iProov

Introducing Human Approval and Presence Specification: HAPS

28 August 2026 ·

Our previous papers on the secure management of AI agents outlined a set of foundational principles and when to apply them. As a near-term response to the problem that AI agents can present technically valid but agent-accessible evidence, such as OAuth Access tokens, that does not establish human approval of the specific action proposed, we are now sharing iProov's Human Approval and Presence Specification (HAPS). HAPS is an experimental procedural specification that develops how relying parties can obtain portable, verifiable evidence of human approval for downstream high-risk actions by requiring agent-resistant factors where demanded by policy.

HAPS (Human Approval and Presence Specification) proposes a common structure for binding an approval ceremony to an exact machine-readable action. It describes proposed requirements for constructing and hashing the action intent and signing view, how a relying party states its proof of human presence and optional identity requirements, and how a presence provider packages the resulting evidence for verification before execution. The relying party remains responsible for deciding which actions require approval, whether the approving human is authorised, and how the action is held and executed.

Evidential Problem Faced by Relying Parties

An AI agent may reach a relying party through an authenticated session, possessing valid credentials and apparently acting within a legitimate delegation. Whilst these facts can establish the agent's identity, the route through which it arrived, and the standing authority it possesses, do not necessarily ensure that the principal intended the particular action now being requested.

This creates the possibility of a confused deputy situation in which the agent exercises legitimate authority contrary to the principal's intentions as a result of miscomprehension, prompt injection, or the misuse of delegated credentials. From the relying party's perspective, the intended and unintended requests may appear identical given that both can arrive through an authenticated channel and satisfy the required access conditions.

Existing protocols provide important parts for the resolution of this challenge:

- OAuth for access delimitation;
- AuthZEN for externalizing authorisation decisions;
- OpenID4VP for credential presentation and transaction-data binding; and

- MCP, through which the HAPS profile can request human interaction while the relying party keeps the action pending.

These mechanisms provide complementary capabilities for authentication, authorisation, credential presentation and transaction-specific binding. HAPS proposes the beginnings of a common structure for combining action data, a derived Signing View and provider assertions of human presence and approval within relying-party-controlled agent workflows. Its intended contribution is to make those relationships explicit across implementations, while retaining the relying party's responsibility for policy and execution.

About Human Approval and Presence Specification

HAPS is an implementation-neutral procedural specification. It proposes suitable data objects, assurance requirements, action-binding properties, a relying-party verification procedure, and provides a common technical description against which independent implementations can be designed and evaluated.

The specification which can be found on GitHub [insert link] defines three primary protocol objects, together with a derived signing view:

- the action intent which provides a canonical, machine-readable description of the proposed action;
- the optional HAPS Challenge, recommended for high-risk actions, which allows the relying party to state the proof of human presence and optional identity requirements that must be satisfied;
- a signing view which specifies the authoritative representation presented to the human; and
- the HAPS Consent Credential for how the resulting evidence can be packaged up and carried back to the relying party in a portable and verifiable form.

To support experimentation with HAPS and invite contributions, we are publishing a partial Rust reference implementation of HAPS's canonicalisation and hashing core, together with a test kit for selected canonicalisation, hashing and malformed-input cases. This is not production ready.

We will be publishing a video demonstration of iProov's internal implementation, in which biometric liveness supplies the human-presence evidence for our specific use cases of managing enterprise Agents, and invite others to experiment with HAPS in a range of directions.

From Procedural Specification to Verifiable Approval

HAPS facilitates the resolution of confused deputy interactions for RPs by defining a holistic evidential relationship across the human, the approval ceremony, the pending action and the relying party.

First, the relying party decides that an action needs human approval. It records exactly what the agent intends to do, holds the action, and issues a HAPS challenge stating what evidence is required.

Second, the presence provider shows the action to the human. HAPS binds the provider's approval assertion to the action and Signing View hashes, enabling the relying party to detect and reject mismatches.

Third, the human explicitly approves the action and completes the required presence check and, where necessary, proves their identity. The presence provider then returns a signed HAPS consent credential.

Finally, the relying party checks that the credential is genuine, sufficiently strong, current, and unused. Only then is the approved action released for execution.

Proportionate and Agent-Resistant Proof

As outlined in our second paper (add name and link here) , the existence of HAPS does not imply that every agentic action should require human-in-the-loop approval. Not only would this forfeit the productivity that autonomy provides, but it would also create approval fatigue and reduce human control to a ceremonial interaction with due consideration of what is being confirmed.

Conditions and Limitations

HAPS depends on a relying party holding the authoritative action intent behind a control which the agent cannot bypass. The relying party must select suitable evidence, verify the credential independently, apply its own authorisation policy and consume the relevant identifiers in accordance with its single-use policy. Where identity is required, it must require and verify an identity binding against its own account records. HAPS cannot establish that the approving human is authorised, compensate for a bypass route, or guarantee that the human read or understood the displayed content. A valid signature does not independently establish that the provider faithfully displayed the action or captured explicit human approval.

Access The GIT

https://github.com/iProov/HAPS/tree/main

HAPS is not a hosted backend service, supported infrastructure component or SDK. The specification and reference implementation are published under Apache-2.0, and conformance is defined by the specification, conformance requirements and test vectors rather than by iProov's example code or biometric technology. The accompanying materials are illustrative. This is not production-ready.