

DISRUKTECH RESEARCH

Enterprise Agent Architecture™

A practical framework for AI agents that use business context, reason over enterprise knowledge, act through authenticated tools and remain under accountable human and operational control.

VERSION 1.0 | WEB AND PDF EDITION | 9 OCTOBER 2026 | ARTURO ESPINOSA

Independent enterprise technology and AI firm

Executive definition

An enterprise agent is defined by accountable action, not conversation.

It combines business context, enterprise knowledge, intelligence and reasoning, authenticated tools and enterprise systems to pursue a measurable outcome inside explicit authorization, governance and human-oversight boundaries.

The framework is designed for executives, enterprise architects, SAP leaders, AI teams, security teams and process owners evaluating systems that move beyond answering questions and begin preparing or executing business actions.

Five principles

- Business outcome and accountable ownership come before model selection.
- Knowledge must retain permission, provenance and scope.
- Every tool call is an authenticated business action, not an informal prompt.
- Autonomy expands only where the organization can observe, constrain and reverse it.
- Governance and human oversight operate across every layer.

The six capability layers

The layers describe responsibilities, not a mandatory vendor stack. A single product may cover multiple layers; ownership must still remain explicit.

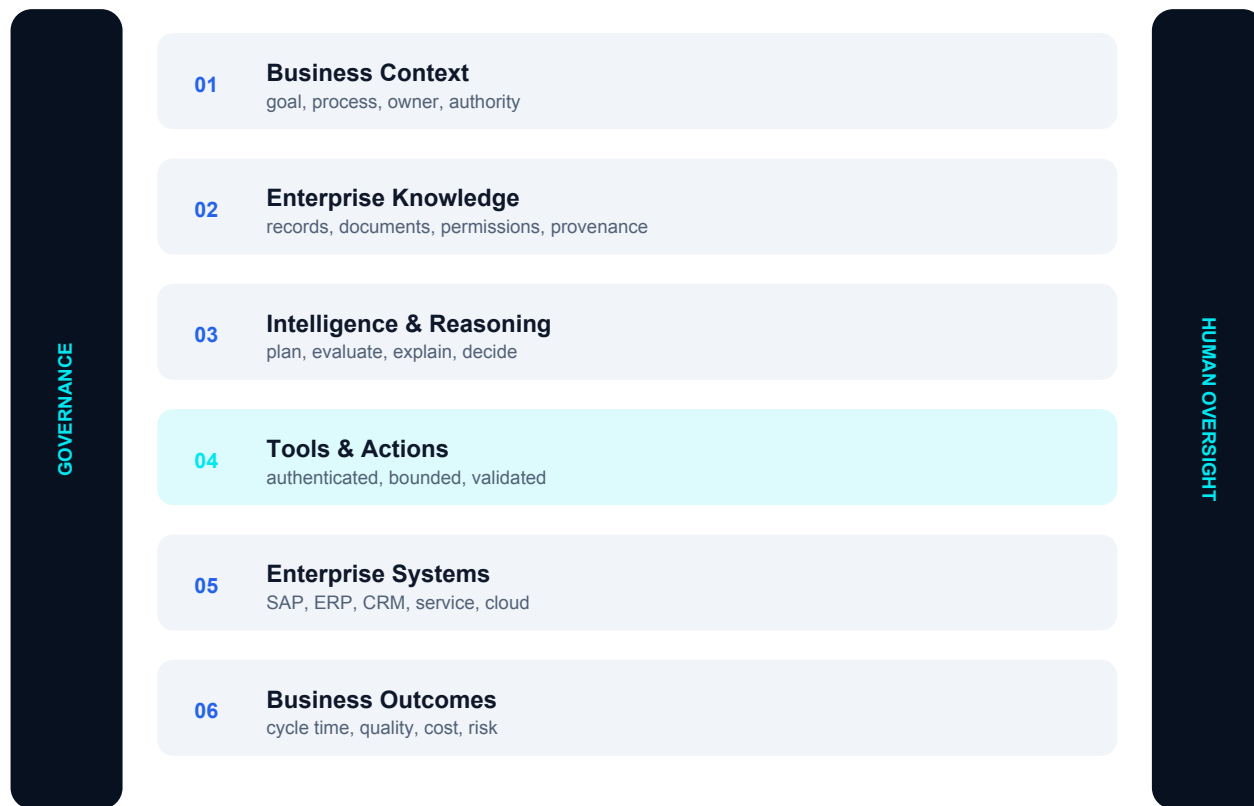


Figure 1. EAA™ capability stack. Governance and human oversight span all six layers.

Layer design questions

Layer	What must be explicit	Executive question
Business Context	Goal, process, owner, policy and authority	Which outcome is the agent accountable for?
Enterprise Knowledge	Allowed data, documents, permissions and sources	Can every answer and action be traced?
Intelligence & Reasoning	Planning, evaluation, explanations and confidence	How is quality tested before and after launch?
Tools & Actions	Authenticated interfaces, input validation and limits	Which actions are allowed and reversible?
Enterprise Systems	Supported APIs and native controls	Does integration preserve system-of-record controls?
Business Outcomes	Cycle time, quality, cost, risk and ownership	Who owns the metric after launch?

Autonomy is a design variable

Level	Agent authority	Required controls	Human role
Analyze	Reads and explains; no business record changes	Read-only identity, source citations, data scope, evaluation	Interprets and acts
Recommend	Prepares an action and its evidence	Policy check, explanation, approval queue, time limit	Approves or rejects
Execute within policy	Acts inside defined thresholds	Dedicated identity, least privilege, segregation of duties, audit, reversal and stop switch	Reviews exceptions and samples

As authority increases, control must become more explicit.

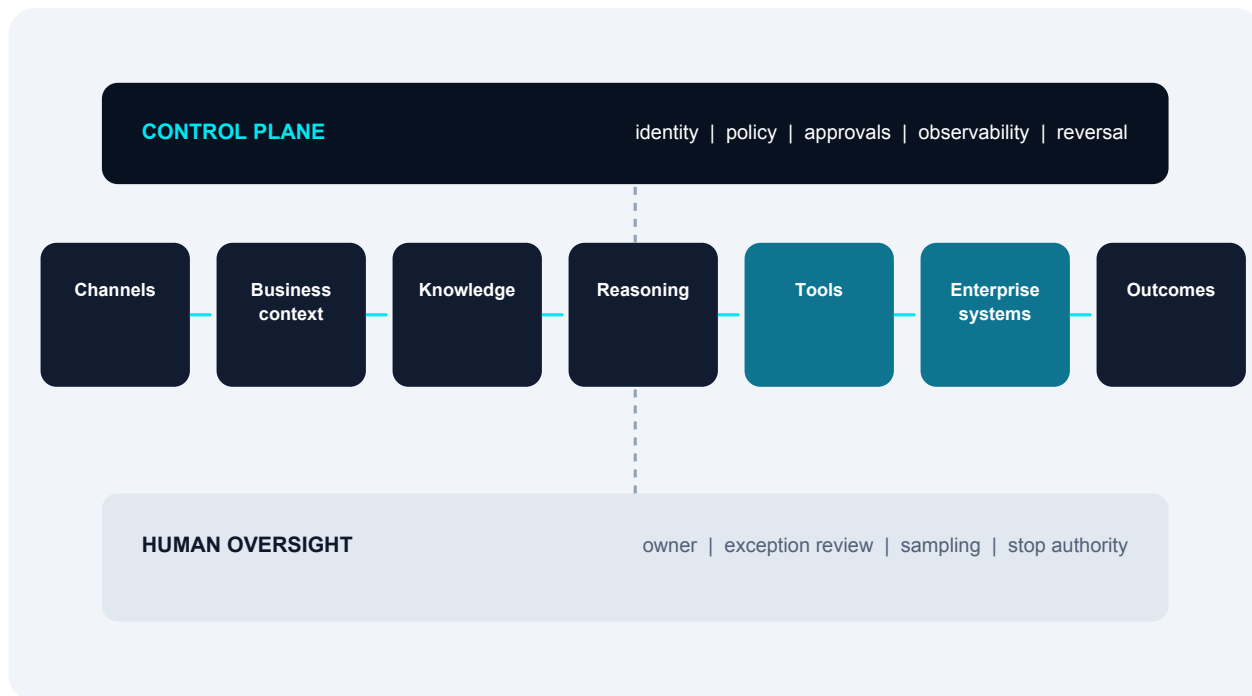
Autonomy should not be assigned once for an entire agent. It should be assigned per action, risk class, value threshold and operating condition.

Reference application: supplier exception

Layer	Worked example
Context	A supplier invoice is blocked by a price variance. The AP owner wants faster resolution without weakening approval policy.
Knowledge	Purchase order, goods receipt, contract amendment, supplier master and prior exception history.
Reasoning	Determine whether evidence supports the variance and whether confidence exceeds the review threshold.
Tools	Read invoice and PO; retrieve evidence; prepare release or escalation; never alter supplier bank data.
Systems	SAP S/4HANA through supported APIs and existing authorization controls.
Outcome	Reduce exception cycle time while preserving payment controls and a complete audit trail.

This reference application is illustrative and uses fictitious data. It is not represented as a client deployment.

Operating control plane



- Identity: the agent has its own identity and cannot borrow a person's credentials.
- Policy: actions are checked against versioned rules and thresholds.
- Evaluation: test sets cover normal cases, edge cases and attempts to exceed authority.
- Observability: inputs, outputs, decisions, tool calls, latency, cost and exceptions are visible.
- Reversal: actions are reversible where possible; non-reversible actions require stronger approval.
- Stop authority: a named owner can suspend the agent immediately.

Applying the framework

Stage	Decision	Deliverable
1. Frame	Which bounded task and outcome?	Agent brief and KPI
2. Map	Which knowledge, tools and systems?	Access and tool contract map
3. Bound	What may it do alone?	Autonomy and control matrix
4. Evaluate	Does it behave under normal and adversarial cases?	Evaluation suite and go-live criteria
5. Operate	Who owns quality, cost and exceptions?	Runbook, dashboards and stop process

The model vendor is selected after the operating requirements are clear. EAA™ is intentionally vendor-neutral.

Publication status and citation

Suggested citation

Espinosa, Arturo. Enterprise Agent Architecture™. Version 1.0. DISRUKTECH Research, 9 October 2026.
<https://disruktech.com/research/enterprise-agent-architecture>

Status

Practitioner framework. This publication does not claim academic peer review. Examples are reference designs unless explicitly identified as client engagements.

Version history

Version	Date	Change
1.0	9 October 2026	Formalized capability layers, autonomy model, control plane, reference application and citation metadata.

Author

Arturo Espinosa is Founder and Managing Partner of DISRUKTECH. His work focuses on SAP transformation, enterprise AI, solution architecture and U.S.-Mexico technology delivery.

disruktech.com | info@disruktech.com