Raziel

MCOMP-04
29th STePS

Prof. Jungpil Hahn (Advisor) **Gita Alekhya Paul** NUS FinTech Lab

NUS National University of Singapore | FinTech Lab School of Computing

Raziel research project

Verifiable data access for AI agents

Useful AI agents need data without blanket access. Raziel gives each agent verifiable permission for an exact task.

Our research question: Can another service verify permission without receiving the private relationships behind it?

A wrong grant can carry a correct signature

01 / The problem

Alice, an AI agent, asks Bob's service for Doc 42. The rule requires an assigned reviewer. Alice has no assignment.

Signed assertion

The issuer uses its genuine key to sign a grant the rule forbids.



Bob trusts the issuer's decision and releases Doc 42.

Raziel checks the computation

Bob requires proof that the rule and committed facts permit the read.



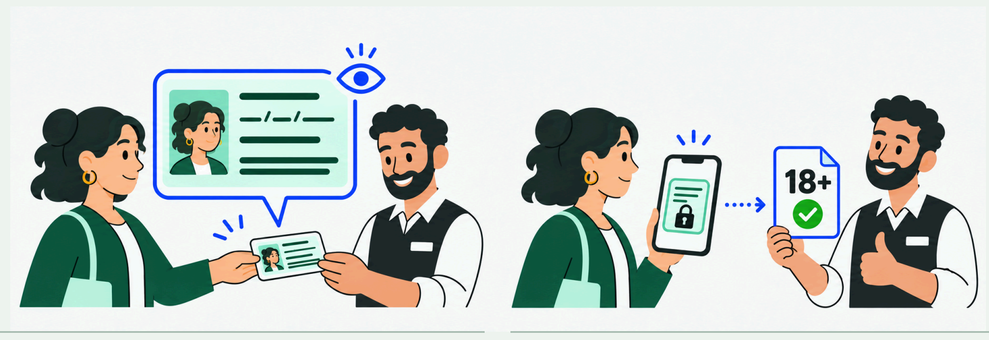
Bob blocks the read without a valid proof.

02 / Zero-knowledge proofs

What does the bar learn?

The bar only needs to know whether a guest is 18 or over. A zero-knowledge proof lets it check that claim from a trusted credential.

In Raziel: The verifier learns that an approved rule allows the request. The underlying relationship evidence stays with the data owner.



Without a ZKP

The guest shows a full ID. The bar sees their name, photo, full birth date and other details.

With a ZKP

The bar checks the 18+ claim. The guest keeps those other details private.

How Raziel works

03 / Architecture and proof

The data owner approves policy and generates the proof. A separate verifier checks it. The gateway controls data release.

1. Match the policy

The proof binds the approved rule and exact agent, task, purpose, action and data scope.

2. Check private evidence

Membership checks bind the hidden relationship facts to an approved snapshot.

3. Prove permission

The policy interpreter returns ALLOW. The verifier checks this result independently.

One grant, repeated protected reads

Request Agent asks for scoped access

Approve Operator selects policy and data

Sign Agent accepts the exact proposal

Prove Owner creates proof and grant

Read Gateway admits each request

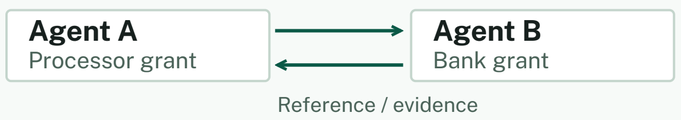
Private evidence stays with the owner

Agents can reuse a verified grant within scope and validity. **The gateway checks the agent's signature, scope, expiry, replay and dataset on every read.**

04 / Working example

Two agents reconcile separate ledgers

One agent reads processor records. The other uses its own grant to read bank settlements. In the synthetic demo, a transaction reference linked the two reads and the agents found a matching settlement. Audit receipts linked both reads to their grants.



05 / Contribution and next research

Research contribution

We tested whether rules based on relationships could be compiled and proved with private evidence. Agents reuse the independently verified grant for new signed requests.

Hypha is our broader vision: join this authorization layer with data discovery, task relevance and cumulative disclosure controls.

Measured prototype evidence

21/21 Fixture decisions matched live SpiceDB

8/8 Tested proof and request mutations rejected

447 ms One retained proof checked with issuer and prover stopped

~16 min One separate local proof generated on eight CPUs

Proving and verification timings come from separate recorded runs.

Design partners and research sponsors

Test one restricted workflow with us

Bring an agent, a data action and the rule that governs it. We are seeking partners to define a sandbox evaluation and sponsors to support proof engineering and compute.

raziel.gpaul.cc

STePS event sponsors / Platinum

DSTA Defence Science & Technology Agency | Garena | sea | Tealseed

STePS event sponsors / Gold

SGInnovate | doing IT right SIMSYS

Research sources: Raziel final report, Chapters 4, 6 and 7. Token checks: RFC 9068 §4, RFC 9449 §7.1.

ChatGPT Codex assisted with copy, diagrams, illustration and export.