

Before your agent can act, define its authority

An authority register and boundary-test plan for agent workflows that can create business consequences.

This conceptual working paper proposes an authority model for a bounded agent workflow. It connects the actor, delegated purpose, resource, action, conditions and expiry of a request to an enforcement point and reviewable evidence. It includes a sample authority register, approval design questions and a boundary test plan. It is an educational design framework rather than an empirical security assessment or a claim that any implementation is secure.

FOR Security architects, Engineering leaders, Platform teams, AI governance owners

Softlander | 2026-09-12 | v0.1

Conceptual working paper, version 0.1. The proposed register and tests have not been validated as a complete security method. Actual enforcement depends on implementation, identity systems, integrations and operational review.

Authority deserves its own design artifact

An assistant that recommends an action and an agent that executes it create different decisions for a business. The move to execution introduces questions about identity, delegation, approval and recovery that deserve to be written down explicitly.

This working paper proposes an authority register for that conversation. Its thesis is practical: **for every consequential action, a reviewer should be able to identify who authorized it, what was allowed and where that permission is enforced.**

The register is a design tool. It is not evidence that a deployed system enforces the intended boundary, and the framework has not been validated as a complete security method.

Begin with the resource and action

List the resources the workflow can reach and the actions available against them. Avoid stopping at tool names. An integration called "customer service" may expose several distinct actions with different consequences; the register should distinguish each one.

Two established sources inform the discussion. NIST SP 800-207 describes zero trust in terms of protecting resources without granting implicit trust solely because of network location or ownership. OWASP identifies excessive functionality, permissions and autonomy as contributors to excessive agency. The register below is a proposed application of those ideas, not a framework supplied or endorsed by either source. [NIST SP 800-207](#), [OWASP: Excessive Agency](#)

For each action, record the actor, delegating principal if applicable, intended purpose, resource scope, conditions and expiry. Add an enforcement point and an owner responsible for confirming how it works.

A proposed authority register

The following is a fictional support workflow. The entries describe intended policy, not an implemented system.

Action	Intended scope	Approval condition	Enforcement and evidence to review
Read a subscription	Authenticated customer's permitted account	Valid user context at access time	Resource authorization decision and scoped query trace
Draft a reply	Current support request	Allowed within the assigned task	Draft state and record access history

Action	Intended scope	Approval condition	Enforcement and evidence to review
Send a reply	Verified recipient for that request	Support specialist approves exact recipient and content	Approval record linked to send transaction
Change a subscription	Excluded from this workflow	No approval route offered here	Tool inventory and resource permission denial
Export account records	Excluded from this workflow	No approval route offered here	Data access scope and attempted export evidence

The register should be checked against the actual credentials and integrations. A missing button in the interface does not answer whether an alternate tool path or broad service credential can perform the action.

Specify what delegation does and does not carry

When an agent acts for a person, write down which rights the workflow may use and for how long. When one agent delegates to another, define whether the receiving agent gets a narrower capability, uses its own permissions or needs a new authorization decision.

The proposed default for review is explicit, bounded delegation. Do not assume that a downstream agent should inherit all of an upstream actor's access. Record how the system handles an expired task, a revoked user, a changed role and work that is already in progress.

Those are requirements to evaluate with the identity and application owners. This paper does not prescribe a universal token format or claim that a named platform provides the required behavior.

Design approval around the pending action

Where approval is required, identify what the person needs to inspect. For a message, that may be the actual recipient and content. For a purchase, it may be the supplier, amount, currency and items. For a record change, it may be the before-and-after values.

The proposed approval design binds the decision to those parameters, identifies the approver and states how long approval remains valid. A material change requires another decision. The execution path checks the approval before the side effect occurs.

Also define the failure path. If the approver is unavailable or execution state is uncertain, the workflow needs a specific disposition. It might retain a draft, route to an owner or reconcile the transaction before continuing. The appropriate choice depends on the action and operating context.

Turn each boundary into a negative test

The authority register becomes useful evidence only after the implementation is examined and exercised. A proposed test plan should include attempts that ought to be denied, alongside permitted actions.

Boundary question	Proposed exercise	Evidence needed
Does access stay within the assigned account?	Request a synthetic record outside that scope	Request identity, resource decision and data returned
Does approval cover the actual action?	Change a material parameter after approval	Approval details, modified request and execution disposition
Does revocation affect pending work as specified?	Revoke authority before a queued action executes	Revocation time, execution time and policy decision
Does delegation narrow permissions as intended?	Ask a downstream agent for an action outside its grant	Delegation record and downstream authorization outcome
Can recovery repeat a side effect?	Simulate an uncertain response after a completed transaction	Correlation records, transaction history and retry disposition

Use authorized nonproduction access and synthetic data for the initial exercise. Agree expected outcomes before running it. Record unknowns when required evidence is unavailable, and avoid treating the agent's narrative as a substitute for the resource's execution record.

Close with an operating decision

The review should leave a maintained action register, evidence references, open findings and owners. The organization can then decide which actions to enable, narrow, require approval for or keep outside the workflow.

Revisit the register when tools, permissions, delegation paths or operating conditions change. The first practical step is to choose one action that creates a real business consequence and trace its authority from request to execution. If the team cannot identify the enforcement point, it has a concrete design question to resolve before relying on that boundary.