

ObsiGate — Analyse complète & Recommandations V1.1

ObsiGate — Analyse complète & Recommandations V1.1

Date : 2026-05-25 | **Analyste :** Hermes-Claw + Audit Code Source | **Version testée :** 1.4.0 (backend) / 1.5.0 (frontend) — <http://openclaw1.dev.home:2020>

☐ Table des matières

1. [Résumé exécutif](#)
 2. [Fonctionnalités actuelles](#)
 3. [Forces](#)
 4. [Points d'amélioration](#)
 5. [Roadmap suggérée](#)
 6. [Checklist de professionnalisation](#)
 7. [Notes de révision V1.1](#)
-

1. Résumé exécutif

ObsiGate est un gestionnaire web de vaults Obsidian Markdown auto-hébergé. La version actuelle (1.4.0/1.5.0) est un produit **riche et mature**, bien au-delà d'une simple V1. Il dispose d'un moteur de recherche TF-IDF avancé, un éditeur CodeMirror 6 intégré, une vue graphique, un support PWA complet, des opérations CRUD sur fichiers/dossiers, et une synchronisation temps réel via SSE + watchdog.

Note globale : 8.5/10 — Produit fonctionnel, performant et professionnel. Les améliorations restantes sont principalement d'ordre sécuritaire (masquage des secrets, rate limiting) et de polish (documentation API, backlinks, tests automatisés).

Recommandation prioritaire : Traiter les 2 points de sécurité (masquage secrets + rate limiting), puis les enrichissements UX (backlinks, conflits Syncthing).

2. Fonctionnalités actuelles

| # | Fonctionnalité | Statut | Détail | ---|-----|:-----|-----| | 1 | Authentification | ☐ | JWT HS256 + Argon2id, sessions persistantes, "Se souvenir de moi", contrôle d'accès par

vault | | 2 | Multi-vaults | ☐ | Configurables via env vars ou API dynamique, support VAULT et DIR | | 3 | Arborescence de fichiers | ☐ | Navigation hiérarchique avec lazy-loading, compteur d'éléments par dossier, icônes par type | | 4 | Rendu Markdown | ☐ | Conversion HTML via mistune, tables, task lists, footnotes, strikethrough | | 5 | Extraction YAML frontmatter | ☐ | Tous les champs parsés, carte frontmatter pliable avec badges, dates formatées, tags | | 6 | Vue Source (raw) | ☐ | Toggle markdown brut + frontmatter | | 7 | Bouton Copier | ☐ | Copie du contenu source | | 8 | Bouton Télécharger | ☐ | Téléchargement du fichier | | 9 | Onglet RÉCENT | ☐ | Onglet sidebar dédié + dashboard avec timestamps relatifs, previews, tags, bookmarks | | 10 | Onglet TAGS | ☐ | Nuage de tags cliquable, compteurs, filtrage des tags template configurables | | 11 | Recherche avancée | ☐ | Moteur TF-IDF, opérateurs (tag: , vault: , title: , path: , ext:), snippets `<mark>` , facettes, pagination, tri | | 12 | Autocomplétion de recherche | ☐ | Dropdown avec historique (localStorage), suggestions fichiers + tags, navigation clavier | | 13 | Filtre fichiers sidebar | ☐ | Filtre par nom, option case-sensitive, recherche sur l'index des chemins | | 14 | Thème Sombre | ☐ | Toggle clair/sombre, persisté localStorage, highlight.js adaptatif | | 15 | Administration utilisateurs | ☐ | Interface CRUD complète (admin uniquement) | | 16 | Configurations | ☐ | Paramètres recherche, backend, watcher, tags, fichiers cachés, diagnostics, à propos | | 17 | Table des matières | ☐ | Panneau droit avec scroll-spy, IntersectionObserver, barre de progression de lecture | | 18 | Bookmarks | ☐ | Dashboard avec favoris, toggle depuis les cartes de fichiers | | 19 | Historique d'ouverture | ☐ | Fichiers récemment ouverts par utilisateur, persisté sur disque | | 20 | Éditeur CodeMirror 6 | ☐ | Coloration syntaxique multi-langages, autocomplétion, Ctrl+F, Ctrl+S | | 21 | Création fichiers/dossiers | ☐ | Via menu contextuel (clic droit) et modals dédiées | | 22 | Renommage fichiers/dossiers | ☐ | Renommage inline via menu contextuel | | 23 | Suppression fichiers/dossiers | ☐ | Avec confirmation, suppression récursive pour les dossiers | | 24 | Menu contextuel | ☐ | Clic droit : copier chemin, vue graphique, créer, renommer, supprimer | | 25 | Wikilinks cliquables | ☐ | `[[lien]]` et `[[lien|alias]]` convertis en liens navigables, liens brisés en rouge | | 26 | Images Obsidian | ☐ | Résolution intelligente (7 stratégies), support `![[image]]` , `![alt](path)` , `<img>` dans liens | | 27 | Vue Graphique | ☐ | Canvas force-directed, nœuds fichiers/dossiers, arêtes parent + wikilinks, zoom/pan | | 28 | Onglets de contenu | ☐ | Barre d'onglets multi-fichiers | | 29 | Fenêtres popout | ☐ | Vue standalone avec éditeur, TOC, thème sombre, login intégré | | 30 | Find in Page | ☐ | Barre Ctrl+F avec case-sensitive, mot entier, regex | | 31 | PWA | ☐ | Service worker, manifest.json, icônes, notification de mise à jour | | 32 | SSE — Synchronisation temps réel | ☐ | Reconnexion automatique, panneau d'état, événements index/vault | | 33 | Watchdog — Surveillance fichiers | ☐ | inotify natif + fallback polling, debounce configurable, mise à jour incrémentale | | 34 | Filtrage dossiers/fichiers cachés | ☐ | Par vault (`hideHiddenFiles`), paramétrable dans Configurations | | 35 | Gestion dynamique des vaults | ☐ | Ajout/suppression via API sans redémarrage | | 36 | API REST | ☐ | 20+ endpoints (CRUD fichiers, recherche, tags, graph, bookmarks, config, diagnostics...) | | 37 | Sécurité | ☐ | Headers CSP/X-Frame/XSS, protection path traversal, non-root, permissions vault | | 38 | Toast notifications | ☐ | Feedback utilisateur (succès, erreur) | | 39 | Barre de progression d'indexation | ☐ | Affichée dans le header pendant l'indexation

initiale | | 40 | Breadcrumb navigable | ☐ | Cliquable, repositionne automatiquement la sidebar | | 41 | Redimensionnement sidebar | ☐ | Drag handle, persisté localStorage, limites min/max | | 42 | Docker multi-plateforme | ☐ | linux/amd64, arm64, arm/v7, i386 |

3. Forces

☐ Design & UX

- **Interface épurée** — layout 3 colonnes (sidebar | contenu | TOC) avec onglets sidebar
- **Français natif** — rare et appréciable pour un outil de cette catégorie
- **Responsive** — s'adapte aux différentes tailles d'écran, menu hamburger mobile
- **Feedback visuel** — timestamps relatifs ("il y a 5 min", "il y a 3 h"), toasts, barres de progression
- **PWA** — installation sur l'écran d'accueil, service worker, notifications de mise à jour

☐ Métadonnées

- **Extraction YAML complète** — tous les champs du frontmatter parsés et affichés dans une carte pliable stylisée
- **Tags cliquables** — dans la vue fichier, l'onglet RÉCENT, et le nuage de tags
- **Résolution wikilinks** — O(1) via table de lookup, support multi-vaults
- **Images** — résolution multi-stratégies (7 stratégies) des syntaxes Obsidian

☐ Architecture

- **Multi-vaults** — supporte plusieurs vaults simultanément avec contrôle d'accès par utilisateur
 - **Auto-détection** — découverte automatique des vaults via variables d'environnement
 - **API dynamique** — ajout/suppression de vaults sans redémarrage
 - **Performance** — TF-IDF avec ThreadPoolExecutor, index mémoire, lookup O(1), SSE temps réel
 - **Stack moderne** — FastAPI + mistune + watchdog (backend), Vanilla JS SPA + CodeMirror 6 + highlight.js (frontend)
-

4. Points d'amélioration

4.1 ☐ Critique — Sécurité

4.1.1 Masquage automatique des secrets

Problème : Les clés API, tokens et secrets sont visibles en clair dans les aperçus de contenu et la vue RÉCENT.

Exemple concret observé : Fichier "AI API" → preview contient des clés OpenAI en clair.

Solution : Implémenter un `SecretRedactor` (backend) qui scanne le contenu avant affichage :

- Patterns : OpenAI/Claude keys, JWT tokens, GitHub tokens, clés hex/base64 > 40 chars
- Preview RÉCENT/Dashboard : Masquer toutes les clés
- Vue Source : Warning "Ce fichier contient N secrets potentiels. [Afficher]"
- Vue Rendu : Masquer avec placeholder visuel
- Export/Téléchargement : Conserver le contenu original

Priorité : ☐ P0 — Corriger avant toute exposition publique

4.1.2 Rate limiting & protection brute-force

Problème : Absence de rate limiting sur le login.

Solution : Middleware rate-limit — 5 tentatives max par IP sur 15 minutes, blocage après 10 échecs.

Priorité : ☐ P0

4.1.3 Filtrage des dossiers sensibles — Amélioration

État actuel : VaultWatcher ignore déjà `.obsidian`, `.trash`, `.git`, `__pycache__`, `node_modules`. Le paramètre `hideHiddenFiles` par vault masque les fichiers/dossiers commençant par `.`.

Amélioration : Rendre la liste `IGNORED_DIRS` configurable.

Priorité : ☐ P2 — Déjà partiellement résolu

4.2 ☐ Important — Performance & Robustesse

4.2.1 Log d'audit

Pas de traçabilité des actions critiques (écritures, suppressions, changements de config).
Logger les actions avec utilisateur, timestamp, vault, chemin dans `data/audit.log`.

Priorité : ☐ P1

4.2.2 Backup automatique avant écriture

Pas de filet de sécurité avant modification/suppression. Sauvegarder le contenu original dans `.obsigate-backup/` horodaté.

Priorité : ☐ P1

4.2.3 Timeout de session configurable

TTL JWT codé en dur. Rendre configurable.

Priorité : ☐ P2

4.3 ☐ UX & Navigation

4.3.1 Backlinks panel

Afficher les fichiers contenant des wikilinks pointant vers le fichier courant. Construire un index inversé des wikilinks au scan initial.

Priorité : ☐ P2

4.3.2 Correction TOC — scroll avec caractères accentués

Le clic sur un titre dans la TOC ne fonctionne pas pour certains titres accentués. Les fonctions slugify backend/frontend utilisent des normalisations Unicode qui divergent.

Solution : Unifier l'algorithme + fallback `scrollIntoView`.

Priorité : ☐ P2

4.3.3 Gestion fichiers non-supportés

Les fichiers binaires non supportés déclenchent une erreur 500. Ajouter un message explicite avec bouton de téléchargement.

Priorité : ☐ P3

4.4 ☐ Fonctionnel — Feature Gaps

4.4.1 Gestion des conflits Syncthing

Les fichiers `.sync-conflict-*` sont indexables mais pas d'outil de résolution. Ajouter section « Conflits » avec diff et choix (garder local/conflit).

Priorité : ☐ P3

4.4.2 Dashboard statistiques

Dashboard avec métriques par vault : fichiers totaux, taille, top tags, orphelins.

Priorité : ☐ P4

4.5 ☐ Architecture & Code

4.5.1 Documentation OpenAPI/Swagger enrichie

L'API REST (20+ endpoints) est documentée via docstrings Python. FastAPI génère automatiquement `/docs` et `/redoc`. Enrichir les modèles Pydantic.

Priorité : ☐ P2

4.5.2 Webhook / Event system

Notifier des systèmes externes lors de changements (création, modification, suppression).

Priorité : ☐ P3

4.5.3 Authentification multi-facteurs

TOTP ou WebAuthn pour l'accès admin.

Priorité : ☐ P5

5. Roadmap suggérée

Phase 1 — Correctifs urgents (1-2 semaines)

#	Tâche	Effort
1	Redacteur de secrets dans les aperçus	2h
2	Rate limiting sur le login	2h
3	Log d'audit (écritures, suppressions)	3h
4	Backup automatique avant écriture	3h

Phase 2 — UX (2-4 semaines)

#	Tâche	Effort
5	Backlinks panel	6h
6	Correction TOC scroll avec accents	3h
7	Gestion des conflits Syncthing (diff + résolution)	8h
8	Liste IGNORED_DIRS configurable	2h
9	Gestion élégante des fichiers non supportés	2h

Phase 3 — Avancé (4-8 semaines)

#	Tâche	Effort
10	Dashboard statistiques	6h
11	Webhooks	5h
12	Enrichissement doc OpenAPI/Swagger	3h
13	Timeout de session configurable	2h

Phase 4 — Polish (continu)

#	Tâche	Effort
14	Tests automatisés (pytest + Playwright)	16h
15	CI/CD pipeline	4h
16	Docker image officielle (registry)	3h
17	i18n (anglais + français)	8h
18	Documentation utilisateur enrichie	8h

6. Checklist de professionnalisation

☐ Sécurité

- ☐ Redacteur de secrets (API keys, tokens, passwords)
- ☒ Exclusion dossiers .obsidian/, .trash/, .git/ (watcher + hideHiddenFiles)
- ☐ Rate limiting sur le login
- ☒ CSP headers (Content-Security-Policy)
- ☐ HTTPS obligatoire en production (reverse proxy recommandé)
- ☐ Session timeout configurable
- ☐ Log d'audit (qui a fait quoi quand)
- ☐ Backup automatique avant écriture

☐ Performance

- ☒ Moteur TF-IDF avec normalisation d'accents
- ☒ Lazy-loading de l'arborescence
- ☒ Pagination des résultats de recherche
- ☒ Table de lookup O(1) pour wikilinks

☐ UX

- ☒

Wikilinks cliquables

- ☒ Tags cliquables
- ☒ Breadcrumb navigable
- ☒ Mode sombre persistant (localStorage)
- ☒ Raccourcis clavier (Ctrl+K, Ctrl+T)
- ☐ Drag & drop pour upload d'images
- ☒ Notifications toast
- ☐ Page 404 personnalisée
- ☒ États vides stylisés

☐ Fonctionnel

- ☒ Éditeur CodeMirror 6
- ☒ Création fichiers/dossiers (menu contextuel)
- ☒ Renommage fichiers/dossiers (menu contextuel)
- ☒ Suppression fichiers/dossiers (menu contextuel)
- ☒ API REST complète (20+ endpoints)
- ☐ Documentation OpenAPI enrichie
- ☐ CLI companion (obsigate-cli)
- ☐ Gestion conflits Syncthing (diff + résolution)
- ☐ Export multi-formats (PDF, HTML standalone)

☐ Qualité

- ☐ Tests unitaires (pytest)
- ☐ Tests E2E (Playwright)
- ☐ CI/CD (GitHub Actions)
- ☒ Image Docker (Dockerfile)
- ☐

Documentation utilisateur

- ☐ CHANGELOG.md
- ☐ Semantic versioning

☐ Communauté

- ☒ Page Git avec README
 - ☐ Capture d'écran animée (GIF/WebM)
 - ☒ Badges (version, license)
 - ☒ Guide de contribution (CONTRIBUTING.md)
 - ☐ Template d'issue (bug / feature request)
-

7. Notes de révision V1.1

Corrections majeures par rapport à l'analyse V1 (qui sous-estimait largement le projet) :

Analyse V1	Réalité V1.1
"Feature Gap — Éditeur Markdown (CodeMirror) Phase 3"	☐ Déjà implémenté avec 15+ langages, autocomplétion, Ctrl+F, Ctrl+S
"Feature Gap — Création fichiers/dossiers Phase 3"	☐ Déjà fonctionnel via menu contextuel + modals
"Feature Gap — Wikilinks cliquables Phase 2"	☐ Déjà fonctionnels avec résolution multi-vaults et liens brisés
"Feature Gap — Tags cliquables Phase 3"	☐ Déjà fonctionnels
"Feature Gap — Breadcrumb navigable Phase 2"	☐ Déjà implémenté
Recherche notée "performances à améliorer"	☐ Moteur TF-IDF complet avec opérateurs avancés, facettes, autocomplétion
"Filtrage dossiers sensibles non implémenté"	☐ Déjà présent (watcher + hideHiddenFiles par vault)
Non mentionné : PWA	☐ Service worker, manifest, icônes, notif de mise à jour
Non mentionné : Vue Graphique	☐ Canvas force-directed avec wikilinks
Non mentionné : Find in Page	☐ Ctrl+F avec case-sensitive, mot entier, regex
Non mentionné : Onglets de contenu	☐ Barre d'onglets multi-fichiers
Non mentionné : Fenêtres Popout	☐ Standalone avec éditeur, TOC, login
Non mentionné : Menu contextuel	☐ Clic droit avec CRUD complet
Non mentionné : Bookmarks	☐ Toggle + dashboard
Non mentionné : Toast notifications	☐ Feedback succès/erreur
Non mentionné : Résolution images	☐ 7 stratégies pour syntaxes Obsidian

Note globale révisée : 7/10 → 8.5/10

Analyse réalisée le 2026-05-25 — Révisée V1.1 après audit complet du code source
Document : ANALYSE_REVIEW.md