PRD — Content Concierge AI

Hybrid RAG–Enabled MVP (Traditional RAG + GraphRAG)

---

1. Purpose of This PRD

This document defines the **full AI technical blueprint** for a Content Concierge MVP that actively leverages **Hybrid RAG**, combining: - Traditional vector-based RAG - GraphRAG powered by TigerGraph

This PRD is designed to be **directly codeable**, with explicit components, data flows, and agent responsibilities.

---

2. AI System Goals

- Deepen personalization beyond APIs
- Capture long-term thematic intelligence
- Improve insight relevance via relationship-aware retrieval
- Maintain explainability and compliance

---

3. High-Level Architecture

Pattern: Agentic Hybrid Intelligence System

- LangGraph → orchestration
- Vector Store → semantic retrieval
- TigerGraph → relational reasoning
- LLM → synthesis only

---

4. Knowledge Layers

4.1 Traditional RAG Layer

Ingested Content: - Bank-approved research - Educational explainers - Historical market commentary

Pipeline: - Chunk → embed → store - Metadata: asset class, risk type, horizon, compliance tags

---

4.2 GraphRAG Layer (TigerGraph)

Core Entities: - User - Asset - Sector - Strategy - Market Theme - Risk Factor

Edges: - OWNS - EXPOSED_TO - ALIGNS_WITH - IMPACTED_BY

Graph enables reasoning such as:

> "Users with X exposure and Y goal often care about Z themes"

---

5. LangGraph Agent Design

5.1 Agent State

```
{
  "user_context": {},
  "vector_context": [],
  "graph_context": [],
  "candidate_insights": [],
  "final_insights": []
}
```

5.2 Graph Nodes

1. **User Context Node**
2. Same as PoC 1
3. **Graph Context Expansion Node**
4. Query TigerGraph for related themes, risks, strategies
5. **Vector Retrieval Node**
6. Fetch semantically relevant documents
7. **Insight Planning Node (LLM)**
8. Merge graph + vector signals into insight candidates
9. **Synthesis Node (LLM)**

10. Generate text insights
11. **Citation Resolution Node**
12. Attach doc + graph provenance
13. **Ranking Node**
14. Select 2–3 best insights

---

6. Citation Model

Each insight must include: - Source document(s) - Graph path explanation (optional, internal)

Example:

> Insight derived from [Doc A] + exposure relationship (User → ETF → Sector → Theme)

---

7. Guardrails & Explainability

- LLM cannot traverse graph directly
- All graph queries are deterministic
- Insight text references only retrieved facts

---

8. Observability

- Graph queries
- Retrieved documents
- Prompt inputs/outputs
- Insight lineage

---

9. Tradeoffs & Risks

- Higher latency
- Increased infra complexity
- Stronger long-term personalization

---

10. Success Criteria

- Higher insight relevance vs API-only baseline
- Reduced repetition over time
- Clear provenance for every insight