Enterprise Agent Foundations: From Autonomous Action to Governed Work

How events, memory, permissions, sandboxes and human intervention support long-horizon work

Online technical whitepaper

6 min read

Published / updated
2026-11-22

Intended readers

CIOs, CTOs, architects, security leaders and enterprise procurement teams

EXECUTIVE ANSWER / DIRECT ANSWER

The purpose of an enterprise agent foundation is not unrestricted autonomy. It is to give every task an explicit goal, traceable state, least-privilege access, verifiable execution and a working intervention path. Models interpret and plan; the foundation turns plans into organized, stoppable and reviewable work.

01 / KEY QUESTIONS

Key questions answered in this paper

Q1

Why cannot a general chat model directly own an enterprise process?

Enterprise work involves state, permissions, tools, handoffs and accountability. Without task state and execution boundaries, a generated answer is not a reliable operational action.

Q2

What should be verified first?

Verify that one real task chain can be recorded, paused, reviewed and recovered before comparing model rankings or agent counts.

Q3

Does controlled self-evolution mean autonomous production changes?

No. Candidate improvements require sandbox validation, human approval, asset versioning and controlled release. Availability depends on the delivered version and project acceptance.

02 / Full paper

01

From conversational output to accountable action

The industry question is shifting from “Does it answer well?” to “Does it act within its authority?”

The 2026 agent governance report from CAICT and China University of Political Science and Law describes perception, memory, planning and action as key agent components. It also highlights a shift from output errors toward uncontrolled action and cascading risk. Enterprise governance therefore needs to cover the full task lifecycle, not only the final response.

NIST launched its AI Agent Standards Initiative in 2026 with work on industry-led standards, community protocols, agent identity and security research. This does not mean one protocol has unified the market. It does show that identity, interoperability and security evaluation are becoming foundational engineering concerns.

- ✓ Goal: outcomes, constraints and stop conditions are explicit.
- ✓ State: inputs, decisions, tool calls, waits and results can be recovered.
- ✓ Accountability: authorization, execution, review and final responsibility remain traceable.

Chapter sources [1] [6]

02

Six foundations of an agent runtime

VROS Agent organizes multi-role work through applications, events, memory, permissions and execution boundaries.

Its self-developed Lianshan programming language and VR Lang are used to write and publish agent applications. A unified conversational entry captures intent, while typed event streams represent decisions, tools, human intervention and Worker execution. Long-horizon work no longer depends on one opaque context window. Actual language features and modules depend on the delivered version.

Session, task and long-term memory serve different time horizons. Personal, public and collaborative spaces govern resource visibility. Decision and execution are separated, with side effects dispatched to Worker sandboxes. Ports and adapters isolate storage, models, tools and Worker implementations so components can evolve without coupling the whole system.

Foundation	Core problem	Acceptance evidence
Applications & versions	Unreusable workflows and uncontrolled changes	Versions, change records and rollback
Event-first runtime	Opaque, unrecoverable long tasks	Event sequence, state and replay points
Persistent memory	Lost context and ungoverned experience	Scope, retrieval basis and deletion
Three spaces	Unclear resource visibility	Authorization across personal, public and shared assets
Worker sandboxes	Direct high-impact side effects	Isolation, tool allowlists and execution logs

Chapter sources [2] [9]

03

Treat prompts, tools and memory as security boundaries

Agent security governs an action system, not just a chat interface.

OWASP’s agentic threat guidance covers goal hijacking, tool misuse, identity and privilege, memory poisoning and inter-agent communication. Web pages, documents, retrieved passages, tool output and messages from other agents should all be treated as untrusted input rather than inherently safe context.

Practical controls include task-scoped least privilege, separation of read and write actions, restricted egress, human confirmation for high-impact steps, version review for tools and Skills, and independent execution logs. A model’s stated intention not to perform an action is not a system control.

- ✓ Identity: users, agents, Workers and tools have verifiable identities.
- ✓ Permissions: denied by default, granted for the task and revocable afterward.
- ✓ Budgets: cap runtime, calls, spend and resource use with circuit breakers.
- ✓ Intervention: high-impact or irreversible actions can pause for accountable human handling.

Chapter sources [9] [3] [4]

04

Evaluate the task chain, not a single demonstration

A successful demo does not establish operational reliability.

Break work into observable stages and record input validity, planning quality, tool-call correctness, deliverable quality, human rework and exception recovery. Run the same task repeatedly, retain failures and test long-horizon work for goal drift, context truncation and inconsistent intermediate state.

NIST AI 800-4 emphasizes post-deployment monitoring across functionality, operations, human factors, security and compliance. Enterprises can use these categories to design dashboards and review routines, but thresholds and owners must be defined for the specific use case. A general benchmark is not an acceptance result.

Dimension	Suggested measures	Conditions to preserve
Task quality	Stage completion and deliverable acceptance	Task difficulty and human baseline
Execution reliability	Retries, recovery and effective stops	Repeated runs and fault injection
Security & permissions	Blocked violations, approvals and log completeness	Indirect injection and tool changes
Business cost	Model, compute and review cost per task	Peak concurrency and failed retries

Chapter sources [7] [5] [1]

05

Boundaries for controlled evolution and interoperability

Evolution should produce reviewable candidate assets, not bypass change management.

VROS Agent positions RSI as controlled self-evolution: execution review, experience extraction, sandbox validation, human approval, asset versioning and controlled release. Unverified experience does not enter live business or device workflows. Availability remains subject to the delivered version and project acceptance.

Interoperability does not mean unconditional interconnection. MCP, other agents and enterprise systems still require explicit identity, data scope, action permissions, protocol versions and failure handling. Standards reduce connection friction; they do not remove supply-chain or authorization risk.

- ✓ Keep candidate improvements separate from production assets.
- ✓ Archive evidence, approver, version and rollback path together.
- ✓ Reassess changed protocols and tool definitions instead of relying on one-time trust.

Chapter sources [6] [9] [1]

06

Procure one acceptable task chain first

Agent counts, polished interfaces and model leaderboards do not prove production fit.

Start with a frequent, bounded task that has a human baseline. Ask suppliers to delineate model and runtime responsibilities, data flows, permission enforcement, log fields, failure behavior, asset ownership and exit arrangements. On-premises tests should include model serving, retrieval, Workers, networking and storage in one compatibility plan.

ISO/IEC 42001 specifies an organizational AI management system and NIST AI RMF provides a voluntary risk-management approach. This whitepaper references their methods only. It does not claim that AIBotStore or VROS Agent is certified, and it does not replace legal or compliance assessment.

- ✓ Run an end-to-end PoC on real work and retain raw evidence.
- ✓ Contract for versions, permissions, data, acceptance, handover, change and exit.
- ✓ Set risk-based review cadence and reassess material model or tool changes.

Chapter sources [8] [4] [7]

03 / CHECKLIST

Pre-project checklist

Use these items to structure a PoC plan, procurement questions or acceptance checklist.

01	Tasks have explicit inputs, outputs, constraints, stop conditions and human baselines
02	Events and logs reconstruct decisions, approvals, tool calls and execution
03	Permissions are least-privilege and sensitive actions can require confirmation
04	Memory, knowledge, tools and Skills have provenance, versioning and change controls
05	Repeated tests, fault injection and post-deployment monitoring are in place
06	Controlled evolution produces candidates that require validation and approval

04 / FAQ

Frequently asked questions, answered directly

01 **Is VROS Agent a foundation model?**

No. VROS Agent is a multi-user agent operating system for applications, task state, events, memory, permissions and Worker execution. Models are connected and adapted according to the project environment.

02 **Why does event sourcing matter for long-horizon work?**

Events turn long tasks into observable, recoverable state changes. Teams can locate and handle a specific failed stage instead of retaining only a final generated answer.

03 **Can the system run completely unattended?**

That depends on task risk, authorization and acceptance. High-risk operations, device control, financial judgment and other consequential actions should retain accountable human oversight.

04 **Does referencing NIST or ISO imply certification?**

No. The frameworks organize technical and governance questions. Certification, compliance and project suitability require separate assessment.

05 / REFERENCES

References and public sources

Chinese references follow GB/T 7714 style and English references follow APA style. Accessed November 22, 2026. External material informs industry and technical methods; it does not imply endorsement or certification of AIBotStore.

[1] China Academy of Information and Communications Technology, & China University of Political Science and Law. (2026). Agent governance research report (2026) [Report].
https://www.caict.ac.cn/kxyj/qwfb/ztbg/202609/P020260916574260931247.pdf →

[2] China Academy of Information and Communications Technology. (2026). AI Native research report 1.0: Enterprise transformation framework and practice [Report].
https://www.caict.ac.cn/kxyj/qwfb/ztbg/202610/P020261022558735347833.pdf →

[3] China Academy of Information and Communications Technology. (2026). Artificial intelligence security governance blue book [Report].
https://www.caict.ac.cn/kxyj/qwfb/bps/202601/P020260109784447548497.pdf →

[4] National Institute of Standards and Technology. (2023). AI Risk Management Framework (AI RMF 1.0).
https://www.nist.gov/itl/ai-risk-management-framework →

[5] National Institute of Standards and Technology. (2024). Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile (NIST AI 600-1).
https://doi.org/10.6028/NIST.AI.600-1 →

[6] National Institute of Standards and Technology. (2026). AI Agent Standards Initiative.
https://www.nist.gov/caisi/ai-agent-standards-initiative →

[7] Rao, A., Keller, A., Kalra, N., Steed, R., Kwegyir-Aggrey, K., Klyman, K., Staheli, D., & Bergman, A. (2026). Challenges to the monitoring of deployed AI systems (NIST AI 800-4).
https://doi.org/10.6028/NIST.AI.800-4 →

[8] International Organization for Standardization. (2023). ISO/IEC 42001:2023—Information technology—Artificial intelligence—Management system.
https://www.iso.org/standard/42001 →

[9] OWASP GenAI Security Project. (2025). Agentic AI—Threats and mitigations.
https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/ →

Scope statement

This whitepaper supports technology selection and project planning and does not replace legal, compliance, safety or investment advice. Actual VROS Agent, AIBotStore, hardware-integration and controlled-evolution scope depends on the delivered version, project contract and acceptance. Industry task examples do not represent delivered customers or achieved outcomes.