

# 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)
```

Revision #1

Created 2026-05-10 15:20:29 UTC by thymon

Updated 2026-05-10 15:20:29 UTC by thymon