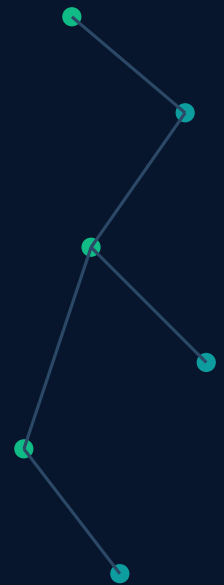


ECZ-ID Agent Passport Technical Reference

External integration reference for the Agent object, persistent identifier, supported context, Resolver representation, lifecycle and fail-safe interpretation.



AGENT PASSPORT · TECHNICAL REFERENCE

PRIMARY READER

Developer, enterprise architect, security engineer, integration engineer and automated-system designer.

SECONDARY READER

Platform engineers, technical evaluators, operations and identity teams.

WHEN TO READ THIS

You understand Agent Passport conceptually and need to integrate its identity and proof into software or architecture.

RELEASE

2026.10 · 15 October 2026

WHAT YOU WILL KNOW AFTER READING

You can model the canonical Agent object, interpret supported bindings/evidence/state/authority, resolve it programmatically and handle absent or stale data safely.

WHAT THIS DOCUMENT DOES NOT ESTABLISH

This is not an internal schema dump and does not promise that every optional field or evidence type will be present for every Passport.

NEXT

Use the Operations guide for recurring checks, change handling, lifecycle and support escalation.

OBJECT MODEL

The canonical object is the AI / software agent

The Agent ECZ-ID is the persistent reference for the intended logical agent. Provider-native IDs and endpoints are context, not authority roots.

What an Agent Passport can represent

Persistent identity at the centre; bounded context around it; Resolver for independent re-check.



Canonical object

AI / software agent

Stable identifier

Persistent Agent ECZ-ID for the intended object

INTEGRATION LAW

Treat endpoints, provider identities, manifests and runtime roles as bindings/evidence around the Agent ECZ-ID - not as replacements for the canonical identity.

IDENTITY AND RESPONSIBILITY

Parent / organisation context is a relationship, not a universal assurance

The integration should evaluate the published relationship and evidence actually present rather than infer responsibility or verification from naming or provider context.

AGENT IDENTITY

The Agent ECZ-ID answers "which agent record is this?" and remains the stable reference while supported context changes.

PARENT / ORGANISATION

The record may publish an organisation or Parent relationship that represents responsibility context where supported.

PROVIDER CONTEXT

AWS account/customer context, endpoints and other provider-native identifiers remain bindings or evidence.

DUPLICATE PREVENTION

Reuse the canonical account, Parent and Passport where appropriate rather than minting a new identity for the same logical agent.

INTERPRETATION BOUNDARY

A relationship can be published without implying that every underlying assertion has been independently verified. Read verification/assurance state from the evidence actually present.

BINDING MODEL

Connect persistent identity to the changing world

Bindings are external surfaces associated with the Agent ECZ-ID according to the current record and evidence.

A2A Agent Card / manifest

Where present, a public agent description or well-known surface can be associated with the Agent ECZ-ID.

MCP-related surface

Connect the Agent to relevant MCP context without treating that surface as the identity root.

API / endpoint

Associate operational endpoints while keeping identity independent of one transient location.

Repository / package

Reference implementation or package context where supported.

Provider-native IDs

Keep cloud/runtime/provider identifiers as bindings, never canonical authority roots.

Other supported machine surfaces

Represent additional relationships when supported by the canonical implementation and evidence.

BINDING SEMANTICS

A binding says the ECZ-ID record is associated with that surface according to current state/evidence; it is not a statement that the external surface itself is the authority root.

EVIDENCE MODEL

Source, scope and freshness matter

Evidence can support claims about identity, relationships, state, bindings or authority. The presence of evidence does not turn the whole Passport into an unlimited “trusted” verdict.

SOURCE

Where the supporting information comes from. Consumers should understand the origin before relying on it.

SCOPE

What exact claim or context the evidence supports. Avoid extending it beyond its intended meaning.

FRESHNESS

When the evidence was observed or updated relative to the decision being made.

Likely evidence categories

Organisation responsibility context, manifest/endpoint evidence, supported relationship evidence and authority/delegation evidence where present.

Absent evidence

If required evidence is absent, treat the claim as not established. Do not infer it from object type, provider context or marketing copy.

SAFE INTERPRETATION

Evaluate the evidence needed for the decision actually being made; re-resolve when freshness policy requires current proof.

CURRENT STATE AND FRESHNESS

State is point-in-time, not a permanent conclusion

Agent estates change. Consumers should treat current state as a signal from the canonical system and re-resolve before consequential use when freshness matters.

CURRENT PUBLISHED STATE

Read the state actually published for the Agent Passport. Do not invent or infer lifecycle labels that are not present.

FRESHNESS POLICY

Define when your integration must re-resolve based on consequence, risk and how quickly the relevant context can change.

RE-CHECK PRINCIPLE

A previously observed proof should not be assumed current forever. Resolve again when endpoint, operator, evidence, state or authority changes could affect the decision.

Safe cache behaviour

Cache according to your use case, but invalidate/re-resolve when the action requires fresher proof.

Unsafe cache behaviour

Treat an old successful resolution as a permanent trust verdict regardless of subsequent changes.

AUTHORITY / DELEGATION SEMANTICS

Action-specific context, not blanket permission

Identity and responsibility can support policy decisions, but they do not themselves grant permission to take a consequential action.

Use identity as input to the decision - not as the decision itself

Resolve the Agent, evaluate the action-specific context you need, then apply local policy.

**BOUNDARY**

Agent Passport can support the decision; it does not automatically grant permission to spend, call APIs or invoke tools.

Expected integration

Resolve identity, evaluate required authority/delegation where present, then apply local policy.

Do not infer

Spend permission, API/tool permission or system access solely from possession of an Agent Passport.

FAIL SAFELY

If action-specific authority or evidence is required and absent, challenge or deny according to your policy rather than assuming permission.

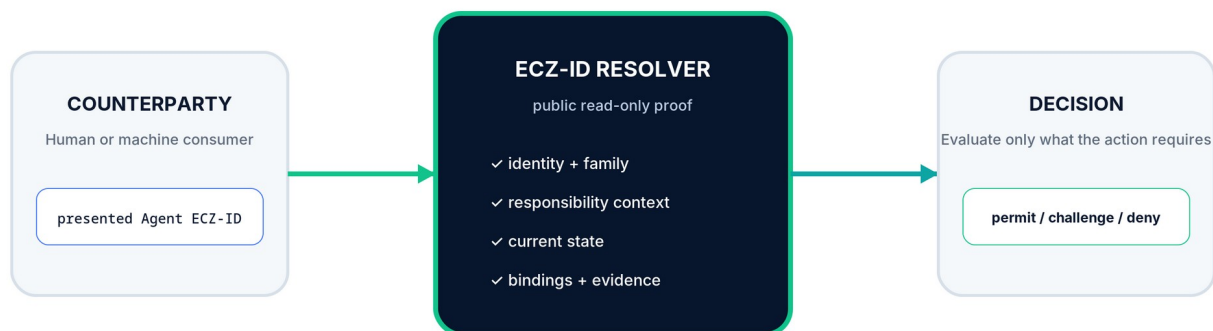
RESOLVER REPRESENTATION

Human and machine-readable re-check

Resolver is the public, read-only proof surface. Automation should evaluate only the fields and evidence actually returned.

Independent re-check when freshness matters

Human and machine consumers can resolve the same Agent ECZ-ID again before consequence.



GET https://api.ecocitizenz.com/api/p/{ecz_id}.json

Absent required data = not established. Re-resolve according to freshness policy.

MACHINE-READABLE ROUTE

GET https://api.ecocitizenz.com/api/p/{ecz_id}.json

- Validate the ECZ-ID and Passport family = Agent.
- Read current published state.
- Compare the expected organisation / Parent relationship where relevant.
- Evaluate only the bindings/evidence needed for the current interaction.
- Evaluate authority/delegation only for the specific action when required.
- Re-resolve according to freshness policy.

INTEGRATION PATTERN

Pre-consequence verification sequence

Use Agent Passport as structured input to your control decision rather than as a universal allow/deny score.

1

Resolve

Resolve the Agent ECZ-ID from the official Resolver route.

2

Validate identity

Confirm family/object and expected persistent identifier.

3

Evaluate context

Inspect responsibility, state, bindings and evidence relevant to this integration.

4

Evaluate authority

If the action requires delegation/authority, verify the supported context separately.

5

Apply local policy

Permit, challenge or deny in your own runtime/control plane.

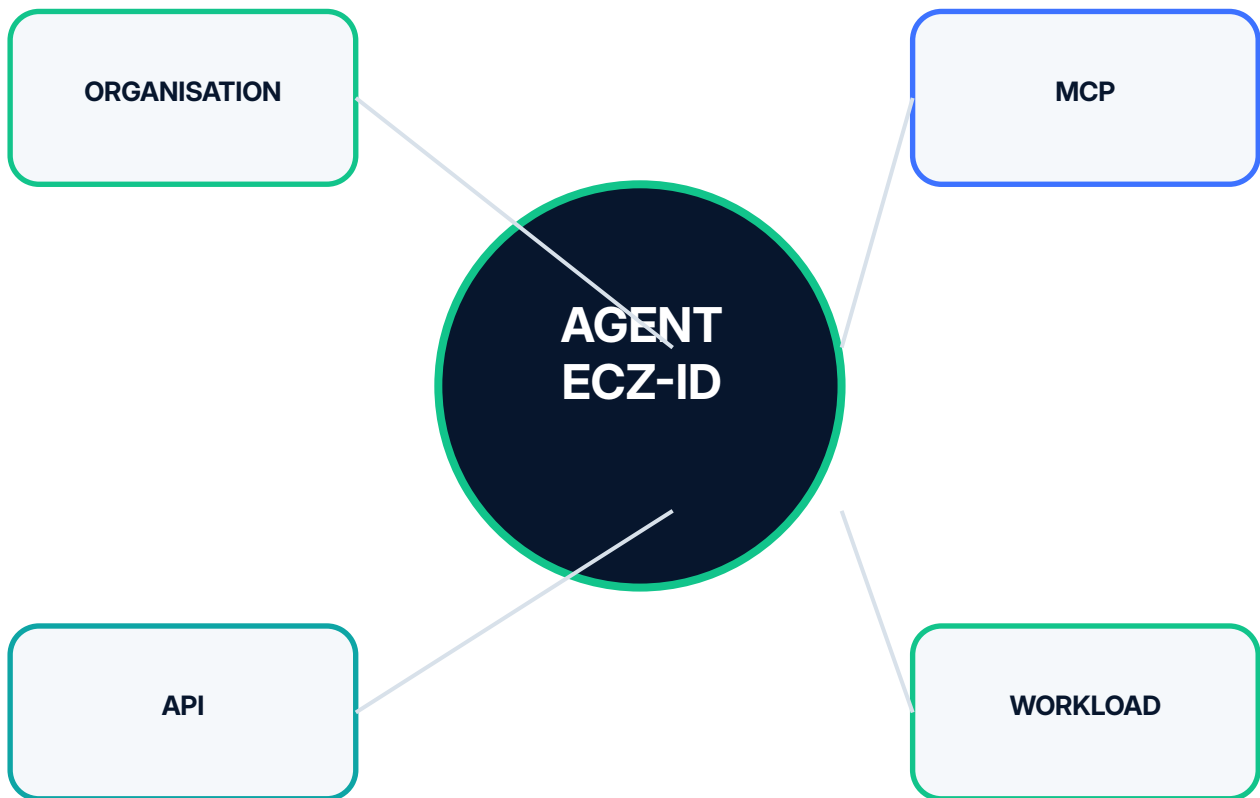
FAILURE-SAFE RULE

Missing required fields/evidence should cause your integration to follow its challenge/deny/fallback policy - not fabricate an identity or trust conclusion.

RELATIONSHIP PATTERNS

Agent ↔ MCP ↔ API ↔ organisation

A real machine interaction can involve several ECZ-ID objects. Reciprocal trust means each relevant counterparty can present or resolve its own identity and context.

**RECIPROCAL TRUST**

The interaction can evaluate relationships and evidence between independently resolvable objects rather than relying on one opaque actor assertion.

LIFECYCLE AND MUTATION MODEL

Keep the identity stable; update the supported context

Operational changes should normally mutate state, bindings or evidence around the canonical Agent ECZ-ID rather than replace the identity.

Operate the Passport as a living proof record

Identity remains stable while operational context is reviewed, updated and re-checked.



Model / endpoint change

Update supported context for the same logical agent.

Responsibility change

Use supported control path and re-check published relationship.

Evidence / authority change

Update the relevant bounded context and re-resolve before consequential use.

DO NOT REPURPOSE

Do not recycle one canonical Agent ECZ-ID for a different logical object.

FAILURE HANDLING

What integrations must do when the record is incomplete, stale or unexpected

Fail-safe integration behaviour is part of the technical contract: absent required data is not evidence of permission or trust.

Passport does not resolve

Confirm identifier and distinguish not-found from network/client failure. Do not fabricate a fallback identity.

Required binding absent

Treat the integration requirement as not established; request appropriate evidence/context or follow challenge/deny policy.

Evidence stale or out of scope

Re-resolve or obtain current evidence rather than extending an old conclusion.

Unexpected Parent / organisation

Challenge the interaction and investigate rather than silently accepting the mismatch.

Authority context absent

If the action requires it, do not infer permission from identity alone.

Resolver/network failure

Follow your availability/failure policy; do not convert transport failure into a positive trust verdict.

SECURITY HANDLING

Never log or transmit passwords, API keys, AWS Marketplace registration tokens, private keys/certificates or other unrelated customer-private data through public proof paths.

DEVELOPER QUICK REFERENCE

The external integration contract in one page

Use this as the implementation review checklist.

DO

- Treat Agent ECZ-ID as persistent identity.
- Validate family/object.
- Evaluate current state.
- Use only relevant bindings/evidence.
- Re-resolve when freshness matters.

DO NOT

- Treat endpoint/provider ID as authority root.
- Infer absent evidence.
- Turn one field into a universal trust score.
- Assume blanket authority.
- Cache a trust conclusion forever.

RESOLVER ROUTE

GET https://api.ecocitizenz.com/api/p/{ecz_id}.json

Canonical object

AI / software agent

Support

support@ecocitizenz.com

NEXT DOCUMENT

For recurring checks, change handling, lifecycle and recovery, use ECZ-ID Agent Passport Operations, Lifecycle and Troubleshooting Guide.