

ECZ-ID Plugin Passport Overview and Customer Guide

Persistent, resolver-verifiable identity for plugins, extensions and integration components across changing stores, repositories/packages, host platforms and versions.

PLUGIN PASSPORT · OVERVIEW

PRIMARY READER

Buyer, technical evaluator, security architect, developer platform owner.

SECONDARY READER

Developers, enterprise architects, software supply chain, governance and procurement.

WHEN TO READ THIS

You are deciding whether a plugin/extension needs durable identity and independently re-checkable publisher/distribution/host context.

RELEASE

2026.10 · 16 October 2026

WHAT YOU WILL KNOW AFTER READING

You can explain the plugin/extension object, publisher/organisation responsibility, store/repository/package/host/version bindings, state, evidence and safe verification boundary.

WHAT THIS DOCUMENT DOES NOT ESTABLISH

A Plugin Passport does not prove plugin safety, vulnerability-free status, package/build integrity, host approval or permission to install, enable or execute.

NEXT Read the AWS Getting Started Guide when you are ready to acquire and activate the free Plugin Passport.

EXECUTIVE EXPLANATION

A durable identity anchor for plugins and extensions that outlives one store, package or host surface

Plugin Passport separates the identity of the intended plugin/extension/integration component from current store listing, repository/package, host/platform, version and capability context.

PERSISTENT PLUGIN IDENTITY

One Plugin ECZ-ID; changing distribution and host context

Keep the canonical Plugin ECZ-ID stable while supported store listings, repository/package references, host/platform relationships, versions, capabi



WHY THIS MATTERS

A plugin name, repository URL or store listing or package coordinate can be useful, but none alone necessarily provides a durable, provider-neutral identity that counterparties can independently re-check.

WHAT STAYS DURABLE

The Plugin ECZ-ID for the intended plugin/extension/integration component where continuity is appropriate.

WHAT CAN CHANGE

Store listing, repository/package, publisher metadata, host/platform relationship, version, declared capabilities/permissions where supported and evidence.

PLUGIN PASSPORT IN ONE VIEW

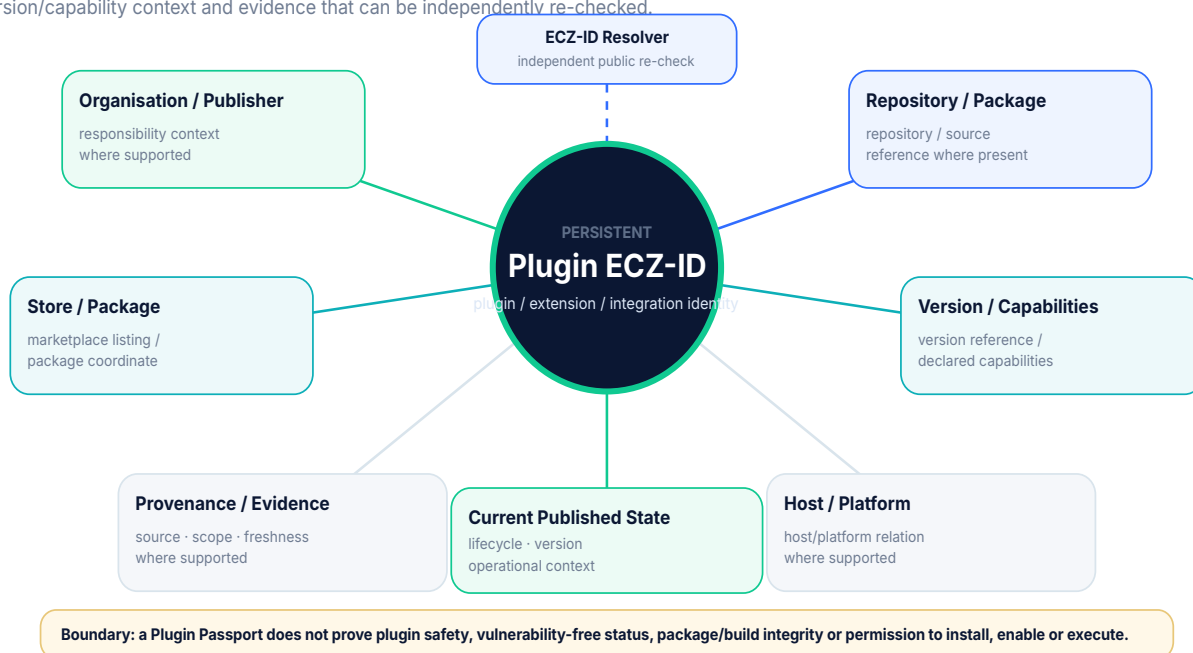
Identity at the centre; publisher, distribution and evidence around it

The Passport can connect a persistent Plugin ECZ-ID to publisher/organisation context, store/repository/package bindings, version/capability context, current state and evidence where present.

PLUGIN PASSPORT PROOF MODEL

What a Plugin Passport can prove and contextualise

A persistent plugin/extension identity with bounded publisher/organisation responsibility, store/repository/package and host/platform bindings, version/capability context and evidence that can be independently re-checked.



INTERPRETATION BOUNDARY

Do not infer plugin safety, vulnerability-free status, cryptographic package provenance, package/build integrity or permission to execute unless the relevant evidence and policy explicitly establish them.

RESOLVER

Public, read-only proof a human or machine can independently re-check.

DASHBOARD

Authenticated customer control for permitted lifecycle and estate actions.

THE CUSTOMER PROBLEM

Plugin identity fragments across stores, packages, hosts and versions

Plugins and extensions can move between marketplaces/stores, repositories/packages, host platforms and integration surfaces. Versions change continuously while the underlying plugin/extension/integration component may remain the same object.

WITHOUT A PERSISTENT IDENTITY

Teams may rely on a plugin name, repository URL, store listing, package coordinate or host-platform reference - each tied to a particular distribution or host context and potentially mutable.

WITH PLUGIN PASSPORT

The plugin can present a stable ECZ-ID with bounded publisher/organisation responsibility, current state, bindings and evidence that counterparties can resolve again.

CORE QUESTION

Which plugin/extension is this, who publishes it, which store/package, host/platform and version does it correspond to, and can I independently verify it?

Plugin Passport complements - rather than replaces - store review, package/signature verification, permissions review, vulnerability management, host/platform controls and runtime policy.

PUBLISHER AND ORGANISATION RESPONSIBILITY

Know who stands behind the plugin - without overclaiming

Where supported, the canonical record can represent organisation/Parent responsibility and publisher context for the plugin/extension object.

RESPONSIBILITY CAN ANSWER

Which organisation claims responsibility for publishing or operating this Plugin identity?

RESPONSIBILITY DOES NOT MEAN

Every version, capability, permission, security property or third-party claim has been independently verified.

SAFE INTERPRETATION

Read the published relationship and its supporting evidence. Do not promote a responsibility relationship into universal assurance.

PUBLISHER CHANGES

A maintainer or publisher relationship can change while the canonical plugin/extension identity remains stable where appropriate.

PARENT CONTEXT

Links the Plugin Passport to the responsible organisation/Parent where present.

EVIDENCE

Supports bounded claims according to source, scope and freshness.

RE-CHECK

Resolve again when publisher/organisation responsibility matters to the decision.

STORE, PACKAGE, HOST AND VERSION BINDINGS

Stable identity; changing distribution and host surfaces

Bindings connect the canonical Plugin ECZ-ID to supported external references such as marketplace/store listings, repositories/packages, host/platform relationships, versions and declared permissions/capabilities where supported.

REPOSITORY

Source repository or project reference where supported.

MARKETPLACE/STORE OR PACKAGE SURFACE

Store listing / package namespace / package coordinate.

VERSION

Plugin/extension version reference where supported.

HOST / PLATFORM

Host/platform relationship where supported.

PERMISSIONS / CAPABILITIES

Declared permissions/capabilities where supported and evidenced.

EVIDENCE LINK

Reference that supports a bounded claim about the plugin/extension identity.

BINDING LAW

External store, package, host and version identifiers are bindings/context. They do not replace ECZ-ID as the canonical identity authority root.

VERSION, PERMISSIONS, STATE, EVIDENCE AND FRESHNESS

Current proof should be treated as point-in-time

Plugin identity may persist while versions, store listings, repositories/packages, host integrations, declared permissions/capabilities and evidence change.

IDENTITY PERSISTS

The ECZ-ID can remain stable across supported store, package, host/platform and version changes when the underlying plugin/extension object remains the same.

CONTEXT CHANGES

Published version/lifecycle state, bindings and evidence should be re-checked when freshness matters.

PROVENANCE BOUNDARY

Repository or version evidence may support provenance/context, but do not infer cryptographic package/build provenance, signature validity or reproducibility unless the record and evidence explicitly establish it.

GOOD CONSUMER BEHAVIOUR

Read source, scope and freshness; re-resolve before consequential adoption or upgrade decisions.

UNSAFE SHORTCUT

Assuming an old version, host, store binding or publisher context is still current because the ECZ-ID itself is unchanged.

IDENTITY IS NOT PLUGIN SAFETY, HOST APPROVAL OR EXECUTION AUTHORITY

Use the Passport before the consequence - not instead of plugin/extension supply-chain controls

Plugin Passport can help establish which plugin/extension is intended and what context supports that identity. It does not make the plugin safe or authorised for your environment.

IDENTITY

Which plugin/extension object is this?

RESPONSIBILITY

Who claims publication/responsibility?

LOCAL POLICY

Is this plugin/version permitted in this environment?

SECURITY CONTROLS

Are permissions, signing, vulnerability, host-policy and runtime requirements satisfied?

BOUNDARY

A valid Passport is not permission to install, enable, load or execute the plugin. Apply package/signature verification, vulnerability, permissions, change-control and runtime policy separately.

HIGH-VALUE USE CASES

Where persistent, re-checkable plugin/extension identity helps most

The strongest use cases are those in which distribution and host context changes while publisher accountability and identity continuity still matter.

REPOSITORY MIGRATION

Preserve plugin/extension identity when source hosting moves.

MARKETPLACE/STORE OR PACKAGE MIGRATION

Keep the same ECZ-ID while supported store/package coordinates change.

VERSION VERIFICATION

Connect a stable plugin/extension identity to current version context.

ENTERPRISE EXTENSION REVIEW

Give architecture/security teams a re-checkable identity anchor for a plugin/extension.

MACHINE CONSUMPTION

Allow automation to resolve bounded plugin/extension identity/context before plugin adoption or upgrade decisions.

CROSS-STORE / CROSS-PLATFORM DISTRIBUTION

Treat store, host and provider-native plugin IDs as bindings rather than the authority root.

VALUE

The outcome is identity continuity plus current, independently re-checkable context - not a universal software-security score.

EXAMPLE - MARKETPLACE/STORE OR PACKAGE MIGRATION

One plugin/extension identity through a store/repository/package move

A publisher moves a Plugin from one marketplace/store or store/package surface to another while preserving the same underlying plugin/extension/integration component.

BEFORE

Consumers know the Plugin by ECZ-ID plus store, repository, package and version bindings.

AFTER

The Plugin ECZ-ID remains stable; supported store/host/package bindings and evidence are updated and Resolver reflects the current record.

WHAT CHANGED

Distribution context, not necessarily the canonical plugin/extension identity.

CONSUMER ACTION

Re-resolve before adopting the new distribution location or version.

SECURITY ACTION

Validate package signatures, vulnerabilities, host/organisation approval and local approval independently.

DEPLOYMENT FIT

Who should use a Plugin Passport?

Adopt where persistent plugin/extension identity and independent re-checkability are valuable across a changing plugin/extension supply chain.

STRONG FIT

Plugins/extensions distributed across stores, repositories/packages, versions or providers where durable identity matters.

SECURITY FIT

Teams that need a stable object identity before applying separate plugin/extension supply-chain controls.

DEVELOPER FIT

Tooling that needs a machine-resolvable plugin/extension identity independent of one package coordinate.

LOWER-VALUE FIT

Single-use or ephemeral plugin/extension artifacts where persistent cross-context identity brings little benefit.

FREE IDENTITY PRIMITIVE

The Plugin Passport itself is a free identity primitive. Paid ECZ-ID capabilities are not prerequisites for basic Passport identity and Resolver verification.

HOW TO VERIFY A PLUGIN PASSPORT

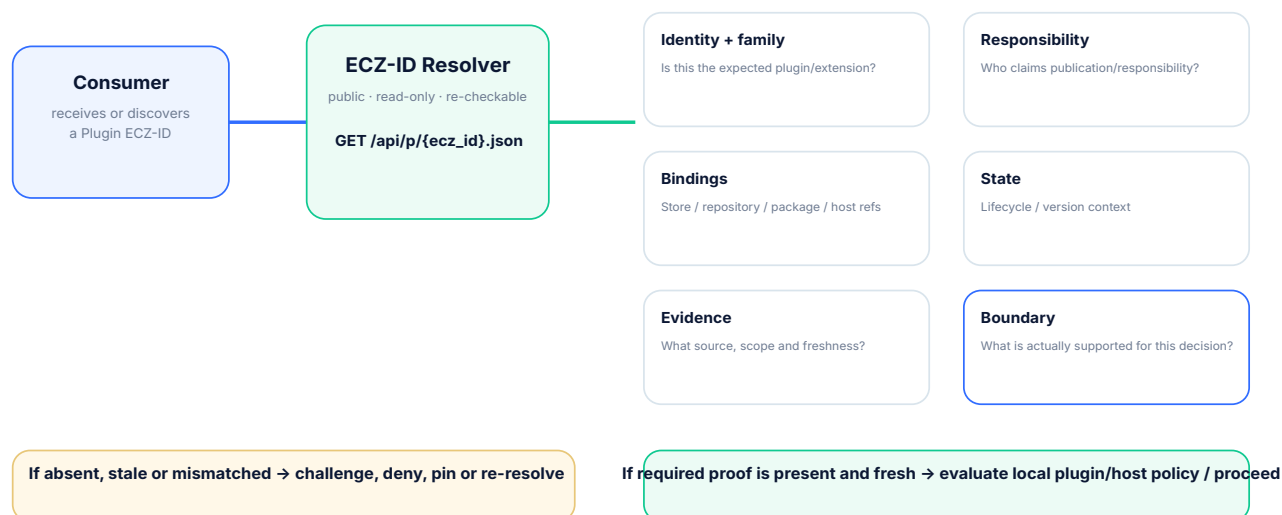
Resolve, inspect, interpret, re-check

Verification is intentionally bounded: identify the plugin/extension object, inspect publisher/organisation responsibility and current distribution and host context, then apply your own policy for adoption.

RESOLVER RE-CHECK

Turn a presented Plugin 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.



MACHINE-READABLE ROUTE

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

IF REQUIRED PROOF IS PRESENT

Evaluate the decision using the returned state, bindings and evidence plus your local plugin/host policy.

IF REQUIRED PROOF IS ABSENT

Treat it as not established; challenge, pin, deny or seek additional evidence rather than infer.

DECISION CARD

The Plugin Passport in 60 seconds

Use this page as the executive summary before acquisition or architecture review.

OBJECT

Plugin / extension / integration component.

PERSISTENT IDENTITY

One canonical ECZ-ID can survive store, repository/package and version changes where object continuity holds.

RESPONSIBILITY

Organisation/Parent and publisher context where present.

BINDINGS

Store, repository/package, docs, version and provider references where supported.

EVIDENCE + STATE

Bounded, point-in-time context; source, scope and freshness matter.

BOUNDARY

Not plugin safety, vulnerability-free status, package/build attestation, host approval or execution permission.

NEXT ACTION

If the model fits your plugin/extension supply chain, acquire the free Plugin Passport through AWS Marketplace or the supported ECZ-ID acquisition surface, then verify the resulting record in Resolver.