Enterprise Agent Deployment

A Technical Brief | 3ZY TECH | 2026

1. From chatbot to closed loop

The first wave of enterprise LLM projects shipped a chat window over internal documents. Most of them stalled: usage decayed, answers could not be traced, and nothing changed inside the workflow.

An agent differs in three ways: it calls systems, it keeps state across steps, and it verifies its own output before handing the result back to a human.

2. Reference architecture

- Planner decompose the goal into auditable steps
- Tools idempotent calls with explicit error codes
- Verifier rule checks, cross-source checks, self review
- Memory short-term task state, long-term preferences
- Guardrail human approval for payment, deletion, outbound send

The verifier is the part most often skipped, and the part that decides whether the system can be trusted in production.

3. Orchestration

A linear chain is rarely enough. Model the task as a directed acyclic graph: parallel branches for independent lookups, retry with fallback to cached data, and a visible degraded flag on any node that had to fall back.

Every node should log its inputs, outputs, latency and fallback state, so a failed run can be replayed instead of guessed at.

4. Go-live checklist

1) Is the success metric end-to-end, not per-step accuracy?
2) Can every write action be rolled back?
3) Are permissions enforced at field level, not just page level?
4) Is every tool call logged with actor, time and input summary?
5) Does the model and its data stay inside the intranet?
6) Which team owns the agent after launch?

If three of these cannot be answered, the project is not ready for production. Narrow the scope to one verifiable task first.

Contact: support@3ziye.com | https://3ziye.com