**1) schemas/meta.py + utils/ids\_meta.py**

**Mục đích:** Đặt **chuẩn dữ liệu & ID ổn định** cho toàn pipeline.  
**Đặt ở đâu:** src/schemas/meta.py, src/utils/ids\_meta.py.  
**Làm gì:**

* Khai báo **schema META** (DocInfo, ChunkMeta) để mọi bước ghi cùng một cấu trúc.
* Hàm tạo **doc\_id/page\_id/chunk\_id** ổn định (deterministic).
* Hàm **slugify/sha1** và **meta\_from\_input\_path(...)** (rút metadata nền từ đường dẫn: corpus/company/year/doc\_type).  
  **Đầu vào/ra:** đường dẫn input → dict metadata nền; text → sha1; (doc,page,idx) → IDs.  
  **Khi nào chạy:** Được gọi bởi A1 (ghi META) và C2 (upsert).  
  **Vì sao cần:** Không có chuẩn/ID ổn định → không thể “reindex theo trang”, không thể trace nguồn.

**2) config/a1\_text\_only\_rules.yaml**

**Mục đích:** Tập luật **regex/rules** để A1 trích **anchor nội dung** (title/chapter/heading/table\_anchor) và **lọc rác** (watermark, quảng cáo).  
**Đặt ở đâu:** config/a1\_text\_only\_rules.yaml.  
**Làm gì:** Cung cấp pattern: TITLE\_PATTERNS, CHAPTER\_PATTERNS, TABLE\_ANCHOR\_PATTERNS, NOISE\_LINES, YEAR\_PATTERN…  
**Đầu vào/ra:** nhận **page\_text** → trả về trường trích xuất (chapter, table\_anchor, year\_hint…).  
**Khi nào chạy:** Trong A1 mỗi khi xử lý một trang.  
**Vì sao cần:** META giàu thông tin giúp **lọc theo metadata** và **citation đúng chỗ**. Không có rules → model dễ trả lời lạc đề.

**3) A1 (ghi \*\_pageN\_meta.json theo trang/chunk)**

**Mục đích:** Biến mỗi **trang** thành nhiều **chunk** có META chi tiết để downstream (C2/RAG) dùng.  
**Đặt ở đâu:** src/a1\_text\_only\_output.py (hoặc runner A1 bạn đang dùng).  
**Làm gì:**

* Tách trang → tách đoạn/thẻ bảng → **chunk** (800–1200 ký tự, overlap ~100).
* Dùng rules YAML (mục 2) để gắn **heading\_path/chapter/table\_anchor/year\_hint**.
* Ghi file …\_page{n}\_meta.json theo **schema (mục 1)**.  
  **Đầu vào/ra:** PDF/DOCX/IMG → \*\_pageN\_text.txt + \*\_pageN\_meta.json.  
  **Khi nào chạy:** Bước tiền xử lý trước khi index.  
  **Vì sao cần:** Đây là nơi tạo **hạt dữ liệu** chuẩn cho vector store. Không có META per-chunk → RAG không bắt đúng trang/mục.

**4) C2 (xóa-nạp theo trang vào Vector DB)**

**Mục đích:** Đưa chunk + META vào vector store **một cách an toàn**, hỗ trợ “in bổ sung” chỉ theo trang.  
**Đặt ở đâu:** src/c2\_clean\_mix\_pre\_vector.py.  
**Làm gì:**

* Đọc \*\_pageN\_meta.json.
* **Delete by page\_id** (xóa đúng trang cần cập nhật).
* **Upsert** từng chunk với **metadata filterable**: corpus/company/year/doc\_type/doc\_id/page/page\_id/chunk\_id/heading\_path/....  
  **Đầu vào/ra:** META per-page → vector store (Chroma/FAISS/Qdrant…).  
  **Khi nào chạy:** Mỗi khi có trang mới/cần cập nhật.  
  **Vì sao cần:** Bảo toàn index, cho phép **partial rebuild** (không chạy lại toàn bộ).

**5) Router YAML + intent\_router() (RAG chọn nguồn tự động)**

**Mục đích:** Tự suy diễn **phạm vi nguồn** (filter metadata) từ câu hỏi để **search đúng kho**.  
**Đặt ở đâu:** config/router\_rules.yaml + hàm intent\_router() trong src/rag\_chain.py (hoặc src/router.py).  
**Làm gì:**

* Map từ khóa → filter (ví dụ “thẩm quyền, MKT2, UIC, 2024” → {corpus: company, company: UIC, doc\_type: policy, year: 2024}).
* Hợp nhất với **UI filter** nếu người dùng chọn phạm vi thủ công.  
  **Đầu vào/ra:** query → filter dict.  
  **Khi nào chạy:** Trước khi gọi similarity\_search.  
  **Vì sao cần:** Không phải lúc nào user cũng chọn đúng thư mục; router giúp **precision cao** mà không đổi dữ liệu.

**6) rag\_chain.py (truy vấn & citation đúng chỗ)**

**Mục đích:** Lấy kết quả từ vector store theo **filter** và in ra **citation chuẩn** để người dùng truy vết.  
**Đặt ở đâu:** src/rag\_chain.py.  
**Làm gì:**

* Gọi similarity\_search(query, filter=...) (từ router/UI).
* Trả lời + **đính kèm nguồn**: doc\_id — Trang — (heading\_path) — [chunk\_id].
* (Tùy chọn) **Hybrid search** + **rerank nông** để tăng độ khớp.  
  **Đầu vào/ra:** query (+filter) → top-k docs (+metadata) → câu trả lời + citation.  
  **Khi nào chạy:** Runtime khi user hỏi.  
  **Vì sao cần:** Citation dựa trên META per-chunk giúp **tin cậy** và **bắt đúng đoạn**.

**7) manifest.json (Ops/kiểm soát phiên bản index)**

**Mục đích:** Đảm bảo **nhất quán** giữa **query embedder** và **index** (tránh lỗi model/dimension).  
**Đặt ở đâu:** Trong thư mục snapshot vector store.  
**Làm gì:**

* Ghi: collection, embedding\_model, embedding\_dim, normalize\_embeddings, danh sách meta\_keys, created\_at, build\_commit.
* Khi load index: **assert** đúng model & dimension.  
  **Đầu vào/ra:** metadata về index → dùng để **validate** khi mở.  
  **Khi nào chạy:** Lúc **build snapshot** và **load** vào app.  
  **Vì sao cần:** Ngăn “query bằng model A” trên “index model B” (ra kết quả sai/không khớp).

**Tóm tắt công dụng theo dòng chảy**

* **(1)** định nghĩa **khung & ID** →
* **(2)** đặt **luật trích xuất** →
* **(3)** A1 **tạo META per-page/chunk** giàu thông tin →
* **(4)** C2 **đưa vào DB** đúng chuẩn + **cập nhật theo trang** →
* **(5)** Router **sinh filter** từ câu hỏi →
* **(6)** RAG **search theo filter** & **trả citation** →
* **(7)** Manifest **giữ nhất quán** mô hình & snapshot khi vận hành.

Nếu bạn muốn, mình gộp 7 mục này thành **một checklist triển khai** trong repo (README.md + ví dụ file) để team copy-paste là chạy được.