes pas responsables.

Revenir en haut de page

Proposer un contenu

Identifiant

Identifiant

Mot de passe

Mot de passe

☐ Connexion automatique

Se connecter

Pas de compte ? S'inscrire...

Derniers commentaires	Étiquettes (tags) populaires	Sites amis	À propos de LinuxFr.org
- C'est lassant à la longue... - Re: pffff - Re: Et à la fin on fait n'im... - Re: Mise au point... - Re: Mise au point... - hommage sympa - Re: Triste - Re: Et à la fin on fait n'im... - Re: moi j'aime bien - Re: Oups - Oups - Re: Et à la fin on fait n'im...	- intelligence_artificielle - grands_modèles_de_lang... - merdification - réchauffement_climatique - linux - canicule - administration_française - capitalisme - startup_nation - états-unis - cybersécurité - fuite_de_données	- Agenda du Libre - April - Éditions D-BookeR - Éditions Diamond - Éditions Eyrolles - Éditions ENI - En Vente Libre - Framasoft - La Quadrature du Net - Lea-Linux - Open Source Initiative - Imprimerie Grafik Plus	- Mentions légales - Faire un don - L'équipe de LinuxFr.org - Informations sur le site - Aide / Foire aux questions - Suivi des suggestions et b... - Wiki du site - Règles de modération - Statistiques - API pour le développement - Code source du site - Plan du site