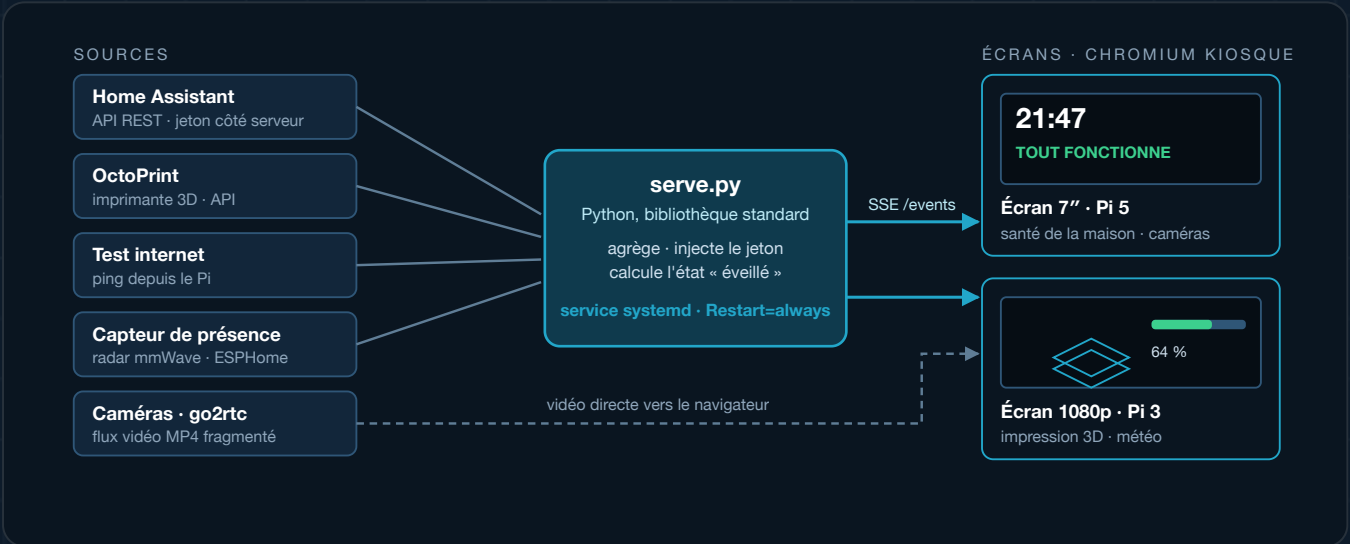


Tableau de bord de supervision sur Raspberry Pi

Deux écrans muraux qui disent en un regard si la maison fonctionne : courant, internet, serveur, caméras, capteurs et imprimante 3D. Un petit service Python agrège les sources et pousse l'état aux écrans ; rien ne s'affiche quand tout va bien.

MATÉRIEL	LOGICIEL	DEPUIS	STATUT
Raspberry Pi 5 · écran 7"	Python · JavaScript	Juillet 2026	En service, 24 h/24



2 écrans
alimentés par un seul service Python

0 jeton
d'accès dans le navigateur : tout passe par le serveur

0 requête
périodique côté page : l'état est poussé en SSE

Noir
quand personne n'est là : écran réveillé par un radar

Raspberry Pi 5

Debian 13

Python

JavaScript

Server-Sent Events

API REST

systemd

Wayland · labwc

Chromium kiosque

Git · déploiement

ESPHome

01 Architecture et choix

Un intermédiaire côté serveur

La page ne parle jamais directement à Home Assistant. Le service local sur le Pi relaie les appels et ajoute le jeton d'accès, stocké dans un fichier lisible par root seulement. Avantages : aucun secret dans le navigateur, pas de problème CORS, et la page s'affiche même si le serveur domotique est en panne (« serveur injoignable »).

L'état poussé, pas interrogé

Le service interroge lui-même les sources et pousse un instantané unique sur /events (Server-Sent Events). Les pages ne font plus aucune requête périodique, ce qui a réglé une fuite de mémoire du navigateur (voir page suivante).

Fail-open

Si le capteur de présence ne répond pas, l'écran reste allumé plutôt que noir : une panne du capteur ne doit jamais cacher une panne de la maison.

```
# serve.py (extrait simplifié)
def do_GET(self):
    if self.path.startswith("/api/"):
        return self.proxy(HA_URL, headers={
            "Authorization": f"Bearer {TOKEN}"})
    if self.path == "/events":
        self.send_header("Content-Type",
            "text/event-stream")
        for snap in SNAPSHOTS.follow():
            self.wfile.write(sse(snap)) #
        "data: {json}\n\n"
```

```
# /etc/systemd/system/dashboard.service
[Service]
EnvironmentFile=/etc/dashboard.env # root, 600
ExecStart=/usr/bin/python3 serve.py
Restart=always
```

Le kiosque est lancé par la session Wayland au démarrage :
chromium --kiosk --app=http://localhost:8080/, avec
une boucle de surveillance qui le relance s'il s'arrête.

02 Une interface qui ne montre que les problèmes

ÉTAT AFFICHÉ	QUAND
Tout fonctionne	aucune anomalie : horloge seule
À vérifier	appareil hors ligne, sonde muette, mise à jour
Sur batterie	coupure de courant, onduleur actif
Serveur injoignable	Home Assistant ne répond plus

- Un sous-système en bonne santé n'occupe aucun pixel ; les problèmes s'affichent à côté de l'horloge (4 lignes maximum).
- Une caméra coupée moins de 15 min reste une simple info bleue ; au-delà, l'état passe à « à vérifier ».
- Une image figée ressemble à une image en direct : chaque tuile vérifie que la vidéo avance réellement et se reconnecte sinon, en espaçant les essais.
- Une caméra morte depuis plus de 3 min libère sa place, l'autre prend toute la largeur.

03 Dépannage : quatre pannes résolues

ÉCRAN NOIR	Symptôme : l'écran reste noir alors que le signal est présent et l'alimentation correcte. Démarche : pilote d'affichage DPI du Pi 5 activé par superpositions KMS, sortie vérifiée connectée en 1024×600, image de test blanche. Cause : l'interrupteur physique du rétroéclairage sur la carte de l'écran était sur arrêt. Leçon : vérifier le matériel le plus simple d'abord.
CAMÉRAS NOIRES	Symptôme : après environ un jour, les deux vidéos deviennent noires alors que l'horloge tourne. Diagnostic : <code>df -h /tmp</code> plein à 98 % alors que du ne trouve rien, soit des fichiers supprimés mais encore ouverts ; près de 1 000 descripteurs « deleted » dans le processus de rendu. Cause : le navigateur ne libérait pas la mémoire partagée de chaque réponse HTTP (~1 Mo par requête), jusqu'à bloquer le décodage vidéo matériel. Correction : passage au push SSE, plus aucune requête périodique.
ÉCRAN BLANC AU DÉMARRAGE	Symptôme : page blanche certains matins, réglée par un simple redémarrage du navigateur. Cause : le navigateur démarrait pendant le saut d'horloge de la synchronisation NTP et son service réseau plantait. Correction : le script de démarrage attend que l'heure soit synchronisée.
LENTEURS	Mesure : 85,6 °C et processeur bridé de 2,4 à 1,5 GHz (<code>vcgencmd get_throttled</code>), sans sous-tension : l'alimentation est saine, il faut un refroidissement actif. Leçon : contrôler température et bridage avant d'accuser le logiciel.

04 Exploitation

Déploiement

Code dans un dépôt Git privé ; le Pi tire les mises à jour avec une clé de déploiement en lecture seule, puis redémarre le service et le kiosque.

Vérification

Capture d'écran à distance de l'affichage réel (grâce à `grim` sous Wayland) après chaque changement, plutôt que de supposer.

Présence

Radar mmWave réglé en portée et sensibilité pour n'allumer les écrans que si quelqu'un est devant, de jour comme de nuit.

Ce que ce projet montre : partir d'un besoin simple (« est-ce que tout marche ? »), garder les secrets côté serveur, concevoir une interface pour des non-informaticiens, puis diagnostiquer méthodiquement des pannes qui touchent à la fois le matériel, le système et le navigateur.