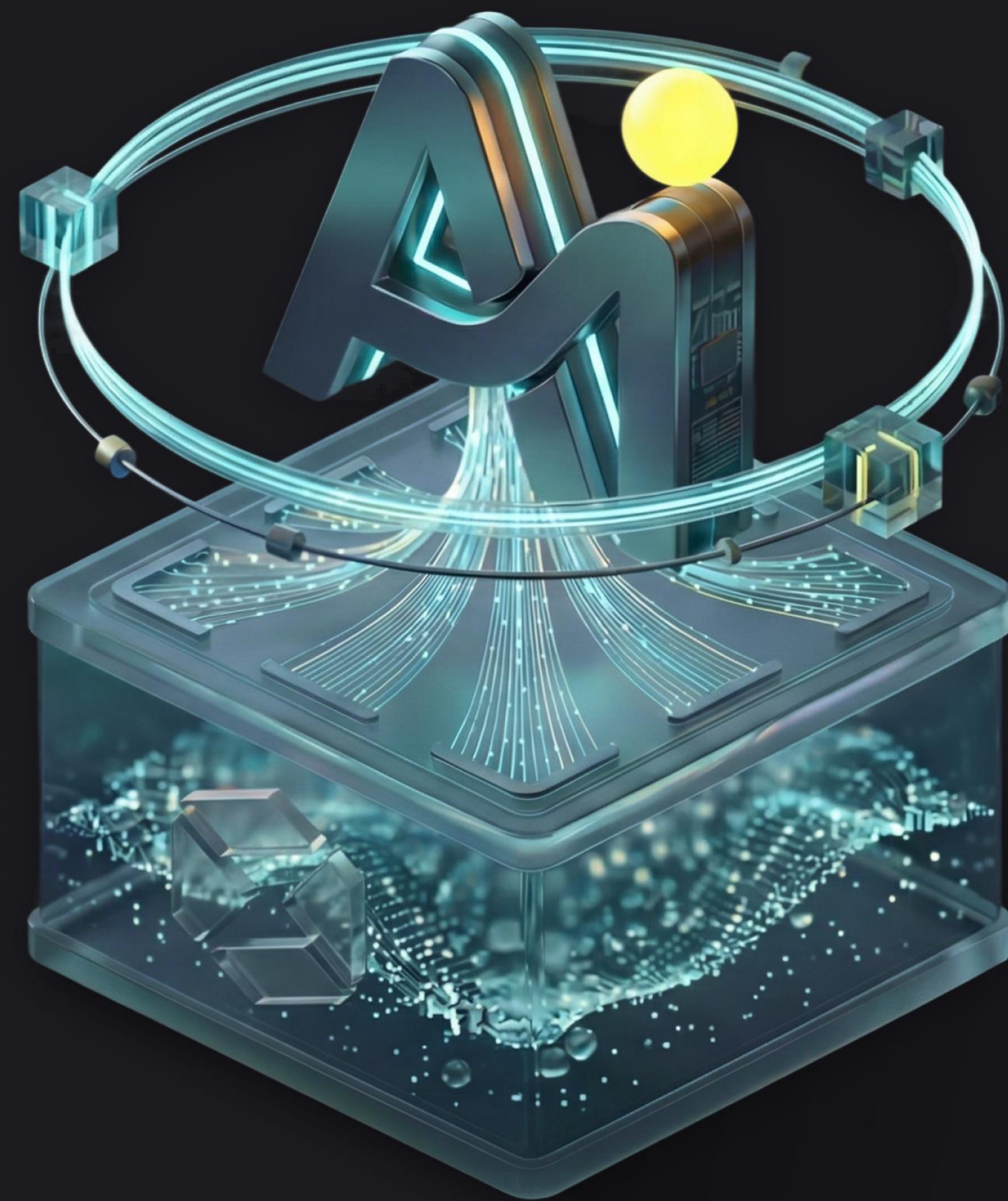


当 AI 遇上 DATA

数据范式正在重写



以本体论重构数据

图技术驱动 AI 理解世界

数据，尤其是私有化数据，是 AI 时代企业的护城河



生产制造

大模型

-
- 高确定性
- 高稳定性
- 高实时性

- "幻觉"
- 概率性
- 迟滞性



- **独占性** (不可复制)
- **累积性** (越积越多)
- **反馈性** (越用越好)

持续领先

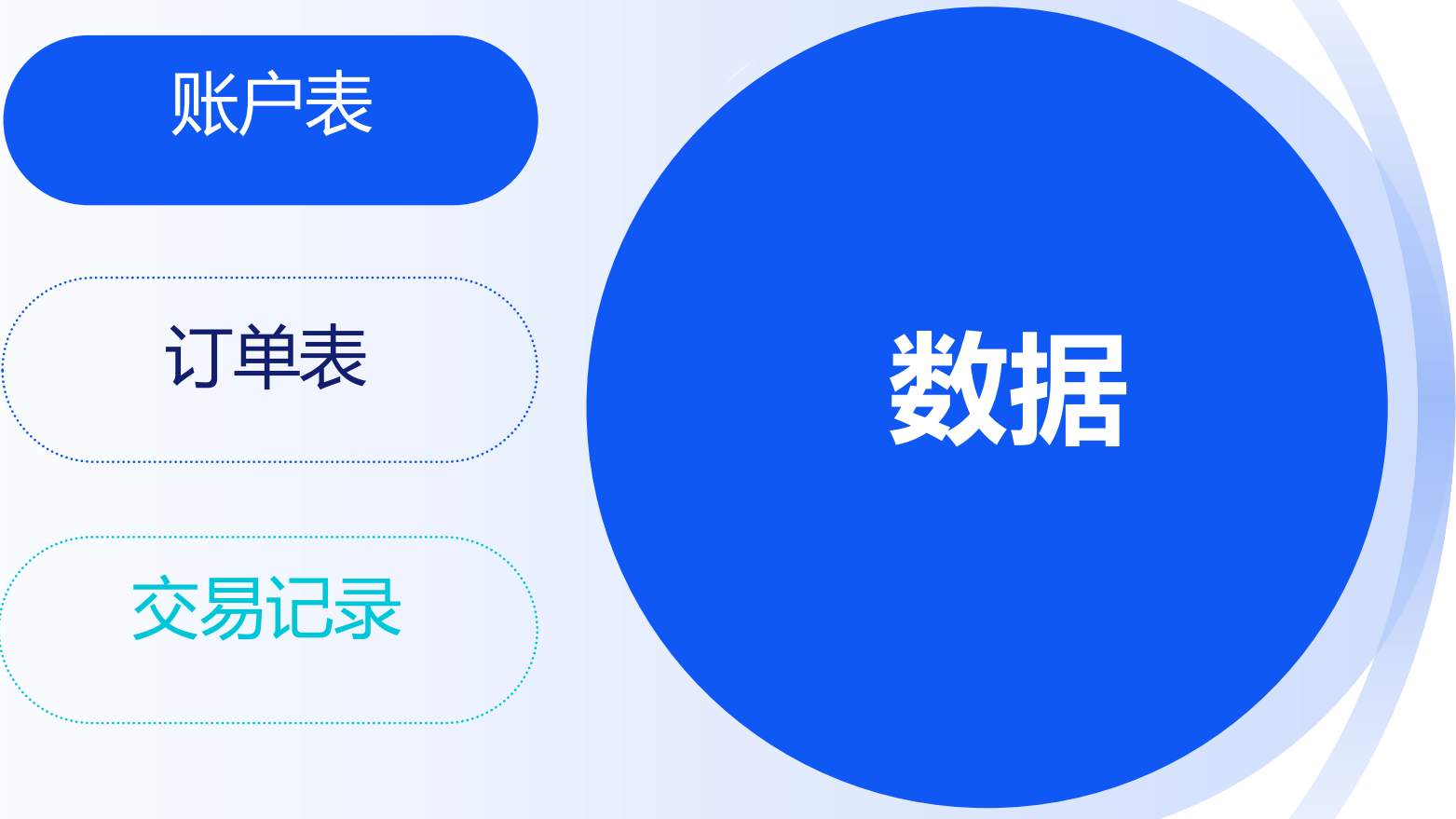
AI 目前主要在研发与销售两端应用，生产制造比例占比仅 **24%**
(数据来源于华为近几年在 AI 领域的经验分享)

从数据到知识，从事实到事理

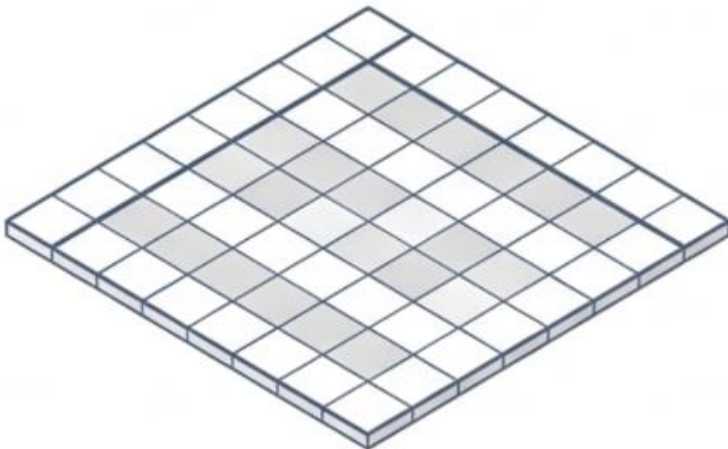
本体定义(Thomas R. Gruber)：

Ontology is a **formal**, explicit **specification** of a **shared conceptualization**

本体是对某一领域“共享概念模型”的形式化、明确化描述



系统中心视角 (System-Centric View)
Table / Row / Column

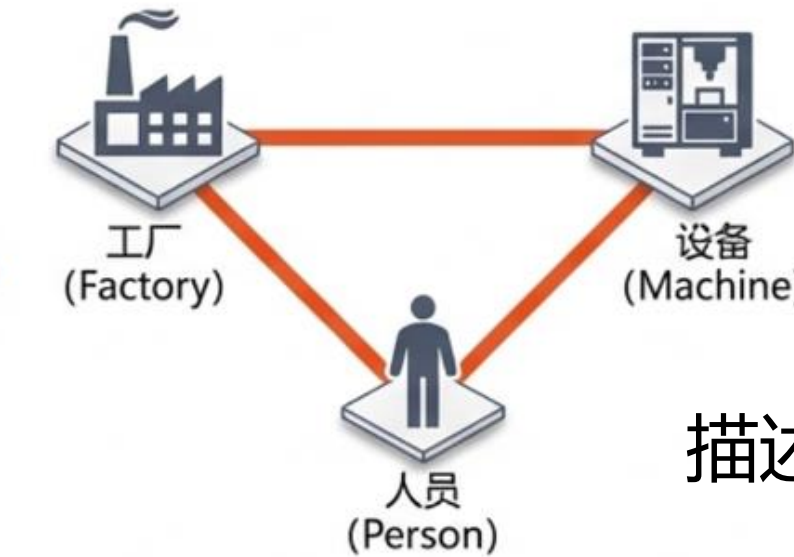


本体定义	作用	企业难题
Conceptualization (概念化)	“世界是什么”	不显性 碎片化
Explicit (显式化)	消除隐式理解	弱语义 歧义性
Formal (形式化)	让机器可处理	弱语义 歧义性
Shared (共享)	统一认知标准	随意性 歧义性 碎片化

本体是 AI 落地企业核心的数字骨架



人类业务视角 (Human-Centric View)
Object / Action / Event



描述的是**事理**

第一次进入了一个“必须让机器理解世界”的时代

过去不是没有语义，而是语义被藏在了人、流程和固定报表里。

任务简单

记录交易、做统计分析、按字段检索、满足流程系统协作。系统更像"记录器"和"执行器"，不是"理解器"

场景稳定

查询路径相对固定，很多判断可以交给报表、规则引擎和人工经验兜底。

固定报表

大量分析是预定义好的，人只消费结果，不要求系统持续解释关系网络。

人肉翻译

业务人员承担了跨系统对齐、补语义和理清上下文的大部分成本。

命名约定

不同系统之间靠经验映射字段、代码和值域，而不是靠统一对象模型。

流程制度

SOP、审批链和部门分工，替系统补上了很多对象边界。

机器理解
机器计算

从“数据驱动” → “语义驱动”

从“人解释” → “机器理解”

从“预定义的系统”进行推理
→ “开放系统”进行推理

信息系统：记录 & 处理数据

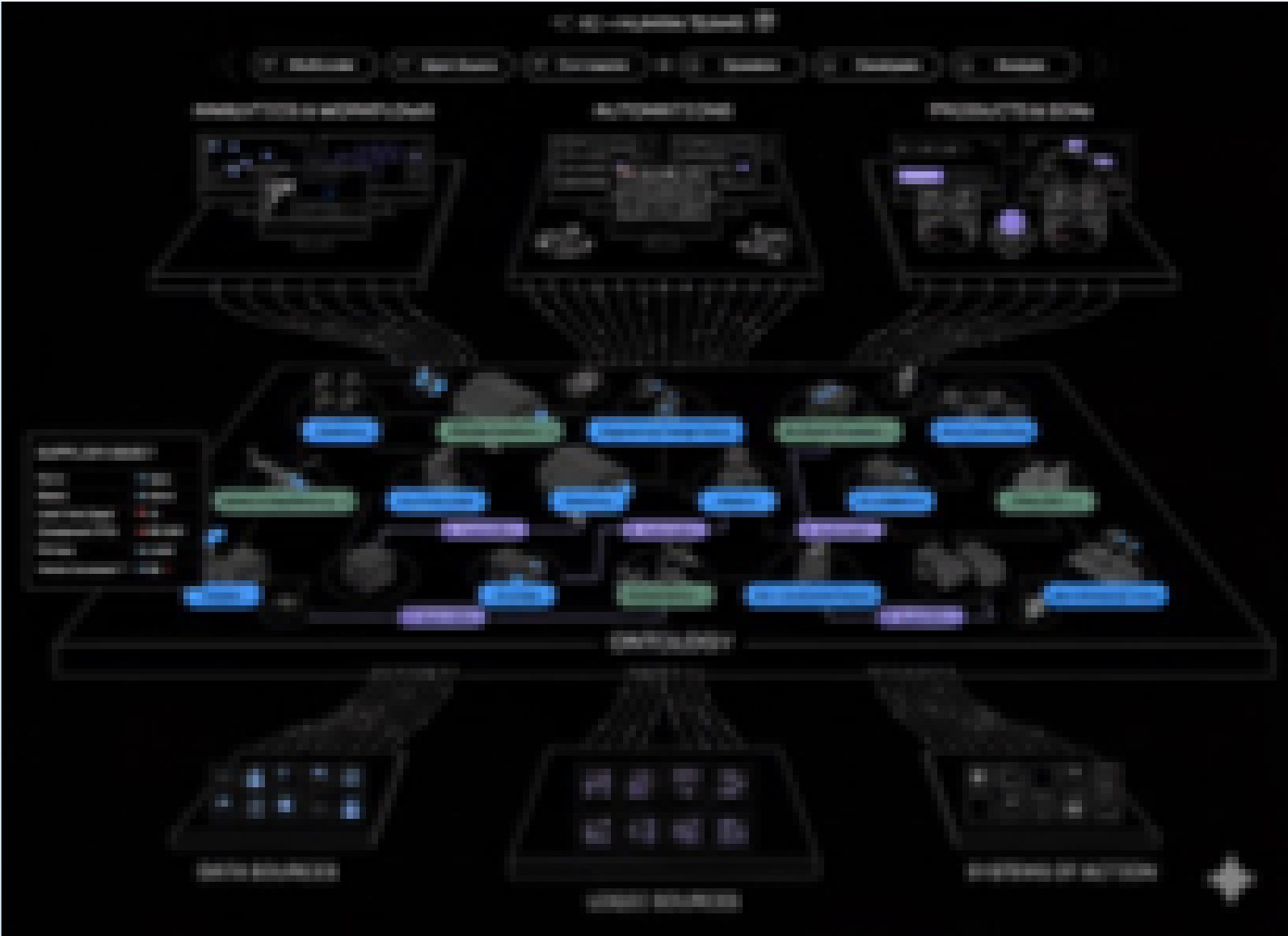
认知系统：理解 + 推理 + 决策



业界实践：Palantir 的操作本体论实践

Palantir Ontology

Palantir在其Foundry和AIP平台中，将本体论从学术概念转化为企业级操作工具。



本体=数据的「业务语义层」

核心思想：
让业务用户和系统能够以统一的「对象语言」操作数据，而不需要理解底层数据库结构，充当了企业数字孪生的语义核心。

对象类型（Object Types）

将原始数据表映射为业务对象（如「客户」「订单」「资产」），屏蔽技术细节

关系链接（Link Types）

定义对象间的关系（一对一/一对多），跨数据源自动关联，形成知识网络

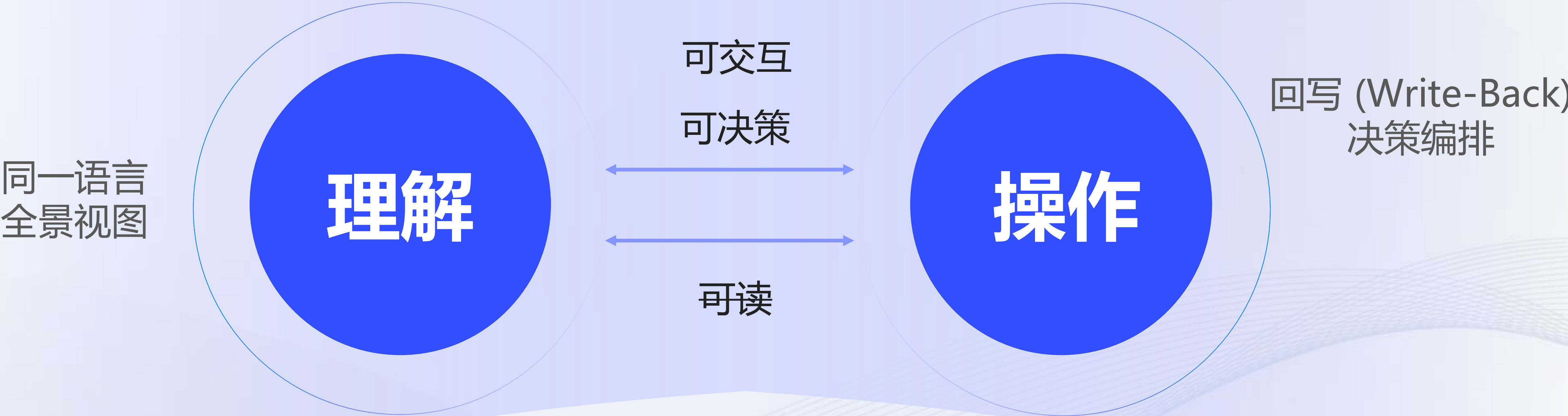
操作动作（Actions）

将业务操作绑定到本体对象，触发下游系统更新，实现「读写一体」的本体

AI 接入层（AIP）

LLM通过本体理解数据语义，Agent可以用自然语言查询和操作企业知识图谱

本体论核心价值：理解+操作

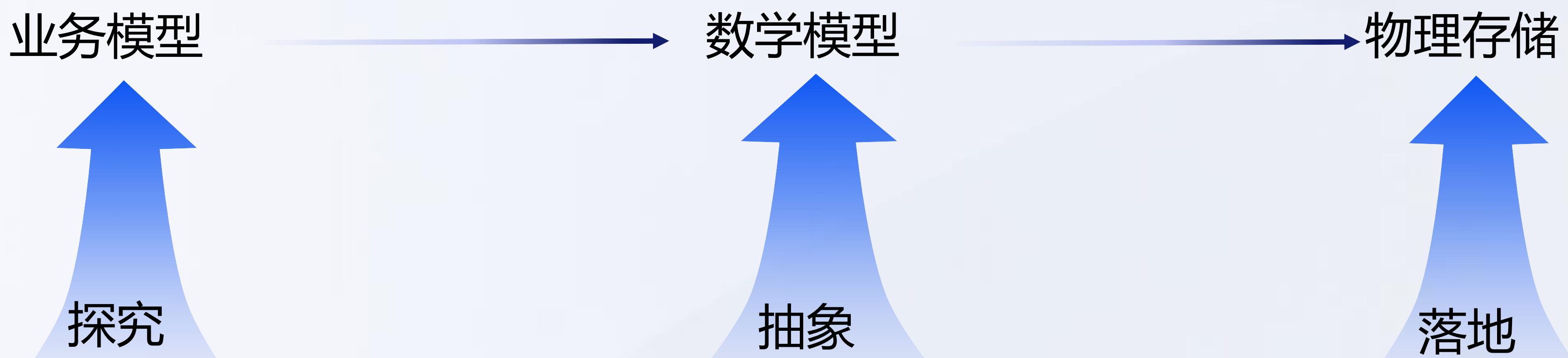


打破部门壁垒，统一业务语义
聚合业务系统，形成 360 度视图

不仅仅是查询，允许回写
模拟仿真，基于关系推演决策后果



透过现象看本质：本体即是“图模型”



哲学本体论概念	Palantir 工程实践概念	图技术概念
Entity (实体)	Object (对象)	Node/Vertex (节点)
Relation (关系)	Link (链接)	Edge/Relationship (边)
Cognition (认知) Inquiry (探问)	Search Around (关联搜索)	Query (图查询) Computing (图计算)

本体论打通了业务模型与数学模型之间的语义鸿沟

图技术打通了数学模型与物理存储之间的实现鸿沟

图技术，让本体 “可计算、可推理、可进化”



原生图存储

基于属性图模型（节点-边-属性）原生存储，支持文本、时间、空间、向量等多类型数据，天然承载本体中的类、实例及其属性信息



多跳关系遍历

支持高性能多跳路径遍历，能够高效实现本体中的继承、传递等关系推导



动态 Schema

支持灵活Schema演进，无需迁移即可扩展实现类型与关系，天然适配本体的持续演化与开放世界假设（OWA）



推理集成

支持相似度计算、关系预测、标签传播等图推导能力，在本体之上实现隐式关系发现与知识补全，让本体具备 “自我扩展” 的能力



GQL模式查询

GQL 通过直观化模式 (a)-[r]-(b)表达本体
查询，使实体关系以接近人类认识的方式进行匹配与检索



实时更新

支持本体实例与关系的实时增量更新与动态推理，使知识库能够持续演化，保证在推理和决策中的数据一致性与可靠性

哪些企业最迫切需要“本体论”破局

多系统	高合规	长链路	高异构	高推理成本
同一对象在多个应用里分裂存在。	需要解释决策链路与对象关系。	判断依赖上下游与传播路径。	结构化、事件、文档和规则同时存在。	人能理解，但系统总在"拼接答案"。

如果一家公司在这五项里命中三项以上，建议需要使用“本体论”。

举例：

同一用户、商家、内容和行为会在多个触点里重复出现。	不仅要给结论，还必须解释对象关系、决策路径和责任边界。	上下游、事件流和规则层持续耦合，字段视角已经不够用了。
平台型企业 	强监管行业 	复杂运营网络 

本体实施整体架构图

场景应用



AI



BI



决策引擎

...

本体层



产品

当事人

账户

介质

渠道

地理位置

协议

资源项

事件

...

原生图存储

图

向量

文本

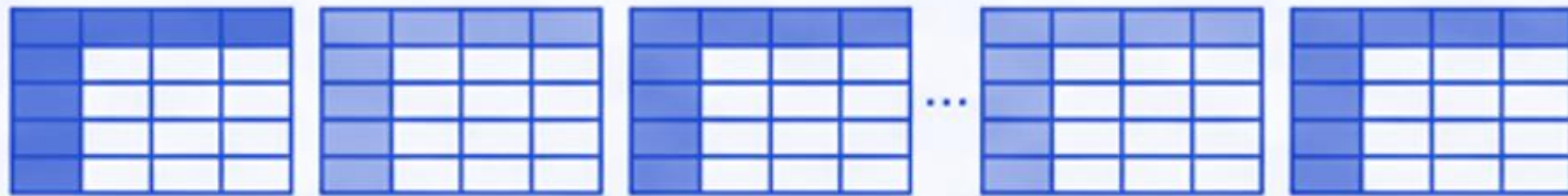
Geo/时序.....



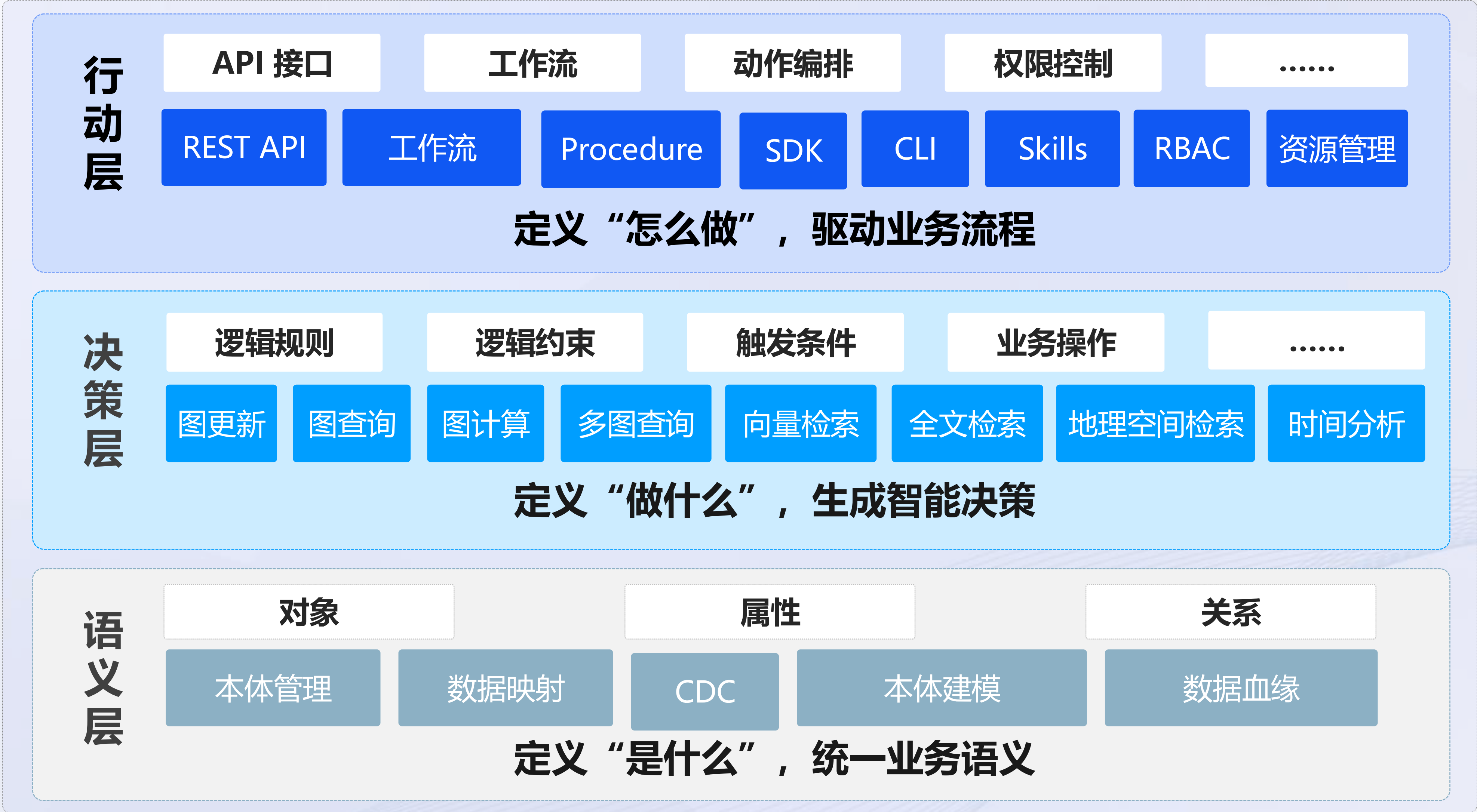
抽取与映射



数据源



本体层实现逻辑



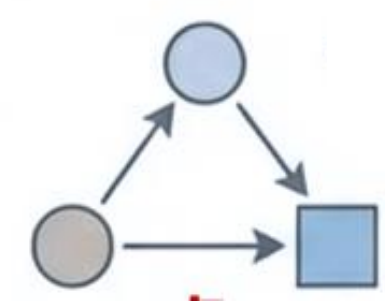
* 《本体智能研究报告》-中国人工智能产业发展联盟（AIIA）



企业本体论转型的四个建议阶段

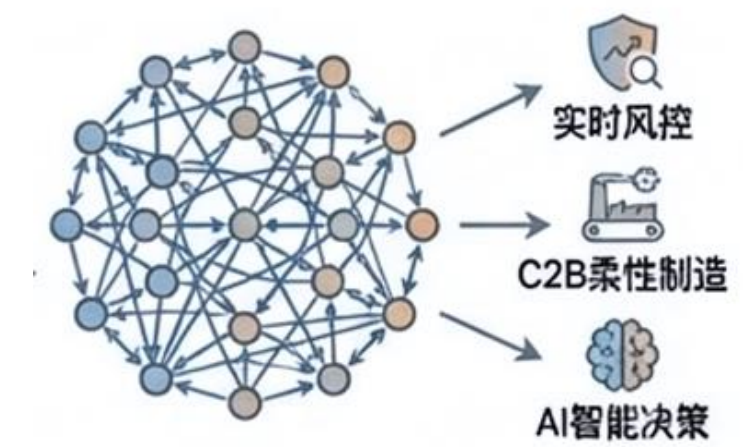
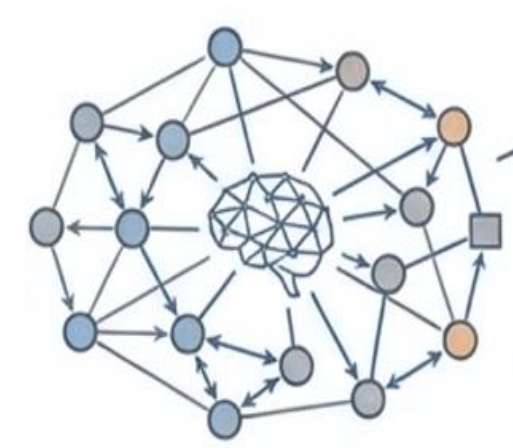
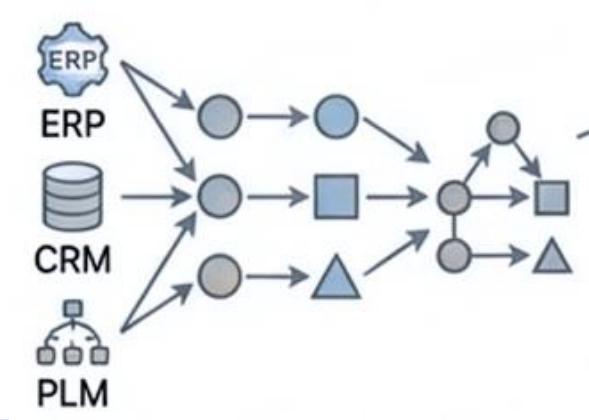
关键动作:

- 1. 高价值业务域, 局部概念建模试点
- 2. 消除多义词与同义词



关键动作:

- 1. 构建覆盖全企业的本体数据;
- 2. 建立数据映射, 数据流动起来



阶段一 业务本质建模

梳理企业核心实体与承诺逻辑, 建立初步的词汇一致性, 补齐语义技术短板

点

阶段二 打破数据孤岛

建立标准化知识工程流程, 跨越物理数据孤岛

面

阶段三 知识迭代

利用图技术深挖实体间的隐形关联, 建立动态企业知识大脑

关键动作:

通过深度关联查询、社区分析、链路预测、标签推导等技术, 自动化补全知识图谱, 实现知识体系的持续更新与精细化管理。

动态

阶段三 智能化驱动

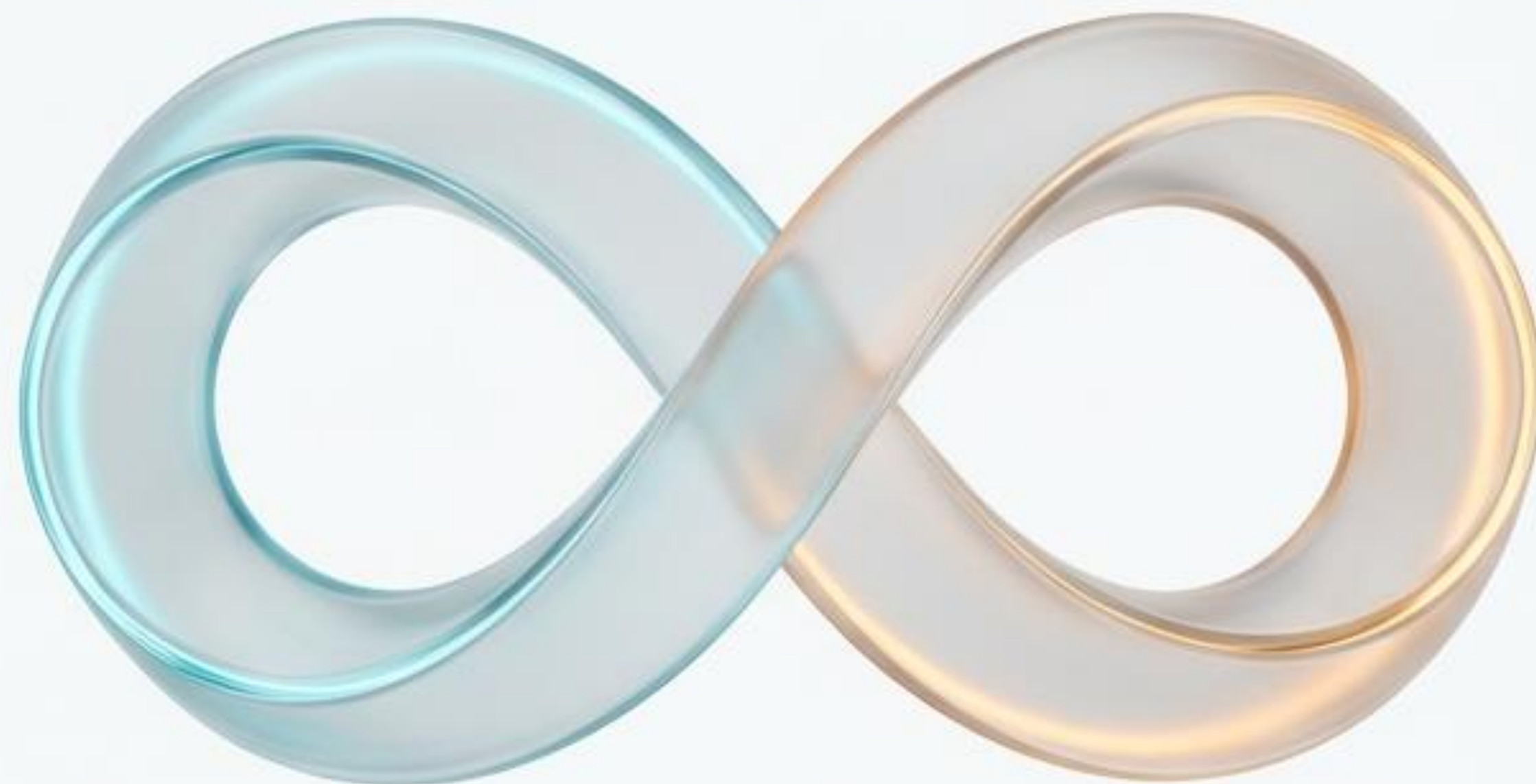
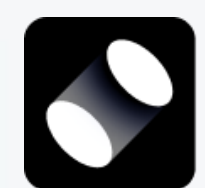
语义层成为驱动 AI 和自动化决策的企业“大脑”

关键动作:

- 1. 对接大模型, 支撑精准、可解释的企业级智能问答;
- 2. 配置推理规则引擎, 实现全企业的隐形风险自动预警与业务洞察;
- 3. 建立闭环反馈机制, 将 AI 生成的新知识沉淀至本体底座

智能

Q&A



Evolving into the Next-Generation Real-Time Intelligent Platform for the Agent Era

The Bi-Directional Evolution: Empowering AI Applications and Being Reshaped by AI

Andy Chen, MirrorShip CTO

Today's Agenda

- 01 Context:** The AI/Agent era reshapes data system requirements
- 02 Outward:** How StarRocks serves as the **real-time data foundation** for AI/Agent
- 03 Inward:** How AI/Agent powers StarRocks' own intelligence and autonomy
- 04 Conclusion:** From analytics engine to self-evolving intelligent data platform

AI and Agents are Reshaping the Boundaries of Data Systems

Historically, data systems served reports, BI, and offline analytics. Today, the primary requesters are **Large Models, Copilots, Business Agents, and Automated Workflows**. We are no longer just serving Human Analysts—we are serving **Machine Decision-Makers**.

	Traditional Data System	AI-Era Data System
Target User	Human Analysts	Models & Autonomous Agents 
Interaction	Manual SQL	Natural Language & Tool Calling 
Goal	Data Viewing & Reporting	Understanding, Judging & Acting 
System Role	Analytical Engine	Intelligent System Base 

New Demands for Real-Time Analytics in the Agent Era

Introducing AI fundamentally changes workload patterns, capability requirements, and system governance.



Absolute Freshness

Second or sub-second data freshness directly dictates Agent decision quality.



Diverse Queries

The rapid convergence of structured, semi-structured, text, and vector retrieval.



Multi-User Diversity

Concurrent usage by BI users, Copilots, automated tasks, and autonomous Agents.



Autonomous Systems

Traditional manual operations fail under high concurrency, micro-requests, and burst traffic.

The new era demands a more real-time, open, and intelligent platform.

Upgrading from Real-Time OLAP to a Real-Time Intelligent Base

“StarRocks is no longer just a faster SQL Engine. It must become the Real-Time Intelligent Data Platform for the AI & Agent Era.”

Outward: Empowering AI & Agents.

Providing data and intelligence as a service to external AI models, agents, and automated workflows.

Inward: Utilizing AI & Agents to reshape the core.

Integrating AI and agentic processes to optimize internal system operations, query execution, and self-management.

Evolution: From Query Capabilities to Decision & Autonomous Capabilities.

Shifting focus from analytical results to direct support for automated decision-making and autonomous actions.

The Bi-Directional Evolution Path

StarRocks Empowering Agents

- Real-Time Data Ingestion & Analysis
- Semantic & Metric Layers
- Open Agent Interfaces
- Unified Structure + Text + Vector Analytics
- Cloud-Native Elasticity & Governance

StarRocks

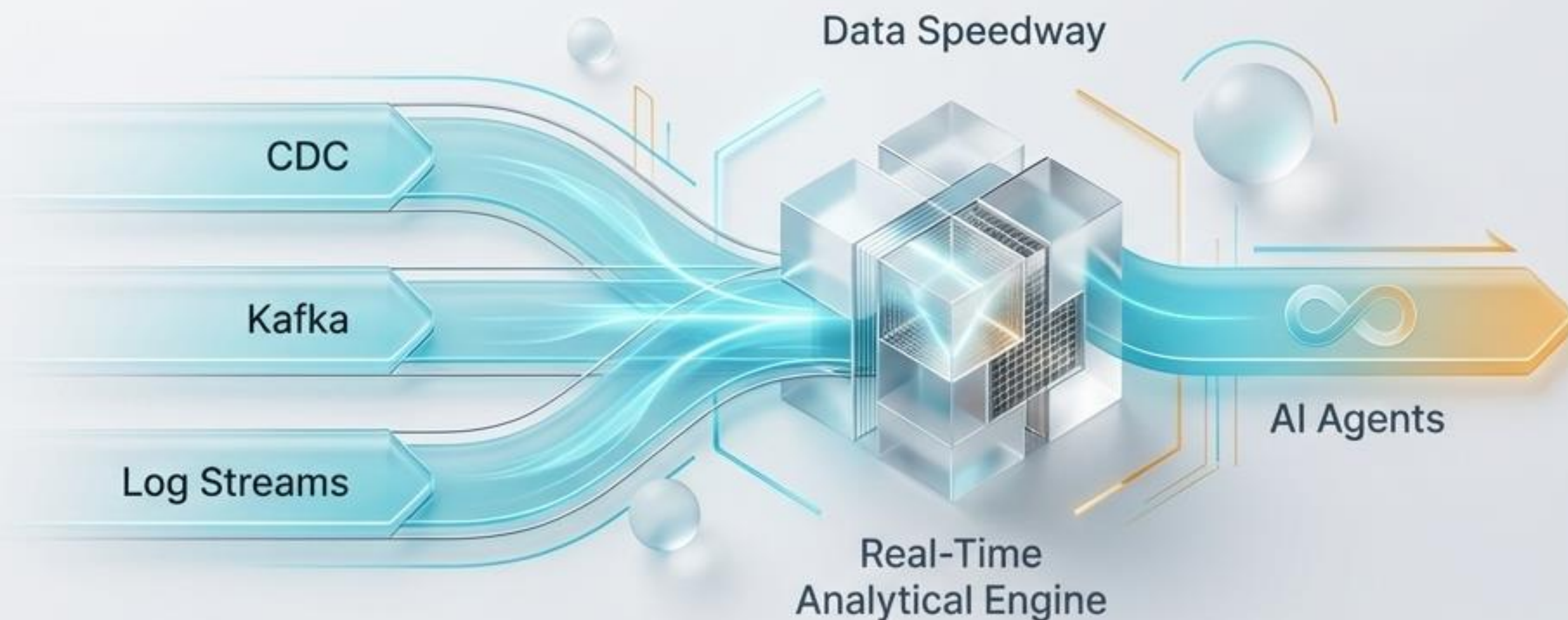
Agents Empowering StarRocks

- Intelligent Optimization
- Intelligent O&M
- Intelligent Inspection
- QA & Expert Copilot
- Progression from Advice to Autonomous Closed Loop

This is a bi-directional enhancement, not a one-way support relationship.

Autonomous Decisions Demand Absolute Data Freshness

For Agents, data value isn't about mere existence—it's dictated by freshness. A few minutes of delay renders data useless for real-time risk control, recommendations, and operational automation.



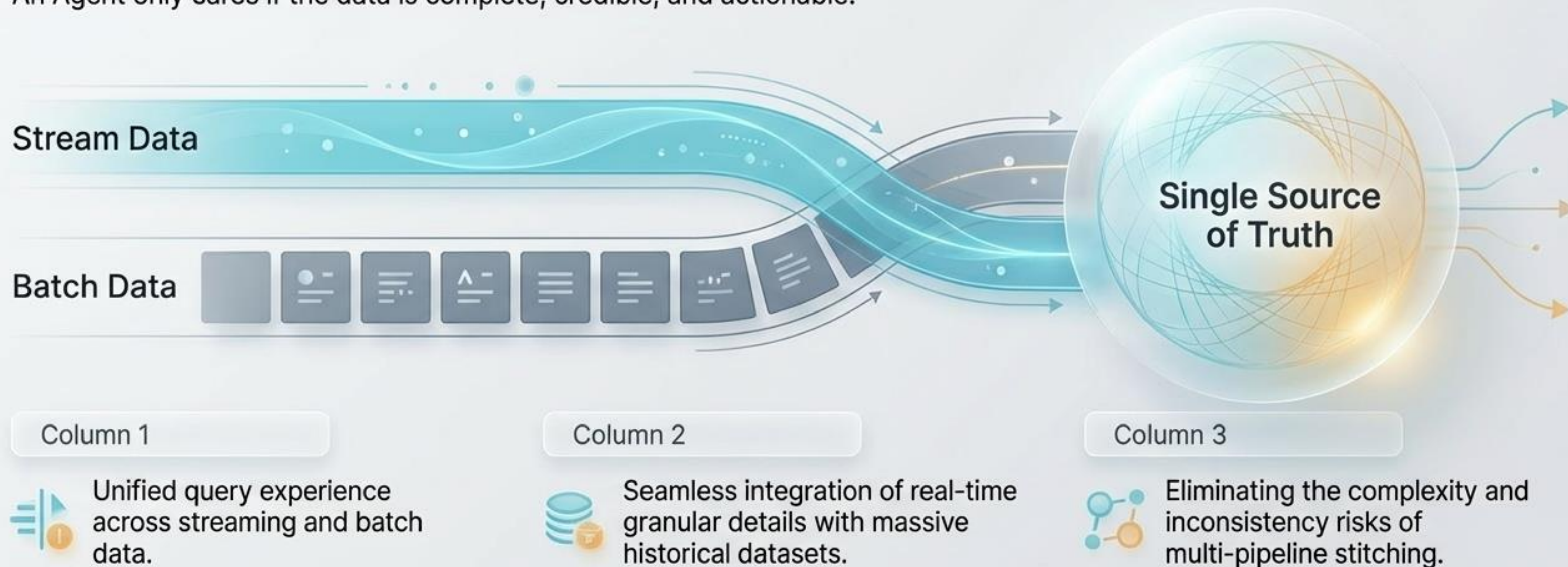
Key Capabilities

- Strengthened ingestion for CDC, Kafka, and changelogs.
- Unified consumption of Lakehouse incremental data.
- Write-and-query, incremental visibility, and ultra-low latency.
- Directly supporting real-time marketing, risk control, and recommendations.

Without high-freshness data, there are no high-quality real-time Agents.

Agents Require Unified Data Freshness and Consistency

Decision chains do not care if data originates from streams, batches, or a Lakehouse. An Agent only cares if the data is complete, credible, and actionable.



Unifying stream and batch is no longer just an architectural debate; it is fundamentally an issue of data credibility in the AI era.

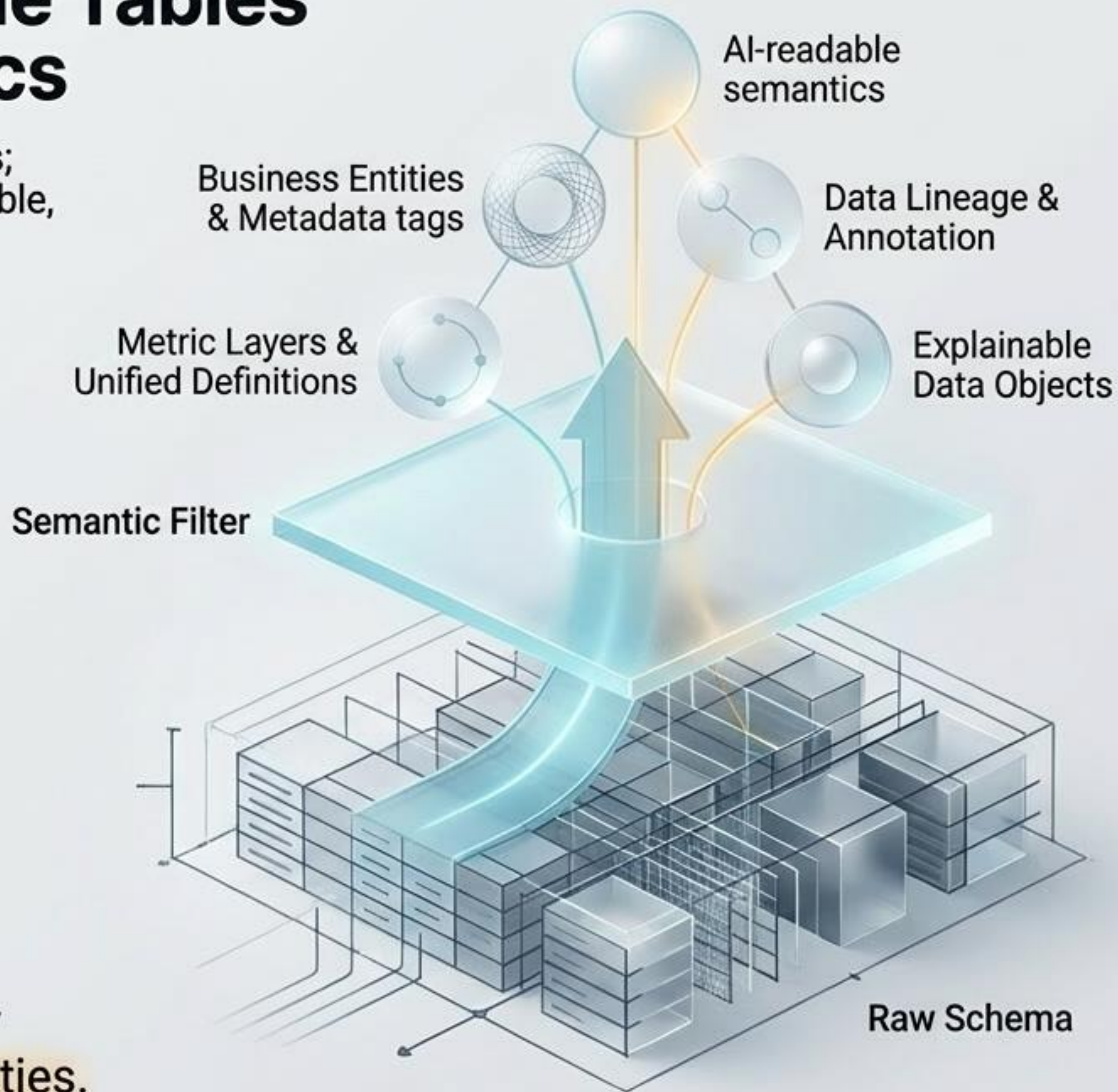
Transitioning from Queryable Tables to Understandable Semantics

Understanding raw schemas is difficult for human engineers; for Models and Agents, the real hurdle isn't accessing the table, it's understanding the business logic behind it.

Key Capabilities

- Constructing robust metric layers, semantic layers, and unified definitions.
- Providing clearer metadata, data lineage, tagging, and annotation systems.
- Enhancing the explainability and reusability of data objects.
- Helping Agents instantly answer: What is this field? How is this calculated?

The future of competition lies not just in query performance, but in **semantic delivery capabilities**.



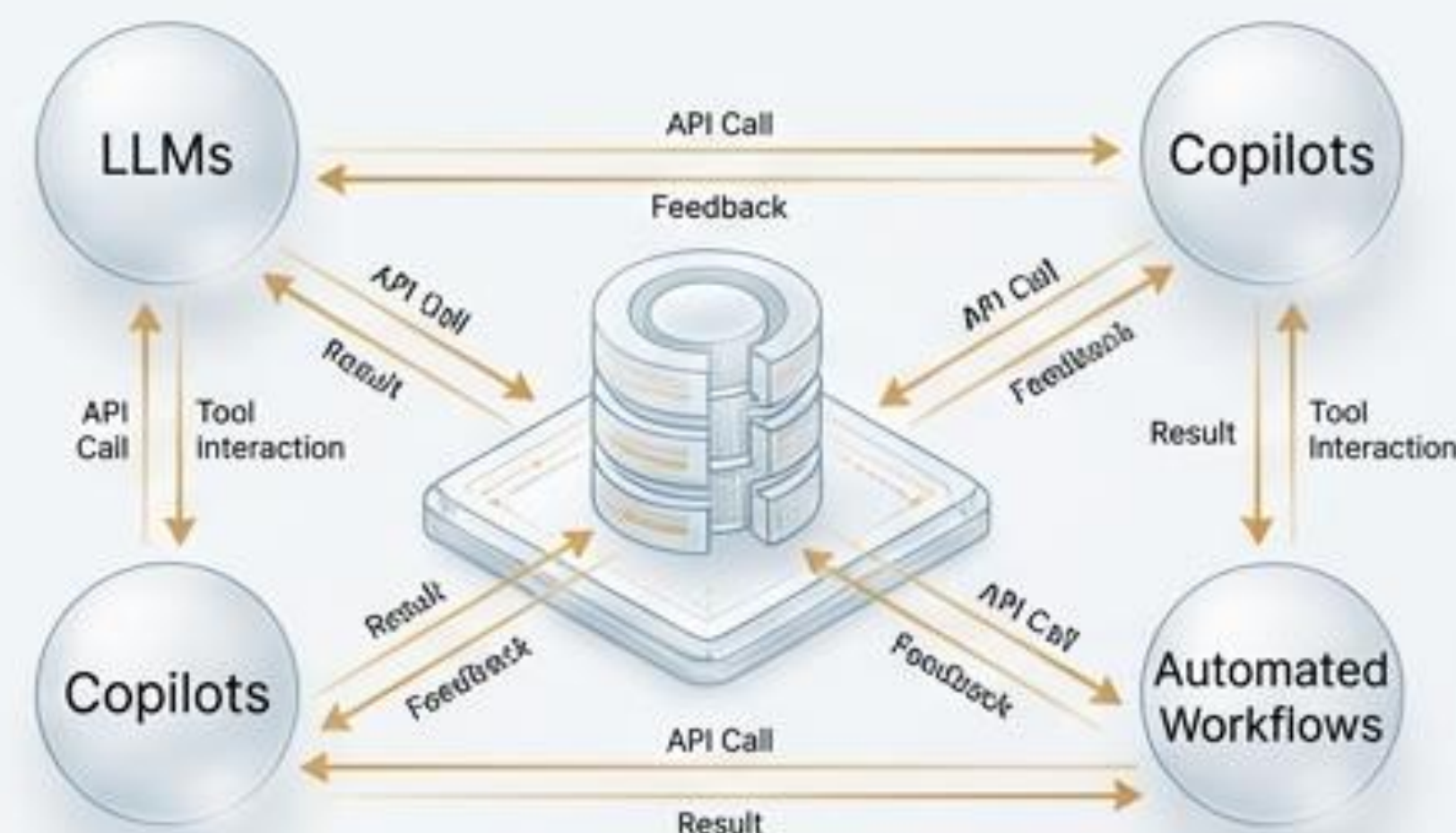
Redesigning Interfaces for Non-Human Users

When the user base expands from analysts to LLMs, Copilots, and Autonomous Agents, legacy JDBC and SQL are insufficient for Agentic calling patterns.

Human Interface (JDBC/SQL)



Agentic Interfaces (API/Tool Calling)



Key Capabilities

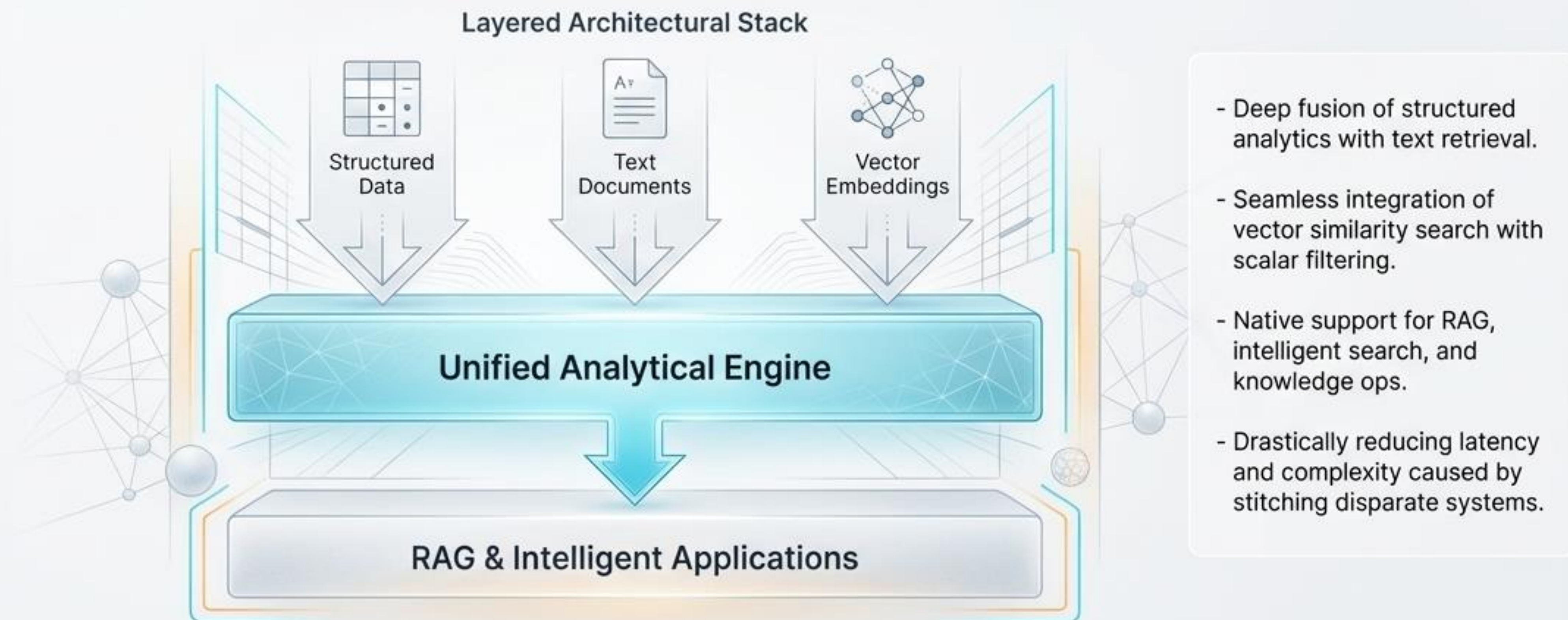
- Comprehensive schema and metadata exposure.
- Closed-loop support for SQL generation, validation, and error correction.

- API-driven, tool-ready, and function-calling query services.
- Context awareness and result-reuse across multi-turn tasks.

An Agent-oriented database isn't just queryable—it must be inherently understandable, tunable, and controllable.

The Future of Analytics Blends Structure, Text, and Vectors

AI workloads are increasingly hybrid. Structured metrics, text snippets, embeddings, and behavioral logs must be jointly analyzed and filtered within a single pipeline.



The system that best handles hybrid data becomes the definitive foundation for AI-native applications.

Agent-Driven Workloads Amplify Elasticity and Governance Needs

Agents don't just bring more requests; they bring **automated**, **bursty**, and **long-chain** execution patterns. **Resource governance, cost control, and security auditing** become paramount.



In the AI era, raw performance must be matched by ironclad governance.

Empowering AI is Only Phase One

If StarRocks only enables AI, it remains merely a system that serves AI. The true next-generation differentiator is bringing AI inward—making Agents active participants in StarRocks' own optimization, operations, and autonomous evolution.

We are making Agents a fundamental capability of the database itself.

Shifting from Manual Tuning to Autonomous Optimization

Database tuning heavily relies on scarce expert intuition. In increasingly complex workloads, manual experience cannot sustain optimal performance. Agents introduce workload-aware, autonomous optimization.



Key Capabilities

- 1** Auto-recommending partitions, indexes, Sort Keys, and Buckets.
- 2** Intelligently identifying opportunities to build Materialized Views.
- 3** Automated SQL diagnostics and dynamic rewrite suggestions.
- 4** Reinforcing Optimizer decisions based on historical workload analysis.

The core value of intelligent optimization is a system that deeply understands its own query workloads.

Shifting from Reactive Response to Proactive Governance

Traditional O&M relies on "troubleshoot after failure". The complexity and velocity of the AI era demand a pivot to proactive governance and pre-emptive intervention.

Key Capabilities

- 1** Automatically identifying hotspots, data skew, resource contention, and slow nodes.
- 2** Fusing logs, metrics, traces, and profiles for instant root-cause analysis.
- 3** Auto-generating scaling, parameter tuning, and capacity planning suggestions via a dedicated DBA Copilot.



**Intelligent O&M does not replace engineers;
it acts as a multiplier for their capabilities.**

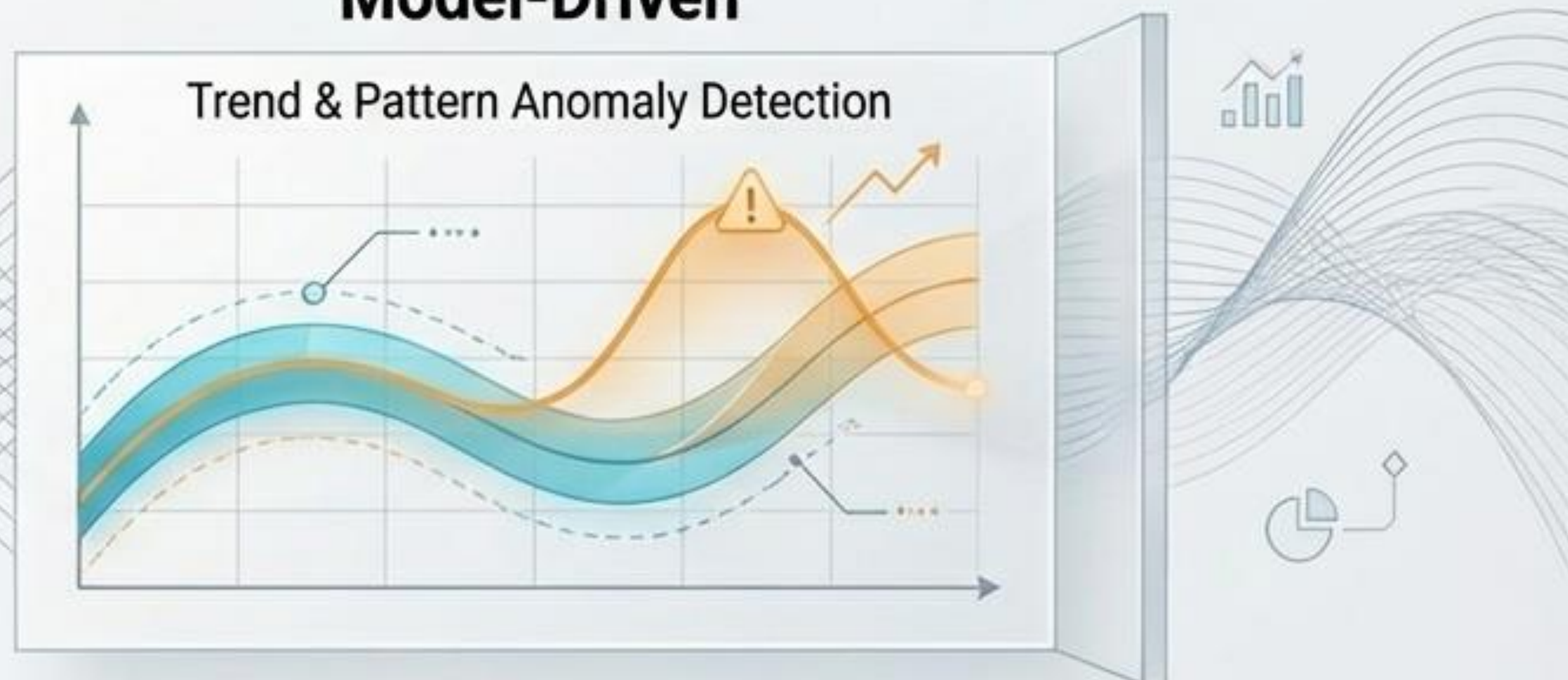
Evolving from Rule-Driven to Model-Driven Inspection

Many system inspections remain trapped in static thresholds and rigid rules. True systemic risks stem from subtle trend shifts, pattern anomalies, and multi-factor stacking—areas where Agents excel.

Rule-Driven



Model-Driven



Key Capabilities



7x24 intelligent inspection of query latency, ingestion pileups, and load volatility.

The future of inspection relies on enhanced contextual understanding, not just more rules.



Identifying anomalous trends rather than just threshold breaches.



Auto-generating inspection reports, risk ratings, and targeted remediation advice.



Consolidating findings into reusable Inspection Agents.

Converting System Knowledge into On-Demand Expert Capabilities

Optimization, O&M, and inspection rely on vast amounts of tacit knowledge. Agents and Copilots make this knowledge explicit, tool-ready, and reusable, scaling platform capabilities beyond a few experts.

```
2023-04-22 13:53:33.316 ERROR [HTPO]:error=3518063-unareevor:2022.21
2023-04-22 13:53:33.328 ERROR [NSPO]:error=732689, logn Gsarxs
2023-04-22 13:53:33.538 ERROR [REPOK]:error=732689, logn Gsarxs
2023-04-22 13:58:33.658 ERROR [KFO9]:error=732689, node "db-63
2023-04-22 13:58:33.533 ERROR [HTPO]:error=7836021-unareevor:2022.21
2023-04-22 13:58:33.323 ERROR [KFO9]:error=732689, logn Gsarxs
2023-04-22 13:58:33.323 ERROR [RSDQ]:error=732689, logn Gsarxs
2023-04-22 13:58:33.516 ERROR [HTPO]:error=2528851-unareevor:2022.21
2023-04-22 13:58:33.533 ERROR [KFO9]:error=732689, node "db-63
2023-04-22 13:53:33.240 ERROR [RAOD]:error=2528851-unareevor:2022.21
2023-04-22 13:58:33.366 ERROR [KFO9]:error=732689, node "db-63
2023-04-22 13:58:33.933 ERROR [KFO9]:error=732689, node "db-63
2023-04-22 13:58:33.836 ERROR [HTCM]:error=732689, node "db-63
2023-04-22 13:58:33.822 ERROR [KFO9]:error=732689, node "db-63
2023-04-22 13:53:32.807 ERROR [KFO9]:error=732689, node "db-63
2023-04-22 13:58:33.909 ERROR [KFO9]:error=732689, node "db-63
2023-04-22 13:58:33.826 ERROR [KFO9]:error=732689, node "db-63
2023-04-22 13:53:33.873 ERROR [KFO9]:error=732689, node "db-63
```

> Interactive Terminal

Output: Based on log analysis, the query latency spike is due to high lock contention on table 'orders' and resource exhaustion on node 'db-03'.

Recommended action: Optimize query plans and consider scaling node db-03.

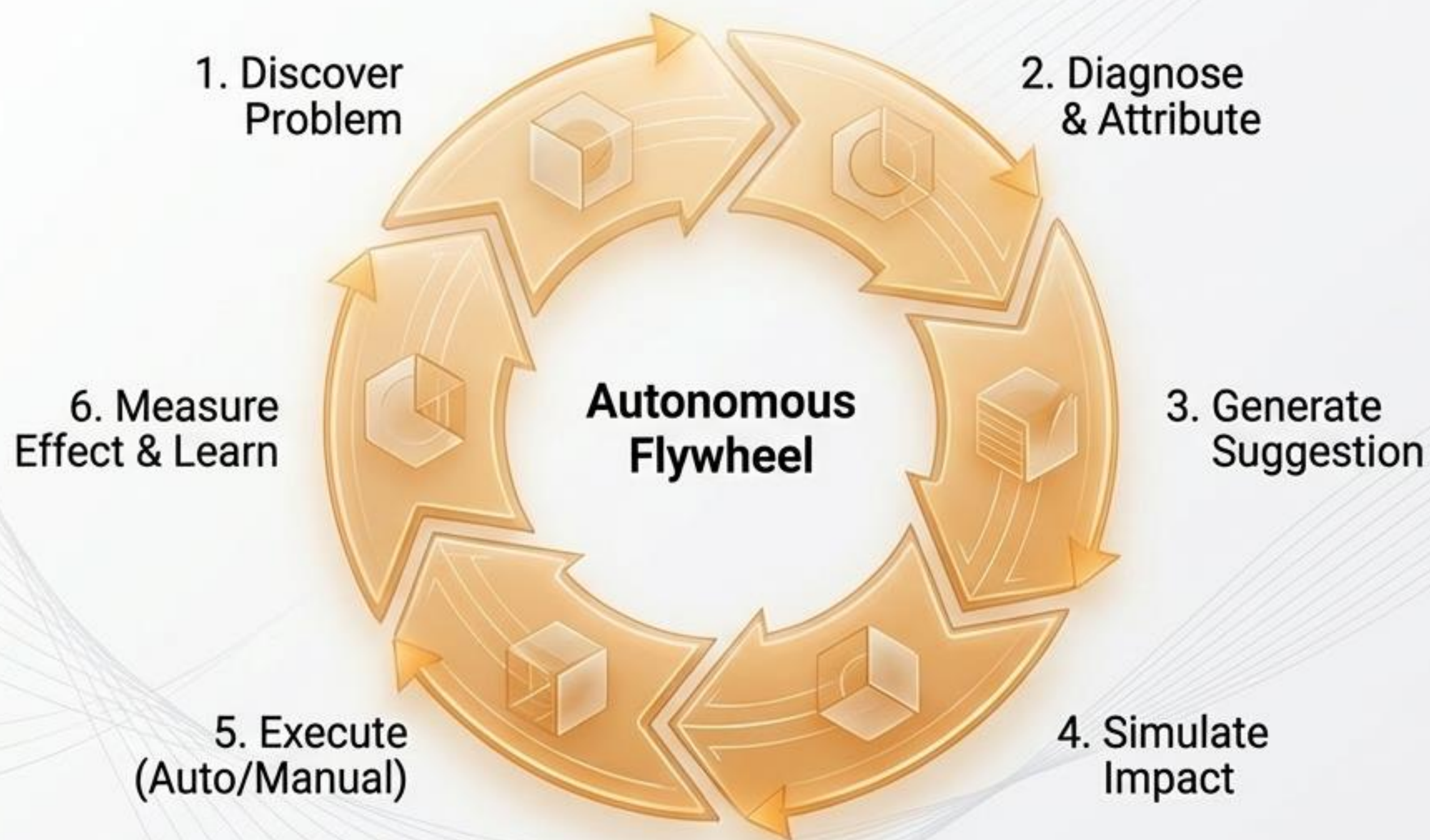
```
{
  "name": "input",
  "type": "string",
  "values": [
    {
      "key": "string",
      "values": [
        {
          "key": "array"
        }
      ]
    }
  ],
  "type": "query",
  "key": "array"
},
{
  "name": "output",
  "type": "string",
  "values": [
    {
      "key": "string",
      "values": [
        {
          "key": "array"
        }
      ]
    }
  ],
  "type": "query",
  "key": "array"
}
}
```

Key Capabilities

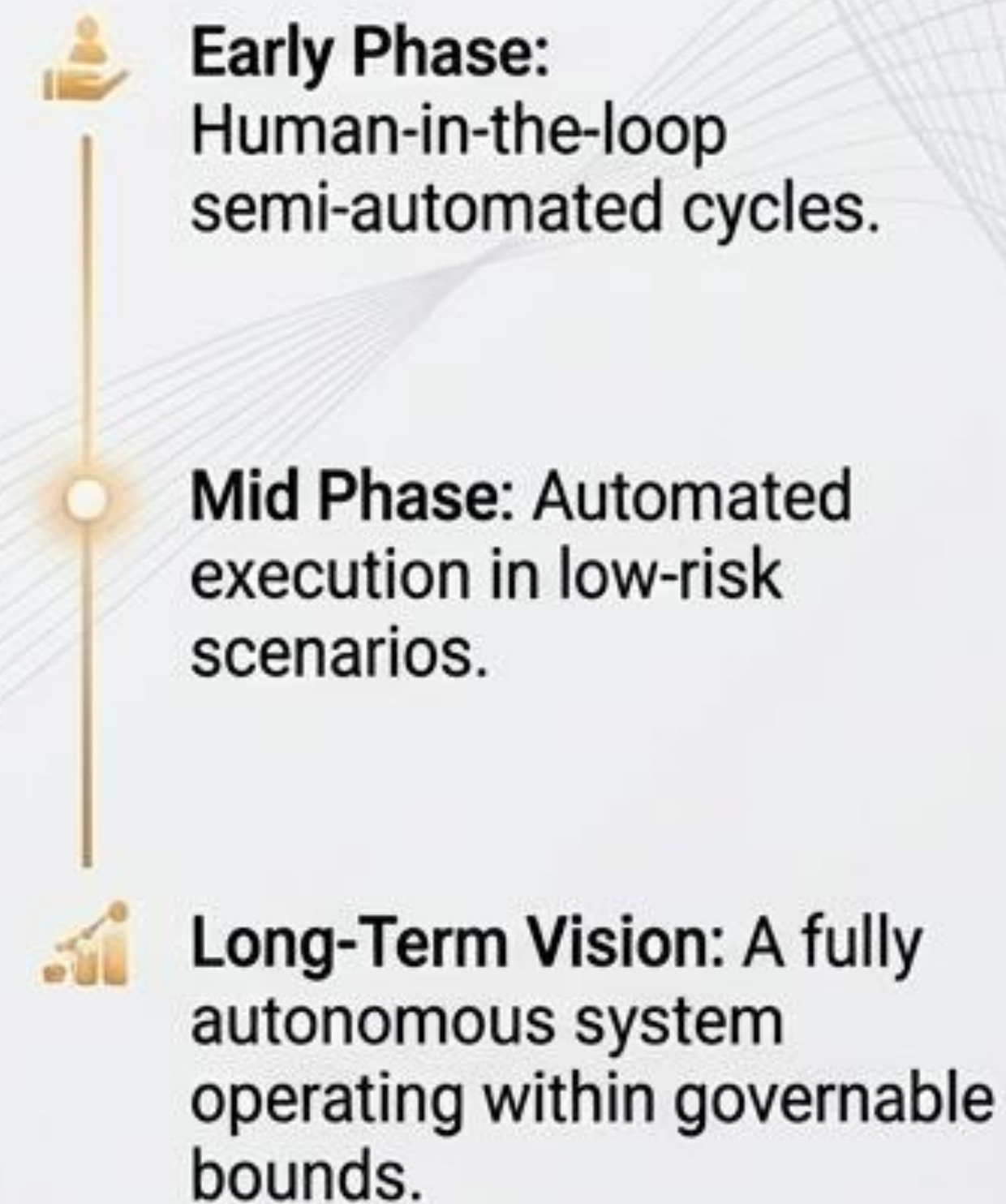
- A dedicated Knowledge QA Assistant for DBAs, developers, and operations.
- Unifying query logs, system metrics, alerts, and docs into a single interactive layer.
- Drastically lowering the barrier to entry for troubleshooting.
- Transforming individual expert best practices into platform-wide capabilities.

The Ultimate Goal is an Autonomous Closed Loop, Not Just Advice

True intelligence isn't just a system offering a recommendation. It is a system forming an explainable, controllable, traceable, and continuously learning closed loop.

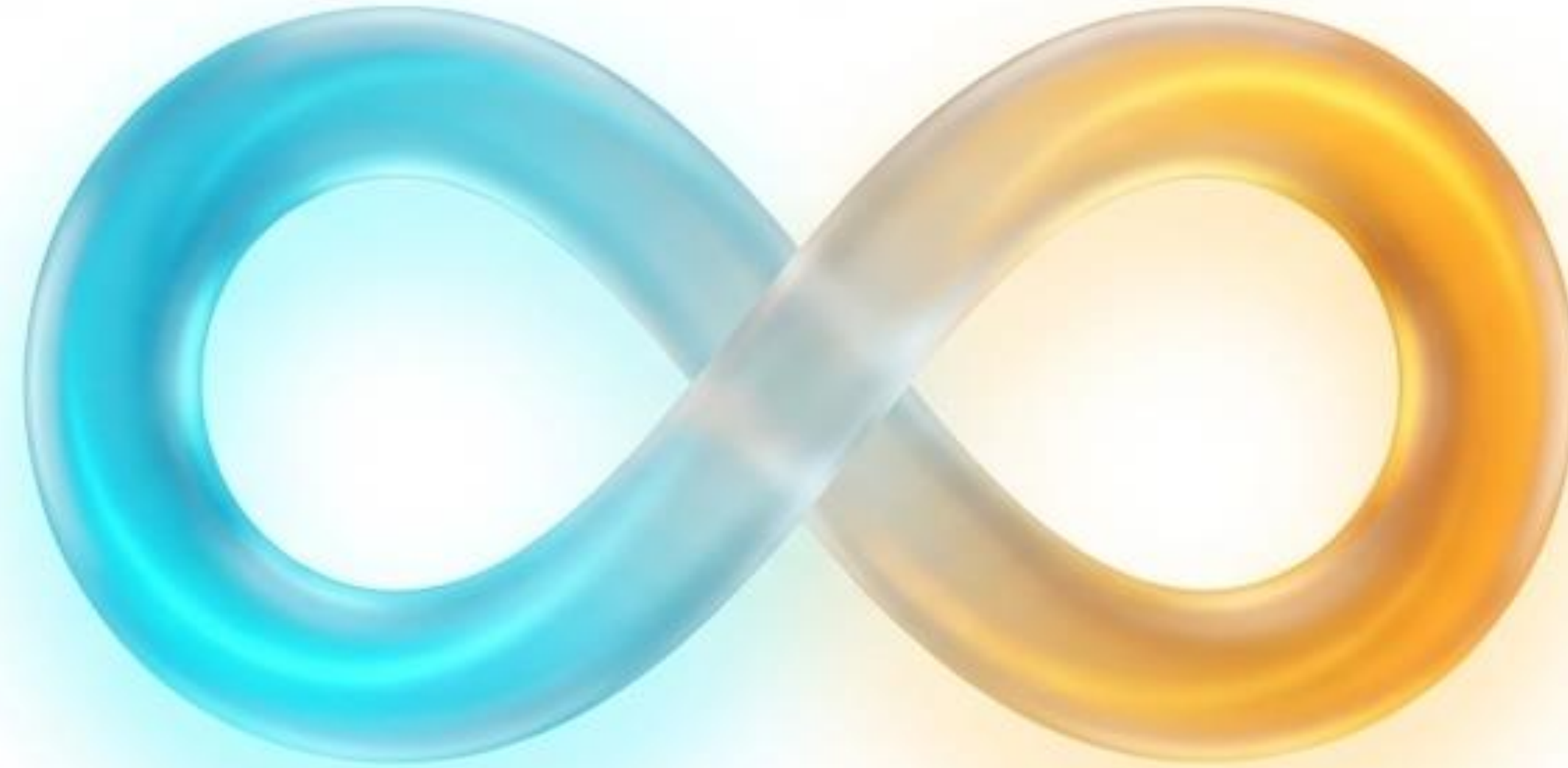


Evolutionary Path



Next-Generation StarRocks: Not Just Faster, But Smarter

Outwardly: Serving as the definitive real-time data base for AI and Agents.



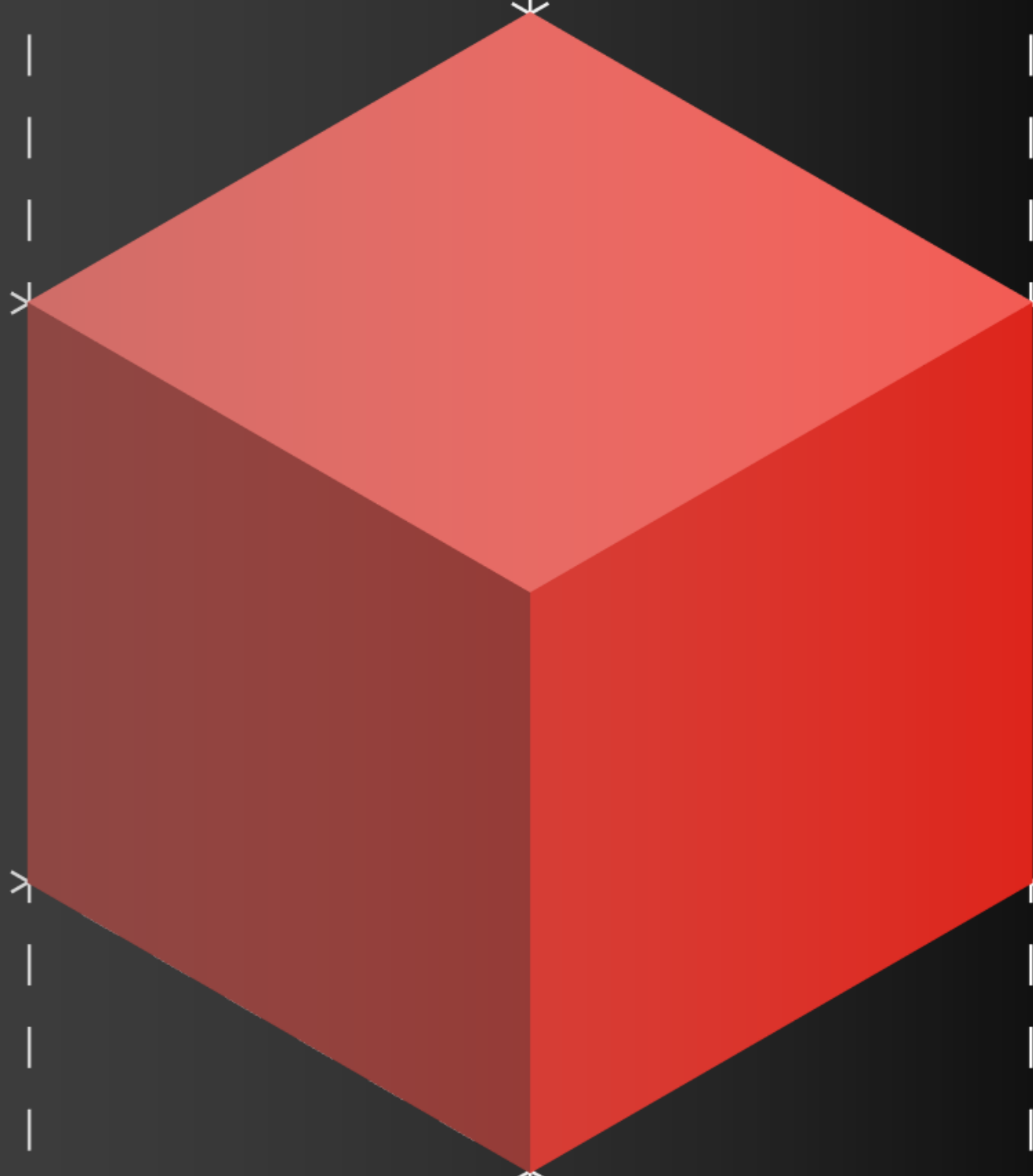
Inwardly: Leveraging Agents to achieve autonomous optimization, operations, and inspection.

In the past, real-time analytics solved the problem of “seeing fast.” Today, it must solve the problem of “understanding, judging, tuning, and acting fast.”

The future of StarRocks is not just to serve AI—it is to grow into a real-time intelligent analytical system continuously driven by AI.

8个月做到 1 亿 ARR

AI Agent 时代的数据底座怎么建？



TiDB 全球化脚步

PingCAP

- TiDB : 开源, 分布式, 云原生 数据库
- 2015 年 : 公司成立並启动 TiDB 开源项目
- 2017 年 : TiDB GA V1.0
- 2025: 4,000+ 全球客户



获得全球化合规认证 —— US/ EU/ JP/ KR/ IN

<https://www.pingcap.com/trust-hub/>

2020/07

SOC2 Type I and II



2021/06

Start planning ePrivacy Seal in September 2020 and pass it in June 2021



2021/07

Internationally recognized information security management system



2022/05

Personal Information Management System



2022/09

Financial Industry Information Security Standards



2023/03

Personal Health Data Protection Regulations



三个真实的客户故事诠释范式转移

Manus

数据库成本威胁到原生智能体应用的产品模型。

成本

Plaud

长上下文和媒体数据导致存储架构严重碎片化。

架构复杂度

某头部基础大模型公司

数据分片支撑了流量规模，但运维负担过重。

运维可扩展性

这三个故事共同解释了为什么 **AI 原生产品** 需要一种 **全新的数据库**。

AI 应用极少在 Demo 阶段失败

Demo 表现惊艳且体验极佳

业务应用极少在
早期阶段失败。

早期用户体验到工
作流的“魔法感”。

模型运行完美无瑕。

业务增长触发底层数
据架构的突然崩溃。

隐藏的增长天花板

当数据库设计从根本上与产品模型脱节时，转折点就会出现。

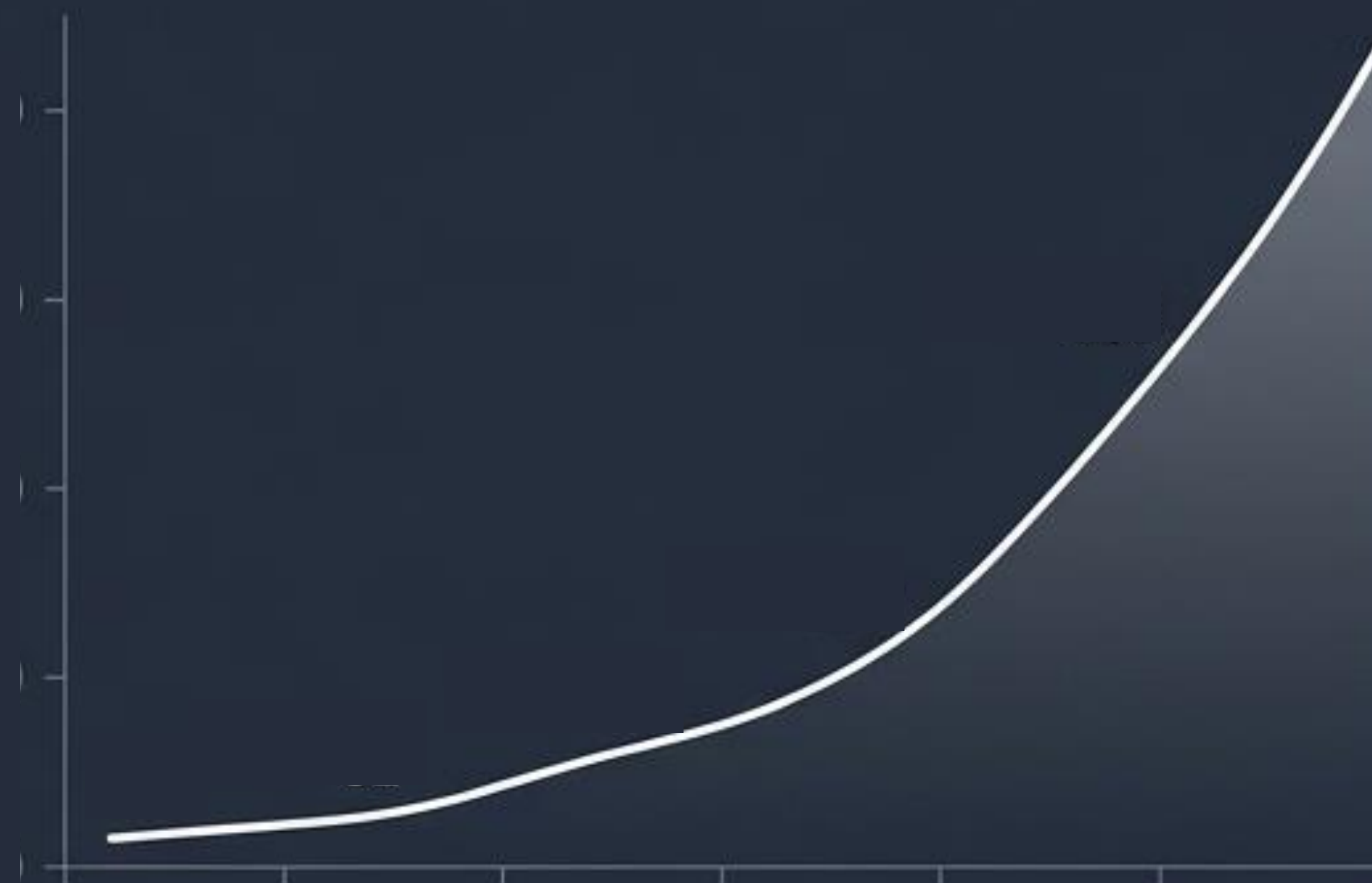


manus

迅速获得市场关注与强劲势头



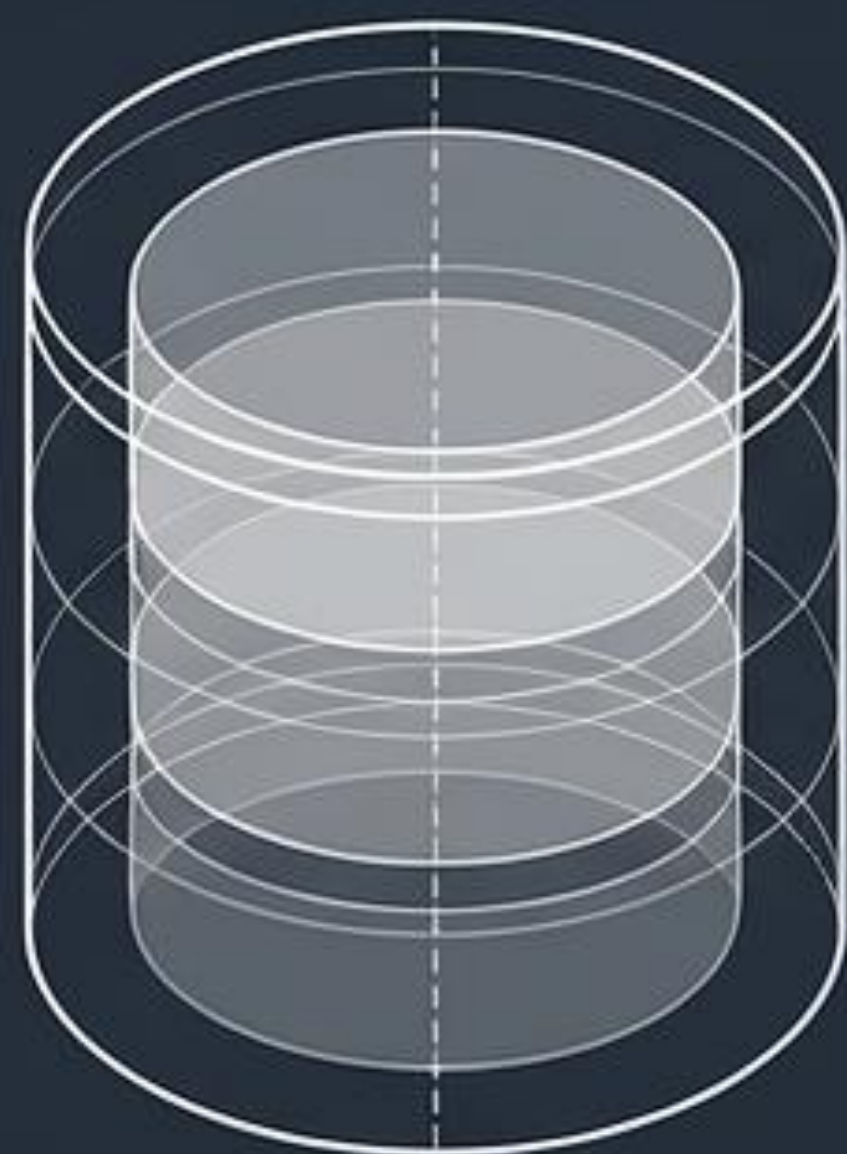
早期市场牵引力与增长



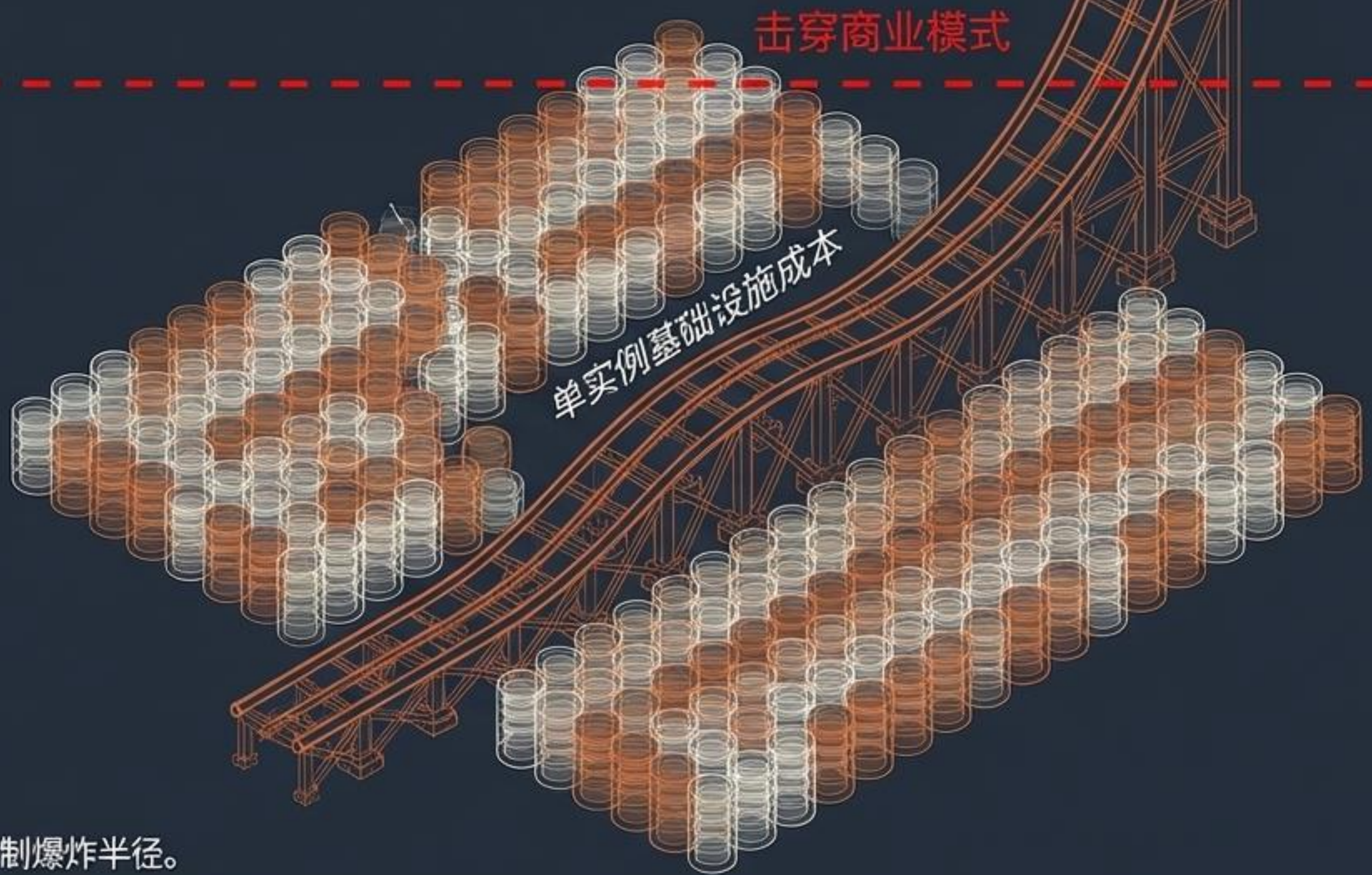
核心洞察：Manus 面临的挑战从来不是产品不够好，市场反响空前热烈。

关键转折点：产品规模化需要从根本上重新思考数据库模型。真正的考验在于架构能否支撑业务的规模化爆发。

传统成本模型面临击穿产品形态的风险



1 个智能体工作空间



Manus 需要细粒度的隔离和独立的 Schema 来控制爆炸半径。

分析任务需要结构化存储和可编程的 SQL 查询，而不仅仅是原始文本。

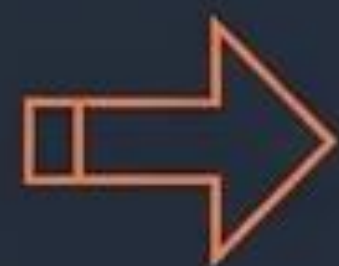
将传统的隔离实例模式映射到数十万的量级，在经济上会让最初的产品愿景变得极度不现实。

与众不同：一个会话几乎等同于一个智能体

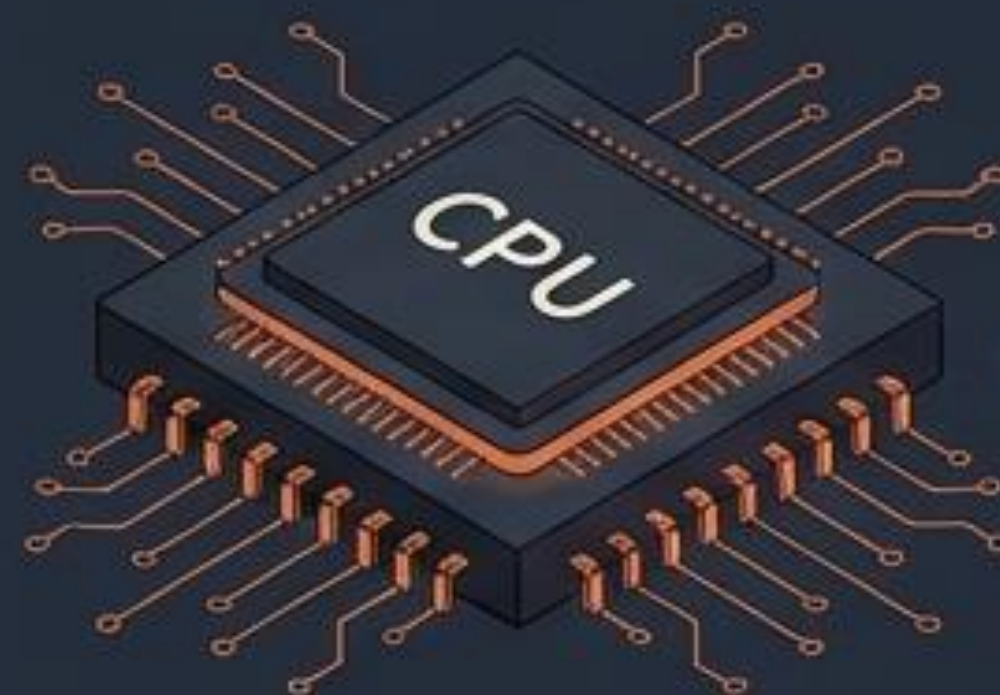


不仅仅是聊天记录

行为表现为执行单元，而
而不仅仅是对话日志。



执行单元与工作环境



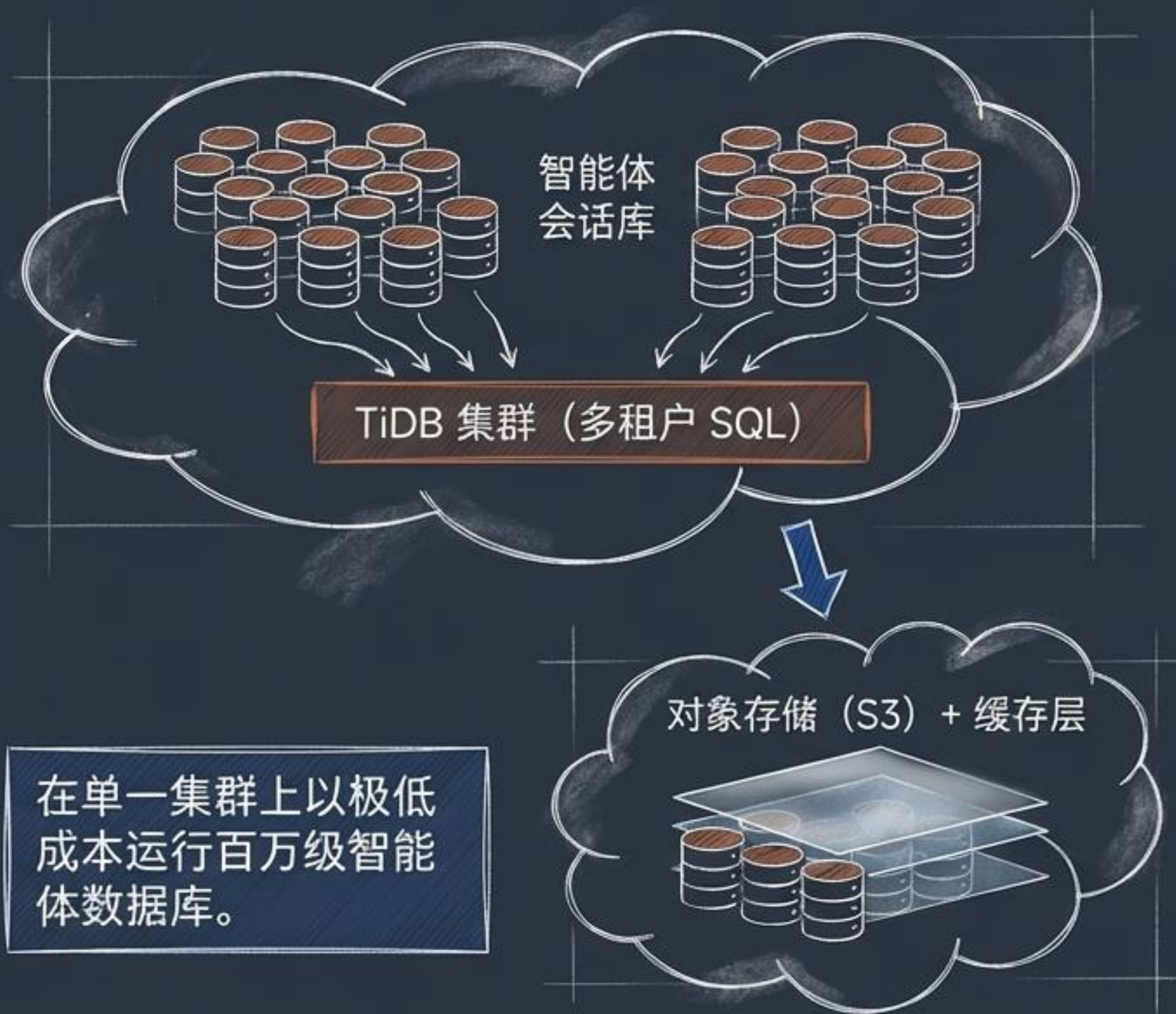
状态、工作流与输出



专属数据空间与
动态演进的 Schema

数据库成为了智能体工作环境的一部分。
需要具备动态 Schema 行为的隔离数据空间。

Manus: 以极低成本实现海量智能体数据库



① 百万级逻辑数据库

- 1 个 TiDB 集群
- 百万级逻辑库
- Per-agent / Per-session 隔离

② 让 AI 业务可行的成本模型

- 传统架构: 1 智能体对应 1 数据库实例
- 成本降低 10 倍

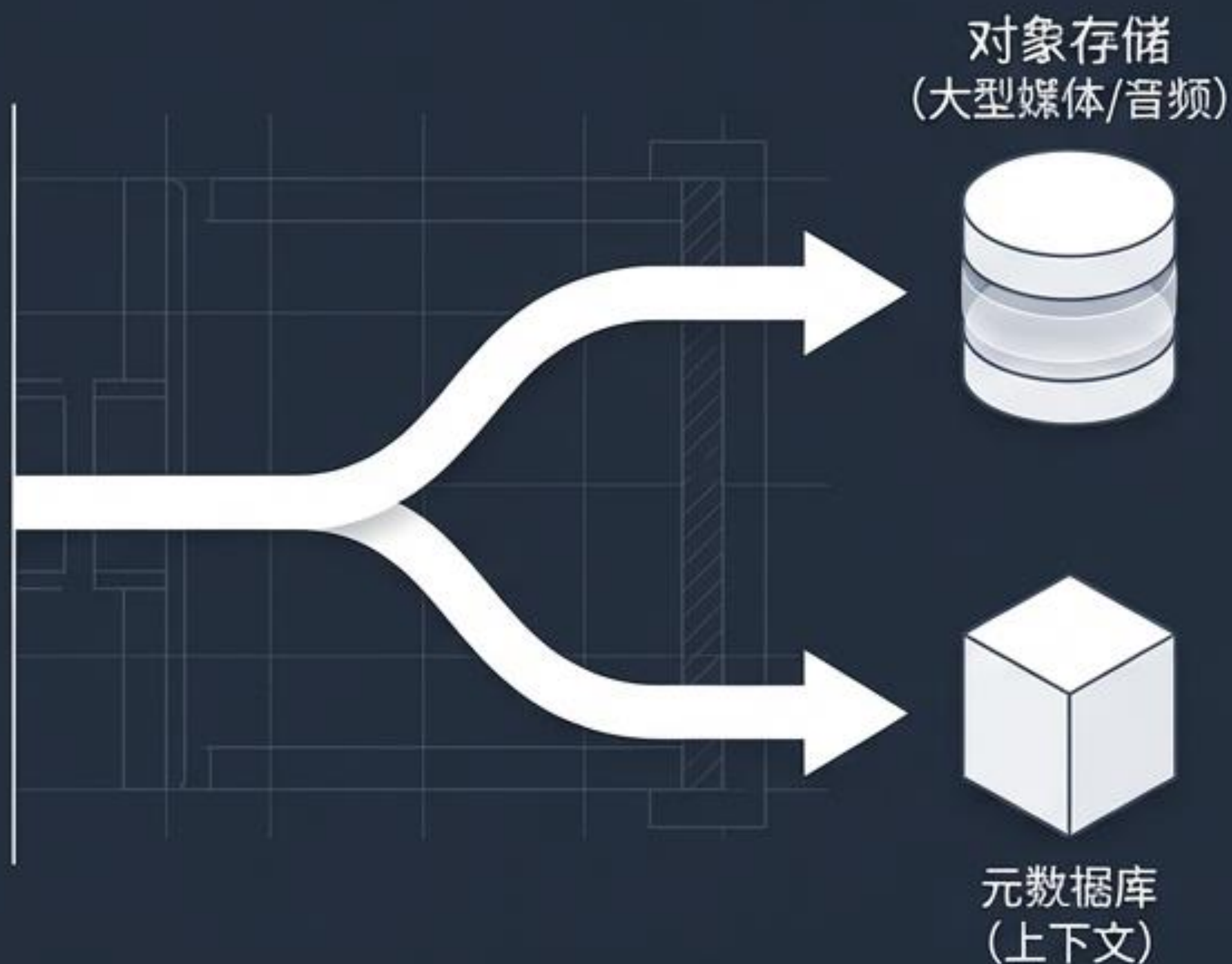
③ 资源管控

- 资源控制
- 工作负载隔离
- 多租户保护

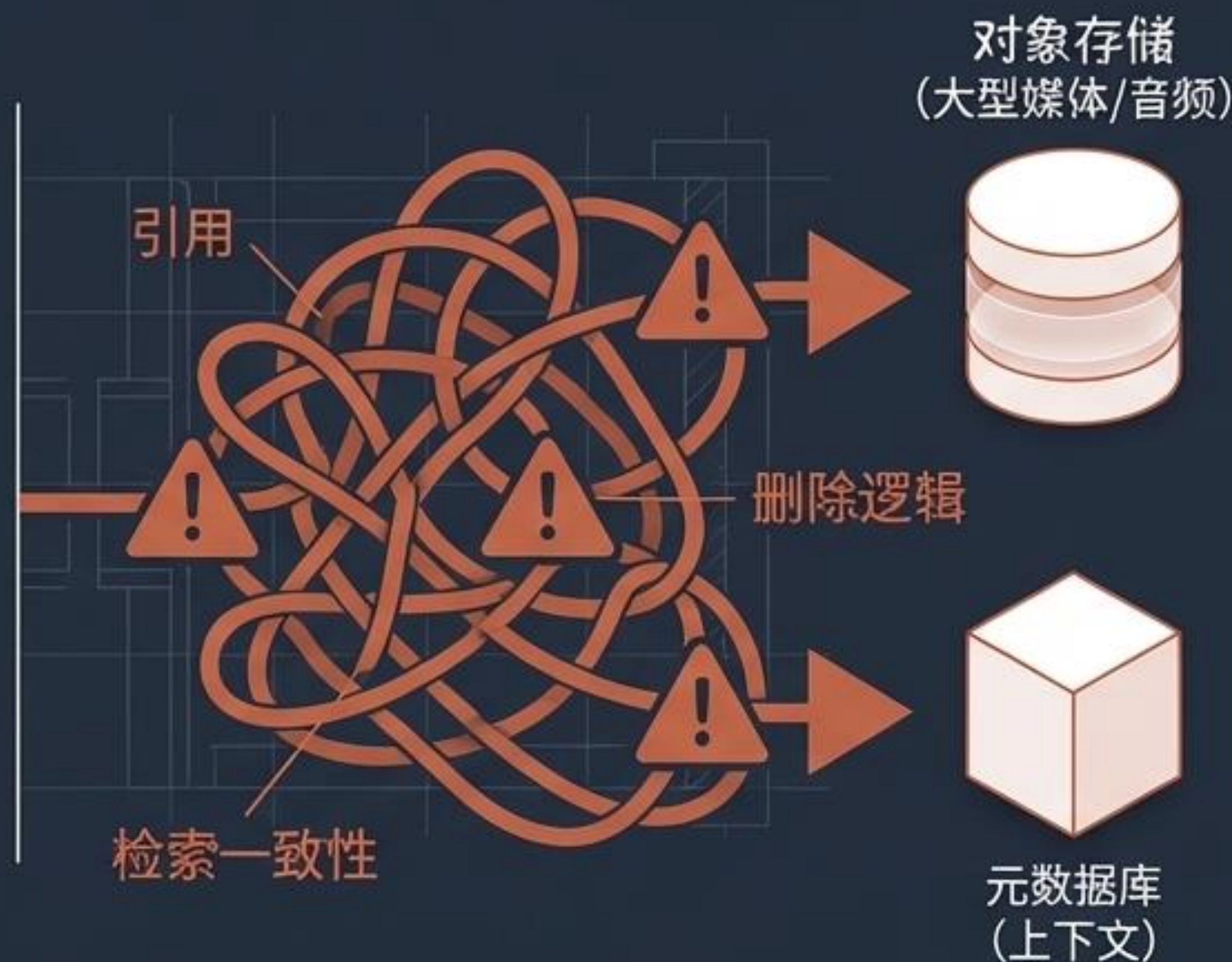


PLAUD AI: 长上下文管理——直到数据碎片化接管系统

初期成功

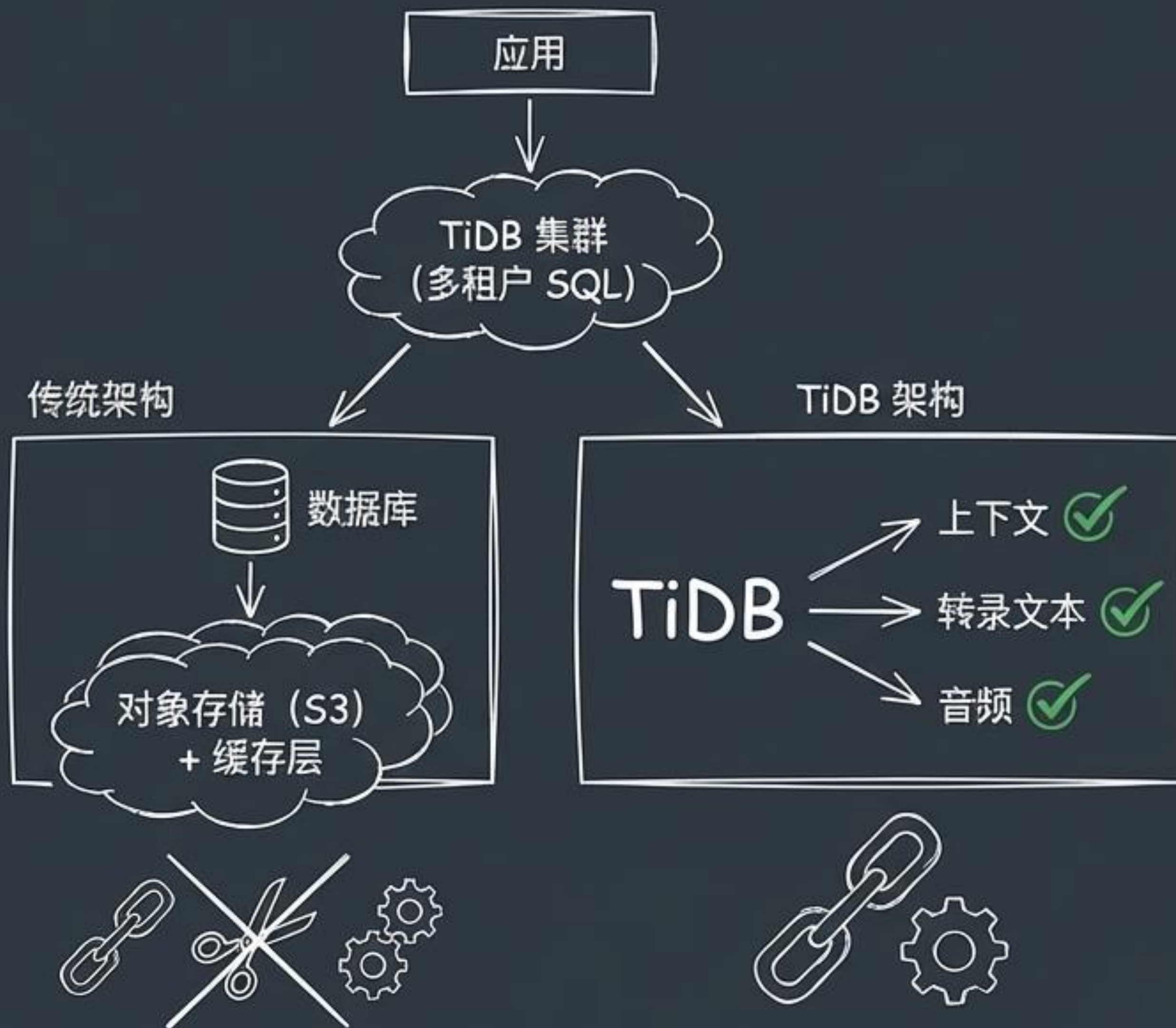


隐藏的高墙



Plaud 的初始策略很奏效：将大型音频/媒体数据与元数据分离。但随着管理这种分离状态的架构越来越难以保持简单、快速和一致，隐藏的瓶颈显现了。

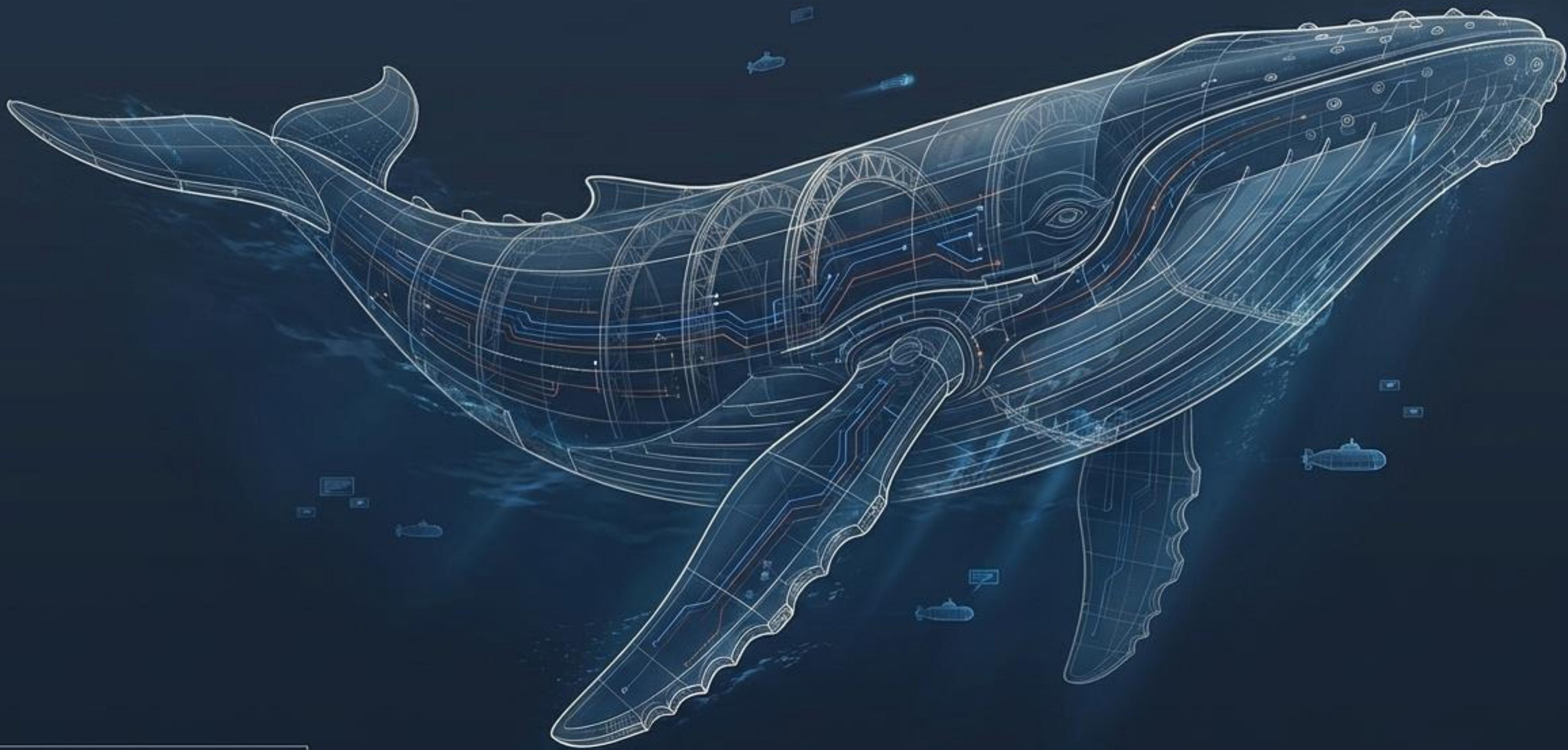
Plaud: 长上下文与多媒体存储



- ① 强一致性
→ ACID 事务保障
- ② 可查询性
→ SQL + Python 结合，直接分析 AI 上下文
- ③ 架构极简
→ 彻底告别“数据库+S3+缓存+同步逻辑”，现在仅需 TiDB

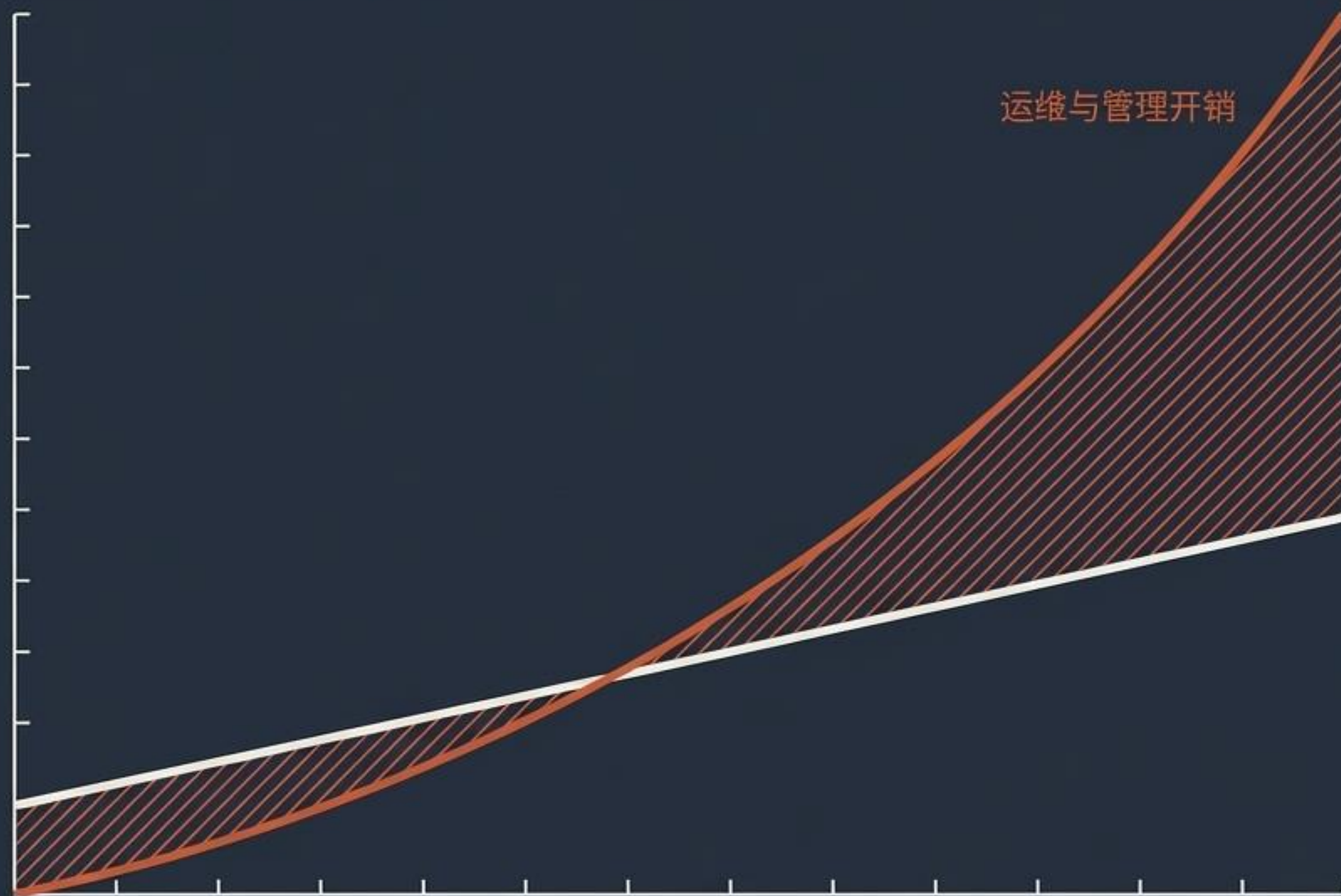
单字段最高 100M

将超长 AI 上下文直接存储在数据库中，同时保留 SQL 查询和事务能力。



某头部基础大模型公司

成功扩张后，打破了原有的运维模型



系统性能 (分片 PostgreSQL)

最初的分片 PostgreSQL 设计非常有效且技术上健全。

但成功带来了新的负担：随着分片数量的增加，运维复杂度、协调摩擦和手动管理工作量呈指数级飙升。

从“无限分片”转向统一的分布式系统

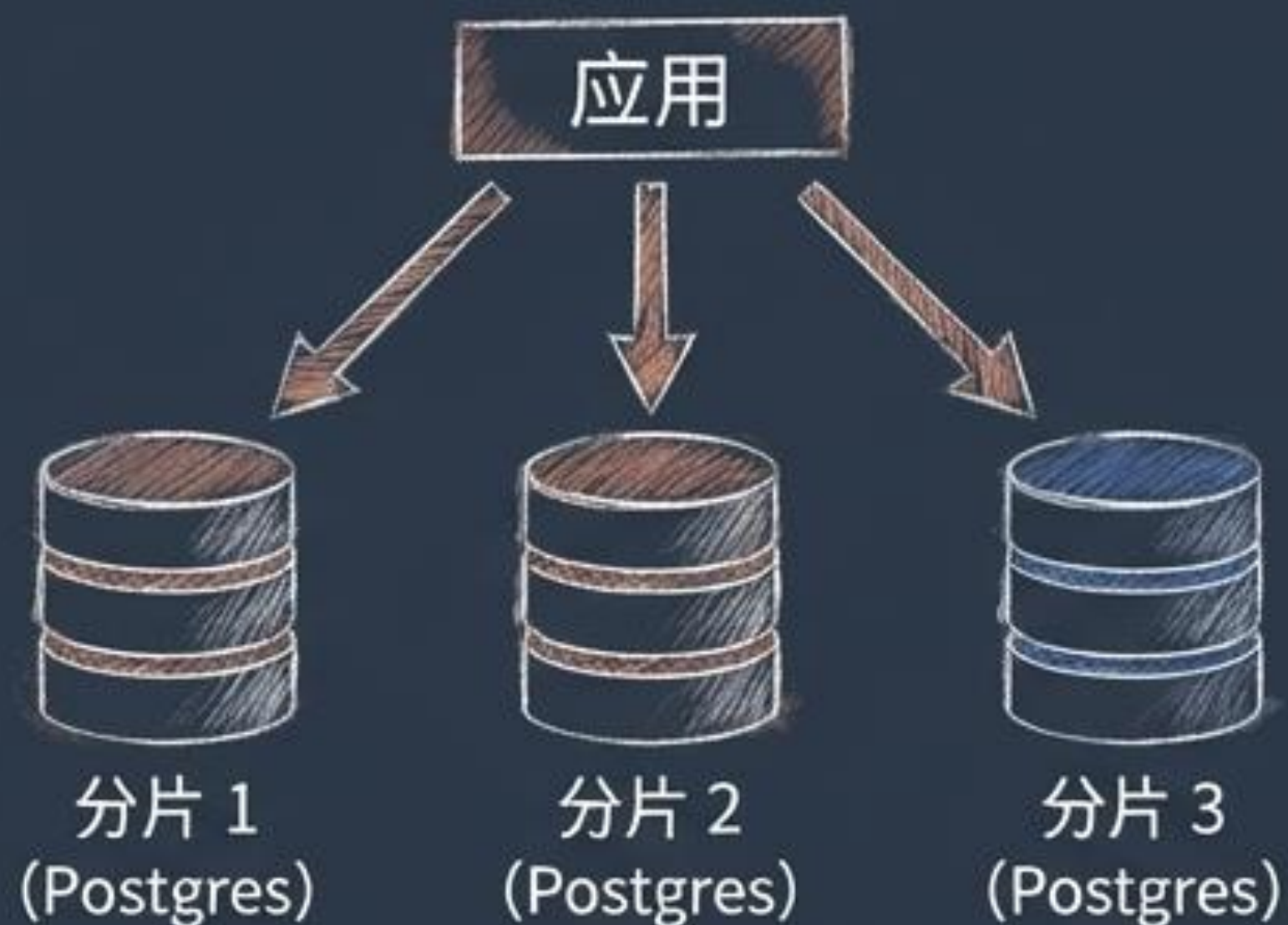


彻底消除手动分片协调和高昂的运维开销。

AI 产品需要能够优雅扩展的系统，而不是压垮管理它们的工程师团队。

某头部基础大模型公司：消除分库分表复杂度

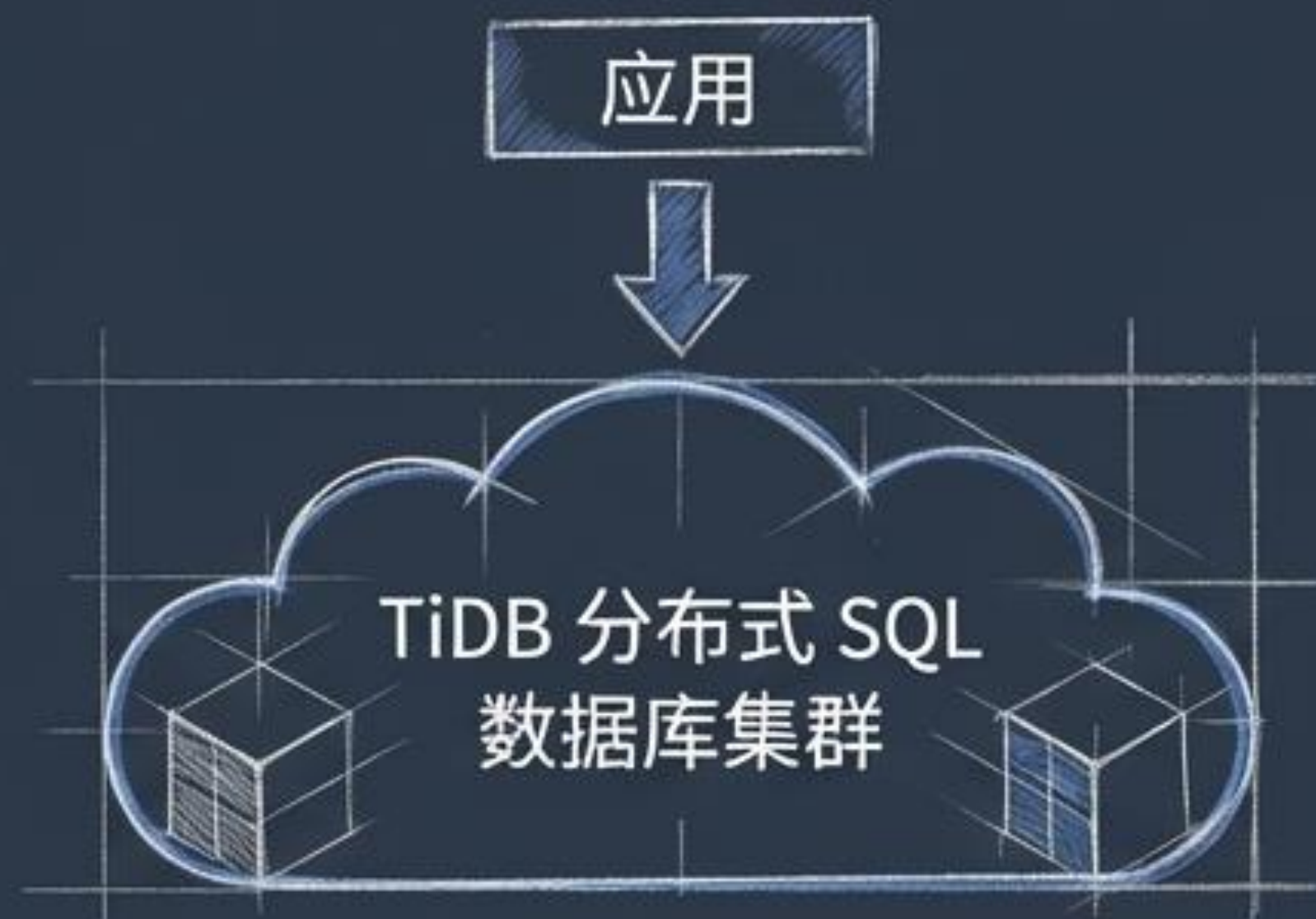
传统架构



运维挑战

- ✗ 分片路由
- ✗ 重新平衡
- ✗ Schema 同步
- ✗ 运维开销

TiDB 架构



TiDB 自动处理

- ✓ 数据分片
- ✓ 重新平衡
- ✓ 数据复制
- ✓ 弹性扩缩容

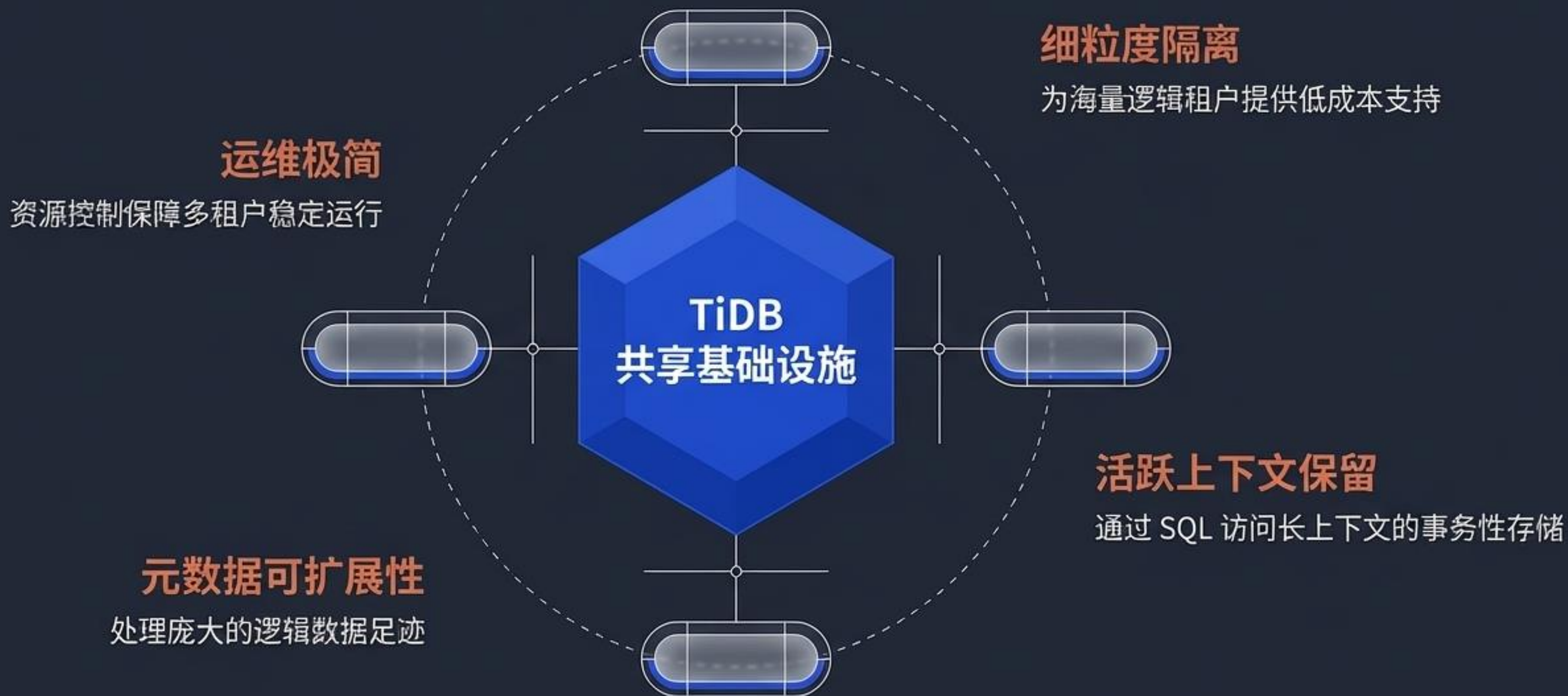
停止管理分片。让数据库自动扩展。

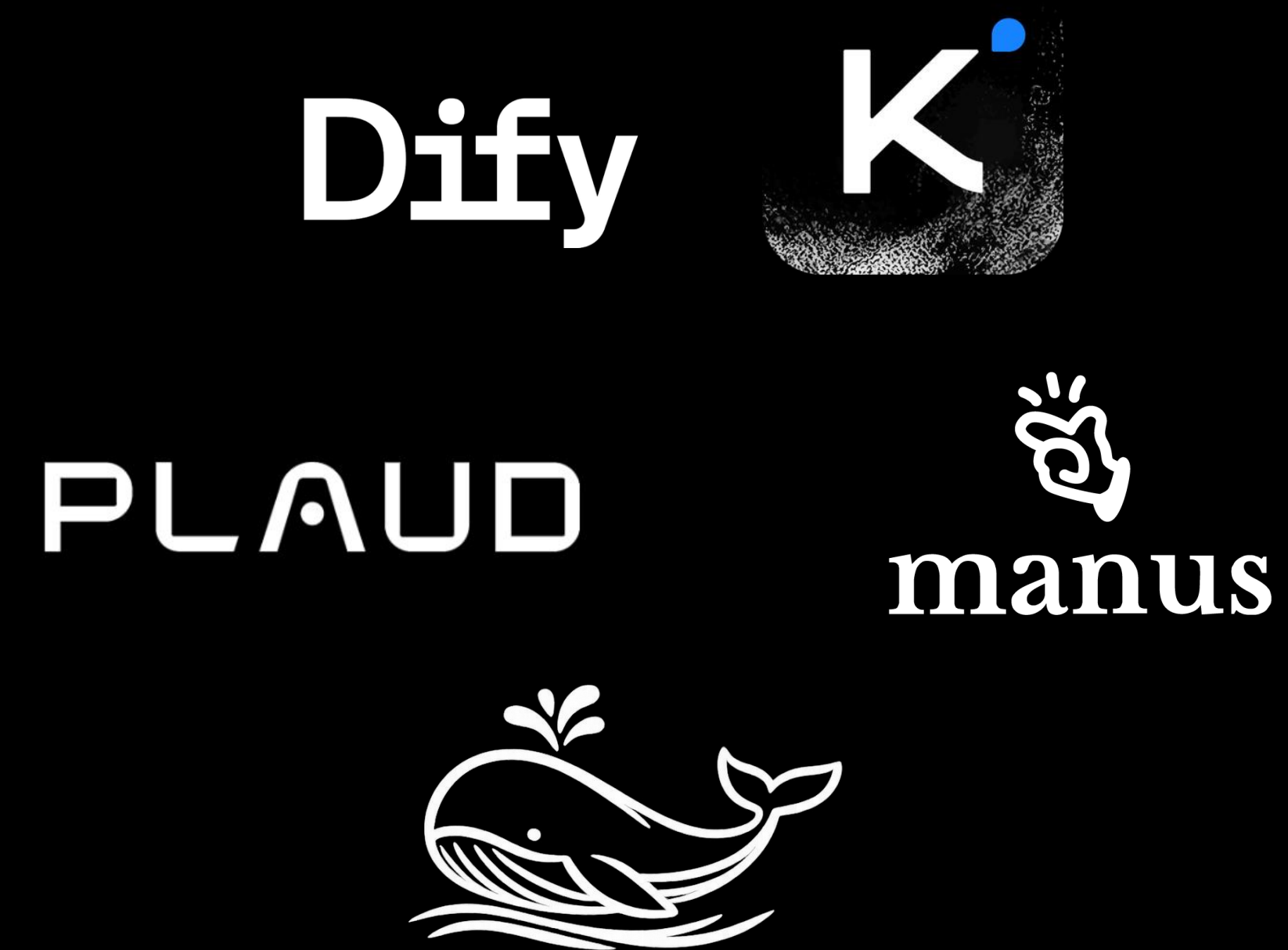
三种截然不同的扩展路径，一个共同的底层范式转移

	AI 增长向量	隐藏的断点	TiDB 破局之道
Manus	智能体 工作空间 	实例成本击穿 商业模式 	共享架构上的 海量逻辑库 
Plaud	活跃长上下文 	数据碎片化 与缓存灾难 	统一的关系型 操作流 
某头部基础 大模型公司	海量并发交易 	分片运维吞噬 研发团队 	统一的分布式 SQL 

AI 原生应用需要的数据库必须同时具备成本效益、架构极致简单和运维高度可扩展。

AI 原生数据基础设施的新形态





MEM9.AI

无限记忆 for OpenClaw

你的 Agent 会在每次会话结束后忘掉一切，mem9 负责修复这件事。它提供持久化记忆基础设施，支持混合搜索、共享空间和跨 Agent 召回，从第一次写入一直保留到未来。

如何安装

阅读 <https://mem9.ai/SKILL.md> ，按照说明为 OpenClaw 安装并配置 mem9



把上面这条命令复制给 OpenClaw 即可完成安装，按提示操作会自动生成 API Key，无需注册申请。

不再遗忘

云端持久记忆可跨越重置、重启和设备切换持续保留。

安全备份

你的 Agent 记忆存放在耐久云存储里，而不是脆弱的本地文件。

无缝接入

从一条指令开始，再逐步迁移已有记忆，不会打断现有工作流。

Network drive with semantic search_

A unified filesystem-like interface for AI agents. Store, retrieve, and search with automatic embedding and full-text indexing. Use familiar commands like a local drive.

CLI



```
$ curl -fsSL https://drive9.ai/install.sh | sh
```

macOS / Linux (x86_64, arm64)

AI Agents

Read <https://drive9.ai/skill.md> and follow instructions

Your agent learns to install, auth, and use drive9 autonomously

-- VECTOR SEARCH ENABLED --

Database Built for Agents.

Zero sign-up, zero config.
Provision a disposable MySQL-compatible database for AI agents, MCP servers, and RAG prototypes with vector and full-text search in milliseconds.

TRY IN BROWSER

CLI QUICKSTART

FOR HUMAN

FOR AGENT

COPY

```
# For Human
curl -X POST https://zero.tidbapi.com/v1beta1/instances
```

```
Method: Post
Endpoint: /v1beta1/instances
Auth: Bearer <your_api_key>
Expected: 200 OK
```

[*]

AI Ready

Vector + relational in one store.

[=]

Full MySQL

Joins, transactions, and standard SQL.

[~]

Disposable

30-day lifespan for demos and CI.

[>]

Zero

Provision in milliseconds, no cold starts.

POPULAR SEARCH JOURNEYS

AI Agent Memory Database >

Store agent state, user preferences, tool outputs, and searchable memory in one disposable SQL backend.

Serverless MySQL for RAG >

Use MySQL-compatible SQL with vector search and full-text search for retrieval-heavy prototypes.

Database for MCP Servers >

Provision a low-friction MySQL backend for MCP tools, protocol demos, and per-session state.

Temporary Database Workflows >

Browse demos, tutorials, evaluation flows, and API-first sandboxes built around disposable databases.

CLI QUICKSTART

Provision a temporary MySQL-compatible database via API, then connect and run one SQL query using curl, jq, and mysql.

01

Step 1. Create an Instance

```
curl -s -X POST https://zero.tidbapi.com/v1beta1/instances \
```

COPY

数据AI应用实战全记录

AskTable研发团队

4.18

让Agent懂你的企业数据库

察言观数· AI 引擎

AI 问数

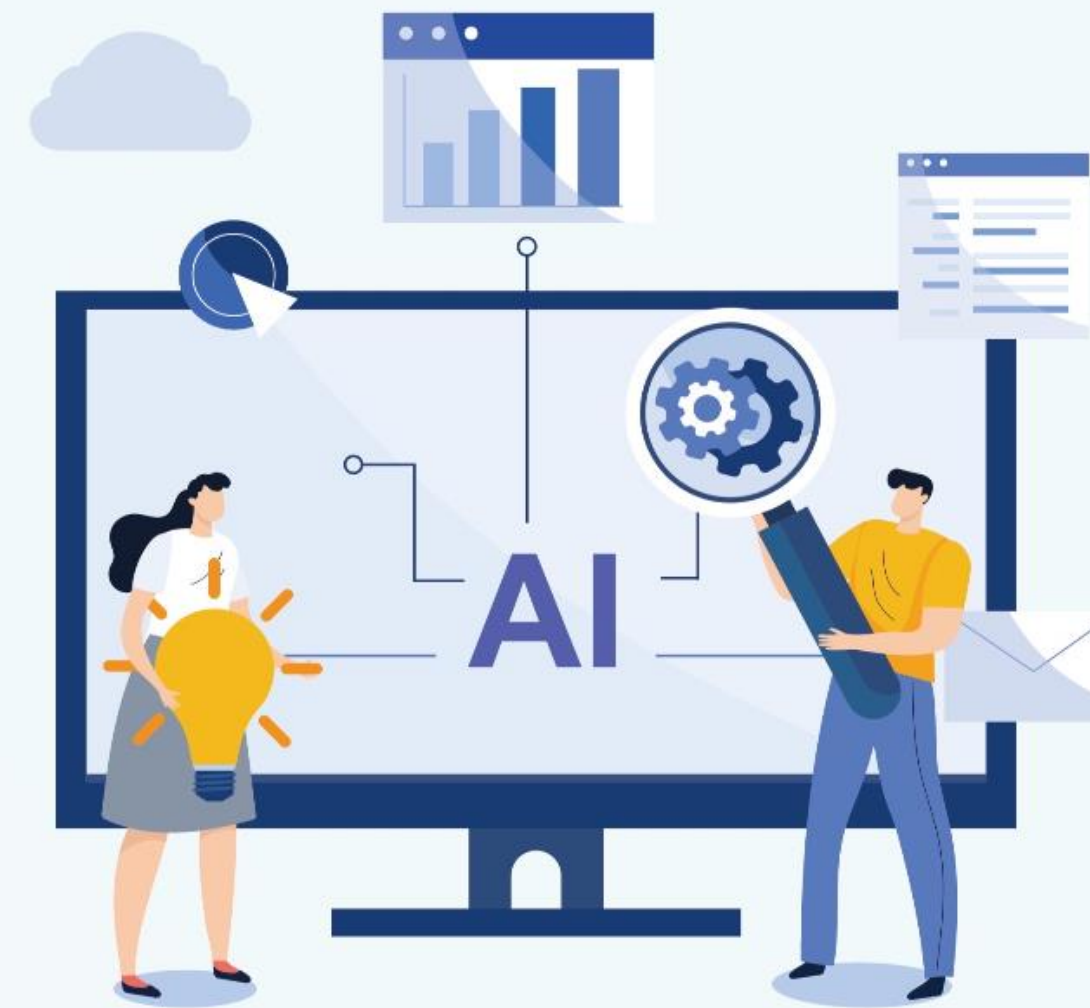
AI 智能体搭建

可被集成

企业专属的“数据”智能体构建平台

聚焦结构化数据，快速搭建可集成的 AI Agent

实现“问数即服务”的创新体验。



三个问题

数据分析领域Agent的现状？

AskTable如何思考和实践？

我是否真的需要Agent？

从开发的实践出发

Agent对AI数据应用的冲击

产品范围的取舍：两个月能做什么

核心技术路径：NL2SQL范式迁移，可视化和多形态

AskTable 的实践复盘与演进方向

RECAP

来自两年前最佳实践

NL2SQL
SFT RAG Dify
Prompt Engineering

NL2SQL easy?

Pre-Processing

- Schema Linking: 找到与问题最相关的表和字段
- Database Content Retrieval: 找到与问题有关的数据库中的数据
- Additional Information Acquisition: 与问题相关的业务知识

NL2SQL Translation Methods

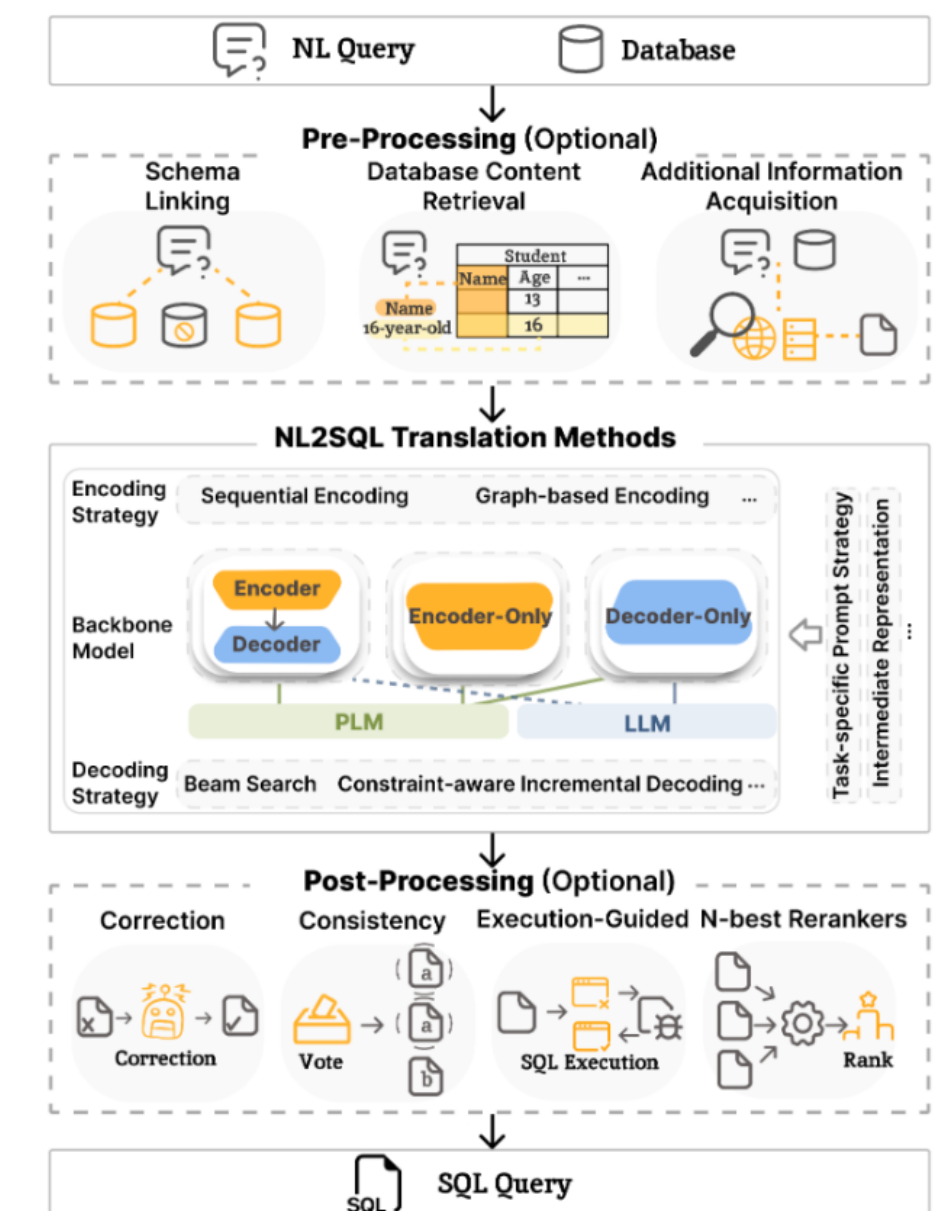
- Encoding Strategy
- Decoding Strategy
- Task-specific Prompt Strategy: 提示词工程
- Intermediate Representation for NL2SQL Translation

Post-Processing

- SQL Correction Strategies
- Output Consistency: 投票决定最终的结果
- Execution-Guided Strategies: 以执行的结果作为反馈修正sql
- N-best Rankers Strategies

AskTable

AskTable



<http://arxiv.org/abs/2408.05109>

RECAP

来自两年前最佳实践

NL2SQL
SFT RAG Dify
Prompt Engineering

NL2SQL easy?

Pre-Processing
- Schema Linking: 找到与问题最相关的表

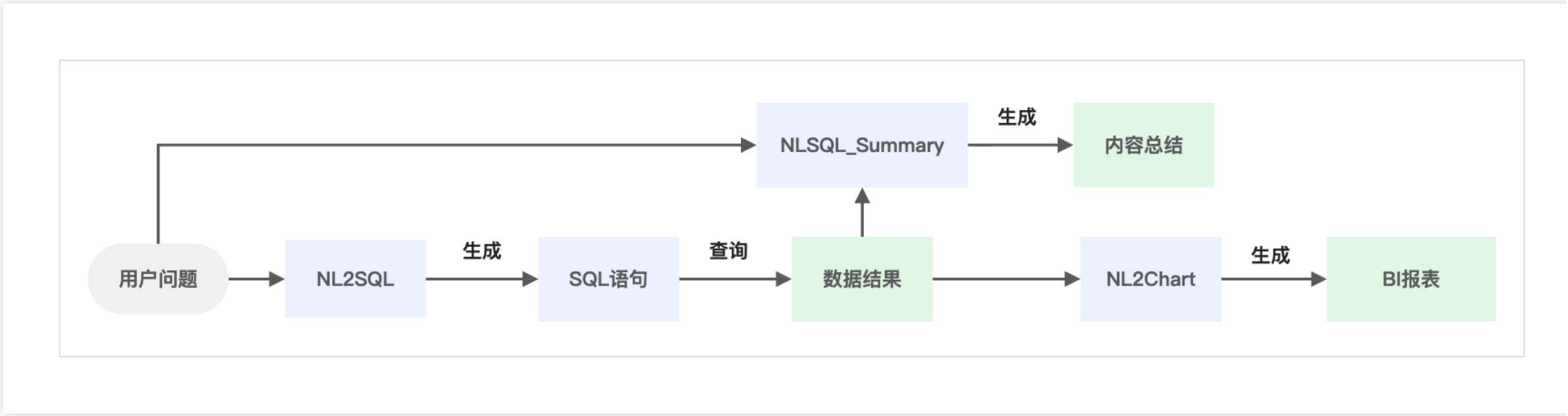
Data-Agent

更新时间：2025-12-29 15:23:52



Data-Agent 是 PolarDB for AI 所提供的一整套数据智能处理能力，涵盖从提问、数据处理到洞察输出的全链路功能。

功能简介



RECAP

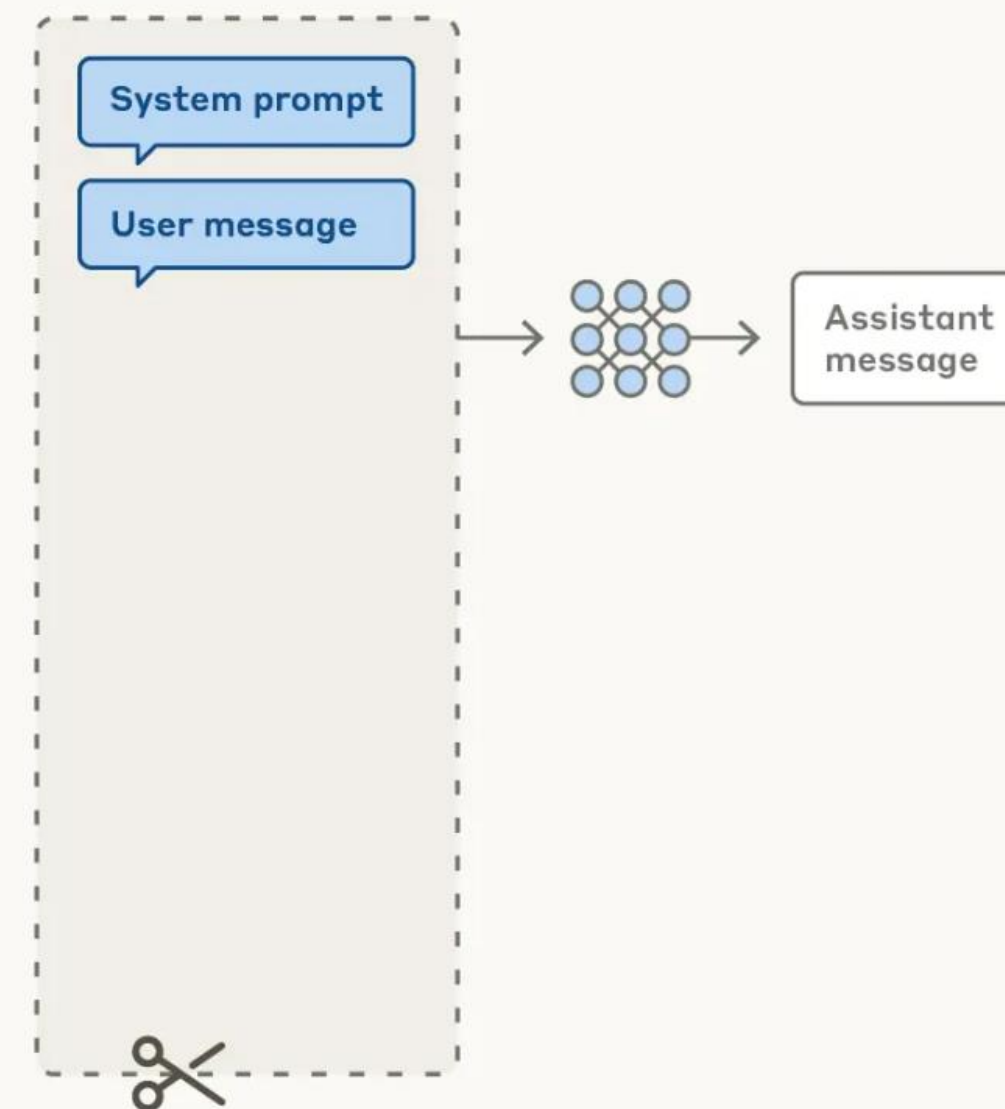
去年迎来了Agent

RLHF
Agentic Loop
Context Engineering

Prompt engineering vs. context engineering

Prompt engineering for single turn queries

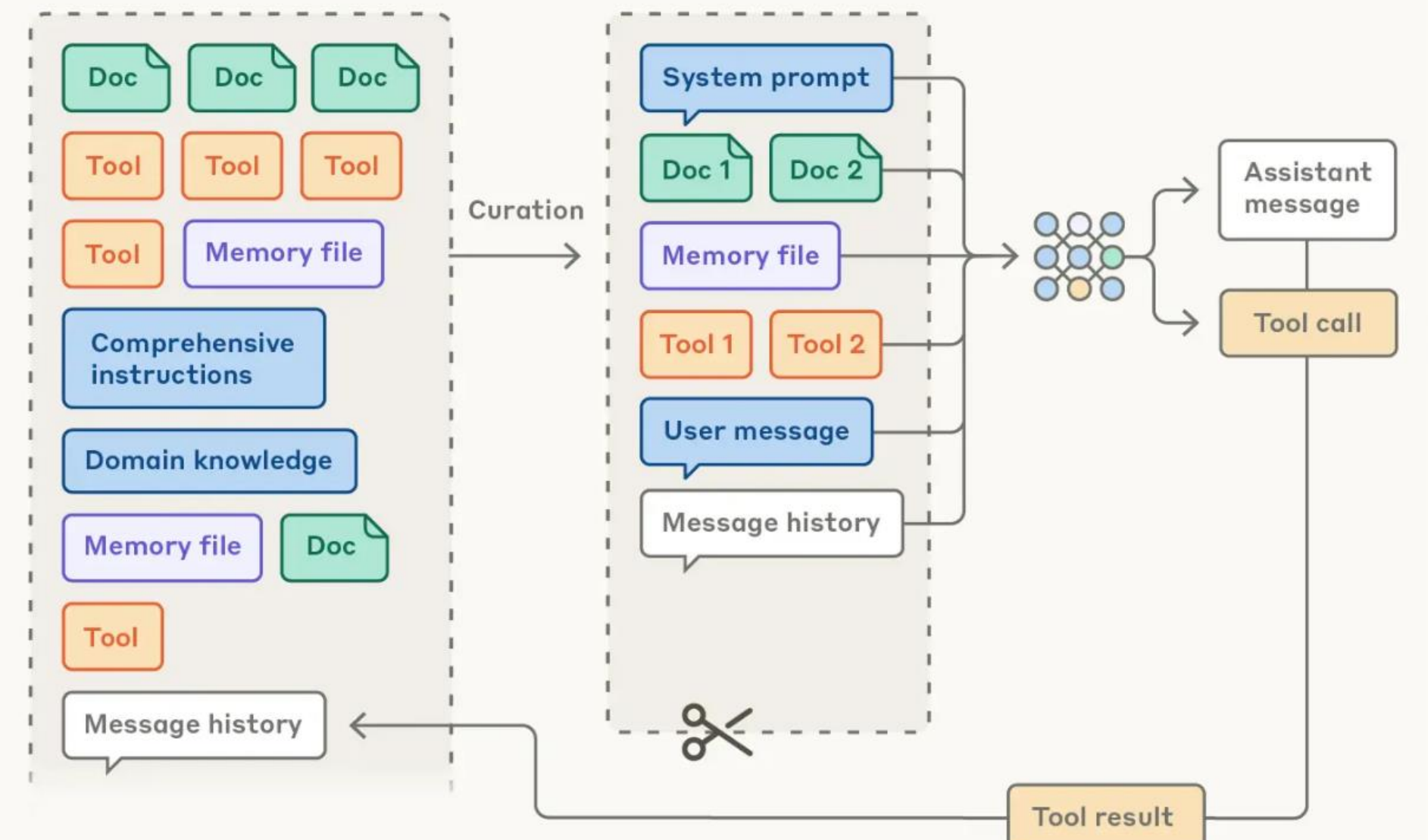
Context window



Context engineering for agents

Possible context to give model

Context window

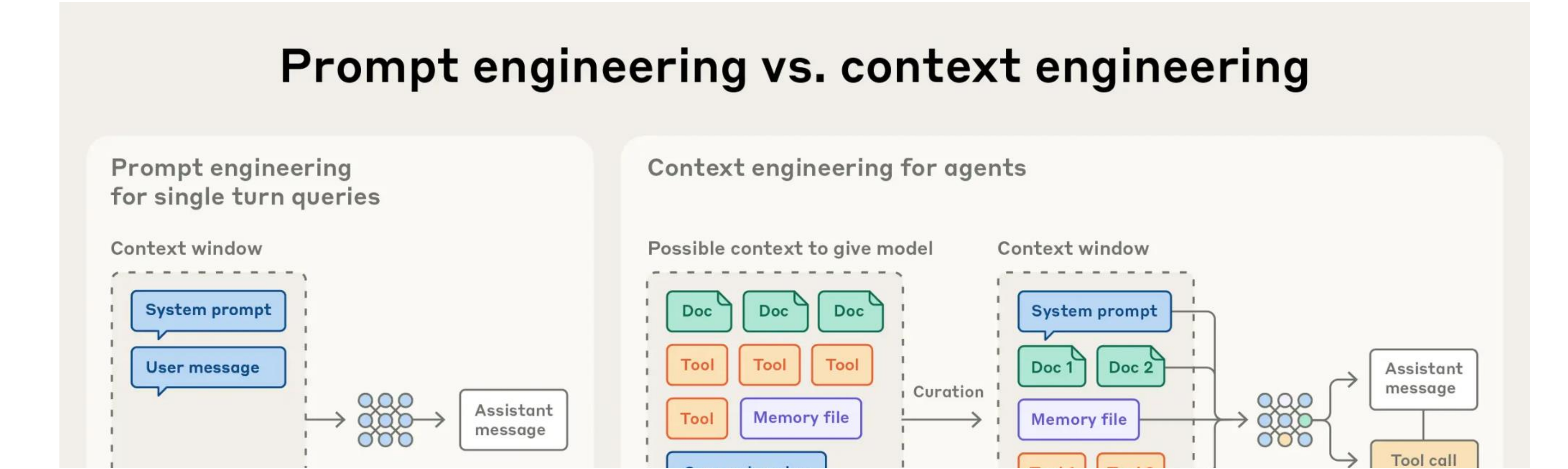


<https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents>

RECAP

去年迎来了Agent

RLHF
Agentic Loop
Context Engineering



+ a b | e a u Tableau MCP Documentation

Tableau MCP

Tableau's official MCP Server. Helping Agents see and understand data.

Get Started



RECAP

去年迎来了Agent

RLHF
Agentic Loop
Context Engineering

← Data Agent

分享

思考完成

好的，根据你的问题，我制定了以下执行计划：

步骤1：确认近30日时间范围及上一个交易日日期

1.1 调用 `execute_one_sql` 工具，查询证券交易明细数据表（表ID 3887891）中 `交易日期` 的最大值（即上一个交易日日期），并验证近30日时间范围（2025-04-21至2025-05-20）是否覆盖所有交易日数据。

步骤2：获取近30日数据概况基础数据

2.1 调用 `execute_one_sql` 工具，按 `交易日期` 分组，查询近30日每日的 `佣金收入`、`交易总金额`、`活跃客户数量`，用于绘制趋势图及计算最大值、最小值。

步骤3：分析分公司维度交易数据环比变化

3.1 调用 `execute_one_sql` 工具，获取上一个交易日各 `所属分公司` 的 `交易总金额`、`交易总笔数`（本期值）。

3.2 调用 `execute_one_sql` 工具，获取上一个交易日往前推7天的各 `所属分公司` 的 `交易总金额`、`交易总笔数`（上期值）。

3.3 调用 `code_interpreter` 工具，基于本期值和上期值计算各分公司的环比增长率及贡献度（使用贡献度公式），并生成表格展示结果。

步骤4：分析活跃用户画像

4.1 调用 `execute_one_sql` 工具，按 `股票名称` 分组，查询近30日 `交易总金额` 和 `交易总笔数`，取前5名用于柱状图。

4.2 调用 `execute_one_sql` 工具，按 `所属城市` 分组，查询近30日活跃客户数量（`count(distinct [用户ID])`），用于绘制城市分布饼图。

4.3 调用 `execute_one_sql` 工具，按 `佣金水平` 分组，查询近30日活跃客户数量（`count(distinct [用户ID])`），用于绘制佣金水平分布饼图。

本月公司经营表现分析



经营分析顾问正在执行任务。

查看结果

执行事件详情

Agent 正在执行...

SQL 选择数据表并执行SQL语句查询数据

已完成

开始执行任务：在证券交易明细数据表（表ID 3887891）中，查询交易日期的最大值，即上一个交易日的日期。此日期用于后续分析上一个交易日各分公司的交易数据以及与上周同期对比。

SQL 选择数据表并执行SQL语句查询数据

已完成

此查询问题无需拆解：在证券交易明细数据表（表ID 3887891）中，查询交易日期的最大值，即上一个交易日的日期。此日期用于后续分析上一个交易日各分公司的交易数据以及与上周同期对比。

思考过程：True

SQL 选择数据表并执行SQL语句查询数据

正在执行

调用模型生成SQL中...

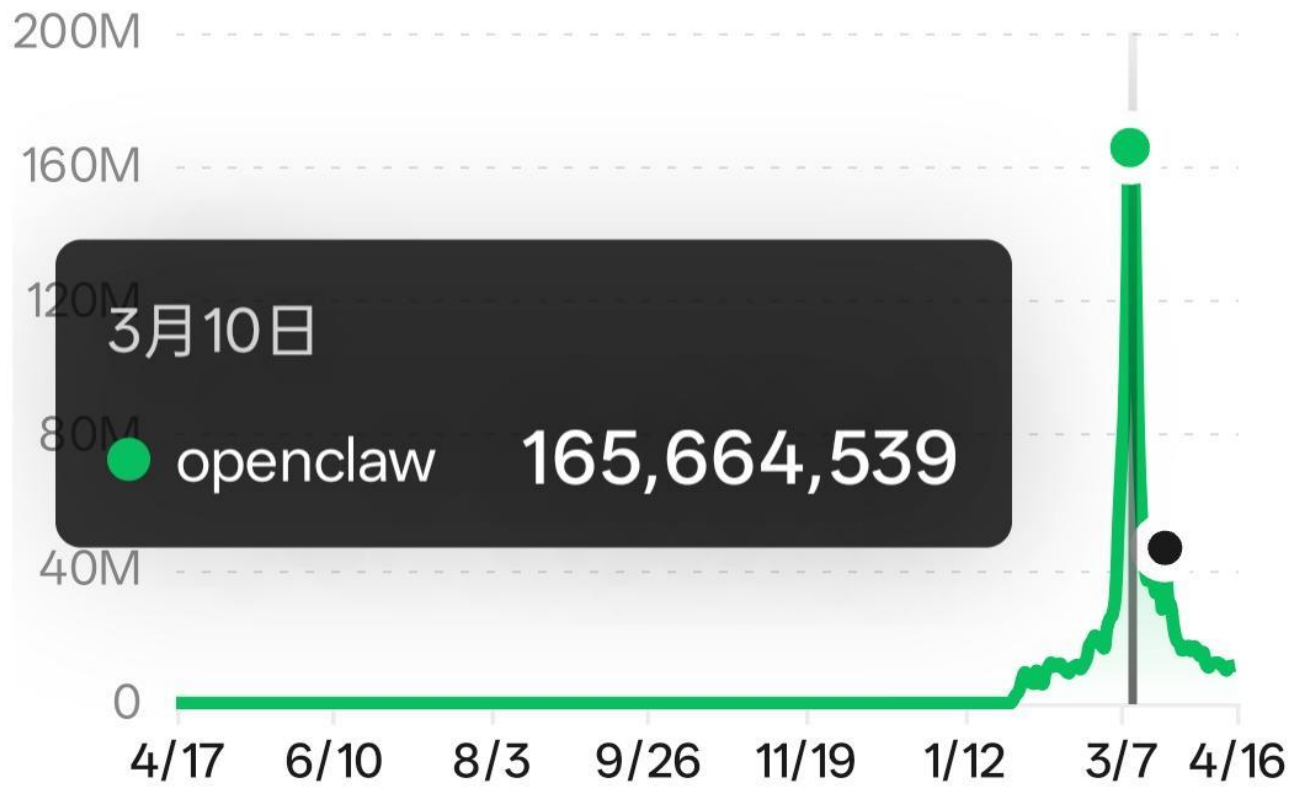
查询问题：在证券交易明细数据表（表ID 3887891）中，查询交易日期的最大值，即上一个交易日的日期。此日期用于后续分析上一个交易日各分公司的交易数据以及与上周同期对比。

RECAP

年初狂热的OpenClaw

指数趋势 ⓘ

近7天 近30天 全部



RECAP

陈巍：“龙虾”们颠覆软件世界？（上）——OpenClaw（原Clawdbot...   :

zhuanlan.zhihu.com › p › 2010119008783987145

3.3 数据流分析. OpenClaw的数据流遵循高效的端到端管道：用户输入→意图解析→任务规划→执行→结果反馈，确保信息在智能体循环中流动顺畅。该流程嵌入Lobster壳中 ...

[【CDA干货】OpenClaw保姆级教程，3分钟高效搞定数据分析！](#)

2026 年初，一款被称为OpenClaw（俗称 小龙虾）的AI工具火了。和只能提供建议的ChatGPT不同，OpenClaw被定义为个人AI智能体执行网关——它能直接操作你的电脑，执行文件整理、数据清洗、网页...

[OpenClaw 深度解析：一只龙虾凭什么震撼全世界](#)

Mar 5, 2026 原因很简单：把小龙虾部署在云服务器上，和用Manus 基本没有区别。上面没有用户的本地数据，没有用户的文件，能干的事非常有限。部署在本地电脑上 ...

给小龙虾装上业务大脑：两个 SKILL 让 OpenClaw 学会查数和归因   :

cloud.tencent.com › developer › article › 2642968

十分钟快速上手小龙虾数据分析工具，通过Aloudata CAN语义层实现智能查询与归因分析。支持自然语言提问，自动生成HTML报告，涵盖销售额、客单价等54个指标和58个维度。提供metric-query和metric-attribution两个SKILL，轻松完成数据查询和深度诊断，无需编写SQL即可获取精准业务洞察...

[火爆全网的“龙虾”OpenClaw，到底是什么来头？一篇文章给你说明白](#)

Mar 9, 2026 三、这只"龙虾"能做什么？ OpenClaw的能力举例：. 图片. 办公自动化：清空收件箱、自动发邮件、管理日历、办理航班值机、整理文档、做数据分析.

data-analysis

INSTALLATION

```
$ npx skills add https://github.com/supercent-io/skills-template --skill data-analysis
```

SUMMARY

Dataset exploration, cleaning, statistical analysis, and visualization in Python or SQL.

- Supports CSV, JSON, and SQL data sources with pandas DataFrames and direct database queries
- Covers the full analysis pipeline: data loading, missing value handling, outlier detection, grouping, correlation analysis, and pivot tables
- Includes visualization templates for histograms, boxplots, heatmaps, and time series using matplotlib and seaborn
- Generates structured markdown reports with dataset overview, key findings, statistical summaries, and actionable recommendations

SKILL .md

Data Analysis

WEEKLY INSTALLS

13.8K

REPOSITORY

supercent-io/sk...template

GITHUB STARS

☆ 88

FIRST SEEN

Jan 24, 2026

SECURITY AUDITS

Gen Agent Trust Hub

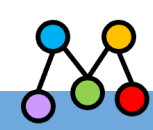
PASS

Socket

PASS

Snyk

PASS

资料整理
火龙果科技



一个事实

所有AI应用，都只是在享受Code Agent的发展红利

所有体验的进步，都来自头部模型厂商的馈赠

所有功能的迭代，都来自软件工程最佳实践的迁移

一个事实

所有AI应用，都只是在享受Code Agent的发展红利

所有体验的进步，都来自头部模型厂商的馈赠

所有功能的迭代，都来自软件工程最佳实践的迁移

数据分析领域，为AI的适配和迭代为零

一个事实

我们没有可靠的统一的对**数据语义层**的定义协议。(Open Semantic Interchange)

缺少对数据存储、数据类型统一的定义。虽然**SQL语法**本来就五花八门（SQL:1999）

对Agent生态充满敌意的**数据访问层**，依赖繁重系统驱动和老旧的SDK。

没有可靠**沙盒环境**，对Agent生成的SQL和分析代码环境隔离，角色权限过滤。

还有数据可视化，如何定义准确率？.....

一个事实

我们没有可靠的统一的对**数据语义层**的定义协议。(Open Semantic Interchange)

缺少对数据存储、数据类型统一的定义。虽然**SQL语法**本来就五花八门（SQL:1999）

对Agent生态充满敌意的**数据访问层**，依赖繁重系统驱动和老旧的SDK。

没有可靠**沙盒环境**，对Agent生成的SQL和分析代码环境隔离，角色权限过滤。

还有数据可视化，如何定义准确率？.....

仅仅为了追赶趋势，所有人就已经精疲力竭了

OpenClaw热度已过，说明了

渠道：入口为王

ChatGPT时期的提问：
Chatbot是否是AI应用的终局？

记忆：自我演进

Agent能否自举？
从对话到知识的沉淀

算力规模

20\$甚至200\$的AI订阅习以为常
用户对算力投入预期越来越高

Agent友好

交互cli化
CLI+Skill成为新潮流

推动生态发生了什么变化

云厂商更加开放了

过去被垄断的入口因为竞争agent的生态位而被打开

模型选择更多了

国产模型也能完成agent任务，更加稳定

对安全的警惕性下降了

降本增效，先发射再瞄准，先出效果再治理

放宽了AI Startup 约束条件

我们决定重写一个新的Ask Table

两个月可以做什么

如何优化开发流程

从内部的需求出发

对话为核心的Table Artifact



仪表盘



AskTable-Cli

用智能体搭建智能体

Excel / CSV

包含员工基本信息及其对应的薪酬详情。

ID: ds_3Q5La... 1个库 1张表 25个列 正常可用 创建于 20 天前

数据概览 文件管理 训练集 评测集

数据概览 同步成功 · 2026-03-28 14:55:15

Sheet1

入职日期	入职时间			未启用	2026/3/28 14:54:44
记录日期	数据记录时间			未启用	2026/3/28 14:54:44
年份	年份			未启用	2026/3/28 14:54:44
季度	季度			未启用	2026/3/28 14:54:44
月份	月份			未启用	2026/3/28 14:54:44
大区	大区			未启用	2026/3/28 14:54:44
省份	省份			未启用	2026/3/28 14:54:44
城市	城市			未启用	2026/3/28 14:54:44
办公地点	办公地点			未启用	2026/3/28 14:54:44
部门	所属部门			未启用	2026/3/28 14:54:44
子部门	子部门			未启用	2026/3/28 14:54:44
职位级别	职级			未启用	2026/3/28 14:54:44



@datamini/asktable-cli

0.0.6 • Public • Published 4 days ago

Readme

```
• Read 1 file (ctrl+o to expand)

分析完成！虽然15个用例全部执行成功，但Agent发现了大量SQL质量问题：

- 高优先级：10个（字段类型不匹配、ClickHouse方言兼容、公式错误等）
- 中优先级：1个（SQL质量优化）

不过我看到报告中的"API操作"部分是空的，说明这些建议没有直接对应的API payload（主要是SQL层面的修正，不是字段备注更新）。让我看看第五步 apply 命令的实现。

• Read 1 file (ctrl+o to expand)

• Bash(cat /private/tmp/at-tuner-test/tuning-report.json | python3 -c "import
  json,sys; data=json.load(sys.stdin);
  print(json.dumps(data.get('suggestions', []))[:2], in...)
  [ [
    {
      "category": "sql_quality",
      ... +25 lines (ctrl+o to expand)

看到了问题。分析步骤生成的建议是SQL修正类型（filter_value_mismatch、dialect_co
patibility等），但 apply 命令期望的是 field_desc、glossary、skill
类型。这些建议无法直接通过API应用。

这些发现的问题可以通过字段备注更新来解决—比如让AI知道 status
是数值类型、credit_limit 是字符串类型等。让我修改 analyze
步骤，让它生成正确的可应用建议格式。

• Read 1 file (ctrl+o to expand)

问题找到了：analyze 步骤的 prompt 要求返回 field_desc | glossary | skill
类型，但 AI 实际上返回了 filter_value_mismatch 等自定义类型。而且 prompt
太弱，没有让 AI 给出具体的字段备注建议。

让我重写 analyze 步骤，直接让 AI 基于发现的问题生成可执行的字段备注更新建议。

• Read 1 file (ctrl+o to expand)

明白了。让我重写 analyze 步骤，让 AI 直接生成可应用的 field_desc/glossary/skil
建议。关键是要让 AI 识别需要更新哪些字段的备注。

* Actioning... (8m 24s · ↓ 1.4k tokens · thought for 4s)
```


Data Agent

技能

管理可复用的指令，让数据智能体在需要时按需激活

技能

+ 创建

人数统计

按维度统计排名分析：用于回答"某表/某字段有多少"、"按XX维度分别是多少"、"最少/最多的N个..."

12 天前

empty

没什么用的技能

17 天前

经费和师资分析

分析各学院的师生比情况，找出资源分配不均的学院，并结合经费预算数据进行深度对比，揭示教...

17 天前

empty

没什么用的技能

创建时间 17 天前

对话打磨

编辑

删除

内容

当智能体激活该技能时，这段内容会原样返回给它。

请输入完整技能指令

Data Agent

技能

管理可复用的指令，让数据智能体在需要时按需激活

技能

人数统计
按维度统计排名分少"、"按XX维度12 天前

empty
没什么用的技能17 天前

经费和师资分析
分析各学院的师生院，并结合经费预17 天前

数据智能体 > 高校数据分析助手

高校数据分析助手

输入描述（可选）

创建时间 2 天前

数据源

技能

记忆

能力配置

<> Python 解释器

数据变换、统计分析等 SQL 难以表达的操作

? 用户澄清

允许智能体向用户提问确认

+

添加

图表可视化

为查询结果创建可视化图表

编辑

...

资料整理
火龙果科技

荐

Data Agent

AskTable

技能

管理可复用的指令，让数据智能体在需要时按需激活

技能

人数统计

按维度统计排名分少"、"按XX维度12 天前

empty

没什么用的技能17 天前

经费和师资分析

分析各学院的师生院，并结合经费预17 天前

数据智能体 > 高校数据分析助手

高校数据分析助手

输入描述（可选）

创建时间 2 天前

数据源

技能

记忆

能力配置

Python 解释器

数据变换、统计分析等 SQL 难以表达的操作

用户澄清

允许智能体向用户提问确认

+

添加

李忠民

@data agent 这周有多少注册用户?

1 条回复

data agent 机器人 | connector for asktable data agent

回复 李忠民: @data agent 这周有多少注册用户?

**本周注册用户数:

统计范围为本周一至今（`deleted\_at IS NULL` 排除了已删除账号）。

如果你需要进一步了解，比如：

每天的注册趋势

按注册渠道（source）分布

与上周对比

随时告诉我！

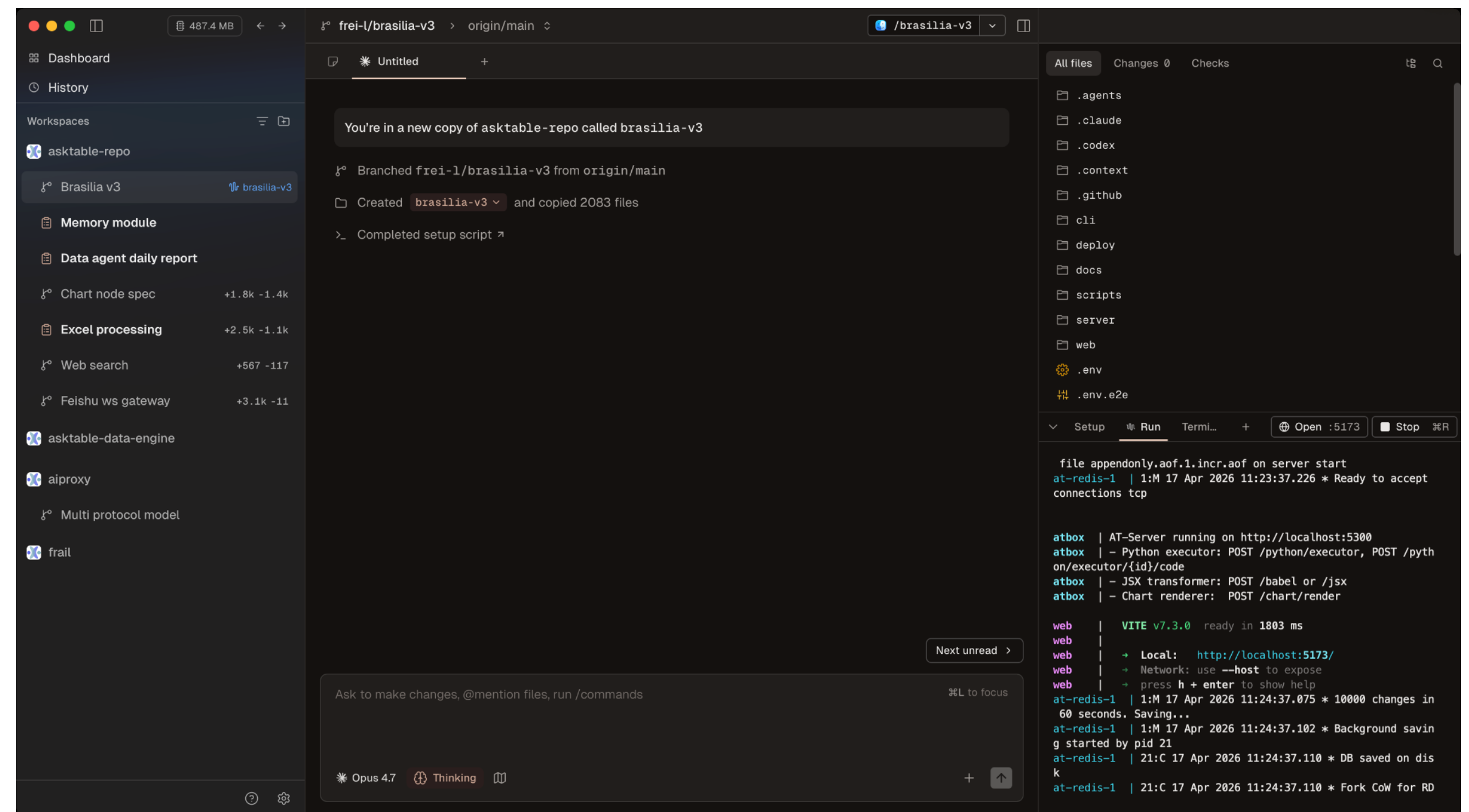
优化产品研发流程

Monorepo

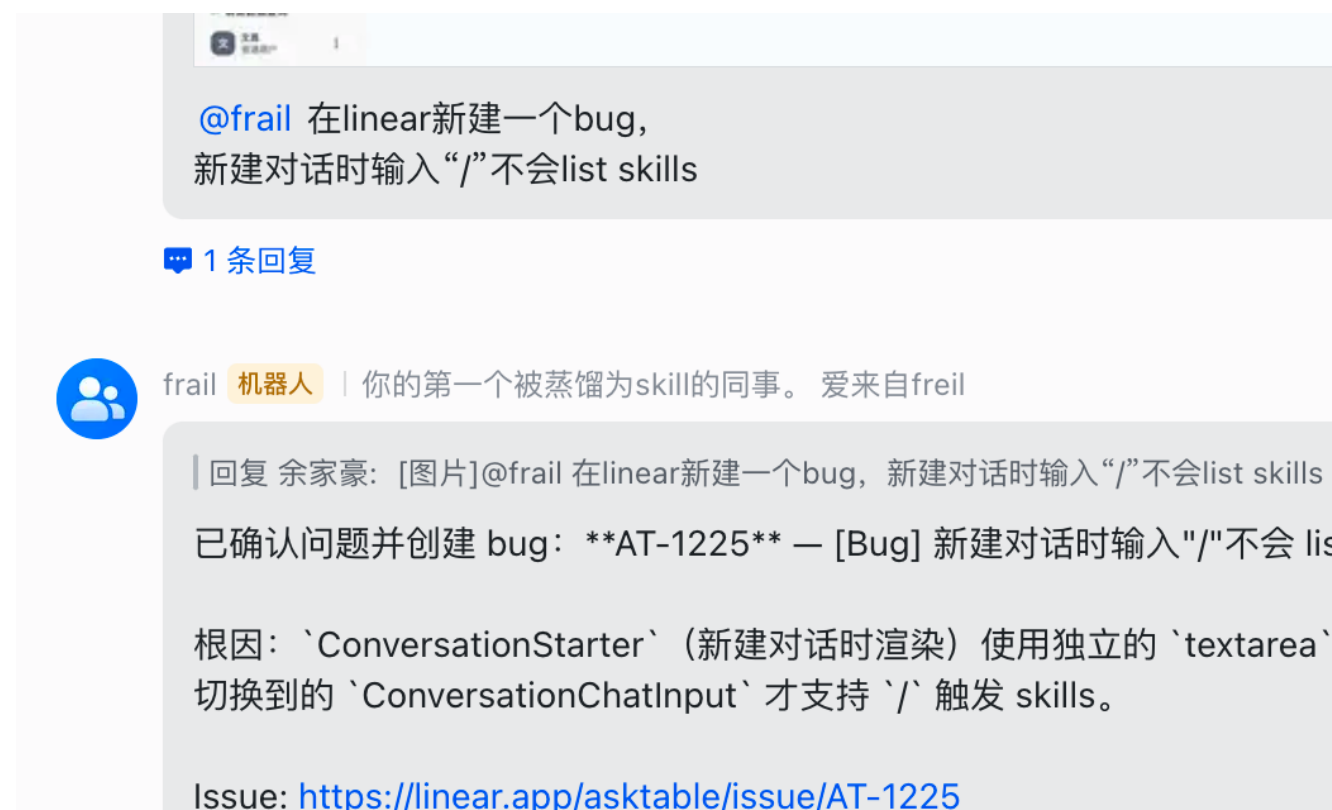
所有产品研发文档从飞书迁移到代码仓库

规范[AGENT.md](#) [DESIGN.md](#)
MCP, Skills

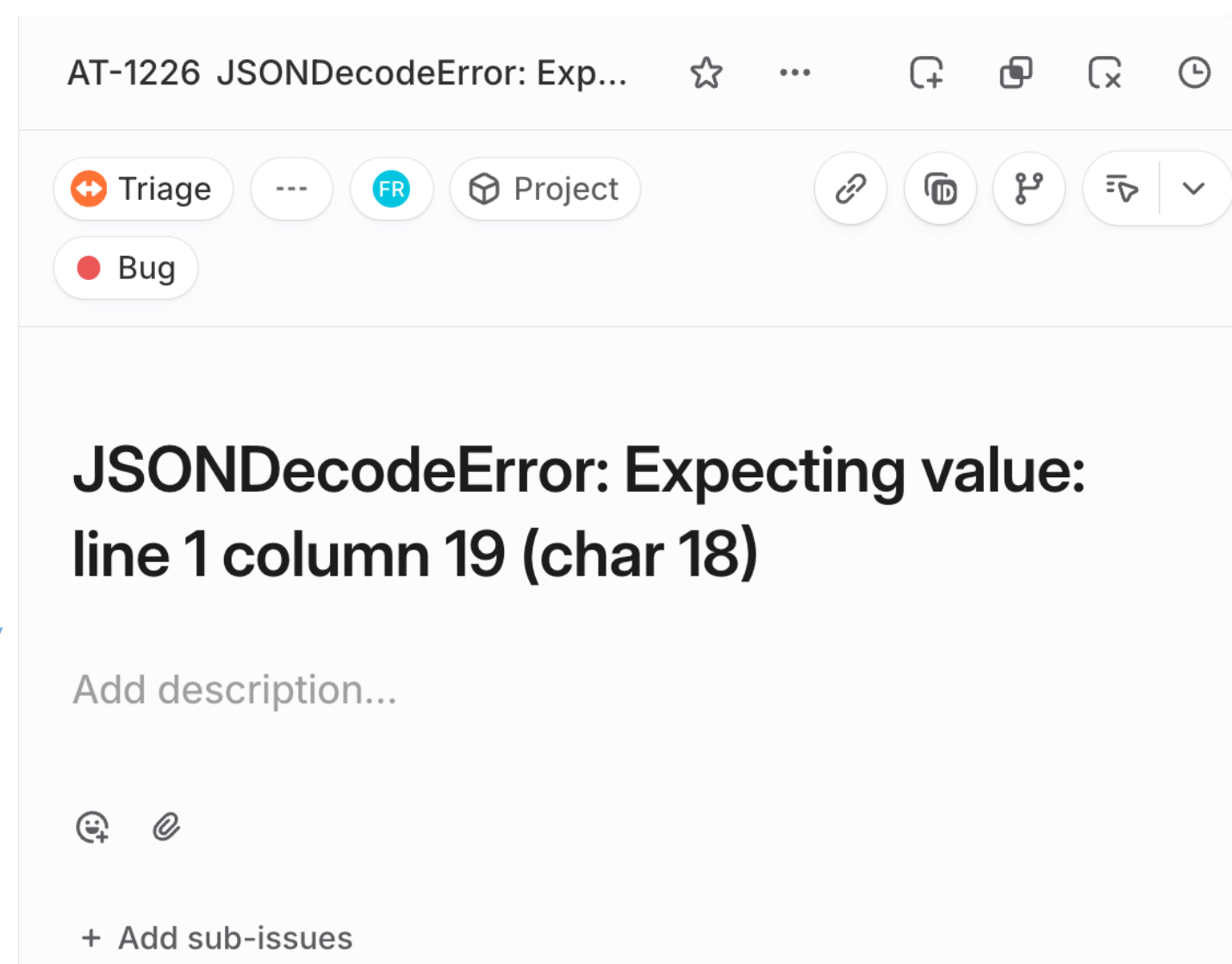
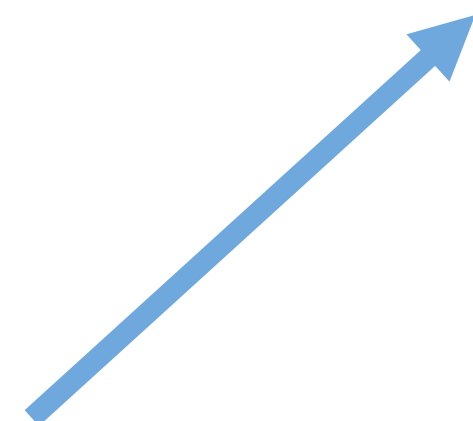
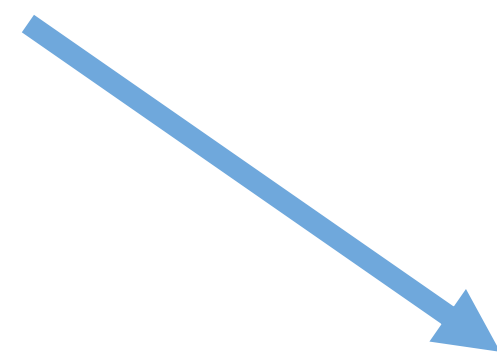
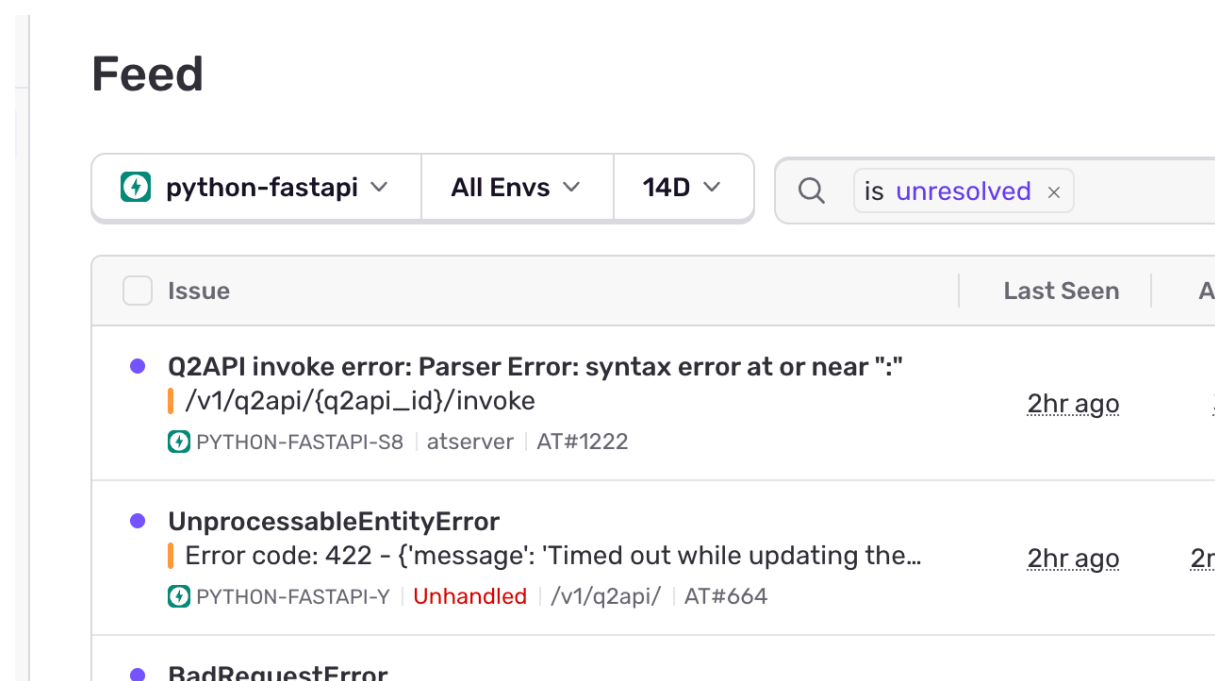
Prompt Request代替Pull Request



Workflow



飞书收集反馈



汇总Linear



AskTable

conductor + skills

/debug

/ceo-review +
/eng-review

/codex + /qa +
agent browser
端到端验证

/release

DogFooding

从团队内部需求出发，反馈具体，迭代迅速

1. 对团队内的协作工具做优先支持，对团队用的数据渠道来源做优先接入
2. **我们需要CLI自动化客户的POC，客户只要提供示例数据，我们就可以生成DEMO**
3. **我们需要每天日报，推送昨天注册用户每个功能的使用情况，以及是否需要跟进**
4. **我们需要每周一份数据，追踪页面的浏览量，平台点击率。**
5. **我们需要把pg的用户数据和飞书多维表格的数据做关联查询**

我是否真的需要DataAgent

1. 技术日新月异，大部分是噪音和再发明。
1. 决策质量的上限是人的带宽，AI只是概率模型，最擅长喷射代码和报告。但是没有被人类理解和验证，就都没有意义。
1. 当数据不再稀缺，阻碍进步是个人判断和组织架构。

我是否真的需要DataAgent

1. 技术日新月异，大部分是噪音和再发明。
1. 决策质量的上限是人的带宽，AI只是概率模型，最擅长喷射代码和报告。但是没有被人类理解和验证，就都没有意义。
1. 当数据不再稀缺，阻碍进步是个人判断和组织架构。

为先进团队减少数据的阻力和噪音



让智能体真正易用

FastGPT Agent 的设计思路与落地价值

FastGPT 产品团队

Design Philosophy & Business Value

从 Chatbot 到真正的任务型 Agent



Core Insight: Agent 不再只是"回答问题", 而是能够理解目标、制定计划、调用工具、持续执行直至任务完成。

五大核心痛点制约 Agent 规模化落地



1. 门槛高

构建 Agent 需要深厚的 Prompt 工程与系统设计经验，普通开发者难以上手



2. 不稳定

大模型输出随机性导致任务执行结果不可预期，难以用于生产环境



3. 工具协同难

多工具调度逻辑复杂，接口异构，缺乏统一的工具管理机制



4. 记忆不足

长任务中上下文丢失，无法维持跨轮次的状态与计划连贯性



5. 黑盒与安全

执行过程不透明，决策链路难以追踪，存在安全与合规风险

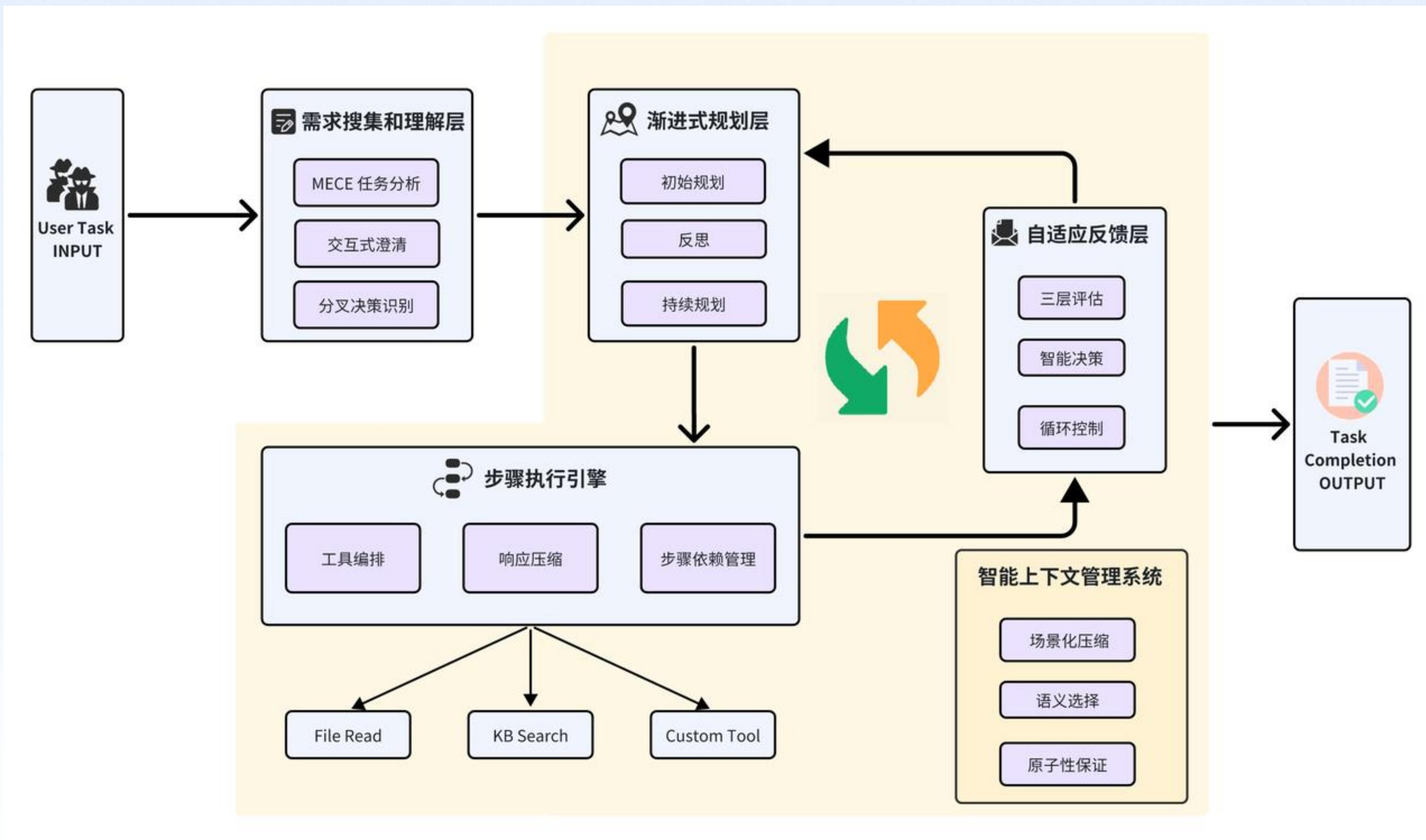
五层协同体系，重新定义 Agent 能力边界

- 面向复杂任务的智能体系统
- 同时解决“创建 Agent”与“运行 Agent”两类核心问题
- 从问答能力走向任务执行能力





整体架构



对话式生成 Agent：降低构建门槛

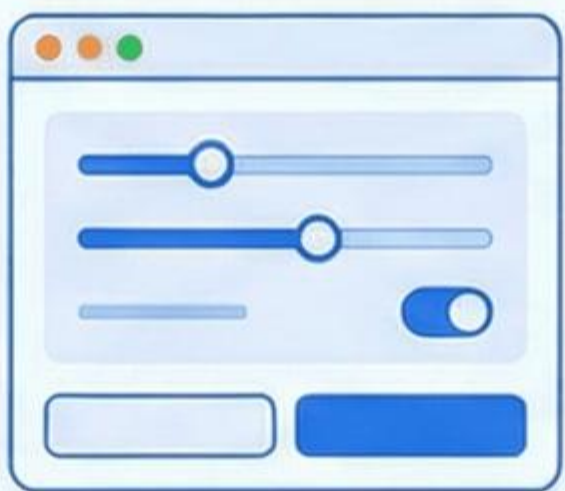
用自然语言描述需求，自动生成完整 Agent 配置



用户输入自然语言需求



FastGPT 自动生成：
Prompt + 工具选择 +
知识库配置 + 能力参数



Agent 即刻可用，无需手动调试

Key Points:



• 无需 Prompt 工程经验，自然语言即配置



• 自动推断所需工具与知识库，零门槛构建



• 支持迭代优化，对话式调整 Agent 行为



决策与路由机制：智能选择执行模式

基于复杂度与上下文的动态决策路由



核心优势

系统根据任务复杂度与当前上下文自动选择最优路径，无需人工干预，避免过度规划或执行不足。



渐进式规划：告别僵化的一次性计划

当前阶段优先，信息不足先问，执行后继续规划

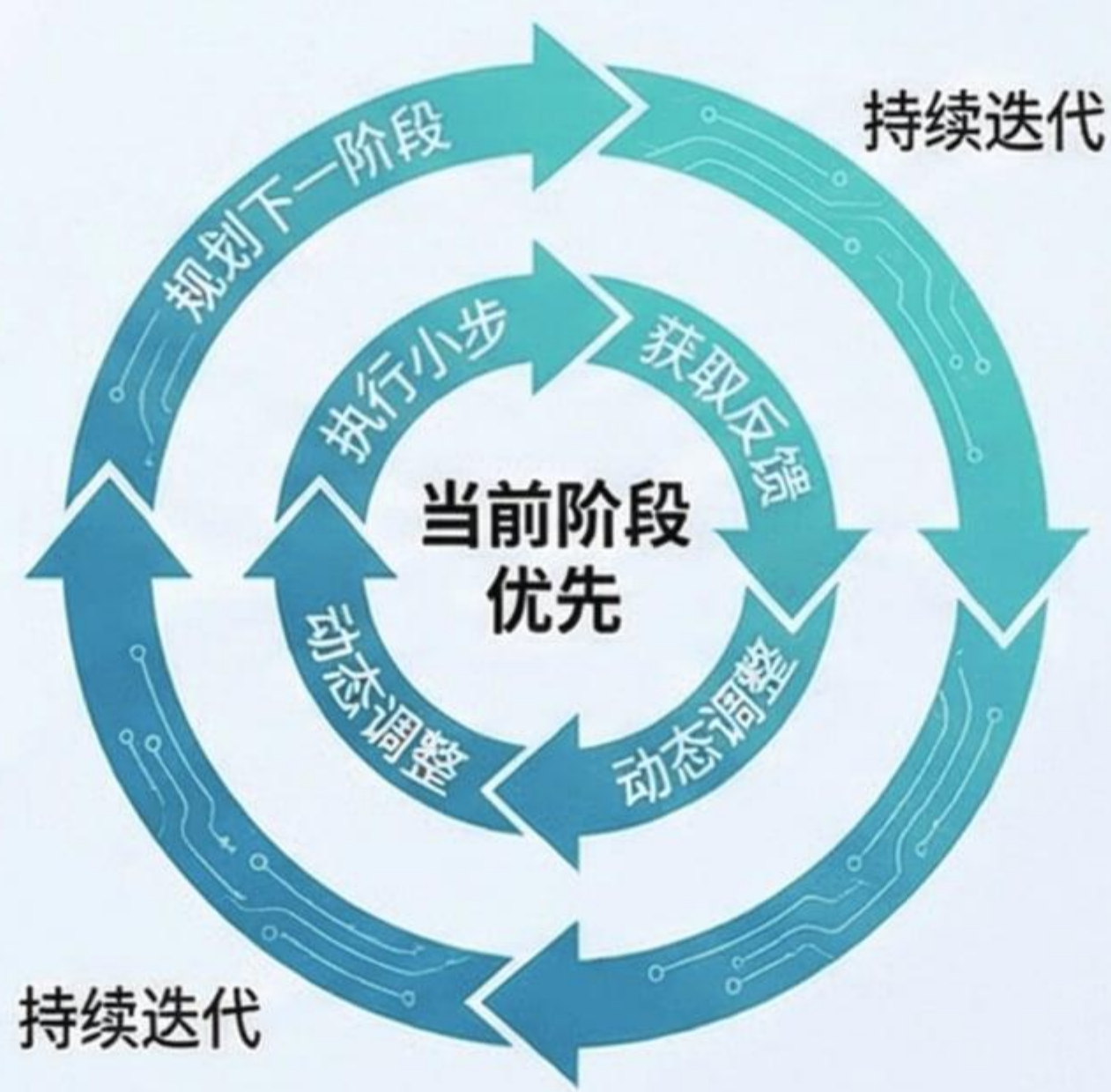


传统方案



僵化全局规划，信息滞后
适应性差

FastGPT 渐进规划



渐进式规划，快速迭代
适应变化、动态调整



当前阶段优先

只规划当前可执行的步骤，避免过早承诺。



信息不足先 Ask

主动向用户澄清歧义，确保执行方向正确。



执行后继续规划

基于中间结果动态调整后续计划。



避免僵化全局规划

拥抱不确定性，保持计划弹性。

多步骤执行与统一工具接入

工具、知识库、文件、沙箱，统一调度一站式执行

- **分步执行：**将复杂任务拆解为可追踪的原子步骤
- **统一调度：**四类工具统一接口，无缝协作
- **中间沉淀：**每步结果保存，支持回溯与调试

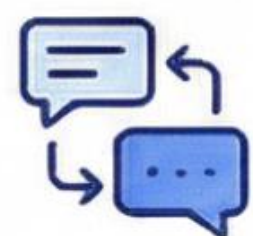




多轮上下文管理与断点恢复

任务可暂停、可恢复，长程执行不丢失状态

三层上下文管理机制



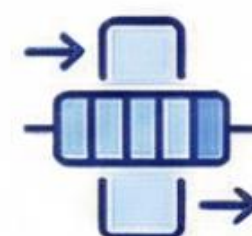
主消息管理

维护完整对话历史，确保上下文连贯



场景化压缩

将当前任务计划序列化保存，支持跨会话恢复



缓冲区管理

中间执行状态与工具输出的临时存储与复用

断点恢复场景



用户中途提供新信息
→ Agent 无缝融入并继续



网络中断或超时
→ 从最近检查点恢复执行



用户主动暂停
→ 保存完整状态，随时续接

FastGPT Agent 的核心价值

四大核心价值，驱动 Agent 真正落地

01



降低构建门槛

对话式生成，无需专家，人人可建 Agent

02



提升复杂任务完成能力

渐进规划 + 多步执行，攻克长链路任务

03



增强执行过程可解释性

每步可追踪，无策链路透明，告别黑盒

04



支持真实业务场景落地

稳定、可控、可集成，生产级可用

未来架构演进与产品 Roadmap

从"能做"走向"可控", 从专家搭建走向全民构建

1. Agent Runtime 可定制

支持接入 Claude Code 或主流开源 Agent 框架, 满足高度定制化需求



2. Sandbox 定制化

用户可自定义预装环境, 一键发布到 FastGPT, 打造专属执行沙箱



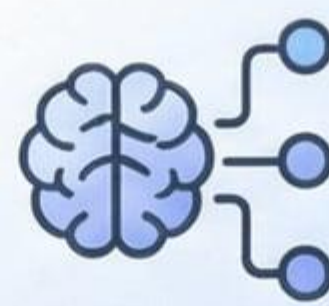
3. Skill 体系 (本月发布测试版) 🚀

模块化技能市场, 即插即用扩展 Agent 能力边界



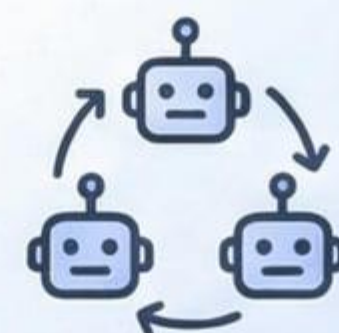
4. 从单模型走向系统能力

多模型协同、专家模型路由, 突破单一模型上限



5. 从单 Agent 走向多 Agent 协同

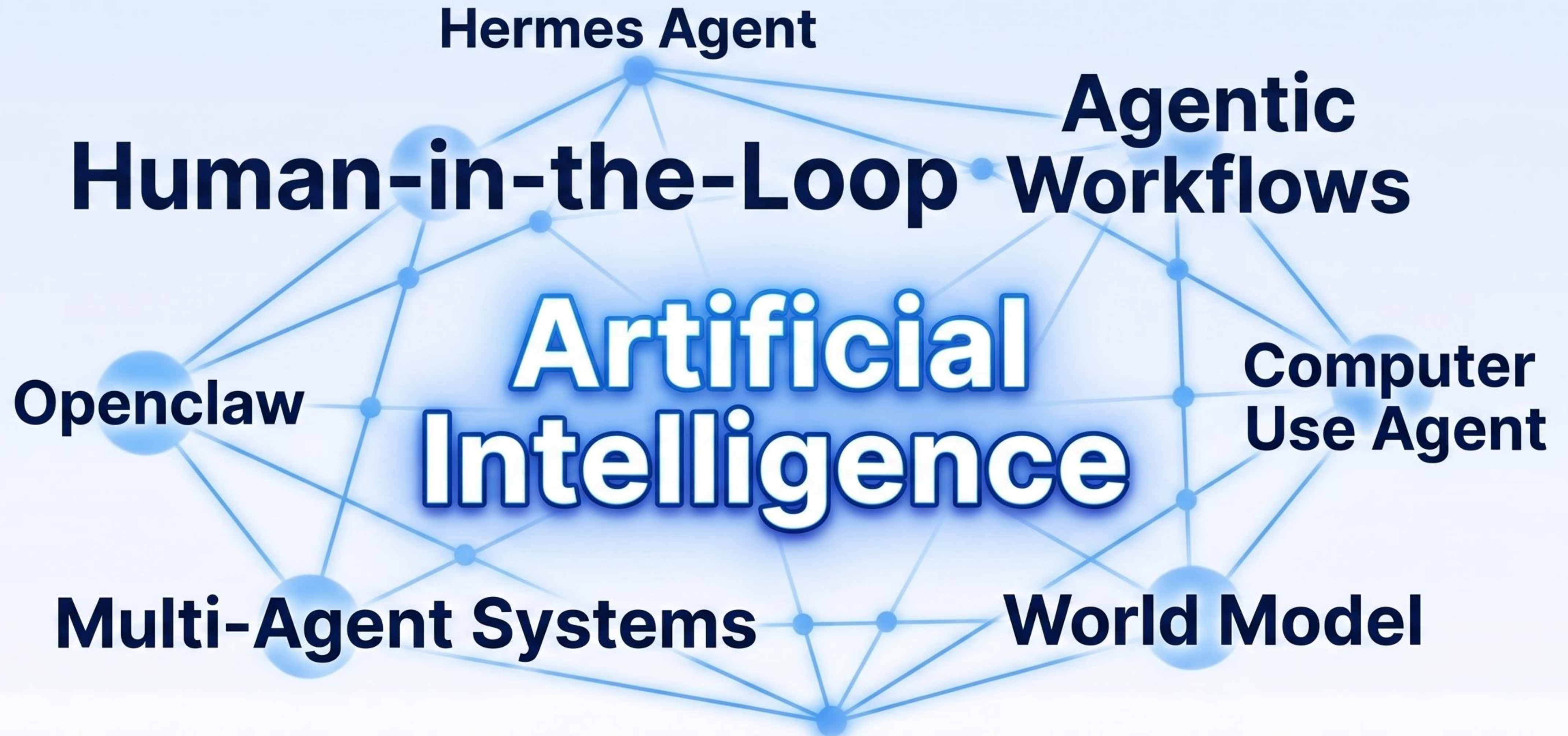
任务分发、角色分工、结果汇聚的多智能体网络



6. 从通用能力走向业务深度集成

与企业系统、行业数据深度融合, 打造垂直领域 Agent







FastGPT Agent，让每个人都能拥有自己的智能助手

真正易用的 Agent，不是让用户适应 AI，而是让 AI 真正理解并服务于用户的目标。