

ECZ-ID

RECIPROCAL TRUST INFRASTRUCTURE
FOR THE MACHINE ECONOMY

PUBLIC CUSTOMER RESOURCE

ECZ-ID API Passport AWS Marketplace Getting Started and Technical Guide

From AWS Marketplace subscription to a canonical API ECZ-ID with independent Resolver proof.

API PASSPORT · AWS GETTING STARTED

PRIMARY READER

AWS Marketplace subscriber, cloud administrator, developer and customer administrator.

SECONDARY READER

Procurement, platform engineering, security operations and API owners.

WHEN TO READ THIS

You have subscribed - or are about to subscribe - and want a working API Passport with minimal friction.

RELEASE

2026.10 · 16 October 2026

WHAT YOU WILL KNOW AFTER READING

You can move from AWS acquisition through TrustOps to one canonical API Passport, Dashboard control and Resolver verification.

WHAT THIS DOCUMENT DOES NOT ESTABLISH

This guide does not make AWS identity the ECZ-ID authority root or replace API authentication, authorization or runtime controls.

NEXT Complete the acquisition, then use the API Technical Reference to integrate Passport proof into your API/service workflow.

[START HERE](#)

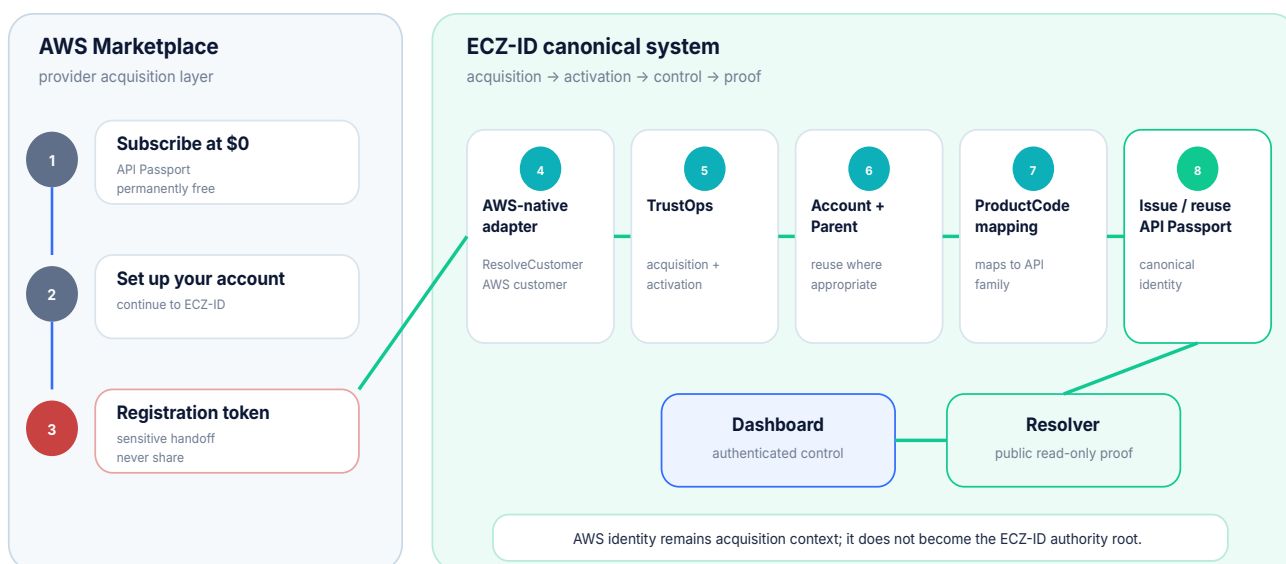
The whole journey in one view

The API Passport is permanently free. Subscribe in AWS, complete the account handoff, then verify the canonical API ECZ-ID in Resolver.

AWS MARKETPLACE ACQUISITION

Activate an API Passport through AWS Marketplace

Provider acquisition is separated from the canonical ECZ-ID identity, control and proof system.



SUCCESS CONDITION

The intended API/service interface has one canonical API ECZ-ID, visible in Dashboard and independently re-checkable in Resolver.

AWS ROLE

Acquisition/provider context and customer handoff.

ECZ-ID ROLE

Canonical identity, activation, control and proof.

STEP-BY-STEP ACQUISITION

1. Subscribe → 2. Set up your account → 3. registration-token handoff

AWS Marketplace provides the acquisition context. The registration token is sensitive and must never be copied into public documentation, screenshots or support channels.

1 · SUBSCRIBE AT \$0

Select the ECZ-ID API Passport listing and subscribe. There is no ECZ-ID software charge for the Passport.

2 · SET UP YOUR ACCOUNT

Use the Marketplace setup action to continue into the ECZ-ID fulfillment path.

3 · AWS REGISTRATION TOKEN

AWS passes a registration token to the fulfillment endpoint. Treat it as sensitive operational input: do not paste it into chat, email, tickets, screenshots or public documentation.

EXPECTED RESULT

The customer handoff reaches the ECZ-ID AWS-native adapter.

IF IT FAILS

Record the visible error and occurrence time.
Do not expose the token.

RESOLVECUSTOMER AND TRUSTOPS

4. identify the AWS customer → 5. activate the ECZ-ID acquisition flow

The AWS-native adapter resolves the Marketplace customer and hands acquisition into TrustOps.

AWS-NATIVE ADAPTER

Uses the appropriate Marketplace integration to ResolveCustomer and associate the acquisition with the ECZ-ID flow.

TRUSTOPS

Processes acquisition, applies the supported entitlement/activation logic and orchestrates the lifecycle handoff.

ARCHITECTURE LAW

AWS customer identity is acquisition/provider context. It does not become the canonical ECZ-ID identity authority.

Fulfillment endpoint

<https://trustops.ecocitizenz.com /aws/m arketplace/fulfill>

DO NOT RETRY BY CREATING NEW IDENTITIES

If acquisition fails, diagnose the failed stage. Repeated retries should not be used as a workaround for account, Parent or Passport issues.

ACCOUNT, PARENT AND PASSPORT REUSE

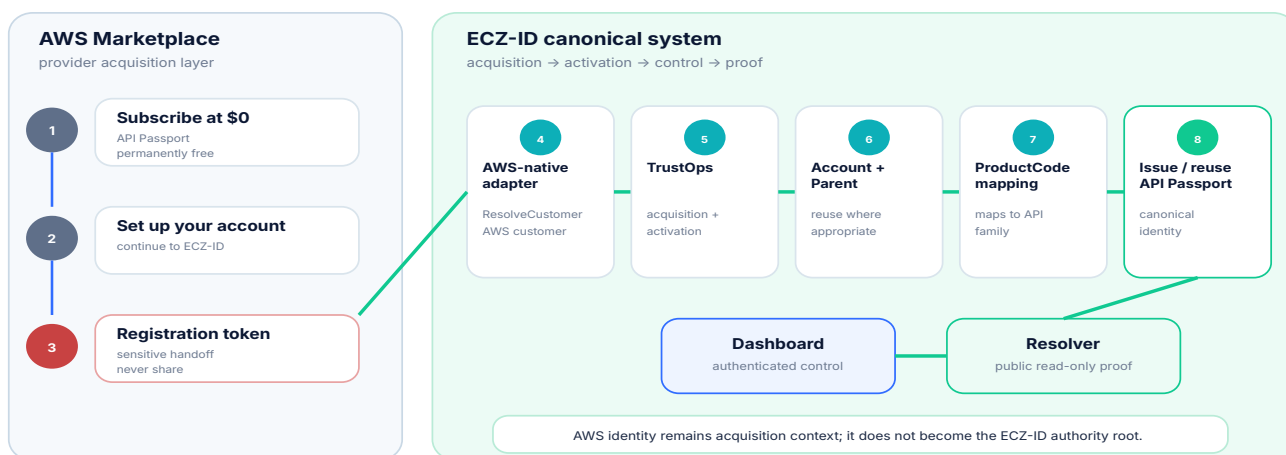
6. reuse or create account + Parent → 7. map ProductCode → 8. issue or reuse Passport

The target outcome is one canonical identity for the intended API/service interface, not a new Passport for every subscription click.

AWS MARKETPLACE ACQUISITION

Activate an API Passport through AWS Marketplace

Provider acquisition is separated from the canonical ECZ-ID identity, control and proof system.

**REUSE FIRST**

Where supported, reuse the existing EcoCitizen account and Parent relationship.

ISSUE OR REUSE

Map the AWS ProductCode to the API Passport family and issue a new API ECZ-ID only when the logical object legitimately needs one.

DUPLICATE PREVENTION

A new AWS subscription event is not, by itself, a reason to mint another canonical API identity.

FIRST CUSTOMER CONTROL

9. continue to Dashboard

Dashboard is the authenticated customer control plane. Resolver is the public proof surface.

DASHBOARD

Use the authenticated control plane to inspect permitted estate actions, the API Passport, relationships, products/entitlements, evidence and sharing surfaces available to your account.

RESOLVER

Use the public, read-only proof surface to inspect what a counterparty can independently re-check.

DO NOT CONFUSE THE TWO

Dashboard can change permitted customer state; Resolver cannot. Resolver proves what is published; it is not the control plane.

FIRST CHECK IN DASHBOARD

Confirm the intended API Passport exists and corresponds to the correct logical interface.

THEN VERIFY PUBLICLY

Resolve the ECZ-ID and confirm the family/object, organisation context, state and supported bindings/evidence.

FIRST RESOLVER VERIFICATION

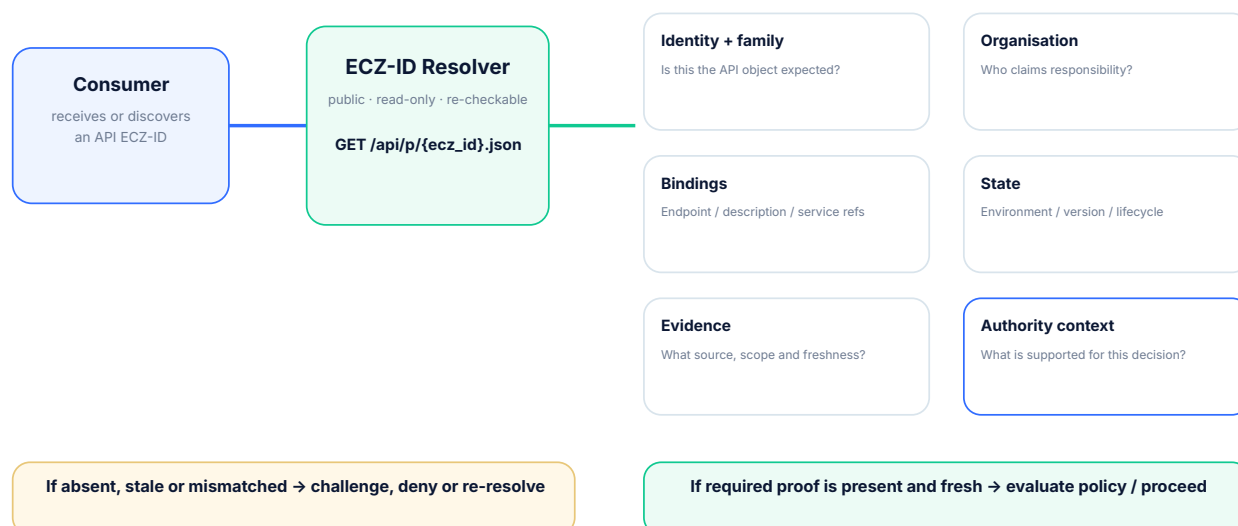
10. re-check the API Passport

A successful AWS acquisition is not complete from the customer perspective until the API ECZ-ID can be independently resolved and interpreted correctly.

RESOLVER RE-CHECK

Turn a presented ECZ-ID into a bounded decision input

Resolve the current public record, inspect only what is present, and fail safely when required proof is missing.



HUMAN VERIFICATION

Open the official Resolver and inspect the API ECZ-ID, family/object, organisation, state, bindings and evidence actually present.

MACHINE VERIFICATION

Use the machine-readable Resolver route and fail safely when required fields or evidence are absent.

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

CALL SAFELY

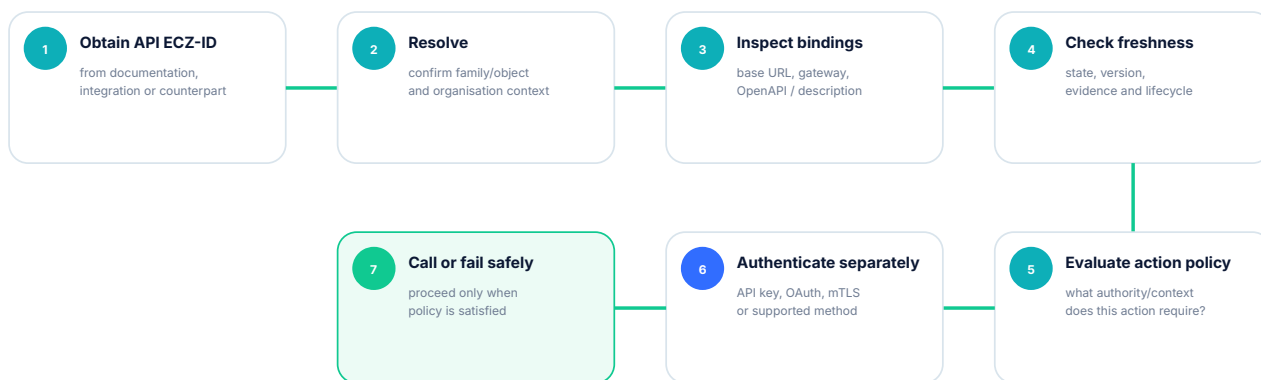
Use Passport proof before connection or invocation

The Passport helps answer “is this the API/service interface I intended, and what current context supports that identity?” It does not answer every security question by itself.

PRE-CALL VERIFICATION

Verify identity and context before an API call

Use Passport proof as one input; keep API authentication and runtime authorization separate.



Boundary: API identity and proof do not replace API authentication or authorization.

If required proof is absent, stale or mismatched, challenge, deny or re-resolve according to policy.

SEPARATE CONTROL

Authenticate to the API using its supported credential/runtime mechanism. Evaluate authorization for the specific operation separately from Passport identity.

PASSPORT INPUT

Identity · responsibility · endpoint/description
bindings · state · evidence.

RUNTIME CONTROL

Authentication · authorization · gateway policy
· application security.

SECURITY AND SECRET HANDLING

What must never enter public proof or support channels

AWS registration tokens and API credentials are operational secrets. The Passport is designed to publish bounded proof, not secrets.

NEVER SEND OR PUBLISH

- AWS Marketplace registration tokens
- passwords
- API keys / OAuth tokens
- private keys or certificates
- secrets from gateway/runtime configuration

SAFE SUPPORT CONTEXT

- AWS account context
- ECZ-ID / Passport ID if available
- failed stage
- visible error
- occurrence time

PUBLIC PROOF RULE

Resolver should expose only the bounded public record. Never use evidence fields, support tickets or screenshots as a place to store credentials.

Support: support@ecocitizenz.com

TROUBLESHOOTING

Diagnose the failed stage before taking action

The fastest recovery path is to identify where the journey stopped, preserve the canonical identity and avoid retries that create ambiguity.

SUBSCRIBE UNAVAILABLE

Confirm you can see the Limited product during testing, or that the listing is Public for general buyers.

SET UP YOUR ACCOUNT FAILS

Record visible error + time. Do not expose the registration token.

RESOLVECUSTOMER / ACTIVATION FAILS

Contact support with AWS context and failed stage; do not create duplicate identities.

EXISTING ACCOUNT NOT RECOGNISED

Use the supported existing-account path rather than intentionally creating another account/Parent.

RESOLVER MISSING EXPECTED PASSPORT

Confirm acquisition completed, then verify the ECZ-ID and distinguish issuance delay/failure from a copied-ID error.

ENDPOINT / API MISMATCH

Do not “fix” by minting another identity; verify bindings and the intended logical API first.

60-SECOND COMPLETION CHECK

Know when you are done

Use this checklist before treating the AWS acquisition as complete.



Subscribed at \$0



Canonical API Passport issued or reused



Set up your account completed



Dashboard shows the intended API identity



AWS handoff reached TrustOps



Resolver independently returns the Passport



Existing account/Parent reused or correctly created



No secrets were copied into public/support channels



ProductCode mapped to API Passport family

NEXT

Use the API Passport Technical Reference to integrate Resolver proof into your API/service workflow.