SABI - vaayaadi

🧠 High-Level ETL + RAG Architecture (Sabi)

Extract (Data Sources)

Data is collected from multiple sources using **connectors**:

- **PostgreSQL** → structured data (daily plans, schedules)
- **Gmail** → emails, attachments (PDFs, text)
- **OCR / Textractor** → text from images or documents
- (Optional later: web, files, notes)

Each source has:

- source_id
- raw content
- metadata

Transform (Processing & Chunking)

Before indexing, data is normalized:

- Clean text
- Split into **chunks**
- Each chunk gets:
  - source_id
  - chunk_id
  - chunk_text
  - metadata

This step ensures **consistent, searchable units**.

Embedding (txtai – vector only)

txtai is used **only for embeddings**, not orchestration.

- txtai.Embeddings.embed(text)
- Converts chunk_text → **vector**
- No indexing done by txtai

✓ Keeps txtai lightweight

✓ Full control over indexing

Load (Elasticsearch – hybrid index)

Each chunk is stored in **Elasticsearch** with:

source_id

chunk_id

text

metadata

vector

This enables:

- Keyword search
- Vector (semantic) search
- Hybrid retrieval

Index example:

healthai_vectors

🧩 Key Design Principles (Mentor-safe)

- No overengineering
- Clear separation of concerns
- txtai ≠ orchestrator
- Pipeline ≠ chatbot
- Production-ready flow

🧠 One-Line Summary (for meeting)

“We built a modular ETL pipeline that converts raw personal and operational data into embeddings, indexes them in Elasticsearch, and powers a real-time conversational AI through retrieval-augmented generation.”