

ECZ-ID MCP Passport Operations, Lifecycle and Troubleshooting Guide

Run the MCP Passport safely as endpoints, tools, evidence and operator context change over time.

MCP PASSPORT · OPERATIONS

PRIMARY READER

Administrator, platform operations, security operations, support engineer and estate owner.

SECONDARY READER

MCP operator, AI platform owner, developer and security architecture teams.

WHEN TO READ THIS

The MCP Passport already exists and you need to keep its published proof accurate and useful.

RELEASE

2026.10 · 15 October 2026

WHAT YOU WILL KNOW AFTER READING

You can perform operational checks, update changing context, re-check Resolver proof, avoid duplicates, recover access and escalate failures safely.

WHAT THIS DOCUMENT DOES NOT ESTABLISH

Operations guidance does not replace runtime IAM, secrets management, authorization or security incident procedures.

NEXT

Use the Technical Reference when you need machine integration semantics or strict failure-handling behaviour.

FIRST OPERATIONAL CHECK

Know the identity before you change anything

Before modifying bindings or evidence, confirm which MCP ECZ-ID is the canonical record for the logical server/tool surface.

Operate the MCP Passport as a living proof record

Keep identity stable while endpoint, tool surface, state and evidence are reviewed and updated.



RULE

Operate the existing identity where supported. Do not create a second Passport merely because the endpoint, tool list, runtime or provider changed.

DASHBOARD VS RESOLVER

Control privately; prove publicly

Operational teams need to know which surface is authoritative for changes and which surface is used by counterparties to verify the result.

DASHBOARD

Authenticated customer control plane for permitted Passport, evidence, sharing and organisation/account actions.

RESOLVER

Public, read-only proof surface for re-checking the bounded record. Resolver does not mutate identity.

OPERATING LAW

Make changes through the supported authenticated control path, then independently read back the public proof through Resolver.

ENDPOINT AND TOOL CHANGES

Update context without losing identity continuity

MCP operations commonly involve endpoint migration, tool additions/removals, manifest changes, new versions or provider/runtime moves.

ENDPOINT

Update the supported endpoint binding/evidence. Then re-check Resolver.

TOOLS / MANIFEST

Update the relevant tool-surface reference/evidence. Treat change as policy-relevant where required.

PROVIDER / RUNTIME

Update provider-native/runtime bindings while keeping the canonical MCP ECZ-ID stable.

DO NOT

Do not “solve” a migration by minting another MCP Passport for the same logical surface unless the object itself is genuinely new.

EVIDENCE AND CHANGE REVIEW

Keep bounded proof current

Evidence should be reviewed according to source, scope and freshness. Where change history or trust-epoch context is available, use it to understand what changed since the last review.

EVIDENCE REVIEW

Confirm the evidence still supports the claim you rely on, has the expected source/scope, and is fresh enough for the intended use.

CHANGE REVIEW

A tool/manifest/endpoint change may require counterparties to re-evaluate policy even when the canonical MCP ECZ-ID remains the same.

SECURITY SIGNAL BOUNDARY

Secret-exposure or dangerous-chain warnings, where supported by evidence/tooling, are signals to investigate - not a universal trust score.

OPERATOR AND ORGANISATION CHANGES

Change the relationship carefully

If the responsible organisation/operator changes, update the supported relationship through the canonical customer control path and verify the resulting public record.

BEFORE CHANGE

Confirm the canonical MCP ECZ-ID, current Parent/operator context, current state and any dependent integrations that rely on the relationship.

AFTER CHANGE

Re-check Resolver, confirm expected relationship/state, and notify dependent policy owners when the change affects authorization or onboarding assumptions.

BOUNDARY

A published relationship says what the ECZ-ID record represents. It does not automatically transfer credentials, tool permissions or private runtime access.

LIFECYCLE STATES

Make lifecycle changes visible to counterparties

Operational truth is useful when counterparties can tell that a surface is current, suspended, degraded, withdrawn or otherwise changed according to the canonical state model.

ACTIVE / CURRENT

Use current published state according to the actual canonical implementation.

SUSPEND / DEGRADE

Reflect the relevant lifecycle state rather than leaving a stale "looks normal" record.

REVOKE / RETIRE

Preserve evidence/history as supported while preventing consumers from treating retired context as current.

DO NOT INVENT STATE NAMES

Use the lifecycle/state values actually supported by the canonical ECZ-ID implementation. This guide describes the operational principle, not an internal enum contract.

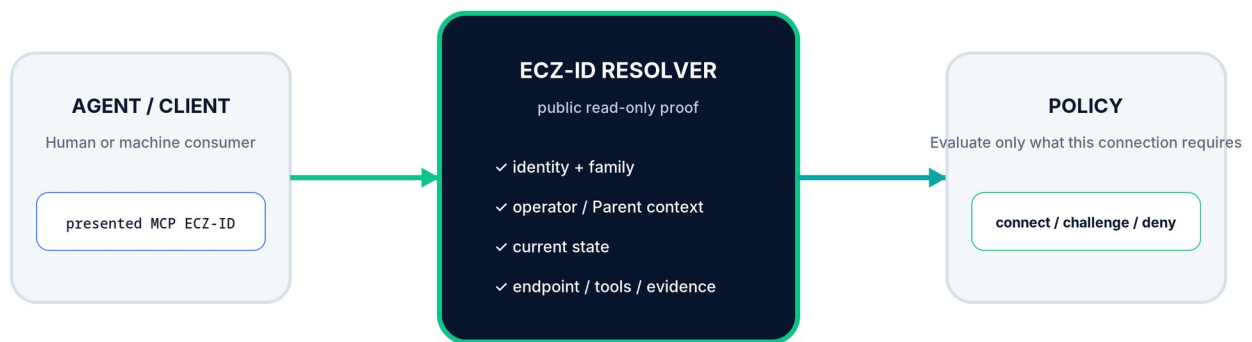
RESOLVER RE-CHECK POLICY

Freshness belongs in operations

Use re-resolution as an operational control whenever endpoint, tools, operator, evidence or state can change faster than your cached decision.

Independent re-check before an agent connects or invokes

Human and machine consumers can resolve the same MCP ECZ-ID again when endpoint, tools or state may have changed.



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

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

POLICY PATTERN

Choose a freshness threshold appropriate to the consequence. Re-resolve immediately before high-consequence connections/invocations if your policy requires it.

DUPLICATE PREVENTION

One logical MCP surface, one canonical identity where supported

Duplicates create ambiguity for operators, agents and counterparties.

WHEN TO REUSE

Endpoint changes, tool-set changes, version/runtime updates and provider moves normally update context around the same logical MCP object.

WHEN A NEW ID MAY BE LEGITIMATE

A genuinely distinct MCP server/tool surface that should be independently identifiable can require a separate Passport.

BEFORE MINTING

Search/inspect the existing estate and confirm you are not creating a second identity because of a discovery, login or binding problem.

ACCOUNT RECOVERY AND ACCESS

Recover control without duplicating the estate

A sign-in or recovery problem is not evidence that the MCP Passport should be reissued.

- Use the supported account recovery/sign-in path.
- Confirm you are accessing the correct organisation/account context.
- Do not create a new Parent or Passport merely to work around access problems.
- After recovery, inspect the existing MCP Passport and re-check public Resolver proof.
- Escalate with ECZ-ID/Passport ID, failed step, visible error and occurrence time - never credentials.

SUPPORT

support@ecocitizenz.com

COMMON FAILURES

Diagnose by layer

Separate acquisition/account/control problems from proof/binding/freshness problems.

WRONG / MISSING ENDPOINT

Check the canonical binding and whether the expected update completed. Do not infer the old endpoint is still current.

TOOL SURFACE MISMATCH

Confirm manifest/tool evidence and whether the consumer is using stale cached context.

MISSING EVIDENCE

Treat it as absent. Do not fill gaps from marketing or previous assumptions.

RESOLVER NOT FOUND

Confirm ECZ-ID, acquisition/lifecycle state and distinguish authoritative not-found from network/client failure.

SUPPORT ESCALATION

Send enough context - never secrets

Good support context shortens diagnosis while preserving security.

SEND

ECZ-ID / Passport ID
Failed step
Visible error
Occurrence time
Expected endpoint/tool surface
Relevant non-secret AWS/account context

NEVER SEND

Passwords
API keys
Bearer tokens
AWS registration tokens
Certificates/private keys
Private MCP credentials/config

SUPPORT

support@ecocitizenz.com

ONE-MINUTE OPERATOR RUNBOOK

The recurring operational rhythm

Use this as the minimum repeatable check for an MCP Passport in service.

- 1. Confirm you are operating the canonical MCP ECZ-ID.
- 2. Inspect current published state.
- 3. Check endpoint/tool bindings against the intended MCP surface.
- 4. Review relevant evidence and change context.
- 5. Confirm operator/Parent relationship where policy depends on it.
- 6. Re-resolve public proof after changes.
- 7. Apply local IAM/auth/tool policy separately.
- 8. Escalate anomalies without sending secrets.

GOAL

Keep identity stable, context current and proof independently re-checkable.