

The Narrative Arc

เส้นทางการสำรวจสถาปัตยกรรม



1
1

1. The Vision & Blueprint

เริ่มจากภาพใหญ่ที่สุด ทำความเข้าใจวิสัยทัศน์และโครงสร้างแม่บทของระบบ



2
2

2. The Core Components

เจาะลึกองค์ประกอบหลักและกลไกการทำงานภายใน



3
3

3. The Industry Benchmark

เปรียบเทียบแต่ละส่วนกับแนวปฏิบัติและเครื่องมือที่เป็นมาตรฐานในปัจจุบัน



4
4

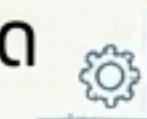
4. The Guiding Philosophy

สกัดแกร่นปรัชญาที่ขับเคลื่อนทุกการตัดสินใจในการออกแบบ

Beyond the Chatbot: A Knowledge Operating System

ไม่ใช่แค่ ‘AI Chat’ แต่คือ
Knowledge Operating System

ที่กำหนดที่เป็นตัวกลางจัดการองค์ความรู้ทั้งหมด



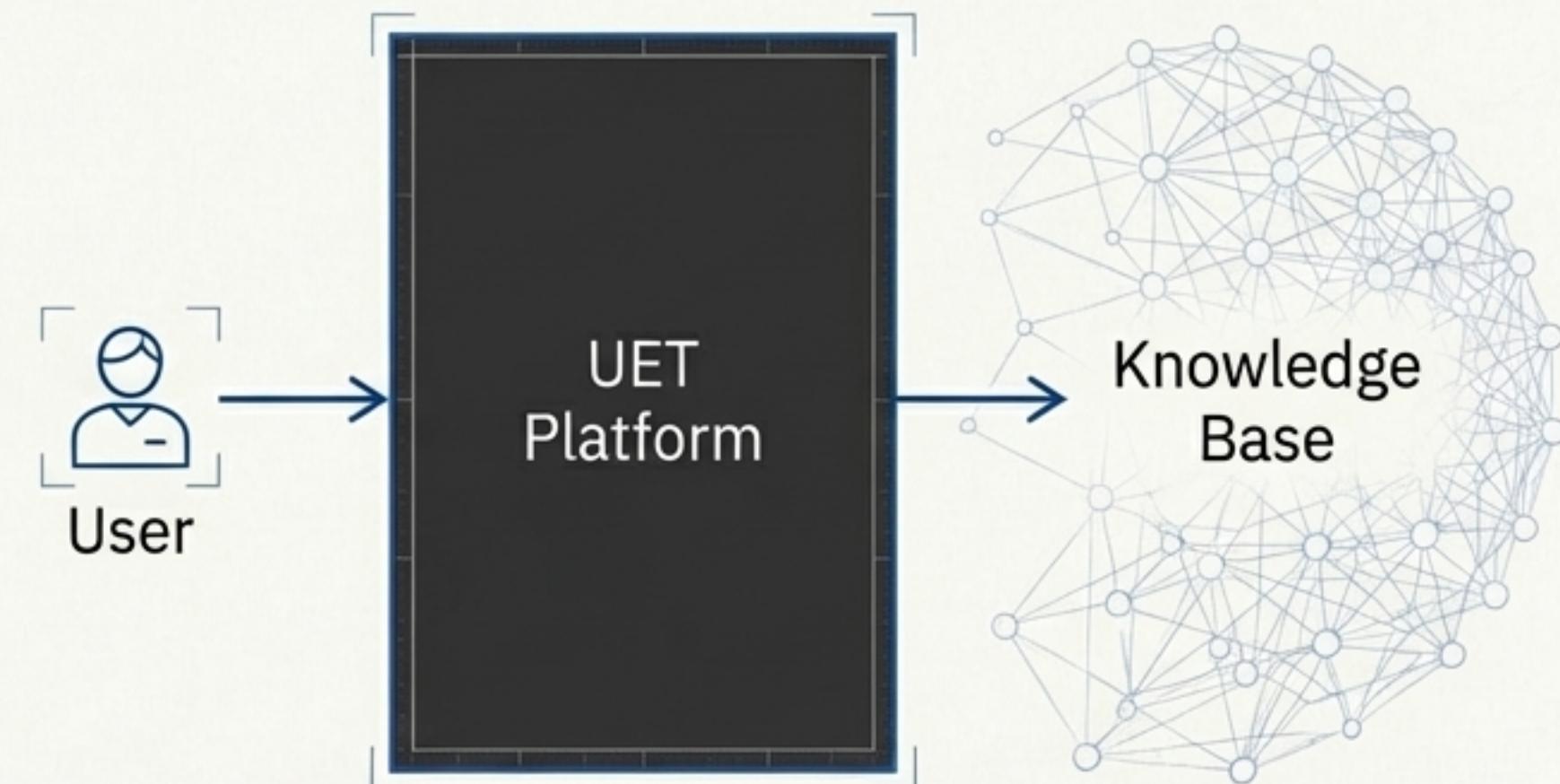
ไม่ใช่แค่ ‘RAG’ แต่คือ
Precision-RAG + Unified Knowledge Graph

ที่ผสานการทำงานตั้งแต่ Text, Chunk,
Embed, Graph ไปจนถึง Agent
อย่างเป็นระบบ (Deterministic)

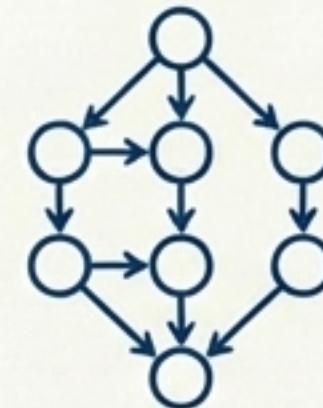


ไม่ใช่แค่ ‘Agent’ แต่คือ
Agent Orchestration System

ที่ควบคุมทุกขั้นตอนการทำงานแบบ
Execution Graph



Core Architectural Patterns: รากฐานการออกแบบระบบ



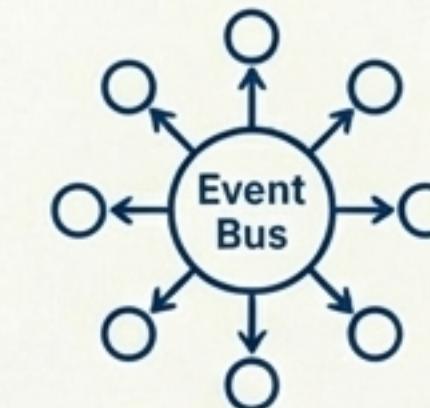
DAG-based Execution

การทำงานเป็นลำดับขั้นที่โปร่งใส ตรวจสอบย้อนกลับได้ ทำให้ทุกการตัดสินใจของ AI มีที่มาที่ไป



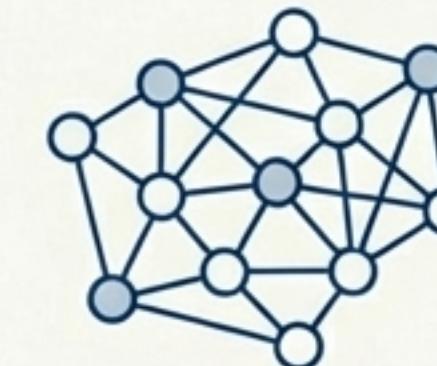
Modular Separation (Room & Responsibility)

การแบ่งขอบเขตความรับผิดชอบของแต่ละโมดูลอย่างชัดเจน 未必 องแต่ละห้องมีหน้าที่ของตัวเอง



Event-Driven Communication

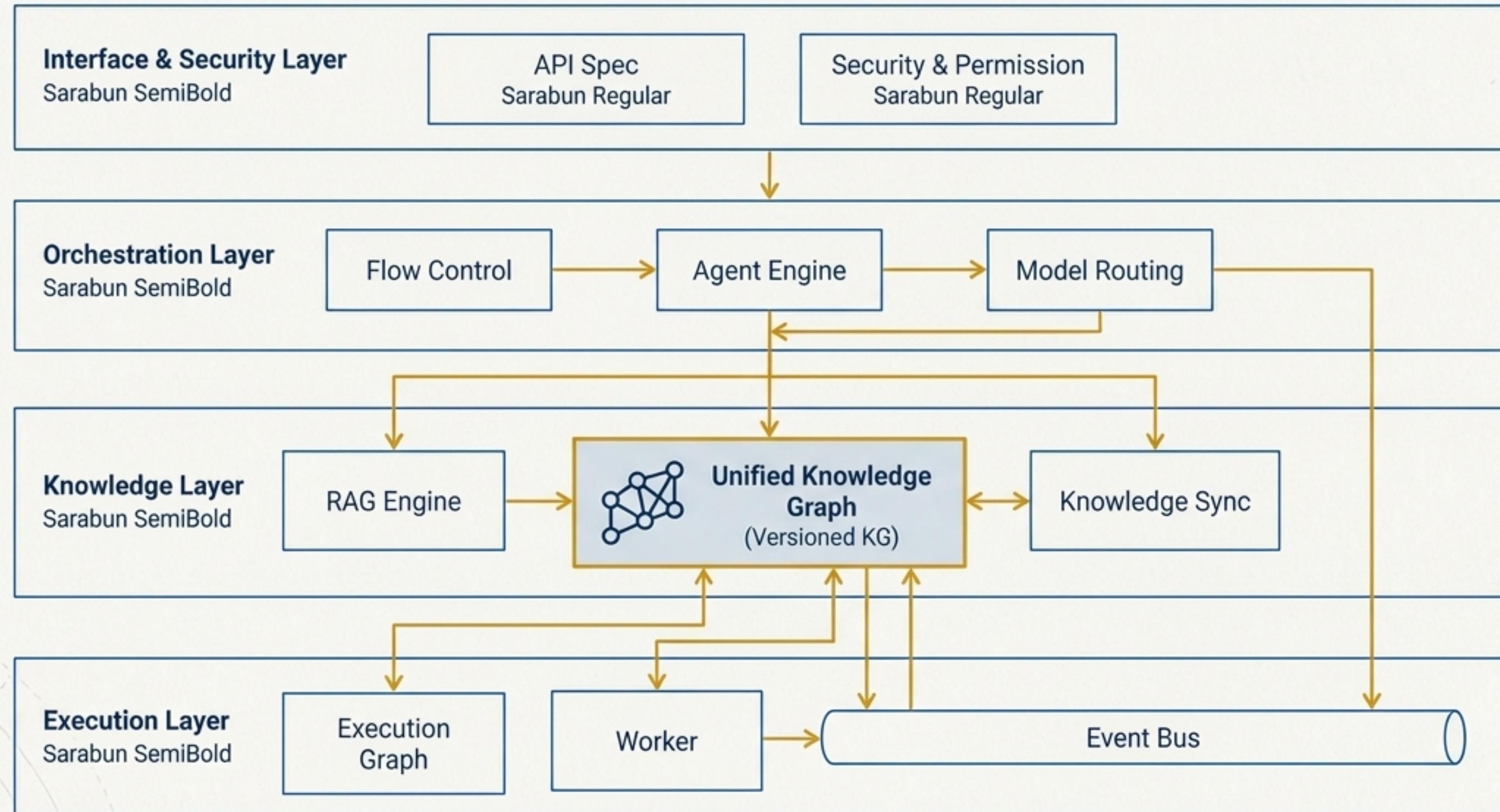
การสื่อสารระหว่างโมดูลผ่าน Event Bus ทำให้แต่ละส่วนทำงานแยกจากกันได้อย่างอิสระ ลดการพึ่งพิง (Decoupling)



Graph-based Knowledge

มององค์ความรู้ทั้งหมดเป็นโครงข่ายความสัมพันธ์ (Graph) ไม่ใช่แค่ข้อมูลที่เก็บแยกส่วน

The UET Module Map



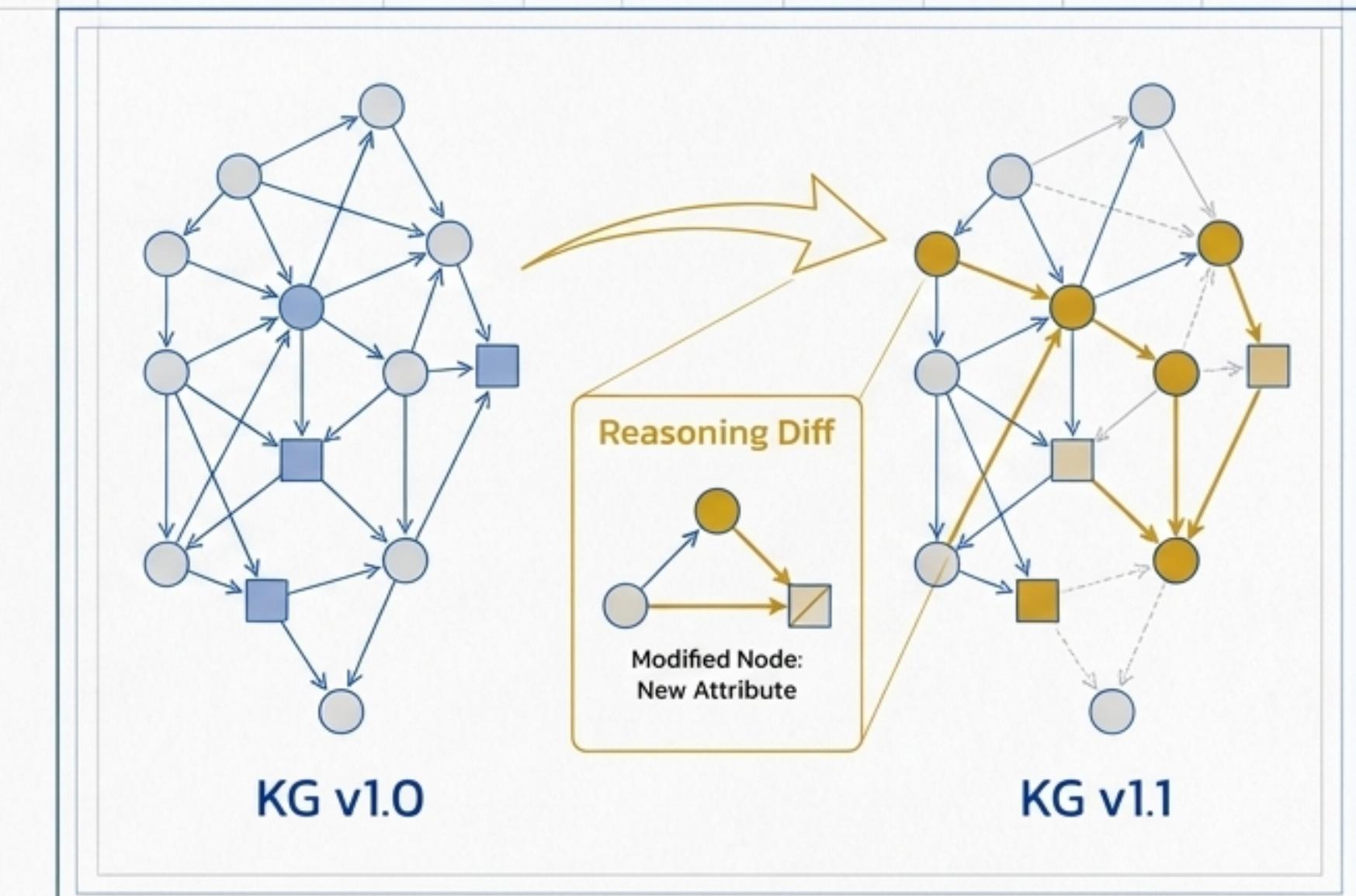
The Knowledge Core: Where UET Redefines RAG

Impact-based Knowledge Management

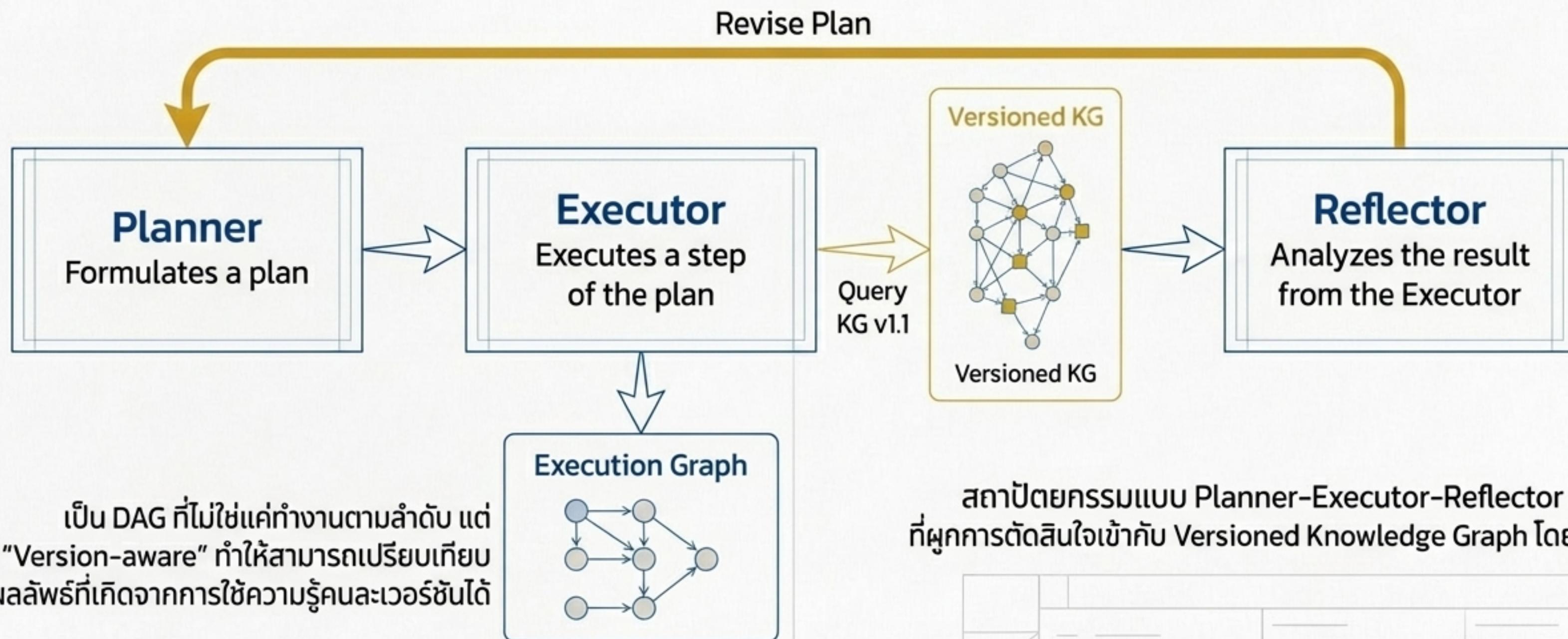
ระบบไม่ได้แค่ดึงข้อมูลที่ 'เกี่ยวข้อง' (Relevant) แต่ประเมินและจัดการความรู้โดยอิงจาก 'ผลกระทบ' (Impact) ขององค์ความรู้นั้นๆ

Versioned Knowledge Graph

ความรู้ไม่ใช่สิ่งคงที่ แต่มี 'เวอร์ชัน' และสามารถ 'เปรียบเทียบความแตกต่างเชิงเหตุผล' (Reasoning Diff) ระหว่างเวอร์ชันได้



The Orchestration Brain: How UET Reasons and Acts



Industry Benchmark: RAG & Agent Engines

RAG Engine

UET RAG Engine	The World
เกี่ยบกับ: LangChain, LlamaIndex, Haystack, ColBERT	เกี่ยบกับ: LangChain, LlamaIndex, Haystack, ColBERT
ใช้สถาปัตยกรรมแบบ Pipeline และ DAG-based Retrieval	ใช้สถาปัตยกรรมแบบ Pipeline และ DAG-based Retrieval
มี Impact-based Knowledge Layer ที่ Framework ทั่วไปไม่มี ทำให้การดึงข้อมูลคำนึงถึงผังระบบ ไม่ใช่แค่ความคล้ายคลึง (Similarity)	

Agent Engine

UET Agent Engine	The World
เกี่ยบกับ: LangGraph, ReAct Framework, CrewAI	เกี่ยบกับ: LangGraph, ReAct Framework, CrewAI
ใช้แนวคิด Planner-Executor หรือการทำงานร่วมกันของหลาย Agent	ใช้แนวคิด Planner-Executor หรือการทำงานร่วมกันของหลาย Agent
พัฒนาการทำงานกับ Versioned Graph โดยตรง ทำให้ Agent สามารถ ให้เหตุผล (Reasoning) บนชุดความรู้ที่เปลี่ยนแปลงและมีวัตถุนาคารได้	

Industry Benchmark: Execution & Knowledge Graphs

Execution Graph

UET Execution Graph

The World

เกี่ยบกับ: DAG Orchestrators (Prefect, Airflow), LangGraph

ความเหมือน: เป็นตัวจัดการการทำงานแบบ DAG (Node-Edge Flow)

ความต่าง: เป็น Version-aware และสามารถทำ Reasoning Diff เพื่อเปรียบเทียบผลลัพธ์จากการใช้ “ความรู้” คุณลักษณะเวอร์ชันในการตัดสินใจได้

Knowledge Graph

UET Knowledge Graph

The World

เกี่ยบกับ: Graph Databases (Neo4j, ArangoDB), RDF Ontologies

ความเหมือน: จัดเก็บและ Query ข้อมูลในรูปแบบ Graph

ความต่าง: ออกแบบมาเพื่อรับ Version Evolution และการทำ Semantic Diff ระหว่างเวอร์ชันของ Knowledge ได้โดยกำเนิด

Industry Benchmark: Routing, Flow Control & Security

Model Routing

เกี่ยบกับ: OpenRouter, LiteLLM, Together AI

ความเหมือน: รองรับการเลือกใช้ LLM ได้หลายใบเดียว

ปรับแต่งเกณฑ์การเลือกโมเดลได้ลึกกว่า โดยอิงตาม **Context** และสถานะปัจจุบันของ Knowledge Graph ไม่ใช่แค่ประเภทของงาน

Flow Control + Event Bus

เกี่ยบกับ: Event-driven Architectures (Kafka, Node-RED)

ความเหมือน: จัดการ Event แบบ Asynchronous

มีกลไก **Context Adaptation** และเชื่อมโยงการ Sync สถานะกับ Knowledge Graph โดยตรง

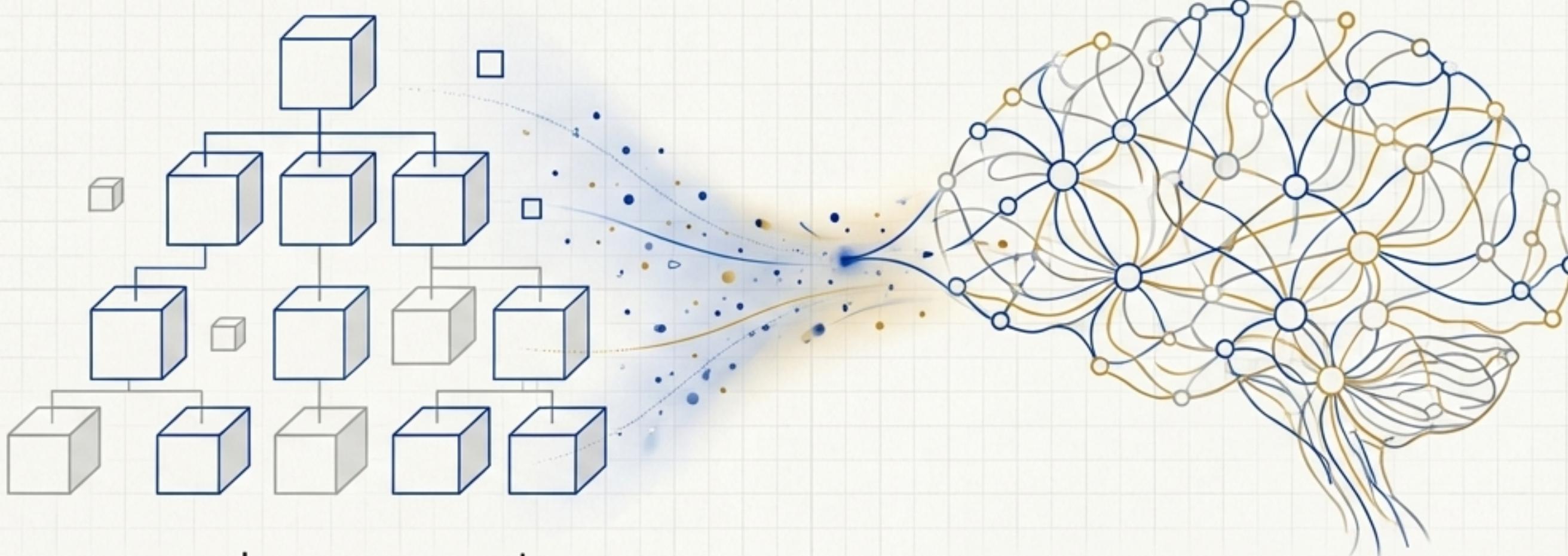
Security + Permission

เกี่ยบกับ: OAuth, RBAC, OpenPolicyAgent

ความเหมือน: เป็น Framework สำหรับ Authentication & Authorization

เชื่อมโยงสิทธิ์เข้ากับ 'เจตนา (Intent)' ของ AI Agent และการให้เหตุผล ไม่ใช่แค่ Role ของผู้ใช้

The Philosophical Shift: From Instruction-Follower to a Knowledge Collective



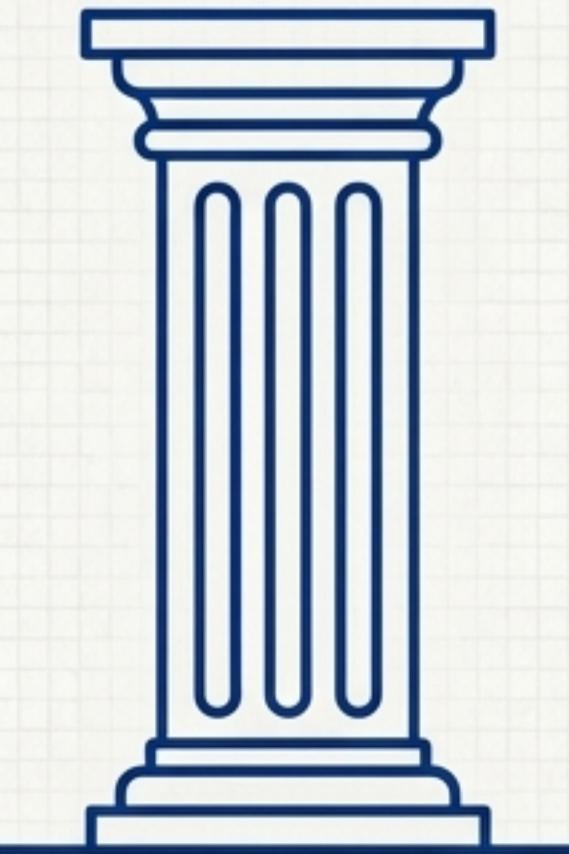
จาก: “ระบบที่ทำงานตามคำสั่ง”
(An Instruction-following System)

สู่: ‘หน่วยร่วมมือของความรู้และเหตุปัจจัย’
(A Systemic Collaboration of Causes
and Knowledge)

เราไม่ได้มองระบบเป็นแค่เครื่องมือ แต่เป็นระบบบิเวศของความรู้ที่มีชีวิตและเปลี่ยนแปลงได้

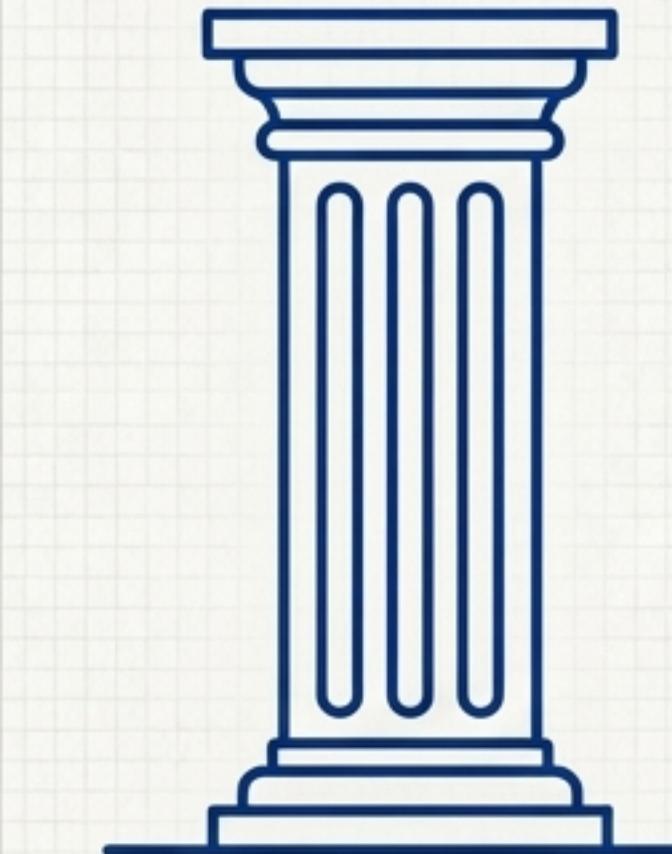
The Four Pillars of UET's Design Philosophy

1. Impact-Based Ethics



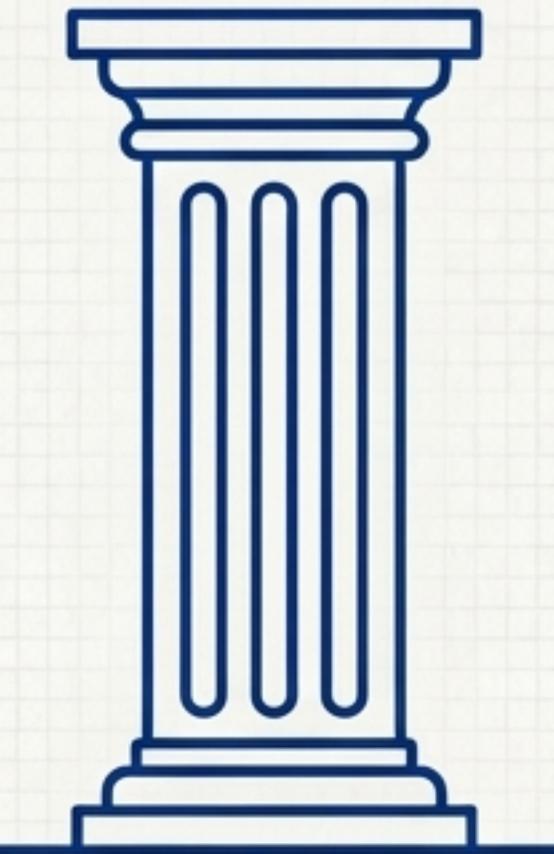
ให้ความสำคัญกับ
‘ความสมดุลของผลกระทบ’
ที่เกิดจากการตัดสินใจของ AI
มากกว่าแค่ Performance
หรือ Accuracy

2. Auditable & Versioned Knowledge



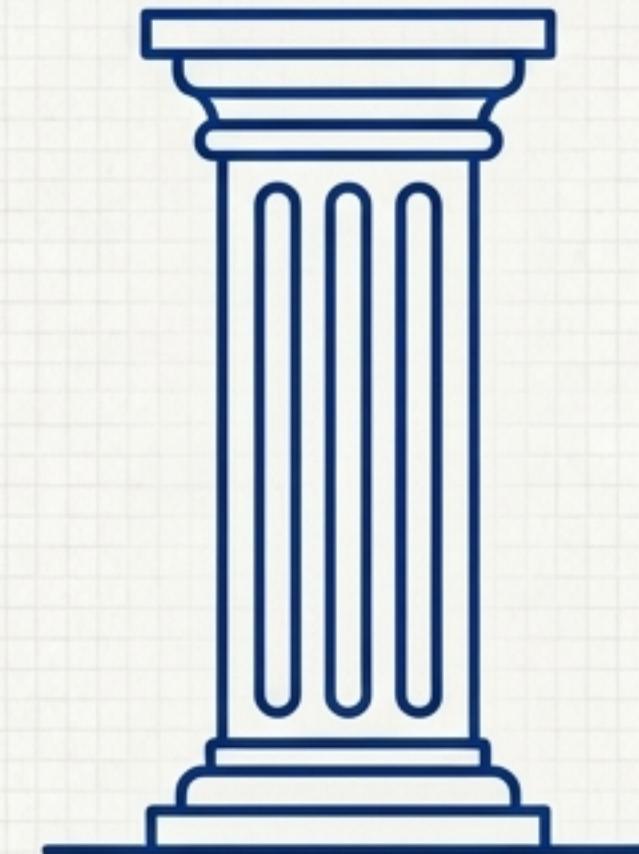
ความรู้คือสิ่งที่ต้องตรวจสอบ
และย้อนรอยได้เสมอ
ทุกการเปลี่ยนแปลงต้องมี
ที่มาที่ไปและเปรียบเทียบได้

3. Systemic Collaboration



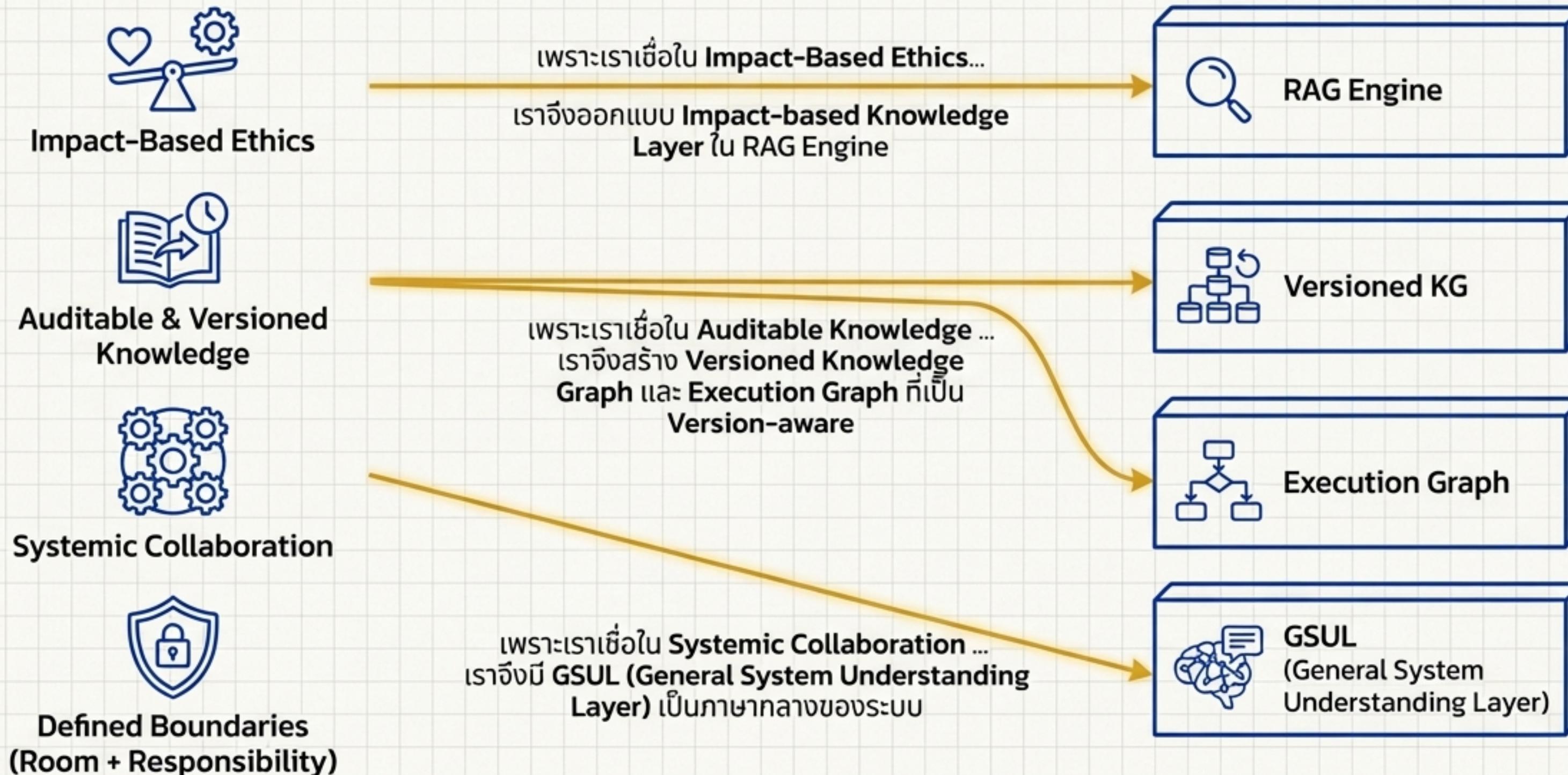
ทุกโน้มถูกล็อกองค์ความพยายามที่
ทำงานร่วมกันผ่านภาษากลาง
(Shared Layer)
ไม่ใช่การทำงานแบบไฮโล

4. Defined Boundaries (Room + Responsibility)



ขอบเขตความรับผิดชอบ
ที่ชัดเจนของแต่ละโมดูล
คือหัวใจของความเสถียร
และความสามารถในการ
บำรุงรักษา

Synthesis: How Philosophy Shapes Architecture



UET Platform: Key Architectural Differentiators



Knowledge is Dynamic, Not Static

ระบบถูกสร้างบนพื้นฐานที่ว่า 'ความรู้' มีเวอร์ชันและวิวัฒนาการได้ (Versioned KG)



Reasoning is Auditable, Not a Black Box

ทุกการตัดสินใจสามารถย้อนรอยและเปรียบเทียบผลลัพธ์ได้ผ่าน Version-aware Execution Graph



Impact Overrides Relevance

การเลือกและจัดการความรู้คำนึงถึง 'ผลกระทบ' เป็นอันดับแรก



Philosophy-Driven Design

ทุกการตัดสินใจทางสถาปัตยกรรมมีรากฐานมาจากปรัชญาที่ชัดเจน ไม่ใช่การเลือกเทคโนโลยีตามกระแส