

Description de l'application

Application web progressive (PWA) de supervision et de contrôle d'installations industrielles de traitement de surface, communiquant en temps réel via MQTT.

Aperçu

VKQVWebApp permet de visualiser et piloter des machines industrielles depuis un navigateur web. L'application se connecte à un broker MQTT pour recevoir les données en temps réel et envoyer des commandes aux équipements.

Version : 2.6.6 **Éditeur :** VK Vision SA

Fonctionnalités

Supervision

- **Synoptiques** — Affichage graphique interactif des installations (vue générale, grille, individuel). En mode individuel, le synoptique est mis à l'échelle automatiquement pour occuper la page entière sans scrollbar.
- **Machine 2D** — Visualisation de la configuration machine avec synoptiques détaillés par cuve au clic
- **Machine 3D** — Vue tridimensionnelle interactive de la machine (Three.js) avec navigation à la souris
- **Alarmes / Messages** — Consultation et acquittement des alarmes et messages (2 onglets) ; onglet actif mis en évidence en jaune
- **Tableau de bord** — Statistiques de production : productivité horaire, charges entrées/sorties, fiabilité, utilisation horaire, avec courbes de tendance
- **Courbes de tendance** — Affichage des courbes de variables enregistrées (sélection par arborescence)
- **Informations charges** — Tableau des charges en cours avec barre de progression de temps (T.Actuel vs T.Min)
- **Horloge hebdomadaire** — État de la programmation horaire de la machine : jours de semaine, weekend, modes spéciaux

Contrôle

- **Télécommandes robots** — Commandes de montée, descente et déplacement des robots (support type 17 avec sélection du type de montée/descente)
- **Redresseurs** — Surveillance des redresseurs de courant
- **Pompes doseuses** — Suivi des pompes de dosage chimique
- **Sliders interactifs** — Réglage de consignes, alarmes et seuils de sécurité via des curseurs graphiques

Mode Tableau de bord

Mode d'affichage automatique en rotation, conçu pour les écrans de supervision sans interaction :

- Liste de pages configurable (synoptique individuel, alarmes, redresseurs, pompes, machine 2D+robots, courbes de tendance, etc.)
- Durée d'affichage paramétrable par page
- Sélection automatique du synoptique, de la variable de courbe et de l'onglet alarmes/messages
- En mode machine 2D : affichage des télécommandes robots en-dessous de la machine
- Connexion automatique après un délai configurable
- Option **plein écran automatique** au démarrage du mode
- Activation depuis la page Maintenance

Internationalisation (i18n)

- Langues supportées : **Français** (FR), **Allemand** (DE), **Anglais** (EN)
- Langue par défaut configurable par connexion MQTT
- Changement de langue automatique selon la connexion sélectionnée et le profil utilisateur (login QuickView)
- Retour à la langue de la connexion après déconnexion du compte

Système

- **Connexions MQTT** — Gestion de connexions multiples avec persistance locale (localStorage), langue par défaut par connexion
 - **Authentification** — Login utilisateur avec JWT ; niveau d'accès et langue lus depuis la réponse QuickView
 - **Mode hors-ligne** — Support PWA avec Service Worker et cache des ressources statiques
 - **Responsive** — Interface adaptée aux écrans mobiles, tablettes et desktop ; menu latéral réductible
 - **Plein écran** — Bouton dans le footer (icône `expand/compress`) pour basculer en plein écran ; l'icône se met à jour automatiquement
 - **Maintenance** — Compteurs de messages MQTT entrants/sortants ; clic sur un compteur (utilisateur connecté) → liste des 200 derniers messages (horodatage, topic, payload)
-

Architecture technique

Technologies

Composant	Technologie
UI	HTML5, CSS3, Bootstrap 5
Logique	JavaScript (ES5, IIFE modules)
Communication	MQTT over WebSocket
Authentification	JWT (jose)
Dessin 2D	Canvas API
Dessin 3D	Three.js r134 (WebGL)
Icônes	FontAwesome 6
PWA	Service Worker, manifest.json

Structure des fichiers

```
TestSynopticViewer/  
├─ app.html           Page principale  
├─ manifest.json     Manifest PWA  
├─ sw.js             Service Worker  
├─ assets/  
│   └─ css/  
│       └─ app.css   Styles applicatifs  
│       └─ user.css  Styles personnalisés  
│       └─ bootstrap.min.css Bootstrap 5  
│       └─ all.min.css  FontAwesome  
│   └─ js/  
│       └─ VKApp.js   Logique principale (navigation, MQTT, pages)  
│       └─ VKI18n.js  Internationalisation (FR / DE / EN)  
│       └─ VKMachine3D.js Vue 3D de la machine (Three.js)  
│       └─ VKDialogs.js Dialogues modaux (slider, clavier, popups)  
│       └─ VKDrawing.js Moteur de rendu canvas des synoptiques  
│       └─ VKConnection.js Gestion des connexions MQTT  
│       └─ VK.js      Variables globales et utilitaires  
│       └─ machine.js Configuration machine  
└─ img/              Logos, icônes, favicons
```

└─ Spec/	
└─ Specifications.md	Spécifications fonctionnelles
└─ Json/	Données JSON de test
└─ img/	Maquettes et captures de référence

Modules JavaScript

Module	Rôle
VKApp	Application principale : navigation, connexion MQTT, gestion des pages, graphiques, mode tableau de bord
VKI18n	Internationalisation : dictionnaires FR/DE/EN, <code>t(key)</code> pour traduire, <code>apply()</code> pour mettre à jour le DOM
VKMachine3D	Vue 3D interactive de la machine (Three.js) : cuves creuses, charges, robots, orbite souris, clic-synoptique
VKDialogs	Boîtes de dialogue : slider vertical interactif avec curseurs d'alarme/sécurité, clavier numérique, menus popup
VKDrawing	Rendu graphique des synoptiques sur canvas HTML5
VKConnection	Persistence et gestion des connexions MQTT (localStorage)

Communication MQTT

Topics souscrits en permanence

Topic	Variable JS	Description
<code>{base}Apps/{clientId}/Login</code>	VKQVLogin	Réponse d'authentification (access, lang, jwtoken)
<code>{base}Status/Connection</code>	VKQVConnectionStatus	État de connexion machine (Online/Offline)
<code>{base}Status/Databanks/#</code>	VKQVDatabanks[]	Registres de données temps réel
<code>{base}Configuration/Synoptics/#</code>	VKQVSynoptics[]	Définitions des synoptiques (winjson)
<code>{base}Configuration/Machine</code>	VKQVMachine	Configuration de la machine (positions, robots)
<code>{base}Alarms</code>	VKQVAlarms	Liste des alarmes actives
<code>{base}Messages</code>	VKQVMessages	Liste des messages actifs
<code>{base}LiftLower</code>	VKQVLiftLower	Types de montées/descentes des robots

Topic	Variable JS	Description
<code>{base}Dailydata</code>	—	Statistiques journalières (tableau de bord)

Topics souscrits par page

Topic	Page	Variable JS
<code>{base}Status/Rectifiers</code>	Redresseurs	<code>VKQVRectifiers</code>
<code>{base}Status/Dosing</code>	Pompes doseuses	<code>VKQVDosing</code>
<code>{base}Status/Machine</code>	Machine / Robots	<code>VKQVMachineData</code>
<code>{base}Status/Timetable</code>	Informations charges	<code>VKQVTimetable</code>
<code>{base}Status/DaysClock</code>	Horloge hebdomadaire	<code>VKQVDaysClock</code>
<code>{base}Configuration/Recorders</code>	Courbes de tendance	<code>VKQVRecorders</code>
<code>{base}Status/Recorder/{ionum}</code>	Courbes de tendance	<code>VKQVRecorderData</code>

Topics publiés

Topic	Description
<code>{base}Apps/{clientId}/Status/Connection</code>	État de l'application (Connected / Disconnected)
<code>{base}Apps/{clientId}/Login</code>	Envoi des credentials utilisateur
<code>{base}Apps/{clientId}/Commands</code>	Commandes envoyées aux équipements

Internationalisation (VKI18n)

Le module `VKI18n` est un IIFE exposant deux fonctions :

```
VKI18n.t('key') // Retourne la chaîne traduite dans la langue active
VKI18n.apply() // Met à jour tous les éléments DOM marqués data-i18n
```

Les éléments HTML sont annotés avec :

- `data-i18n="key"` → `element.textContent`
- `data-i18n-placeholder="key"` → `element.placeholder`
- `data-i18n-title="key"` → `element.title`

Flux de résolution de la langue :

1. Démarrage → `FR` (défaut)
2. Sélection d'une connexion dans le dropdown → langue de la connexion (`defaultLang`)
3. Connexion MQTT → langue de la connexion
4. Login QuickView → langue du profil utilisateur (`lang 100=FR, 101=DE, 102=EN`)
5. Logout → retour à la langue de la connexion

Mode Tableau de bord

Activé depuis la page **Maintenance**. L'application tourne en boucle sur une liste de pages configurées, chacune affichée pendant une durée définie.

Pages disponibles dans la rotation :

- Accueil, Commandes générales, Tous les synoptiques
- Synoptique individuel (*sélection du synoptique*)
- Alarmes / Messages (*sélection de l'onglet*)
- Redresseurs, Pompes doseuses
- Tableau de bord, Machine 2D + robots, Machine 3D
- Courbes de tendance (*sélection de la variable*)
- Télécommandes robots, Informations charges, Horloge hebdomadaire

En mode tableau de bord, les synoptiques individuels sont mis à l'échelle pour occuper toute la page (contain-scaling), avec le titre affiché au-dessus. Si l'option plein écran est activée, le navigateur passe automatiquement en plein écran au démarrage du mode (via clic utilisateur ou au premier contact pour le démarrage automatique).

Vue Machine 3D

La vue 3D (`VKMachine3D`) est un rendu WebGL temps réel basé sur **Three.js r134**.

Navigation à la souris

Geste	Action
Clic gauche + glisser	Orbite (rotation autour de la machine)
Shift + clic gauche + glisser	Panoramique horizontal
Molette	Zoom avant / arrière
Pinch (tactile)	Zoom

Éléments dessinés

Élément	Description
Cuves	Boîtes creuses à 5 parois (fond + 4 côtés). Couleur : vert si actif, rouge si alarme, gris si hors process. Étiquettes numéro + nom.
Charges	Cubes colorés à l'intérieur des cuves selon le statut du job (<code>js</code> 0-7). Numéro affiché sur le dessus.
Robots	Structure en L/Γ jaune (style monorail). Numéro affiché en sprite face caméra. Charge portée dessinée sous le bras.
Rails	Monorails horizontaux en hauteur, un par ligne de machine.

Clic sur une cuve

Un clic gauche simple ouvre le synoptique de la cuve dans la zone sous la vue 3D. Le bouton × ferme le synoptique.

Page Horloge hebdomadaire

Affiche la programmation horaire de démarrage/arrêt de la machine, organisée en 3 onglets :

Onglet	Contenu
Semaine	Jours 2 à 6 (lundi → vendredi)
Weekend	Samedi (jour 7) et dimanche (jour 1)
Modes spéciaux	Tableau des actions programmées (jour, heure, mode, actif)

Pour chaque jour : nom complet du jour (le jour courant affiché en rouge), section Start (badge + heure + schéma + timeline), section Stop (badge + heure + schéma + timeline). Les détails ne s'affichent que si le démarrage/arrêt est activé. Les données sont rechargées à chaque visite de la page (pas de cache).

Source MQTT : `{base}Status/DaysClock`

Page Maintenance

- **Messages entrants / sortants** : compteurs mis à jour en temps réel. Si l'utilisateur est connecté (login), un clic sur un compteur ouvre la liste des 200 derniers messages (horodatage, topic, payload), du plus récent au plus ancien.
 - **Mode Tableau de bord** : configuration de la rotation automatique des pages, du délai d'auto-connexion et du plein écran automatique.
 - **Renvoyer les synoptiques** : visible uniquement si connecté à QuickView (accès au niveau d'accès requis).
-

Format des couleurs

Les couleurs provenant de la configuration utilisent le format `#AARRGGBB` (alpha + rouge + vert + bleu) et sont converties en `#RRGGBB` pour le CSS.

Démarrage

1. Héberger les fichiers sur un serveur web (ou ouvrir `app.html` localement)
 2. Configurer une connexion MQTT (host, port, path, SSL, langue par défaut)
 3. Se connecter au broker
 4. Naviguer entre les pages via la barre latérale
-

Licence

© 2026 VK Vision SA — Tous droits réservés.

Revision #2

Created 2026-04-13 15:53:04 UTC by Jean-Noël Voirol

Updated 2026-04-13 15:54:54 UTC by Jean-Noël Voirol