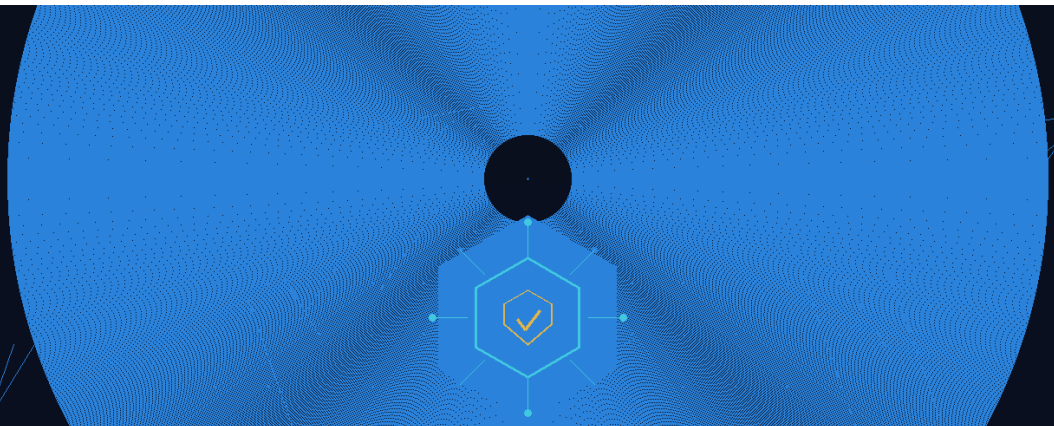


Trusted Enterprise Agent Whitepaper

From "Smarter AI" to "More Trustworthy AI"

Governance Paradigms, Technical Architecture & Business Pathways for Enterprise-Grade Agents

Agent = Intelligence + Control + Responsibility

Controlled

Default Deny
Explicit Authorization
Bounded Execution

Trusted

Verifiable Identity
Traceable Decisions
Attributable Responsibility

Auditable

Reasoning Chain Logs
Full Process Tracking
Continuous Improvement

Version: v2.0
Date: June 2026
Classification: Industry Whitepaper

Trusted Enterprise Agent Whitepaper

> **Governance Paradigms, Technical Architecture, and Business Pathways for Enterprise-Grade Agents — From Internal Collaboration to Cross-Enterprise Orchestration**

Abstract

The rapid evolution of Large Language Models (LLMs) has endowed Agents with unprecedented capabilities in understanding, reasoning, and planning. Yet the industry's attention has long been focused on a single direction — how to use Agents **within the enterprise** to replace employees, improve efficiency, and optimize processes. This is important, but it reveals only one dimension of the problem.

This whitepaper advances a proposition that has been widely overlooked but is critically important:

> **The primary battleground for "Controlled and Trusted" is not only internal enterprise Agents — cross-enterprise B2B Agents are equally vital.**

Internal enterprise Agents address **organizational efficiency** — how a single company can operate faster. Here, Controlled-and-Trusted represents a "best practice": within a unified trust domain, mechanisms such as approvals, permissions, and audits ensure Agents operate safely.

Cross-enterprise Agents (B2B Agents) address **collaboration efficiency** — how different companies can complete procurement, supply chain coordination, logistics, and financial transactions through Agents. Here, Controlled-and-Trusted escalates to a **hard requirement**: How do Agents from different enterprises mutually authenticate identity? How is information visibility controlled? How do we ensure every cross-enterprise transaction has an auditable chain of responsibility? How can Agents safely make commitments on behalf of their enterprises when no single party holds absolute authority?

Whether internal or cross-enterprise, the core issue is not "how smart is the Agent" — it is whether the Agent can be trusted. The difference is that in cross-enterprise scenarios, the complexity and risk level are orders of magnitude higher.

This whitepaper proposes an **enterprise-grade Agent governance architecture centered on "Controlled and Trusted"** — applicable to both internal and cross-enterprise scenarios, but facing higher requirements and richer implications in cross-enterprise collaboration. Its core proposition:

> **Agent = Intelligence + Control + Responsibility**

It is organized across the following dimensions:

1. **The Dual Battlefields of "Controlled and Trusted"** — Internal enterprise Agent governance as the foundation; cross-enterprise B2B Agent governance as the extension and upgrade.
2. **The Agent Governance Model (CAEM)** — Default Deny, Explicit Capability, Decision Trace, and Human Accountability, progressively strengthened from internal to cross-enterprise.
3. **Six-Layer Runtime Architecture** — Identity, Capability, Governance, Context, Execution, and Audit, extended with dual-mode (intra-domain/cross-domain) execution in cross-enterprise scenarios.

4. **Seven-Layer Agent Communication Governance Model** — From identity authentication to information classification to delegation tokens; organizational governance internally, cross-organizational governance externally.
5. **Business Pathway: Enterprise Agent Network** — Why the greatest opportunity is not building more Agents, but becoming the connective layer between them.

This whitepaper provides enterprise decision-makers, technical architects, and industry practitioners with a new perspective: **the greatest future business opportunity may not lie in developing more standalone Agents, but in becoming the infrastructure for cross-enterprise Agent collaboration — defining collaboration protocols, identity authentication, permission systems, context sharing, and process orchestration. Whoever enables Agents from different enterprises to complete business collaboration securely, at low cost, and with high trustworthiness, will have the opportunity to become the core infrastructure of the next-generation enterprise internet.**

Table of Contents

Prologue: TEA Core Design Principles

Before delving into market analysis and technical architecture, this chapter establishes the most fundamental design principles of TEA. These are not implementation details — they define the **essential difference** between TEA and most Agent frameworks today.

Principle 1: Business First, Models Second

> **Enterprise AI applications should be developed against business capabilities rather than foundation models.**

Why This Principle Matters

Looking back at decades of enterprise software evolution, one constant pattern emerges: **enterprises never build around technology itself — they build around business.**

- ERP was not built around Oracle, but around financial management and supply chain management.
- CRM was not built around MySQL, but around customer management and sales processes.
- OA was not built around Redis, but around approval workflows and collaborative办公.

Technology constantly changes — databases go from Oracle to MySQL to TiDB — but business logic remains relatively stable. **Technology is a replaceable execution resource; business capabilities are long-term, stable software assets.**

AI Is Making the Same Mistake Today

Many enterprises today write AI application code like this:

```
# Model-oriented development — switching models nearly requires a rewrite
deepseek.chat(prompt)
gpt.chat(prompt)
claude.messages(prompt)
```

The entire business logic is tightly coupled to specific models. Model upgrade → Prompt rewrite → Workflow modification → Application follows suit. This is the biggest architectural problem in AI applications today.

The TEA Approach

Enterprises should develop against **Business Capabilities**, with models fading into the background as replaceable execution resources:

```
# Business-capability-oriented development — zero changes when switching models
contract.review(document)       # Contract review
customer.reply(ticket)          # Customer reply
finance.analyze(report)         # Financial analysis
purchase.approve(order)         # Purchase approval
```

Business code never knows which model sits behind it. GPT today, DeepSeek tomorrow, Huawei Ascend Cloud the day after — **zero changes to business code**.

What This Means

TEA's abstraction level is **higher than both models and Agents**. Models, Agents, GPUs, and tools are merely resources scheduled by the TEA Runtime. The interfaces exposed to developers and enterprises are not model APIs, but **Business Capability Interfaces**.

```
Business Layer (Business Capabilities)
    Contract Review / Customer Reply / Sales Analysis
    Purchase Approval / Knowledge Query
                        │
                        ▼
        Business Capability API
                        │
                        ▼
             TEA Runtime
     Intent / Planning / Policy / Routing / Memory / Audit
                        │
                        ▼
        Foundation Model Layer
     GPT / DeepSeek / Qwen / GLM / Claude / Llama
```

Models have already been reduced to infrastructure. Applications developed against models will be endlessly dragged along by technological iteration; applications developed against business capabilities can isolate technological changes beneath the Runtime layer.

Principle 2: Bounded Intelligence

> **Every Enterprise Agent must operate within an explicitly defined capability boundary, authorization boundary, and decision boundary.**

Why Boundaries Are Necessary

Most Agent frameworks today focus on a single question:

> *How can the Agent do more?*

TEA focuses on a fundamentally different question:

> *How do we ensure the Agent only does what it is supposed to do?*

What enterprises truly fear is not insufficient Agent capability — it is the **Agent making decisions on behalf of the enterprise without proper authorization, sufficient evidence, or within its designated scope of responsibility.**

The Concrete Meaning of Capability Boundaries

Just as human employees in an enterprise each have their scope of responsibility — Finance doesn't approve legal matters, HR doesn't design chips, the CEO doesn't write device drivers — Agents should have clearly defined boundaries:

Finance Agent:

- ✅ Invoice audit, expense approval, financial analysis, cost forecasting
- ❌ Modify corporate strategy, delete databases, restructure人事

Legal Agent:

- ✅ Contract review, risk alerts, regulatory lookup
- ❌ Auto-sign contracts, auto-execute payments

R&D Agent:

- ✅ Write code, fix bugs, submit PRs
- ❌ Auto-deploy to production

Bounded Intelligence and CAEM

Bounded Intelligence is the philosophical foundation of CAEM (Controlled Agent Execution Model). The four CAEM principles — Default Deny, Explicit Capability, Decision Trace, and Human Accountability — are essentially **Bounded Intelligence** implemented across different technical dimensions.

Bounded Intelligence does not limit an Agent's capability — it **makes that capability trustworthy to the enterprise**. An Agent without boundaries, no matter how capable, cannot enter core business operations; a boundary-defined Agent can grow from "intelligent assistant" to "trusted digital employee."

Principle 3: Four-Tier Output Classification

TEA classifies Agent outputs into four tiers, each corresponding to different execution strategies and governance requirements:

Output Tier	Meaning	Example	Auto-Execute?
🟢 **Information**	Query, summarize, explain	"Inventory is 350 units"	✅ Auto
🔵 **Recommendation**	Provide options and suggestions	"Recommend Supplier B, lowest quote"	✅ Auto
🟡 **Decision Proposal**	Involves business decisions	"Recommend approving the ¥2.4M purchase order"	⚠️ Human confirmation
🔴 **Action Execution**	Involves real business actions	Payment, contract signing, production deployment	⚠️ Authorization + Approval required

Core insight: Human confirmation at 🟡 and 🔴 tiers is not because the Agent "can't" or "isn't smart enough" — it is because **the action exceeds the Agent's decision boundary**. This is a hard requirement of enterprise governance, not a technical limitation.

The TEA Runtime automatically determines which tier the current output belongs to based on task type and risk level. If it exceeds the Agent's decision boundary, it automatically escalates to "requires human verification" — with complete decision rationale and audit records throughout the process.

---

1. Introduction: From "Big Models" to "Small Applications" — The Return of AI Business Value

1.1 The Two Phases of Market Evolution

The AI industry has traversed two clear phases over the past two years:

- **2023–2024: Capability Building Phase** — The market focused on "building a bigger model," chasing parameter scale and benchmark score improvements.
- **2025–2026: Application Explosion Phase** — The market began shifting toward "building countless smaller, more vertical, more deeply embedded applications," integrating AI capabilities into specific business scenarios.

The real money-making opportunity has shifted from "building a bigger model" to "building countless smaller, more vertical, more deeply embedded applications."

1.2 Big Models Are Becoming Infrastructure

Large language models increasingly resemble electricity: no one pays for electricity itself — people pay for refrigerators, air conditioners, washing machines, and phones. Similarly, no one pays for GPT, Claude, or Gemini itself — people pay for **AI + a specific scenario**: AI contract writing, AI interview assistant, AI sales assistant, AI procurement assistant, AI financial assistant, AI medical assistant... Each represents a market worth billions or even tens of billions.

1.3 What Do Enterprises Buy?

Enterprises don't buy AI. Enterprises buy **cost reduction** or **revenue growth**.

- **Sales Follow-up AI**: Automatically analyze customer conversations, remind sales staff, draft quotes, predict close rates — enterprises willingly pay hundreds of thousands annually because it directly increases sales.
- **Customer Service Bot**: 100-person team → 30-person team, 70% cost reduction, ROI immediately visible.
- **Contract Review System**: AI checks risks, highlights clauses, provides revision suggestions — saves 30 minutes per contract, tens of thousands of contracts per year, enormous value.

1.4 What Truly Creates Value Is the Workflow

AI products are not equal to chat boxes. What truly generates value is the complete workflow:

```
Input → AI → Tools → Database → Approval → Notification → Execution → Feedback → Continue AI
```

AI is just one step; the entire workflow is where the money is. Take the recruitment process:

```
Receive Résumé → AI Scoring → Schedule Interview → Send Email → Generate Offer → Onboarding Process
```

1.5 Two Dimensions of Controlled and Trusted: Internal and Cross-Enterprise

Most AI applications and Agent products today are designed **around a single enterprise**: helping internal sales, customer service, legal, and HR improve efficiency. Within the enterprise, Controlled-and-Trusted manifests as approval processes, permission management, and operational audits — these are baseline governance capabilities every enterprise-grade Agent needs.

But the bigger imagination space lies between enterprises.

When Agents need to represent different enterprises in collaboration — a procurement Agent sending RFQs to a supplier Agent, a logistics Agent interfacing with a customs Agent, a bank Agent reviewing loan materials submitted by an enterprise Agent — the requirements for Controlled-and-Trusted escalate from "best practice" to "hard requirement." There is no unified permission system here, no common boss, no natural information symmetry.

Controlled-and-Trusted is a baseline governance capability within the enterprise; in cross-enterprise collaboration, it becomes the very prerequisite of the business model. This whitepaper covers both dimensions, with particular emphasis on the higher requirements that cross-enterprise scenarios impose on governance systems.

1.6 The Evolution of Value Measurement: From Token to Business Outcome

The pricing model for large models is undergoing a profound transformation, and this directly impacts the architectural design of enterprise-grade Agents.

From "Selling Models" to "Selling Compute": DeepSeek V4's introduction of peak/off-peak pricing is a landmark signal — GPU compute is becoming a demand-priced resource, just like electricity and cloud computing. As model capabilities converge, what is truly scarce is GPU compute, inference throughput, and response latency.

Token is not the destination — it is a transition: A thousand tokens of plain text summarization versus a thousand tokens involving deep reasoning with 20 tool calls — the GPU consumption can differ by orders of magnitude. An Agent executing a task may output only 500 tokens, yet think through 100k tokens, call 30 tools, and run OCR, Vision, and Embedding behind the scenes. **1000 tokens ≠ equivalent compute cost.**

The evolution path of value measurement:

```
Token → Compute → Task → Business Outcome
```

Stage	Measurement Unit	Stakeholder Concern	Typical Question
Token Era	Price per million tokens	Model vendors, developers	"Which model is cheapest?"
Compute Era	GPU time, FLOPs	Cloud providers, infra	"How to schedule peak/off-peak?"
Task Era	Cost per Completed Task	Enterprise IT	"How much to review one contract?"
Outcome Era	Cost per Business Outcome	Enterprise leadership	"How much to acquire one customer?"

What this means for TEA: The optimization objective of TEA Runtime cannot remain at "pick the cheapest model" or "use the fewest tokens." The true optimization target of TEA Runtime should be:

> **Complete a business objective at the lowest cost, highest quality, and with full governability.**

From "Cost per Token" to "Cost per Business Outcome" — this is not just a billing change, but an **elevation in abstraction level**. Tokens will increasingly resemble CPU clock cycles today — still present, but primarily an internal scheduling metric, not the core metric enterprises care about.

In the future, enterprises won't purchase "API calls"; they will purchase "100,000 customer service conversations completed per day" or "1 million contracts auto-reviewed per month." The platform internally decides which model to use, how much GPU to allocate, whether to enable deep reasoning, and whether to schedule during off-peak hours. TEA Runtime is the execution layer designed for this era.

---

2. Core Proposition: Controlled and Trusted — From Internal Enterprise Agents to Cross-Enterprise B2B Agents

2.1 Essential Differences and Progressive Relationship Between the Two Types of Agents

Internal enterprise Agents and cross-enterprise Agents (B2B Agents) solve problems in two different dimensions, but they are not opposed — B2B Agent Controlled-and-Trusted requirements are extensions and upgrades built upon the foundation of internal Agent governance:

	Internal Enterprise Agent	**Cross-Enterprise B2B Agent**
Problem Solved	Organizational efficiency (one company running faster)	Collaboration efficiency (multiple companies collaborating more smoothly)
Trust Domain	Single trust domain (one boss, one set of rules)	Cross-trust-domain (independent, no hierarchical relationship)
Permission System	Unified management (corporate IT decides)	Independent + peer-to-peer negotiation protocols
Information Visibility	Internally controllable (boss decides who sees what)	Highly sensitive (core data must not leak to counterparties)
Responsibility Attribution	Internal accountability	Cross-enterprise legal/commercial liability
Controlled-and-Trusted Requirement	**Best Practice** (approvals, permissions, audits are baseline)	**Hard Requirement** (identity auth, info classification, dual audit are prerequisites)

Internal enterprise Agent Controlled-and-Trusted determines "how well it can be used"; cross-enterprise Agent Controlled-and-Trusted determines "whether it can be used at all."

2.2 The Necessity of B2B Agents: The Current State of Enterprise Collaboration

Today, when two enterprises collaborate, most processes look like:

```
Company A → Send Email → Send WeChat → Phone Call → Excel → PDF → Manual Confirmation → Company B
```

Most of the work is merely **information synchronization + confirmation + negotiation**. Agents are perfectly suited for this.

The future is more likely to be:

```
Enterprise A Agent → Natural Language Negotiation → Invoke Systems → Confirm Rules →
Enterprise B Agent
```

Humans only intervene at key approval nodes.

2.3 Which Cross-Enterprise Businesses Are Most Suitable for Agent-ification?

Category 1: Procurement (The Largest B2B Agent Opportunity)

Today's procurement process: Buyer → Find suppliers → RFQ → Compare prices → Confirm delivery → Contract → Place order.

The future:

```
Procurement Agent → Send RFQ to 20 Supplier Agents → Auto-analyze →
Compare inventory → Compare delivery → Recommend optimal solution → Submit for boss
approval
```

The Supplier Agent responds: Is inventory available? Can production be arranged? Is the price acceptable? Is partial delivery supported? The entire process could be completed in ten minutes.

Category 2: Supply Chain Coordination

An automotive part involves: OEM → Tier 1 Supplier → Tier 2 Supplier → Tier 3 Supplier. Today, it all runs on email, Excel, SAP, and phone calls.

The future:

```
OEM Agent → Notify demand change → Supplier Agent re-plan production →
Logistics Agent re-arrange → Auto-reply with risk assessment
```

The entire supply chain synchronizes in real time.

Category 3: Cross-Enterprise Sales

A customer says "I need 500 units." The Sales Agent auto-coordinates:

```
Requirement Analysis → Inventory Agent → Procurement Agent → Logistics Agent → Quoting
Agent → Generate Proposal
```

The Customer Agent continues negotiating price, payment terms, delivery, and after-sales — many negotiation rounds can be automated.

Category 4: Logistics

Logistics involves freight forwarders, shipping companies, insurance, warehousing, customs declaration, customs clearance, and transportation — all different enterprises. The future:

```
Logistics Agent → Notify port → Notify customs → Notify warehouse → Notify truck → Sync status
```

All enterprises share status — but with **strict data permission isolation for each party**.

Category 5: Finance (Corporate Lending)

The Enterprise Agent auto-prepares financial data, ERP data, tax records, contracts, and invoices; the Bank Agent auto-reviews. Final human signature required.

2.4 What Higher Requirements Do Cross-Enterprise Agents Impose on Governance Systems?

Within a single enterprise, Agent governance is a natural extension of IT management — the CTO can set unified rules, and all Agents operate within the same organizational structure. Internal Agents equally require governance capabilities like approval, permissions, and audit, but all of this can be achieved relatively efficiently within a single trust domain.

Cross-enterprise Agent collaboration adds entirely new challenges on top of internal governance:

- **No central authority**: Company A's Agent and Company B's Agent are not subordinate to each other. Who decides "who can do what"?
- **Information asymmetry is a hard requirement**: The supplier's inventory cost, the customer's budget cap, the logistics provider's internal scheduling — these are all core commercial secrets that absolutely cannot be fully exposed in peer-to-peer collaboration.
- **Complex responsibility attribution**: If Company A's procurement Agent places an order based on incorrect information from Company B's supplier Agent, who is responsible?
- **High difficulty of identity authentication**: How do you know the Agent you're communicating with genuinely represents the enterprise it claims? Has it been tampered with?
- **Irreversible execution consequences**: An internal Agent making a mistake affects its own company; a cross-enterprise Agent making a mistake could trigger cascading cross-enterprise commercial disputes.

Therefore, cross-enterprise Agents require not "smarter LLMs," but a governance infrastructure that is independent of any single party and can be trusted by all participants.

2.5 From "Free Intelligence" to "Authorized Intelligence": Core Principles Spanning Internal and Cross-Enterprise

The industry's current technical definition of an Agent is:

> Agent = LLM + Memory + Tools + Planning

But from an enterprise perspective — whether internal or cross-enterprise — it must be redefined as:

> **Agent = Intelligence + Control + Responsibility**

This means:

- **LLM provides power, not governance** — The Agent is not a free-acting "digital employee" but a strictly constrained "enterprise digital representative." This is best practice internally; it is a **hard requirement** in cross-enterprise scenarios.
- **Ability and Authority must be thoroughly separated** — The Agent does not act because it "can," but because it is "authorized to."
- **The smarter the LLM, the more critical the governance layer** — An Agent capable of autonomously negotiating with dozens of supplier Agents, adjusting prices, and signing contracts, without governance constraints, would face exponentially escalating commercial risk. This holds equally true for both internal and cross-enterprise scenarios.

2.6 Zero Trust Principle: The Security Foundation of Enterprise-Grade Agents

The foundational security philosophy of enterprise-grade Agents (whether internal or cross-enterprise) must be **Zero Trust**:

> **An Agent has zero capabilities by default; all capabilities are granted through authorization. In cross-enterprise scenarios, add: information from other enterprises' Agents is distrusted by default and must be verified.**

Corresponding to four fundamental questions:

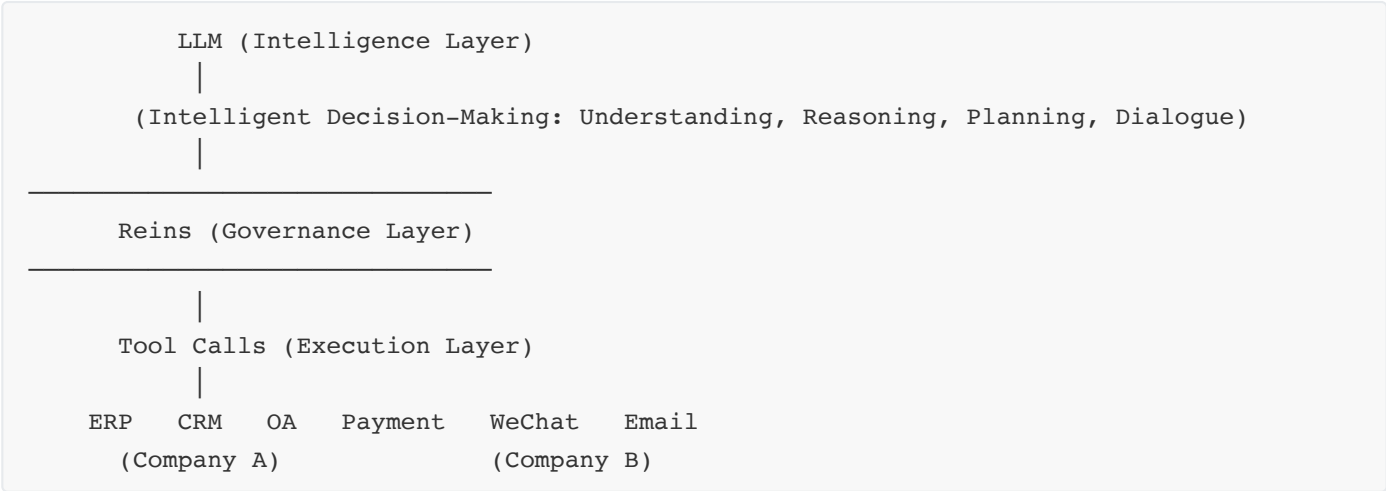
1. **What can it do?** — Allow List (capability whitelist), must be explicitly declared by the owning enterprise
2. **What must it absolutely never do?** — Deny List (prohibition list), e.g., read-only, RFQ-only-no-contract
3. **What if there is no explicit authorization?** — Default Deny, never permitted because "AI thought it would be more convenient"
4. **What has it done?** — Complete cross-enterprise Decision Trace, auditable by both parties

---

3. Theoretical Foundation: The Agent Governance Model (CAEM) — From Internal Governance to Cross-Enterprise Extension

3.1 Three-Layer Architecture

Decomposing an enterprise-grade Agent into three distinct layers is the starting point for understanding enterprise Agent governance. This architecture applies equally to both internal and cross-enterprise scenarios:



- **Intelligence Layer (Brain)**: Provided by model vendors (OpenAI, Anthropic, Google, Alibaba, etc.), responsible for understanding, reasoning, planning, and dialogue.
- **Governance Layer (Reins)**: Determines what an Agent can do, when it can do it, and what it cannot do. Within the enterprise, the governance layer is a rules engine; in cross-enterprise scenarios, it additionally supports peer-to-peer governance protocols — it is not a simple extension of internal rules, but an upgrade to a collaboration control plane.
- **Execution Layer**: Actually invokes enterprise systems. In cross-enterprise scenarios, Company A's Agent calls Company A's ERP; Company B's Agent calls Company B's ERP — **they cannot directly access each other's systems across domains**.

What truly controls collaboration risk is the middle layer — the Governance Layer. Within the enterprise, it is the execution engine for management rules; in cross-enterprise collaboration, it upgrades to a "collaboration control plane" independent of any single enterprise.

3.2 Controlled Agent Execution Model (CAEM)

CAEM comprises four non-negotiable principles, **serving as the governance baseline internally and further escalating in cross-enterprise scenarios**:

Principle 1: Default Deny

Any capability not explicitly authorized is categorically prohibited from execution. In cross-enterprise scenarios, this means: Company A's Agent cannot access any system of Company B unless Company B explicitly authorizes it. An Agent cannot spontaneously request the counterparty to disclose information just because "negotiation needs more data."

Principle 2: Explicit Capability

An Agent can only execute actions within its capability manifest, and these capabilities must be **visible to collaborating parties**. Each capability is configured with: permission requirements, applicable conditions, approval requirements, callable systems, and **which external Agents it is exposed to**.

Principle 3: Decision Trace

Records not only what was executed, but also: Why was it understood this way? How was it planned? What facts was it based on? How was risk assessed? How was the result ultimately verified? In cross-enterprise scenarios, **the decision chain must be verifiable by both (or all) parties** — this goes deeper than traditional API logs; it records the Cross-Enterprise Reasoning Log.

Principle 4: Human Accountability

For high-risk or out-of-scope cross-enterprise actions, **both parties'** human approval must be obtained. Every cross-enterprise transaction has clearly identified responsible persons and approval records. The Agent can never bypass this step. The ultimate responsible party is always human.

3.3 Six Layers of Governance

Governance is not a single "rein" but a complete control system:

Layer	Name	Core Question	Cross-Enterprise Example
1	Rule	What are the fixed boundaries?	Payments cannot exceed ¥500K; can only RFQ certified suppliers
2	Policy	What are the business conditions?	VIP customers get 10% discount; new suppliers require additional approval
3	Permission	What are the data access boundaries?	Sales Agent cannot disclose cost data to Customer Agent
4	Verification	Is the information trustworthy?	Counterparty Agent claims 100 units in stock → request verifiable proof
5	Approval	Is human confirmation needed?	Cross-enterprise contract signing → legal approval from both parties
6	Rollback	What if execution goes wrong?	Cross-enterprise order error → both parties confirm rollback protocol

3.4 Four-Layer Model for Decision Chain Auditing

Auditing does not mean recording API calls — it means recording the complete cross-enterprise reasoning chain:

Layer 1: Think — Why did the Agent understand it this way? E.g., counterparty Agent said "ship ASAP" → our Agent understood "Priority = P1" — record the understanding basis and source.

Layer 2: Plan — What execution plan did the Agent formulate? Which external systems are involved?

Layer 3: Action — Which systems were actually invoked? Which are our own systems, and which are interactions with external Agents?

Layer 4: Evaluate — Was execution successful? Did the counterparty confirm? If it failed, was it our fault or theirs? Is re-negotiation needed?

3.5 Seven Unavoidable Questions

For every Action by a cross-enterprise Agent, the following must be answerable:

1. **Who** — Who is it? Which enterprise does it represent? (Identity)
2. **Why** — Why is it doing this? Based on which cross-enterprise task? (Task / Intent)
3. **What** — What is it preparing to do? Is it within the authorized scope of both parties? (Capability)
4. **Based on What** — What information is it based on? Is the source our enterprise or the counterparty? (Context)
5. **Who Approved** — Who authorized it? Who approved it? Single-party or dual-party approval? (Governance)
6. **What Happened** — What was ultimately executed? How did the counterparty respond? (Execution)
7. **How Do We Prove It** — How can the entire process be proven afterwards, to both our side and the counterparty? (Audit)

If a system can completely answer these seven questions — and the answers are verifiable by both parties — then that Agent possesses the fundamental characteristics of a cross-enterprise "Controlled and Trusted Agent."

3.6 Bounded Intelligence: The Philosophical Foundation of CAEM

The four CAEM principles — Default Deny, Explicit Capability, Decision Trace, and Human Accountability — share a common philosophical foundation: **Bounded Intelligence**.

> **Every Enterprise Agent must operate within an explicitly defined capability boundary, authorization boundary, and decision boundary.**

Why Boundaries Matter More Than Capability

Most Agent frameworks today ask:

> *How can the Agent do more?*

TEA asks:

> *How do we ensure the Agent only does what it is supposed to do?*

This is not a conservative technical choice — it is a hard requirement of enterprise-grade software. What enterprises truly fear is not insufficient Agent capability, but the **Agent making decisions on behalf of the enterprise without proper authorization, sufficient evidence, or within its designated scope.** An Agent without boundaries, no matter how capable, cannot enter core business operations; a boundary-defined Agent can grow from "intelligent assistant" to "trusted digital employee."

Three Dimensions of Bounded Intelligence:

Dimension	Meaning	CAEM Correspondence
Capability Boundary	What the Agent can and cannot do	Explicit Capability (Allow List + Deny List)
Authorization Boundary	Under what conditions the Agent may execute	Default Deny + Governance Rules + Approval
Decision Boundary	What the Agent can decide autonomously vs. what requires human confirmation	Human Accountability + Output Classification

The Core Corollary of Bounded Intelligence: Human verification is not about insufficient model capability — it is about enterprise governance requirements.

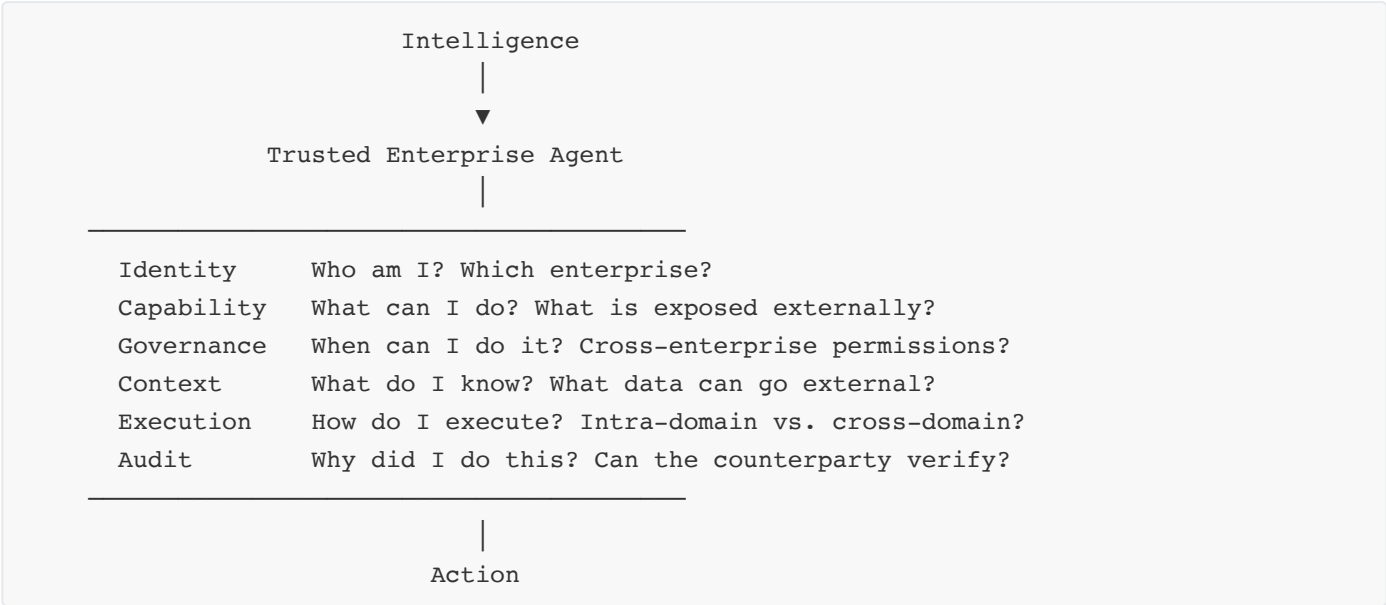
When a Finance Agent analyzes an invoice (✅ within capability boundary) but prepares to make a payment (❌ beyond decision boundary), the system automatically escalates to "🟡 Requires Human Verification." This is not because the Agent "isn't smart enough" or "can't do it" — it can, but it is **not allowed to**. This is the direct manifestation of "Ability and Authority must be thoroughly separated."

---

4. Technical Architecture: Six-Layer Runtime Model — From Intra-Domain Execution to Cross-Domain Collaboration

4.1 Complete Model

The complete lifecycle of an enterprise-grade Trusted Agent comprises six layers, sharing the same architectural framework across both internal and cross-enterprise scenarios, with each layer having additional extension requirements in cross-enterprise contexts:



4.2 Detailed Layer Definitions (Internal Baseline + Cross-Enterprise Extensions)

Layer 1: Identity

Core Question: Who am I? Which enterprise do I represent? How can the counterparty Agent verify my identity?

```
Procurement Agent
  Enterprise: A Manufacturing Co., Ltd. (Unified Social Credit Code: XXX)
  Department: Procurement
  Supervisor: Procurement Manager Zhang San
  Represents: Company A Procurement Department
  Employee ID: AGT-PUR-A001
  Trust Level: L2
  External Identity Credential: Issued by XX CA
```

Every cross-enterprise Trusted Agent must possess a digital badge containing not only internal identity information but, more importantly, **externally verifiable credentials** — like HTTPS certificates, allowing counterparties to verify via a CA that this Agent genuinely represents the enterprise it claims.

Layer 2: Capability

Core Question: What can I do? Which capabilities are exposed to collaborators?

```
√ Query own enterprise inventory
√ Create own enterprise purchase orders
√ Send RFQ to certified supplier Agents
√ Receive and compare quotations
× Access supplier internal systems
× Modify supplier quotations
× Direct payment
× View supplier cost data
```

Key distinction: In cross-enterprise scenarios, the capability whitelist must differentiate "intra-domain capabilities" from "external-facing capabilities" — some capabilities are internal-only; some can be exposed to external Agents.

Layer 3: Governance — The Control Plane

Core Question: In cross-enterprise collaboration, when can I act?

Governance is the **control plane that runs through the entire cross-enterprise execution process**:

```
Governance (Cross-Enterprise)
├── Policy: Cross-enterprise orders > ¥1M require dual-party approval
├── Permission: External Agents can only read fields I authorize
├── Approval: Cross-enterprise contract signing → our legal + counterparty legal
├── Communication: Only communicate with certified enterprise Agents
├── Information Classification: L0-L4, different levels for different external
enterprises
├── Risk Control: Anomalous RFQ pattern detection, anti-fraud
└── Delegation: Temporary authorization tokens, auto-revoked after task completion
```

Layer 4: Context

Core Question: What do I know? What information can be disclosed to external Agents?

This is where cross-enterprise Agents differ most from internal Agents. The Agent does not know everything; it obtains bounded context based on counterparty enterprise identity, bilateral agreements, and clearance levels:

- Which information is internal-only? (Complete internally)
- Which is visible to collaborators? (Aggregated info after authorized desensitization)
- Which is invisible to collaborators? (Cost, profit, salaries — core secrets)
- Which comes from collaborators? (Counterparty-provided info, requires verification)

Example: Company A's Procurement Agent sends an RFQ to Company B's Supplier Agent. Company B's Agent may know the exact inventory is 350 units (Warehouse A: 100, B: 200, C: 50), but only replies "approximately 350 units available." Company A does not need to know Company B's warehouse distribution.

Layer 5: Execution

Core Question: How do I execute? Which are internal operations, and which are cross-enterprise interactions?

In cross-enterprise scenarios, execution is divided into two domains:

- **Intra-Domain Execution**: Invoke own enterprise ERP, CRM, OA — same as internal Agents
- **Cross-Domain Execution**: Send requests to external enterprise Agents, receive responses, exchange structured context — this is an entirely new capability

All cross-domain execution actions must pass through the Governance layer and **cannot directly invoke the counterparty's internal systems**. Company A's Agent calls Company B's Agent's **peer interface**, not Company B's ERP.

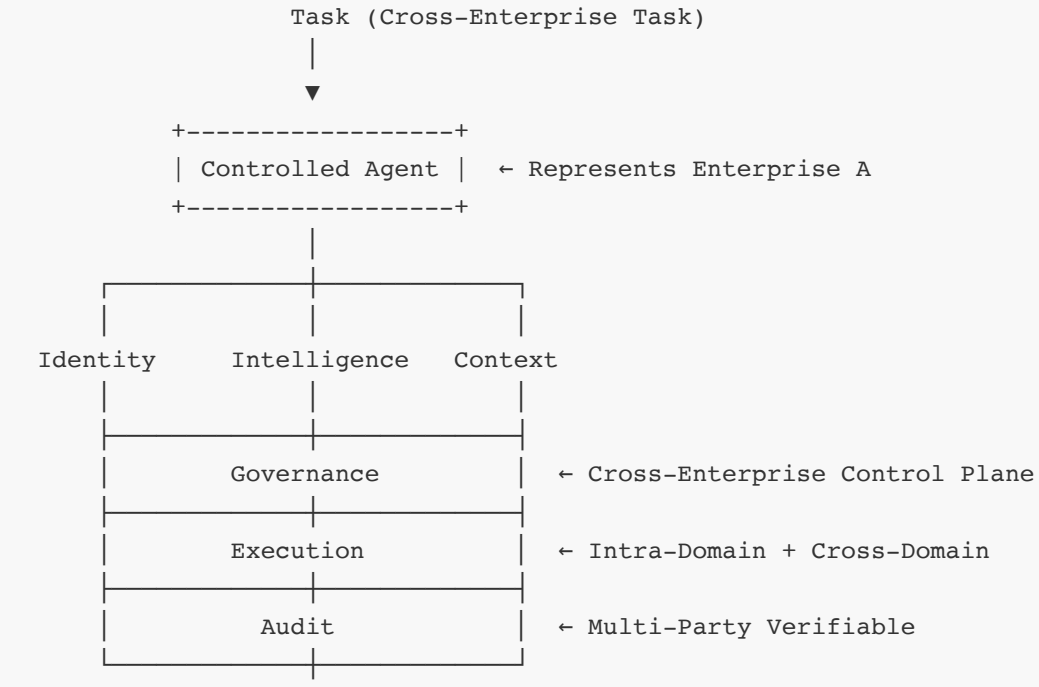
Layer 6: Audit

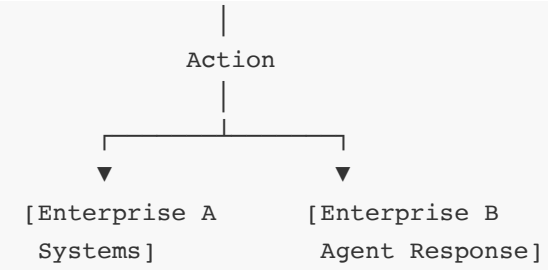
Core Question: Why did I do this? Can both parties independently verify?

The core requirement of cross-enterprise auditing is: **audit records must be verifiable by both (or all) parties**. This means:

```
Cross-Enterprise Decision Chain:
  [Company A] Basis: Purchase requisition PO20260012
  [Company A] Why RFQ Company B? → B is on qualified supplier list
  [Company A] Why accept this quote? → Lowest among 3 bids
  [Company B] Why this price? → Based on current inventory and cost, approved
  [Company B] Why confirm this delivery date? → Production schedule available
  Final Execution: [A] Create PO + [B] Confirm acceptance
  Dual Sign-off Verification: A Approver ✓ | B Approver ✓
```

4.3 Architecture Overview





Core Design Philosophy: Intelligence is not the center; Action is the center. Governance serves as the cross-enterprise control plane running through the entire lifecycle. Execution is explicitly divided into intra-domain and cross-domain, with isolation between the two spaces. All cross-domain interactions pass through peer governance protocols.

---

5. Agent Communication Governance: A Seven-Layer Model from Identity to Authorization

5.1 From "Network Governance" to "Organizational Governance" to "Cross-Organizational Governance"

Many people design Agent communication by default as:

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

Equal, peer-to-peer communication. But real-world scenarios involve progressively escalating communication governance complexity.

Within an enterprise, it is **organizational governance** — information asymmetry, clear hierarchies, different permissions for different roles. Agent communication must follow organizational hierarchy: a Procurement Agent cannot directly access the CEO Agent; the Finance Agent's data is invisible to the Sales Agent.

Cross-enterprise is **cross-organizational governance** — not only does the enterprise have internal hierarchy, but there are also asymmetric collaboration relationships between enterprises:

> **Between Company A's Agent and Company B's Agent, there is no hierarchical relationship, but there are clear collaboration boundaries.**

In cross-enterprise scenarios, Agents are subject to three additional layers of constraints beyond internal organizational governance:

1. What permissions does its own enterprise grant it?
2. What does the counterparty enterprise allow it to see?
3. What does the bilateral collaboration protocol allow it to do?

5.2 Three-Layer Permissions for Cross-Enterprise Agent Communication

Layer 1: Communication Permission

```
Company A Procurement Agent → × → Company B CEO Agent (no skip-level communication)
Company A Procurement Agent → √ → Company B Supplier Agent (via established bilateral
collaboration channel)
```

Communication permissions are not only determined by the Agent's own enterprise but also controlled by the counterparty enterprise. Company B can reject communication requests from certain types of Company A Agents.

Layer 2: Visibility

```
Company B Inventory Agent knows: Warehouse A: 100, Warehouse B: 200, Warehouse C: 50
Company A Procurement Agent can see: Available inventory: 350 (desensitized aggregate,
warehouse distribution hidden)
```

Information is filtered before transmission. What Company A's Agent receives is not raw data, but **authorized information after desensitization by the counterparty's governance layer**.

Layer 3: Depth

```
Company A CEO Agent asks: Why is this procurement cost 15% higher than last time?
Company B Supplier Agent answers: Raw material cost increase + tight delivery window (with
analysis basis)

Company A Regular Procurement Agent asks the same question:
Company B Supplier Agent answers: Current quote is ¥XX million (pricing logic not exposed)
```

Same question, same counterparty Agent, but different inquirers from different levels/enterprises — get answers of different depths.

5.3 Cross-Enterprise Information Classification System

Level	Definition	Visible To	Cross-Enterprise Example
L0	Public information	All enterprise Agents	Product catalog, public pricing
L1	Collaboration shared	Signed collaboration partners	Order status, delivery date, logistics status
L2	Collaboration restricted	Specific partners + specific levels	Supplier pricing (procurement only), quality data (QC only)
L3	Enterprise confidential	Own enterprise management only	Profit, budget, cost structure
L4	Core secret	Very few own enterprise authorized persons	Acquisition plans, financing, pricing strategy

For each cross-enterprise Agent communication, both parties' governance layers sequentially evaluate:

```
Requesting Agent Identity → Affiliated Enterprise → Bilateral Collaboration Protocol →
Request Clearance Level → Counterparty Enterprise Info Level →
Is it permitted? → How much is permitted? → Does it need our approval? → Return
(desensitized) information
```

5.4 Structured Communication Protocol: Agents Should Not Transmit Natural Language

Cross-enterprise Agents should transmit not natural language (too vague, unauditable) but **authorized structured context**:

```
{
  "message_id": "msg_20260629_001",
  "sender": {
    "agent_id": "AGT-PUR-A001",
    "enterprise": "A Manufacturing Co., Ltd.",
    "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": "Procurement Manager Zhang San",
    "delegation_token": "dt_abc123",
    "expires_at": "2026-06-29T18:00:00Z",
    "max_budget_per_unit": null
  },
  "signature": "sha256:..."
}
```

The receiving Agent decides what information to return based on the `sender` identity, `request` content, bilateral **collaboration protocol**, and its own enterprise's governance rules. The entire process is not "LLM casually answering" but **rule-driven + authorization verification + returning desensitized results**.

5.5 Delegation Token

When cross-enterprise execution of a specific task is needed, a time-scoped, scope-limited **Delegation Token** is issued:

```
Issuer: Company A CEO Agent
Grantee: Company A Procurement Agent
Task: RFQ and comparison across 10 suppliers
Permissions: Read L2 supplier data, send RFQ to L1-L2 external enterprise Agents
Validity: 2 hours
Non-transferable
Cannot sign contracts (RFQ only, no order placement)
Cannot view own enterprise cost data
```

After 2 hours, permissions automatically expire. The counterparty enterprise will also verify the validity of this Token — if expired, Company B's Agent can refuse to respond.

5.6 Seven-Layer Model for Cross-Enterprise Agent Governance

Layer	Governance Content	Core Question	Cross-Enterprise Specific Requirement
Identity	Agent Identity	Who are you? Which enterprise?	Enterprise CA issued, dual-party verifiable
Capability	Capability Whitelist	What can you do?	Differentiate intra-domain vs. externally exposed capabilities
Authorization	Authorization Mechanism	Who allowed you to do it?	Own-party authorization + counterparty admission
Communication	Communication Control	Who can you communicate with?	Based on inter-enterprise collaboration protocols
Information Clearance	Information Classification	How much can you know?	Cross-enterprise L0-L4; counterparty decides how much to expose
Decision Trace	Decision Audit	Why did you decide this?	Both parties independently verifiable
Execution Control	Execution Control	Who approves final execution?	High-risk cross-enterprise operations require dual-party approval

5.7 The Meeting Paradigm Shift: From Internal Meetings to Cross-Enterprise Agent Negotiation

The evolution of meeting scenarios particularly illustrates the value of cross-enterprise Agents:

- **Phase 1 (Already Happening)**: Agent as internal meeting secretary — recording, minutes, task distribution.
- **Phase 2 (Next 2–5 Years)**: Agent represents departments in internal meetings — real-time data retrieval, analysis, suggestions.
- **Phase 3 (Key Transformation)**: **Agent represents the enterprise in cross-enterprise negotiations** — Procurement Agent and Supplier Agent automatically negotiate price, delivery, and payment terms; both CEOs only do final approval.
- **Phase 4 (Future)**: Cross-enterprise Agent autonomous coordination — every day at 2 AM, Company A's Planning Agent, Company B's Supply Agent, and Company C's Logistics Agent automatically synchronize the day's plan; the boss wakes up to a consolidated "Today's Operational Recommendations."

Meetings evolve from a synchronous communication method into computable, traceable, auto-executable cross-enterprise decision processes.

---

6. Key Business Scenarios and Implementation Pathways

6.1 Scenario 1: Cross-Enterprise Agent Identity Authentication and Admission (The Foundation of Foundations)

Before any B2B Agent application, one question must be answered: **How do you know the Agent you're communicating with genuinely represents the enterprise it claims?**

Just as the internet evolved from "naked HTTP" to "HTTPS + CA certificate system," the Agent internet needs:

- Agent identity registration (analogous to domain registration)
- Agent identity verification (analogous to SSL certificate issuance)
- Enterprise admission whitelist (analogous to firewall rules)
- Agent revocation mechanism (certificate expiration or revocation)

This is the most fundamental, most critical, and most easily overlooked infrastructure layer of cross-enterprise Agents.

6.2 Scenario 2: Cross-Enterprise Agent Approval Center (B2B Approval Gateway)

In cross-enterprise scenarios, what enterprises fear most is not the Agent answering incorrectly, but the **Agent making commitments on behalf of the enterprise that it shouldn't**.

```
Company A Agent prepares to place order with Company B
    ↓
Generate cross-enterprise execution plan
    ↓
Risk assessment (amount, counterparty trustworthiness, historical fulfillment rate)
    ↓
Approval required?
    ↓
Company A Procurement Manager approval + Company B Sales Manager confirmation
    ↓
Both Agents execute + dual sign-off
```

Product positioning: Not a chat platform, not an Agent development platform, but the last checkpoint for cross-enterprise AI actions.

6.3 Scenario 3: Cross-Enterprise Agent Operational Audit (B2B Audit Trail)

The key to cross-enterprise operations is "being able to clearly determine responsibility when something goes wrong":

- Which enterprise's Agent? At what time?
- Based on what authorization?

- What request was sent to which enterprise's Agent?
- How did the counterparty respond?
- What was ultimately executed?
- Who approved it on both sides?

This is not just a log — it is a dual-party-verifiable, legally significant cross-enterprise decision chain. Analogous to letters of credit and bills of lading in international trade today, but the object has shifted from "paper documents" to "Agent decision records."

6.4 Scenario 4: Cross-Enterprise Information Classification and Permission Management

Different collaboration relationships, different information visibility:

- Long-term strategic supplier: Can see inventory forecasts, production plans
- One-time RFQ supplier: Can only see the material specifications of the current RFQ
- Logistics partner: Can only see goods requiring transport, not pricing
- Bank: Can see financial statements, but not customer lists

This classification is not static — it is dynamically adjusted based on collaboration protocols.

6.5 Scenario 5: B2B Verification Engine

In cross-enterprise scenarios, you must never trust a counterparty Agent's "verbal answer":

```
Company B Agent answers: Inventory is 350 units
Company A Agent: Please provide verifiable proof
    ↓
Company B Agent: Generate ERP inventory snapshot + digital signature
    ↓
Company A Verification Engine: Verify digital signature validity AND inventory snapshot
authenticity
    ↓
Verification passed → "Verified: Company B currently has 350 units available"
Verification failed → "Unverified, requesting manual confirmation from Company B"
```

This means cross-enterprise Agent outputs always have verifiable factual sources as support.

6.6 Scenario 6: Cross-Enterprise Agent Collaboration Scheduler (B2B Coordinator)

When Enterprise A has 10 Agents and Enterprise B has 10 Agents, the possible interaction combinations explode:

- Two Agents mutually calling each other, infinite loops
- Multiple Agents performing duplicate operations on the same transaction
- Priority conflicts (Company A's urgent order vs. Company B's capacity ceiling)

This requires a **cross-enterprise Coordinator** — not belonging to any single enterprise, but serving as a neutral collaboration scheduling layer.

---

7. Business Model: Enterprise Agent Network

7.1 The Biggest Opportunity Is Not Developing Agents, but Becoming the Connective Layer Between Agents

If every company has an "Enterprise Digital Representative (Enterprise Agent)" and enterprises complete massive standardized collaboration through Agents, then:

> **1 million companies → 1 Agent per company → communicating with each other**

This forms a brand-new collaboration network analogous to the internet. And the real business opportunity lies in: **Who builds the connective layer of this network?**

Not an "Agent Store" (selling Agents), but:

> **Enterprise Agent Network**
>
> Enabling Agents from different enterprises to:
>
> - **Discover each other**: Procurement Agent finds Supplier Agent (analogous to DNS)
> - **Authenticate each other**: Verify counterparty identity and trust level (analogous to SSL/CA)
> - **Communicate securely**: Structured, auditable, clearance-leveled communication (analogous to HTTPS + advanced protocols)
> - **Collaborate trustworthily**: Negotiate, contract, execute, settle (cross-enterprise business processes)

7.2 Don't Build an "Agent Governance Platform" — Build "B2B Agent Infrastructure"

Just as in the security industry: no one buys "network security" — people buy firewalls, bastion hosts, database audit, and vulnerability scanners.

Similarly, don't build an "Agent Governance Platform" on day one. Instead, take one specific cross-enterprise business scenario and push governance to the extreme.

Recommended entry point priorities:

1. **Cross-Enterprise Agent Identity Authentication Service**: Most foundational, most critical — no identity, no collaboration
2. **Cross-Enterprise Agent Approval Center (B2B Approval Gateway)**: The mandatory checkpoint for all cross-enterprise high-risk operations
3. **Cross-Enterprise Agent Audit Service (B2B Audit Trail)**: Dual-party-verifiable decision chain records
4. **Industry-Vertical B2B Agent Networks**: E.g., manufacturing procurement Agent network, cross-border logistics Agent network

7.3 The Three-Stage Internet Evolution Analogy

- **Stage 1: Web Interconnection** (Browser accessing websites) — Infrastructure: DNS + HTTP + Browsers

- **Stage 2: API Interconnection** (System calling system) — Infrastructure: REST + OAuth + API Gateway
- **Stage 3: Agent Interconnection** (Intelligent collaboration between enterprises) — Requires new infrastructure: Agent Identity Authentication + Permission Protocols + Context Sharing + Process Orchestration + Audit

Each generation of interconnection gave birth to new infrastructure giants. The Agent interconnection era will be no different.

7.4 The Complete Picture of Next-Generation Infrastructure

If Agents truly begin communicating at scale, a complete set of new requirements will emerge, forming the full "Agent Internet" infrastructure stack:

```
┌──────────────────────────────────────┐
│   Industry B2B Agent Applications │  ← Procurement / Logistics / Finance / Supply Chain
...
├──────────────────────────────────────┤
│   Cross-Enterprise Process      │  ← Procurement → Supplier → Logistics → Payment →
Legal
│   Orchestration                 │
├──────────────────────────────────────┤
│   Cross-Enterprise              │  ← Shared Context, Unified Protocols, Event-Driven
│   Collaboration Hub             │
├──────────────────────────────────────┤
│   Governance Layer              │  ← Policy, Permission, Approval, Risk
├──────────────────────────────────────┤
│   Identity & Trust Layer        │  ← CA, Digital Signatures, Identity Auth, Admission
├──────────────────────────────────────┤
│   Communication Protocol Layer  │  ← Structured Messaging, Encryption, Conversation
Clearance
└──────────────────────────────────────┘
```

The greatest business opportunity may not be developing more standalone Agents, but becoming the connective layer between Agents: defining collaboration protocols, identity authentication, permission systems, context sharing, and process orchestration. Whoever enables Agents from different enterprises to complete business collaboration securely, at low cost, and with high trustworthiness, will have the opportunity to become the core infrastructure of the next-generation enterprise internet.

7.5 Responsibility Boundaries: Cloud Providers vs. Enterprise Runtime

In the evolution of Agent infrastructure, a key question is: **Which capabilities should be provided by cloud providers, and which must be the responsibility of the enterprise's own Runtime?**

Layer	Responsible Party	Core Objective	Why?
GPU Scheduling	Cloud Provider	GPU utilization, SLA, cost	Enterprises shouldn't care which GPU is idle — just like not caring which server handles an HTTP request
Model Serving	Model Vendor	Model capability, inference performance	Training and inference optimization is the model vendor's core competency
Model Routing	**Enterprise Runtime**	Select optimal model for each business task	Cloud providers don't know your business semantics — the same sentence could be legal or financial; only the enterprise knows
Workflow Orchestration	**Enterprise Runtime**	Agent collaboration, task decomposition	Business processes are unique organizational knowledge
Cost per Task	**Enterprise Runtime**	Lowest cost per completed business task	Enterprises care about "how much per customer service ticket," not "how much per million tokens"
Business KPI	Enterprise	ROI, revenue, efficiency	Final business judgment always rests with the enterprise

Core conclusion: GPU scheduling and model serving will be standardized by cloud providers, like today's IaaS. Model Routing, Workflow Orchestration, and Cost-per-Task optimization — **capabilities closer to business should be borne by the enterprise Runtime**. This is precisely the positioning of TEA Runtime — it does not replace cloud providers, but serves as the business execution layer built atop them.

7.6 Inference Planner: From Model Routing to Inference Planning

The core capability of TEA Runtime is not simple "model routing" — it is **Inference Planning**.

Why Model Router Is Not Enough

Traditional model routing is a simple mapping: `Task → Model`. But Agent-era tasks are far beyond what "pick one model" can solve. A single user request might be decomposed into search (small model), reasoning (strong model), verification (small model), and polishing (small model) — **it's not about handing the entire task to one model, but decomposing the task into subtasks and selecting the most suitable model for each**.

This is precisely the core responsibility of the Inference Planner:

```
User Request
     │
     ▼
Intent Agent (Understand Intent)
     │
     ▼
Task Planner (Decompose Tasks)
     │
```

```
      ├── Search → Small Model (7B)
      ├── Summarize → Small Model (7B)
      ├── Reason → Strong Model (DeepSeek V4)
      ├── Verify → Verification Model
      └── Polish → Small Model (7B)
      │
      ▼
Result Synthesis
```

Mechanisms for Ensuring Routing Accuracy

No system can guarantee every routing decision is "correct." TEA's approach is not to pursue "never wrong," but to establish a continuous improvement mechanism:

1. **Data-Driven Routing**: The Runtime continuously records each task's model selection, cost, latency, and user satisfaction. After accumulating data, it automatically learns the optimal strategy. It's not because an engineer thinks DeepSeek is best for contract review — it's because **data proves** it offers the lowest cost with the highest satisfaction in this scenario.
2. **Confidence-Driven Degradation**: When the Router's confidence in its selection is below a threshold (e.g., <70%), it doesn't bet on a single model. Instead, it automatically upgrades to multi-model parallel execution + voting, or even escalates to human confirmation. Like autonomous driving — when unsure, hand over to the pilot.
3. **Benchmark Agent**: A dedicated Agent that automatically executes tens of thousands of standard test questions every night at 2 AM, continuously evaluating each model's latest performance across domains like sales, legal, R&D, and finance, forming a daily-updated leaderboard. The Runtime automatically adjusts routing strategies based on this.

The Ultimate Form of Inference Planner

It does not merely "select models" — it plans the entire inference process: understanding intent, decomposing tasks, allocating models, orchestrating execution, verifying results, and re-planning based on feedback. It has essentially become a **high-intelligence Planning Agent**. This is also why it must be borne by the enterprise Runtime, not cloud providers: understanding business semantics, grasping enterprise context, and complying with organizational governance rules — only the enterprise itself can do this.

7.7 Three-Tier Architecture for AI Application Development

Based on the above analysis, TEA proposes a **three-tier architecture** for enterprise AI application development. This is both a technical architecture and an organizational division of labor:

```
Business Layer (Business Capabilities)
──────────────────────────────────────
Contract Review / Customer Reply / Sales Analysis
Purchase Approval / Knowledge Query / Code Generation
         │
         ▼
Business Capability API
──────────────────────────────────────
reviewContract() / replyCustomer()
analyzeSales()  / approveExpense()
```

```
          │
          ▼
      TEA Runtime
──────────────────────────────────
Intent / Planning / Policy
Routing / Memory / Audit
          │
          ▼
Foundation Model Layer
──────────────────────────────────
GPT / DeepSeek / Qwen / GLM
Claude / Llama / Huawei Ascend Cloud
```

This architecture follows the core design principle established in the Prologue: **Business First, Models Second**. The business layer only calls Business Capability APIs — it has no knowledge of which model lies beneath. Models can be swapped at any time — GPT today, DeepSeek tomorrow, Huawei Ascend Cloud the day after — **zero changes to business code**.

Enterprises express not "which model to use," but Business Policy:

```
Contract Review:
    quality: high
    budget: medium
    latency: 10s

Customer Reply:
    quality: medium
    budget: low
    latency: 2s    # Must respond within seconds
```

The Runtime translates these policies into concrete execution plans — which model to select, whether to decompose tasks, whether to use multi-model voting, whether to execute during off-peak hours. **Models are merely resources for implementing policies; policies are the truly stable enterprise assets.**

The core value of this positioning: When DeepSeek, OpenAI, Huawei Ascend Cloud, or any future new model emerges, enterprises need not modify business logic. The Runtime automatically re-selects the optimal execution plan based on policies and data. This is like migrating databases from Oracle to MySQL — if the code was written through an ORM with business logic, migration cost is minimal; if it was raw SQL, massive rewrites are needed. **TEA Runtime is the ORM layer of the AI era — isolating business logic from the continuous evolution of underlying models.**

---

8. Conclusion: From Agent Development to Agent Interconnection — Two Battlefields, One Governance Logic

8.1 Core Assertions

From the unique perspective of cross-enterprise Agent collaboration, this whitepaper has demonstrated the following core assertions:

1. **AI's business value lies not in the model itself, but in "model + specific scenario + workflow."** Big models are infrastructure; small applications are the value carriers.
2. **"Controlled and Trusted" applies equally to internal enterprise Agents and cross-enterprise B2B Agents, but the two impose governance requirements at different levels.** Internally, Controlled-and-Trusted is a "best practice" — approvals, permissions, and audits are baseline for safe Agent operation. In cross-enterprise collaboration, it escalates to a "hard requirement" — without identity authentication, information classification, and dual-party auditing, collaboration cannot even begin.
3. **Cross-enterprise Agents add entirely new dimensions of challenge atop internal governance.** No central authority, information asymmetry as a hard requirement, complex responsibility attribution, high difficulty of identity authentication — all of these demand that the governance system upgrade from "enterprise-internal rules engine" to "cross-enterprise collaboration control plane."
4. **The core of enterprise-grade Agents is: separation of capability and permission, Default Deny, Decision Trace, and Human Accountability.** These four CAEM principles are the governance baseline internally and the prerequisite for safe operation in cross-enterprise scenarios.
5. **Agent communication must transcend natural language, moving toward structured, clearance-leveled, verifiable protocols.** Internally, this is an efficiency improvement; in cross-enterprise collaboration, every communication must include identity credentials, authorization tokens, clearance labels, and digital signatures — not "chat," but "formal inter-enterprise commercial exchange."
6. **The greatest business opportunity is not developing Agents, but becoming the connective layer between Agents.** Enterprise Agent Network — defining collaboration protocols, identity authentication, permission systems, context sharing, and process orchestration — will be the core infrastructure of the next-generation enterprise internet.

8.2 From "Smarter AI" to "More Trustworthy AI"

The most commercially valuable AI products in the coming years will not be the "smartest AI," but:

- **Within the enterprise: governance platforms that enable Agents to integrate into workflows safely and compliantly**
- **Between enterprises: connective infrastructure that enables Agents from different enterprises to collaborate securely**
- **Protocol layers that define Agent identity, permission, and auditing standards**
- **Applications that deeply understand industry processes and seamlessly integrate into enterprise workflows**

Big models provide intelligence; **Controlled-and-Trusted Agent governance architecture — whether internal or cross-enterprise — is responsible for transforming intelligence into stable, measurable, sustainable, and auditable business value.**

8.3 The Final Analogy

If AI is electricity and Agents are appliances, then:

- **Internal enterprise Agents** are like appliances in a single building — same meter, same distribution box, same property management. Controlled-and-Trusted here is the "building electrical code" — ensuring every appliance is safe, no short circuits, no overloads.
- **Cross-enterprise B2B Agents** are like power trading between different cities and different grids — requiring substations, transmission protocols, metering and settlement standards, and mutual trust mechanisms. Controlled-and-Trusted here upgrades to "cross-grid dispatch protocols" — without these protocols, the power cannot even be transmitted.

Today, everyone is building appliances (developing Agents). But the truly great opportunity is building the power grid (Agent interconnection infrastructure) — and a unified Controlled-and-Trusted governance system that extends from internal to cross-enterprise is the most critical "dispatch protocol" of this grid.

8.4 Paradigm Shift: From "How to Apply AI in the Enterprise" to "How AI Becomes Part of Enterprise Software"

Reviewing the entire argument of this whitepaper, one can clearly discern a **Paradigm Shift**:

For the past month — indeed, the past two years — the AI industry has been asking the same question:

> "How do we apply AI in the enterprise?"

This question naturally places AI at the center, with the enterprise as the object being transformed. The resulting approach: find a scenario → match a model → write prompts → fine-tune. When the model upgrades, everything gets torn down and rebuilt.

But TEA approaches from the opposite direction:

> **"How does AI become part of enterprise software?"**

The premise of this question is: **Enterprise software itself is the center; AI is merely one type of execution resource.** Just like databases, message queues, and cloud servers — they were once "new technologies" too, but ultimately all faded into the background, becoming the underlying infrastructure of enterprise software. Models will follow the same path.

Every major abstraction in enterprise software over the past decades — databases turning storage into data services, operating systems turning hardware into compute resources, Kubernetes turning servers into compute clusters — has fundamentally done one thing:

> **Abstract rapidly changing underlying technologies into stable upper-layer capabilities.**

The most valuable platform of the AI era likely won't be the one with the strongest model, but the one that **abstracts continuously evolving models, Agents, tools, and compute into business capabilities that enterprises can rely on for the long term.**

This is the ultimate positioning of TEA: **not "yet another Agent framework," but a stable abstraction layer between enterprise software and AI technology — enabling enterprises to develop against business capabilities while AI technology can continuously evolve without disrupting upper-layer business.**

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

Appendix: Key Terminology

Term	Definition
Controlled Agent	An Agent that completes tasks within explicitly defined boundaries, and is auditable, traceable, and predictable. In cross-enterprise scenarios, boundaries are jointly defined by bilateral collaboration protocols.
Trusted Enterprise Agent	An Agent that an enterprise dares to entrust with critical business (including cross-enterprise business), satisfying six trust requirements: Identity, Capability, Information, Decision, Execution, and Responsibility. Represents the enterprise's digital identity externally.
Cross-Enterprise Agent / B2B Agent	An Agent representing different enterprises in collaboration. Its fundamental distinction from internal enterprise Agents lies in operating within a cross-trust-domain environment.
Enterprise Agent Network	A network formed by multiple enterprises' Agents communicating and collaborating with each other. Analogous to the internet, but the communicating entities are Agents rather than browsers or APIs.
CAEM (Controlled Agent Execution Model)	A four-principle governance model centered on Default Deny, Explicit Capability, Decision Trace, and Human Accountability. Each principle is stricter in cross-enterprise scenarios than in internal scenarios.
Bounded Intelligence	The philosophical foundation of CAEM. Every Agent operates within explicitly defined capability, authorization, and decision boundaries.
Decision Trace	Records not only what was executed, but also the complete reasoning chain of why it was understood that way, how it was planned, what facts it was based on, and how results were verified. Must be dual-party verifiable in cross-enterprise scenarios.
Enterprise AI Action Gateway	The last checkpoint for all enterprise AI actions. In cross-enterprise scenarios, it is the mandatory approval and verification layer before both parties' Agents execute operations.
Delegation Token	A time-scoped, scope-limited, non-transferable temporary permission credential. Automatically revoked after cross-enterprise task completion.
Conversation Clearance	The clearance label (L0-L4) for each cross-enterprise Agent communication, determining what information and how much the receiver may return.
Agent IAM	Transplanting enterprise Identity and Access Management (IAM) systems into the Agent domain. In cross-enterprise scenarios, extended to cross-domain identity and access management.

Cross-Enterprise Collaboration Protocol	The Agent collaboration rules agreed between two enterprises, including communication permissions, information clearance levels, approval requirements, and responsibility attribution.
Inference Planner	The core component of TEA Runtime. Not simple model routing, but planning the entire inference process — intent understanding, task decomposition, model allocation, execution orchestration, and result verification.
Business Capability Interface	The interface abstraction TEA exposes to developers. Enterprises call `contract.review()` rather than `gpt.chat()`; underlying models can be swapped at any time.
Four-Tier Output Classification	Agent outputs classified by risk level into 🟢Information, 🔵Recommendation, 🟡 Decision Proposal, and 🔴Action Execution, each with different automation and approval strategies.

Version: v2.0
Date: June 2026

Publisher: Chaos Lab

Contact: 2819699195@qq.com