

SHARED FOUNDATION · 01/06

ECZ-ID Reciprocal Trust Technical Overview

A customer-level explanation of persistent machine identity, responsibility, supported authority, current state, bindings, evidence and re-checkability.

PRIMARY READER

CTO, CISO, enterprise architect, security architect, platform architect, technical evaluator

SECONDARY READER

Procurement, governance, developers and operational stakeholders

WHEN TO READ THIS

You have just discovered ECZ-ID and need to understand the system, its boundaries and whether it is relevant to your estate.

RELEASE

2026.10
16 October 2026

What you will know after reading

You can explain the ECZ-ID trust model, the seven free Passport families, the system boundaries, the difference between identity and authority, and how proof can be independently re-checked.

What this document does not establish

It does not certify safety, replace IAM or runtime controls, grant blanket authority, or turn every item of evidence into an independently verified fact.

Permanent route: <https://developers.ecocitizenz.com/guides/reciprocal-trust-technical-overview/> · Support: support@ecocitizenz.com

EXECUTIVE EXPLANATION

Why machine interactions need a durable trust layer

Machine actors can change endpoint, provider, runtime, model, package or deployment context. The identity and proof used to evaluate them should not disappear every time that surrounding infrastructure changes.

The problem

Authentication can prove possession of a credential at a moment in time, but it does not by itself answer which durable object is being represented, who claims responsibility, what published context is current, or how a counterparty can re-check that context later.

The ECZ-ID role

ECZ-ID adds a persistent identity and bounded evidence layer. It is designed to complement - not replace - authentication, authorisation, IAM, runtime policy, security controls and provider-native identity.

Reciprocal trust means relevant counterparties can present or resolve their own identity and context, then evaluate the relationship between them before consequence.

Persistent identity

Keep one stable ECZ-ID while supported bindings and surrounding infrastructure evolve.

Bounded context

Represent responsibility, state, bindings, evidence and supported authority/delegation where present.

Independent re-check

Use Resolver to refresh the bounded public proof instead of relying only on a stale assertion.

SYSTEM ARCHITECTURE

One canonical truth layer, separate public and customer surfaces

The architecture deliberately separates where truth lives from where customers acquire, manage, document and publicly verify it.

TrustOps

Acquisition, entitlement where applicable, activation and lifecycle handoff.

Core / Backend

Canonical identity, authority, state, custody and evidence truth.

Resolver

Public, read-only proof for human and machine re-check.

Provider integrations

Provider-native adapters and reference estates. Provider IDs remain bindings/context, not authority roots.

Developer Gateway

Documentation, interoperability, APIs, schemas, SDKs and implementation guidance.

Dashboard

Authenticated customer control plane for permitted estate actions.

One canonical truth layer; separate acquisition, control, public proof and integration surfaces.

Architectural law

Core / Backend is the canonical truth source. TrustOps, Dashboard, Resolver, Developer Gateway and provider adapters have distinct roles and must not become competing identity authorities.

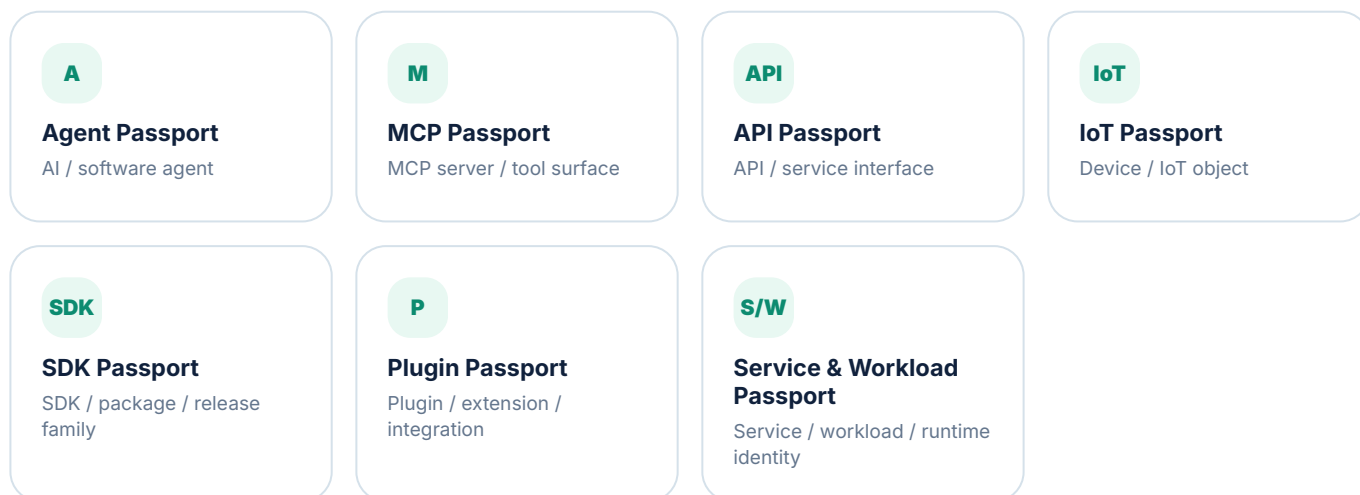
Provider neutrality

AWS, Azure, Google Cloud or another provider may contribute acquisition context, bindings or evidence. A provider-native identifier does not replace the ECZ-ID authority root.

PASSPORT FAMILIES

Seven free Passport families specialise the identity layer

A Passport applies the same ECZ-ID trust grammar to a specific machine/software object. The seven Passport identity primitives are free.

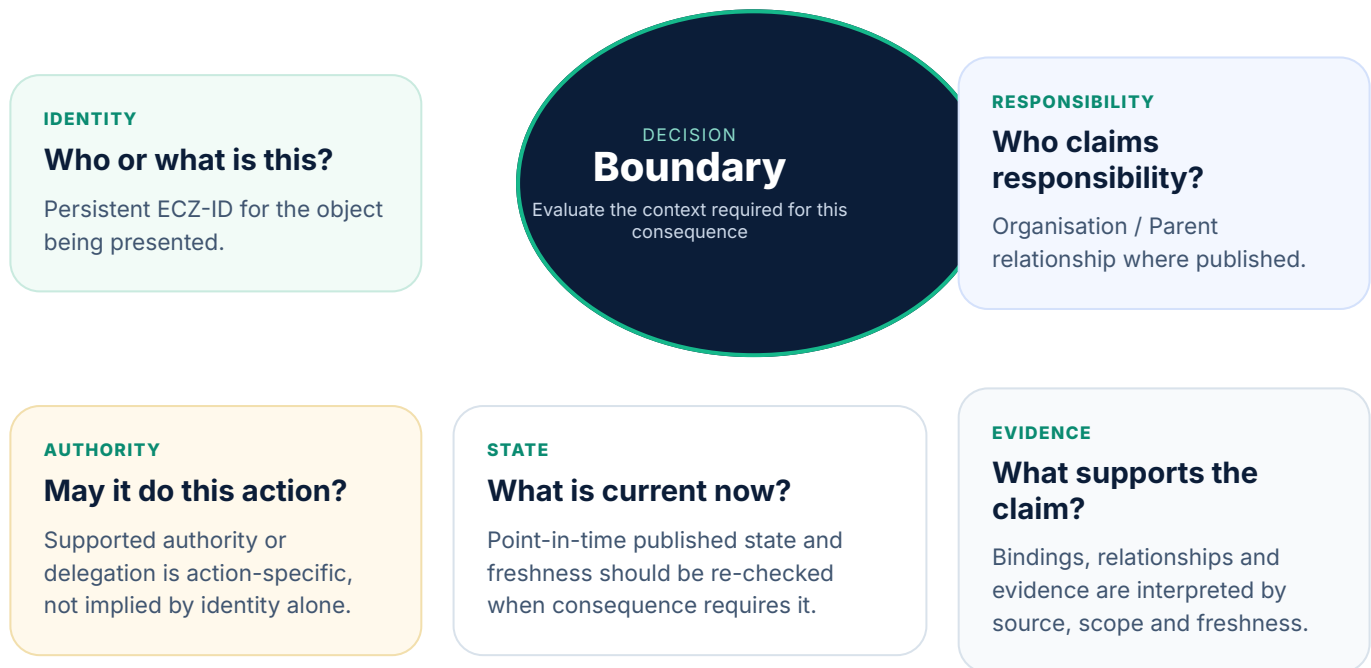


Shared model, different object. Every family answers the same high-level questions - identity, responsibility, current state, bindings, evidence and supported authority context - but the actual object and relevant bindings are family-specific.

DECISION BOUNDARY

Identity, responsibility and authority are different questions

A safe machine decision should not collapse identity, responsibility, authority and evidence into a single “trusted” label.



Boundary: A Passport is identity and accountability infrastructure. It is not, by itself, permission to spend, call an API, invoke a tool, control a device or perform another consequential action.

CURRENT STATE

State and freshness make proof usable over time

Identity may remain stable while published context changes. A previous observation should not be treated as current forever.

Identity can persist

The ECZ-ID can remain referenceable as endpoints, versions, runtimes or provider context change.

State can change

Published lifecycle/current state is point-in-time context. Interpret exactly what is present.

Freshness is a policy question

Re-resolve when the consequence requires current proof. Higher-consequence decisions usually justify fresher evidence.

1

Resolve

Obtain the ECZ-ID and retrieve current public proof.

2

Inspect

Read family/object, responsibility, state, bindings, evidence and supported authority context.

3

Evaluate

Apply your local policy for the interaction.

4

Re-check

Refresh the record when freshness matters.

Re-checkability is part of the trust model. ECZ-ID does not ask counterparties to accept a one-time screenshot or assertion indefinitely.

BINDINGS & EVIDENCE

Bindings connect stable identity to a changing world

Bindings associate an ECZ-ID with external surfaces; evidence supports bounded claims. Neither should be interpreted more broadly than its source and scope allow.

Bindings may point to
Provider-native identities, endpoints, manifests, repositories, package registries, API descriptions, device/service references, runtime identifiers and other supported machine identities.

Evidence may support
Identity, responsibility, relationships, current state, bindings and authority/delegation context where the canonical record contains appropriate support.

QUESTION	SAFE INTERPRETATION	UNSAFE LEAP
Is a provider ID present?	It is published as a binding/context reference.	The provider ID has become the ECZ-ID authority root.
Is evidence present?	Evaluate its source, scope and freshness.	The mere presence of evidence means the object is universally safe.
Is a relationship present?	The record publishes that relationship.	Every underlying assertion has been independently verified.

RECIPROCAL TRUST

Reciprocal trust evaluates the interaction, not one opaque actor

Machine-to-machine consequence often involves more than one identity. An agent, MCP server, API, workload and organisation may all matter to the same decision.



Reciprocal evaluation

Each relevant counterparty can resolve its own context. Policy evaluates the relationships and evidence between them rather than trusting one party's self-description.

No universal trust score

The output should be decision-specific: allow, challenge, deny, require fresher evidence, require authority, or route to another control.

PROVIDER INTEGRATIONS

Provider integrations add reach without taking over identity authority

ECZ-ID is intended to remain portable across provider and runtime contexts. Provider-native identities are useful - but they remain provider context.

Acquisition context

A marketplace or provider can be the place where a customer discovers and activates a Passport.

Bindings & evidence

Provider-native account, project, service, workload or other identifiers can be recorded when supported.

Canonical identity

The ECZ-ID remains the persistent identity that counterparties resolve and re-check.

Design implication: moving an object to another provider or endpoint should normally update supported bindings/context, not force a new ECZ-ID merely because infrastructure changed.

CONTROL VS PROOF

Dashboard controls; Resolver proves

Authenticated customer operations and public verification are intentionally separate.

DASHBOARD**Authenticated control plane**

- View and manage permitted estate actions.
- Work with Passports & Identity, Products & Entitlements, Operations, Evidence & Proof, sharing and organisation areas where available.
- Use account recovery and authenticated customer controls.
- Do not treat Dashboard as the public proof surface.

RESOLVER**Public read-only proof**

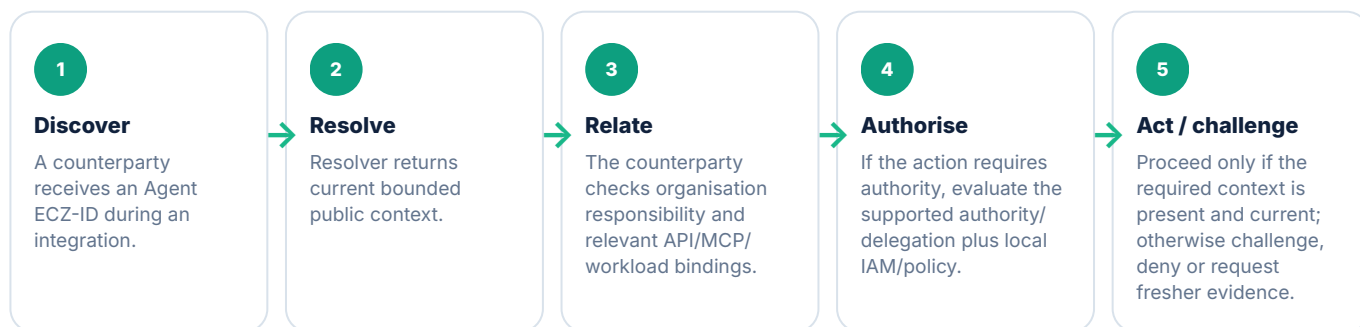
- Independently re-check bounded published proof.
- Inspect identity, family/object, responsibility, state, bindings, evidence and supported authority/delegation where present.
- Use from human or machine workflows.
- Do not mutate canonical identity from Resolver.

Machine-readable route: GET https://api.ecocitizenz.com/api/p/{ecz_id}.json

EXAMPLE INTERACTION

Example: from discovery to a bounded machine decision

This example is illustrative. The required evidence and policy depend on the actual action and risk.



Important: ECZ-ID contributes identity, accountability context and re-checkable evidence. Authentication, authorisation and runtime enforcement still belong to the systems responsible for those controls.

QUICK REFERENCE

What ECZ-ID establishes - and what it does not

Use this page as the executive boundary check before adopting or integrating ECZ-ID.

ECZ-ID is designed to establish

- A persistent ECZ-ID for the relevant object.
- Published responsibility/Parent context where present.
- Current published state and relationships.
- Bindings and evidence references where supported.
- Supported authority/delegation context where present.
- A public Resolver path for independent re-check.

ECZ-ID does not establish by itself

- Universal safety or correct future behaviour.
- Blanket permission to spend, call APIs, use tools or control systems.
- Certification, standards compliance or independent verification of every declaration.
- Replacement for IAM, authentication, authorisation, runtime controls or security testing.

Next: Read Passport and Parent Identity Model, then the relevant Passport family guide. For verification workflows, use the Resolver and Passport Verification Guide.