

The Business Capability Advantage

Governed Execution for Autonomous Business

AUTONOMY | The governed execution platform for autonomous business

Central thesis

AI agents do not primarily need another development studio. They need a stable, governed way to understand and act on the business.

Release qualification note. This paper describes the intended first-public-release architecture. Runtime behaviors still in the r12 installed-system qualification gate are explicitly identified rather than presented as already shipped.

Executive opening

Enterprise AI has become dramatically better at understanding language, summarizing information, planning work, and choosing tools. Yet the hardest question remains: how should an intelligent agent safely act on a real business?

A business operates through durable business capabilities: retrieve a purchase order, create a supplier, approve a requisition, explain an invoice hold, update a customer, onboard an employee, resolve a service case. Those capabilities must keep their meaning even as AI models, agent frameworks, applications, integrations, and implementation technologies change.

Sirvisetti Autonomy is the governed execution platform for autonomous business. Applications remain systems of record. Autonomy becomes the governed system of action between business intent and consequential execution.

What Autonomy is - and is not

Autonomy is not primarily an AI agent builder, low-code workflow studio, iPaaS, API gateway, or connector catalog. Agent Studio is declarative: purpose, Model Profile, eligible capabilities, authority/policy, confirmations, invocation modes, and execution limits. It is not a generic node-and-wire agent workflow canvas.

Applications remain systems of record. Autonomy becomes the governed system of action.

Ten differentiators

1. Business capability is the universal primitive

The same canonical business capability can be used by people, built-in Agents, external agents, APIs, Triggers, Actions, and Processes. Actions and Processes can both implement governed capabilities; Policies constrain execution; Application Packs supply target-system implementations; Bindings resolve the appropriate implementation at runtime. Example: Procurement Agent, REST client, and Chatbot user can all request Retrieve Purchase Order through the same capability.

Release status. Core model established; full cross-initiator r12 qualification is pending.

2. ABCS provides vendor-neutral business semantics

ABCS - the Autonomous Business Capability Specification - defines canonical business semantics independently from Sirvisetti, AI vendors, workflow engines, integration tools, and backend applications. Retrieve Purchase Order remains a business capability whether Oracle EBS, SAP S/4HANA, REST, Camel, BPMN, or another mechanism implements it.

Release status. Specification boundary established; not dependent on runtime qualification.

3. Business semantics are separated from implementations

Business Packs define portable capabilities. Application Packs declare compatible implementations. Environment-specific Bindings connect the canonical capability to the correct implementation and Connection. A capability can therefore remain stable while applications, Environments, and technologies change.

Release status. Business/Application Pack and Binding contracts statically qualified in r11; installed-system gate pending.

4. Any agent can use Autonomy

The built-in Agent is optional. Customer-built agents, external agent platforms, applications, and headless clients can use the same governed capability surface instead of receiving raw ERP or CRM credentials.

Release status. Headless boundary established; standalone Agent Run REST/A2A equivalence is in r12 qualification.

5. Implementations are resolved dynamically

Requesting Agents do not hard-code an application, tenant, Connection, or implementation. Environment Bindings resolve the compatible implementation and Connection at runtime, while portable Packs carry no production secrets.

Release status. Binding/Connection/Environment model statically qualified; installed-system no-restart journey pending.

6. Authority is evaluated for every execution

Autonomy evaluates who initiated the work, which Agent is acting, eligible capabilities, organizational scope, risk and limits, invocation mode, Environment, policy, and delegated authority. API authentication alone is not sufficient business authority.

Release status. Foundational authority/policy model established; full per-Agent-Run qualification is part of r12.

7. Policy and human decisions are part of execution

Policies, confirmations, approvals, separation of duties, and requests for more input participate in the same governed run. Low-risk work can continue while higher-risk work pauses at the decision point requiring an authorized person.

Release status. Policy and Task concepts established; Agent WAITING_APPROVAL/WAITING_INPUT resume is in r12 qualification.

8. Agent Runs survive real business delays

An Agent Run is designed as durable execution toward a goal, not merely an in-memory chat. First-release states include CREATED, RUNNING, WAITING_INPUT, WAITING_APPROVAL, PAUSED, COMPLETED, FAILED, and CANCELLED, with resume from the same run context.

Release status. Qualification in progress; durable Agent Run persistence and resume are major r12 primitives and are not yet claimed shipped.

9. Business outcomes are verified and evidenced

Autonomy is designed to retain intent, initiator and Agent identity, authority and policy decisions, approvals, selected Binding, implementation, Connection, execution result, verification/readback, events, and recovery/replay information. Evidence is structured; private model chain-of-thought is not exposed.

Release status. Execution/Evidence direction established; comprehensive Agent Run evidence and installed-system verification remain r12 gates.

10. Packs create a continuously expanding business platform

Business Packs define what the business can do. Application Packs define how target applications implement compatible capabilities. Agent Packs define reusable autonomous business workers that depend on capabilities rather than a particular ERP or CRM. This lets the platform remain stable while coverage grows.

Release status. Business/Application packaging statically qualified; first-class Agent Pack runtime and complete r12 no-restart qualification are in progress.

Agent-builder model vs. Autonomy capability-execution model

This is not an argument against agent builders. Autonomy addresses the complementary problem: a stable governed business execution layer underneath whichever agents an organization chooses to use.

Dimension	Agent-builder model	Autonomy model
Primary abstraction	Agent graph / tool chain	Canonical business capability
Business meaning	Often embedded in agent/tool	Separated from implementation / ABCS-aligned
Backend access	Agent-specific tools/credentials	Governed capability + Binding + Connection
Governance	Often wrapped around agent	Identity, authority, policy, approval, verification, evidence
Agent choice	Builder-centered runtime	Built-in Agent optional; external agents supported
Growth model	Add agent flows	Add Business, Application, Agent Packs

Purchase Order Exception: one governed Agent Run

- **1.** Business intent: investigate the exception on purchase order 12345 and resolve it if authorized.
- **2.** Capability discovery: Retrieve Purchase Order, Retrieve Receipt, Retrieve Supplier, Update Purchase Order.
- **3.** User and Agent authority: record both identities and evaluate eligibility, scope, and delegated limits.
- **4.** Policy evaluation: investigation is allowed; a corrective update returns `require_approval`.
- **5.** Runtime Binding resolution: resolve the compatible implementation, Connection, application, and Environment.
- **6.** Approval pause: the Agent Run enters `WAITING_APPROVAL` and creates an Autonomy Task.
- **7.** Same-run resumption: approval resumes the same durable Agent Run rather than restarting the conversation.
- **8.** Governed execution: execute through the selected Action or Process implementation and application Binding.
- **9.** Verification: read back or otherwise verify the resulting business state.
- **10.** Evidence: retain intent, identities, authority, policy, approval, Binding, implementation, Connection, execution, verification, events, and recovery/replay information.

Three Pack types, one stable platform

Business Packs - portable canonical business capabilities and supporting content. They answer: What can the business do?

Application Packs - compatible implementations for target applications and Environments. They answer: How does this application implement the capability?

Agent Packs - reusable business-role autonomous workers depending on governed capabilities rather than a specific backend. They answer: Which autonomous worker reasons toward the goal?

The Pack model lets the governed platform change slowly while business coverage grows through capabilities, application implementations, and departmental Agents.

Initial departmental Agent examples

- Procurement Agent - purchase-order exception

- Finance Agent - invoice-hold investigation
- Sales Agent - customer/revenue situation
- Customer Service Agent - service escalation
- HR Agent - onboarding exception

Closing vision

Agents can improve. Models can change. Applications can be replaced. Integrations can be modernized. Business capabilities remain the stable contract between intent and action, while Autonomy governs who may act, how implementation is selected, which policies apply, whether a person must decide, what happened, and what evidence remains.

Autonomize Your Business.

Contact: contact@sirvisetti.com | sirvisetti.com