

ECZ-ID Service & Workload Passport Overview and Customer Guide

Persistent, resolver-verifiable identity for services and workloads across changing cloud providers, accounts/projects/subscriptions, clusters, namespaces, runtimes and deployment environments.

SERVICE & WORKLOAD PASSPORT · OVERVIEW

PRIMARY READER

Buyer, technical evaluator, security architect, platform/service owner.

SECONDARY READER

Platform engineers, cloud architects, security teams, operations, governance and procurement.

WHEN TO READ THIS

You are deciding whether a service/workload needs durable identity and independently re-checkable owner/operator, provider, runtime and deployment context.

RELEASE

2026.10 · 16 October 2026

WHAT YOU WILL KNOW AFTER READING

You can explain the service/workload object, organisation/owner/operator responsibility, provider/runtime/deployment/service bindings, state, evidence and safe verification boundary.

WHAT THIS DOCUMENT DOES NOT ESTABLISH

A Service & Workload Passport does not by itself prove workload safety, runtime/image/software integrity, configuration correctness, availability or permission to perform an action.

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

EXECUTIVE EXPLANATION

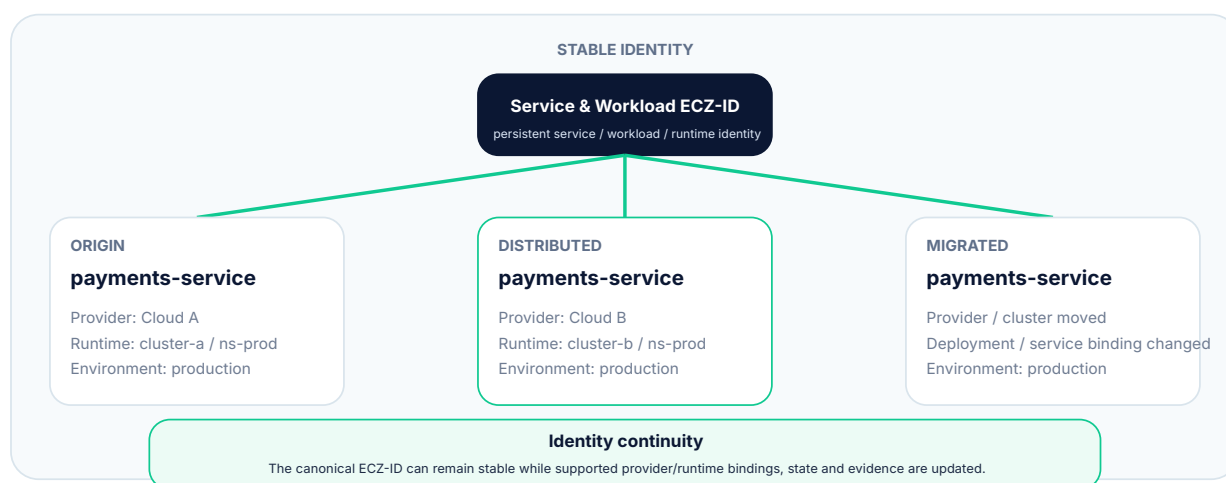
A durable identity anchor for services and workloads across changing provider and runtime context

Service & Workload Passport separates the identity of the intended service/workload/runtime object from current provider-native workload IDs, cloud account/project/subscription references, runtime/deployment/service bindings, namespace/service-account/container references, environment and current state.

PERSISTENT SERVICE / WORKLOAD IDENTITY

One Service & Workload ECZ-ID; changing provider and runtime context

Keep the canonical Service & Workload ECZ-ID stable while supported provider-native workload IDs, cloud account/project references, clusters, namespaces, runtime/deployment/service bindings, environment, state and evidence change.



WHY THIS MATTERS

A service name, provider-native workload ID, cloud resource ID, namespace/service-account/container reference or runtime endpoint can be useful, but none alone necessarily provides a durable, provider-neutral identity that counterparties can independently re-check.

WHAT STAYS DURABLE

The Service & Workload ECZ-ID for the intended service/workload/runtime object where continuity is appropriate.

WHAT CAN CHANGE

Provider-native workload IDs, cloud account/project/subscription references, clusters/namespaces/service accounts, runtime/deployment/service bindings, environment, current state and evidence.

SERVICE & WORKLOAD PASSPORT IN ONE VIEW

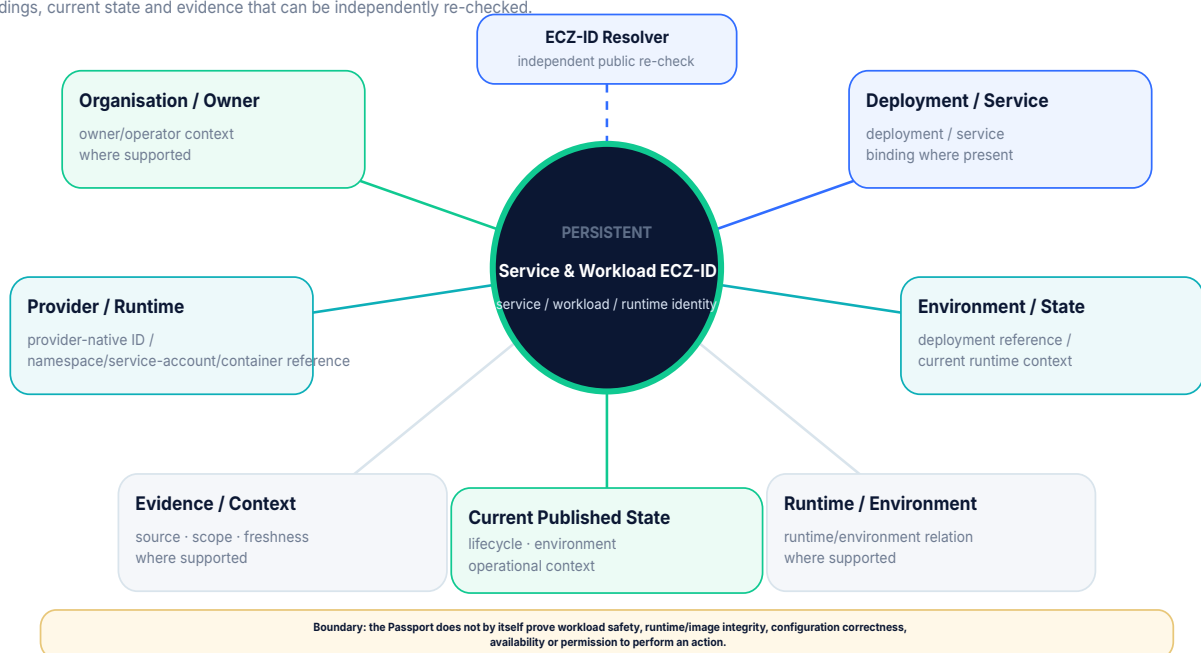
Identity at the centre; responsibility, provider/runtime context and evidence around it

The Passport can connect a persistent Service & Workload ECZ-ID to organisation/owner/operator context, provider/runtime/deployment bindings, runtime/environment context, current state and evidence where present.

SERVICE & WORKLOAD PASSPORT PROOF MODEL

What a Service & Workload Passport can prove and contextualise

A persistent service/workload identity with bounded organisation/owner/operator responsibility, provider/runtime/deployment bindings, current state and evidence that can be independently re-checked.

**INTERPRETATION BOUNDARY**

Do not infer workload safety, configuration correctness, image/runtime integrity, availability or permission to act 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

Service/workload identity fragments across providers, accounts, clusters, namespaces, runtimes and deployments

Services and workloads can move between cloud providers, accounts/projects/subscriptions, clusters, namespaces, service accounts, containers and runtime environments while the underlying service/workload identity remains the same.

WITHOUT A PERSISTENT IDENTITY

Teams may rely on a service name, provider-native workload ID, cloud resource ID, cluster/namespaces reference, service account, container/workload ID or runtime endpoint - each tied to one provider/runtime context and potentially mutable.

WITH SERVICE & WORKLOAD PASSPORT

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

CORE QUESTION

Which service/workload is this, who is responsible for it, what provider/runtime/deployment bindings support it, and can counterparties independently re-check the canonical ECZ-ID?

Service & Workload Passport complements - rather than replaces - provider/platform review, container/image integrity verification, IAM/policy review, vulnerability management, runtime/environment controls and runtime policy.

ORGANISATION, OWNER AND OPERATOR RESPONSIBILITY

Know who is responsible for the service/workload - without overclaiming

Where supported, the canonical record can represent organisation/Parent responsibility and owner/operator context for the service/workload object.

RESPONSIBILITY CAN ANSWER

Which organisation, owner or operator claims responsibility for this service/workload identity?

RESPONSIBILITY DOES NOT MEAN

Every deployment, authority, 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.

OWNER/OPERATOR CHANGES

Owner/operator responsibility can change while the canonical service/workload identity remains stable where the underlying object remains the same.

PARENT CONTEXT

Links the Service & Workload Passport to the responsible organisation/Parent where present.

EVIDENCE

Supports bounded claims according to source, scope and freshness.

RE-CHECK

Resolve again when organisation/owner/operator responsibility matters to the decision.

PROVIDER, RUNTIME, DEPLOYMENT AND SERVICE BINDINGS

Stable identity; changing provider and runtime context

Bindings connect the canonical Service & Workload ECZ-ID to supported provider-native workload IDs, cloud account/project/subscription references, clusters, namespaces, service accounts, containers, runtime/deployment/service references and other supported workload identifiers.

CLOUD CONTEXT

Cloud account, project or subscription reference where supported.

RUNTIME IDENTITY

Provider-native workload ID, namespace, service account, container or workload reference where supported.

DEPLOYMENT

Deployment, service or runtime reference where supported.

ENVIRONMENT

Cluster, namespace, runtime or environment relationship where supported.

AUTHORITY CONTEXT

Supported authority/delegation or policy context where present and evidenced.

EVIDENCE LINK

Reference that supports a bounded claim about the service/workload identity.

BINDING LAW

Provider-native workload IDs and runtime/deployment/service identifiers are bindings/context. They do not replace ECZ-ID as the canonical identity authority root.

RUNTIME, STATE, EVIDENCE AND FRESHNESS

Current proof should be treated as point-in-time

Service/workload identity may persist while provider-native workload IDs, accounts/projects/subscriptions, clusters, namespaces, service accounts, runtime/deployment/service bindings, current state and evidence change.

IDENTITY PERSISTS

The ECZ-ID can remain stable across supported provider, account/project, cluster, namespace, runtime and deployment changes when the underlying service/workload object remains the same.

CONTEXT CHANGES

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

PROVENANCE BOUNDARY

Runtime, deployment or provider evidence may support bounded provenance/context, but do not infer cryptographic image/build provenance or runtime integrity 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 deployment, runtime, provider binding or owner/operator context is still current because the ECZ-ID itself is unchanged.

IDENTITY IS NOT WORKLOAD SAFETY, RUNTIME APPROVAL OR EXECUTION AUTHORITY

Use the Passport before the consequence - not instead of IAM, runtime, deployment or security controls

Service & Workload Passport can help establish which service/workload is intended and what context supports that identity. It does not make the workload safe or authorised for your environment.

IDENTITY

Which service/workload object is this?

RESPONSIBILITY

Who claims publication/responsibility?

LOCAL POLICY

Is this workload/deployment permitted in this environment?

SECURITY CONTROLS

Are authority/policy context, signing, vulnerability, runtime-policy and runtime requirements satisfied?

BOUNDARY

A valid Passport is not authority or policy context to deploy, start, invoke, modify or otherwise operate the workload. Apply container/image integrity verification, vulnerability, authority/policy context, change-control and runtime policy separately.

HIGH-VALUE USE CASES

Where persistent, re-checkable service/workload identity helps most

The strongest use cases are those in which provider and runtime context changes while owner/operator accountability and identity continuity still matter.

PROVIDER MIGRATION

Preserve service/workload identity when cloud/provider context changes.

CLUSTER/RUNTIME MIGRATION

Keep the same ECZ-ID while supported cluster, namespace, runtime or provider bindings change.

DEPLOYMENT VERIFICATION

Connect a stable service/workload identity to current runtime/deployment context.

ENTERPRISE WORKLOAD REVIEW

Give architecture/security teams a re-checkable identity anchor for a service/workload.

MACHINE CONSUMPTION

Allow automation to resolve bounded service/workload identity/context before service/workload deployment or migration decisions.

CROSS-PROVIDER / CROSS-RUNTIME PORTABILITY

Treat provider, runtime and provider-native workload 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 - PROVIDER / CLUSTER / RUNTIME MIGRATION

One service/workload identity through a provider/runtime/deployment move

An operator moves a service/workload from one cloud/provider or runtime environment to another while preserving the same underlying service/workload/runtime object.

BEFORE

Consumers know the service/workload by ECZ-ID plus provider, account/project, cluster/namespace, runtime and deployment bindings.

AFTER

The Service & Workload ECZ-ID remains stable; supported provider/runtime/deployment bindings and evidence are updated and Resolver reflects the current record.

WHAT CHANGED

Provider/runtime/deployment context, not necessarily the canonical service/workload identity.

CONSUMER ACTION

Re-resolve before relying on the new provider/runtime/deployment context.

SECURITY ACTION

Validate workload authentication, IAM/authorization, container/image integrity, vulnerability/configuration and local runtime approval independently.

DEPLOYMENT FIT

Who should use a Service & Workload Passport?

Adopt where persistent service/workload identity and independent re-checkability are valuable across changing cloud, cluster, runtime and deployment context.

STRONG FIT

Services/workloads that move across providers, accounts/projects, clusters, namespaces, runtimes or deployments where durable identity matters.

SECURITY FIT

Teams that need a stable object identity before applying separate IAM, workload-security and runtime controls.

DEVELOPER FIT

Tooling that needs a machine-resolvable service/workload identity independent of one namespace/service-account/container reference.

LOWER-VALUE FIT

Single-use or ephemeral service/runtime/deployment references where persistent cross-context identity brings little benefit.

FREE IDENTITY PRIMITIVE

The Service & Workload 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 SERVICE & WORKLOAD PASSPORT

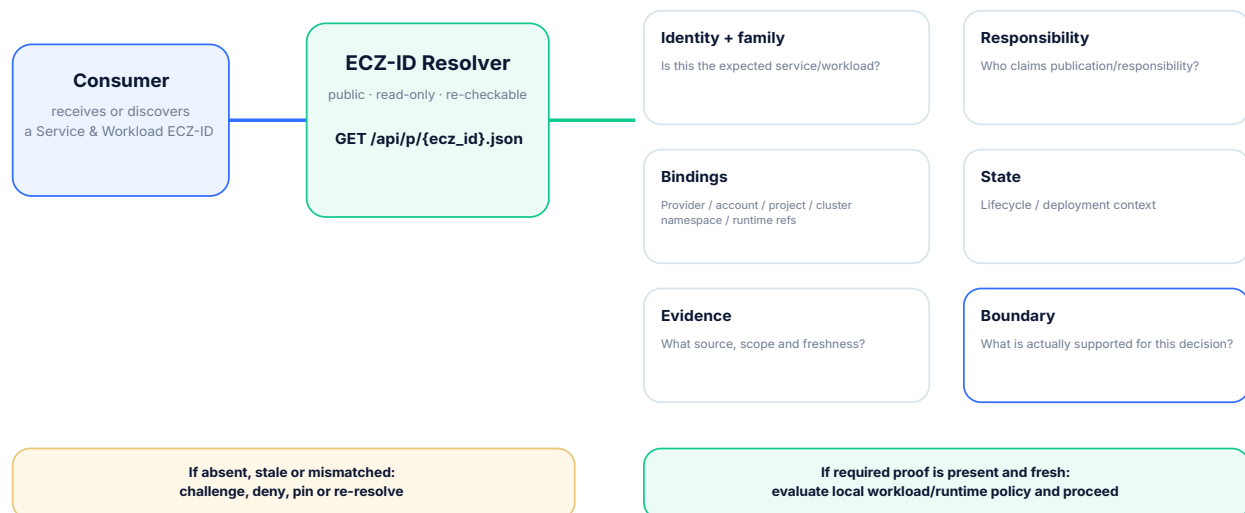
Resolve, inspect, interpret, re-check

Verification is intentionally bounded: identify the service/workload object, inspect organisation/owner/operator responsibility and current provider/runtime/deployment context, then apply your own policy for the intended action.

RESOLVER RE-CHECK

Turn a Service & Workload 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 workload/runtime policy.

IF REQUIRED PROOF IS ABSENT

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

DECISION CARD

The Service & Workload Passport in 60 seconds

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

OBJECT

Service / workload / runtime identity.

PERSISTENT IDENTITY

One canonical ECZ-ID can survive provider/runtime/deployment reference and deployment changes where object continuity holds.

RESPONSIBILITY

Organisation/Parent and owner/operator context where present.

BINDINGS

Provider-native workload IDs, cloud account/project/subscription, cluster/namespace/service-account/container, runtime/deployment/service and documentation references where supported.

EVIDENCE + STATE

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

BOUNDARY

Not workload safety, configuration correctness, runtime/image integrity, availability or permission to perform an action.

NEXT ACTION

If the model fits your service/workload estate, acquire the free Service & Workload Passport through AWS Marketplace or the supported ECZ-ID acquisition surface, then verify the resulting record in Resolver.