

Vue d'ensemble

- [Choix structurants](#)
- [Présentation du projet](#)
- [Schéma d'architecture](#)
- [Stack technique](#)

Choix structurants

Choix structurants

📅 Dernière mise à jour : 2026-05-10

Liste rapide des décisions d'architecture qui définissent le projet. Chaque entrée pointe vers son ADR détaillé dans le chapitre **Adrs**.

Choix	Pourquoi	ADR
Hono plutôt qu'Express/Fastify	SSE natif, perfs, types meilleurs, moins de middleware tiers	ADR-001
SQLite + Drizzle plutôt que Postgres	Self-host minimal, zéro service supplémentaire, ACID suffisant pour l'usage actuel	ADR-002
Claude via SSH plutôt que API Anthropic directe	Quota Pro/Team du compte personnel (vs API à crédits), accès à l'outil <code>Read</code> de Claude Code pour la vision PDF, no-coût marginal sur appels	ADR-003
Vision inline au moment de la question, pas pendant l'ingestion	Coût zéro à l'ingestion, lecture PNG ciblée 1 page max au moment opportun, latence acceptable (~30s)	ADR-004
RAG Fusion v2 (pondération question×2 + blending position-aware)	Préserve les exact-matches de la question brute contre la dilution HyDE, et garde le signal RRF quand le reranker est peu confiant sur du contenu technique abstrait	ADR-005
Repository pattern (Phase 2)	Centralisation des requêtes Drizzle dans <code>repositories/</code> , jamais d'accès direct depuis routes/services/cron	ADR-006
Handlers MVC (Phase 4)	Séparation routes (HTTP) → handlers (logique pure) → services → repos. Result discriminés pour les erreurs attendues	ADR-007

Autres décisions notables (sans ADR formel)

- **TEI bge-m3 plutôt qu'Ollama mxbai-embed-large** : initialement Ollama, switch vers TEI pour les perfs (GPU dédié + serving optimisé HuggingFace) et le support BM25 sparse natif Qdrant.
- **Contextual Retrieval B** plutôt qu'A : full-LLM par chunk avec document complet + position hiérarchique. Plus cher mais qualité max sur les règles de jeux où le contexte est critique.
- **Pas de Helmet Hono** : tous les headers sécurité (HSTS, CSP, X-Frame-Options) sont gérés par NPM en amont. Évite la duplication.
- **CORS whitelist + CIDR LAN** : support des origines exactes ET des plages IPv4 (`192.168.10.0/24`) pour autoriser le subnet local sans exposer publiquement.
- **hasCardDatabase** : colonne nullable sur `games` qui pointe vers la collection Qdrant correspondante (`magic-cards` , etc.). Permet d'activer dynamiquement l'autocomplete `@card` , le deck import, le mode deckbuilding par jeu.
- **Sticky mentions** plutôt que long context window : on borne explicitement (cap 20 ou 80 si deck) au lieu de balancer tout l'historique. Maîtrise des tokens, prévisibilité du coût.
- **Logger central + env vars centralisées** (`config.ts`) : aucun `console.*` ni `process.env.*` ailleurs. Garantit que les logs sont filtrables et que les vars sont validées au boot.

Présentation du projet

Présentation du projet

📅 Dernière mise à jour : 2026-05-10

But

Webapp RAG (Retrieval-Augmented Generation) self-hosted qui répond en français aux questions sur les règles de jeux de société et de TCG, en citant les sources (page exacte du livret PDF, cartes, règles d'extensions).

Public

Initialement : Thymon en solo pendant ses parties / lors de l'analyse de decks. Architecture multi-utilisateur avec auth en place (argon2 + sessions cookies HTTP-only) pour future ouverture.

État

- **Production** : tourne sur Unraid de Thymon, exposée sur <https://rules.thymon.fr> via Nginx Proxy Manager.
- **Stack mature** : 6401 lignes TypeScript backend, 50 composants Vue, 34 endpoints API, 6 TCG intégrés, 4 phases de roadmap RAG livrées.
- **CI/CD** : Gitea Runners, build multi-stage, push registry Gitea self-hosted.

Historique court

Date	Étape clé
2026 (début)	Setup Hono + Vue + Qdrant, premier RAG simple
2026 Q1	Embeddings TEI, contextual retrieval B, hierarchy LLM
2026-04	Vision inline (Claude lit PNG), HyDE, RAG Fusion v2, sticky mentions, deck import FAB, mode deckbuilding
2026-04-26	Méta-game (17Lands, MTGGoldfish, MTGTop8, Mobalytics, RiftboundStats)
2026-04-27	Lorcana intégration full premium, Phase 2 OCR roadmap préparée
2026-05-06	OCR auto Phase 1 (tesseract)
2026-05-09	Sécurité SSH renforcée (user oracle dédié, ForceCommand wrapper)

Caractéristiques clés

- **Self-hosted total** : aucune dépendance cloud sauf appels Claude via VM oracle (qui est aussi self-hosted, mais le compte Anthropic est externe).
- **Multilingue** : règles FR ou EN par jeu, HyDE bilingue, traduction question si nécessaire.
- **Vision inline** : Claude lit la PNG de la page la plus pertinente directement via son outil `Read` côté VM SSH (zéro pré-ingestion vision).
- **Méta-game** : ingest tier lists + tournois (17Lands, MTGGoldfish, MTGTop8, Mobalytics, RiftboundStats, fabtcg.com) → indexés en chunks `[META]` aux côtés des règles.
- **Mode deckbuilding** : intent dédié, spec Haiku, prompt Opus spécialisé, validation structurelle + retry auto.
- **Sécurité SSH** : 4 couches (user oracle séquestré, ForceCommand wrapper, validateModel(), ed25519 dédiée).

Schéma d'architecture

Schéma d'architecture

📅 Dernière mise à jour : 2026-05-10

Vue containers

```
flowchart LR
    User[Utilisateur Web]
    NPM[Nginx Proxy Manager<br/>rules.thymon.fr]
    App[Container boardgame-referee<br/>Hono + Vue dist + SQLite + PDFs]
    Qdrant[Container Qdrant<br/>:6333]
    TEI[TEI bge-m3<br/>:8099 RTX 3060]
    Reranker[TEI Reranker bge-v2-m3<br/>:8990 RTX 3060]
    SSH[VM oracle<br/>Claude Code CLI<br/>+ Read tool sur /app/pdfs]

    User -- HTTPS --> NPM
    NPM -- HTTP :3000 --> App
    App -- HTTP :6333 --> Qdrant
    App -- HTTP :8099 --> TEI
    App -- HTTP :8990 --> Reranker
    App -- SSH ed25519 --> SSH
    App -- volume PDF + PNG --> SSH
```

Vue d'une question (flux RAG)

```
sequenceDiagram
    participant U as Frontend Vue
```

```

participant H as Hono /api/ask/stream
participant CLF as classify (Haiku)
participant HYDE as HyDE (Haiku)
participant TEI as TEI bge-m3
participant Q as Qdrant (dense+BM25)
participant R as Reranker
participant SSH as Claude Code SSH

U->>H: POST { gameId, question, cardMentions, history }
H->>CLF: classify(question)
par parallèle
    H->>HYDE: génère passage hypothétique
    H->>TEI: embed question brute
end
HYDE-->>H: passage
H->>TEI: embed passage HyDE
H->>Q: search hybride (dense × 2 vecteurs + BM25)
Q-->>H: candidats RRF v2 fusionnés
H->>R: rerank top-50
R-->>H: top-K final + scores
H->>SSH: prompt Opus avec chunks + cartes citées + image PNG
SSH-->>H: stream tokens
H-->>U: SSE phases / context / token / done
H->>H: persist questions.answer + diagnostics

```

Vue d'une ingestion (flux PDF → Qdrant)

```

flowchart TB
    PDF[PDF uploadé]
    Extract[pdfjs-dist : extract text]
    OCR{Auto OCR<br/>nécessaire ?}
    Tess[tesseract +<br/>pdftoppm 300dpi]
    Chunk[Chunking sémantique]
    Hier[Hierarchy LLM<br/>chapter / section]
    Ctx[Contextual LLM B<br/>10 SSH parallèles]

```

```

Embed[TEI bge-m3<br/>batch 32]
QdrantUp[Qdrant upsert<br/>rules_slug]
Confl{Extension ?}
ConflDetect[Conflict detect<br/>vs jeu base]
PNG[pdftoppm rendu<br/>page-XX.png 300dpi]

PDF --> Extract
Extract --> OCR
OCR -- oui --> Tess --> Chunk
OCR -- non --> Chunk
Chunk --> Hier --> Ctx --> Embed --> QdrantUp
QdrantUp --> Confl
Confl -- oui --> ConflDetect --> PNG
Confl -- non --> PNG

```

Couches backend

```

src/
├─ routes/           ← Contrats HTTP (Hono), validation Zod, auth
├─ handlers/         ← Logique métier pure (Phase 4 MVC), retourne Result discriminés
├─ services/         ← Domaine : RAG, Qdrant, TEI, Claude SSH, cards, méta, OCR
└─ repositories/     ← Data access (Drizzle) – SEUL endroit qui importe drizzle-orm

```

Règles : `routes` ne fait que parser+valider+déléguer. `handlers` ne connaît pas Hono. `services` ne touche pas la DB. `repositories` ne contient pas de logique métier.

Couches frontend

```

frontend/src/
├─ views/            ← Pages routables (HomeView, PlayView, AdminView...)
├─ components/        ← Composants par domaine (admin/, play/, deck-import/, card-zoom/, home/)
├─ composables/       ← Hooks logique métier réutilisable (useAskStream,
useMentionAutocomplete...)
├─ stores/            ← Pinia (auth, games, session)
├─ services/          ← `api.ts` (client unique vers backend)
└─ lib/               ← Helpers TCG-specific (mana.ts, fab-symbols.ts, ...)

```

Stack technique

Stack technique

📅 Dernière mise à jour : 2026-05-10

Runtime

Composant	Version	Rôle
Node.js	22 (image <code>node:22-bookworm-slim</code>)	Runtime backend
Vue 3	3.5.31	Framework frontend
Tailwind CSS	4.2.2 (<code>@tailwindcss/vite</code>)	Utility-first CSS
Vite	(via <code>@tailwindcss/vite</code>)	Bundler frontend
Pinia	3.0.4	State management
Vue Router	5.0.4	Routing client

Backend principal

Lib	Version	Rôle
Hono	4.7.6	Web framework (routes + middleware + SSE streaming)
<code>@hono/node-server</code>	1.14.1	Adapter HTTP Node
Drizzle ORM	0.44.2	ORM SQLite type-safe
<code>better-sqlite3</code>	11.9.1	Driver SQLite synchrone (rapide, low-overhead)
Zod	3.24.4	Validation schémas
Argon2	0.43.0	Password hashing

Lib	Version	Rôle
ssh2	1.16.0	Client SSH (pool vers VM oracle)
pdfjs-dist	4.10.38	Extraction texte PDF
sharp	0.34.5	Resize cache images cartes
cheerio	1.2.0	HTML parsing (méta-game scraping)
nodemailer	8.0.5	SMTP optionnel
stream-chain + stream-json	3.6.1 / 1.9.1	Streaming JSON parser (Claude SSH)
tsx	4.19.4	TS executor (dev + scripts)
uuid	11.1.0	UUIDs partout

Stockage / persistance

Composant	Rôle
SQLite (better-sqlite3)	Users, games, questions, feedbacks
Drizzle Kit	Migrations
Qdrant	Vector DB (1024 dims, dense + BM25 sparse) — une collection par jeu (rules_<slug>) + collections cartes par TCG (magic-cards, lorcana-cards, etc.)
Filesystem (/app/pdfs, /app/data)	PDFs uploadés, PNG rendus, caches JSON (OCR / contextual / conflicts), data sources cartes

ML / RAG

Composant	Rôle
TEI (Text Embeddings Inference)	Service HTTP local sur RTX 3060 — modèle bge-m3 (1024 dim dense + sparse BM25)
TEI Reranker	Service HTTP local — modèle bge-reranker-v2-m3 cross-encoder
Claude Code CLI (Anthropic)	Génération streamée via SSH vers VM oracle
Modèle Haiku	claude-haiku-4-5-20251001 — HyDE, decompose-query, conflit detect, classify intent
Modèle Opus	(par défaut) — réponses RAG, deckbuilding

TCG (sources de données)

TCG	Source
MTG	Scryfall bulk JSON (téléchargé) + traduction Haiku
Lorcana	LorcanaJSON.org (MIT, bulk JSON FR/EN)
FAB	Package npm <code>@flesh-and-blood/cards</code> (bundlé dans l'image Docker)
Riftbound	API Riot card-gallery (live JSON)
Terraforming Mars	HTML parsing local + cards.json
Ark Nova	Sprite sheets découpées + cards.json

Méta-game

Source	TCG
17Lands API	MTG (draft analytics)
MTGGoldfish (scrape)	MTG (constructed metagame)
MTGTop8 (scrape)	MTG (top 8 tournois)
Mobalytics (scrape)	Riftbound (tier list)
RiftboundStats (API)	Riftbound (tournois)
fabtcg.com (scrape)	FAB (tournois officiels LSS)
BoardGameGeek XML v2	Tous (forums Rules)

Infra

Composant	Rôle
Unraid	OS hôte des containers
Docker	Containerisation (image multi-stage)
Gitea (<code>gitea.thymon.fr</code>)	Git + CI Runners + registry
Nginx Proxy Manager (NPM)	Reverse proxy + Let's Encrypt sur <code>rules.thymon.fr</code>
VM SSH <code>oracle</code>	Exécute Claude Code CLI (compte dédié least-privilege)
RTX 3060	TEI bge-m3 + reranker (GPU)