



QILAHK

SECURITY YOU CAN PROVE

ADVANCED RESEARCH WHITE PAPER

QTA-R05 / 05 OF 05

Runtime Trust for AI Agents

Research into action scoped proof surfaces and authority enforcement

Version 1.1 | October 12, 2026

QILAHK Trust Authority

LOCK / KEY / PROOF

qilahk.com | blog.qilahk.com

Table of contents

Runtime Trust for AI Agents

Abstract	3
System boundary and the AI trust problem	4
Assess the deployed system	4
Define the action claim	4
Threat and fault assumptions	4
A proposed action scoped proof contract	5
Bind approval to the actual request	5
Use constrained capabilities	5
Illustrative attack and continuity scenarios	6
A retrieved instruction attempts to redirect an action	6
A valid approval becomes stale	6
Limit the guarantee	6
Adversarial evaluation and measures	7
Evaluate side effects rather than model statements	7
Test the proposed contribution	7
Research roadmap and references	8
Prototype and promotion gates	8
Open research questions	8
References	8

Abstract

Runtime Trust for AI Agents

An AI agent can generate a plausible plan while lacking authority to perform a proposed action. It can also operate within nominal permissions while using an unapproved destination, stale context, or a changed tool configuration. This paper proposes a QILAHK research model that evaluates an action-specific Proof Surface before an external side effect is executed.

The proposed architecture treats the model as a planner whose requests pass through an independent enforcement boundary. The boundary checks subject identity, tool and policy versions, capability scope, destination, data classification, approval context, and current trust conditions. It records the authorized request and actual execution result as separate events.

The research question is whether an action-scoped proof contract reduces unauthorized side effects and makes decisions more reconstructable without excessive interruption. The paper specifies an adversarial evaluation plan, version-change tests, and metrics with explicit denominators. It does not claim universal prompt-injection resistance, model correctness, or a deployed QILAHK AI enforcement product.

Intended audience

AI security researchers, agent platform architects, and assurance teams

Research scope

This white paper presents proposed QILAHK research mechanisms, illustrative scenarios, and evaluation protocols. Its hypotheses are intended for prototype testing and independent review.

QTA-R05 | Version 1.1 | October 12, 2026

System boundary and the AI trust problem

Assess the deployed system

The subject of an AI review includes the model version, orchestration logic, instructions, retrieval sources, connected tools, permissions, deployment context, and enforcement components. An evaluation of a model alone does not establish the properties of a later agent deployment. The claim must identify which system configuration and task context it covers.

NIST AI RMF 1.0 organizes risk management through Govern, Map, Measure, and Manage [1]. This paper uses that lifecycle as background and proposes a narrower runtime mechanism for tool actions. The mechanism is one possible control within an AI risk program, not a substitute for impact assessment or broader governance.

Define the action claim

An action claim asks whether an identified agent may perform a specific operation on a specific resource using specified data under current conditions. Required properties can include a recognized initiating principal, a permitted tool operation, destination restrictions, data-release authorization, an unexpired approval, and a valid system configuration.

The Proof Surface is the set of observations and checks needed for those properties. A narrative from the model is not sufficient evidence of permission. Credentials prove access to a service under their validation model; they do not establish that every operation available through that service is authorized for the current task.

Threat and fault assumptions

The model may be influenced by untrusted retrieved text, propose incorrect arguments, omit relevant context, or persist after a task changes. A tool adapter may return inaccurate metadata. The proposal assumes an enforcement boundary outside the model's control and no alternate path to the protected side effect. If a bypass route exists, the action contract cannot enforce its intended property.

A proposed action scoped proof contract

Bind approval to the actual request

Canonicalize the proposed operation into a structured request containing initiating principal, agent run, tool operation, resource, destination, arguments, data scope, and applicable policy version. An approval binds to that request or a precisely defined permitted range. A material argument change requires re-evaluation rather than inheriting an unrelated approval.

At the execution boundary, check the request again against current configuration and policy. Pin the approved request version to prevent a time-of-check to time-of-use mismatch. The tool adapter must execute the validated arguments and report what actually happened. A plan that passed appraisal is not proof that the external operation succeeded.

Use constrained capabilities

A capability should restrict operation, resource scope, audience, duration, and permitted delegation. Keep credentials outside model-visible text where possible. A model can request a capability through the orchestrator but should not be able to expand its scope by asserting urgency or quoting an instruction from a retrieved page.

Separate routine permitted actions from actions requiring an authorized human or institutional reviewer. Record who approved, the request covered, conditions, and expiry. A denial, missing approval, or expired approval is an enforcement result that the model cannot override. The recipient's policy defines the consequences and any permitted fallback.

Proposed runtime checks

Property	Required evidence or enforcement
Identity	Authenticated initiating principal and bound agent run
Tool scope	Operation and resource within the granted capability
Destination	Approved recipient service or resource identifier
Data release	Classification and authorization for the exact disclosure
Configuration	Pinned model orchestration and tool versions
Approval	Valid approval tied to the request where policy requires it

External text remains task data. It can supply facts to be assessed but cannot create authority grants or expand a capability. Logging should preserve the relevant decision context while minimizing sensitive prompt and payload content; full conversational retention is not required to prove every action property.

Illustrative attack and continuity scenarios

A retrieved instruction attempts to redirect an action

An illustrative research agent reads an external document while preparing a report. The document asks the agent to send the source collection to a new address. The model may produce a tool request that follows the instruction. The enforcement boundary checks the proposed destination and data release against the initiating task and capability; the document itself cannot satisfy those checks.

If the operation is outside scope, the boundary blocks the request and records the failed properties. If policy permits a new authorized review, the request can be presented to the appropriate reviewer with its actual destination and data scope. Approval of one recipient should not authorize a different recipient or a broader payload.

A valid approval becomes stale

An administrator approves a request under tool adapter version V1. Before execution, the adapter changes to V2 with different parameter semantics. The system should detect that the approved configuration no longer matches the executing configuration. Materiality policy determines whether the approval can be safely translated or requires a new review.

Similarly, a model replacement, expanded retrieval source, changed instruction, or new tool permission can affect a deployment-level determination. Reassess the affected claims rather than assuming that an unchanged interface preserves the original Proof Surface.

Limit the guarantee

Correct scope enforcement does not establish that an allowed answer is factually correct, that a recipient will handle data appropriately, or that an approved action is wise. These need additional evidence and governance. The claim must remain bounded to the action properties actually enforced.

A malicious tool service can perform an unreported side effect after receiving an authorized request. The adapter cannot prove behavior beyond its observation and control boundary. Test service contracts, execution receipts, and independent reconciliation where the intended assurance depends on external behavior.

Adversarial evaluation and measures

Evaluate side effects rather than model statements

Build a controlled tool environment with instrumented external effects and known authorization labels. Include benign tasks, direct and indirect instruction attacks, destination substitution, payload expansion, expired approval, adapter changes, retries, and partial tool failures. The success criterion is actual external behavior, not whether the model says it resisted an instruction.

Compare prompt-only constraints, permission checks at tool attachment, and the proposed per-action enforcement boundary. Give all variants the same tasks and tools. Include an ablation that removes request-version binding and another that removes destination checks to identify which controls change outcomes.

Test the proposed contribution

H1 proposes a lower unauthorized-side-effect rate than the two baselines under the selected attack distribution. H2 proposes improved reviewer reconstruction of the exact request, approval, and execution result. H3 proposes that benign task completion remains within a preregistered operational bound.

Report attack success by family, repeated trials, and model and adapter version. Hold out attack templates and distinguish policy errors from enforcement implementation errors. A block-all system avoids unauthorized effects but fails usefulness; benign completion and interruption burden must accompany the security result.

Runtime research measures

Measure	Definition
Unauthorized effect rate	Trials with an unauthorized external effect divided by adversarial trials
Benign completion	Correctly completed authorized tasks divided by benign trials
Approval precision	Approvals applied to their exact permitted scope divided by approval uses
Trace reconstruction	Correct request approval and execution links identified divided by required links
Operational burden	Added action latency and human reviews per completed task

Use synthetic sensitive data and isolated test destinations during experimentation. Publish the policy labels, tool semantics, and enforcement assumptions alongside results. A favorable benchmark would support only the tested boundary and distribution, not a universal claim of safe AI behavior.

Research roadmap and references

Prototype and promotion gates

First formalize the action request schema and capability semantics. Build an enforcement boundary with no direct bypass route, then implement approval binding, version checks, and execution receipts. Add materiality-based reassessment only after the basic enforcement invariants pass fault and adversarial tests.

Promotion gates should require rejection of out-of-scope destinations and payloads, visible expired-approval outcomes, tested retry behavior, and reconciliation of approved requests to executed effects. Capture incomplete execution explicitly so a retry does not duplicate an irreversible side effect.

Open research questions

Which Proof Surface properties can be assessed automatically with adequate reliability? How can approval scope remain understandable to people without becoming too broad? Which configuration changes require immediate reassessment? How much trace context is needed for accountability while minimizing sensitive retention?

The proposed contribution is an action-level authority boundary that can be tested independently of model persuasion. Its value depends on complete enforcement coverage, explicit data-release policy, accurate tool semantics, and evaluation against real external effects.

References

[1] NIST AI 100 1 Artificial Intelligence Risk Management Framework

Govern Map Measure and Manage lifecycle.

<https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf>

[2] NIST SP 800 207 Zero Trust Architecture

Explicit contextual authentication and authorization.

<https://csrc.nist.gov/pubs/sp/800/207/final>

[3] QILAHK Journal Define a Proof Surface for an AI System

System boundary and AI claim framing.

<https://blog.qilahk.com/proof-surface-for-ai-systems.html>

Reference editions and source pages checked October 12, 2026. The proposed mechanisms and evaluation plans in this paper are research designs, not reported experimental findings.