CEDAR AUTHORIZATION PRIMER

Cedar

Policy and decision model

How a complete request becomes an authorization decision

An architecture primer based on Cedar documentation, the Cedar tutorial, and CNCF project status.

FIRST-PARTY SOURCES / INDEPENDENT LUMINITY SYNTHESIS



Cedar evaluates a complete authorization request

A visual primer on requests, policy scope, decision semantics, entity data, trusted context, and schema-backed validation.

AT A GLANCE

4
request components
principal, action, resource, context

2
policy effects
permit and forbid

3
policy results
true, false, or error

2
final outcomes
Allow or Deny

Decision mechanics

01

A complete PARC request, selected policies, and relevant entities produce one decision plus diagnostics.

Trust boundary

02

The application authenticates callers, supplies trusted facts, validates the model, and enforces Cedar's result.

IN THIS PRIMER

Key Points

01 **Request and inputs**
PARC, policies, entities, decision, and diagnostics

02 **Policy structure and scope**
Effects, constraints, conditions, and overpermission

03 **Decision semantics**
Forbid precedence, default deny, and errors

04 **Entities and relationships**
Groups, containers, attributes, and data completeness

05 **Trusted request context**
Authentication boundary, verification, and normalization

06 **Schema and reuse**
Validation, templates, change impact, and runtime separation

Cedar does not authenticate users or enforce application operations. Those responsibilities remain outside the authorizer.

© 2026 Luminity Digital, Inc. | Independent synthesis of Cedar first-party sources.

Cedar documentation and CNCF project page · verified Nov 13, 2026

Cedar policy and decision model

The authorizer evaluates one principal, action, resource, and context against policies and relevant entity data.

01

What enters the request?

Principal	ENTITY REFERENCE
Action	ENTITY REFERENCE
Resource	ENTITY REFERENCE
Context	RECORD
Authentication	OUTSIDE CEDAR

Complete PARC request → Cedar authorizer

02

How a decision is produced

Request · P A R C | Policies · permit + forbid | Entities · attributes + hierarchy

1. **Select** — Applicable policies
2. **Evaluate** — Scope and conditions
3. **Combine** — Forbid before permit
4. **Return** — Allow or Deny

POLICIES
Permit and forbid policies

ENTITIES
Only relevant attributes and ancestors

DIAGNOSTICS
Determining policies and evaluation errors

03

Four request boundaries

Context is a record — TYPE

Context carries request-specific values. Principal, action, and resource remain entity references.

Entity data is supplied — INPUT

The application provides the attributes and hierarchy required by the selected policies.

Diagnostics explain — WHY

The result includes policy identifiers for determining policies and evaluation errors.

The caller enforces — BOUNDARY

Cedar returns the authorization decision. The application decides how to act on it.

REQUEST BOUNDARY

- Authenticate before authorization
- Supply complete entity data
- Retain diagnostics
- Enforce in the application

Luminity Digital
INTELLIGENCE, ENGINEERED, DELIVERED.

© 2026 Luminity Digital, Inc. | Independent synthesis of Cedar first-party sources.

Cedar policy and decision model

Every policy declares an effect and complete scope. Conditions refine when that policy applies.

01

What a policy contains

- Effect — PERMIT OR FORBID
- Principal scope — REQUIRED
- Action scope — REQUIRED
- Resource scope — REQUIRED
- when and unless — OPTIONAL

Static policy → Template-linked policy

02

How scope expands

Principal · exact or group | Action · exact or set | Resource · exact or container

1 **Exact** — Named entities

2 **Grouped** — Role or container

3 **Conditional** — Attribute tests

4 **Broad** — Unconstrained scope

MATCH SET

Broader scope matches more requests

CONDITIONS

when must be true; unless must be false

BOUNDARY

Annotations do not affect evaluation

03

Four scope interpretations

Exact entity scope — NARROW

A policy can target one principal, action, or resource by identifier.

Group or container — HIERARCHY

Membership supports role-based and resource-hierarchy authorization.

Attribute conditions — ABAC

Conditions compare request and entity values after scope matches.

Unconstrained elements — REVIEW

An omitted constraint expands the request set that the policy can match.

SCOPE CHECK

- Effect and PAR scope are mandatory
- Conditions refine the match
- Broad scope needs review
- Annotations are metadata

© 2026 Luminity Digital, Inc. | Independent synthesis of Cedar first-party sources.

Basic Cedar Syntax and Cedar Policy Templates · verified Nov 13, 2026

Cedar policy and decision model

Cedar evaluates every policy, then combines the results through forbid precedence and default deny.

01

What each policy returns

true	POLICY SATISFIED
false	NOT SATISFIED
error	SKIPPED + REPORTED
Policy order	NO EFFECT
Final result	ALLOW OR DENY

Evaluate policies → Combine results

02

How the final result is chosen

Rule 1 · satisfied forbid | Rule 2 · satisfied permit | Rule 3 · otherwise

1 **Evaluate** — Every policy

2 **Forbid?** — Any true forbid

3 **Permit?** — Any true permit

4 **Decide** — Deny or Allow

FORBID
One satisfied forbid yields Deny

PERMIT
A permit matters only when no forbid matches

DEFAULT
No satisfied permit yields Deny

03

Four decision interpretations

Default deny — BASELINE

A request is denied when no permit policy evaluates to true.

Forbid precedence — OVERRIDE

Any satisfied forbid overrides all satisfied permit policies.

Order independence — STABLE

Reordering policies does not change the authorization outcome.

Errors remain visible — DIAGNOSTIC

Evaluation errors do not grant access and appear in diagnostics.

DECISION RULE

- Forbid overrides permit
- Permit is explicit
- Otherwise deny
- Errors never authorize
- Order does not matter

© 2026 Luminity Digital, Inc. | Independent synthesis of Cedar first-party sources.

Cedar policy and decision model

Entity data carries group membership, containment, and attributes used by policy scope and conditions.

01

What entity data supplies

- ✓ Principal attributes — DEPARTMENT, STATUS
- ✓ Principal hierarchy — ROLES AND GROUPS
- ✓ Resource attributes — OWNER, CLASSIFICATION
- ✓ Resource hierarchy — CONTAINERS
- ! Missing required data — EVALUATION RISK

Entity slice → Policy evaluation

02

How relationships are used

Principal · user to role | Resource · object to container | Attributes · values on entities



RBAC
principal in Role::reviewer

RELATIONSHIP
resource in Album::entries

ABAC
Compare principal and resource attributes

03

Four entity interpretations

Membership is modeled — GRAPH

Groups and containers appear as explicit entity relationships.

Attribute conditions — VALUES

Policies can compare typed values stored on principal and resource entities.

Ancestors matter — ANCESTRY

The request needs every relevant parent used by membership tests.

The caller selects data — BOUNDARY

Cedar evaluates the entity slice supplied for that request.

ENTITY BOUNDARY

- Model relationships explicitly
- Supply required ancestors
- Use typed attributes
- Treat missing data as material

Luminity Digital
INTELLIGENCE, ENGINEERED, DELIVERED.

© 2026 Luminity Digital, Inc. | Independent synthesis of Cedar first-party sources.

Cedar policy and decision model

Context carries request-specific facts. Persistent identity and resource facts remain on their entities.

01

Which facts belong where?

✓ User department	PRINCIPAL ENTITY
✓ Resource owner	RESOURCE ENTITY
✓ MFA used this session	CONTEXT
✓ Current request parameters	CONTEXT
! Caller-supplied identity claim	UNTRUSTED

Trusted source → Normalized context

02

How context reaches Cedar

Source · identity or session system | Check · signature and provenance | Shape · action-specific context

1 **Establish** — Authenticate caller

2 **Verify** — Signed claims

3 **Normalize** — Expected types

4 **Authorize** — PARC request

AUTHENTICATION	PROVENANCE	VALIDATION
Occurs before Cedar	Retain where sensitive facts came from	Check the action-specific context shape

03

Four trust interpretations

Cedar does not authenticate — AUTHN

The application establishes who or what is making the request.

Context is request-specific — REQUEST

Session posture and current parameters belong in the context record.

Entities hold durable facts — PERSISTENT

Departments, owners, groups, and containers belong on entities.

Trust is upstream — BOUNDARY

Cedar evaluates the facts it receives. Trusted systems must establish them.

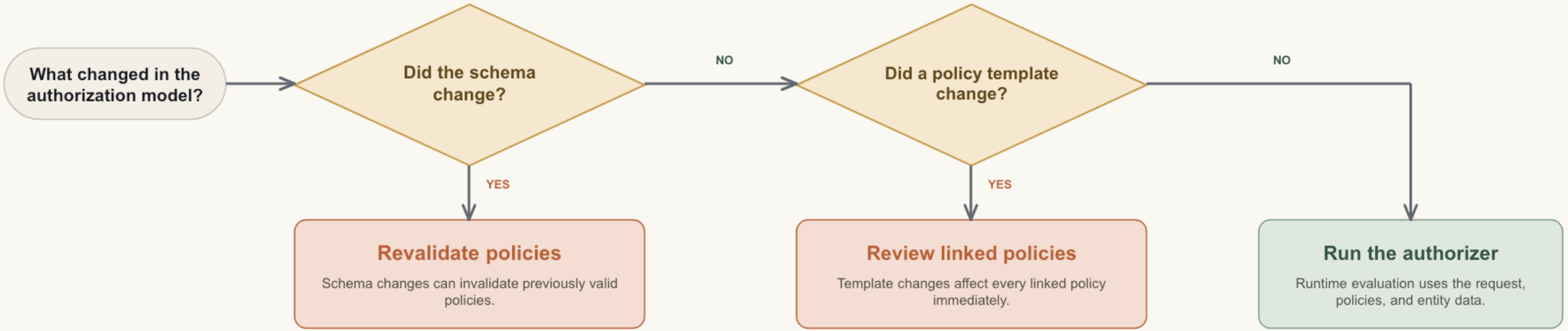
TRUST CHECK

- Authenticate the caller
- Verify sensitive claims
- Normalize values
- Validate context shape
- Record provenance

© 2026 Luminity Digital, Inc. | Independent synthesis of Cedar first-party sources.

Cedar policy and decision model

Schema and policy validation govern design-time quality. Runtime authorization still uses requests, policies, and entities.



RETAIN THESE CONTROLS WITH EVERY CHANGE

Entity and action types — MODEL

The schema declares attributes, hierarchy, actions, and applies-to types.

Policy validation — CHECK

Validation detects type, action, attribute, and optional-value errors.

Template linkage — REUSE

Templates reuse policy structure with principal and resource placeholders.

Runtime separation — BOUNDARY

The authorizer does not use the schema to decide a request.

ARCHITECTURE RULE

- Validate after schema changes
- Review template blast radius
- Keep schema out of runtime
- Keep deployment controls separate

Luminity Digital
INTELLIGENCE, ENGINEERED, DELIVERED.

© 2026 Luminity Digital, Inc. | Independent synthesis of Cedar first-party sources.

Cedar Schema and Cedar Policy Templates · verified Nov 13, 2026