WorkingAgents

Technical Brief v1.2

The Execution Control Layer for AI agents. Enforced inside enterprise infrastructure.

Audience: VP Platform, Security Architect, CISO, Staff and Principal Engineers, Infrastructure teams, Technical partners.
Classification: Public material. Implementation detail is withheld and released only under NDA backed technical review.
Document: WorkingAgents Technical Brief v1.2
Contact: liem@workingagents.ai · workingagents.ai

1. At a glance

WorkingAgents is the Execution Control Layer between AI agents and enterprise systems. Every agent action is evaluated against policy before execution. Enforcement occurs at the point of action. Every evaluated action produces a decision record. The layer runs inside customer infrastructure. No data leaves the environment unless explicitly permitted by policy.

It is infrastructure. It is not an AI application. It is not observability. It is not compliance software.

Core properties

- Every action is evaluated against policy before execution. Unevaluated actions do not execute.
- The layer runs inside your infrastructure. Agent traffic does not cross the boundary.
- Chain of custody is maintained from request to outcome.
- Every evaluated action produces a structured decision record.
- User scoped and service scoped identity are preserved end to end. Privilege elevation is not possible within the execution path.

2. Deployment boundary

The Execution Control Layer is installed as an inline control plane between agents and enterprise systems, inside the customer environment. Agents, tools, connectors, and target enterprise systems remain on the customer side of the boundary. No request payload, tool argument, retrieved record, or resulting artifact transits to an external service. Egress is rejected unless explicitly permitted by policy.

Remains inside customer infrastructure

- Agent traffic and tool invocation payloads.
- Source systems, datasets, and retrieved records.
- Identity assertions and session context.
- Decision records and associated evidence.

Leaves only when explicitly permitted by policy

- Aggregate operational telemetry, under customer controlled configuration.
- Policy bundles authored by the customer or shared with an approved partner.

3. Request lifecycle

Every agent action passes through a deterministic sequence. No stage is skippable and no execution path bypasses evaluation.

Stage	Behaviour
`request`	Agent submits an intended action with identity, target system, and arguments.
`evaluate`	Action is evaluated against the applicable policy set before any execution occurs.
`redact`	Fields that exceed the caller's data boundary are removed or transformed before routing.
`route`	Permitted action is routed to the target system under scoped credentials only.
`execute`	Target system performs the action. Enforcement is applied at the point of action.
`record`	Outcome and decision metadata are written to the decision record. Chain of custody is sealed.

4. Control points

Enforcement is applied at three mandatory checkpoints. Before execution prevents authorization failures. During execution prevents constraint violations. After execution preserves audit integrity.

Before execution

- Identity and scope are resolved for the calling agent and the upstream user.
- The requested action, target, and arguments are evaluated against policy.
- Fields that exceed boundary are redacted or transformed at the argument level.
- The decision is produced deterministically: allow, deny, or allow-with-constraints.

During execution

- Execution proceeds only under the scoped credentials the decision permits.
- Runtime constraints are enforced: field projection, row limits, rate limits.
- A constraint violation halts the action and triggers the escalation path.

After execution

- The outcome is bound to the originating decision by identifier.
- The decision record is finalised and sealed. Chain of custody is preserved.
- Downstream effects inherit the same identity and policy context.

5. Identity and access model

A two layer identity model is preserved across every evaluated action. Privilege elevation is not possible within the execution path.

User scoped access

Actions taken on behalf of a named user are evaluated against that user's entitlements in the target system. The user's identity is carried through and is not collapsed into a shared service principal.

Service scoped access

Service level actions, such as scheduled maintenance or infrastructure telemetry, are evaluated against narrowly scoped service identities with their own policy surface. Service identities cannot impersonate users.

6. Policy evaluation

Policies are referenced by stable identifiers so that every decision is traceable to the exact rule that produced it. References take the shape of a dotted namespace path with a version suffix.

```
policy.governance.2.1
policy.data.pii.redaction.1.4
policy.tool.finance.payments.execute.3.0
```

Inputs that determine the evaluation outcome

- Caller identity: user, agent, and service context.
- Requested action and target system.
- Action arguments, with field level inspection where policy requires it.
- Environment signals such as deployment stage and sensitivity class.

Inputs determine the evaluation outcome, and the outcome determines execution. Rule engine internals, authoring surface, and evaluation ordering are withheld and released only under NDA backed technical

review.

7. Decision records

Every evaluated action produces a structured decision record. The decision record is the authoritative source of truth for why an action was allowed or blocked. A minimal record carries the following fields.

Field	Purpose
`id`	Stable identifier for the decision. Binds the outcome and any downstream effects.
`action`	The requested action, including target system and operation.
`agent`	The calling agent identity and its upstream user or service context.
`policy`	Reference to the policy that produced the decision, for example policy.governance.2.1.
`decision`	Allow, deny, or allow-with-constraints.
`reason`	Rationale aligned to the evaluated rule.

Additional fields, storage representation, retention behaviour, and export paths are released only under technical review.

8. Data handling and privacy boundary

No data leaves the environment unless explicitly permitted by policy. Denial is the default. Egress is evaluated as a policy controlled action on the same path as every evaluated action.

Does not leave the environment

- Prompts, tool arguments, and retrieved records.
- Source system contents and derived artifacts.
- Identity assertions, session tokens, and credentials.
- Decision records and their evidentiary payloads.

Permitted egress is scoped, not implicit

- Egress is evaluated as a policy controlled action. No implicit path exists.
- Field level redaction is enforced at the boundary where policy requires it.
- Exports require a named destination, an owner, and a matching policy reference.

9. What is not exposed in this brief

The following categories are withheld from public material. Absence of detail is intentional and exists to avoid expanding the attack surface. These details are released only under NDA backed technical review.

- Internal component topology, service boundaries, and queue behaviour.
- Storage format, indexing strategy, and retention mechanics for decision records.
- Policy engine internals, rule authoring surface, and evaluation ordering.
- Specific third party dependencies, vendors, and version pinning.
- Key management, secret distribution, and trust roots inside the deployment.
- Performance characteristics under adversarial load.

10. Technical review checklist

Reviewers should arrive with the following questions. These surface the assumptions that govern a production deployment.

- Where does the control plane run in our environment, and who operates it?
- Which identity provider is authoritative for user and service identities in evaluation?
- How are policy bundles authored, reviewed, versioned, and promoted to production?
- What is the failure mode when a policy evaluation cannot be completed?
- What is the chain of custody guarantee between a decision record and the executed outcome?
- Which fields in the decision record are tamper evident, and how?
- What must a tool or connector implement to be admitted behind the layer?
- How is egress modelled as a policy controlled action, and who can author those policies?
- What operational telemetry, if any, crosses the boundary, and under whose control?
- What is the upgrade path, and how are policy semantics preserved across versions?

11. Call to action

If your environment is running or planning to run autonomous agents, the next step is a scoped technical review of your environment.

- Schedule a security review focused on the deployment boundary and the decision record.
- Request an architecture review covering control points and the request lifecycle.
- Talk to the team about a scoped pilot inside your infrastructure.

Contact: liem@workingagents.ai · workingagents.ai

Document reference: WorkingAgents Technical Brief v1.2