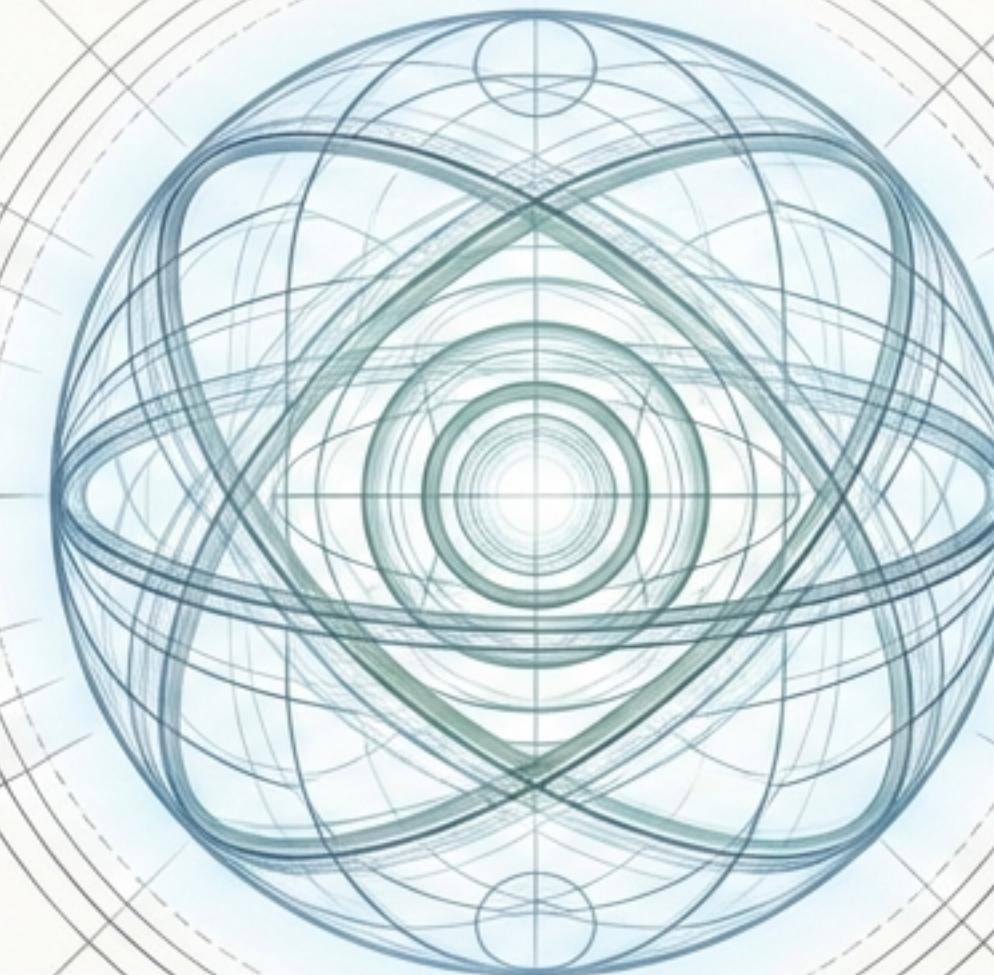


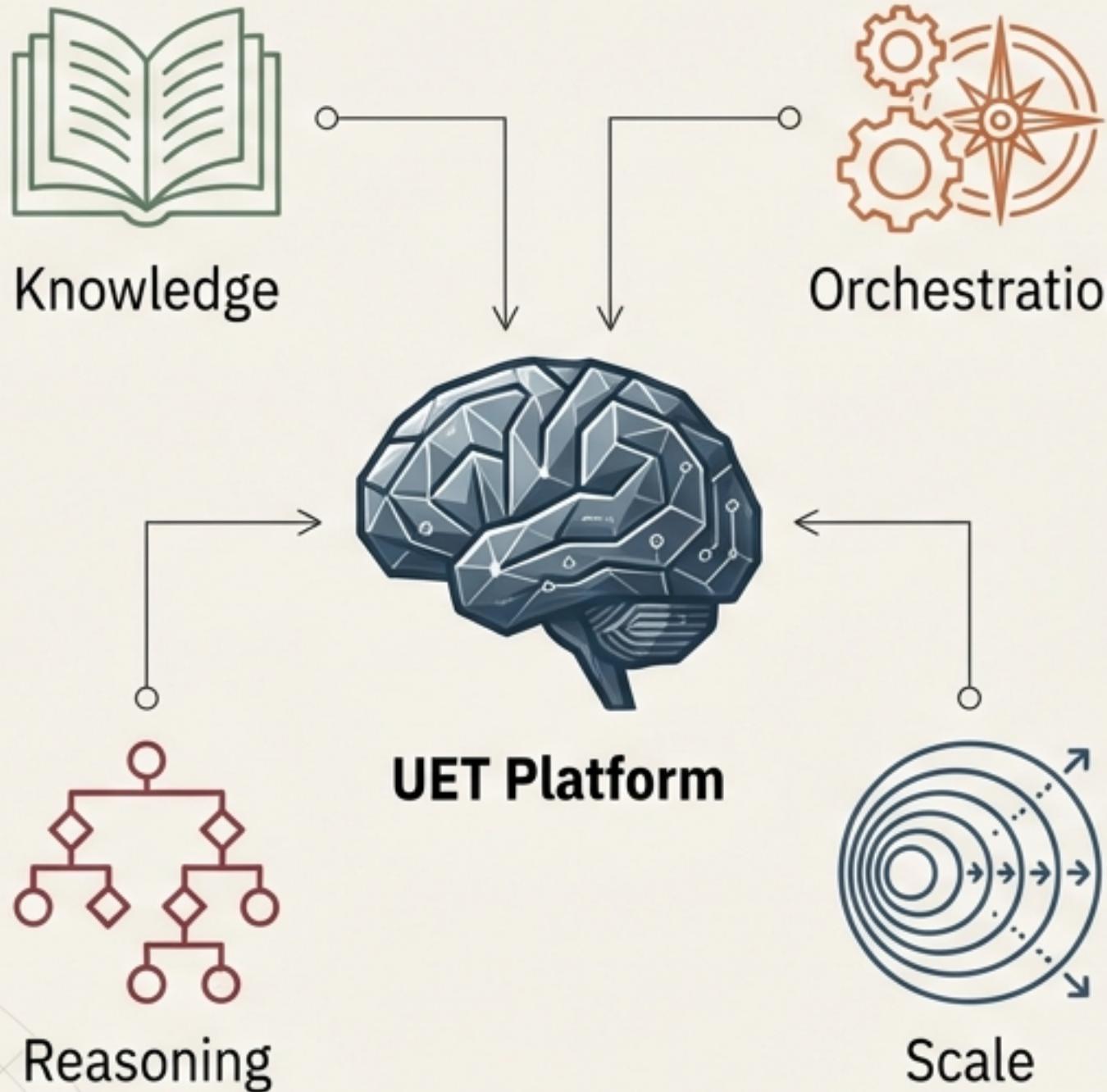
UET Platform Architecture v3.0

Master Blueprint – ภาพรวมระดับจักรวาลของแพลตฟอร์ม UET AI



ระบบทั้งหมดของ UET ถูกออกแบบให้ deterministic, สามารถอัปเดตความรู้แบบกันที,
และทำงานร่วมกันผ่านสัญญาณ (System Contract) ที่บังคับทุกโมดูลให้เชื่อมกันแบบ 100%

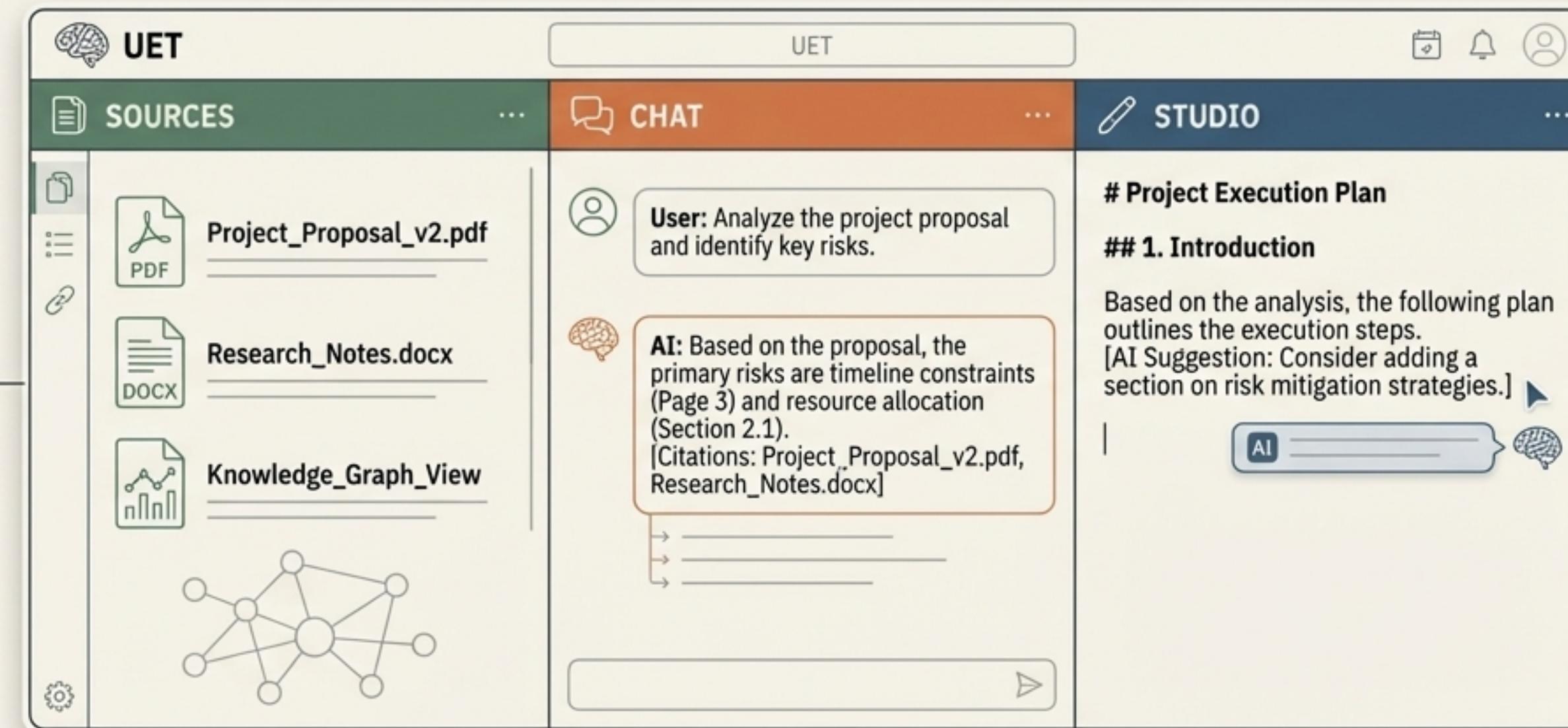
Purpose: เหตุผลที่ระบบนี้ต้องมีอยู่



- **Knowledge Operating System:** เป็นตัวกลางจัดการทุกความรู้ของผู้ใช้
- **Precision-RAG + Unified Knowledge Graph:** ผ่าน text → chunk → graph → agent → answer แบบ deterministic
- **Agent Orchestration System:** ไม่ใช่ “LLM chat” แต่เป็นระบบที่ควบคุมทุก step ของ AI
- **Execution-Graph AI:** pipeline reasoning ที่มี state และเกติกาชัดเจน ไม่ใช่ AI ที่คิดเมื่อ
- **Scalable by Design:** สร้างขึ้นเพื่อรับผู้ใช้จำนวนมากและโหลดงานหนัก

👉 พูดง่าย ๆ: นี่คือระบบที่ “อ่าน → เข้าใจ → สร้างเหตุผล → ตอบ” แบบมีสมองจริง

The User's Universe: A 3-Panel Knowledge Workspace



Sources Panel (คลังวัตถุดิบ):
ที่รวมไฟล์, เอกสาร, และกราฟ
ความรู้ทั้งหมดของโปรเจกต์

Chat Panel (สมอง AI):
ช่องสนทนากับ AI เพื่อวิเคราะห์,
สรุป, และเชื่อมโยงข้อมูลจาก Sources

Studio Panel (ห้องตัดต่อความรู้):
พื้นที่สำหรับเขียน, สร้างสรรค์, และ
สังเคราะห์ผลงานใหม่โดยใช้ AI ช่วย

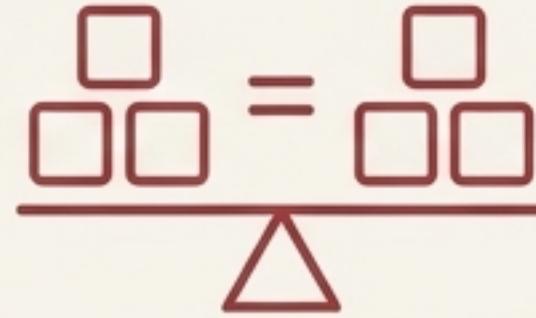
The 8 Laws of the Universe: หลักการแกนที่ห้ามละเมิด



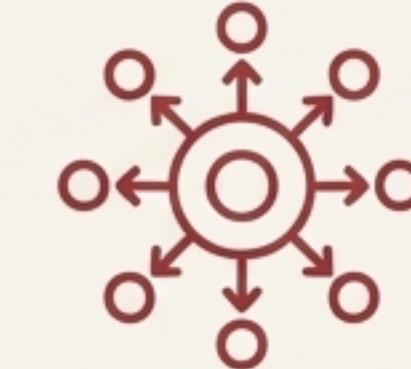
Zero-Stale:
ข้อมูลเก่าต้องไม่ถูกใช้ในการ
ตอบสนองเดี๋ยวนี้



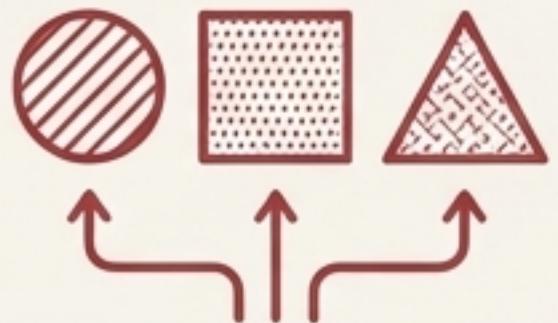
Versioned Everything:
ทุกอย่างในระบบต้องมีเวอร์ชัน
กำกับและตรวจสอบย้อนหลังได้



Deterministic Reasoning:
Input เมื่อันกันต้องให้ผลลัพธ์
เหมือนกันเสมอ



Event-Driven Architecture:
ทุกส่วนของระบบเชื่อมต่อและ
อัปเดตสถานะผ่าน Event Bus



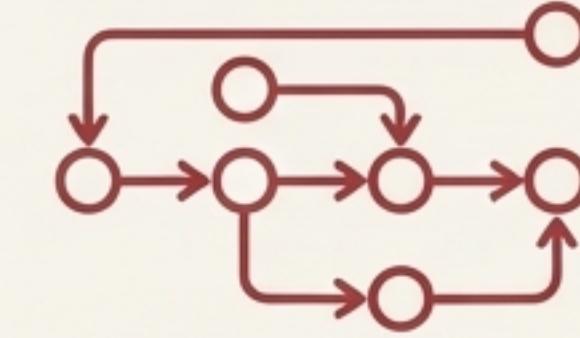
Separation of Concerns:
Agent ≠ RAG ≠ Sync; แต่ละ
Engine มีหน้าที่ชัดเจน ไม่ปนกัน



Consistency First: ความ
ถูกต้องของข้อมูลและความสอด
คล้องของระบบมาก่อนความเร็ว

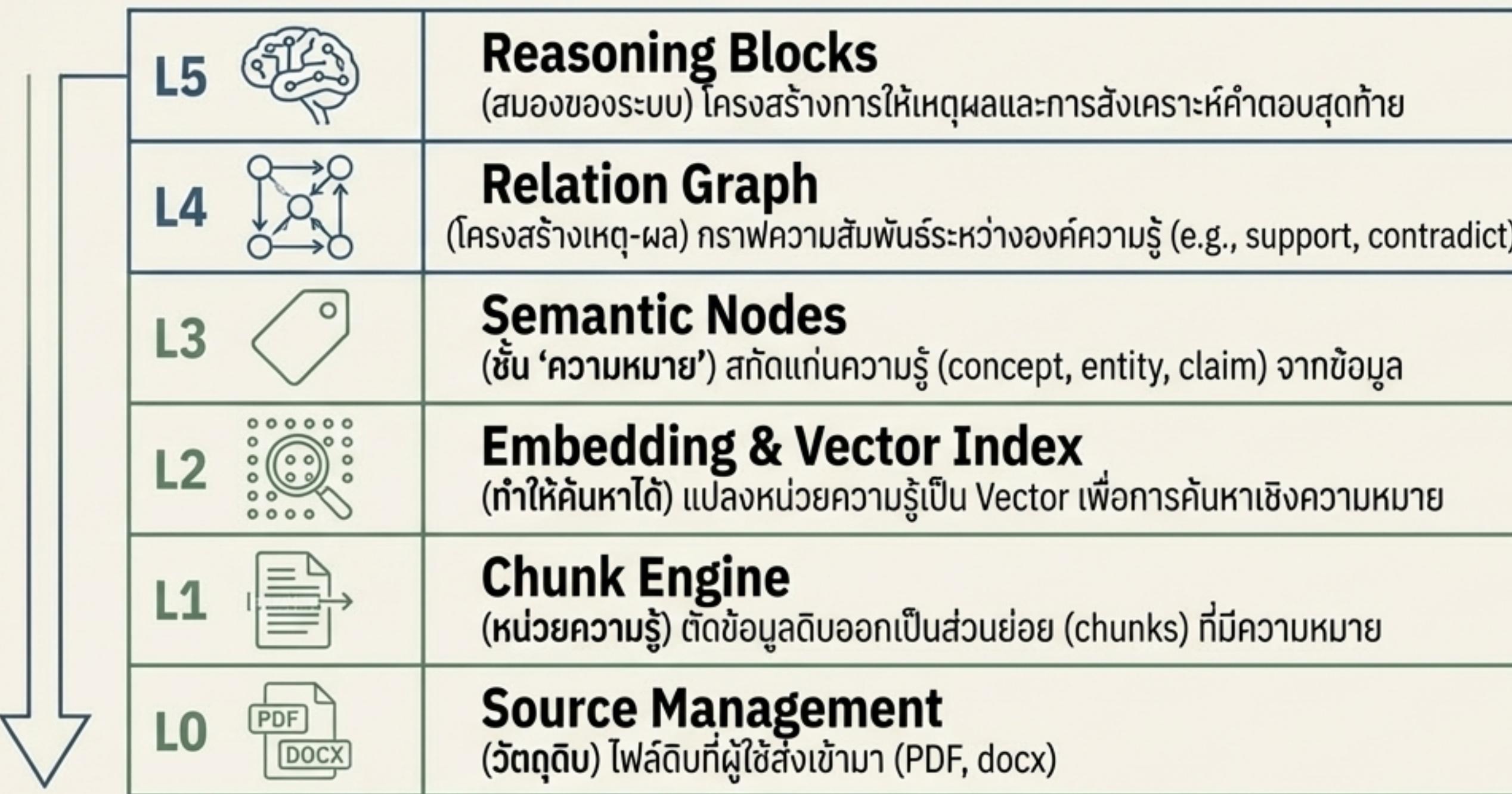


Permission Safety: ทุกการ
กระทำต้องผ่านการตรวจสอบ
สิทธิ์ตาม Permission Matrix



Explainability: ทุกขั้นตอน
การทำงานของ AI ต้องสามารถ
ติดตามและอธิบายย้อนกลับได้

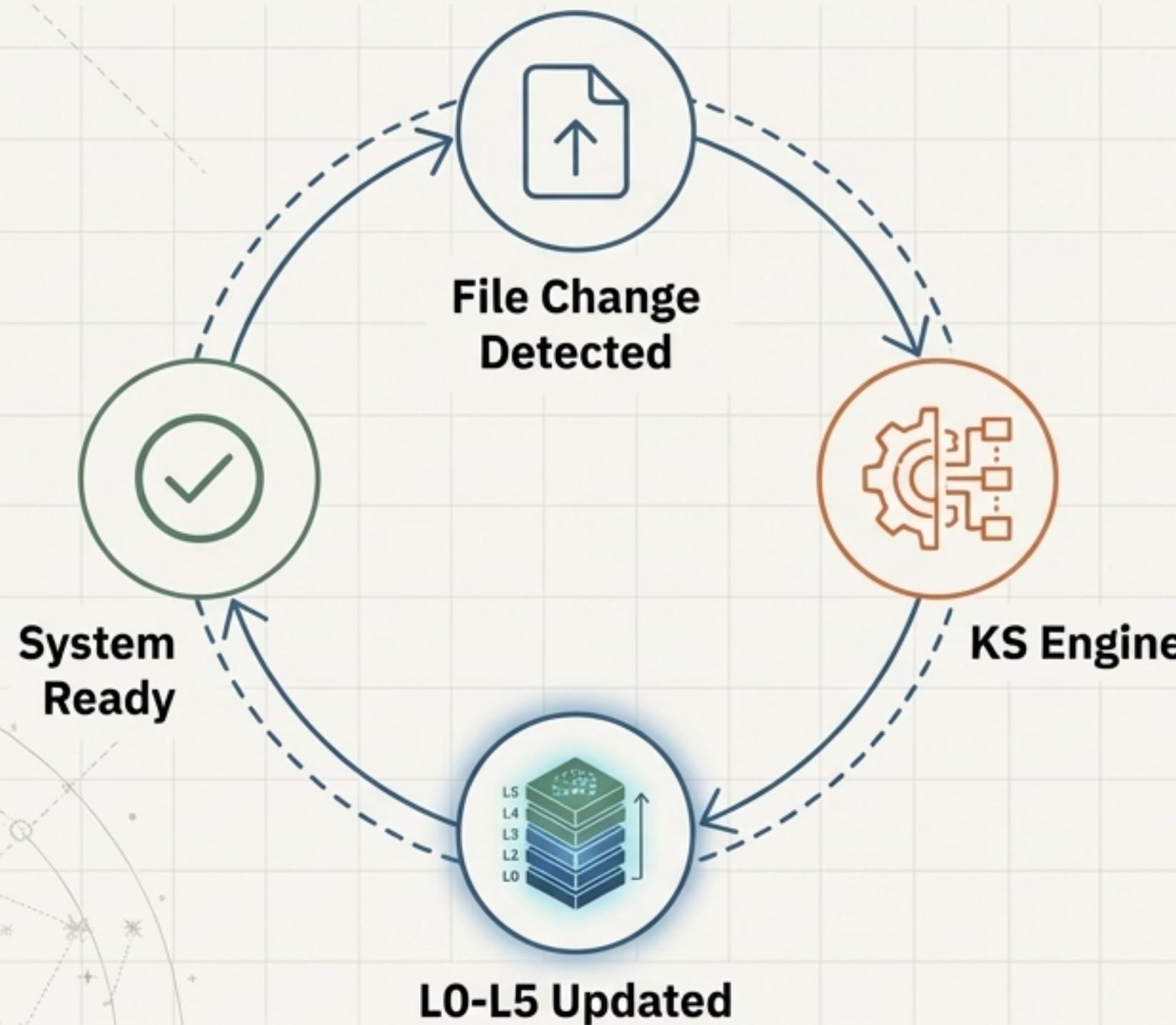
The AI Stack: โครงสร้างจักรวาลของระบบ (The 6-Layer Model)



การไหลของข้อมูลและ AI reasoning จะวิ่งขึ้น-ลงผ่าน 6 ชั้นนี้อย่างเป็นระบบ

The Heart of the System: UET Knowledge Loop

ทุกครั้งที่มีไฟล์เปลี่ยน → ทั้งระบบอัปเดตความรู้แบบ deterministic ในไม่กี่วินาที

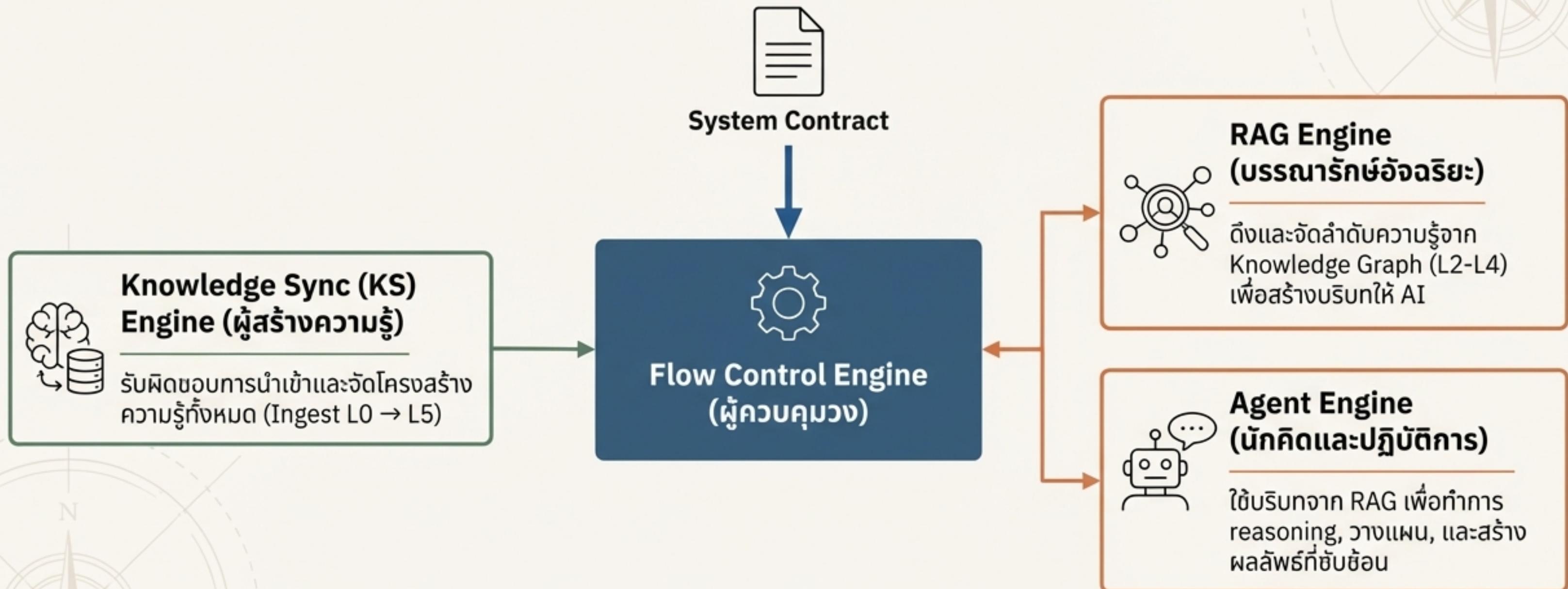


- **File Change Detected:** ผู้ใช้อัปโหลดหรือแก้ไขไฟล์
- **Knowledge Sync Triggered:** KS Engine เริ่บกระบวนการ ingest, clean, chunk, embed, และ update graph
- **L0-L5 Updated:** ทุกชั้นของ AI Stack ถูกปรับปรุงให้เป็นเวอร์ชันล่าสุด
- **System Ready:** RAG และ Agent Engines พร้อมใช้งานข้อมูลใหม่ทันที



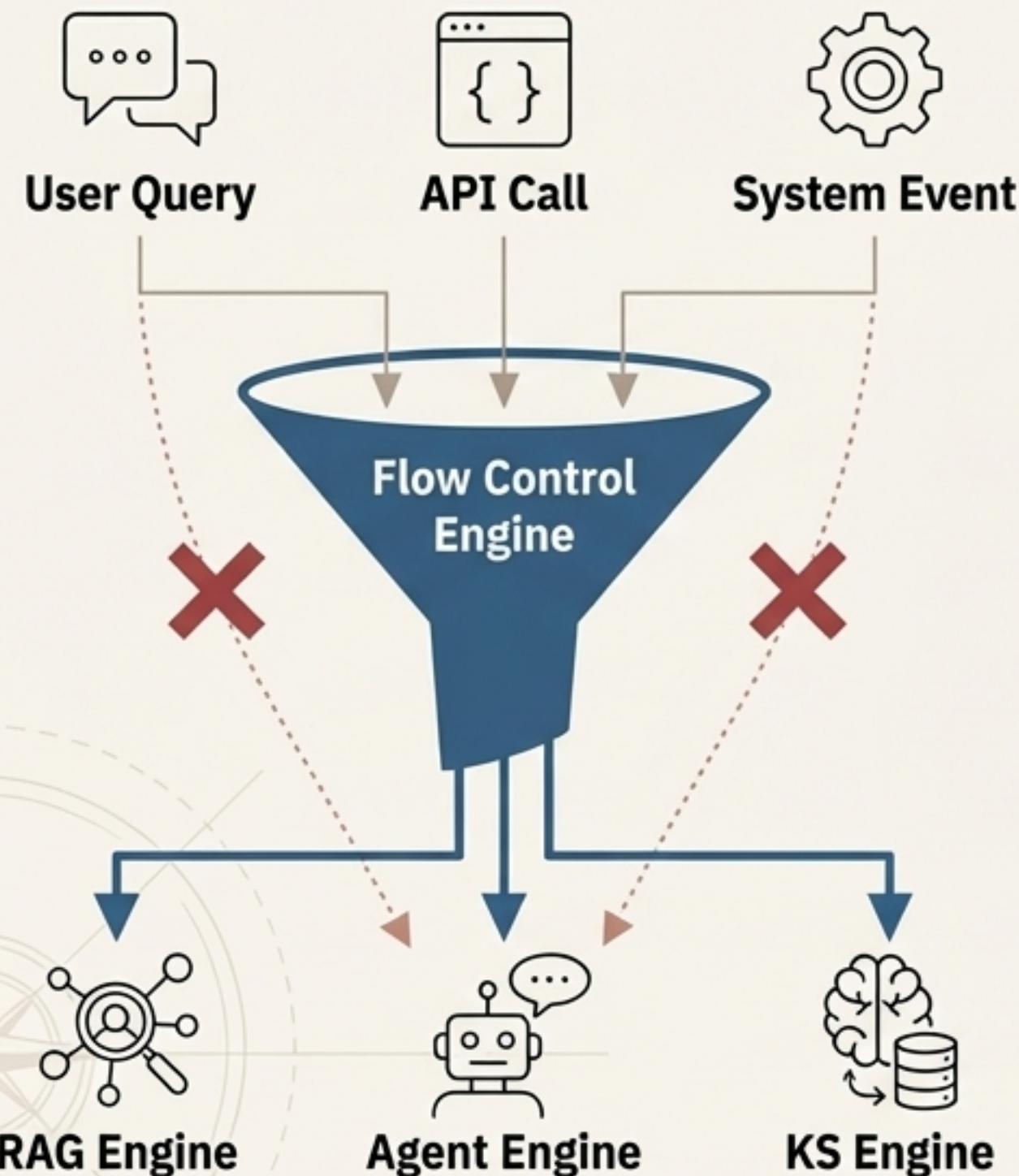
Zero-Stale
Principle

The Engines of UET: ผู้ขับเคลื่อนจักรวาลแห่งความรู้



**Separation of Concerns
Principle**

The Conductor: Flow Control Engine – ประดู่ใหญ่ของทุก Request



- **Classify Request:** ตีความทุกคำสั่งที่เข้ามา (simple, knowledge, reasoning, tool)
- **Orchestrate Execution:** จัดการลำดับการทำงานของ Engine ต่างๆ และควบคุม multi-step execution
- **Enforce Contracts:** ตรวจสอบ Permission, Safety, และ Version Integrity ก่อนอนุญาตให้ทำงาน
- **Handle Errors & Fallback:** เป็นศูนย์กลางการจัดการความผิดพลาดและกำหนดเส้นทางสำรอง

Flow Engine เป็น "ตัวเดียว"
ที่เริ่มต้นทุก request ในระบบ



Deterministic Reasoning
Principle

The Rules of Engagement: Engine Interaction Matrix

Caller ↓ / Callee →	Flow	Safety	Routing	RAG	Agent	...	Data Layer
Flow Control	—	✓	✓	✓	✓	...	✗
Agent Engine	✗	✗	✓	✓	—	...	✓
RAG	✗	✗	✗	—	✓	...	✓
...							

Key Insights

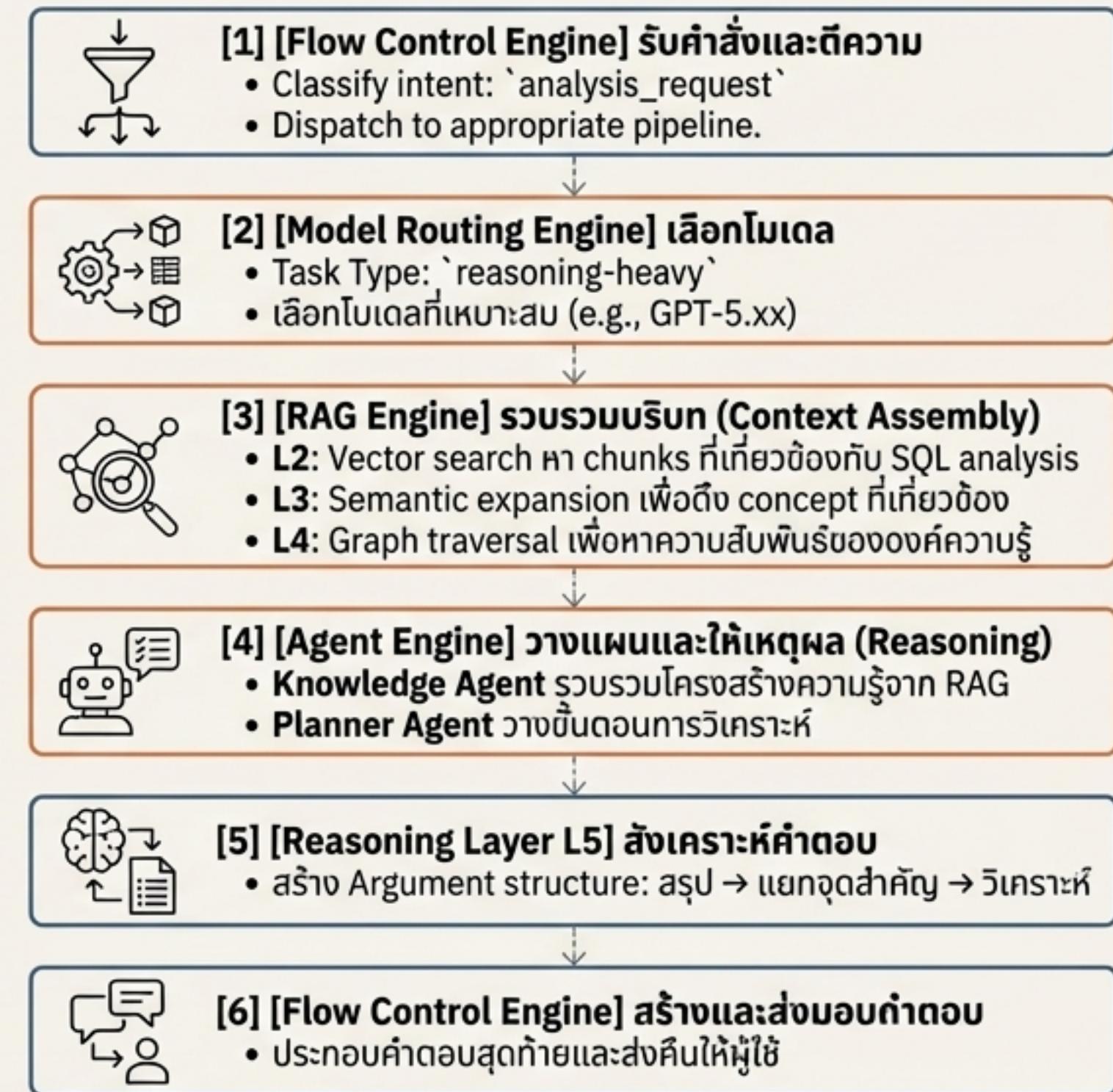
- Flow is the Master:** Flow Control สามารถเรียกทุก Engine ได้ (ยกเว้น Data Layer โดยตรง)
- Strict Boundaries:** Agent ห้ามเข้าถึง Database โดยตรง ต้องผ่าน RAG/KS เท่านั้น
- Clear Roles:** RAG เป็นเพียงผู้สร้าง บริบท (context creator) และไม่มีสิทธิ์ ตัดสินใจแทน Flow
- Passive Data:** Data Layer ไม่เคยเป็น ผู้เริ่มต้นการกระทำใดๆ

⚙️ Separation of Concerns Principle

⌚ Consistency First Principle

End-to-End Flow: Tracing a Single Query

“ช่วยวิเคราะห์ไฟล์ SQL ให้หน่อย”



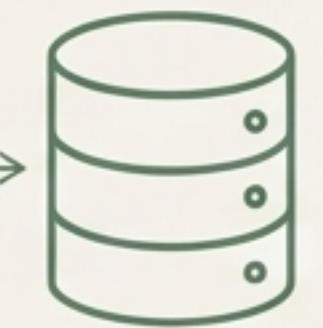
ทุกขั้นตอนเกิดขึ้นตาม Blueprint อย่างเป็นระเบียบและตรวจสอบย้อนกลับได้ (Explainability)



Deterministic Reasoning Principle
Separation of Concerns Principle

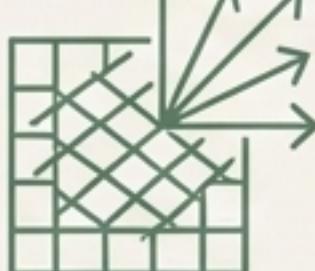
The Foundation: Data & Infrastructure Architecture

Data Architecture (Unified Knowledge Graph)



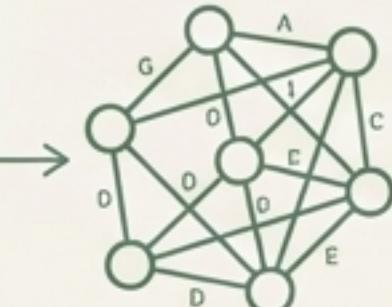
Relational DB (PostgreSQL)

เก็บ metadata (files, chunks, nodes, edges), agent state, logs



Vector Index Store (Qdrant)

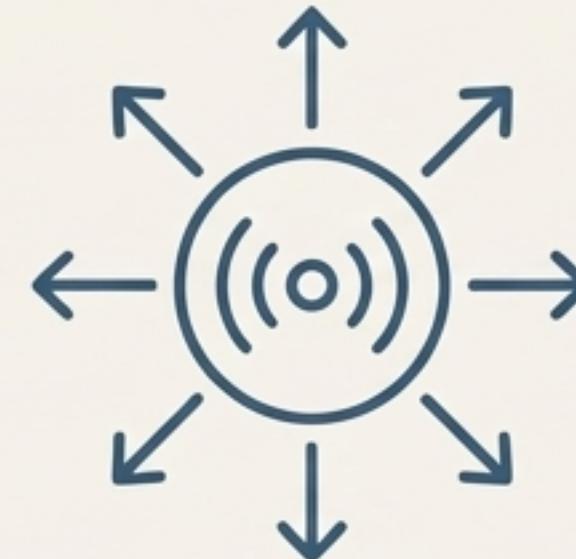
เก็บ embedding vectors เพื่อการค้นหาความหมายที่รวดเร็ว



Graph Structure

โครงสร้างเชิงตรรกะของ semantic nodes และ relations

Communication Backbone



Event Bus System

ระบบสื่อสารกลางแบบ publish/subscribe สำหรับแจ้งเตือนและยิงค์สถาบันทั่วทั้งระบบ (e.g., Redis)

Support Infrastructure



Cache System

เพิ่มความเร็วและลดภาระงานด้วย multi-layer caching strategy



Worker/Task System

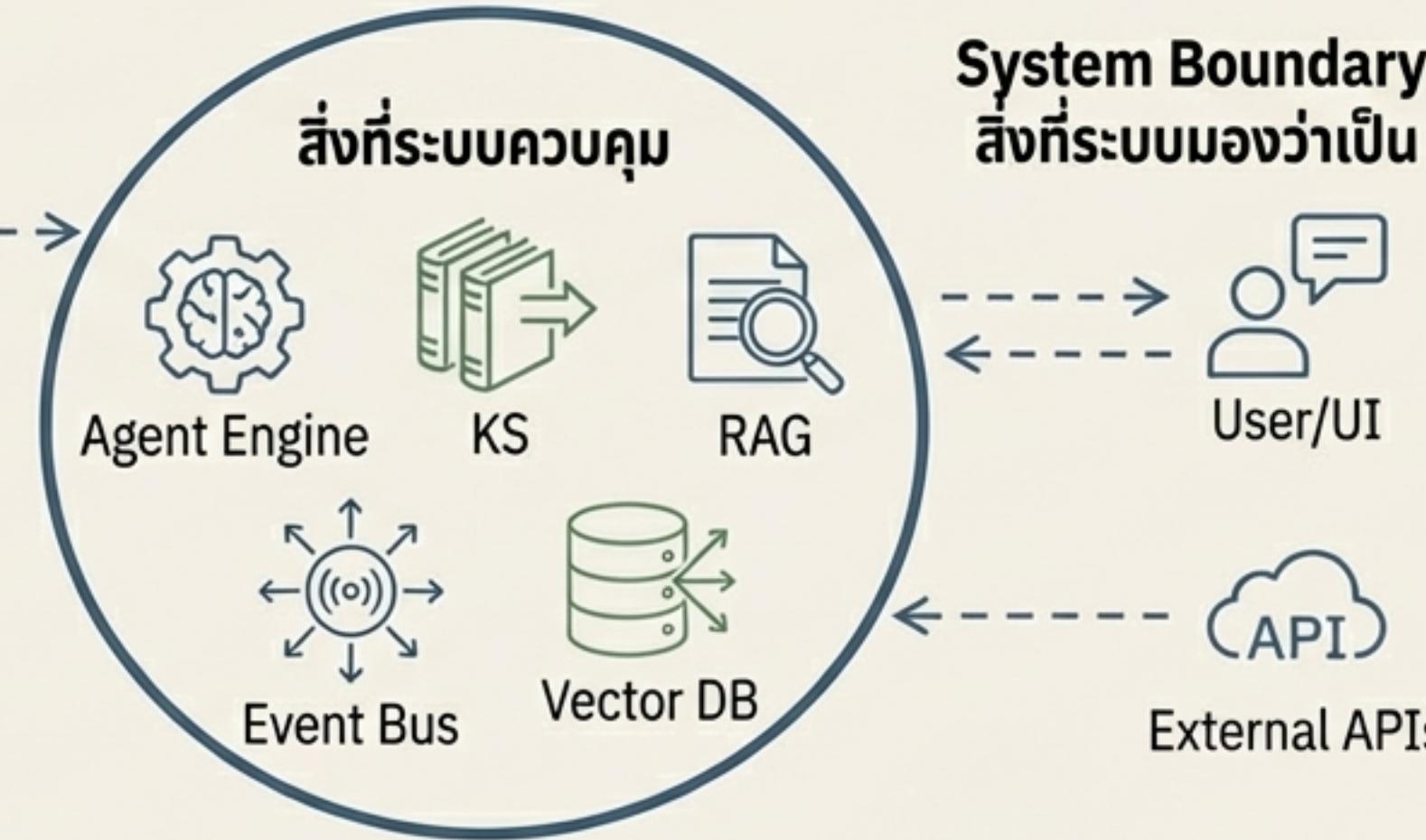
จัดการงานเบื้องหลังที่ใช้เวลาบาน (e.g., embedding, re-indexing)

System Integrity: Boundaries, Failures, and Safeguards

System Boundary Diagram



LLM Providers



System Boundary Diagram สิ่งที่ระบบมองว่าเป็น External



User/UI



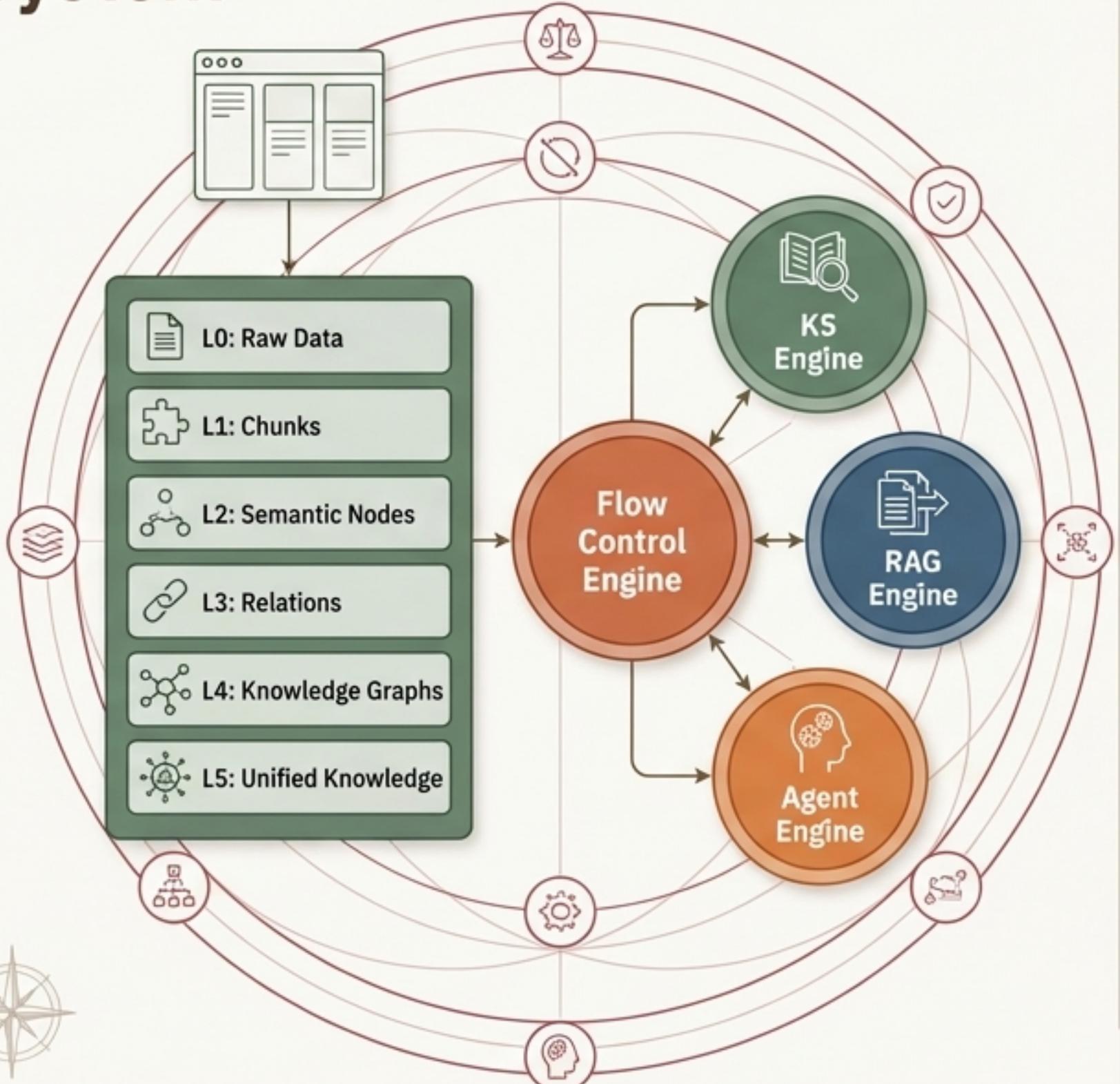
External APIs

Failure Mode Overview

Failure Mode	ลักษณะปัญหา	วิธีแก้ไขในสถาปัตยกรรม
Stale Knowledge Error	RAG ใช้ KB เวอร์ชันเก่า	'Zero-Stale' principle + Cache invalidation ผ่าน Event Bus
Routing Error	เลือกไม่เดลกที่ประสิทธิภาพไม่พอ	Routing fallback logic + Cost/performance monitoring
Agent Loop Error	Agent reasoning วนซ้ำไม่รู้จบ	Step limit + Loop detection ใน Flow Control Engine
Sync Mismatch Error	Chunk ใน DB ไม่ตรงกับ Vector	KS rebuild process + Checksum validation
Permission Denied	Agent พยายามทำงานนอกขอบเขตสิทธิ์	Flow Control block + Permission Matrix enforcement
Event Bus Delay/Miss	Event หล่าไปครบท ทำให้สถานะไม่ตรงกับ	Retry mechanism + Ordering guarantee



The Blueprint Realized: A Deterministic Knowledge Operating System



Architectural Strengths

- **Deterministic & Explainable:** ทุกการทำงาน ถูกควบคุมและตรวจสอบได้
- **Zero-Stale & Event-Driven:** ความรู้สดใหม่อยู่เสมอผ่าน Knowledge Loop ที่ขับเคลื่อนด้วย Event
- **Robust & Resilient:** ออกแบบมารองรับ Failure modes และการขยายตัวในอนาคต
- **Unified & Consistent:** ทุกส่วนประกอบทำงานภายใต้กฎเดียวกันของ System Contract

Blueprint นี้คือ **หัวใจของสถาปัตยกรรมทั้งหมด** – เป็นแกนกลางที่ทำให้เอกสาร, API, และทุกโมดูลของ UET Platform สอดคล้องกัน 100% และเป็นรากรฐานที่มั่นคงสำหรับการพัฒนาระบบท่อไป

