Surface d'exposition

Surface d'exposition

> *Dernière mise à jour : 2026-05-10*

Endpoints publics vs internes

Public (via NPM)

- **Tout** `/api/*` est exposé sur `https://rules.thymon.fr`
- **Frontend SPA** servi à `https://rules.thymon.fr/`
- Mais : auth requise sur quasi tous les endpoints (sauf `/api/health`, `/api/auth/login`, `/api/auth/register`, `/api/confirm-user/:token`)

Interne (LAN seulement)

- **Container app** `:3000` : pas exposé sur l'hôte (`docker-compose.yml` n'a pas de `ports:`)
- **Qdrant** `:6333` : exposé sur LAN (`192.168.10.100:6333`) pour que d'autres apps puissent le consommer (pas idéal — voir ci-dessous)
- **TEI** `:8099`**, Reranker** `:8990` : LAN seulement

CORS

`config.CORS_ORIGIN` whitelist :

- Origines exactes : `https://rules.thymon.fr`
- CIDR IPv4 LAN : `192.168.10.0/24` (matche n'importe quelle origine `http://192.168.10.X:Y`)

Parsing : `splitAndTrimArray()` + matching CIDR via `ipaddr.js`. Parsé une fois au boot, comparé à chaque requête.

```
# Exemple .env prod
CORS_ORIGIN=https://rules.thymon.fr,192.168.10.0/24
```

⚠ **Ne pas laisser `CORS_ORIGIN=*` en prod** — annule la protection.

Headers sécurité

Gérés par **NPM en amont** (Advanced → Custom Nginx Configuration) :

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
Content-Security-Policy: default-src 'self'; ... (à durcir si nécessaire)
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
Referrer-Policy: strict-origin-when-cross-origin
Permissions-Policy: ...
```

Vérifier : `curl -I https://rules.thymon.fr`.

Ports ouverts (Unraid host)

À auditer périodiquement avec `nmap -p- localhost` depuis Unraid CLI :

Port	Service	Exposition
80, 443	NPM (entrée publique)	Internet
22	SSH Unraid	LAN seulement (firewall)
6333	Qdrant	LAN
8099	TEI	LAN
8990	TEI Reranker	LAN
3000	App	LAN seulement

⚠ **Qdrant exposé sur LAN sans auth** : si quelqu'un a accès au LAN, il peut lire / modifier toutes les collections vectorielles. Acceptable pour un home network privé, mais à durcir si le LAN devient moins de confiance (genre invités, IoT) :

- Activer auth Qdrant : `QDRANT__SERVICE__API_KEY=<random>` côté config Qdrant
- Mettre cette clé dans `QDRANT_API_KEY` env var côté backend
- Le client Qdrant la passe en header `api-key`

Injection / validation

- **Inputs API** : tous validés via Zod (`src/lib/schemas.ts`). UUID v4 partout, longueurs bornées, enums stricts.
- **SQL injection** : impossible — Drizzle ORM utilise des prepared statements. Pas de SQL string-concat.
- **Shell injection (Claude SSH)** : `validateModel()` valide tout `model` user-provided avant interpolation. Le wrapper oracle valide le préfixe strict. Le system prompt est passé via stdin JSON, pas en argument.
- **Path traversal (page-image)** : `GET /api/games/:id/page-image/:page` valide que `:page` est un nombre, et résout via `resolvePageImageFile()` qui contraint au dossier `/app/pdfs/images/<slug>/`. Pas d'accès `../../etc/passwd` possible.

Rate limiting

- **Login** : 5 tentatives / IP / 15 min
- **Ask** : 20 questions / user / min
- **Deck parse** : 10 / user / min

Rate-limiters in-memory (Map). Restart container = compteurs reset.

Couverture log

Les actions sensibles sont logées :

- Login (succès et échec) avec IP + username
- Création / suppression de user
- Action admin (sync cards, export feedback, send password reset)
- Erreurs SSH / quota / wrapper rejet

Filtrer dans `/app/data/logs/server.log` :

```
grep -E "auth|admin|ssh|quota" /app/data/logs/server.log
```

CVE / dépendances vulnérables

`npm audit` régulièrement. Niveau acceptable : `--audit-level=high`. Critical → fix immédiat.

Pas de scanner CI auto actuellement. À ajouter : un job `audit` dans `.gitea/workflows/build.yml` qui fail sur high+.

Pas de WAF

NPM ne fait pas de WAF (rules custom Nginx limitées). Si un jour tu veux du WAF :

- Cloudflare en frontal (mais SSL passthrough OK avec NPM)
- ModSecurity dans NPM

Pour l'usage actuel (single user, LAN-friendly), c'est overkill.

Données collectées / RGPD

Ce que l'app stocke :

- **Users** : username, email, password hash, role
- **Questions** : contenu de la question, réponse Claude, vote, comment, diagnostics RAG
- **PDFs** : règles de jeux uploadés (potentiellement copyrightées — usage personnel acceptable)
- **Sessions** : in-memory, perdues au restart

Pas d'analytics tiers (Google Analytics, Hotjar, etc.). Logs serveur en local.

Pour un usage perso, RGPD n'est pas un sujet. Si tu ouvres à des tiers :

- Politique de confidentialité claire
- Droit à l'effacement (DELETE user supprime cascade les questions)
- Droit d'accès (export CSV des questions de l'utilisateur — pas implémenté actuellement)

Revision #1
Created 2026-05-10 15:20:20 UTC by thymon
Updated 2026-05-10 15:20:20 UTC by thymon