RAG Document Ingestion & Retrieval System — One-Page Design Note

1. Purpose

Design a **trust-first RAG ingestion and retrieval system** for large, mixed-content enterprise PDFs (legal, technical, scanned, tabular). The system must scale, minimize hallucinations, remain auditable, and support downstream vector-based retrieval and graph-based reasoning.

---

2. Core Guarantees (Non-Negotiable)

- **No hallucinated content** enters retrieval systems.
- **Every extracted fact is traceable** to document → page → extraction method.
- **Original data is never overwritten**; derived data is additive.
- **Reprocessing is always possible** with improved parsers/models.

These guarantees define the architecture.

---

3. Data Lifecycle (High-Level)

1. PDF arrives as an opaque object
2. Document is probed and classified (no extraction)
3. Deterministic parsing is attempted first
4. ML/OCR is applied only when required
5. Extracted blocks are scored for quality and risk
6. Trusted blocks are promoted to retrieval systems
7. All stages remain auditable via lineage

---

4. Decision-Driven Architecture (Key Insight)

This system is **rule-driven, not model-driven**.

All intelligence lives in **decision points**, expressed as IF–THEN rules: - If text layer exists → deterministic parsing - If confidence < threshold → escalate to ML/OCR - If content expresses relationships → graph candidate - If content is semantic and trusted → vector candidate - If trust is low → store only, do not retrieve

---

5. Storage Roles (Clear Separation of Concerns)

Authoritative Store — SQL (Always On)

- Stores **every page and block**
- Captures extraction method, confidence, lineage
- Serves as audit log, replay source, and compliance layer

Retrieval Surface — Vector Database (Primary Consumption)

- Stores **only high-confidence semantic blocks**
- References SQL block IDs (not raw text)
- Optimized for fast user query answering

Reasoning Surface — Knowledge Graph (Selective)

- Stores structured relationships (procedures, dependencies)
- Built only when document structure justifies it
- Nodes and edges reference SQL block IDs

Rule: **No system may ingest data unless it originates from SQL.**

---

6. Pipeline Structure (Conceptual)

- **Ingestion Pipeline** → registers document
- **Probing Pipeline** → computes logical parameters
- **Parsing Pipeline** → deterministic first, ML second
- **Qualification Pipeline** → scoring and trust evaluation
- **Promotion Pipeline** → vector / graph eligibility

Each pipeline is simple; the workflow is intelligent.

---

7. Why This Design Works

- Scales to 1,000+ page documents
- Minimizes ML cost and risk
- Prevents hallucination by construction
- Fully auditable and replayable
- Aligns with investor, legal, and enterprise expectations

---

8. One-Line Mental Model

“SQL decides what exists, Vector decides what’s retrieved, and the Knowledge Graph decides what’s connected.”

End of design note.