RAG Document Ingestion & Retrieval — Implementation Approach Note

1. Objective

Translate the **design guarantees** into a concrete, buildable system by defining **components, responsibilities, and data flow** — without committing to specific tools or code.

This document answers **HOW the system is built**, not **WHAT libraries are used**.

---

2. Architectural Principle

The system is composed of **small, single-responsibility components** coordinated by a **workflow/ orchestration layer**.

- No component performs decision-making *and* heavy data processing
- Intelligence lives in **routing rules**, not in individual modules
- All derived data flows through a single authoritative store

---

3. Core Components (System Decomposition)

3.1 Document Registry

Responsibility - Register incoming documents - Assign immutable document identifiers - Track high-level document status

Knows - Source, storage location, metadata

Does NOT Know - Parsing logic - Retrieval systems

3.2 Probe Engine

Responsibility - Rapidly inspect document structure - Sample pages to detect layout complexity - Generate structural signals

Input - document_id

Output - probe_signals

Does NOT Know - Vector DB - Knowledge Graph - ML models

3.3 Logical Decision Engine (Control Plane)

Responsibility - Convert probe signals into logical parameters - Decide parsing strategy, risk tolerance, promotion eligibility

Input - probe_signals

Output - logical_parameters

Key Insight This is the *brain* of the system. No extraction happens here.

3.4 Parsing Engine (Data Plane)

Responsibility - Extract content deterministically wherever possible - Delegate to ML/OCR only when required

Input - document_id - logical_parameters

Output - extracted_blocks

Guarantees - No hallucinated content - Every block has lineage

---

3.5 Quality & Trust Evaluator

Responsibility - Assign confidence, risk, and trust levels - Enforce promotion thresholds

Input - extracted_blocks

Output - qualified_blocks

Does NOT Know - How content will be retrieved

3.6 Promotion Engine

Responsibility - Decide where qualified blocks are promoted - Enforce strict promotion rules

Promotion Rules - High-confidence semantic → Vector - Relational / procedural → Knowledge Graph - Low-confidence → SQL only

Critical Rule No promotion occurs without SQL persistence.

3.7 Retrieval Interfaces

Responsibility - Serve user queries - Combine vector retrieval and graph reasoning - Resolve answers using SQL-backed references

Guarantee - All responses are traceable to source documents

---

4. Control Plane vs Data Plane

Control Plane

- Probe Engine
- Logical Decision Engine
- Promotion Engine

Focus: decisions, rules, orchestration

Data Plane

- Parsing Engine
- OCR / ML sub-processes
- Embedding generation

Focus: computation, extraction, indexing

This separation enables scaling, auditing, and failure isolation.

---

5. Data Flow Summary

PDF → Registry → Probe → Decision Engine → Parsing → Quality Evaluation → SQL → Promotion → Vector / Knowledge Graph

At no point does raw content bypass SQL.

---

6. Why This Approach Is Industry-Grade

- Encourages modular development
- Allows tool replacement without redesign
- Supports replay and reprocessing
- Prevents hallucination by construction

- Easy to explain to PMs, mentors, and investors

---

7. Readiness for Next Phase

With this approach defined, the system is ready for: - Tool selection - Service boundaries - File / repository structure - Workflow engine implementation

---

End of implementation approach note.