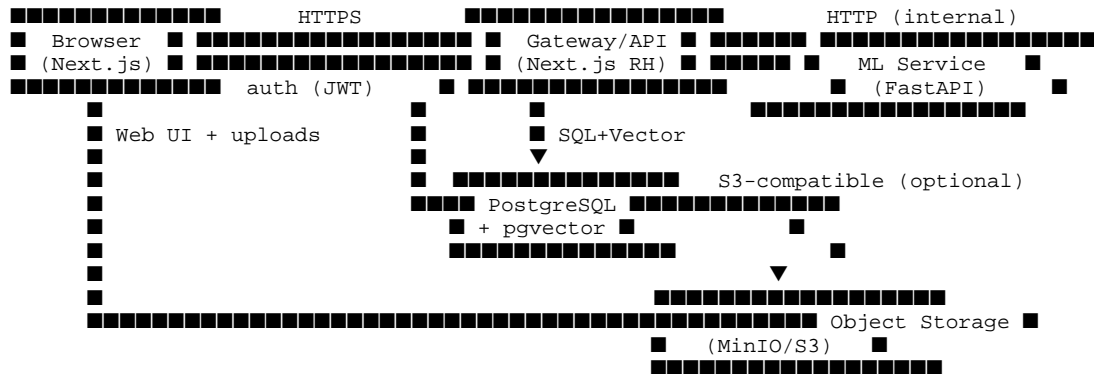


# MelodyMind — System Architecture (MVP)



## Components (MVP)

Frontend (Next.js, TypeScript): Pages (/login, /dashboard, /analyze/text, /analyze/audio, /matches). Duties: auth, uploads, charts, players.

Gateway/API (Next.js Route Handlers): Duties: JWT auth, validation, DB access, orchestrates ML calls.

ML Service (FastAPI, Python): Endpoints: /health, /analyze/text, /analyze/audio, /embed/text, /embed/audio.

PostgreSQL + pgvector: Stores users, items, metrics (JSONB), and embeddings.

Object Storage (MinIO/S3): Stores raw audio and plots (optional at MVP).

## Data Flow (Happy Path)

1. User logs in → Gateway issues JWT.
2. Upload Text → Gateway → DB row → ML analyze → save metrics+embedding.
3. Upload Audio → Gateway → Object store → ML analyze → save metrics+embedding.
4. Cross-modal matches → pgvector similarity → top-k results.
5. Frontend shows charts + playback + export.

## Minimal API Contract (Frontend ↔ Gateway)

```
POST /api/auth/login → {token}
POST /api/texts → {textId}
POST /api/audio → {audioId}
POST /api/analyze/text/:textId → {jobId}
POST /api/analyze/audio/:audioId → {jobId}
GET /api/jobs/:jobId → {status, progress}
GET /api/texts/:id /api/audio/:id → metrics JSON
POST /api/match/from-text → top-k audio
POST /api/match/from-audio → top-k texts
```

## Minimal ML Contract (Gateway ↔ ML)

```
POST /analyze/text {textId, text} → {metrics, embedding}
POST /analyze/audio {audioId, path} → {metrics, embedding}
POST /embed/text {text} → {embedding}
POST /embed/audio {path} → {embedding}
```

## Core Tables (Postgres)

users(id, email, hash, created\_at)  
texts(id, user\_id, title, language, raw, created\_at)  
audio(id, user\_id, title, url, duration\_s, created\_at)  
analyses(id, target\_type, target\_id, status, summary\_json, created\_at)  
embeddings(id, modality, dim, vector VECTOR(768), target\_type, target\_id, created\_at)  
\* Index: ivfflat/hnsw on vector for similarity.

## Similarity Query (pgvector)

```
SELECT target_id, 1 - (vector <=> $1) AS score  
FROM embeddings  
WHERE modality = 'audio'  
ORDER BY vector <=> $1  
LIMIT 10;
```

## Security & Ops (MVP)

JWT (httpOnly cookie), schema validation.  
File type/size checks; optional AV scan.  
Structured logs, request timing, health probes.  
Docker Compose for local, GitHub Actions later.

## Acceptance (Phase 1)

Architecture frozen; API contracts noted; tables listed.  
You can explain the end-to-end flow in 2 minutes.  
No coding yet — just clarity.