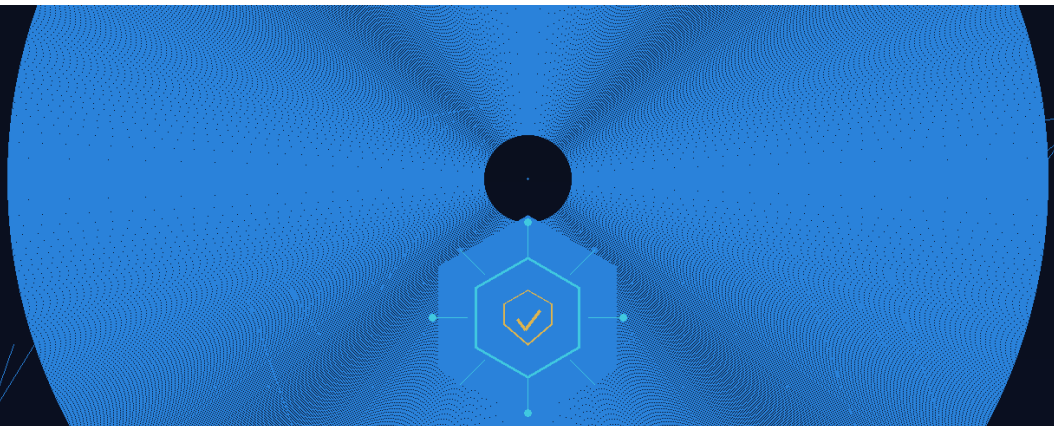


可控可信企业智能体白皮书

从“更聪明的AI”到“更可信的AI”

企业级智能体的治理范式、技术架构与商业路径

Agent = Intelligence + Control + Responsibility

可控
Controlled
默认拒绝
显式授权
边界内执行

可信
Trusted
身份可验证
决策可追溯
责任可归属

可审计
Auditable
推理链记录
全过程追踪
持续改进

版本：v1.0
日期：2026 年 6 月
分类：行业白皮书

可控可信企业智能体白皮书

> **企业级 Agent 的治理范式、技术架构与商业路径——从内部协同到跨企业协作**

摘要

大语言模型（LLM）的快速演进使智能体（Agent）具备了前所未有的理解、推理和规划能力。然而，行业对 Agent 的关注长期集中在一个方向上——**企业内部**如何用 Agent 替代员工、提高效率、优化流程。这固然重要，但它只揭示了问题的一个维度。

本白皮书提出一个被行业普遍忽视但至关重要的命题：

> **"可控可信" 的主战场，不只是企业内部 Agent，企业之间的 B2B Agent 也是。**

企业内部 Agent 解决的是**组织效率**——一个公司内部怎么跑得更快。可控可信在这里是"最佳实践"：在统一信任域内，通过审批、权限、审计等机制确保 Agent 安全运行。

跨企业 Agent（B2B Agent）解决的是**协作效率**——不同公司之间如何通过 Agent 完成采购、供应链协同、物流、金融等标准化协作。可控可信在这里升级为**刚性要求**：不同企业的 Agent 如何互相认证身份？如何控制信息可见性？如何确保每一笔跨企业交易都有可追溯的责任链？如何在没有任何一方拥有绝对权威的情况下，让 Agent 安全地代表企业做出承诺?

无论是内部还是跨企业，核心问题都不是"Agent 聪明不聪明"，而是 Agent 可不可信——只是在跨企业场景中，这个问题的复杂度和风险等级是数量级提升的。

本白皮书提出一套以 **"可控可信"为核心的企业级 Agent 治理架构**——它同时适用于企业内部和跨企业场景，但在跨企业协作中面临更高的要求和更丰富的内涵。核心主张是：

> **Agent = Intelligence（智能）+ Control（控制）+ Responsibility（责任）**

并从以下五个维度展开论述：

1. **"可控可信"的双重战场**——企业内部 Agent 治理是基础，跨企业 B2B Agent 治理是升级和扩展
2. **Agent 治理模型（CAEM）**——默认拒绝、显式能力、决策链追踪、人类最终负责，从内部到跨企业逐级增强
3. **六层运行时架构**——身份、能力、治理、上下文、执行、审计，在跨企业场景中扩展本域/跨域双模执行
4. **Agent 通信的七层治理模型**——从身份认证到信息分级到临时授权，在企业内部是组织治理，在跨企业场景升级为跨组织治理
5. **商业路径：Enterprise Agent Network（企业 Agent 网络）**——为什么最大的机会不是开发 Agent，而是成为 Agent 之间的连接层

本白皮书为企业决策者、技术架构师和行业从业者提供一个全新的视角：**未来最大的商业机会未必是开发更多单点 Agent，而是成为跨企业 Agent 协作的基础设施——定义协作协议、身份认证、权限体系、上下文共享和流程编排。谁能让不同企业的 Agent 安全、低成本、高可信地完成业务协同，谁就有机会成为下一代企业互联网的核心基础设施。**

目录

序章：TEA 核心设计原则

- 原则一：Business First, Models Second（业务优先，模型其次）
- 原则二：Bounded Intelligence（边界智能）
- 原则三：四级输出分级（Output Classification）

序章：TEA 核心设计原则

在正式展开市场分析和技术架构之前，本章先确立 TEA 最根本的几条设计原则。它们不是实现细节，而是决定了 TEA 与今天大多数 Agent 框架的**本质区别**。

原则一：Business First, Models Second（业务优先，模型其次）

> **企业级 AI 应用，不应面向模型开发，而应面向业务能力开发。**
>
> *Enterprise AI applications should be developed against business capabilities rather than foundation models.*

为什么这个原则重要?

回顾企业软件几十年的发展，有一个不变的规律：**企业永远不会围绕技术本身开发，而是围绕业务开发。**

- ERP 不会围绕 Oracle 开发，而是围绕财务管理、供应链管理。
- CRM 不会围绕 MySQL 开发，而是围绕客户管理、销售流程。
- OA 不会围绕 Redis 开发，而是围绕审批流程、协同办公。

技术一直在变，数据库从 Oracle 到 MySQL 到 TiDB，业务逻辑相对稳定。**技术是可替换的执行资源，业务能力才是长期稳定的软件资产。**

AI 今天正在犯同样的错误

今天很多企业的 AI 应用代码是这样的：

```
# 面向模型开发 —— 换模型几乎等于重写
deepseek.chat(prompt)
gpt.chat(prompt)
claude.messages(prompt)
```

整个业务逻辑绑死在具体模型上。模型升级 → Prompt 重写 → Workflow 修改 → 应用跟着升级。这是今天 AI 应用最大的架构问题。

TEA 主张的方式

企业应该面向**业务能力（Business Capability）**开发，模型退居幕后成为可替换的执行资源：

```
# 面向业务能力开发 —— 模型切换零改动
contract.review(document)        # 合同审核
customer.reply(ticket)           # 客服回复
finance.analyze(report)          # 财务分析
purchase.approve(order)          # 采购审批
```

业务代码永远不知道后面是什么模型。今天用 GPT，明天用 DeepSeek——业务代码**零修改**。

这意味着什么?

TEA 的抽象层次**高于模型和 Agent**。模型、Agent、GPU、工具都只是 TEA Runtime 调度的资源。真正面向开发者和企业暴露的接口，不是模型接口，而是**业务能力接口（Business Capability Interface）**。

```
业务层（Business Capabilities）
    合同审核 / 客服回复 / 销售分析 / 采购审批 / 知识查询
                        │
                        ▼
        业务能力接口（Business Capability API）
                        │
                        ▼
              TEA Runtime
     Intent / Planning / Policy / Routing / Memory / Audit
                        │
                        ▼
        模型层（Foundation Models）
     GPT / DeepSeek / Qwen / GLM / Claude / Llama
```

模型已经退化成基础设施。面向模型开发的应用会不断被技术迭代牵着走；面向业务能力开发的应用，则能把技术变化隔离在 Runtime 之下。

原则二：Bounded Intelligence（边界智能）

> **每个企业 Agent 都必须运行在明确规定的能力边界、权限边界和决策边界之内。**
>
> *Every Enterprise Agent must operate within an explicitly defined capability boundary, authorization boundary, and decision boundary.*

为什么需要边界?

现在很多 Agent 框架关注的核心问题是：

> **How can the Agent do more?（如何让 Agent 做更多事情？）**

而 TEA 关注的核心问题是：

> **How do we ensure the Agent only does what it is supposed to do?（如何确保 Agent 只做它应该做的事情？）**

企业真正担心的通常不是 Agent 能力不够，而是 **Agent 在没有适当授权、没有充分依据或超出职责范围时替企业做了决定**。

能力边界的具体含义

就像人类企业中的员工各有职责范围——财务不会审批法律，HR 不会设计芯片，CEO 不会写驱动程序——Agent 也应该有明确的边界：

财务 Agent：

- ✅ 发票审核、报销审批、财务分析、成本预测

- ❌ 修改公司战略、删除数据库、调整人事组织

法务 Agent：

- ✅ 合同审查、风险提示、法规查询
- ❌ 自动签合同、自动付款

研发 Agent：

- ✅ 写代码、修 Bug、提交 PR
- ❌ 自动发布生产

边界智能与 CAEM 的关系

Bounded Intelligence 是 CAEM（受控智能体执行模型）的哲学基础。CAEM 的四条原则——Default Deny、Explicit Capability、Decision Trace、Human Accountability——本质上都是**边界智能**在不同维度的技术实现。

边界智能不是限制 Agent 的能力，而是**让 Agent 的能力可以被企业信任**。一个没有边界的 Agent，能力再强也无法进入企业核心业务；一个边界清晰的 Agent，才能从"智能助手"成长为"可信的数字员工"。

原则三：四级输出分级（Output Classification）

TEA 将 Agent 的输出划分为四个等级，每一级对应不同的执行策略和治理要求：

输出级别	含义	示例	自动执行?
🟢 **Information**（信息）	查询、总结、解释	"库存还有 350 件"	✅ 自动
🔵 **Recommendation**（建议）	给出方案和建议	"建议采购 B公司，报价最低"	✅ 自动
🟡 **Decision Proposal**（决策建议）	涉及业务决策	"建议批准 240 万采购订单"	⚠️ 人工确认
🔴 **Action Execution**（执行）	涉及真实业务动作	付款、签约、发布生产	⚠️ 必须授权+审批

核心要点：🟡 和 🔴 级别的人工确认，不是因为 Agent "不会"或"不够聪明"，而是因为**它超出了 Agent 的决策边界**。这是企业治理的刚性要求，不是技术能力的限制。

TEA Runtime 根据任务类型和风险等级自动判断当前输出属于哪个层级。如果超过 Agent 的决策边界，自动升级为"需人工核实"，整个过程都有决策依据和审计记录。

---

1. 引言：从“大模型”到“小应用”——AI 商业价值的回归

1.1 市场演进的两个阶段

过去两年，AI 行业经历了清晰的两个阶段：

- **2023–2024：能力建设期**——市场聚焦于“做一个更大的模型”，追求参数规模与基准测试分数的提升。
- **2025–2026：应用爆发期**——市场开始转向“做无数个更小、更垂直、更深入的小应用”，将 AI 能力嵌入具体业务场景。

真正赚钱的机会，已经从“做一个更大的模型”转向“做无数个更小、更垂直、更深入的小应用”。

1.2 大模型正在成为基础设施

大语言模型越来越像电力：没有人因为有电就付钱——人们付钱是因为冰箱、空调、洗衣机、手机。同样，没有人因为 GPT、Claude 或 Gemini 本身而付钱——人们付钱是因为 **AI + 某一个具体场景**：AI 写合同、AI 面试助手、AI 销售助手、AI 采购助手、AI 财务助手、AI 医疗助手……每一个都是一个数十亿甚至上百亿的市场。

1.3 企业买的是什么?

企业不会买 AI。企业买的是**降本**或者**增收**。

- **销售跟进 AI**：自动分析客户沟通、提醒销售、撰写报价、预测成交率——企业愿意年付数十万，因为它直接提升销售额。
- **客服机器人**：100 人客服 → 30 人客服，成本下降 70%，ROI 立即可见。
- **合同审核系统**：AI 检查风险、标红条款、提供修改建议，每份合同节省 30 分钟，一年数万份，价值巨大。

1.4 真正值钱的是 Workflow

AI 产品不等于聊天框。真正产生价值的是完整的工作流：

```
输入 → AI → 工具 → 数据库 → 审批 → 通知 → 执行 → 反馈 → 继续 AI
```

AI 只是其中一步，真正赚钱的是整个 Workflow。例如招聘流程：

```
收到简历 → AI 评分 → 安排面试 → 发送邮件 → 生成 Offer → 入职流程
```

1.5 可控可信的两个维度：企业内部与跨企业

目前大多数 AI 应用和 Agent 产品的思路，都是**围绕单个企业内部**来设计的：帮助企业内部的销售、客服、法务、HR 提高效率。在企业内部，可控可信体现为审批流程、权限管理、操作审计——这些是每一个企业级 Agent 都需要的基本治理能力。

但更大的想象空间，在企业之间。

当 Agent 需要代表不同的企业互相协作时——采购 Agent 向供应商 Agent 询价、物流 Agent 与海关 Agent 对接、银行 Agent 审核企业 Agent 提交的贷款材料——可控可信的要求就从“最佳实践”升级为“刚性要求”。这里没有统一的权限体系，没有共同的老板，没有天然的信息对称。

可控可信在企业内部是治理能力的标配，在跨企业协作中是整个商业模式的前提。本白皮书将同时覆盖这两个维度，并重点阐述跨企业场景对治理体系提出的更高要求。

1.6 价值衡量的演进：从 Token 到 Business Outcome

大模型正在经历一场计费模式的深刻变革，这对企业级 Agent 的架构设计有直接影响。

从"卖模型"到"卖算力"：DeepSeek V4 推出的峰谷计费是一个标志性信号——GPU 算力正在像电力、云计算一样成为按需定价的资源。模型能力趋近后，真正稀缺的是 GPU 算力、推理吞吐量和响应延迟。

Token 不是终点，而是过渡：同样输出 1000 Token，普通文本总结和调用 20 个工具的深度推理任务，GPU 消耗可能相差几十倍。Agent 执行一个任务，最终输出可能只有 500 Token，但后台思考了 10 万 Token、调用了 OCR、Vision、Embedding——**1000 Token ≠ 相同算力成本**。

价值衡量单位的演进路径：

```
Token → Compute（算力）→ Task（任务）→ Business Outcome（业务结果）
```

阶段	企业关心的问题	TEA 的应对
Token 时代	"哪个模型最便宜？"	模型价格监控
Compute 时代	"高峰时段能不能自动切到低价模型？"	峰谷调度 + 模型切换
Task 时代	"完成一份合同审核多少钱？"	Cost per Task 统计与优化
Outcome 时代	"获取一个客户多少钱？"	业务指标关联 + ROI 分析

这对 TEA 意味着：**TEA Runtime 的优化目标不是"选最便宜的模型"，而是"以最低成本、最高质量、可治理地完成一个业务目标"。** Token 会像 CPU 时钟周期一样退居幕后——平台内部调度的指标，而非企业关心的核心指标。企业未来购买的不是"API 调用次数"，而是"每天完成 10 万次客服对话"或"每月审核 100 万份合同"——平台内部再决定模型、算力和调度策略。

---

2. 核心命题：可控可信——从企业内部 Agent 到跨企业 B2B Agent

2.1 两类 Agent 问题的本质区别与递进关系

企业内部 Agent 和跨企业 Agent（B2B Agent）解决的是两类不同维度的问题，但它们不是对立的——B2B Agent 的可控可信要求，是在内部 Agent 治理基础之上的扩展和升级：

	企业内部 Agent	跨企业 B2B Agent
解决的问题	组织效率（一个公司跑得更快）	协作效率（多个公司协作得更顺畅）
信任域	单一信任域（一个老板、一套规则）	跨信任域（各自独立、互不隶属）
权限体系	统一管理（公司 IT 说了算）	各自独立 + 对等协商协议
信息可见性	企业内可控（老板决定谁看什么）	高度敏感（核心数据不能泄露给对方）
责任归属	内部追责	跨企业法律/商业责任
可控可信的要求	**最佳实践**（审批、权限、审计是标配）	**刚性要求**（身份认证、信息分级、双方审计是前提）

企业内部 Agent 的可控可信决定了“能用多好”；跨企业 Agent 的可控可信决定了“能不能用”。

2.2 从企业协作的现状看 B2B Agent 的必要性

今天两个企业合作，大多数流程是：

```
A公司 → 发邮件 → 发微信 → 电话 → Excel → PDF → 人工确认 → B公司
```

大量工作其实只是**信息同步 + 确认 + 协商**。Agent 非常适合做这件事情。

未来更可能是：

```
A企业Agent → 自然语言协商 → 调用系统 → 确认规则 → B企业Agent
```

人只在关键节点审批。

2.3 哪些跨企业业务最适合 Agent 化?

第一类：采购（最大的 B2B Agent 机会）

今天的采购流程：采购员 → 找供应商 → 询价 → 比较价格 → 确认交期 → 合同 → 下单。

未来：

```
采购Agent → 向20家供应商Agent询价 → 自动分析 →
比较库存 → 比较交期 → 推荐最优方案 → 提交老板审批
```

供应商 Agent 负责回复：是否有库存、是否能生产、是否接受价格、是否支持分批交货。整个过程可能十分钟完成。

第二类：供应链协同

一个汽车零件涉及：主机厂 → 一级供应商 → 二级供应商 → 三级供应商。今天全部靠邮件、Excel、SAP、电话。

未来：

```
主机厂Agent → 通知需求变化 → 供应商Agent重新排产 →
物流Agent重新安排 → 自动回复风险
```

整个供应链实时同步。

第三类：跨企业销售

客户提出“我要 500 台设备”，销售 Agent 自动协调：

```
需求分析 → 库存Agent → 采购Agent → 物流Agent → 报价Agent → 生成方案
```

客户 Agent 继续就价格、付款方式、交期、售后进行多轮谈判——很多轮可以自动完成。

第四类：物流

物流涉及货代、船公司、保险、仓储、报关、清关、运输——全部都是不同的企业。未来：

```
物流Agent → 通知港口 → 通知海关 → 通知仓库 → 通知卡车 → 同步状态
```

所有企业共享状态，但**各自的数据权限严格隔离**。

第五类：金融（企业贷款）

企业 Agent 自动准备财务数据、ERP 数据、纳税记录、合同、发票；银行 Agent 自动审核。最后人工签字。

2.4 跨企业 Agent 对治理体系提出了哪些更高要求?

在单个企业内部，Agent 治理是 IT 管理问题的自然延伸——CTO 可以制定统一规则，所有的 Agent 都在同一个组织架构下运行。内部 Agent 同样需要审批、权限、审计等治理能力，但这一切在单一信任域内可以相对高效地实现。

跨企业 Agent 协作则在内部治理的基础之上，叠加了全新的挑战：

- **没有中央权威**：A 公司的 Agent 和 B 公司的 Agent 互不隶属，谁来决定“谁能干什么”?
- **信息不对称是刚需**：供应商的库存成本、客户的预算上限、物流的内部排期——这些都是各自的核心商业机密，绝不能在对等协作中全量暴露。
- **责任归属复杂**：如果 A 公司的采购 Agent 根据 B 公司供应商 Agent 的错误信息下了单，谁负责?

- **身份认证难度高**：你怎么知道和你通信的 Agent 真的代表它声称的那家公司？它有没有被篡改？
- **执行后果不可逆**：一个内部 Agent 犯了错，影响的是自己公司；一个跨企业 Agent 犯了错，可能引发连锁的跨企业商业纠纷。

因此，跨企业 Agent 需要的不是“更聪明的 LLM”，而是一套独立于任何单一方、可以被所有参与方信任的治理基础设施。

2.5 从“自由智能”到“授权智能”：贯穿内部与跨企业的核心原则

当前行业对 Agent 的技术定义是：

> Agent = LLM + Memory + Tools + Planning

但从企业级（无论是内部还是跨企业）的角度，必须重新定义为：

> **Agent = Intelligence（智能）+ Control（控制）+ Responsibility（责任）**

这意味着：

- **LLM 提供的是动力，而不是治理**——Agent 不是自由行动的“数字员工”，而是受严格约束的“企业数字代表”。在企业内部这是最佳实践，在跨企业场景中这是**刚性要求**。
- **能力（Ability）与权限（Authority）必须彻底分离**——Agent 不是因为“会”所以去做，而是因为“被授权”才可以做。
- **LLM 越聪明，治理层越重要**——一个能够自动与几十家供应商 Agent 谈判、调价、签约的 Agent，如果没有治理约束，商业风险将指数级上升。这一规律在企业内部和跨企业协作中同样成立。

2.6 零信任原则：企业级 Agent 的安全基石

企业级 Agent（无论是内部还是跨企业）的底层安全哲学必须是**零信任（Zero Trust）**：

> **Agent 默认没有任何能力，所有能力都是授权获得的。在跨企业场景中，还需加上：对来自其他企业的 Agent，默认不信任任何信息，所有信息都需要验证。**

对应四个根本问题：

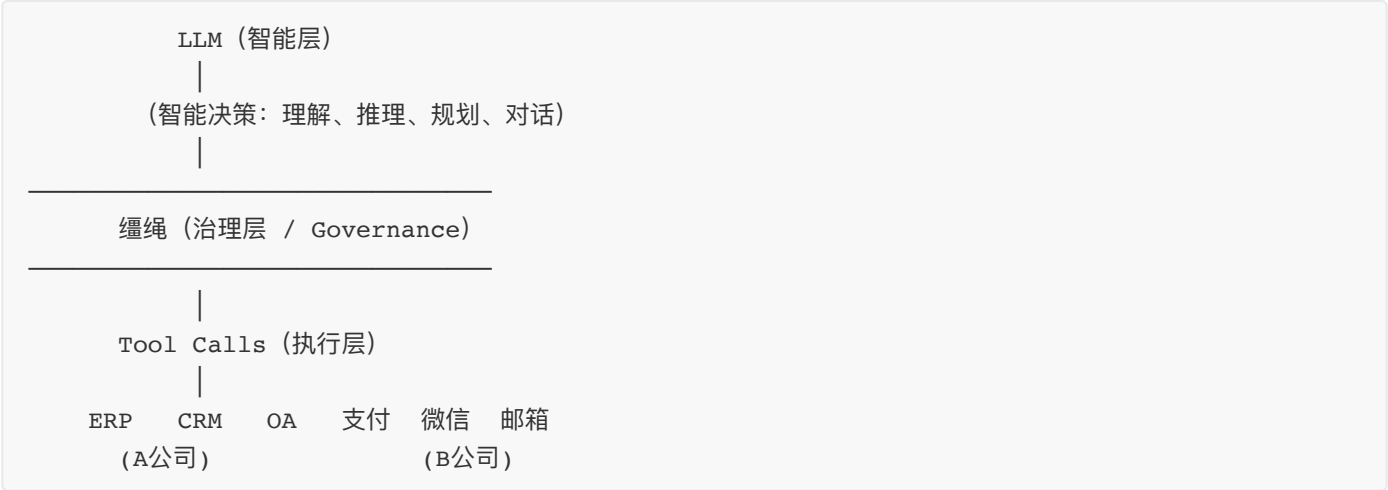
1. **它可以干什么？**——Allow List（能力白名单），必须由所属企业显式声明
2. **它绝对不能干什么？**——Deny List（禁止清单），只能读不能写、只能询价不能签约
3. **没有明确授权怎么办？**——默认禁止（Default Deny），不因“AI 觉得方便”而放行
4. **它干过什么？**——完整的跨企业 Decision Trace（决策链追踪），双方可审计

---

3. 理论基础：Agent 治理模型（CAEM）——从内部治理到跨企业扩展

3.1 三层架构

将企业级 Agent 拆分为三个清晰的分层，这一架构同时适用于企业内部和跨企业场景：



- **智能层（Brain）**：由模型厂商提供，负责理解、推理、规划和对话。
- **治理层（Reins）**：决定 Agent 能干什么、什么时候能干、不能干什么。在企业内部，治理层是规则引擎；在跨企业场景中，治理层还需支持对等治理协议——它不是内部规则的简单外延，而是协作控制平面的升级。
- **执行层**：真正调用企业系统。在跨企业场景中，A 公司的 Agent 调用 A 公司的 ERP，B 公司的 Agent 调用 B 公司的 ERP，**互相不能直接跨域访问对方的系统**。

真正控制协作风险的是中间这一层——治理层。在企业内部，它是管理规则的执行引擎；在跨企业协作中，它升级为独立于任何单一企业的"协作控制平面"。

3.2 受控智能体执行模型（CAEM：Controlled Agent Execution Model）

CAEM 包含四条不可妥协的原则，**在企业内部是治理基线，在跨企业场景中每条都进一步升级**：

原则一：Default Deny（默认拒绝）

任何未明确授权的能力，一律禁止执行。在跨企业场景中，这意味着：A 公司的 Agent 不能访问 B 公司的任何系统，除非 B 公司显式授权。Agent 不能因为"谈判需要更多数据"而自行请求对方开放信息。

原则二：Explicit Capability（显式能力）

Agent 只能执行能力清单中的动作，且这些能力必须**对协作方可见**。每项能力配置：权限要求、适用条件、审批要求、可调用系统、**对哪些外部 Agent 开放**。

原则三：Decision Trace（决策链追踪）

不仅记录执行了什么，更记录：为什么这样理解？如何规划？依据哪些事实？如何评估风险？最终如何验证结果？在跨企业场景中，**决策链必须是双方（或多方）可验证的**——这比传统 API 日志更深层，它记录的是跨企业推理链（Cross-Enterprise Reasoning Log）。

原则四：Human Accountability（人类最终负责）

对于高风险或超出授权范围的跨企业行动，必须进入**双方**人工审批。每一笔跨企业交易都有明确的责任人和审批记录。Agent 永远不能绕过这一环节。最终的责任人始终是人。

3.3 治理六层级

治理不是一根单一的"缰绳"，而是一套完整的控制系统：

层级	名称	核心问题	跨企业场景示例
第一层	Rule（规则）	固定边界是什么?	付款不能超过 50 万；只能向已认证供应商询价
第二层	Policy（策略）	业务条件是什么?	VIP 客户可以优惠 10%；新供应商需要额外审批
第三层	Permission（权限）	数据访问边界?	销售 Agent 不能透露成本数据给客户 Agent
第四层	Verification（验证）	事实是否可信?	对方 Agent 说库存 100 件 → 要求提供可验证凭证
第五层	Approval（审批）	是否需要人工确认?	跨企业合同签约 → 双方法务审批
第六层	Rollback（回滚）	执行错误怎么办?	跨企业订单错误 → 双方确认回滚协议

3.4 决策链审计的四层模型

审计不是记录 API 调用，而是记录完整的跨企业推理链：

第一层：Think（思考）——Agent 为什么这么理解？例：对方 Agent 说"尽快发货" → 本企业 Agent 理解为"优先级 = P1"，记录理解依据与来源。

第二层：Plan（规划）——Agent 制定了怎样的执行计划？涉及哪些外部系统?

第三层：Action（执行）——具体调用了哪些系统？哪些是本公司系统，哪些是与外部 Agent 的交互?

第四层：Evaluate（评估）——执行后是否成功？对方是否确认？如果失败，是己方原因还是对方原因？是否需要重新协商?

3.5 七个不可回避的问题

对跨企业 Agent 的每一次 Action，都必须能够回答：

1. **Who** —— 是谁？代表哪个企业？（Identity）
2. **Why** —— 为什么要做？基于哪个跨企业任务？（Task / Intent）
3. **What** —— 准备做什么？是否在双方授权范围内？（Capability）
4. **Based on What** —— 依据什么信息？信息来源是本企业还是对方企业？（Context）
5. **Who Approved** —— 谁授权、谁批准？是单方审批还是双方审批？（Governance）

6. **What Happened** —— 最终执行了什么？对方如何响应？（Execution）
7. **How Do We Prove It** —— 事后如何向己方和对方证明全过程？（Audit）

如果一个系统能完整回答这七个问题——而且答案是双方可验证的——那么这个 Agent 就已经具备了跨企业"可控可信智能体"的基本特征。

3.6 Bounded Intelligence（边界智能）：CAEM 的哲学基础

CAEM 的四条原则——Default Deny、Explicit Capability、Decision Trace、Human Accountability——背后有一个共同的哲学基础：**边界智能（Bounded Intelligence）**。

> **每个企业 Agent 都必须运行在明确规定的能力边界、权限边界和决策边界之内。**
>
> *Every Enterprise Agent must operate within an explicitly defined capability boundary, authorization boundary, and decision boundary.*

为什么边界比能力更重要?

现在很多 Agent 框架关注的核心问题是：

> *How can the Agent do more?*（如何让 Agent 做更多事情？）

而 TEA 关注的核心问题是：

> *How do we ensure the Agent only does what it is supposed to do?*（如何确保 Agent 只做它应该做的事情？）

这不是一个保守的技术选择，而是企业级软件的刚性需求。企业真正担心的通常不是 Agent 能力不够，而是 **Agent 在没有适当授权、没有充分依据或超出职责范围时替企业做了决定**。一个没有边界的 Agent，能力再强也无法进入企业核心业务；一个边界清晰的 Agent，才能从"智能助手"成长为"可信的数字员工"。

边界智能的三个维度：

维度	含义	CAEM 对应
能力边界	Agent 能做什么、不能做什么	Explicit Capability（Allow List + Deny List）
权限边界	Agent 在什么条件下可以执行	Default Deny + Governance Rules + Approval
决策边界	Agent 可以自主决定什么、什么必须人工确认	Human Accountability + 输出分级

边界智能的核心推论：人工核实不是模型能力不足，而是企业治理要求。

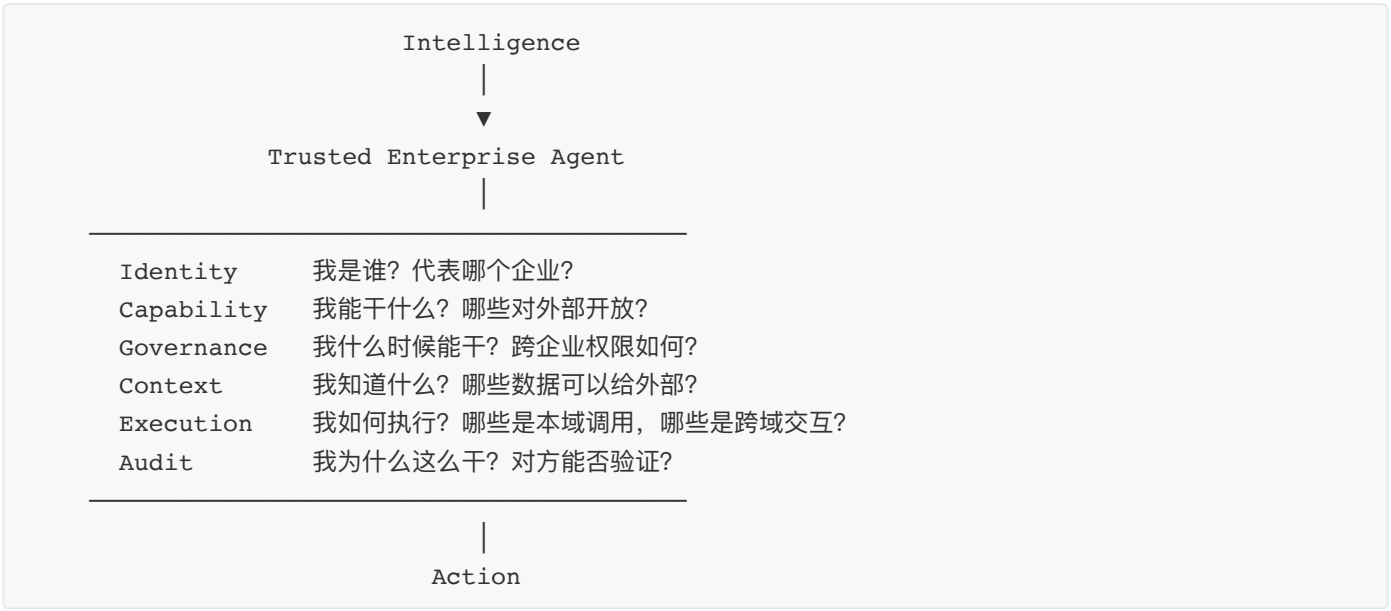
当财务 Agent 分析发票（✅ 能力边界内）但准备付款（❌ 超出决策边界）时，系统自动升级为"🟡 需人工核实"。这不是因为 Agent "不够聪明"、不会付款——它能做，但**不被允许做**。这正是"能力（Ability）与权限（Authority）必须彻底分离"的直观体现。

---

4. 技术架构：六层运行时模型——从本域执行到跨域协作

4.1 完整模型

一个企业级可信智能体的完整生命周期包含六层，在企业内部和跨企业场景中共享同一架构框架，但在跨企业场景中每层都有额外的扩展要求：



4.2 各层详细定义（含企业内部基础 + 跨企业扩展）

第一层：Identity（身份）

核心问题：我是谁？代表哪个企业？对方 Agent 如何验证我的身份？

```
采购Agent
  所属企业：A制造有限公司（统一信用代码：XXX）
  部门：采购部
  负责人：采购经理 张三
  代表：A公司采购部门
  工号：AGT-PUR-A001
  可信等级：L2
  对外身份凭证：由XX CA机构签发
```

每个跨企业可信智能体必须拥有数字工牌，包含的不仅是内部身份信息，更重要的是**对外的可验证凭证**——就像 HTTPS 证书，对方可以通过 CA 验证这个 Agent 确实代表它所声称的企业。

第二层：Capability（能力）

核心问题：我能干什么？哪些能力对协作方开放？

```
√ 查询本企业库存
√ 创建本企业采购订单
√ 向已认证供应商 Agent 发起询价
√ 接收并比较报价
× 访问供应商内部系统
× 修改供应商报价
× 直接付款
× 查看供应商成本数据
```

关键区别：在跨企业场景中，能力白名单需要区分"本域能力"和"对外能力"——有些能力只在企业内部使用，有些可以向外部 Agent 暴露。

第三层：Governance（治理）——控制平面

核心问题：在跨企业协作中，我什么时候能干？

治理是**贯穿整个跨企业运行过程的控制平面**：

```
Governance（跨企业治理）
├── Policy（策略）：跨企业订单 > 100 万必须双方审批
├── Permission（权限）：外部 Agent 只能读取我授权的数据字段
├── Approval（审批）：跨企业签约 → 本方 + 对方法务审批
├── Communication（通信）：只能与已认证的企业 Agent 通信
├── Information Classification（信息分级）：L0-L4，不同外部企业不同密级
├── Risk Control（风险）：异常询价模式检测、反欺诈
└── Delegation（授权）：临时授权令牌，跨企业任务完成后自动回收
```

第四层：Context（上下文）

核心问题：我知道什么？哪些信息可以透露给外部 Agent？

这是跨企业 Agent 与内部 Agent 差异最大的地方。Agent 不是知道所有东西，而是根据对方企业身份、双方协议和密级获得有边界的上下文：

- 哪些本企业信息？（内部完整）
- 哪些对协作方可见？（经授权脱敏后的聚合信息）
- 哪些对协作方不可见？（成本、利润、工资等核心机密）
- 哪些来自协作方？（对方提供的信息，需要验证）

例：A 公司采购 Agent 向 B 公司供应商 Agent 询价，B 公司的 Agent 可能知道精确库存 350 件（A 仓 100、B 仓 200、C 仓 50），但只回复"可供应约 350 件"。A 公司不需要知道 B 公司的仓储分布。

第五层：Execution（执行）

核心问题：我如何执行？哪些是内部操作，哪些是跨企业交互？

在跨企业场景中，执行被分为两个域：

- **本域执行**：调用本企业 ERP、CRM、OA——与内部 Agent 相同
- **跨域执行**：向外部企业 Agent 发送请求、接收响应、交换结构化上下文——这是全新的能力

所有跨域执行动作必须经过 Governance 层，且**不能直接调用对方企业的内部系统**。A 公司的 Agent 调用的是 B 公司 Agent 的**对等接口**，而不是 B 公司的 ERP。

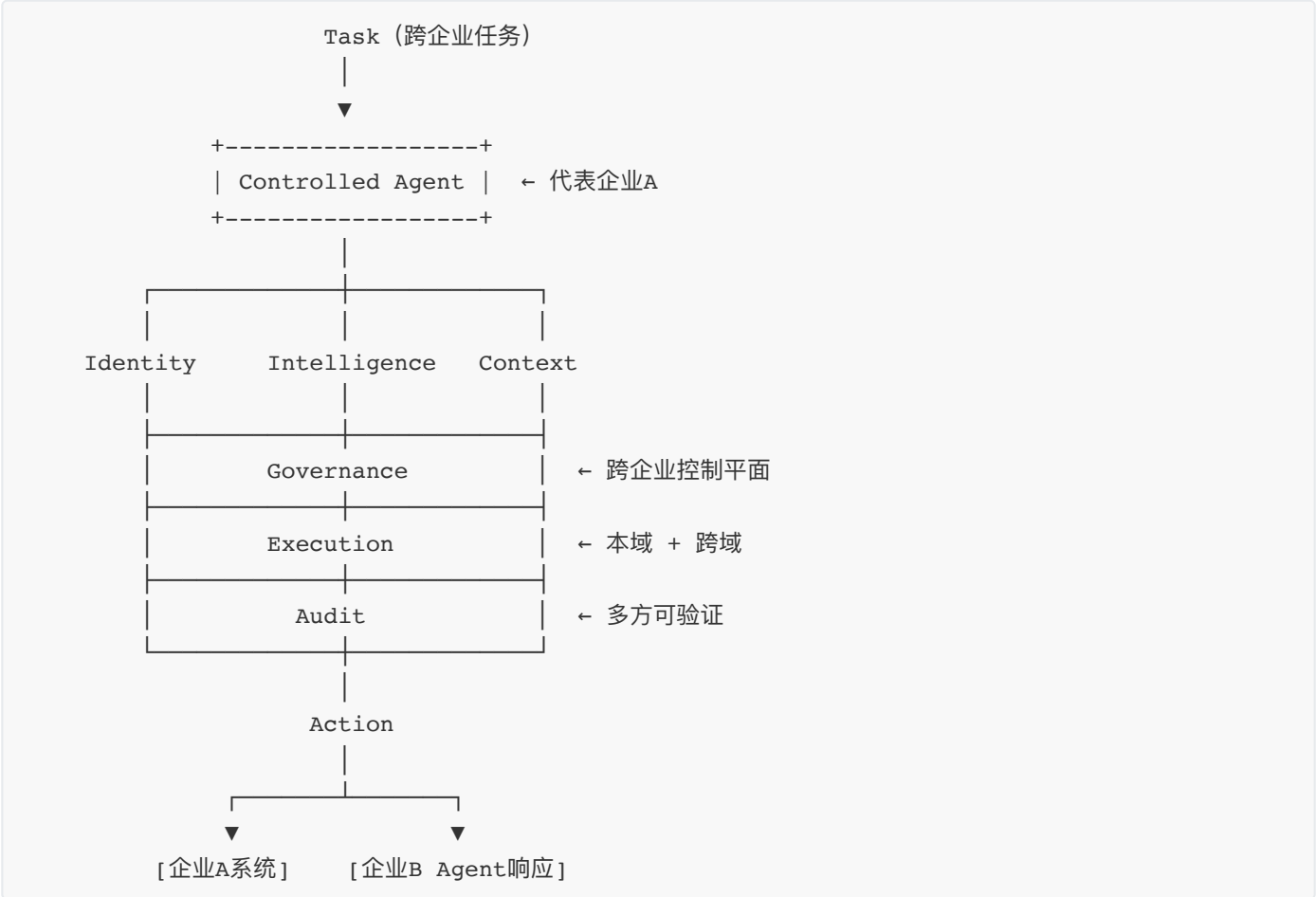
第六层：Audit（审计）

核心问题：我为什么这么干？双方能否各自独立验证？

跨企业审计的核心要求是：**审计记录必须双方（或多方）可验证**。这意味着：

```
跨企业决策链：
  [A公司] 依据：采购需求 PO20260012
  [A公司] 为什么向B公司询价？→ B公司在合格供应商名录中
  [A公司] 为什么接受此报价？→ 三家比价中最低
  [B公司] 为什么报此价格？→ 基于当前库存和成本，经审批
  [B公司] 为什么确认交期？→ 生产排期可用
  最终执行：[A] 创建采购订单 + [B] 确认接单
  双方签章验证：A公司审批人 ✓ | B公司审批人 ✓
```

4.3 架构总览



核心设计理念：Intelligence 不是中心，Action 才是中心。Governance 作为跨企业控制平面贯穿整个生命周期。执行被明确划分为本域和跨域两个隔离的空间。所有跨域交互都经过对等的治理协议。

---

5. Agent 通信治理：从身份到授权的七层模型

5.1 从“网络治理”到“组织治理”再到“跨组织治理”

很多人设计 Agent 通信时默认：

```
Agent A  <---> Agent B
```

大家平等通信。但实际场景中，通信治理的复杂度是逐级递增的。

企业内部是**组织治理**——信息不对称，层级分明，不同岗位不同权限。Agent 之间的通信需要遵循组织层级：采购 Agent 不能直接访问 CEO Agent，财务 Agent 的数据对销售 Agent 不可见。

跨企业则是**跨组织治理**——不仅企业内部有层级，企业之间也有不对等的协作关系：

> **A 公司的 Agent 和 B 公司的 Agent 之间，没有上下级关系，但有明确的协作边界。**

在跨企业场景中，Agent 在内部组织治理的基础上，额外受到三重约束：

1. 本企业给它什么权限?
2. 对方企业允许它看到什么?
3. 双方的协作协议允许它做什么?

5.2 跨企业 Agent 通信的三层权限

第一层：能不能通信（Communication Permission）

```
A公司采购Agent → × → B公司CEO Agent（不可越级通信）
A公司采购Agent → √ → B公司供应商Agent（通过双方企业已建立的协作通道）
```

通信权限不仅由本企业决定，还受对方企业控制。B 公司可以拒绝 A 公司某类 Agent 的通信请求。

第二层：可以看什么（Visibility）

```
B公司库存Agent 已知：A仓 100，B仓 200，C仓 50
A公司采购Agent 可见：可采购库存 350（脱敏后的聚合，不暴露仓储分布）
```

信息在传递前就被过滤。A 公司的 Agent 收到的不是原始数据，而是**经过对方治理层脱敏后的授权信息**。

第三层：可以知道多少（Depth）

```
A公司 CEO Agent 问：为什么这次采购成本比上次高15%?
B公司 供应商Agent 回答：原材料成本上涨 + 交期紧迫（给出分析依据）

A公司 普通采购Agent 问同样问题：
B公司 供应商Agent 回答：当前报价为XX万元（不暴露定价逻辑）
```

同一个问题，同一个对方 Agent，但来自不同层级/企业的询问者，得到不同深度的答案。

5.3 跨企业信息分级体系

等级	定义	对谁可见	跨企业示例
L0	公开信息	所有企业 Agent	产品目录、公开报价
L1	协作共享	已签约协作方	订单状态、交期、物流状态
L2	协作受限	特定协作方 + 特定层级	供应商标价（仅采购可见）、质量数据（仅品控可见）
L3	企业机密	仅本企业管理层	利润、预算、成本构成
L4	核心机密	仅极少数本企业授权者	收购计划、融资、定价策略

每次跨企业 Agent 通信，双方治理层依次判断：

```
请求Agent身份 → 所属企业 → 双方协作协议 → 请求密级 → 对方企业信息等级 →
是否允许 → 允许多少 → 是否需要本方审批 → 返回（脱敏后的）信息
```

5.4 结构化通信协议：Agent 之间不应该传自然语言

跨企业 Agent 之间传递的不应是自然语言（太模糊、不可审计），而应该是**授权后的结构化上下文**：

```
{
  "message_id": "msg_20260629_001",
  "sender": {
    "agent_id": "AGT-PUR-A001",
    "enterprise": "A制造有限公司",
    "enterprise_ca": "CN=AManufacturing,O=...,C=CN",
    "role": "purchase_agent",
    "clearance_level": "L2"
  },
  "request": {
    "type": "rfq",
    "action": "price_inquiry",
    "scope": "purchase",
    "items": ["SKU-001", "SKU-002"],
    "quantity": 500,
    "delivery_window": "2026-07-15/2026-07-30"
  },
  "authorization": {
    "granted_by": "采购经理 张三",
    "delegation_token": "dt_abc123",
    "expires_at": "2026-06-29T18:00:00Z",
    "max_budget_per_unit": null
  },
  "signature": "sha256:..."
}
```

接收方 Agent 根据 `sender` 身份、`request` 内容、双方**协作协议**，以及本企业的治理规则，决定返回什么信息。整个过程不是 LLM "随意回答"，而是**规则驱动 + 授权校验 + 返回脱敏结果**。

5.5 临时授权令牌（Delegation Token）

当需要跨企业执行特定任务时，颁发有时效、有范围的**临时授权令牌**：

```
授权人：A公司 CEO Agent
被授权人：A公司 采购Agent
任务：向10家供应商询价并比价
权限：读取 L2 供应商数据，向 L1-L2 外部企业Agent发起询价
有效期：2小时
不可转授权
不可签约（只能询价，不能下订单）
不可查看本企业成本数据
```

2 小时后权限自动失效。对方企业也会验证这个 Token 的有效性——如果过期了，B 公司的 Agent 可以拒绝响应。

5.6 跨企业 Agent 治理的七层模型

层	治理内容	核心问题	跨企业特有要求
Identity	Agent 身份	你是谁？代表哪个企业?	企业 CA 签发、双方可验证
Capability	能力白名单	你能做什么?	区分本域能力与对外暴露能力
Authorization	授权机制	谁允许你做?	本方授权 + 对方准入
Communication	通信控制	你可以和谁沟通?	基于企业间协作协议
Information Clearance	信息分级	你可以知道多少?	跨企业 L0-L4，对方决定可暴露多少
Decision Trace	决策审计	为什么这么决定?	双方可各自独立验证
Execution Control	执行控制	谁批准最终执行?	高风险跨企业操作需双方审批

5.7 会议范式的变革：从企业内部开会到跨企业 Agent 协商

会议场景的变化特别能说明跨企业 Agent 的价值：

- **第一阶段（已发生）**：Agent 作为企业内部会议秘书——录音、纪要、任务分发。
- **第二阶段（未来 2–5 年）**：Agent 代表部门参与企业内部会议——实时拉取数据、分析、建议。
- **第三阶段（关键变革）**：**Agent 代表企业参与跨企业协商**——采购 Agent 与供应商 Agent 自动谈判价格、交期、付款条件，双方 CEO 只做最终批准。

- **第四阶段（未来）**：跨企业 Agent 自主协调——每天凌晨，A 公司的计划 Agent 与 B 公司的供应 Agent、C 公司的物流 Agent 自动同步当日计划，老板醒来只收到整合后的“今日经营建议”。

会议从一种同步沟通方式，演变成一种可计算、可追踪、可自动执行的跨企业决策流程。

---

6. 关键业务场景与落地路径

6.1 场景一：跨企业 Agent 身份认证与准入（基础中的基础）

在所有 B2B Agent 应用之前，必须先解决一个问题：**你怎么知道和你通信的 Agent 真的代表它声称的那家企业?**

类似于互联网从“裸奔的 HTTP”进化到“HTTPS + CA 证书体系”，Agent 互联网也需要一套：

- Agent 身份注册（类似域名注册）
- Agent 身份验证（类似 SSL 证书签发）
- 企业准入清单（白名单，类似防火墙规则）
- Agent 吊销机制（证书到期或被撤销）

这是跨企业 Agent 最基础、最刚需、也最容易被忽视的基础设施层。

6.2 场景二：跨企业 Agent 审批中心（B2B Approval Gateway）

跨企业场景中，最让企业担心的不是 Agent 回答错，而是 **Agent 代表企业做了不该做的承诺**。

```
A公司Agent准备向B公司下单
    ↓
生成跨企业执行计划
    ↓
风险评估（金额、对方可信度、历史履约率）
    ↓
需要审批?
    ↓
A公司采购经理审批 + B公司销售经理确认
    ↓
双方Agent执行 + 双方签章
```

产品定位：不是聊天平台，不是 Agent 开发平台，而是跨企业 AI 行动的最后一道关口。

6.3 场景三：跨企业 Agent 操作审计（B2B Audit Trail）

跨企业操作的关键是“出了事能说得清是谁的责任”：

- 哪个企业的哪个 Agent? 在什么时间?
- 基于什么授权?
- 向哪个企业的哪个 Agent 发起了什么请求?
- 对方如何响应?
- 最终执行了什么?
- 双方谁批准的?

这不只是日志，而是双方可验证的、具有法律效力的跨企业决策链。 类似于今天国际贸易中的信用证和提单，但对象从“纸质单据”变成了“Agent 决策记录”。

6.4 场景四：跨企业信息分级与权限管理

不同协作关系，不同信息可见度：

- 长期战略供应商：可以看到库存预测、生产计划
- 一次性询价供应商：只能看到当前询价的物料规格
- 物流合作伙伴：只能看到需要运输的货物信息，看不到价格
- 银行：可以看到财务报表，但看不到客户名单

这套分级不是静态的，而是基于协作协议动态调整的。

6.5 场景五：Agent 验证引擎（B2B Verification Engine）

跨企业场景中，绝不能相信对方 Agent 的“口头回答”：

```
B公司Agent回答：库存还有 350 件
A公司Agent：请提供可验证凭证
    ↓
B公司Agent：生成ERP库存快照 + 数字签名
    ↓
A公司验证引擎：验证数字签名有效性 and 库存快照真实性
    ↓
验证通过 → “已验证：B公司当前可供应库存 350 件”
验证失败 → “未验证，要求B公司人工确认”
```

这意味着跨企业 Agent 的输出永远有可验证的事实来源作为支撑。

6.6 场景六：跨企业 Agent 协作调度（B2B Coordinator）

当企业 A 有 10 个 Agent、企业 B 有 10 个 Agent，它们之间可能的交互组合爆炸式增长：

- 两个 Agent 互相调用，无限循环
- 多个 Agent 对同一笔交易重复操作
- 优先级冲突（A 公司的紧急订单 vs B 公司的产能上限）

需要一个**跨企业 Coordinator**——不属于任何单一企业，而是作为中立的协作调度层。

---

7. 商业模式：Enterprise Agent Network

7.1 最大的机会不是开发 Agent，而是成为 Agent 之间的连接层

如果每家公司都有一个“企业数字代表（Enterprise Agent）”，企业之间通过 Agent 完成大量标准化协作，那么：

> **100 万家公司 → 每家公司一个 Agent → 互相通信**

这会形成一个类似于互联网的全新协作网络。而真正的商业机会在于：**谁来做这个网络的连接层?**

不做“Agent 商店”（卖 Agent），而是做：

> **Enterprise Agent Network（企业 Agent 网络）**
>
> 让不同企业的 Agent 能够：
>
> - **互相发现**：采购 Agent 找到供应商 Agent（类似 DNS）
> - **互相认证**：验证对方身份和可信等级（类似 SSL/CA）
> - **安全通信**：结构化、可审计、有密级的通信（类似 HTTPS + 更高级协议）
> - **可信协作**：协商、签约、执行、结算（跨企业业务流程）

7.2 不做“Agent 治理平台”，做“B2B Agent Infrastructure”

正如安全行业：没有人买“网络安全”，大家买的是防火墙、堡垒机、数据库审计、漏洞扫描。

同样，不要在第一天就做一个“Agent 治理平台”。而是围绕一个具体的跨企业业务场景，把治理做到极致。

推荐切入点优先级：

1. **跨企业 Agent 身份认证服务**：最基础、最刚需——没有身份就没有协作
2. **跨企业 Agent 审批中心（B2B Approval Gateway）**：所有跨企业高风险操作的必经关口
3. **跨企业 Agent 审计服务（B2B Audit Trail）**：双方可验证的决策链记录
4. **行业垂直 B2B Agent 网络**：比如专注制造业采购 Agent 网络、专注跨境物流 Agent 网络

7.3 互联网演进的三阶段类比

- **第一阶段：网页互联**（浏览器访问网站）——基础设施是 DNS + HTTP + 浏览器
- **第二阶段：API 互联**（系统调用系统）——基础设施是 REST + OAuth + API Gateway
- **第三阶段：Agent 互联**（企业之间的智能协作）——需要新的基础设施：Agent 身份认证 + 权限协议 + 上下文共享 + 流程编排 + 审计

每一代互联方式都诞生了新的基础设施巨头。Agent 互联时代同样如此。

7.4 下一代基础设施的完整图景

如果 Agent 之间真的开始大量通信，就会出现一整套新的需求，构成完整的“Agent 互联网”基础设施栈：

最大的商业机会未必是开发更多单点 Agent，而是成为 Agent 之间的连接层：定义协作协议、身份认证、权限体系、上下文共享和流程编排。谁能让不同企业的 Agent 安全、低成本、高可信地完成业务协同，谁就有机会成为下一代企业互联网的核心基础设施。

7.5 责任边界：云厂商 vs 企业 Runtime

在 Agent 基础设施的演进中，一个关键问题是：**哪些能力应该由云厂商提供，哪些必须由企业自己的 Runtime 负责?**

层	谁负责	核心目标	为什么?
GPU 调度	云厂商	GPU 利用率、SLA、成本	企业不应该关心哪块 GPU 空闲——就像不关心 HTTP 请求落到哪台服务器
模型服务	模型厂商	模型能力、推理性能	训练和推理优化是模型厂商的核心竞争力
Model Routing	**企业 Runtime**	为每个业务任务选最优模型	云厂商不知道你的业务语义——同一句话是法律还是财务，只有企业知道
Workflow 编排	**企业 Runtime**	Agent 协作、任务拆分	业务流程是企业独有的组织知识
Cost per Task	**企业 Runtime**	每完成一个业务任务成本最低	企业关心的是“一个客服工单多少钱”，不是“100万 Token多少钱”
Business KPI	企业	ROI、收入、效率	最终的业务判断永远在企业手中

核心结论：GPU 调度和模型服务会像今天的 IaaS 一样被云厂商标准化；而 Model Routing、Workflow 编排、Cost-per-Task 优化这些**离业务越近的能力，越应该由企业 Runtime 承担**。这正是 TEA Runtime 的定位——它不取代云厂商，而是建立在云厂商之上的业务执行层。

7.6 Inference Planner：从模型路由到推理规划

TEA Runtime 的核心能力不是简单的“模型路由”，而是**推理规划（Inference Planning）**。

为什么 Model Router 不够?

传统的模型路由是一个简单的映射：`Task → Model`。但 Agent 时代的任务远不是"选一个模型"能解决的。一个用户请求可能被拆分为搜索（小模型）、推理（强模型）、校验（小模型）、润色（小模型）——**不是整个任务交给一个模型，而是整个任务拆成多个子任务，每个子任务选择最合适的模型**。

这正是 Inference Planner 的核心职责：

```
用户请求
    │
    ▼
Intent Agent（理解意图）
    │
    ▼
Task Planner（拆分任务）
    │
    ├── 搜索 → 小模型（7B）
    ├── 摘要 → 小模型（7B）
    ├── 推理 → 强模型（DeepSeek V4）
    ├── 校验 → 校验模型
    └── 润色 → 小模型（7B）
    │
    ▼
结果合成
```

路由准确性的保证机制：

没有任何系统能保证每次路由都"选对"。TEA 的做法不是追求"永不出错"，而是建立一套持续改进的机制：

1. **数据驱动路由**：Runtime 持续记录每次任务的模型选择、成本、耗时、用户满意度，积累数据后自动学习最优策略。不是因为工程师觉得 DeepSeek 适合合同审核，而是**数据证明**它在这个场景下成本最低且满意度最高。
2. **置信度驱动的降级**：当 Router 对选择的置信度低于阈值（如 < 70%），不赌单一模型，而是自动升级为多模型并行 + 投票，甚至升级为人工确认。就像自动驾驶——没把握时交给飞行员。
3. **Benchmark Agent（基准测试 Agent）**：一个专门的 Agent，每天凌晨自动执行数万个标准测试问题，持续评估各模型在销售、法务、研发、财务等领域的最新表现，形成每日更新的排行榜，Runtime 据此自动调整路由策略。

Inference Planner 的最终形态：

它做的不只是"选模型"，而是规划整个推理过程——理解意图、拆分任务、分配模型、编排执行、校验结果、根据反馈重新规划。它本质上已经成为一个**高智商的 Planning Agent**。这也是为什么它必须由企业 Runtime 承担，而非云厂商：理解业务语义、掌握企业上下文、遵守组织治理规则——这些只有企业自己能做到。

7.7 AI 应用开发三层架构

基于以上分析，TEA 提出企业 AI 应用开发的**三层架构**，这既是技术架构，也是组织分工：

```
业务层（Business Capabilities）
————————————————————————————
合同审核 / 客服回复 / 销售分析
采购审批 / 知识查询 / 代码生成
```

```
          │
          ▼
业务能力接口（Business Capability API）
──────────────────────────────────
reviewContract() / replyCustomer()
analyzeSales()  / approveExpense()
          │
          ▼
     TEA Runtime
──────────────────────────────────
Intent / Planning / Policy
Routing / Memory / Audit
          │
          ▼
   模型层（Foundation Models）
──────────────────────────────────
GPT / DeepSeek / Qwen / GLM
Claude / Llama / 华为昇腾云
```

这一架构遵循的核心设计原则是序章中确立的 **Business First, Models Second**。业务层只调用业务能力接口，完全不知道底层用的是哪个模型。模型可以随时替换——今天 GPT，明天 DeepSeek，后天华为昇腾云——业务代码零修改。

企业表达的不是"用哪个模型"，而是业务策略（Business Policy）：

```
合同审核：
    quality: high
    budget: medium
    latency: 10s

客服回复：
    quality: medium
    budget: low
    latency: 2s    # 必须秒回
```

Runtime 把这些策略翻译成具体的执行方案——选择哪个模型、是否拆分任务、是否多模型投票、是否低峰执行。**模型只是实现策略的一种资源，策略才是企业真正稳定的资产。**

这一定位的核心价值：当 DeepSeek、OpenAI、华为昇腾云或未来任何新模型出现时，企业无需修改业务逻辑。Runtime 根据策略和数据自动重新选择最优执行方案。这就像数据库从 Oracle 换成 MySQL——如果代码是通过 ORM 写的业务逻辑，迁移成本极低；如果是裸 SQL，就需要大规模重写。**TEA Runtime 就是 AI 时代的 ORM 层——隔离业务逻辑与底层模型的持续演进。**

---

8. 结语：从 Agent 开发到 Agent 互联——两个战场，同一套治理逻辑

8.1 核心论断

本白皮书从跨企业 Agent 协作的独特视角出发，论证了以下核心论断：

1. **AI 的商业价值不在模型本身，而在"模型 + 具体场景 + 工作流"**。大模型是基础设施，小应用才是价值载体。
2. **"可控可信"同时适用于企业内部 Agent 和跨企业 B2B Agent，但两者对治理体系的要求处于不同层级。** 在企业内部，可控可信是"最佳实践"——审批、权限、审计是 Agent 安全运行的标配；在跨企业协作中，可控可信升级为"刚性要求"——没有身份认证、信息分级、双方审计，协作根本无法开始。
3. **跨企业 Agent 在内部治理的基础之上叠加了全新维度的挑战。** 没有中央权威、信息不对称是刚需、责任归属复杂、身份认证难度高——这些都要求治理体系从"企业内规则引擎"升级为"跨企业协作控制平面"。
4. **企业级 Agent 的核心是：能力与权限分离、默认拒绝、决策链追踪、人类最终负责。** 这四条原则（CAEM）在企业内部是治理基线，在跨企业场景中是安全运行的前提条件。
5. **Agent 间的通信需要超越自然语言，走向结构化、有密级、可验证的协议。** 在企业内部，这是效率提升；在跨企业协作中，每一次通信都必须包含身份凭证、授权令牌、密级标签和数字签名——不是"聊天"，而是"企业间的正式商务往来"。
6. **最大的商业机会不是开发 Agent，而是成为 Agent 之间的连接层。** Enterprise Agent Network——定义协作协议、身份认证、权限体系、上下文共享和流程编排——这将是下一代企业互联网的核心基础设施。

8.2 从"更聪明的 AI"到"更可信的 AI"

未来几年最有商业价值的 AI 产品，不会是"最聪明的 AI"，而是：

- **在企业内部，最能让 Agent 安全、合规地融入工作流的治理平台**
- **在企业之间，最能让不同企业的 Agent 安全协作的连接基础设施**
- **最能定义 Agent 身份、权限、审计标准的协议层**
- **最懂某个行业流程、最能融入企业工作流的应用**

大模型负责提供智能，而**可控可信的 Agent 治理架构——无论是在企业内部还是跨企业——负责把智能转化为稳定、可衡量、可持续、可追责的商业价值。**

8.3 最后的比喻

如果 AI 是电，Agent 是电器，那么：

- **企业内部 Agent** 就像一栋楼里的电器——同一个电表、同一个配电箱、同一个物业管着。可控可信在这里是"楼宇电路规范"——确保每个电器安全、不短路、不超载。
- **跨企业 B2B Agent** 就像不同城市、不同电网之间的电力交易——需要变电站、需要输电协议、需要计量和结算标准、需要互信机制。可控可信在这里升级为"跨电网调度协议"——没有这套协议，电根本送不过去。

今天，所有人都在造电器（开发 Agent）。但真正的大机会，是建电网（Agent 互联基础设施）——而一套统一的、从内部延伸到跨企业的可控可信治理体系，就是这张电网最核心的"调度协议"。

8.4 范式转换：从“AI 怎么用到企业”到“AI 怎么成为企业软件的一部分”

回顾本白皮书的整个论述，可以清晰地看到一次**范式转换（Paradigm Shift）**：

过去一个月甚至过去两年，AI 行业一直在问同一个问题：

> “怎么把 AI 用到企业里？”

这个问题天然把 AI 放在中心位置，企业是被改造的对象。于是大家的思路是：找一个场景 → 套一个模型 → 写 Prompt → 调优。模型一升级，全部推倒重来。

但 TEA 从另一个方向出发：

> **“怎么把 AI 变成企业软件的一部分？”**

这个问题的前提是：**企业软件本身才是中心，AI 只是其中一种执行资源。** 就像数据库、消息队列、云服务器一样——它们曾经也是“新技术”，但最终都退居幕后，成为企业软件的底层基础设施。模型也会走同样的路。

企业软件过去几十年的每一次重大抽象——数据库把存储变成数据服务、操作系统把硬件变成计算资源、Kubernetes 把服务器变成计算集群——本质上都在做同一件事：

> **把快速变化的底层技术，抽象成稳定的上层能力。**

AI 时代最有价值的平台，很可能也不是拥有最强模型的平台，而是**把不断变化的模型、Agent、工具和算力，抽象成企业可以长期依赖的业务能力的平台。**

这正是 TEA 的最终定位：**不是“又一个 Agent 框架”，而是在企业软件与 AI 技术之间建立一层稳定的抽象，让企业可以面向业务能力开发，而 AI 技术可以持续演进而不影响上层业务。**

Enterprise AI is not about building on models. It is about building on business capabilities.

附录：关键术语表

术语	英文	定义
可控智能体	Controlled Agent	在明确边界内完成任务，且可审计、可追责、可预测的智能体。在跨企业场景中，边界由双方协作协议共同定义。
可信企业智能体	Trusted Enterprise Agent	企业敢把关键业务（包括跨企业业务）交给它完成的智能体，满足身份、能力、信息、决策、执行、责任六项可信要求。对外代表企业的数字身份。
跨企业 Agent / B2B Agent	Cross-Enterprise Agent / B2B Agent	代表不同企业进行协作的智能体。与企业内部 Agent 的根本区别在于运行在跨信任域环境中。
Enterprise Agent Network	企业 Agent 网络	由多个企业的 Agent 互相通信、协作构成的网络。类比于互联网，但通信主体是 Agent 而不是浏览器或 API。
受控智能体执行模型	CAEM (Controlled Agent Execution Model)	以 Default Deny、Explicit Capability、Decision Trace、Human Accountability 为核心的四原则治理模型。在跨企业场景中每条原则都比内部场景更严格。
边界智能	Bounded Intelligence	CAEM 的哲学基础。每个 Agent 都在明确规定的能力边界、权限边界和决策边界内运行。
决策链追踪	Decision Trace	不仅记录执行了什么，更记录为什么这样理解、如何规划、依据哪些事实、如何验证的完整推理链。跨企业场景中需双方可验证。
企业 AI 行动网关	Enterprise AI Action Gateway	企业所有 AI 行动的最后一道关口。在跨企业场景中，是双方 Agent 执行操作之前的必经审批和校验层。
临时授权令牌	Delegation Token	有时效、有范围、不可转授权的临时权限凭证。在跨企业任务完成后自动回收。
会话密级	Conversation Clearance	Agent 间每次跨企业通信的密级标签（L0-L4），决定接收方可以返回什么信息以及多少信息。
智能体 IAM	Agent IAM	将企业身份与访问管理（IAM）体系移植到 Agent 领域，在跨企业场景中扩展为跨域身份与访问管理。
跨企业协作协议	Cross-Enterprise Collaboration Protocol	两个企业之间约定的 Agent 协作规则，包括通信权限、信息密级、审批要求、责任归属等。
推理规划器	Inference Planner	TEA Runtime 的核心组件，不是简单的模型路由，而是规划整个推理过程——意图理解、任务拆分、模型分配、执行编排和结果校验。
	Business	

业务能力接口	Capability Interface	TEA 面向开发者暴露的接口抽象。企业调用的是 `contract.review()` 而非 `gpt.chat()`，底层模型可随时替换。
四级输出分级	Output Classification	Agent 输出按风险等级分为 🟢信息、🔵建议、🟡决策建议、🔴执行四级，不同级别对应不同的自动化和审批策略。

版本：v2.0
日期：2026 年 6 月

出品方: Chaos Lab

联系邮箱: 2819699195@qq.com