

Intégrations externes

Intégrations externes

“ Dernière mise à jour : 2026-05-10

Pour chaque service tiers : à quoi il sert, endpoints consommés, où sont les credentials, quel est le mode dégradé.

Qdrant

- **Rôle** : Vector DB. Une collection par jeu (`rules_<slug>`) + une par TCG (`magic-cards`, etc.).
- **Endpoint** : `QDRANT_URL` (ex. `http://192.168.10.100:6333`)
- **Credentials** : aucun (cluster privé LAN, accès filtré au niveau réseau Docker)
- **Vecteurs** : dense 1024 (bge-m3) + sparse BM25 natif
- **Mode dégradé** : aucun. Si Qdrant est down, le retrieval échoue, l'app renvoie 500.
- **Health** : `GET /collections` (vérifié par `/api/admin/health`)

TEI bge-m3 (embeddings)

- **Rôle** : Génère les embeddings dense 1024 dim pour les questions et chunks.
- **Endpoint** : `TEI_URL` (ex. `http://192.168.10.100:8099`)
- **Credentials** : aucun
- **Modèle** : `BAAI/bge-m3` servi par TEI (HuggingFace)
- **Latence typique** : ~50-100ms pour un batch de 32 sur RTX 3060
- **Mode dégradé** : aucun fallback. Sans TEI : pas de retrieval.
- **Health** : `GET /health`

TEI Reranker bge-v2-m3

- **Rôle** : Cross-encoder qui re-classe les top-50 candidats fusionnés (RRF) avant la génération.
- **Endpoint** : `TEI_RERANKER_URL` (ex. `http://192.168.10.100:8990`)
- **Credentials** : aucun
- **Modèle** : `BAAI/bge-reranker-v2-m3`
- **Mode dégradé** : si reranker down, on retombe sur le score RRF brut (suboptimal mais fonctionnel).

Claude Code CLI (Anthropic)

- **Rôle** : Génération streamée des réponses RAG, HyDE, decompose-query, contextual retrieval, hierarchy, conflict detect, classify intent.
- **Accès** : SSH ed25519 vers la VM `oracle` (compte dédié).
- **Endpoint** : `CLAUDE_SSH_HOST` + `CLAUDE_SSH_USER=oracle` + clé privée `CLAUDE_SSH_KEY_PATH`
- **Credentials Anthropic** : stockés sur la VM dans `/home/oracle/.claude/.credentials.json` (compte personnel Pro/Team)
- **Modèles** :
 - `claude-haiku-4-5-20251001` : HyDE, decompose, classify, contextual, conflict
 - `claude-opus-...` (par défaut) : réponses RAG, deckbuilding
- **Quota** : géré par `services/claude-quota.ts`. Pause auto + reprise programmée à `resetAt + 2 min`.
- **Sécurité** : ForceCommand wrapper `/home/oracle/bin/oracle-claude.sh` valide préfixe `cd /home/oracle/work && claude . validateModel()` côté backend bloque l'injection.

BoardGameGeek (XML API v2)

- **Rôle** : Récupère metadata jeux (image, mécaniques, catégories), liste extensions, scrape forums "Rules" pour ingest dans le RAG.
- **Endpoint** : [https://boardgamegeek.com/xmlapi2/...](https://boardgamegeek.com/xmlapi2/)
- **Credentials** : `BGG_API_TOKEN` (optionnel). Sans token, rate-limit plus strict mais fonctionnel.
- **Cache** : hot games 6h (mémoire), forums via cron quotidien (configurable)
- **Mode dégradé** : sans BGG, tu peux toujours créer un jeu manuellement (skip étape 1 du wizard).

Sources de cartes par TCG

TCG	Endpoint / source
-----	-------------------

MTG	https://api.scryfall.com/bulk-data → JSON <code>all_cards</code> (téléchargé localement)
Lorcana	https://github.com/LorcanaJSON/LorcanaJSON (raw GitHub, MIT)
FAB	npm <code>@flesh-and-blood/cards</code> + <code>@flesh-and-blood/types</code> (bundlé dans l'image Docker)
Riftbound	https://riftbound.leagueoflegends.com/en-us/card-gallery (API JSON Riot)
Terraforming Mars	HTML parsing local + cards.json
Ark Nova	JSON local + sprites découpées

Sources méta-game

Source	TCG	Méthode
17Lands	MTG (draft analytics)	API publique
MTGGoldfish	MTG (constructed metagame)	Scrape (Cheerio) + rate-limit 1.5s
MTGTop8	MTG (top 8s tournois)	Scrape + rate-limit 1.5s
Mobalytics	Riftbound (tier list)	Scrape
RiftboundStats	Riftbound (tournois)	API
fabtcg.com	FAB (tournois LSS)	Scrape (header <code>META_FAB_USER_AGENT</code> Cloudflare-compatible)

Tous gérés via `src/services/meta/*.ts`. Cron `meta-sync.ts` pilote la fréquence (par défaut hebdo, configurable).

Email (SMTP)

- **Rôle** : Notification admin au register, password reset.
- **Credentials** : `SMTP_HOST`, `SMTP_PORT`, `SMTP_USER`, `SMTP_PASS`, `SMTP_FROM`, `SMTP_SECURE`
- **Mode dégradé** : si `SMTP_HOST` vide, tous les emails sont silencieusement skippés. L'admin doit checker `/admin` à la main.

Reverse proxy (Nginx Proxy Manager)

- **Rôle** : TLS Let's Encrypt + reverse proxy `rules.thymon.fr` → container `:3000`.
- **Pas dans le code** : config NPM via UI, hors scope du repo.
- **Heartbeats SSE 8s** : nécessaires pour ne pas se faire fermer la connexion par le `proxy_read_timeout 60s` par défaut.
- **TRUST_PROXY=true** : Hono fait confiance aux `X-Forwarded-*` headers de NPM.

Aucune autre intégration

- Pas de Sentry / Datadog / monitoring tiers
- Pas de S3 / cloud storage (tout en local sur Unraid)
- Pas de service d'analytics (Umami pourrait être ajouté facilement, pas en place actuellement)
- Pas de webhook entrant (tout est trigger soit user, soit cron)

Revision #1

Created 2026-05-10 15:19:58 UTC by thymon

Updated 2026-05-10 15:19:58 UTC by thymon