STIDS 2026 | May 27-28, 2026 | Arlington, Virginia

Zenia

Explainable Zero Trust Decisions

with Knowledge Graphs and AI

An RDF-native, Graph RAG, and multi-agent reference architecture for explainable Zero Trust access control.

Lowell Vizenor

Director of Product Innovations, NIRA, Inc.

Co-authors: Wen Zhu, Jay Desai, John Price

Data-centric. Explainable. Interoperable.

Overview

01 Why Zero Trust access decisions are hard

02 Why ABAC fits, but is not enough

03 LEGACII and triple-level enforcement

04 Graph RAG, hybrid reasoning, and MCP orchestration

05 Scenario walkthrough and research trajectory

Why Zero Trust Access Decisions Are Hard

Access control is no longer a simple yes/no permission check.

01 Data crosses boundaries
Systems, organizations, and classification domains.

02 Context changes fast
Role, clearance, location, and mission need.

03 Policies conflict
Multiple authorities can govern the same data.

04 Attributes drift
Roles, labels, and mappings fall out of sync.

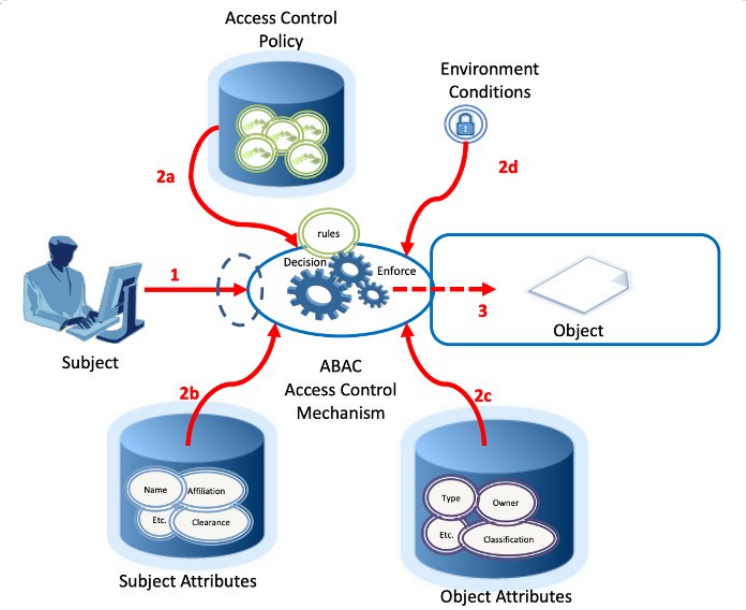
05 Denials need explanation
What rule blocked access, and why?

06 Mission work cannot wait
Delays create workarounds and erode trust.

Zero Trust decisions must be precise, contextual, and explainable.

Why ABAC Fits - But Is Not Enough

ABAC is the right decision model for Zero Trust, but it is hard to govern at enterprise scale.



Source: NIST SP 800-162, Guide to Attribute Based Access Control (ABAC) Definition and Considerations

Why it fits Zero Trust

Request-specific - evaluates each access attempt

Context-aware - uses mission and environment

Fine-grained - can vary by data element

Data-centric control - Policy follows the data, not the system boundary.

Why it is not enough

Rules multiply - policy sets become hard to govern

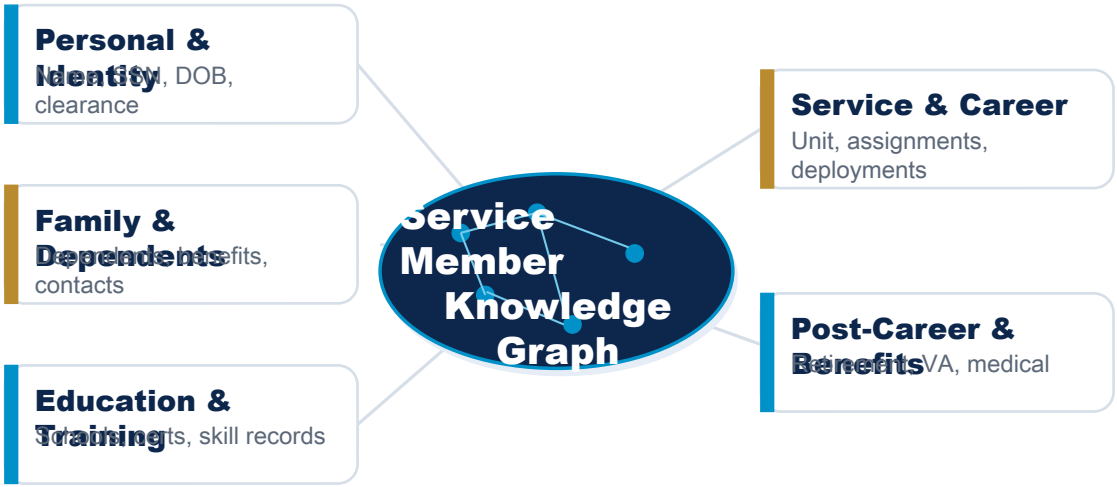
Terms diverge - organizations use different labels

Attributes drift - roles and mappings fall out of sync

Denials need evidence - users need to know why

LEGACII: One Record, Many Sensitivity Levels

A service-member knowledge graph brings lifecycle data together - but access cannot be granted at the whole-record level.



The access-control problem

One graph contains facts with different access rules.

Rank	unrestricted
Unit	need-to-know
Deployment location	classified
SSN	HR / FSO
Medical record	special handling

Zenia's Answer: Explainable Triple-Level Access Control

Zenia applies Zero Trust at the data layer — down to individual RDF triples — and makes each decision explainable.

From LEGACII: one service-member graph contains facts with different access rules.

1 RDF-Native ABAC

Policy bound to data

- **Predicate-level rules** — Govern RDF properties.
- **Ontology inference** — Derive implied permissions.
- **SPARQL enforcement** — Filter triples at query time.

2 Graph RAG

Evidence from the graph

- **Policy subgraphs** — Retrieve relevant rules.
- **Relationships preserved** — Keep context and provenance.
- **Conflicts surfaced** — Flag contradictory rules.

3 MCP Orchestration

Agents with clear roles

- **Identity binding** — Resolve requester attributes.
- **Policy evaluation** — Apply rules first.
- **Retrieval + explanation** — Assemble evidence.
- **Audit trail** — Record sources and steps.

Zenia evaluates access to statements in the graph, not just records in a system.

RDF-Native ABAC: Predicate-Level Enforcement

Policy binds directly to RDF predicates, so access can be evaluated before triples leave the graph.

01 Native RDF format
Policy enforcement works directly on the knowledge graph.

02 Statement-level granularity
Rules govern predicates within the same service-member record.

03 Ontology-backed inference
Clearance, role, and property hierarchies reduce rule enumeration.

04 SPARQL-time enforcement
Queries are filtered or masked before results are returned.

Example: statement-level access rule

```
# 1. The data triple
:SvcMbr_12345  sm:deploymentLocation  :Fort_Brag .

# 2. Bind the predicate to a rule
sm:deploymentLocation  abac:governedBy
policy:DepLocationRule .

# 3. The access rule itself
policy:DepLocationRule  a  abac:PropertyAccessRule ;
  abac:requiredClearance   abac:TS_SCI ;
  abac:requiredAttribute   abac:NeedToKnow_Ops ;
  abac:effect              abac:Permit .
```

Security Rule Design Alternatives

The design question: where should access-control rules attach in an RDF knowledge graph?

N-Quads

Security ID per statement

~

Good: precise statement labels

Issue: maintenance-heavy

Rule or attribute changes can force updates to statement IDs and security metadata across the data.

Subgraphs

Partition by sensitivity

✗

Good: simple graph boundaries

Issue: duplicated data

The same entity may be copied across graphs, fragmenting the record and complicating updates.

Predicate Rules

governedBy on predicate

✓

Good: policy visible in graph

Chosen: update policy, not data

Rules attach to predicates, remain readable in SPARQL, and can change without restructuring records.

Legend: ✓ best fit ~ workable but costly ✗ rejected for this architecture

Design decision **Start with property-based rules as the baseline, then add context-aware extensions only where needed.**

Graph RAG: Retrieval With Structure

RAG grounds AI answers in retrieved evidence. Graph RAG keeps relationships intact.

RAG

Retrieval-Augmented Generation

Find sources — retrieve relevant documents or passages.

Add context — supply evidence to the model.

Generate answer — respond using retrieved material.

Plain meaning: AI answers with evidence, not just memory.

Graph RAG

Retrieval from a knowledge graph

Retrieve entities — pull relevant people, data, and policies.

Preserve relationships — keep graph structure intact.

Track provenance — show where facts came from.

Support reasoning — use hierarchy and inference.

Plain meaning: AI gets structured evidence, not just text.

Why Zenia needs it

Access decisions depend on relationships

User → clearance — requester attributes drive access.

Predicate → rule — data elements cite their policy.

Rule → attribute — conditions are explicit.

Decision → evidence — explanations cite the chain.

Plain meaning: The explanation follows the evidence chain.

Rule-Based vs. Judgment-Based Decisions

Zenia automates clear cases and assembles evidence when policy is incomplete or conflicting.

Rule-Based Decisions

When policy is complete

1. **Facts are known** — requester, data, action, and context are available.
2. **Rules are clear** — applicable policy produces a definite outcome.
3. **Reasoning is deterministic** — ontology inference resolves hierarchy and inheritance.
4. **System enforces** — permit, deny, or mask is applied automatically.

Judgment-Based Decisions

When policy is incomplete or conflicting

1. **Facts are missing** — required context is absent or uncertain.
2. **Rules conflict** — multiple policies point in different directions.
3. **Hybrid reasoning** — Graph RAG retrieves structure; traditional RAG retrieves guidance and precedents.
4. **Human reviews** — Zenia explains the issue; it does not guess.

Evidence path

Graph RAG policy structure → **Traditional RAG** guidance + precedents → **Fusion agent** combined rationale → **Human review** ambiguous cases

MCP Orchestration: Agents With Clear Roles

Model Context Protocol (MCP) is a standard way for AI agents to call tools and data services through defined interfaces.

In Zenia: Graph RAG retrieves structured evidence. MCP coordinates the agents and tool calls around the decision workflow.

Graph RAG

What evidence is retrieved

- **Policy subgraphs** — rules relevant to the request.
- **Attributes + relationships** — who, what, why, and context.
- **Hierarchies + provenance** — why the evidence applies.

MCP

How agents coordinate work

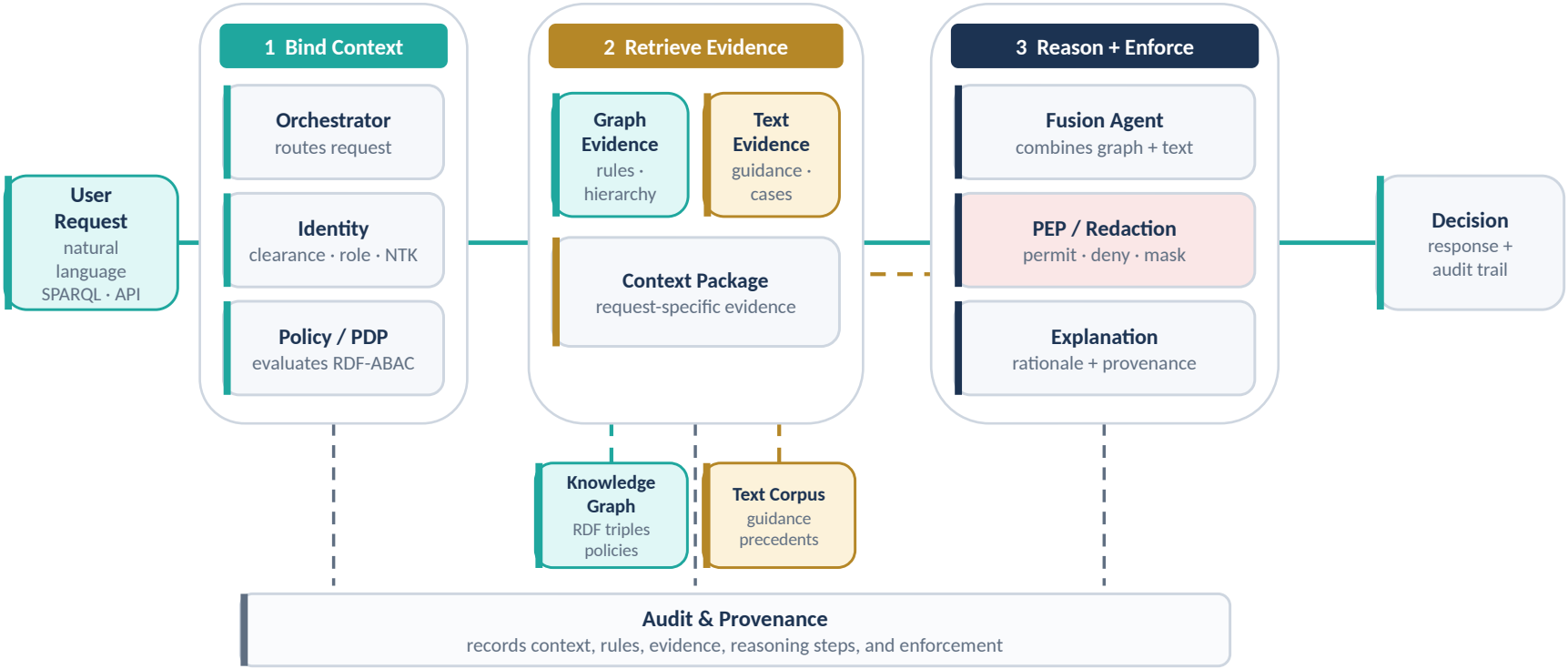
- **Standard interfaces** — tool and service calls are defined.
- **Controlled tool calls** — agents use approved capabilities.
- **Inspectable steps** — workflow events can be reviewed.

Decision workflow

MCP does not replace Graph RAG. It keeps agent-to-tool interactions modular, controlled, and auditable.

Zenia Reference Architecture

Identity, policy, retrieval, reasoning, enforcement, and audit in one explainable Zero Trust workflow.



Scenario: Deployment History

A policy-enforced SPARQL query returns authorized data and masks or filters restricted statements.

Request

"Show me the deployment history for Service Member 12345."

Requester

Role: Analyst

Clearance: Secret

Need-to-know: Confirmed

Decision

Approved with masking

Historical deployments returned; restricted fields filtered or generalized.

Generated SPARQL Query

```
# 1. Requester attributes injected by Identity Agent
VALUES (?user ?ntk) { (ex:analyst true) }

SELECT ?member ?dep ?date ?location
WHERE {
  ?member a leg:ServiceMember ;
          leg:id "SM-12345" ;
          leg:hasDeployment ?dep .
  ?dep    leg:location ?location ;
          leg:date ?date .

  # 2. Predicate rule binding applied
  leg:hasDeployment leg:governedBy leg:Rule_DEPLOY .
  leg:Rule_DEPLOY acl:requiresClearance
acl:SecretClearance .

  # 3. Clearance + need-to-know enforced inline
  FILTER EXISTS { ?user acl:hasClearance ?uc .
                  ?uc acl:higherThan*
acl:SecretClearance . }
  FILTER (?ntk = true)
}
```

1 Attributes injected 2 Rule binding applied 3 Policy enforced inline

Findings and Research Trajectory

Zenia tests how far AI can extend deterministic ABAC while keeping enforcement grounded in explicit policy.

Prototype findings

Decompose the workflow
Monolithic query + enforcement was brittle; staged execution performed better.

Guide graph traversal
Multi-hop paths often needed ontology-aware hints or query templates.

Constrain model behavior
Non-determinism, context confusion, and prompt limits make AI unsuitable as sole enforcement.

Still to test

Policy deconfliction
overlapping or contradictory rules

Denial review
evidence when a denial may be wrong

Approval acceleration
guidance, precedents, similar cases

Explanation validation
cited evidence vs. applied rules

Research trajectory

Near term
AI assembles graph + text evidence for human review after denial.

Far term
Evaluate whether AI can safely override deterministic denial in high-stakes cases.

Current position
Not there yet: AI assists, explains, and recommends; rules and human review remain necessary.

Zenia's goal: determine how far AI can responsibly extend deterministic ABAC — not replace rule-based enforcement prematurely.

Questions?

Zenia: Explainable Zero Trust Decisions with Knowledge Graphs and AI

Lowell Vizenor
Director of Product Innovations, NIRA, Inc.
lvizenor@nira-inc.com