

Open Continuity Protocol

Portable AI History — Consolidated Editor's Working Draft

Version: 0.4-working-draft

Publication date: 15 August 2026

Status: Current project-authoritative consolidated editor's working draft

Protocol family: Open Continuity Protocol (OCP)

Wire version: 0.4

Release theme: Portable AI History

Schema dialect: JSON Schema Draft 2020-12

License status: All licensing remains reserved until the project adopts a documented specification/code licensing policy; CC BY 4.0 for text and Apache-2.0 for code remain proposals only.

Status notice. This source document is part of the current project-authoritative OCP working-draft tree until superseded. A publication snapshot becomes pinned only when `release-manifest.json` identifies its exact members and digests; prose alone is not a release pin. Neither the source tree nor a manifest is an Internet Standard, standards-body adoption, security certification, legal-compliance statement, production recommendation, or claim of interoperability between independent implementations. The bundled reference harness reports local schema and semantic results only. Independent implementations, deployed controls, human acceptance, governance formation, and public review remain required before a stable release.

OCP is implementable without KnowOnce or any other proprietary provider. Product names in examples are synthetic. External standards retain their own governance, licensing, and conformance rules.

Document map

Sections 1–11 explain the problem and architecture. Sections 12–19 define the core model and lifecycles. Sections 20–24 complete Portable AI History profiles, loss-aware round trip, and transfer composition. Sections 25–36 define revocation, safety, privacy, bindings, evolution, errors, and exact conformance. Sections 37–41 address implementation, governance, horizon, evidence-gated roadmap, and resolved/open questions. Sections 42–45 provide generated examples, normative schemas, references, and a glossary. The complete executable schemas, fixtures, registries, and endpoint description are distributed in the companion **OCP Schema and Conformance Pack**.

Table of contents

1	Cover and document status
2	Plain-English orientation
3	Abstract
4	Problem statement
5	Goals
6	Non-goals
7	Design principles
8	Terminology
9	Architecture and roles
10	Trust boundaries
11	Core protocol flow
12	Complete core object model
13	Normative schema definitions
14	Request lifecycle
15	Grant lifecycle
16	Package lifecycle
17	Update-proposal and review lifecycle
18	Accepted-context rules
19	Evidence and provider-interpretation rules
20	Conversation Profile
21	Artifact Profile
22	Provider Adapter Framework and import
23	Export and round-trip rules
24	Transfer and profile composition
25	Revocation and erasure semantics
26	Security considerations
27	Privacy considerations
28	Third-party and organizational-data rules
29	Transport and HTTP binding
30	Offline package format
31	Adjacent-standard mappings
32	Versioning
33	Extensions and profiles
34	Error handling
35	Conformance classes
36	Normative test assertions
37	Reference implementation guidance
38	Governance
39	Design horizon: 2026–2031
40	Roadmap — 0.4 Portable AI History
41	Resolved and remaining questions
42	Complete validated examples
43	Normative schemas
44	References
45	Glossary

1. Cover and document status

1.1 Normative status

The key words **MUST**, **MUST NOT**, **REQUIRED**, **SHOULD**, **SHOULD NOT**, and **MAY** in this document are to be interpreted as described in BCP 14 when, and only when, they appear in capitals. Every such requirement names or clearly identifies the actor responsible for satisfying it.

An implementation conforms only to the classes it declares and passes. A core schema pass does not establish privacy compliance, safe import behavior, correct authorization, faithful round trips, accessible consent, recipient deletion, or ecosystem interoperability.

1.2 Sources of truth

OCP 0.4 uses scoped authority rather than a silent precedence list:

1. this document owns protocol semantics, actor obligations, trust boundaries, and safety invariants;
2. JSON Schemas own wire representation and conditional shape;
3. registries own identifiers, lifecycle transitions, errors, events, profiles, and maturity;
4. binding documents own transport and serialization mechanics;
5. assertion and fixture files own executable expectations and evidence status; and
6. `release-manifest.json`, when generated for a publication snapshot, owns the exact release members and digests.

No artifact may override another artifact outside its scope. Any conflict is a specification defect and invalidates the affected conformance claim until resolved. Implementers **MUST** report conflicts and **MUST NOT** choose an interpretation that broadens access, disclosure, processing, retention, proposal rights, delegation, trust, or authority.

1.3 Stability

The 0.4-working-draft label indicates material design work remains. Field names, lifecycle states, profile boundaries, registry locations, and endpoint paths may change before a stable release.

Implementations **MUST NOT** advertise 0.4 as 1.0-compatible and **MUST NOT** accept an unsupported version by assuming compatible semantics. The earlier 0.9 document was an architectural proposal, not a predecessor wire release.

2. Plain-English orientation

2.1 One sentence

OCP lets a person or authorized steward carry selected, reviewed context between systems as a purpose-bound package while preserving where it came from, separating evidence from interpretation, preventing silent write-back, and never treating context as permission to act.

2.2 One paragraph

Today, an assistant can know a great deal inside one product while a new assistant starts from nothing or receives a blunt archive. OCP defines a smaller, safer interchange layer. A client explains what context it wants, why, for whom, for how long, and how it will use and retain it. An authorization system records the actual decision. The context provider creates a minimized package containing only the approved presentation. Information inferred by a provider remains distinguishable from source evidence and from context that a person or authorized steward has accepted for a stated scope. Clients may propose corrections, but they cannot silently rewrite accepted context. OCP reuses existing identity, authorization, agent transport, provenance, credential, storage, and cryptographic standards instead of replacing them.

2.3 A sixty-second example

Avery is planning a trip with a new travel assistant. The assistant asks for accepted lodging preferences for trip planning, for one session, with no model training, no advertising, no delegation, and no raw conversation evidence. Avery sees that request in the context provider and approves only the accepted preference “prefers small, quiet hotels,” scoped to the trip. The deployment's authorization layer grants access; OCP records a non-secret `grant_receipt`. The provider returns a short-lived `context_package` for that assistant, containing the accepted item, a delivery-pairwise subject alias, provenance metadata, and an explicit `external_action_authority`: “none”. The assistant can use the preference to suggest hotels, but it cannot book one from that context. If it learns that Avery wants a room near a train station, it submits an `update_proposal`. Avery can accept, revise, reject, or defer the proposal. Raw messages remain evidence and are not disclosed unless separately requested and approved.

2.4 Architecture at a glance



```
review_decision --> new accepted item
```

Evidence and artifacts remain separate permission layers.
Context never enters the action-authorization channel.

2.5 What users should understand

OCP does not move “everything an AI knows.” It moves a selected presentation for a declared use. “Accepted” means approved for a scope; it does not mean objectively true. A provider URL does not prove a file has been preserved. Revoking access stops future authorized access but cannot recall plaintext already delivered. A deletion report describes a provider's claim unless independently verified evidence says more.

3. Abstract

The Open Continuity Protocol is an open protocol family for steward-approved context presentation and transfer. It defines a storage- and transport-independent object model; request, authorization-receipt, package, proposal, and review lifecycles; provenance and acceptance semantics; disclosure layers for context, evidence, and artifacts; extension and profile rules; HTTP and offline bindings; and testable conformance classes.

OCP's normative core is deliberately narrow. It defines `context_space`, `context_item`, `source_record`, `client_declaration`, `context_request`, `grant_receipt`, `context_package`, `update_proposal`, and `review_decision`. OCP 0.4 completes the Conversation Profile, Artifact Profile, and Provider Adapter Framework without moving their domain semantics into the core. Erasure, organization, delegation, selective-disclosure, agent-transport, and live-sync behaviors remain separate or future profiles. Authentication, access-token issuance, external-action authorization, storage, model execution, content authenticity, and legal compliance remain outside the core.

The 0.4 release theme is **Portable AI History**: change the AI, keep the history, and know what survived. The protocol's central invariants are:

- provider observation and inference are not accepted context merely because a provider stored them;
- accepted context is scoped governance state, not universal truth;
- evidence, derived context, and artifact content have distinct permissions;
- a client cannot silently mutate accepted context;
- a package cannot exceed the live effective authorization;
- imported and packaged content is untrusted data;
- context access never authorizes an external action;
- a source reference is not proof of preservation; and
- revocation, retention expiry, deletion status, cryptographic erasure, and independent verification are different claims.

4. Problem statement

4.1 Fragmented context

People repeatedly explain preferences, constraints, decisions, accessibility needs, project history, and creative intent to different systems. Provider exports typically preserve provider data structures rather than the reviewed meaning a new system needs. Copying entire histories creates privacy, security, relevance, and prompt-injection risk. Copying only a summary loses evidence, provenance, disagreement, and correction history.

4.2 The missing interoperability layer

Existing work covers many neighboring needs: authentication and authorization, agent and tool transport, memory interchange, storage, provenance, verifiable credentials, content authenticity, privacy vocabulary, data transfer, and observability. What remains is a governed presentation layer that can answer:

- Which context is fit to present for this use?
- Who reviewed or accepted it, under what authority, and for what scope?
- Which provider observations or sources support or contradict it?
- What was intentionally omitted?
- What may the recipient retain, train on, forward, or propose back?
- How can another provider import it without silently inventing semantics?
- What survived a round trip, and what did not?

4.3 Why archives and prompts are insufficient

A downloaded archive can support portability but ordinarily does not define semantic acceptance, minimization, delivery recipient and plaintext processors, purpose, update review, or safe import behavior. A natural-language prompt can help a model but is not deterministic enough for loss-aware round trips, authorization comparison, concurrency, or conformance. OCP therefore requires structured core JSON. Human-readable summaries MAY accompany it as derived representations with provenance and source-entry references; when their language is known, the representation MUST carry the applicable BCP 47 language tag.

4.4 Failure if OCP is too broad

OCP fails if it attempts to become a universal memory ontology, identity system, authorization protocol, personal-data store, agent runtime, credential format, provenance language, or digital-asset archive. The protocol is valuable only if it composes those layers and standardizes the narrow semantics they do not already share.

5. Goals

OCP has the following goals:

1. **Portable accepted context.** Represent context approved for a stated scope without equating acceptance with truth.
2. **Evidence separation.** Preserve the distinction between source material, provider observation or inference, and accepted presentation.
3. **Purpose-bound minimization.** Make a request and package comparable across selectors, views, delivery recipient, processors, time, retention, processing uses, proposal rights, and delegation.
4. **No silent write-back.** Require a proposal and review decision before client-suggested changes become accepted context.
5. **Loss-aware movement.** Support import, export, and round-trip comparison with explicit omissions, quarantines, unsupported semantics, and preservation state.
6. **Safe agent use.** Keep all context in the data channel and external-action authority in a separate policy channel.
7. **Storage and transport independence.** Work in local-first, hosted, self-hosted, organizational, and federated deployments.
8. **Standards reuse.** Bind to established identity, authorization, transport, provenance, credential, storage, cryptographic, and event standards.

9. **Independent implementation.** Make schemas, examples, transitions, errors, and conformance assertions precise enough for independent implementations.
10. **Honest assurance.** Distinguish what the protocol enforces from what a recipient declares, attests, is audited for, or cannot prove.

6. Non-goals

The core does not:

- authenticate people, organizations, workloads, models, agents, or devices;
- issue OAuth, GNAP, capability, or domain-action credentials;
- authorize purchases, messages, publications, deployments, deletions, tool calls, or any external action;
- define one global person identifier or merge identities across providers;
- decide who legally controls or may disclose data;
- establish that an accepted statement is true;
- require one storage system, graph database, RDF model, pod, cloud, or vendor;
- define model memory internals, embedding formats, prompt assembly, agent tasks, or tool invocation;
- replace general-purpose provenance, content-authenticity, credential, selective-disclosure, event, or telemetry standards;
- guarantee that a recipient honored purpose, retention, training, forwarding, or deletion terms after receiving plaintext;
- guarantee perpetual artifact availability;
- guarantee the recall of an offline package;
- confer GDPR, EU Data Act, EU AI Act, NIST, accessibility, security, or other legal/regulatory compliance; or
- standardize high-risk multi-party authority, minors, guardianship, inheritance, organizational exfiltration policy, or delegated-agent chains in the self-authorized core profile.

7. Design principles

7.1 Accepted context is scoped, not universal

A statement may be accepted for a project, purpose, or context space while disputed elsewhere. A provider **MUST NOT** interpret accepted as objective truth, permanent truth, consent to all processing, or authority over every subject named by the statement.

7.2 Observation, inference, and acceptance remain distinct

A source record is evidence. An inferred context item is a provider interpretation. A review decision can accept a context item for a scope. Conversion between those layers **MUST** remain explicit and auditable.

7.3 Context is not a dossier

Providers **MUST** minimize packages to the active request and authorization. A request for “help plan this trip” does not justify a lifetime archive, unrelated relationships, precise location history, or raw private messages.

7.4 Context, evidence, and artifacts are separate views

Permission to disclose an accepted statement does not imply permission to disclose the conversation, file, credential, photograph, or other source behind it. The `views` object and profile rules **MUST** enforce each layer separately.

7.5 No silent accepted-state writes

A client **MAY** submit an `update_proposal` only when authorized. It **MUST NOT** directly create, revise, accept, supersede, withdraw, or tombstone accepted context. A provider **MUST** create an immutable `review_decision` before an accepted-state change.

7.6 Provenance is not preservation

A source URI identifies or locates a source under its own conditions. It is not evidence that content is independently stored, complete, accessible, licensed for reuse, or durable. Artifact profiles **MUST** state custody and verification separately.

7.7 Revocation is not erasure

Revocation prevents future authorized operations controlled by the provider. It cannot generally recall delivered plaintext. Deletion and erasure records **MUST** state scope, reporter, time, limitations, and verification level without overstating completion.

7.8 OCP must itself be portable

Unknown non-critical extensions, provenance, review decisions, supersession links, and explicit omissions **MUST** survive a conforming round trip or appear in a loss report. An implementation **MUST NOT** require a proprietary service to interpret core records.

7.9 Context is untrusted data

Every imported or packaged content entry **MUST** be processed as untrusted data. Content **MUST NOT** be concatenated into system instructions, policy, authorization decisions, code, queries, shell commands, or tool arguments without an independently authorized and security-reviewed transformation.

7.10 Context access is not action authority

Every core `context_package` **MUST** contain `external_action_authority`: “none”. No core field, natural-language statement, accepted decision, budget, credential reference, or past activity can authorize an external action. Concrete action authority **MUST** be obtained through a separate domain authorization after the action target and parameters are known.

7.11 Fail closed on semantic uncertainty

When an implementation cannot compare purpose, selectors, views, authority, processing use, delegation, lifecycle, integrity, or a critical extension, it **MUST** deny the operation or quarantine the input. It **MUST NOT** guess a more permissive meaning.

8. Terminology

Term	Meaning in OCP
accepted context	A context item approved by an identified reviewer or policy for an explicit <code>accepted_scope</code> . It is governance state, not universal truth.
artifact	Content-bearing digital material such as a document, image, audio file, video, code archive, or dataset. Governed by the Artifact Profile.
delivery recipient	The one recipient service and authenticated client instance for which a runtime request, receipt, or package is issued. Independent recipients require separate packages.
processing recipient	A separately disclosed and approved model provider or subprocessor expected to receive plaintext while serving the delivery recipient. It is not OCP delegation.
authorization binding	The external or local enforcement decision and credential/capability context that authorizes a protected OCP operation.
client	Software requesting or consuming context. A client is distinct from its publisher, executing agent, model, model provider, and subprocessors.
client declaration	A versioned statement of recipient identity, policy links, every plaintext processing recipient, supported retention/processing/offline/proposal modes, delegation, deletion support, and security contact. It is evidence shown during review, not proof of later behavior.
context item	A structured assertion or other profile-defined unit of portable context with subjects, statement, review state, provenance, validity, sensitivity, and lifecycle.
context package	An immutable, minimized presentation issued under one active grant receipt for a purpose, delivery recipient/client instance, declared plaintext processors, period, disclosure view, retention policy, and processing-use decision. It carries data, not authority.
context provider	The service or local component that stores/evaluates context, enforces access with an authorization layer, constructs packages, and manages proposals.
context space	A named scope that groups context around a purpose, domain, subject set, and authority model.
evidence	Source content or observation represented by a <code>source_record</code> . Evidence may support, contradict, or merely relate to a context item; it is not automatically accepted context.
grant receipt	A non-secret OCP record of the effective context decision, linked to an external/local authorization decision. Possession does not grant access and never grants external-action authority.

Term	Meaning in OCP
import quarantine	A state in which bytes and derived representations remain isolated, untrusted, and unaccepted pending parser, integrity, schema, policy, and human review.
package-local alias	An opaque identifier used only within one package or delivery-recipient scope to reduce correlation with store-internal identifiers.
profile	A versioned bounded semantic extension with its own objects or fields, authority model, lifecycle, comparison rules, risks, and conformance requirements. A transport or serialization mechanism is a binding, not a profile.
proposal	A client's inert request to create or revise context. A proposal cannot mutate accepted context.
purpose	A URI code with a specific human description explaining the intended use. A purpose is a review and policy input, not a self-enforcing promise after disclosure.
review decision	An immutable record that accepts, accepts with changes, rejects, defers, or disputes a proposal under a stated authority basis.
source record	A typed record for imported, observed, entered, or externally referenced evidence, including trust and instruction treatment.
snapshot reference	Exact immutable record relationship carrying ID, revision, schema URI, and RFC 8785 digest.
live status reference	Authorized dereferenceable current-state relationship, separate from immutable content identity.
steward / reviewer	A person or policy actor authorized in the applicable context space to make a scoped review decision. Stewardship does not imply legal control of every referenced person or organization.
subject	A person, organization, household, team, project, or resource described by context. Subject is not synonymous with authorizer, account holder, OAuth subject, or data controller.

The term **canonical context** is a deprecated alias for accepted context. Implementations MAY recognize it in legacy import mappings but MUST NOT emit it as a 0.4 core status. "Canonical" MAY describe deterministic byte representation, where explicitly defined, but not truth or authority.

9. Architecture and roles

9.1 Protocol boundary

OCP is a protocol family with a small semantic core and versioned bindings and profiles. The core owns portable record meaning, scoped acceptance, purpose-bound request/package comparison, proposal review, and loss-aware transfer semantics. Other layers keep their existing authority:

Layer	Primary owner	OCP relationship
Human authentication	OpenID Connect or deployment-specific authentication	References authenticated actors and assurance; does not create identity.
Workload and agent identity	Deployment security, SPIFFE-like systems, platform identity	Carries typed references; never equates agent, client, model, and publisher.
Access authorization	OAuth/RAR, G NAP, OS policy, local capability broker	Carries OCP request semantics and records the outcome as a non-credential receipt.
Agent/tool transport	MCP, A2A, application APIs	Carries OCP objects or references through a binding.
Memory interchange	Portable AI Memory and provider exports	Requires an explicit field/loss mapping; does not claim all memory as OCP accepted context.
Storage	Local database, Solid, cloud, file archive, user-controlled store	Stores OCP representations without becoming mandatory to the data model.
Provenance/authenticity	PROV, C2PA, VC/Data Integrity, signature profiles	Referenced or mapped; validity is not equated with truth or acceptance.
Events/telemetry	CloudEvents, OpenTelemetry	Carries privacy-minimized notifications or measures; never becomes the record of authority.

9.2 Actors

The architecture distinguishes these roles even when one component performs several:

- **context subject:** entity described by context;
- **authorizer:** authenticated actor or policy entitled to decide a protected OCP access request;
- **steward/reviewer:** actor entitled to decide whether a proposed context item becomes accepted within a context space;
- **context provider:** protected resource server or local broker that stores records, enforces authorization, and constructs packages;
- **client application:** software that requests and consumes context;
- **delivery recipient:** the one client service and authenticated client instance to which a request, receipt, or runtime package is bound;

- **processing recipient:** a separately declared and approved entity expected to receive plaintext while processing for the delivery recipient; it is not OCP delegation;
- **client publisher:** organization or person responsible for the client declaration;
- **executing agent:** automated process acting within the client;
- **model and model provider:** inference system and its operator, which may be subprocessors;
- **source authority:** system or actor from which evidence or an artifact originated;
- **runtime source:** adapter, importer, observer, or evaluator that captured or transformed it;
- **importer/exporter:** components that parse or emit an offline or provider-transfer package;
- **authorization server or local policy broker:** component that authenticates, presents/evaluates the decision, and issues enforceable credentials or capabilities; and
- **independent verifier:** a party separate enough from the reporting provider to support a specifically scoped assurance statement.

An implementation **MUST** preserve role distinctions in records and audit. A model name **MUST NOT** substitute for a client identity. An OAuth client identifier **MUST NOT** be treated as the subject. A source provider **MUST NOT** be presumed to be the authorizer or reviewer.

9.3 Deployment models

OCP supports:

- a device-local context provider using an OS broker or authenticated local IPC;
- a hosted personal provider exposed as an OAuth protected resource;
- a self-hosted provider using user-controlled storage;
- an organization-managed provider under a dedicated organization profile; and
- provider-to-provider export/import using a transfer archive and, for non-public data, a selected registered protection profile. This draft does not supply an executable protection profile.

All deployments **MUST** produce the same semantic decision inputs: authenticated delivery recipient and client instance, all declared plaintext processing recipients, authorizer and authority basis, request, effective selectors/views, purpose, time, retention, processing uses, proposal rights, delegation, and offline-use decision. The mechanism that proves or enforces them varies by binding.

9.4 Core and profile split

The core record/protocol objects are:

1. context_space;
2. source_record;
3. context_item;
4. client_declaration;
5. context_request;
6. grant_receipt;
7. context_package;
8. update_proposal; and

9. review_decision.

The companion completes the editorial artifacts for Conversation Profile 0.4, Artifact Profile 0.4, and Provider Adapter Framework 0.4 while keeping them outside the nine-record semantic core. Erasure Status, Continuity Observation, and Human Interaction Boundary remain draft profile work. HTTP, transfer, and offline archives are bindings. Profile completion does not establish implementation conformance or independent interoperability.

10. Trust boundaries

10.1 Boundary map

OCP assumes each of these boundaries can fail independently:

Boundary	Untrusted input	Required handling
client → provider	request fields, purpose language, identifiers, proposal content	authenticate client; validate schema; rate-limit; compare with declaration; treat request as inert; authorize separately.
source → importer	archives, JSON, markup, media, URLs, provider labels, embedded prompts	sandbox and bound parsing; no ambient network/credentials; preserve taint; quarantine unknown semantics.
provider → package	selected records, renderings, omissions, external references	apply effective authorization; minimize; use aliases; verify digests; mark all entries untrusted.
package → client/model	structured content, text summaries, source artifacts	verify package; keep in data channel; prevent tool/policy effects; apply recipient controls.
reviewer → accepted state	identity, authority basis, decision, scope	authenticate; enforce authority policy; immutable decision; optimistic concurrency; audit.
provider → external recipient	plaintext disclosure	bind recipient and encrypt where required; acknowledge that authorized plaintext copying remains a residual risk.
provider → audit/telemetry	access metadata and sensitive relationships	exclude content/tokens; minimize categories and identifiers; segregate, encrypt, and retain narrowly.

10.2 Trust assumptions

The provider is trusted to apply policy and construct packages correctly, but OCP does not assume the provider is infallible or benevolent. Signed receipts, immutable revisions, export exception reports, and independent comparison make some failures detectable; they do not prevent a malicious provider from omitting information it alone controls.

The client is not trusted merely because its publisher is known. Its declaration is a claim to show the user and bind into a decision. Post-disclosure retention, training, deletion, and no-forwarding behavior are not locally provable unless a specific mechanism, attestation, audit, or controlled execution environment supplies evidence.

The authorizer is not presumed to control every named subject. The self-authorized core profile is safe only for a competent individual deciding about their own portable personal context. Multi-party, child, guardian, organizational, incapacity, inheritance, and coercion-sensitive decisions require profiles and review beyond the core.

10.4 Context and policy channels

A conforming client **MUST** maintain a control-flow separation between:

- the **context data channel**, which carries package entries, summaries, artifacts, credentials, instructions-as-descriptions, goals, past decisions, and all imported content; and
- the **authorization/policy channel**, which carries authenticated deployment decisions for resource access and external actions.

An LLM reading a warning is not an enforcement boundary. Application code **MUST** prevent context content from directly releasing tools, changing policy, selecting credentials, weakening safeguards, or approving external actions.

11. Core protocol flow

11.1 Happy path

1. The client publishes or references a current `client_declaration`.
2. The client submits a `context_request` stating purpose, selectors, views, time, retention, processing uses, offline use, proposal rights, and delegation.
3. The provider validates the request and invokes the deployment authorization flow. For remote HTTP, an OAuth Rich Authorization Request **MAY** carry equivalent structured details. For local-first, an OS or application broker **MAY** mediate the decision.
4. The provider creates a `grant_receipt` recording the effective OCP decision and a reference to the actual authorization decision. The receipt is not a credential.
5. On a separately authorized operation, the provider computes the effective authorization and constructs an immutable `context_package` that is no broader than the request, receipt, credential/capability, client instance, and current policy.
6. The client verifies the package, treats every entry as untrusted data, and uses it only within the authorized application workflow.
7. If authorized, the client **MAY** submit an `update_proposal`.
8. A reviewer or explicit policy creates an immutable `review_decision`. Acceptance creates a new context-item revision; rejection leaves accepted context unchanged.
9. Revocation prevents future package issuance and online reads. It does not recall delivered plaintext.

11.2 Effective authorization

For every protected operation, the provider **MUST** compute:

```
effective_authorization =
  authorization_credential_or_local_capability
  n active_grant_receipt
  n authenticated_client_instance
  n requested_operation
  n purpose
  n delivery_recipient_and_client_instance
```

```

n approved_processing_recipients
n selectors_and_views
n time_window
n retention_and_processing_uses
n proposal_and_delegation_limits
n offline_use_decision
n local_policy

```

The intersection notation means “the most restrictive result on every dimension.” It does not imply that arbitrary natural-language purposes form a mathematically ordered set. A profile **MUST** define comparison for every semantic field it adds. Unknown or incomparable values fail closed.

11.3 Required containment

The following invariants are normative. `snapshotMatches(ref, record)` means equality of `id`, `revision`, `schema_uri`, and the SHA-256 digest of the record's RFC 8785 canonical JSON.

`processorSet(A) <= processorSet(B)` is order-independent: processors match by exact recipient, and each matched processor's roles and `purpose_codes` are compared as mathematical sets while `receives_plaintext` remains exactly true. `processorSet(A) == processorSet(B)` means mutual containment. Array order is never authority.

```

snapshotMatches(context_request.delivery_recipient.client_declaration_ref,
                 client_declaration)
context_request.delivery_recipient.recipient == client_declaration.client
processorSet(context_request.processing_recipients)
  <= processorSet(client_declaration.processing_recipients)
context_request.retention.mode in client_declaration.retention_supported
context_request.retention.mode != "fixed" in the baseline declaration contract
requested processing-use keys in client_declaration.processing_uses_supported
context_request.offline_use == "prohibited"
  OR "explicitly_authorized" in client_declaration.offline_use_supported
context_request.proposal_rights.mode in client_declaration.proposal_rights_supported
context_request.delegation.mode == "none"
client_declaration.delegation_supported == false

snapshotMatches(grant_receipt.request_ref, context_request)
grant_receipt.purpose == context_request.purpose
grant_receipt.approved_selectors ⊆ context_request.selectors
grant_receipt.approved_views <= context_request.views
grant_receipt.expires_at <= context_request.requested_not_after
grant_receipt.retention <= context_request.retention
grant_receipt.processing_uses <= context_request.processing_uses
grant_receipt.delivery_recipient == context_request.delivery_recipient
processorSet(grant_receipt.processing_recipients)
  <= processorSet(context_request.processing_recipients)
grant_receipt.proposal_rights <= context_request.proposal_rights
grant_receipt.delegation <= context_request.delegation
grant_receipt.offline_use <= context_request.offline_use

snapshotMatches(context_package.grant_receipt_ref, grant_receipt)
context_package.grant_status_ref.uri == grant_receipt.status_ref.uri
context_package.grant_status_ref.last_checked_at >= grant_receipt.status_ref.last_checked_at
authoritative_grant_status.uri == context_package.grant_status_ref.uri
authoritative_grant_status.grant_id == grant_receipt.id
authoritative_grant_status.revision == grant_receipt.revision
authoritative_grant_status.status == "active"
authoritative_grant_status.checked_at == context_package.grant_status_ref.last_checked_at
authoritative_grant_status.checked_at is within advertised age/skew limits at trusted evaluation_time
abs(context_package.issued_at - trusted_evaluation_time) <= advertised_clock_skew
context_package.purpose == grant_receipt.purpose
context_package.delivery_recipient == grant_receipt.delivery_recipient
processorSet(context_package.processing_recipients)
  == processorSet(grant_receipt.processing_recipients)
context_package.entries satisfy grant_receipt.approved_selectors
context_package.disclosure <= grant_receipt.approved_views
context_package.expires_at <= grant_receipt.expires_at
context_package.retention <= grant_receipt.retention
context_package.processing_uses <= grant_receipt.processing_uses
context_package.offline_use <= grant_receipt.offline_use
context_package.proposal_rights <= grant_receipt.proposal_rights
context_package.delegation <= grant_receipt.delegation
context_package.external_action_authority == "none"

```

Here $\leq$ means equal or more restrictive under the field's registered partial order. Purpose comparison is exact over the complete `{code, description}` object in 0.4; a matching code with changed human review text is not the same approved purpose. Time comparison uses parsed instants, not strings. Retention uses the exact partial order and operational boundaries in Section 11.4. Omission of a necessary comparison input, an unknown value, or an incomparable profile field is an error, not permission.

11.4 Retention modes and deletion boundaries

Retention covers the received package envelope, entries, renderings, and every copy or transformation under the client or a named processing recipient's control from which disclosed content or sensitive package relationships can be recovered. Covered data therefore includes prompts and model inputs, excerpts, caches, embeddings and indexes, queued jobs, temporary files, logs and traces, crash dumps, replicas, and backups when they contain or can retrieve such information. Content-free, bounded security/audit facts permitted by Section 27.5 are governed by their own declared policy. An approved processing use does not enlarge retention, and sending data to a processing recipient does not move the deletion boundary.

The **deletion boundary** is the latest time at which covered data may remain recoverable for ordinary processing under the package. At that boundary, the client and every processing recipient **MUST** stop ordinary use and new reads; remove controlled working, online, and durable copies; invalidate retrievable indexes, caches, and links; destroy applicable keys; and propagate the same boundary to subprocessors. A backup or replica that cannot be rewritten by the boundary **MUST** either exclude the covered data or use a scoped cryptographic partition whose decryption key is destroyed by the boundary; a restore process **MUST** preserve the deletion state. Merely suppressing ordinary access while waiting for a later backup rotation does not satisfy the mode. A deployment unable to meet these requirements **MUST NOT** declare or request that mode. These duties are recipient obligations unless a separately identified enforcement or assurance mechanism proves more; an OCP field, key-destruction report, or deletion request alone is not independently verified erasure evidence.

The four modes have these exact operational meanings:

- **no_store**: covered data **MAY** exist only in volatile working memory for the currently authorized, bounded operation. It **MUST NOT** be written to durable storage or made recoverable by a cache, queue, log, trace, crash dump, swap/hibernation image, backup, or downstream store. Its deletion boundary is completion, cancellation, or failure of that operation, or any backgrounding, suspension, process termination, or crash that prevents the operation from continuing—whichever occurs first. A restart or reconnect **MUST NOT** recover the data.
- **session**: covered data **MAY** remain only within the authenticated application session, for the same delivery recipient and client instance, that received it. Its deletion boundary is the earliest of explicit session close or sign-out, authentication-session termination or timeout, and client-process termination or crash. A transport interruption and reconnect **MAY** continue the same session only while the original process and authenticated application session remain live; reconnect does not create permission to reload a persisted copy. Background suspension **MAY** preserve volatile state only if the process and session remain live; process eviction, relaunch, state restoration from durable storage, or a new login is a new session. A client **MUST NOT** advertise `session` unless it has enforceable session-ending events and documents them.
- **while_connected**: covered data **MAY** remain only during one continuously live, authenticated logical connection for the same delivery recipient and client instance. A raw TCP keep-alive is not sufficient; the applicable binding or deployment **MUST** provide a way to determine authenticated liveness. Its deletion boundary is orderly connection close, transport or

authentication loss, failed liveness determination, backgrounding or suspension that interrupts liveness, or process termination or crash—whichever occurs first. Every reconnect starts a new connection and MUST NOT recover data from the prior one. A client MUST NOT advertise `while_connected` for a binding in which it cannot enforce and observe this boundary.

- `fixed`: covered data MAY be stored, including durably, only until the RFC 3339 instant in `delete_by`. The deletion boundary is that instant; clock, restart, reconnect, crash recovery, background execution, replication, and backup rotation MUST NOT extend it. `delete_by` is REQUIRED for `fixed` and prohibited for every other mode.

A mode establishes a latest retention boundary, not a guaranteed minimum; local policy or another authorization dimension MAY require earlier deletion. `legal_retention_may_apply` is REQUIRED and is part of the authorization decision. `false` states that no legal-retention exception is declared for this package. `true` warns that a binding legal duty or order may require an exceptional copy beyond the ordinary mode boundary; it is not blanket permission to retain, does not authorize ordinary processing, and does not prove that an exception exists. When invoked, the deployment MUST minimize and segregate that copy, block ordinary use, record the authority and applicable period outside the package, and delete it when the duty ends. If disclosure of those details is legally prohibited, the deployment MUST state that limitation when permitted. OCP 0.4 supplies no legal-hold evidence profile, so the Boolean alone cannot support an assurance claim.

For retention values `a` and `b`, `a <= b` is true if and only if `a.legal_retention_may_apply == b.legal_retention_may_apply` and one of the following holds: (1) `a.mode == "no_store"`; (2) `a.mode == b.mode` and that mode is `session` or `while_connected`; or (3) both modes are `fixed` and `a.delete_by` is no later than `b.delete_by`, comparing parsed instants. No other pair is ordered. In particular, `session`, `while_connected`, and `fixed` are mutually incomparable because their boundaries cannot be ordered in advance. The baseline client declaration cannot advertise `fixed`, because a bare supported-mode token carries no maximum instant; a registered declaration profile must add a bound and comparison rule before `fixed` retention can be declared supported.

11.5 Failure paths

The provider MUST reject or quarantine:

- an invalid or unsupported schema/version;
- an unknown critical extension;
- a request without meaningful purpose text and URI purpose code;
- an inactive, mismatched, or stale grant receipt;
- an authorization credential presented to the wrong resource or client instance;
- a package request that widens any approved dimension;
- a content or manifest digest mismatch;
- an accepted item without a valid review decision reference;
- a stale proposal base revision;
- an import with unsafe paths, active content, exceeded limits, unresolved critical semantics, or integrity failure; and
- any attempt to derive external-action authority from context.

12. Complete core object model

12.1 Envelope

Persistent OCP records share these fields:

Field	Type	Requirement	Meaning
ocp_version	string	required	Core semantic wire version; exactly 0.4 here.
schema_uri	absolute URI	required	Exact schema for the concrete object.
type	string	required	Concrete object type fixed by its schema.
id	absolute URI	required	Record identifier. A urn:uuid payload MUST be a valid UUID.
revision	positive integer	required	Immutable record revision; not protocol version.
created_at	RFC 3339 date-time	required	OCP record creation time.
created_by	actor reference	required	Actor that emitted this representation.
correlation_id	absolute URI	optional	Cross-resource transaction correlation without embedding content.
extensions	URI-keyed object	required	Namespaced extensions; empty object when none.

Discovery metadata, RFC 9457 problems, and CloudEvents use their own native envelope families. They MUST NOT be forced into the persistent-record envelope.

Actor references contain `id`, `type`, and optional `display_name`. Display names are untrusted labels and MUST NOT be used for authorization. Subject references use a separate type set and MUST NOT be assumed to be authenticated principals.

12.2 context_space

User problem. Context needs a scope. “Avery's preference” may be accepted for travel but not for every purchase, work setting, or future date.

Example. A space named “Japan trip planning” contains Avery's accepted trip-planning context under a self-authorized personal policy.

Formal rule. A space identifies its subjects, domain, authority model, lifecycle, and optional authority-policy reference. A provider MUST evaluate review authority in the selected space. A space MUST NOT silently inherit authority from an unrelated parent or account.

Required fields: envelope plus name, subjects, domain, authority_model, and lifecycle.
Optional fields: description, authority_policy_ref.

The core domain enum is exactly personal, organizational, household, project, and mixed; only personal self-authorization is a baseline authorization profile. In that baseline, the space MUST be active, use domain: "personal" and authority_model: "self", contain exactly one person subject, and authorize only a decision whose reviewed_by identity and independently authenticated reviewer identity have that same {id, type} and whose candidate and result each concern exactly that subject in the exact space snapshot. Display labels never participate in the identity comparison. The other domain or authority-model values signal the need for a registered matching authority profile and MUST fail closed when none applies.

12.3 source_record

User problem. A statement needs inspectable origin without exposing all source content or treating source instructions as commands.

Example. An imported assistant message records the source authority that emitted the underlying evidence, the runtime source that captured or transformed it, a source object ID, import method, acquisition time, digest/content reference, sensitivity, and instruction_treatment: "untrusted_data".

Formal rule. A source record is evidence, not accepted context. It MUST state source_type, source_authority, runtime_source, acquisition_method, acquired_at, trust_treatment, instruction_treatment, availability, sensitivity, and lifecycle. source_authority identifies the originator/provider of the evidence; runtime_source identifies the adapter, importer, observer, or evaluator that captured or transformed it; created_by identifies the emitter of this representation. None implies another. If content is present or referenced, its media type, byte size, digest, reference/redaction state MUST satisfy the source schema. A source URI MUST NOT be interpreted as a preserved copy.

signature_verified and credential_proof_verified require verification evidence naming verifier, profile, issuer, covered digest, verification time, status method, and limitations. A verification profile or provider policy MUST define its maximum status age; the baseline comparison receives that value as an explicit policy input rather than inferring it from the grant-status limit. For an immutable source-record revision, acquired_at ≤ status_checked_at ≤ verified_at ≤ created_at; later verification evidence creates a new revision. Verification proves only the stated cryptographic relationship. It does not prove truth, lawful authority, safety, sensitivity, or acceptance.

12.4 context_item

User problem. A recipient needs a structured, reviewable statement rather than an opaque summary.

Example. The predicate prefers-hotel-style has the typed string value boutique, provenance pointing to an inference source, epistemic_basis: "subject_confirmed", and an accepted review decision scoped to one space and purpose.

Formal rule. A context item MUST contain subjects, exact space snapshot, non-control category, structured statement, epistemic basis, review object, provenance array, sensitivity, and lifecycle. The statement uses a URI predicate and typed value. The baseline core review axis is exactly candidate or accepted; dispute/reaffirmation requires a registered profile with a typed decision and authority model. superseded, withdrawn, and tombstoned are record-lifecycle states and MUST NOT appear as review status. A context item with review.status: "accepted" MUST include both an

exact decision snapshot and `accepted_scope`. A provider **MUST NOT** emit `subject_confirmed` solely from model inference. Acceptance **MUST NOT** rewrite the epistemic basis.

Core categories are `statement`, `behavior_preference`, `goal`, `decision_note`, and `note`. These labels are descriptive selection/display categories, not policy or execution semantics.

`behavior_preference` **MUST** carry `instruction_treatment`: `"untrusted_data"`; no category may enter system/developer instructions, tool policy, credential selection, or action authorization. Rich goal/decision/work semantics require a registered Work Context Profile.

Core typed values are exactly string, boolean, number, decimal, date, date-time, quantity, entity reference, and structured value. JSON number values are limited to the interoperable range -9007199254740991 through 9007199254740991. Decimal and quantity values use one canonical lexical form: base ten without exponent or leading plus; zero is exactly `0`; an integer has no leading zero; a fractional value has at least one digit on each side of the decimal point and no trailing fractional zero; negative zero is invalid. Comparison is exact over that canonical string. The baseline timestamp lexical form uses uppercase `T/Z`, seconds `00` through `59`, and at most three fractional digits; it deliberately excludes leap-second lexical values so independent ECMAScript-compatible comparators agree. A structured value identifies its `schema_uri`. Unknown domain meaning belongs in a profile or quarantined extension; it **MUST NOT** be guessed from free text.

12.5 client_declaration

User problem. A person cannot evaluate a request if the recipient's model providers, subprocessors, retention support, training uses, delegation, and deletion capability are hidden.

Formal rule. A declaration **MUST** identify the client and include policy/security contacts, every model provider or subprocessor expected to receive plaintext as a `processing_recipient`, supported retention and processing uses, supported offline-use and proposal-right modes, `delegation_supported`: `false`, deletion support, status, and expiry. Terms and conformance URI **SHOULD** be included when applicable. `processing_recipients`: `[]` means explicitly none; omission is invalid and never means none. A request **MUST** reference the exact declaration reviewed, bind the same client as delivery recipient, and use only declared processor roles/purpose codes and declared modes under the order-independent comparator in Section 11.3. Listing `session` or `while_connected` is a capability claim that the client and every named plaintext processor can enforce the corresponding Section 11.4 boundary; it is not a label for indefinite retention. A client without discrete session-ending events **MUST NOT** list `session`, and a client unable to determine authenticated connection liveness **MUST NOT** list `while_connected`. The baseline declaration advertises only `no_store`, `session`, and `while_connected`; a registered extension is required to declare bounded `fixed` retention. A declaration is not proof that the client later complied.

12.6 context_request

A stored request is inert client intent. It carries exactly one `delivery_recipient` (client service, client instance, and exact declaration snapshot), every declared `processing_recipient` expected to receive plaintext, purpose, selectors, disclosure views, requested time, retention, processing uses, offline-use request, bounded proposal rights, delegation, and provider-maintained lifecycle status. A request-create input **MUST NOT** let the client choose an approved or other server-controlled state; the provider assigns the initial `submitted` state atomically when it creates the record. The request contains no access token and grants nothing. Independent delivery recipients require separate requests, receipts, and packages.

12.7 grant_receipt

A grant receipt is a non-secret record of the approved OCP semantics. It binds the exact request snapshot and separately references current request, authorization-decision, and receipt status; names authorization system, authorizer, authority basis, purpose, approved/denied selectors, views, one delivery recipient/client instance, all approved plaintext processors, time, retention, processing uses, offline-use outcome, bounded proposal rights, delegation, and status. It MUST NOT contain or function as a bearer credential, token, capability, or action authority.

12.8 context_package

A package is an immutable data presentation. It contains an exact receipt snapshot plus live status reference, purpose, one delivery recipient/client instance, all approved plaintext processors, issue/expiry times, delivery-pairwise subject aliases, entries, omissions, warnings, retention, processing uses, offline-use decision, bounded proposal rights, delegation, `external_action_authority`: "none", and integrity metadata. The baseline has three closed entry forms: embedded accepted context, embedded reference-only evidence metadata, and embedded reference-only artifact metadata. Each declares layer, permitted disclosure, media type, exact content schema URI, digest, and `instruction_treatment`: "untrusted_data"; the two metadata forms exclude raw bytes, filenames, bearer/store URLs, and store digests. They are conservative core package projections, not complete source or artifact records and not evidence of trust, custody, preservation, or Artifact Profile conformance. Richer disputed context, evidence excerpts/full content, artifact content or content-reference delivery, and structured-value package projections require separately registered profiles. Arbitrary `{ }` content is never valid. Each optional human-readable rendering identifies its source entries, media type, language when known, content reference, digest, and untrusted-data treatment.

12.9 update_proposal

A proposal identifies submitter, exact grant-receipt snapshot, live grant status, operation (`create` or `revise`), exact source snapshots, status, and operation-specific target/proposed item. Both operations include `proposed_item`; `revise` additionally includes an exact `target_ref`. Its `submitted_by {id, type}` MUST equal an independently authenticated submitter, and a separate current authorization decision MUST permit that principal to submit this operation under the receipt; `display_name` remains non-authoritative. Its operation, exact current target, selectors, sensitivity, and source attachments MUST remain inside the receipt's bounded proposal rights. The proposed item's provenance-source set MUST equal `source_refs`; every source MUST resolve to an exact active authorized source snapshot with coherent critical-extension, verification, and chronology state. At decision time the provider MUST resolve that exact receipt again, obtain an authoritative active grant-status result, and reject an expired, suspended, revoked, superseded, stale, or mismatched authorization. The status `checked_at` MUST be no later than `decided_at + limits.clock_skew_seconds`, and `decided_at - checked_at` MUST be no greater than `limits.grant_status_max_age_seconds`; both limits come from discovery and have no binding default. A proposal is inert and never writes accepted state. Baseline 0.4 does not define an accepted-context withdrawal proposal: that operation requires a registered profile with an operation-specific decision, tombstone/result commitment, and authority model.

12.10 review_decision

A decision binds the exact proposal snapshot and records reviewer, authority basis, outcome, decision time, optional reason, and—only when the outcome is accepted or `accepted_with_changes`—a commitment to the resulting item's semantic projection. A non-accepting outcome MUST NOT carry a resulting-item commitment. To avoid an impossible decision/item digest cycle, the projection excludes

only envelope members `ocp_version`, `schema_uri`, `type`, `id`, and `revision`, plus the complete review object including `review.decision_ref`; it includes the creation/audit semantics and accepted scope named below. The resulting accepted item then binds the exact immutable decision snapshot. Decisions are immutable. A correction uses a new proposal and decision and, when accepted state changes, a new item revision; exact proposal, target, item, and decision snapshots provide the correction lineage. Baseline 0.4 does not define a generic decision-to-decision supersession relation.

The `ocp-0.4-context-item-accepted-result-v1` projection is a JSON object containing, without relying on member order, required `created_at`, `created_by`, `extensions`, `subjects`, `space_ref`, `category`, `statement`, `epistemic_basis`, `provenance`, `sensitivity`, `lifecycle`, and top-level `accepted_scope`, plus optional `correlation_id`, `instruction_treatment`, `validity`, and `rights`. It excludes `ocp_version`, `schema_uri`, `type`, `id`, and `revision`—the result commitment carries item identity and revision separately—and excludes the entire review object, including the cyclic `review.decision_ref`; `extensions` remains covered because a critical extension can change semantics, while creation actor/time and correlation remain covered audit semantics. The `accepted_scope` in the projection MUST equal both the accepted item's `review.accepted_scope` and the commitment's separately carried `accepted_scope`. The decision's outcome MUST be `accepted` or `accepted_with_changes` for this commitment to exist. Its `semantic_digest` is SHA-256 of the RFC 8785 canonical JSON for that projection. A provider MUST recompute and compare it before emitting the accepted item. A mismatch MUST reject the state change; it MUST NOT be repaired by silently changing the decision or item.

12.11 Profile objects

The non-core companion now contains:

- complete working-draft editorial packages for `conversation`, `message`, `content parts`, `artifacts`, `representations`, `lineage`, `adapter manifests/mappings/runs`, and `loss reports`;
- draft Continuity Observation and Human Interaction Boundary profiles with no authority effect;
- Transfer and Offline Archive bindings;
- draft Erasure Status records; and
- legacy core metadata-only artifact projection vectors retained solely for core package compatibility tests.

Only the three Portable AI History profile families have the complete 0.4 editorial package. Every implementation claim still requires its exact conformance class and evidence.

13. Normative schema definitions

13.1 Schema dialect and validation

All OCP 0.4 schemas use JSON Schema Draft 2020-12. A Schema Conformance implementation MUST:

- compile the complete schema set;
- enforce format checks used by OCP, including URI, date, and date-time;
- reject duplicate JSON member names before schema evaluation;
- reject values outside the selected JSON processing limits;
- resolve only trusted schema locations or a pinned local catalog;

- reject unknown unevaluated core properties; and
- validate the declared `schema_uri` against the concrete object type.

JSON Schema validates one representation. Referential integrity, authorization containment, current state, lifecycle legality, digest bytes, accepted-state linkage, round-trip preservation, and authority require behavioral validation.

13.2 Identifier schema

The common URI definition has two branches: a strict lowercase RFC 9562 UUID-URN pattern, or another absolute URI that is not prefixed `urn:uuid:`. This prevents mnemonic labels such as `urn:uuid:claim-1` from passing as UUID URNs. Implementations SHOULD use UUID URNs for opaque local identifiers and controlled HTTPS URIs for governed vocabularies and schemas. The canonical OCP 0.4 working-draft namespace is `https://opencontinuity.org/`. Identifiers in that namespace are project-controlled and versioned where their semantics require it; their publication does not make OCP 0.4 a stable standard or imply IANA registration. Implementations MUST preserve the exact identifier and version declared by an object and MUST NOT silently rewrite older provisional identifiers.

A `snapshotRef` contains `id`, positive `revision`, exact `schema_uri`, and SHA-256 digest of the referenced RFC 8785 canonical JSON bytes. A `liveStatusRef` contains an authorized status URI and optional last-check time. Security-significant relationships across trust boundaries MUST use a complete snapshot reference; live status MUST NOT substitute for immutable content identity, and a snapshot MUST NOT be treated as current status.

A Package Producer MUST keep store-internal identifiers private. It MUST replace every disclosed subject, package, entry, item, space, decision, source, artifact, relationship, and other store identifier—and every intra-package reference to one—with delivery-pairwise or package-local values while preserving referential integrity. This rule covers statement `entity_ref` values and qualifier values, structured entries, embedded objects, renderings, warnings, omissions, and content references. The baseline rewrites a typed `entity_ref` only through a declared `relationship` alias and fails closed for structured values until a registered package-projection profile defines identifier traversal. A producer MUST generate aliases using a cryptographically secure random mapping or keyed domain-separated PRF over the verified delivery service, client instance, purpose/scope domain, and rotation epoch; it MUST NOT expose an unkeyed hash of a stable identifier. The producer MUST collect the values of identity-, reference-, content-reference-, key-reference-, and digest-bearing store fields, then compare that set against every emitted package string. A URI-shaped vocabulary or genuine free-text value is not a store identifier merely because of its lexical form, but placing a known store identifier or digest in a generic string field does not declassify it. Every emitted alias table value, package ID, entry ID, package item reference, rendering reference, and digest is within this scan; unused alias rows and identity mappings MUST fail. The internal-to-disclosed mapping remains provider-side. A registered profile may require a stable identifier only when separately authorized. Receivers MUST NOT merge subjects solely because labels, names, or unverified IDs match.

13.3 Typed values and predicates

Every core statement uses a URI predicate and typed object. A vocabulary URI identifies semantics and version. Free text MAY appear as a typed string and is still untrusted. A structured value MUST identify its schema URI. Profile authors MUST publish comparison, normalization, and loss behavior; they MUST NOT rely on an LLM to infer wire semantics.

13.4 Provenance

A provenance entry identifies an exact `source_ref` snapshot and method; it may include exact activity reference, observation/acquisition times, producer, `runtime_source`, adapter version, and redaction state. Allowed core methods include subject entry, import, observation, inference, transformation, and credential verification.

Provenance records who or what produced a representation and how; it does not prove truth, lawful collection, subject authority, authenticity, or current availability. An implementation MAY map provenance to W3C PROV but MUST preserve OCP acceptance and review semantics that PROV alone does not supply.

13.5 Integrity

The core digest string is `sha-256`: followed by 64 lowercase hexadecimal characters. JSON decoders MUST use UTF-8, reject a byte-order mark, duplicate member names, lone Unicode surrogates, non-finite numbers, and integers outside the interoperable range unless represented by the core decimal string form. Digested JSON MUST use RFC 8785 JCS. The package integrity digest covers the complete package with the top-level `integrity` member omitted; each embedded entry digest covers its embedded value; each referenced binary digest covers exact referenced bytes. Implementations claiming serialization conformance MUST pass published JCS vectors in at least two independent language implementations before an interoperability claim.

This digest detects accidental or malicious byte changes. It does not authenticate an issuer. A remote security profile MUST select a reviewed signature format, trust model, key identifiers, algorithms, covered representation, rotation, and recipient-confidentiality mechanism. Implementations MUST NOT invent an ad hoc signature or encryption scheme.

13.6 Extensions

Extensions are URI-keyed values with `critical` and `data`. A receiver MUST reject an unknown critical extension. It MAY ignore the semantics of an unknown non-critical extension during runtime use, but an importer/exporter MUST preserve the JSON value and critical flag across a conforming round trip or report loss. Unknown extension data MUST NOT be rendered, indexed, disclosed, or supplied to a model until explicitly understood and authorized.

13.7 Schema inventory

The synchronized companion contains 37 schemas, 32 positive examples, 45 single-mutation negative examples, 243 inherited semantic reference vectors, 29 registry/OpenAPI/evidence-accounting gates, 24 Portable AI History cases, and 134 registered assertions with 134 result rows. The status split is zero implementation passes, zero implementation failures, 90 partial local-reference results, four manual-required results, three unavailable results, and 37 not executed. These are artifact evidence-accounting values, not independent interoperability, deployed assurance, or conformance of an identified implementation.

14. Request lifecycle

14.1 Request creation

A client MUST create a new request when purpose, client declaration, selectors, views, time, retention, processing uses, offline use, proposal rights, or delegation materially changes. It MUST NOT reuse an earlier approval through a broader reinterpretation. The HTTP request-create input carries client-

supplied intent but not a client-selected lifecycle outcome; the provider assigns `submitted` and all later request states.

The request schema requires `purpose.code` for machine comparison and `purpose.description` for human review. A provider **MUST** reject an empty, generic, deceptive, or internally inconsistent description when a reasonable reviewer could not understand the requested use. Purpose codes do not remove the need for plain language.

Selectors use structured layer, space references, `item_categories`, predicates, optional profile, time range, and maximum sensitivity. The baseline containment algorithm is conservative: a grant selector must be structurally equal to or narrower than a request selector on every present dimension. A new ontology relationship is incomparable until a registered profile defines it.

In the Core Personal Profile, a client **MUST** request exactly `delegation: {"mode": "none"}`. A provider **MUST** reject `named_recipients` with `OCP_UNSUPPORTED_PROFILE` unless the request selects a supported, registered Delegation Profile.

14.2 States

```
submitted -> awaiting_decision
submitted -> withdrawn | superseded
awaiting_decision -> partially_approved | approved | denied | expired | superseded
partially_approved -> approved | denied | expired
approved | denied | expired | withdrawn | superseded -> terminal
```

Every legal transition **MUST** be atomic, revision-checked, idempotent under the same idempotency key, and recorded as an audit event. A terminal request **MUST NOT** be reopened. A materially revised request receives a new identifier and correlation/supersession link.

14.3 Partial approval

The provider **MAY** approve a strict subset. It **MUST** create a receipt containing the effective approved selectors/views and **MAY** record denied selectors. The user interface **MUST** distinguish approved from denied parts and **MUST NOT** present partial approval as full approval.

14.4 Illegal transitions

The following are illegal: client-created `approved`; terminal-to-active; expiry extension in place; purpose substitution; widening selectors after review; changing the declaration reference without a new review; and approval without a deployment authorization decision. The provider returns `OCP_ILLEGAL_TRANSITION`, `OCP_STALE_REVISION`, or the appropriate authorization error.

15. Grant lifecycle

15.1 Grant receipt versus authorization

The `grant_receipt` documents effective OCP semantics so clients, providers, auditors, and users can correlate the decision. It is not an OAuth access token, ID token, refresh token, capability, session, API key, or external-action authorization.

A provider **MUST NOT** authorize a protected operation solely because a caller presents a valid request, receipt, receipt ID, package ID, signature, subject ID, or review decision. Every protected operation **MUST** be authorized at operation time by the deployment's authorization layer.

In the Core Personal Profile, a provider **MUST** issue exactly `delegation: {"mode": "none"}`. It **MUST NOT** issue `named_recipients` without a supported, registered Delegation Profile and a new authorization decision covering that profile.

15.2 Remote binding

In the remote HTTP binding:

- the provider acts as an OAuth protected resource and publishes exact HTTPS resource metadata;
- OCP request semantics SHOULD be carried in a registered or collision-resistant RFC 9396 `authorization_details` type;
- authorization code clients use PKCE under current OAuth security practice;
- tokens are resource-restricted to the OCP protected resource and bound to the authenticated client instance where the selected security profile requires it;
- high-sensitivity profiles require sender-constrained tokens such as DPoP or mTLS;
- detailed context inventories SHOULD remain out of broadly visible self-contained tokens; and
- the provider compares effective authorization details, not only scope strings or JSON object equality.

OpenID Connect MAY authenticate a person, but an ID Token is not OCP resource authorization. OAuth token revocation and OCP receipt revocation SHOULD be coordinated while remaining distinct from recipient deletion.

15.3 Local binding

A device-local deployment MAY use an OS broker, code-signing identity, authenticated Unix-domain socket, secure pairing, device-bound key, and user presence instead of a remote OAuth server. It MUST NOT equate loopback, device unlock, filesystem access, or same-user processes with blanket trust.

If TCP loopback is used, the provider MUST implement application-instance authentication, unguessable pairing or key exchange, host/origin checks, DNS-rebinding defense, CSRF defense, restrictive CORS, short-lived capabilities, and OS-protected secret storage. A browser origin without explicit pairing MUST be denied.

15.4 States

```
active -> suspended -> active
active | suspended -> expired | revoked | superseded
expired | revoked | superseded -> terminal
```

Widening or renewal creates a new request and receipt. Narrowing creates a superseding receipt revision or new receipt according to the binding. Revocation is terminal; restoration requires new authorization. Suspension denies package creation and item reads until reactivation.

15.5 Audit events

Required events include activation, suspension, reactivation, expiry, revocation, and supersession. Events MUST contain references and revisions, not tokens, raw context, free-text purpose, or evidence. Status reads MUST be authorized and MUST return a fresh `checked_at`; a `liveStatusRef` that records the same observation uses `last_checked_at`.

16. Package lifecycle

16.1 Construction

A provider MUST construct a package from a named accepted-context snapshot, the exact active receipt referenced by `grant_receipt_ref`, a freshly resolved authoritative state from the URI in `grant_status_ref`, authenticated client instance, current policy, and provider-controlled evaluation time. The resolved status MUST bind the receipt ID and revision, be `active`, and have `checked_at` exactly equal to the package reference's `last_checked_at`; a client-supplied `requested_at`, package timestamp, or copied receipt status is never evaluation time or current-state evidence. It MUST apply the effective-authorization intersection and MUST fail closed if a selector, purpose, policy, source, accepted decision, integrity value, current status, or critical extension cannot be evaluated. Before emission it MUST enforce the complete receipt-to-package containment block in Section 11.3, including exact purpose, receipt/status binding, order-independent processor-set equality, proposal rights, and delegation.

The provider MUST include only entries necessary for the approved purpose and views. It MUST NOT insert raw evidence, artifact content, stable internal identifiers, unrelated context, credentials, or unknown extension data merely because they are available.

Before emission, the Package Producer MUST rewrite every disclosed identifier and matching intra-package reference to delivery-pairwise or package-local values. It MUST compare all emitted URI and digest values against the store-side identifier and digest set and fail closed if an unauthorized stable identifier or correlatable store digest remains. The mapping MUST stay provider-side and MUST NOT appear in the package. Package entry content, omission selectors, warning references, and packaged proposal-right selectors use package-local URI aliases—not `snapshotRef`—for subjects, spaces, items, sources, decisions, artifacts, and relationships. An alias with a copied store digest is invalid because it is neither an exact snapshot nor unlinkable.

16.2 Entry rules

Each entry MUST:

- declare layer context, evidence, or artifact;
- use a disclosure value permitted by the receipt's views;
- identify media type, content schema URI, and digest;
- contain exactly one of embedded content or `content_ref`;
- use an opaque entry ID;
- include `instruction_treatment`: "untrusted_data"; and
- satisfy all active selector, sensitivity, validity, third-party, organization, and profile rules.

The schema discriminates context, evidence, and artifact branches and closes every layer/disclosure combination. Baseline context disclosure is accepted; baseline evidence and artifact disclosure is metadata. An accepted-context entry MUST validate against the package-context entry schema, carry accepted review state, and preserve package-local subject, space, item, decision, and provenance relationships.

A baseline evidence-metadata entry MUST embed `packageEvidenceMetadata`, including `package_source_ref`, `package_space_ref`, and one or more `supports_package_item_refs`. Every support reference MUST identify an accepted-context entry

in the same package, that entry's `package_space_ref` MUST equal `package_space_ref`, its `package_provenance_refs` MUST include `package_source_ref`, and its resolved context-item revision MUST contain an exact provenance snapshot matching the source record resolved from that alias. The resolved source MUST satisfy current lifecycle, selector, sensitivity, and authorization rules, and an unknown critical extension on it fails package construction. The projection may carry only `unassessed`, `user_supplied`, or `suspected_tampering` as `trust_treatment`; a verified trust label or its proof requires a separately registered evidence/verification profile.

A baseline artifact-metadata entry MUST embed `packageArtifactMetadata`, including `package_artifact_ref`, `package_space_ref`, one or more `supports_package_item_refs`, and `content_disclosed: false`. Every support reference MUST identify packaged accepted context in that same package and space. The artifact descriptor resolved from `package_artifact_ref` MUST carry an exact `space_ref` identifying the context-space snapshot represented by `package_space_ref`, satisfy current lifecycle, selector, sensitivity, and authorization rules, and contain no unknown critical extension. The small evidence and artifact metadata objects are conservative core package projections defined by the context-package schema. They are not complete source records or artifact descriptors, do not establish conformance to the incomplete Artifact Profile, and carry no verified-source, custody, preservation, availability-independent, rights, or chain-of-custody assurance. In particular, baseline artifact metadata omits `custody_mode`, content/store references, store digests, and preservation state.

Full evidence or artifact bytes are not deliverable through the baseline HTTP binding: `content_ref` dereference remains unavailable until a registered content-reference profile defines authorization, transport, digest coverage, parser limits, and revocation/currentness checks. Disputed context, evidence excerpts or full content, artifact content, and any other broader projection likewise require an applicable registered profile. A rendering MUST identify the entries from which it was derived, its media type, digest, content reference, and `instruction_treatment: "untrusted_data"`; it MUST carry a BCP 47 language tag when the language is known, and omission means unknown rather than a locale default. A natural-language rendering cannot replace structured entries for core round-trip conformance.

16.3 Omissions and warnings

The provider MUST record material omissions needed to interpret completeness, using registered reasons such as not granted, minimized, unavailable, unsupported profile, third-party restricted, organizational policy, stale, or integrity failure. It SHOULD coarsen or suppress an omission when naming the withheld category would itself disclose sensitive information to an unauthorized recipient.

Warnings identify stale, disputed, degraded, unavailable, or profile-specific conditions. Baseline warning code and message pairs MUST come from the provider's registered, reviewed warning-template set; arbitrary record, source, client, or model text MUST NOT populate message. Warning text is untrusted and MUST NOT expose raw context or policy internals.

16.4 Runtime and transfer packages

A runtime context package is short-lived, delivery-recipient-specific, minimized for one use, and offline-prohibited by default. A transfer archive is user-directed, potentially large, entry-manifested, and processed through import quarantine. When it contains non-public data it requires a selected registered protection profile that binds confidentiality to a destination- or user-controlled key; this working draft does not provide an executable protection profile. A transfer archive MUST NOT be accepted directly as context merely because any available signature and digests validate.

16.5 States and immutability

```
built -> available -> delivered
built -> rejected
available | delivered -> expired | invalidated
expired | invalidated | rejected -> terminal
```

Package bytes are immutable. Lifecycle status is maintained by status service/events; mutation requires a new package ID. `invalidated` signals a current policy, integrity, or source failure but cannot retract already delivered plaintext. A profile **MUST NOT** claim one-time plaintext access. It may claim one-time provider acceptance only when an online atomic counter or separately reviewed hardware mechanism enforces it, defines replay storage and race behavior, and tells the user that an authorized process may retain plaintext after first release.

16.6 Offline use

Core runtime packages use `offline_use`: "prohibited". A profile **MAY** permit `explicitly_authorized` only when the request says `requested`, the grant receipt says `permitted`, the consent interface makes offline limitations clear, and the selected security profile protects the offline package. Even then, the provider cannot guarantee timely revocation, usage counting, staleness refresh, or plaintext recall.

16.7 Forwarding controls and limits

Non-public remote packages **MUST** use recipient confidentiality and issuer-authenticity controls selected by a registered security profile. A runtime package **SHOULD** bind one client-instance key and **MUST** use package-local aliases. Baseline delegation is `none`: a recipient **MUST NOT** forward or reuse the original package, plaintext, token, capability, receipt, or URL for another recipient.

A future Delegation Profile may authorize a hop only by authenticating the child principal, creating a separately enforceable child authorization and child receipt, constructing a new delivery-recipient-specific package, encrypting it to the child recipient/client instance, preserving the root/parent/actor chain, and strictly attenuating selectors, views, purpose, delivery recipient, lifetime, retention, processing uses, proposal rights, delegation depth, offline use, and local policy. The original package and plaintext remain non-forwardable.

Cryptography can stop an unintended holder from opening ciphertext and can constrain stolen tokens. It cannot prevent an authorized process from copying plaintext. Conformance statements **MUST** distinguish cryptographically recipient-bound, recipient-declared no-forwarding, attested, audited, and unknown downstream behavior.

17. Update-proposal and review lifecycle

17.1 Proposal rights

A client may propose only when `proposal_rights.mode` is bounded and the operation, target space, category/predicate, sensitivity, and source-attachment behavior are inside the listed constraints. Because every baseline candidate requires exact provenance, bounded baseline create/revise rights use `source_attachment`: "referenced_authorized"; `prohibited` is not a usable bounded baseline grant and a profile that wants no optional attachments **MUST** first separate required provenance from attachment semantics. A client with `mode`: "none" **MUST** be denied. The provider **MUST** re-evaluate the exact target/source authorization at submission and decision time. Proposal rights never permit direct writes or review decisions.

The provider MUST rate-limit proposals by client, subject/space, sensitivity, and review burden. It MUST render proposals in provider-controlled, accessible UI. It SHOULD batch only low-risk related proposals and MUST NOT use acceptance fatigue as an implicit approval mechanism.

17.2 Optimistic concurrency

A create proposal omits `target_ref`. A revise proposal identifies an exact `target_ref`; the provider MUST compare its ID, revision, schema, and digest to the current active accepted authorized target atomically. The revise candidate uses the same item ID and exactly the next revision. The exact reference replaces a separate ambiguous `expected_revision` field. A mismatch returns `OCP_STALE_REVISION`; the provider MUST NOT silently rebase the proposal or reveal unauthorized replacement content. Accepted-context withdrawal is outside the baseline proposal operation vocabulary until a registered profile defines an operation-sensitive decision and withdrawn/tombstone result commitment.

17.3 Proposal states

```
submitted -> under_review | withdrawn | expired | superseded
under_review -> deferred | accepted | accepted_with_changes | rejected | disputed | expired
deferred -> under_review | rejected | withdrawn | expired
disputed -> under_review
accepted | accepted_with_changes | rejected | withdrawn | expired | superseded -> terminal
```

Each proposal revision is immutable. A proposal-create input MUST NOT let the client select `under_review`, an acceptance outcome, or any other server-controlled state; the provider assigns `submitted` when it creates the proposal. Later state transitions are represented by immutable resource revisions or a binding status record. A review decision MUST bind the exact pre-decision proposal revision from which its outcome is an exact registered transition: `under_review` for `deferred`, `accepted`, `accepted_with_changes`, `rejected`, or `disputed`, and additionally `deferred` for `rejected`. It MUST NOT bind a merely submitted proposal, a disputed proposal pending return to review, or an already terminal revision. The terminal proposal status is maintained separately and MUST be coherent with the decision, but it is not the proposal snapshot the decision reviews.

17.4 Review decisions

A review decision MUST identify the reviewer, authority basis, proposal, outcome, and decision time. For every outcome, the provider MUST resolve and evaluate every proposal-governing space: `proposed_item.space_ref` for create/revise and the exact resolved target's `space_ref` for revise. It MUST bind `reviewed_by` to an independently authenticated reviewer by `{id, type}` and evaluate that reviewer against each space's authority model. An accepting create/revise decision additionally evaluates the result space and every `accepted_scope.space_refs` member. Baseline self authority is limited to the one-person personal-space rule in Sections 12.2 and 28.1; an `authority_policy_ref`, `authority_basis.reference`, or richer authority model fails closed without a registered profile that resolves and evaluates it. Only `accepted` and `accepted_with_changes` outcomes carry a resulting-item commitment; the resulting item MUST have `review.status: "accepted"` and `lifecycle.status: "active"`, and the committed projection MUST contain `accepted_scope`, MUST exclude `decision_ref`, and MUST recompute to that result. In baseline 0.4, every `accepted-scope` space MUST satisfy a complete approved grant selector for the resulting item's category, predicate, sensitivity, and space, and every `accepted_scope.purpose_codes` member MUST equal the exact active grant purpose; richer multi-purpose authority requires a registered profile. An `accepted` outcome preserves `ocp-0.4-context-item-candidate-promotion-v1`: the complete candidate `context_item` with only

`id`, `revision`, `created_at`, `created_by`, `optional_correlation_id`, and the complete review object omitted; the provider compares its RFC 8785 canonical JSON to the identically projected result. This covers version/schema/type, extensions, subjects, space, category, statement, epistemic basis, validity, provenance, sensitivity, rights, lifecycle, and any other schema-permitted member. Any changed covered semantic requires `accepted_with_changes`, which MUST capture the reviewed resulting item rather than mutating the client proposal invisibly. A create result uses a provider-issued result ID that is fresh in the provider's context-item namespace and begins at revision 1. A revise result MUST retain the exact target/proposed-item ID and use `target_ref.revision + 1`. `rejected`, `disputed`, and `deferred` decisions MUST NOT claim a resulting item. Rejection and deferral SHOULD include a human-appropriate reason when safe.

Automated review MAY be used only under an explicit, versioned, inspectable policy within the authority model. High-sensitivity, multi-party, disputed, credential, organization-managed, or rights-affecting changes MUST NOT be auto-accepted in the self-authorized core profile.

17.5 Required audit events

Submit, review start, defer, dispute, accept, accept-with-changes, reject, withdraw, expire, supersede, and stale-revision rejection MUST be auditable without logging raw proposal content. Acceptance events reference both decision and resulting item revisions.

18. Accepted-context rules

18.1 Meaning of accepted

accepted means: an authorized reviewer or explicit policy approved this exact context item revision for the recorded `accepted_scope`. It does not mean true, complete, current forever, agreed by every subject, lawful for every recipient, or authorized for action.

18.2 Acceptance requirements

An accepted item MUST:

- reference an immutable review decision;
- state the spaces and/or purpose codes of its accepted scope;
- retain provenance to the source records and transformations that produced it;
- retain contradictory or superseded evidence references when policy permits;
- carry validity/review times appropriate to the statement;
- carry sensitivity and lifecycle state; and
- create a new revision rather than overwriting history when materially changed.

18.3 Epistemic basis

The core `epistemic_basis` enum is exactly `subject_asserted`, `subject_confirmed`, `provider_observed`, `provider_inferred`, `provider_interpreted`, `third_party_asserted`, `credential_derived`, and `imported_unclassified`. It is independent of review status. Verification of a credential proves only what the credential proof/status profile establishes; it does not automatically make the underlying claim true or accepted.

18.4 Dispute and contradiction

Contradictory items MAY coexist. A provider MUST NOT collapse conflict by last-write-wins, model confidence, provider preference, or lexical similarity. Baseline core records do not define a disputed review state because this draft has no portable typed dispute decision, prior-acceptance/dispute-scope representation, or reaffirmation authority model. A future registered dispute profile must define immutable decision and authority evidence, revision rules, retained acceptance scope, presentation, comparison, and exact dispute/reaffirmation events before disputed state or `accepted_and_disputed` can appear on the wire.

18.5 Supersession, withdrawal, and tombstones

A superseding item points to prior revisions; the prior record remains addressable to authorized audit/round-trip processes. Withdrawal removes an item from current accepted presentation without claiming the source never existed. A tombstone preserves the minimum identifier, revision, transition, and policy-required lineage needed to prevent silent resurrection while minimizing content.

18.6 Time semantics

`created_at` is the OCP representation time. `observed_at` is when the source event or state was observed. `acquired_at` is when OCP obtained it. `valid_from` and `valid_until` describe the statement's claimed validity. `last_confirmed_at` records review evidence. `review_after` signals re-review. Implementations MUST NOT substitute one for another. Normative comparisons parse instants; a range with both endpoints requires `from < until`. Package issuance uses provider-controlled evaluation time and the binding's advertised clock-skew/status-age limits, not an object-authored substitute.

An ordinary current-context package MUST NOT outlive an item's `valid_until`: when that member is present, the package `expires_at` MUST be no later. The baseline also requires the entire package lifetime to precede `review_after`: when that member is present, the package `expires_at` MUST be earlier. A registered profile may define review-due inclusion only by specifying warning semantics, renewed evaluation, recipient behavior, and a stricter package lifetime; the baseline has no such exception. Historical-purpose profiles MUST state their different evaluation time explicitly.

18.7 Concurrency

Accepted-state writes require an exact target snapshot and a strong `If-Match` or equivalent atomic condition. Arrival time and wall-clock skew do not decide conflicts. The provider uses revision, digest, and explicit supersession. Full CRDT/vector-clock behavior is deferred until offline multi-writer experiments demonstrate a common need.

19. Evidence and provider-interpretation rules

19.1 Three layers

OCP separates:

1. **source evidence** — imported bytes, observations, messages, documents, credentials, or references represented by `source_record` or a profile;
2. **provider interpretation** — a context item inferred, summarized, transformed, or proposed by a provider/client; and
3. **accepted context** — an exact context-item revision approved for a recorded scope.

Implementations **MUST** preserve this separation in storage, package construction, user interfaces, and exports. A provider **MAY** derive an interpretation from evidence, but it **MUST** record the method and **MUST NOT** label it subject-confirmed or accepted without the corresponding review evidence.

19.2 Evidence relationships

Evidence **MAY** support, contradict, qualify, supersede, or merely relate to an item. The relationship **SHOULD** be an explicit, profile-versioned link rather than inferred from co-location. Absence of cited evidence does not prove that none exists; a package **MAY** omit evidence because it was not requested, not approved, unavailable, minimized, or restricted.

19.3 Confidence and uncertainty

Confidence is not a universal scalar. Core context items do not require a single numeric confidence. A profile **MAY** define calibrated confidence only when it specifies producer, method, range, interpretation, aggregation limits, and update behavior. Unknown confidence **MUST** be omitted, not encoded as `null`, zero, or "certain."

An importer unable to understand a provider confidence field **MUST** preserve or quarantine the source field and report uncertainty. It **MUST NOT** normalize arbitrary provider scores to a shared scale without a mapping profile.

19.4 Source availability and preservation

A source record distinguishes availability from integrity and custody. `source_uri` **MAY** identify an external source. Embedded or referenced content **MAY** state byte size and digest. A source dependency remains true until an independently stored copy has been verified under an artifact or preservation profile.

19.5 Provider disagreements

Two providers may offer different interpretations of the same evidence or subject. A conforming provider **MUST** retain provider identity, acquisition method, time, and accepted scope. It **MUST NOT** merge items solely by matching normalized text. A context provider **MAY** present a conflict review, but the result is a new decision; it does not erase either provider's provenance.

19.6 Credential and attestation evidence

Credentials and attestations belong in a credential/evidence profile. A Credential Profile **SHOULD** reference an independently issued Verifiable Credential, SD-JWT presentation, or other credential in its native form and record verification result, time, verifier, status method, and disclosed fields. A conforming provider **MUST** distinguish:

- proof format valid;
- issuer trusted for the claimed purpose;
- credential status acceptable at verification time;
- credential statement accepted as context; and
- external action authorized.

These are separate decisions.

20. Conversation Profile

20.1 Profile identity and scope

The **OCP Conversation Profile 0.4** has stable identifier

<https://opencontinuity.org/profiles/conversation>, profile version 0.4, and compatible core range ≥ 0.4 < 0.5 . Its normative schemas are `conversation.schema.json`, `message.schema.json`, and `conversation-content-part.schema.json`. The profile preserves source conversation history as evidence. It does not make messages accepted context, define model memory, require disclosure of private chain-of-thought, or authorize tools or actions.

A conforming implementation declares one or more of `OCP-Conversation-Import-0.4`, `OCP-Conversation-Package-0.4`, and `OCP-Conversation-RoundTrip-0.4`; it **MUST NOT** claim simply “OCP 0.4 compliant.” Profile data remains in the evidence/data channel. Import, validation, source authentication, provider confidence, and display do not change that status.

20.2 Graph model and the turn-object decision

A conversation is a directed acyclic graph of message revisions, not an ordered transcript. A resolved edge points from each `parent_ref` to its child message. The schema permits one or more roots, multiple children, multiple parents, alternative generations, regenerated assistant responses, edited or reissued user turns, selected leaves, abandoned leaves, unresolved parents, and partial exports.

OCP 0.4 does **not** define a universal turn object. Providers disagree on whether a turn groups one user message, a fan-out of assistant candidates, tool events, edits, or an entire interaction transaction. Encoding one interpretation would manufacture semantics. Providers **MAY** preserve a native turn identifier in a non-critical provider extension. A later profile may standardize a turn only after independent mappings demonstrate equivalent semantics.

For a complete graph, an importer or producer **MUST** enforce all of the following:

1. every `message_ref` resolves to exactly one message snapshot in the declared package scope;
2. every resolved message has the same stable `conversation_ref` identity and revision as the containing conversation;
3. every `parent_ref` resolves to a member of the same graph;
4. the resolved parent graph is acyclic;
5. `root_message_refs` is exactly the set of messages with no parents;
6. every selected leaf and every declared branch leaf resolves to a graph leaf;
7. message-array order has no semantic meaning; parents define order, with `sibling_order` used only when the source supplies a stable sibling order; and
8. duplicate exact references, duplicate provider IDs mapped to one identity, and contradictory leaf states are errors.

For a non-complete graph, unresolved parents are permitted only when `graph_state` is `partial` for an intentionally bounded export or `source_incomplete` when the supplied source is itself known to be incomplete. Every missing parent is listed in `unresolved_parent_refs`, and `CONV_PARENT_UNKNOWN` is recorded. A child whose parent is unresolved is an exported component root, not proof of a source root. A producer **MUST NOT** flatten branches into a single sequence or discard unselected generations without `CONV_BRANCH_LOST`. A non-complete graph cannot be called round-trip lossless.

`conversation.message_refs`, `roots`, `leaves`, `branches`, and `message` parents are exact snapshot references. `message.conversation_ref` is deliberately a stable record identity and revision without a digest: making both sides exact snapshot references would create an impossible reciprocal hash cycle. This exception **MUST NOT** be generalized to non-cyclic references.

20.3 Conversation record

The conversation record uses the core record envelope and defines:

Member	Requirement and meaning
<code>profile_id</code> , <code>profile_version</code> , <code>compatible_core</code>	Exact profile identity and compatible core range.
<code>subjects</code> , <code>space_ref</code>	Governed subjects and context space; not inferred global identity.
<code>source_authority</code>	Authority that supplied the source history; not current control authority.
<code>runtime_source</code>	Importing/mapping runtime identity and build evidence.
<code>provider_conversation_id</code>	Optional provider-local provenance identifier, never a universal ID.
<code>title</code>	Optional source or user title.
<code>source_created_at</code> , <code>source_updated_at</code>	Optional source timestamps with source semantics.
<code>imported_at</code>	Optional time the source entered the importing system.
<code>representation_created_at</code>	Required time this OCP representation was created.
<code>root_message_refs</code> , <code>message_refs</code> , <code>selected_leaf_refs</code>	Exact graph snapshot references.
<code>branches</code>	Optional portable branch labels, leaves, and selected, current, alternate, abandoned, or <code>provider_defined</code> state.
<code>graph_state</code> , <code>unresolved_parent_refs</code>	complete, intentionally partial, or <code>source_incomplete</code> , plus missing-parent evidence.
<code>participants</code>	Optional source participants; absence is not evidence of no participant.
<code>ordering_semantics</code>	Optional description of provider ordering beyond parent edges.
<code>integrity</code>	Optional semantic graph digest declaration.
<code>provider_extensions</code>	Inert provider metadata.
<code>sensitivity</code> , <code>lifecycle</code>	Disclosure classification and record lifecycle.

Only `representation_created_at` is universally required among the four time concepts. A producer **MUST NOT** copy import time into source time or invent source precision. If source precision is lost, the adapter records `CONV_TIMESTAMP_PRECISION_LOST`.

20.4 Message record

A message identifies its conversation, sender, source role, kind, parent snapshots, typed content parts, redaction, sensitivity, and lifecycle. `sent_at`, `completed_at`, `source_timestamp_precision`, provider message ID, sibling order, generation metadata, and artifact/citation/tool references are optional because source exports vary.

`source_role` preserves the source label. `message_kind` gives a bounded portable class: `user_turn`, `assistant_response`, `system_source`, `developer_source`, `tool_source`, `agent_source`, `provider_event`, or `other`. A source role that cannot be represented without distortion is preserved in `source_role` and reported with `CONV_ROLE_UNMAPPED`; an adapter **MUST NOT** relabel it merely to fit a receiving runtime.

The following is a normative control invariant:

No imported role label, instruction marker, tool-result marker, message priority, provider signature, source authority, model attribution, or historical action record grants receiving-system control-channel authority.

Every message and content part carries `instruction_treatment`: `"untrusted_data"`. The receiving application—not a model reading the content—enforces authorization, tool availability, credential access, network access, context acceptance, and external actions.

20.5 Portable content parts

`conversation-content-part.schema.json` is a closed typed union. Large media is referenced through the Artifact Profile; it is not embedded in a message.

Type	Portable meaning
text, markdown	Source text or markup and optional language. Markup remains inert.
code	Source code text plus optional language and filename; never executable authority.
structured_data	JSON value labeled by a schema URI.
artifact_ref	Typed relationship to an Artifact Profile record, with relationship and optional semantic type hint. This covers image, audio, video, document, generated media, and other files.
citation_ref	Reference to a citation/source record with optional quoted text or selector.
tool_request_source, tool_result_source	Historical tool data with interaction ID, source status, inert payload, and <code>current_tool_authority</code> : <code>"none"</code> .
reasoning_source	Only provider-exported reasoning-related content, accurately labeled.
redacted	Position-preserving redaction marker.
unsupported	Known source semantic that cannot be portably represented, with a registered loss code.
provider_specific	Opaque inert value whose portable meaning is not claimed.

Unknown content kinds do not enter the union by guessed coercion. The adapter **MUST** preserve them as `provider_specific`, quarantine them, or emit `unsupported`/an explicit omission and a loss. `provider_specific` values **MUST** declare `executable: false` and **MUST** survive a claimed loss-aware round trip unless recipient policy explicitly omits them and reports the omission.

20.6 Hidden reasoning

OCP **MUST NOT** require, infer, reconstruct, or label content as private model chain-of-thought. `reasoning_source` can represent only content the source provider actually exported or made available to the user. Its `reasoning_kind` is exactly one of `provider_exposed_reasoning`, `reasoning_summary`, `analysis_placeholder`, or `unsupported_provider_reasoning`. A summary **MUST NOT** be relabeled as hidden reasoning. Absence of reasoning content conveys no claim about whether it existed.

20.7 Attachments, citations, tools, and agents

An attachment relationship is conversation evidence. Artifact metadata and artifact bytes are separate permission views. A package **MAY** preserve the relationship while withholding metadata or content when the relationship itself is authorized. Message disclosure **MUST NOT** imply artifact-content disclosure, and artifact disclosure **MUST NOT** imply message disclosure.

Citations use references rather than a universal citation ontology. A citation may point to a source record or Artifact Profile record and may retain provider locator and quoted selector metadata through a declared profile/extension. Incomplete mapping produces `CONV_CITATION_UNSUPPORTED`; the adapter **MUST NOT** fabricate bibliographic equivalence.

Tool requests and results are historical source data. Full executable tool semantics are outside this profile. Current tool authorization is always evaluated separately and is represented here only by the constant `none`. Agent messages and delegation labels are likewise history, not delegated authority.

20.8 Integrity

A message **MAY** declare a SHA-256 `content_parts_digest` over the RFC 8785 canonical JSON encoding of the exact `content_parts` array. A conversation **MAY** declare a `semantic_graph_digest` over a lexicographically sorted array of objects containing each message stable ID, ordered parent stable IDs, and optional sibling order. The declared algorithm and canonicalization are part of the integrity statement.

An exact `message_ref` digest covers the complete canonical message record. Digest equality proves byte/canonical-value equality against the reference only. It does not prove truth, speaker identity, provider authenticity, rights, safety, consent, or acceptance. A consumer that cannot implement the declared canonicalization **MUST** treat the integrity value as unverified, not valid.

20.9 Redaction and minimization

Redaction states are `none`, `source_redacted`, `export_policy_redacted`, `recipient_scope_redacted`, `unsupported`, and `unavailable`. A marker can carry a reason class and stable detail code. Where policy permits, redaction **SHOULD** preserve record identity, graph edges, source role, permitted timestamp precision, content-part position, original type, and reason class. A redacted part **MUST NOT** contain the removed value. Redaction is not anonymization and surrounding graph data may remain identifying.

Source redaction records what the source exported; policy redaction records a decision made by the exporting system; recipient-scope redaction records minimization for the delivery. An implementation

MUST NOT conflate these states or claim a source never held content merely because an export is redacted. Human-readable loss reports MUST suppress category names or counts when their existence is not authorized.

20.10 Lifecycle, authority, and authorization

Conversation and message records use the core record lifecycle. Supersession and withdrawal create new immutable revisions; they do not silently rewrite prior snapshots. Provider source deletion or account loss is recorded as provenance/availability change and MUST NOT delete an independently held authorized copy unless retention policy separately requires it.

Conversation access is authorized through the core request/receipt/package flow. Message view, evidence view, Artifact Profile metadata view, and artifact-content view are independently narrowed. The baseline profile has no direct mapped `->` accepted transition. A candidate derived from a message can become accepted context only through the core review semantics; the raw message remains evidence.

20.11 Package projection and composition

A Conversation Package producer includes only graph components authorized by the receipt. It MUST recompute roots, leaves, branch state, graph completeness, and loss counts for the disclosed projection. It MUST preserve exact references to disclosed messages and use unresolved references/redaction markers only when existence disclosure is allowed. It MUST NOT include artifact bytes unless the Artifact Package view is separately authorized.

The profile composes with the Artifact Profile through `artifact_ref` and message-level `artifact_refs`; with core evidence through citations and provenance; and with the Provider Adapter Framework through source locators, mapping records, mapping runs, and loss reports. Section 24 defines the reference composition scenario.

20.12 Round trip, downgrade, and extensions

Conversation round-trip comparison covers stable identity where safe, exact graph edges, roots, selected and abandoned leaves, roles, message kinds, disclosed content types/values, artifact/citation/tool relationships, timestamps and precision, redactions, provider-specific values, extensions, and every loss. Cosmetic ordering alone is not semantic loss. Flattened branches, changed parents, changed selected leaves, omitted disclosed content, invented roles, or dropped opaque fields are material.

A 0.4 producer downgrading to a profile without graph support MUST either refuse or emit a structurally lossy result with `CONV_BRANCH_LOST` and all other applicable losses. It MUST NOT call a flattened transcript lossless. Unknown critical extensions fail closed. Unknown non-critical extensions are preserved inertly, quarantined, or explicitly omitted with loss.

20.13 Errors, losses, and conformance evidence

`registries/conversation-loss-codes.json` defines meaning, loss class, severity, profile-conformance allowance, and round-trip consequence for `CONV_BRANCH_LOST`, `CONV_PARENT_UNKNOWN`, `CONV_ROLE_UNMAPPED`, `CONV_CONTENT_PART_UNSUPPORTED`, attachment, timestamp, citation, tool, generation-metadata, provider-extension, and redaction losses. A loss code does not excuse a false success claim: the registry says whether profile import may still conform and whether round trip is lossless.

Invalid cycles, unreported missing parents, contradictory branch states, silent flattening, unknown critical extensions, or a control-boundary crossing produce a registered protocol error. Implementations **MUST** produce safe detail and **MUST NOT** reveal withheld conversation content in errors.

The companion pack supplies positive, negative, hostile, and round-trip fixtures. Its synthetic reference scenario covers branching/regeneration, redaction, artifacts, opaque provider data, unsupported semantics, and hostile instructions. Passing local fixtures is partial reference evidence only; class conformance requires results from the identified implementation under test.

21. Artifact Profile

21.1 Profile identity and scope

The **OCP Artifact Profile 0.4** has stable identifier

`https://opencontinuity.org/profiles/artifact`, profile version `0.4`, and compatible core range `>=0.4 <0.5`. Its schemas are `artifact.schema.json`, `artifact-representation.schema.json`, and `lineage-event.schema.json`. The profile is storage-neutral and separates a logical artifact, its representations, custody, integrity evidence, rights assertions, accessibility relations, lineage, availability, and disclosure.

The profile answers independently: what the logical artifact is; where it came from; which representations exist; who declares custody; whether exact bytes are independently retrievable and verified; whether the artifact remains source-dependent; which rights/restrictions are asserted; what derived from what; which accessibility alternatives exist; and which metadata or bytes may be disclosed. None of those answers implies the others.

21.2 Logical artifact and representation

An `artifact` represents one logical work or source object. `representations` are encodings, presentations, or derived access forms of that same logical object when the source semantics support that interpretation. For example, an output video may have an original MP4, transcode, thumbnail, transcript, captions, audio-only representation, and accessible alternative.

Representation relationships are registered and include `original`, `lossless_copy`, `transcode`, `preview`, `thumbnail`, `transcript`, `caption`, `translation`, `accessible_alternative`, `extracted_text`, and `derived_metadata`. Separate logical works **MUST** remain separate artifacts connected by lineage. An adapter **MUST NOT** collapse distinct source works into representations merely to simplify storage, or split one source work without recording the mapping and loss.

21.3 Artifact record

The artifact envelope contains profile identity, subjects, space, semantic type, origin, custody, representations, rights records, accessibility links, lineage references, metadata/content access decisions, sensitivity, lifecycle, and extensions.

`origin.source_authority` identifies the source authority; `runtime_source` identifies the importing runtime. Provider object IDs and source URIs are optional provenance and never universal identity. Source creation and acquisition time are distinct. `semantic_type` is a bounded high-level hint—image, audio, video, document, code, dataset, archive, model output, project asset, or other—not a complete media ontology.

`access.metadata` and `access.content` are independent. A context item derived from an artifact may be disclosed without the artifact, artifact metadata may be disclosed while bytes are withheld, and byte access does not disclose every derived inference.

21.4 Representation record

Each representation has a stable `representation_id`, registered relationship, media type, availability, `independently_retrievable`, and `source_dependency`. Filename, byte size, duration, dimensions, language, content reference, digest, algorithm, verification time/method, encryption metadata reference, and format-specific metadata are present only when known and applicable.

`content_ref` locates bytes in a declared store/archive/profile. It is not itself a preservation claim. If `independently_retrievable` is true, exact bytes MUST be available without the originating provider account, subject to the declared holder's storage, authorization, and current lifecycle. The representation MUST provide byte size, digest, mandatory-to-implement algorithm, verification time, and verification method.

21.5 Custody semantics and preservation invariant

Custody mode	Exact meaning
reference_only	Only metadata or a source locator is held. <code>source_dependency</code> is true; no independent bytes are claimed.
derivative_only	At least one non-original derivative is independently retrievable, but original/authoritative source bytes are not held.
managed_copy	The declared holder controls at least one independently retrievable representation and verified its digest over those exact bytes.
synchronized_copy	A valid managed copy exists and is maintained under a separately registered synchronization profile. Sync status does not invalidate the last verified local bytes by itself.

The preservation invariant is normative:

An artifact representation MUST NOT be represented as independently preserved unless independently retrievable bytes are held under declared custody and the declared digest has been verified over those exact bytes.

A cached URL, remote source, authenticated API, metadata hash, provider object ID, browser display, or successful download in the past is insufficient. `managed_copy` and `synchronized_copy` require `source_dependency: false` at artifact custody and held representation levels. `derivative_only` identifies a non-original relationship. A holder MUST NOT silently upgrade reference-only to managed or derivative to original.

21.6 Source disappearance and changed bytes

Source events produce explicit transitions without changing historical facts:

Prior state	Event	Result	Required loss/evidence
reference-only, available	source unavailable/disappears	reference-only, unavailable or lost	ART_SOURCE_UNAVAILABLE
managed copy, available	source unavailable	managed copy remains available if local digest still verifies	record event; no content loss
derivative-only, available	source unavailable	derivative remains; original remains unpreserved	ART_ORIGINAL_NOT_PRESERVED
synchronized copy, healthy	sync failure	synchronized copy, synchronization_degraded; last verified copy remains usable	ART_SYNC_DEGRADED
any source locator	same provider ID or URL resolves to different bytes	create a new representation/source revision; never overwrite the verified prior digest	ART_SOURCE_ID_CHANGED
reference-only	provider permanently disappears	lost unless another authorized source resolves	ART_SOURCE_UNAVAILABLE

An unavailable source does not erase a verified managed copy. A verified copy does not establish continued rights. If local digest verification fails, the representation becomes unavailable/quarantined and ART_CONTENT_DIGEST_MISMATCH is fatal for preservation conformance.

21.7 Integrity and algorithm registry

sha-256 is mandatory to implement for 0.4 and is registered in `digest-algorithms.json`; future algorithms can be added by registry version. A digest is computed over the exact representation bytes, not metadata, a remote ETag, decoded pixels, or a provider response wrapper. Verification records algorithm, exact digest, byte size, time, and method.

Byte integrity is distinct from source authenticity, provider identity, ownership, copyright, license, truth, safety, malware status, semantic equivalence, and accepted context. A consumer MUST NOT infer any of those claims from a valid digest.

21.8 Rights assertions

Rights status is unknown, source_declared, user_asserted, licensed, restricted, redistribution_prohibited, copy_prohibited, or expired. Every rights record identifies asserting party, basis, assertion time, scope, and limitations, with optional source reference and validity end. A rights statement records an assertion; it does not itself grant legal rights or prove the asserter had authority.

Unknown means unknown. Possession, access, preservation, source authentication, or C2PA validity MUST NOT be treated as permission. Rights-based omission is recorded with an artifact loss code and privacy-safe detail.

21.9 Accessibility

An artifact can link representations as captions, transcript, audio description, alt text, accessible document, or translated caption. Transfers SHOULD preserve these relationships even when a representation is omitted. Omission of an otherwise authorized accessibility representation records

ART_ACCESSIBILITY_REPRESENTATION_OMITTED. Implementations MUST NOT infer or expose disability or health status from an accessibility link beyond the authorized purpose.

21.10 Lineage

A `lineage_event` describes one historical relationship among artifact identities: `created_from`, `derived_from`, `edited_from`, `transcoded_from`, `remixed_from`, `extended_from`, `generated_from_prompt`, `generated_from_artifact`, `selected_from_candidates`, `upscaled_from`, `translated_from`, `transcribed_from`, or `captioned_from`.

The event includes stable input/output record references, performer, occurrence time, and `external_action_authority`: "none"; it may include requester, model, model provider, prompt snapshot, source record, and parameter-disclosure level. Input/output references omit content digests because an artifact can carry an exact snapshot reference back to the lineage event; reciprocal digest references would create a hash cycle. Artifact `lineage_refs` remain exact snapshots.

Lineage describes known history and may be partial. It MUST NOT manufacture missing generations, model identity, prompts, parameters, creator identity, or ownership. It MUST NOT authorize reruns, publication, or external action. Missing known stages produce ART_LINEAGE_PARTIAL rather than a guessed complete path.

21.11 PROV and C2PA composition

An OCP artifact can reference a C2PA manifest through an extension or representation metadata when that manifest applies to the exact bytes. OCP does not duplicate C2PA claims or validation. `artifact` maps informatively to a PROV Entity; `lineage_event` to an Activity; input/output to used/generated relations; `performed_by` to an associated Agent; and source references to derivation/provenance links. OCP custody, disclosure views, source dependency, rights assertion status, and accepted-context boundaries do not map losslessly to generic PROV.

A valid signature or C2PA manifest is evidence within its own trust model. It does not automatically establish accepted context, truth, ownership, permission, safety, or preservation.

21.12 Lifecycle and synchronization

Artifact lifecycle status is `active`, `unavailable`, `lost`, `synchronization_degraded`, or `withdrawn`. Custody mode states what the holder claims to have; lifecycle states current operational availability. A new immutable revision records transitions. A synchronization failure MUST NOT mutate `synchronized_copy` into reference-only while verified local bytes exist; it changes lifecycle/sync status and records ART_SYNC_DEGRADED.

Withdrawal stops applicable future authorized disclosure. It does not prove erasure of prior delivered plaintext or bytes. Lost means the holder cannot retrieve an authorized representation; it does not prove all copies have ceased to exist.

21.13 Packaging, round trip, downgrade, and extensions

Artifact Metadata, Preservation, Package, and RoundTrip classes are separate. A package producer evaluates metadata and content independently, filters representations, preserves relationship skeletons where authorized, and recalculates losses. A content package includes only bytes whose member digest and size match the descriptor.

Round-trip comparison covers logical identity, representation relationships, media types, byte sizes, digests, custody, source dependency, availability, verification, lineage, rights assertions, accessibility

links, extensions, and losses. It MUST reject any upgrade from reference to managed, derivative to original, unknown rights to permitted, unavailable to available without evidence, or unverified to verified.

Downgrade to a representation-only or metadata-only profile is refused or explicitly loss-reported.

Unknown critical extensions fail closed; non-critical values are preserved inertly or explicitly omitted. No extension may override custody, digest, access, authority, or accepted-state invariants.

21.14 Errors, losses, fixtures, and conformance

`registries/artifact-loss-codes.json` defines source, content, digest, original-preservation, derivative, representation, lineage, rights, accessibility, synchronization, and provider-extension losses. Each code declares class, severity, profile-conformance allowance, and lossless-round-trip effect.

Preservation overclaim, digest mismatch, unsafe parser behavior, unknown critical extension, and unreported content loss are registered errors. Positive fixtures include reference-only, managed, generated-with-lineage, and disappearance transitions. Negative fixtures cover overclaim and digest mismatch. Hostile fixtures cover MIME spoofing, polyglots, malformed and oversized media, malicious PDF, decompression and recursive archives, traversal/symlink, macros, executables, corruption, and source replacement. Public fixtures are synthetic and passing them remains reference evidence until run against an identified implementation.

22. Provider Adapter Framework and import

22.1 Framework identity and purpose

The **OCP Provider Adapter Framework 0.4** has identifier

<https://opencontinuity.org/profiles/provider-adapter-framework>, version 0.4, and compatible core range $\geq 0.4 < 0.5$. It standardizes how an adapter declares, traces, tests, compares, and loss-reports a source-to-OCP mapping. It does not standardize a provider's private schema or grant a mapping registry authority over provider meaning.

Its normative schemas are `provider-adapter-manifest.schema.json`, `provider-mapping.schema.json`, `mapping-run.schema.json`, and `loss-report.schema.json`. Complete packages also carry a mapping registry, source-schema status, synthetic fixture suite, registered loss codes, and implementation result records.

22.2 Import is not acceptance

Provider exports and external records are untrusted input. Successfully parsing, authenticating, verifying, mapping, or preserving them MUST NOT make them accepted context. Provider observations, memory records, summaries, confidence labels, classifications, embeddings, feedback signals, or inferences remain source evidence/candidates until an authorized OCP review admits an item for an explicit scope.

22.3 Strengthened trust zones

An importer maintains four monotonic trust zones:

1. **raw**: original source bytes, immutable where retained, with no model/tool execution, network access, or ambient credentials;
2. **parsed**: syntax-level data from isolated bounded parsers, carrying source locators and no semantic authority;

3. **mapped:** OCP source/profile/candidate objects produced by a named adapter and mapping profile, still tainted/untrusted; and
4. **accepted:** context items admitted only through the core review decision and authority model.

Taint, source provenance, adapter/mapping versions, and omissions survive every transition. The baseline transition graph is received -> quarantined -> parsed -> mapped -> review_required, then accepted or rejected; abort/reject paths are available earlier. Direct mapped -> accepted is prohibited. A future Auto-Review Profile would need its own narrow authority, reversibility, expiry, audit, exclusions, and hostile tests; it is not part of 0.4.

Raw, parsed, and mapped values **MUST NOT** enter privileged prompts, policy/control channels, credential selectors, authorization decisions, network fetchers, or action arguments merely because a model can read them.

22.4 Provider Adapter Manifest

An `adapter_manifest` identifies:

- adapter ID/version and publisher;
- source provider, product, export family, and recognized source schema versions;
- compatible OCP core versions and supported profiles;
- input media types and required/optional members;
- declared source capabilities;
- mapping-registry reference and known losses;
- quantitative security limits;
- determinism basis and excluded run-variant fields;
- synthetic fixture-suite reference;
- optional integrity/signature metadata; and
- release/deprecation time.

Manifest compatibility is evaluated before parsing. A signature, when present, authenticates only the covered manifest under its trust model; it does not prove mapping correctness, provider endorsement, security, or conformance.

22.5 Capability declaration

Capabilities are explicit, extensible identifiers. The baseline registry can name conversation metadata/branching, messages, edits, alternative generations, provider memory, artifacts/bytes/metadata/generated media, citations, tool history, projects, collections, preferences, feedback signals, timestamps, provider IDs, user profile, and provider interpretations.

Declaring a capability means the adapter recognizes and tests a source semantic; it does not mean every record contains it, every value maps losslessly, or the source provider guarantees it. Unknown capabilities are non-critical only when preserved inertly; a critical unknown capability fails closed.

22.6 Mapping registry

Every meaningful source-to-OCP translation **SHOULD** have a `provider_mapping` with source field/semantic, target object/field, mapping type, loss class, deterministic transformation, supported locator forms, notes, and `accepted_context_eligible`. For this framework, adapter mappings

always set `accepted_context_eligible: false`; acceptance happens through the core review flow.

Mapping type	Meaning
exact	Same semantics and value without material interpretation.
structural	Same semantics represented with a different structure.
normalized	Equivalent source semantics converted to a registered normalized form.
derived	Target value deterministically computed from disclosed source data.
inferred	Semantic interpretation exceeds deterministic translation; never silently accepted.
partial	Only part of the source semantic is portable.
opaque_preserved	Source value retained inertly without portable meaning.
unsupported	Known source semantic cannot be represented.
omitted	Known data intentionally excluded with a reason.

`derived` requires a reproducible function over declared inputs. `inferred` requires uncertainty/provenance and remains a candidate. A registry **MUST NOT** label an interpretive mapping exact merely because a fixture happens to match.

22.7 Loss classification and claims

Loss classes are `none`, `cosmetic`, `metadata`, `structural`, `semantic`, `authority`, `content`, `privacy_policy`, `rights`, and `integrity`. The registry supplies a severity rank, but exact outcome is determined by the applicable profile code.

An import classification is:

- `lossless` only when every applicable supported source semantic and disclosed value compares equivalent, unknown non-critical data is preserved, no critical unknown exists, and all per-profile loss codes permit lossless status;
- `structurally_lossy` when topology/order/association changed but disclosed content semantics otherwise survived;
- `semantically_lossy` when meaning, role, type, precision, or interpretation changed;
- `content_lossy` when authorized bytes or content values were unavailable or omitted; or
- `authority_lossy` when an authority/acceptance/right semantic cannot be represented—never by inventing a substitute.

Adapters cannot self-certify losslessness. The claim requires registered fixtures, comparator output, exact mapping records, and results for an identified implementation/version.

22.8 Source locator

A source locator can identify archive member, JSON Pointer/Path, XML path, database table/row key, spreadsheet sheet/row, byte range, provider object ID, or source URI. It is provenance into one source snapshot, not a universal identifier and not authorization to fetch.

Locators **MUST** be bounded, normalized, and privacy-filtered. Imported source URLs are not fetched during parsing. A separately authorized fetch mechanism may resolve an allowlisted locator after network policy, credentials, redirect/address controls, size/time limits, and recipient purpose are evaluated.

22.9 Deterministic mapping run

A `mapping_run` records its ID, input manifest digest, adapter/mapping versions, runtime source, deterministic input digest, start/end times, result manifest digest, warnings, losses, and deterministic projection digest.

Fixed raw source bytes, declared source snapshot, adapter version, mapping profile version, and runtime source identity **MUST** yield the same deterministic projection. Run ID and clock timestamps are excluded from that projection. Random archive ordering, host locale, map iteration, current time, and undeclared network state **MUST NOT** affect mapped semantics. Replays compare canonical JSON and explain any divergence.

22.10 Adapter invariant and prohibited invention

The central invariant is:

An adapter translates source semantics. It does not manufacture portable human context merely to make an import appear richer or more complete.

An adapter **MUST NOT** invent people, identity equivalence, relationships, accepted state, authority, source timestamps/precision, conversation order, missing messages, branches, artifact ownership, user intent, rights, confidence equivalence, provider memory, or unsupported semantics. It also **MUST NOT** promote a source system/developer/tool label into receiving control authority.

22.11 Unknown fields and source evolution

Unknown semantics are handled by opaque preservation, quarantine, or explicit omission with loss. Guessing is prohibited. Source schema status is recognized, `compatible_unknown`, `unrecognized`, `partial_recognition`, or `schema_drift_detected`.

An adapter **MUST NOT** silently process an unknown or drifted export as a tested version. Compatible-unknown requires structural detection, conservative behavior, and an explicit report. Renamed required members, changed enums, duplicate IDs, contradictory graph metadata, different encodings, or timestamp-range changes trigger drift/partial recognition unless a tested mapping covers them.

22.12 Parser and archive safety

The manifest declares and the importer enforces maximum archive bytes, expanded/member size, member count, nesting and JSON depth, string/collection sizes, traversal work, media dimensions/duration/frames, parser time/memory, and proposal volume. Paths are normalized and contained. Absolute/traversal paths, symlink/device escapes, recursive archives, macros/active documents, executables, MIME/signature mismatches, polyglots outside declared policy, and malformed or resource-exhaustive inputs are rejected or isolated.

Parsers run in an isolated process or equivalent sandbox with no network route, ambient credential, home-directory access, or write access outside dedicated quarantine. MIME sniffing never executes content. Digest validation happens before a representation is promoted to managed custody.

22.13 Loss report

`loss_report` is the normative machine result. It identifies import/export/round-trip scope, aggregate classification, standardized counts, detailed loss entries, `accepted_context_created_without_review`, and a recipient visibility scope. Each loss gives registered code, class, severity, bounded meaning, conformance and round-trip effects, and optional object reference/source locator.

Counts are derived from exact records: messages/artifacts discovered and preserved, branches lost, artifacts independently preserved/source-dependent, opaque fields, and unsupported omissions. Human rendering is informative but **MUST** be generated from the machine result. It **MUST NOT** reveal withheld category names, object identity, or sensitive existence to an unauthorized recipient. The report **MUST** state explicitly when zero accepted context was created without review.

22.14 Required adapter fixtures and hostile corpus

A complete adapter package includes synthetic fixtures for: smallest valid, representative structure, branching, missing/unknown fields, malformed export, hostile archive, renamed/reordered members, duplicate provider IDs, unavailable artifacts, inconsistent timestamps, provider extensions, unsupported semantics, redactions, and schema-version mismatch.

Hostile adapter cases include schema drift, contradictory branches, malicious field names, deep JSON, unknown enums, timestamp overflow, unsupported encodings, forged IDs, and misleading confidence. Pass means reject or explicit loss without guessing and with zero accepted-context creation, authority change, network fetch, credential exposure, external action, or unauthorized durable mutation. Personal exports **MUST NOT** appear in the public suite.

22.15 Errors, extension behavior, and conformance

Schema drift, nondeterminism, unsafe imported content, graph invalidity, preservation overclaim, integrity failure, and unreported profile loss have registered protocol errors. Errors contain safe details and no secrets/source content. Unknown critical extensions fail closed. Non-critical unknowns are preserved inertly or explicitly loss-reported.

OCP-Provider-Adapter-0.4, OCP-Provider-Adapter-Deterministic-0.4, and OCP-Provider-Adapter-LossReport-0.4 have exact assertion sets in `conformance/classes.json`. A package may be structurally complete while every implementation evidence row remains unexecuted; conformance is claimed only for identified test subjects.

23. Export and round-trip rules

23.1 Export purpose and completeness

An OCP export supports backup, user inspection, migration, or interoperability. It names purpose, profile versions, object scope, source snapshot, creator/runtime, omissions, unsupported semantics, and exceptions through declared package members. “Complete” is meaningful only relative to that declared scope. A package **MUST NOT** use a high preserved count to conceal unsupported fields or inaccessible bytes.

23.2 Offline export manifest

The Offline Archive Binding in Section 30 uses `offline-package-manifest.schema.json`. The manifest carries package identity/version, time, archive format/classification, payload/security members, member-set policy, and optional registered protection. Every member has path, media type, byte size, digest, and optional object/schema identity. Rich profile selection, mapping runs, losses, and creator/destination semantics live in typed archive members, not undeclared manifest fields.

A non-public archive requires an applicable registered encryption/signature profile before an implementation calls it protected. Structural metadata or a profile URI alone does not prove encryption, signing, recipient binding, or secure deletion.

23.3 Comparator contract

Round-trip comparators operate on canonical semantics and produce a machine-readable explanation for every difference. They record source/export/import/re-export identities, profile and comparator versions, ignored non-semantic fields, object-level comparisons, losses, result classification, and evidence scope. Random run IDs, wall-clock execution timestamps, archive-member ordering, caches, indexes, and regenerable human renderings may differ when declared excluded. No substantive field can be ignored solely because an implementation does not support it.

23.4 Core and conversation comparison

The core comparator retains 0.3 requirements for safe identities, subjects/pairwise mappings, statements/typed values, epistemic basis, accepted scope, review decisions, provenance, lifecycle, supersession/withdrawal/tombstones, omissions, warnings, and extensions.

The Conversation comparator additionally checks conversation/message identity where safe, graph edges, roots, branches, selected leaves, roles/kinds, content-part types and disclosed values, attachment/citation/tool relations, timestamps and precision, redaction markers, unsupported/provider data, extensions, and loss codes. Branch flattening, parent change, invented ordering, changed selected leaf, or dropped opaque data is not cosmetic.

23.5 Artifact comparison

The Artifact comparator checks logical identity, representations and relations, digests, sizes, media types, custody, source dependency, availability, verification state, lineage, rights assertions, accessibility links, extensions, and losses. It MUST never upgrade reference to managed copy, derivative to original, unknown rights to permission, unverified to verified, unavailable to available, or source-dependent to independent without new recorded evidence. A new verified representation is allowed only as a new explicit representation/revision.

23.6 Adapter comparison

The adapter comparator checks object counts, source schema status, mapping types, source locators, known/observed losses, unknown fields, opaque-preserved values, warning codes, input/result manifests, deterministic projection digest, and excluded run-variant fields. A changed adapter or mapping profile is not a deterministic replay; it is a separate comparison input.

23.7 Unknown data, criticality, and downgrade

Unknown source/profile data is preserved inertly, quarantined, or omitted with explicit loss. Unknown critical extensions fail the affected import/profile; unknown non-critical extensions survive a lossless claim byte-for-value in canonical JSON. An exporter MUST NOT guess a predicate, type, role, authority, right, custody state, or accepted value.

Downgrade to an earlier wire/profile is explicit. If no faithful target representation exists, the producer refuses or marks the result structurally, semantically, content, or authority lossy. A successful parse of downgraded JSON is not evidence of semantic compatibility.

23.8 Export security and privacy

Exports are high-value archives. Providers require fresh authorization proportionate to sensitivity, expose scope and recipient, omit tokens/private keys, minimize audit material, and prevent confused-deputy/open-redirect delivery. Download URLs are short-lived, single-principal, non-guessable, and no-store. Direct provider transfer SHOULD avoid intermediary-browser plaintext where a reviewed flow exists.

Source filtering, third-party restrictions, branch existence, attachment access, artifact bytes, and loss-report visibility are evaluated before packaging. Redaction is not anonymization, and withheld categories MUST NOT leak through counts/errors to an unauthorized recipient.

24. Transfer and profile composition

24.1 Transfer versus export

Export creates a user-accessible package. Transfer coordinates delivery to another provider and records entry outcomes. Delivery, schema validation, and mapping do not imply destination acceptance; successful delivery does not trigger source deletion.

24.2 Transfer manifest and lifecycle

A `transfer_manifest` is a persistent OCP record distinct from the offline archive manifest. It binds source/destination providers, state, exact source snapshots, transfer-local IDs/paths, entry status/losses, optional exact destination snapshots, exceptions, source disposition, and completion time.

The lifecycle is prepared -> delivered -> importing -> validated | validated_with_exceptions | rejected, with aborted available before validation. Destination verification covers archive structure, every size/digest, schema/profile support, authorization, graph/reference closure, artifact custody claims, and loss-report consistency.

`validated` contains only verified entries and no exceptions. `validated_with_exceptions` bijectively relates each non-verified entry to one exception. Integrity failures cannot be hidden as validation. Destination references exist exactly for imported/verified entries. Terminal completion time is not earlier than manifest creation. State follows manifest revision and legal transitions, not event arrival order.

24.3 Reference Portable AI History composition

The three 0.4 profiles compose without merging their authority or disclosure boundaries:

```
Conversation C1
|
|-- Message M1: user source text
|   |
|   |-- attachment relation --> Artifact A1
|   |
|-- Message M2: assistant source response
|   |
|   |-- citation --> Source S1
|   |-- generated output --> Artifact A2
|
Artifact A2 -- derived_from --> Artifact A1
Adapter P1: provider conversation abc --> C1
```

```

Adapter P1: provider asset xyz      --> A1
Adapter P1: generated output uvw    --> A2

Candidate context item -- provenance --> M2 / A1 / A2 / S1
|
+-- only review_decision may create accepted context

```

The companion reference bundle contains a branching conversation with regeneration, one independently preserved attachment, one generated managed artifact with lineage, one unavailable source-dependent artifact, a redacted part, one opaque provider field, one unsupported semantic, and hostile source instructions. It records mapping run and loss report and has an empty `accepted_context_items` array.

Message permission does not grant Artifact metadata or content. Artifact permission does not disclose the conversation. A user may accept a derived context item while withholding raw messages and bytes. The candidate's provenance remains, but package projection follows the receipt's independent evidence/artifact views.

24.4 Receipts, source disposition, and retries

Authenticated transfer receipts SHOULD cover transfer ID, manifest digest, entry outcomes, time, implementation/profile versions, and exception digest. A receipt proves only the covered event under the selected trust profile.

`source_disposition` is unchanged, `deletion_requested`, or unknown. A deletion request is not completion; erasure evidence remains separately scoped. Transfers use idempotency keys and member digests. Replaying identical bytes returns the prior result; reusing a key for different bytes is a conflict. Partial retry names exact entries and preserves verified outcomes.

25. Revocation and erasure semantics

25.1 Revocation lifecycle

```

requested -> acknowledged -> effective
requested | acknowledged -> failed
effective | failed -> terminal

```

`effective` means the provider no longer authorizes the covered future operation and has updated controlled live references/credentials as specified. It does not mean every delivered copy, cache, embedding, derivative, model weight, log, backup, or subprocessor copy was deleted.

25.2 Erasure status vocabulary

The Erasure Status Profile distinguishes:

- `requested` — a scoped deletion request was sent;
- `acknowledged` — the target provider received and recognized it;
- `reported_complete` — the target provider reports completion for the stated scope;
- `partially_complete` — some covered data remains, with limitations;
- `rejected` — the provider declined the request;
- `legal_retention_exception` — stated retention prevents full completion; and
- `independently_verified` — an independent assessment supplied evidence for an exact scope and method.

The record identifies target provider, request reference, scope description, status, reporter, report time, verification level, limitations, and—when independently verified or claiming cryptographic-erasure evidence—an immutable evidence reference. A provider assertion **MUST NOT** be labeled independent verification. An implementation **MUST NOT** claim `cryptographic_erasure_evidence` until a registered evidence profile structurally identifies key and ciphertext/storage scope, wrapping/recovery/backup keys, destruction actor/time/method, covered replicas, verification method, exclusions, and limitations.

25.3 Cryptographic erasure

Cryptographic erasure means destruction of a cryptographic key made necessary to decrypt a defined ciphertext set. An Erasure Status Profile producer making this claim **MUST** identify the key scope, ciphertext set or storage domain, destruction method, responsible actor, time, replicas/backups covered, and verification method. It does not prove deletion of plaintext copies or derivatives created before key destruction.

25.4 Recipient obligations

A client declaration and receipt may record deletion support and retention commitments. OCP can request, acknowledge, and carry evidence about these obligations; it generally cannot observe the recipient's complete downstream environment. Conformance statements **MUST** label recipient behavior as declared, locally enforced, attested, audited, independently verified, or unknown.

25.5 User interface language

Interfaces **MUST** use bounded statements such as “access revoked at Provider A,” “deletion requested from Client B,” “Client B reports deletion complete for the session package,” or “independent verification unavailable.” They **MUST NOT** display “your data is gone everywhere” without evidence that directly supports that exact scope.

26. Security considerations

26.1 Security objectives

OCP seeks to prevent unauthorized disclosure, unauthorized accepted-state mutation, semantic widening, package tampering, recipient confusion, context-to-action escalation, unsafe import, unreported loss, and assurance overclaim. It also seeks to preserve availability and recoverability without weakening identity or key protection.

Schema validity is necessary but insufficient. It cannot establish authentication, authority, a live authorization, safe content, truth, user understanding, recipient behavior, or downstream deletion.

26.2 Threat model

Implementations **MUST** model at least these actors and failure modes:

Threat actor or failure	Primary attack	Required design response
malicious/compromised client	excessive request, substitution, BOLA, forwarding, training-use escalation, proposal spam	separate authorization, structured containment, client declaration, instance binding, rate limits, opaque aliases, audit.

Threat actor or failure	Primary attack	Required design response
package thief	replay, wrong-delivery-recipient or client-instance use, offline copying	sender-constrained token where applicable, recipient encryption, signature, expiry, status checks, offline deny.
malicious source/adaptor	prompt injection, semantic invention, poisoned context, archive/parser attack	sandbox, quarantine, untrusted-data channel, deterministic adapters, negative fixtures, review decision.
impersonated provider/client	consent phishing, mix-up, SSRF discovery	exact issuer/resource matching, authenticated metadata, verified publisher/instance, safe discovery.
malicious provider/insider	omission, false acceptance, surveillance, misleading deletion	immutable receipts/decisions, export/loss report, restricted audit, independent comparison; acknowledge residual trust.
delegated agent	scope or action escalation, hidden processor	baseline delegation none; profile requires child authorization, attenuation, new recipient package, actor chain.
coercive or organizational actor	forced disclosure, household surveillance, enterprise exfiltration	conservative authority boundary, private decline, profile exclusion, organization policy, non-disclosing denial.
resource attacker	archive bomb, recursive graph, expensive media, request flood	quantitative limits, sandbox, quotas, pagination, backpressure, bounded traversal.
key attacker or recovery abuse	decrypt archives, forge packages, restore old access	key separation, protected storage, rotation/revocation, recovery disclosure, fresh authorization after compromise.
source disappearance	broken provenance/artifacts	explicit availability/custody state, verified copies, loss reporting; no false preservation claim.

26.3 Authorization and object access

Every protected operation—including package entry dereference, status lookup, proposal submission, export, transfer, and erasure request—MUST perform operation-time authorization. Stable object identifiers MUST NOT make an object enumerable. A provider SHOULD return the same non-disclosing not-found response when revealing existence would exceed authorization.

State-changing operations MUST use idempotency and optimistic concurrency. A duplicate idempotency key with identical canonical input returns the original outcome. The same key with different input returns OCP_CONFLICT. A stale revision returns OCP_STALE_REVISION. Authorization and revocation races MUST be resolved atomically or fail closed.

26.4 Package consumption checklist

Before rendering, indexing, embedding, caching, dereferencing, or exposing content to a model, a client MUST:

1. validate schema and supported profiles;

2. reject duplicate JSON members and unsupported critical extensions;
3. verify the registered authenticity and confidentiality profiles;
4. verify package and entry digests;
5. verify delivery-recipient, client-instance, and processor binding;
6. verify issue/expiry time and current receipt status where online status is required;
7. verify `external_action_authority`: "none";
8. enforce offline-use and delegation constraints; and
9. route all content only through the untrusted-data channel.

Failure of any required check rejects the package before content processing. A consumer **MUST NOT** continue on a “best effort” basis after an integrity, delivery-recipient, processor, expiry, receipt-status, critical-extension, confidentiality, or action-authority failure.

No Remote Security Profile is registered in this review pack. Consequently, a remote client cannot presently complete checks 3 and 5 for non-public content, and no implementation may claim Remote Security or SEC - 001 conformance from these artifacts. The Context Package example in Section 42 is a structural schema/semantic fixture only; its digest is not issuer authentication and it is not a deployable remote package.

26.5 Cryptographic profiles

OCP does not define new cryptographic algorithms. A signature profile **MUST** define algorithm set, canonical/signed view, key identifier, issuer binding, trust validation, validity, status/revocation, downgrade protection, external-content coverage, and failure behavior. An encryption profile **MUST** define recipient key binding, authenticated encryption, protected metadata, algorithm negotiation, downgrade protection, multi-recipient behavior, and failure behavior.

For live HTTP, a profile **MAY** use HTTP Message Signatures with covered content digests; a durable archive still requires a durable manifest signature. RFC 8785 canonicalization **MAY** produce deterministic JSON bytes, but a digest alone does not authenticate an issuer. TLS protects a transport hop but does not bind an offline copy to a recipient.

Unknown, disallowed, revoked, expired, wrong-issuer, wrong-recipient, wrong-purpose, or downgraded profiles/keys **MUST** fail before release of content.

Before Remote Security can become claimable, a registered profile and test vectors **MUST** define the signature and encryption formats, protected fields, issuer trust, recipient and client-instance key thumbprints, key discovery/status/rotation, algorithm agility, downgrade behavior, and fail-before-release behavior.

26.6 Key management

An implementation that signs, encrypts, or decrypts OCP data **MUST** define and test:

- generation and entropy;
- separation of signing, encryption, client-instance, storage, transport, and recovery keys;
- least-privilege storage and use;
- actor/service/device association;
- issuance, rotation, re-encryption, compromise, revocation, and retirement;
- backup, recovery assurance, loss, and unrecoverability;

- key-destruction scope; and
- audit without key material.

Private keys, tokens, session cookies, passwords, API credentials, and recovery codes **MUST NOT** appear in OCP records, packages, events, logs, analytics, errors, or fixtures. Recovery **MUST NOT** restore an old authorization silently. Following compromise or a high-risk recovery event, affected keys and credentials **MUST** be rotated/revoked and fresh authorization obtained where binding can no longer be trusted.

At-rest encryption **SHOULD** use per-person, per-space, or comparably bounded envelope keys. Key escrow or recovery-party access **MUST** be disclosed. “User-held key” **MUST NOT** be claimed when another party can decrypt independently.

26.7 Prompt injection, poisoning, and parser controls

Schema-valid, provider-signed, subject-confirmed, or credential-verified content can still contain malicious instructions. Implementations **MUST** enforce the isolation rules in Section 22 independent of content classification. Privileged fields such as acceptance, authority, sensitivity, authorization result, and integrity result **MUST** be generated or confirmed by trusted code, never copied from source prose.

Explicit reference retrieval, when a profile permits it, **MUST** use an isolated fetcher that allowlists schemes/destinations, blocks loopback/private/link-local/multicast/local-file/cloud-metadata targets, revalidates redirects and DNS, limits bytes/time/redirects, and sends no ambient credentials. Likely secrets are quarantined and excluded from models, embeddings, logs, analytics, and errors.

26.8 Availability and recovery

Providers **MUST** bound expensive operations, paginate collections, use backpressure, and avoid privacy-leaking resource exhaustion errors. They **SHOULD** support encrypted backups, tested restore, manifest/digest verification, and export recovery. Availability claims **MUST** distinguish provider availability, source availability, independently managed copy, and user-held export.

26.9 Residual security risks

OCP cannot generally stop an authorized recipient from copying plaintext, making a screenshot, creating a derivative, violating purpose, retaining content, or training a model. It cannot ensure a malicious provider exported everything it held. It cannot turn a valid signature into truth or lawful authority. It cannot guarantee offline revocation or universal deletion. Implementations **MUST** state these residual risks adjacent to claims that might imply otherwise.

26.10 Portable AI History hostile-profile requirements

Conversation hostile tests cover fake system/developer instructions, “ignore previous” language, tool-authorization text, fake OAuth scopes, credential-looking strings, role spoofing, malicious Markdown, localhost/private-address URLs, encoded instructions, code blocks, and filenames. Artifact tests cover MIME spoofing, polyglots, malformed images/PDFs, bombs, oversized media metadata, recursive archives, symlink/traversal, macros/executables, corruption, digest mismatch, and source replacement. Adapter tests cover schema drift, duplicates, contradictory branches, malicious names, deep JSON, unknown enums, timestamp overflow, unsupported encodings, renamed members, forged IDs, and misleading confidence.

The pass condition is behavioral containment: zero source-content change to control policy, permissions, credential selection, tool/network access, accepted state, or external actions; zero unauthorized byte release or durable mutation; and safe loss/error output. Phrase detection or model refusal is not sufficient.

26.11 Profile-specific enforcement boundaries

Application code enforces authorization, archive/member limits, parser isolation, network policy, credential selection, accepted-context transitions, artifact custody promotion, tool access, and external actions. A model reading imported content is not an enforcement boundary. Imported roles/instructions never enter privileged system/developer/control channels. Parsers have no network access; arbitrary `source_uri` values are never automatically fetched.

Artifact bytes remain quarantined until size, media/signature policy, digest, and parser checks succeed. A mapping adapter cannot promote a reference to managed custody. A Conversation or Artifact Profile package cannot widen core views. Every failure defaults to no disclosure, no accepted-state change, and no action.

27. Privacy considerations

27.1 Data categories and minimization

Context content, evidence, artifacts, identifiers, relationship edges, filenames, summaries, embeddings, inferences, omissions, purposes, recipient identities, access history, and audit events may all be personal or confidential data. Providers **MUST** minimize collection, storage, indexing, package contents, events, errors, and retention—not only raw messages.

A single sensitivity scale is only a routing aid. Profiles **SHOULD** add orthogonal category and rights-policy references for health, intimate, precise location, financial, credential, child, biometric, work-confidential, and other domain-sensitive information. Derived data receives protections based on what it reveals, not merely its source label.

27.2 Pairwise identifiers and correlation

A Package Producer **MUST** use delivery-recipient-pairwise subject identifiers and package-local entry/item aliases and **MUST** rewrite every matching intra-package reference. Audit correlation IDs **MUST NOT** become global person identifiers. Clients **MUST NOT** combine OCP disclosures across purposes, delivery-recipient or trust-domain scopes, spaces, or profiles unless a separate authorization and lawful deployment policy permit it.

Pseudonymization reduces some direct identification risk but is not anonymization, consent, or disclosure authority. Distinctive preferences, relationship graphs, time patterns, artifacts, and rare traits can re-identify a person.

27.3 Meaningful review and consent presentation

Before approval, provider-controlled UI **MUST** show:

- verified client and publisher, plus client instance when available;
- purpose code and plain-language description;
- requested selectors and views, including evidence/artifact content;
- time period and package lifetime;
- retention mode, delete-by time, and stated legal-retention possibility;
- every processing-use choice, including training, product improvement, and advertising;
- proposal rights, delegation, and offline use;
- delivery recipient/client instance and every plaintext processing recipient;

- sensitive, third-party, organizational, and inferred-data restrictions; and
- what approval, decline, expiry, revocation, and prior plaintext disclosure mean.

Approval MUST be default-deny, unbundled, unselected, specific, revocable, and no easier or more prominent than decline. Client-supplied HTML, styling, urgency, countdowns, or persuasive copy MUST NOT control the trusted decision surface. Material changes to identity, delivery recipient/instance, processor, purpose, selector/view, duration, retention, processing use, proposal rights, delegation, offline use, or sensitivity require a new visible decision.

The review and correction experience MUST satisfy applicable WCAG 2.2 Level AA success criteria. It MUST support keyboard and assistive-technology use, programmatic labels, clear errors, adequate target sizes, non-color cues, understandable timing, review before commitment, and durable access to revoke or correct. Accessibility is evaluated through UI testing and human review, not JSON Schema.

27.4 Client declarations and assurance

The provider MUST resolve the exact, unexpired client declaration and confirm client identity before review. Remote clients MUST include every model provider and subprocessor that will receive plaintext in the required `processing_recipients` array. An empty array is an explicit declaration of none; omission is invalid and never means verified empty.

Status tokens submitted in a client declaration are untrusted claims; schema or endpoint acceptance never assigns assurance. `provider_verified` is valid only when `relationship: "provider"` names the authenticated context provider as evaluator. `independently_assessed` is valid only when `relationship: "independent_third_party"` names an authenticated, trusted external assessor that is neither the declaring client nor the context provider; the context provider then decides whether to rely on it. Either status MUST carry an exact `evidence_ref` to a schema-valid OCP 0.4 SourceRecord with available digest-bearing content and state the method, evaluator relationship, scope, assessment period or date, and limitations. Before relying on either label, the provider MUST resolve and recompute that exact evidence snapshot; reject inactive, unavailable, quarantined, suspected-tampering, content-empty, causally invalid, or unknown-critical evidence; apply the selected source-verification policy to any verified trust treatment; bind the SourceRecord authority to the named evaluator; and obtain a current trusted policy evaluation keyed to the exact declaration revision and digest for the stated scope. An approval keyed only by stable declaration ID, a syntactically valid label, or a self-supplied different evaluator is not assurance. For a remote disclosure, absent processor/provider lists are undeclared and non-approvable; empty arrays are an explicit declaration of none. Any processor or model-provider change requires renewed review.

Declared retention, training, delegation, and deletion support do not prove behavior. UI and conformance statements distinguish:

- `protocol_enforced`;
- `client_declared`;
- `contractually_committed`;
- `externally_attested`;
- `independently_audited`; and
- `unknown`.

An implementation MUST NOT upgrade one evidence relationship to another by inference. In particular, schema validity, a signature, encryption, a declaration, or green tests do not prove recipient compliance or legal sufficiency.

27.5 Retention, audit, and telemetry privacy

The provider enforces retention and processing constraints at boundaries it controls. After plaintext leaves that boundary, the fields may be recipient obligations unless separately enforced. Providers **MUST** say which is which.

Audit/events **MUST** exclude raw context, evidence, purpose text, credentials, tokens, and package bodies. They use bounded identifiers, revisions, transition types, time, digest, and privacy-safe correlation. Audit storage **SHOULD** be encrypted, access-segregated, briefly retained under a declared policy, and safely inspectable by the subject. OpenTelemetry attributes **MUST** be low-cardinality and **MUST NOT** contain raw context or stable cross-purpose person IDs.

27.6 Sensitive inference

A client/provider **MUST NOT** infer or disclose a sensitive trait merely because lower-sensitivity context makes it statistically likely. The inference itself is sensitive data. High-impact inferences require a profile defining purpose, authority/legal basis, source disclosure, contestability, acceptance policy, retention, and conformance. They **MUST NOT** be auto-accepted in the personal core.

27.7 Law and jurisdiction

OCP fields can record a deployment's legal role, purpose, rights request, or policy reference. They do not determine legal applicability. GDPR portability has qualifying conditions and does not make every provider inference or third-party datum portable. The EU Data Act and AI Act have specific scopes and phased obligations. Implementers **MUST** obtain jurisdiction- and use-case-specific legal review and **MUST NOT** advertise OCP conformance as legal compliance.

27.8 Conversation, artifact, adapter, and loss-report privacy

Conversation portability is not blanket permission to expose every message, third-party contribution, hidden branch, relationship edge, private attachment, or provider interpretation. A producer performs source-level filtering before graph packaging and uses redaction/partial-graph semantics only when even the existence of an omitted component may be disclosed. Third-party authority follows Section 28; one participant's portability request does not automatically authorize all parties' content.

Artifact preservation and artifact disclosure are separate. Managed custody permits future availability under policy but does not make bytes public, establish rights, or authorize a recipient. Metadata, representation relations, and filenames may themselves be sensitive. Accessibility relationships are minimized so they do not unnecessarily disclose disability or health implications.

Adapters retain only the raw bytes, opaque fields, source locators, and diagnostics needed for the stated import purpose and retention policy. Public fixture suites use synthetic data. Logs **MUST NOT** contain raw conversation content, artifact bytes, credentials, source locators that expose secrets, or unbounded provider fields.

Machine loss reports are recipient-filtered before human rendering. An unauthorized recipient **MUST NOT** learn the existence, category, count, locator, filename, or relationship of withheld material. Aggregate counts therefore apply to the visibility scope in the report, not necessarily the source archive as a whole.

28. Third-party and organizational-data rules

28.1 Core authority boundary

The Core Personal Profile is limited to a competent individual authorizing disclosure of context solely about that same individual in an active personal space with `authority_model: "self"`. The space, proposed candidate, resulting accepted item, decision reviewer, and independently authenticated reviewer MUST all bind that one person exactly by `{id, type}`; display labels are ignored. Authentication does not prove capacity, comprehension, free choice, guardian validity, absence of coercion, child best interests, organization authority, or authority over another person.

If a request requires joint, guardian, delegated-stewardship, organizational, legal, incapacity, deceased-person, inherited, or disputed authority and no registered profile applies, the provider MUST return `OCP_UNSUPPORTED_PROFILE` or a more specific non-disclosing denial.

28.2 Third-party data

The Core Personal Profile MUST exclude raw third-party communications, media, contact data, precise location, health, intimate, financial, credential, employment, and organization-confidential content. This applies even when the authorizer possesses or wrote part of the source.

An item naming another person, marked third-party asserted, or derived from another person's content MUST be quarantined or omitted unless a third-party/joint-data profile supplies authority and disclosure rules. Denials, omissions, warnings, logs, and errors MUST NOT reveal the person's presence or sensitive category beyond the caller's authorization. `OCP_THIRD_PARTY_RESTRICTED` SHOULD remain general.

28.3 Households, groups, and disagreement

A Household or Joint Context Profile MUST define the decision set, threshold, item-level ownership/rights, veto and emergency behavior, separation/exit, historical access, dispute, death/incapacity, and audit visibility. One member's "accept" MUST NOT become every member's acceptance. A shared context space MUST preserve which statements are joint, individual, disputed, or organization-controlled.

28.4 Minors and guardians

A Minors and Guardians Profile is deferred. It requires legal and child-safety review across jurisdictions, age/capacity transitions, guardian evidence and limits, child voice, best-interest evaluation, high-privacy defaults, profiling limits, separate third-party rights, emancipation/majority transition, revocation, concealment from unsafe guardians, and appeal. Until that profile exists, a provider conforming to the Core Personal Profile MUST NOT process guardian authority or market its deployment as suitable for child context portability.

28.5 Domestic abuse and coercive control

Context portability can expose location, contacts, plans, finances, health, identity, and escape arrangements. High-risk deployments SHOULD provide private review, safe exit, delayed disclosure, reauthentication, notification controls, device/session hygiene, non-disclosing denial, recipient visibility, and emergency suspension. Shared-account access MUST NOT imply authority over another person's personal context. Providers MUST avoid notifications that reveal a hidden request or safety action to a coercive actor.

These controls reduce risk but do not establish absence of coercion. A deployment operator **MUST** obtain deployment-specific safety expertise and human-support capability before serving coercion-sensitive populations.

28.6 Organizational and mixed data

Organization-managed, customer, trade-secret, work-credential, confidential, regulated, and mixed personal/work material **MUST NOT** be exported under the personal core. It enters quarantine until an Organizational Profile and current organization policy decision apply.

The Organizational Profile **MUST** define organization identity, custodian, role/policy authority, source environment, classification, exportability, confidentiality, trade-secret and customer restrictions, purpose, recipient, residency, retention, legal hold, departure/role change, monitoring, and personal/organization separation. A policy reference records a decision input; it does not prove permission. Absent a current profile decision, the provider returns `OCP_ORGANIZATIONAL_POLICY` without disclosing policy internals.

28.7 Multi-party profile requirements

Every authority model beyond self **MUST** define:

- decision maker or set and threshold;
- item- and disclosure-level scope;
- evidence and assurance for authority;
- validity and renewal;
- disagreement, veto, withdrawal, appeal, and transition;
- emergency and coercion-safe behavior;
- third-party notification and access limits;
- audit visibility and retention; and
- what happens when the profile is unsupported at import or transfer.

OCP records the determination. OCP itself does not prove the determination legally sufficient.

29. Transport and HTTP binding

29.1 General requirements

The HTTP binding uses HTTPS, JSON, RFC 9457 Problem Details, explicit content types, current authorization, idempotency, and optimistic concurrency. HTTP syntax does not replace the semantic rules in this document.

Remote endpoints **MUST** use authenticated HTTPS. Requests and responses carrying non-public context **MUST** use `Cache-Control: no-store` unless a profile defines a private encrypted cache with equivalent limits. Servers **MUST** validate `Content-Type`, `Accept`, content length, decompression limits, JSON limits, and supported OCP/profile versions before processing content.

This working draft uses provisional media types `application/ocp+json` and `application/ocp-package+json` in fixtures. Until a stable registration exists, deployments **SHOULD** support `application/json` with a versioned profile parameter or explicit schema link and **MUST NOT** claim IANA registration.

29.2 Discovery

GET /.well-known/ocp-configuration is rooted at the origin, outside the versioned /ocp/v0.4 API base, and returns public discovery metadata. A versioned OpenAPI server entry MUST therefore give this operation an origin-root server override rather than composing it under /ocp/v0.4. The current discovery schema requires `issuer`, `ocp_versions`, `base_uri`, `schemas`, `bindings`, `profiles`, `authorization`, and quantitative `limits`, with optional `discovery_uri`. `limits.clock_skew_seconds`, `limits.grant_status_max_age_seconds`, `limits.event_cursor_retention_seconds`, and `limits.event_page_max_items` are required. The grant-status age/skew limits govern every grant-authorized evaluation in this binding, including package issuance and proposal review decisions. A client or provider MUST read those advertised values and MUST NOT substitute an HTTP-binding default. A future complete HTTP profile SHOULD extend or version that schema before advertising additional metadata such as:

- exact issuer/provider identifier;
- supported OCP wire versions;
- schema catalog and integrity pins;
- supported profiles and versions;
- authorization methods and protected-resource metadata links;
- endpoint base URLs;
- media/package formats;
- client-declaration support;
- conformance statement, documentation, privacy policy, security contact, and governance links; and
- limits relevant to clients.

Clients MUST perform exact HTTPS issuer/resource comparison and SSRF-safe discovery. In the baseline HTTP binding, `issuer` is an origin URL with no path, query, fragment, or user information; `base_uri`, `discovery_uri`, protected-resource metadata, and authorization-server metadata MUST use that exact origin, and `discovery_uri` MUST use the exact /.well-known/ocp-configuration path. A separately registered federation profile may authorize a different metadata origin only by defining issuer binding, authenticated metadata, redirect and SSRF rules, and credential-audience checks. Clients MUST NOT follow an untrusted discovery link into local/private networks or send credentials to a different origin without such a selected authenticated profile.

29.3 Core operations

Method and path	Purpose	Success	Required controls
POST /ocp/v0.4/client-declarations	register/reference a declaration	201 declaration	authenticated publisher; schema; expiry/status; idempotency.
POST /ocp/v0.4/context-requests	submit request	201 request	client/instance authentication; exact current declaration; idempotency; request limits.

Method and path	Purpose	Success	Required controls
GET /ocp/v0.4/context-requests/{id}	read immutable/current request record	200 request + ETag	caller/requester/steward authorization; non-disclosing 404.
GET /ocp/v0.4/context-requests/{id}/status	authoritative request status	200 typed status	caller/requester/steward authorization; non-disclosing 404; approved or partially approved responses include exact grant_receipt_ref and grant_status_ref.
GET /ocp/v0.4/grant-receipts/{id}	read non-secret receipt	200 receipt + ETag	authorization; receipt is not bearer authority.
GET /ocp/v0.4/grant-receipts/{id}/status	authoritative grant status	200 typed status	active authorization; no sensitive scope leakage.
POST /ocp/v0.4/context-packages	construct package	201 immutable package	enforceable credential/local capability; active exact receipt; trusted server evaluation time; idempotency; containment.
GET /ocp/v0.4/context-packages/{id}	retrieve package	200 package	current delivery recipient/instance authorization, expiry, no-store.
GET /ocp/v0.4/context-packages/{id}/status	authoritative package status	200 typed status	current delivery recipient/instance authorization; non-disclosing 404; expiry-aware 410.
GET /ocp/v0.4/context-packages/{id}/entries/{entry}	retrieve referenced entry	200 entry	repeat package, receipt, purpose, view, recipient, processor, and item-level checks.
POST /ocp/v0.4/update-proposals	submit proposal	201 proposal + ETag	bounded proposal right; rate limit; idempotency; exact target; stale target returns HTTP 412 / OCP_STALE_REVISION.
GET /ocp/v0.4/update-proposals/{id}/status	authoritative proposal status	200 typed status	submitter/reviewer/steward authorization; non-disclosing 404.

Method and path	Purpose	Success	Required controls
POST /ocp/v0.4/update-proposals/{id}/decisions	create review decision and optional accepted result	201 decision/result wrapper	exact path/proposal/If-Match binding; authenticated reviewer authority; exact reviewed result input for accepting outcomes; provider-authored identities, envelopes, decision time, lifecycle/review state, and derived commitment; atomicity; idempotency.
POST /ocp/v0.4/transfer-manifests	start binding transfer	201 manifest	Transfer Binding; fresh authorization; digest/entry/loss limits.
GET /ocp/v0.4/transfer-manifests/{id}	retrieve full transfer manifest	200 manifest + ETag	source/destination/steward authorization; includes entry results, losses, and exceptions.
GET /ocp/v0.4/transfer-manifests/{id}/status	authoritative transfer status	200 typed status	source/destination/steward authorization; non-disclosing 404.
GET /ocp/v0.4/events	event batch page	200 CloudEvents batch + Next-Cursor	filtered authorization; cursor and privacy rules.

The typed status endpoints are authoritative state-of-record projections; events only notify clients that a read may be warranted. A status response uses `checked_at` for the server observation time of its returned state. A `liveStatusRef` uses `last_checked_at` for its own referenced-status observation; the values are not interchangeable and need not be equal. `status_checked_at` is not an HTTP response member. An `approved` or `partially_approved` request status MUST contain both the exact immutable `grant_receipt_ref` and the live `grant_status_ref`; every other request state MUST omit both. The full transfer-manifest GET, rather than its status projection, is the authoritative representation for entry results, losses, and exceptions.

The context-package create input contains the exact `grant_receipt_ref`, current `grant_status_ref`, and an optional caller `requested_at`. `requested_at` is only a non-authoritative wall-clock hint for tracing or scheduling. It is not an authorization fact, trusted evaluation time, status-observation time, or package `issued_at`; the provider MUST use its own trusted clock, re-read current status, and author `issued_at`.

The review-decision create input contains caller-supplied `proposal_ref`, `reviewed_by`, `authority_basis`, `outcome`, optional `reason`, and—only for accepted or `accepted_with_changes`—a closed `resulting_item_input`. That input is the complete reviewed result semantics: extensions, subjects, space, category, optional untrusted-data treatment, statement, epistemic basis, optional validity/rights, provenance, sensitivity, and accepted scope. It deliberately excludes item ID/revision, creation audit fields, lifecycle, review state, and decision reference. The server MUST bind `proposal_ref.id` to the `{id}` path parameter, bind its revision and digest to the proposal revision selected by If-Match, and verify that `reviewed_by` is the authenticated actor with current review authority. For an accepting outcome, the server atomically authors a fresh/next result identity, both persisted envelopes and creation audit fields, result lifecycle `active`, result review status `accepted`, the exact decision reference, and the resulting-item commitment derived from the authored accepted-result projection. The 201 response returns a typed

wrapper containing the decision and, only for an accepting outcome, the resulting item. A nonaccepting outcome prohibits `resulting_item_input` and returns no resulting item. This makes `accepted_with_changes` reconstructible and prevents the caller from guessing provider-authored commitment fields.

The OpenAPI 3.1 file in the companion pack expresses this outline. A production binding **MUST** add its selected authorization and cryptographic profiles, exact operation responses, and asynchronous transfer behavior while honoring the advertised quantitative limits.

29.4 Idempotency

Every state-creating POST **MUST** accept an Idempotency-Key of 16–128 characters. The server binds the key to authenticated principal, endpoint, OCP version, and canonical request digest for a documented retention period. Repetition of the same tuple returns the original status/resource. Reuse with different canonical content returns `OCP_CONFLICT` and performs no side effect. HTTP 429 and every response carrying a retryable OCP problem **MUST** include a syntactically valid `Retry-After` delta-seconds or `IMF-fixdate` value. A client **MUST NOT** retry before that value and **MUST** treat a retryable problem without `Retry-After` as non-retryable rather than inventing a delay.

The binding constrains problem bodies to registered code/status/retryability tuples. Every advertised error response uses a status-specific response schema, so the problem body's status **MUST** equal the actual HTTP response status; a registered tuple for another status is invalid even though it is valid elsewhere. In particular, HTTP 429 carries `OCP_RATE_LIMITED` with `retryable: true`; HTTP 503 carries `OCP_TEMPORARILY_UNAVAILABLE` with `retryable: true`; an expired context-package read carries HTTP 410 / `OCP_PACKAGE_EXPIRED` / `retryable: false`; and event-cursor HTTP 410 carries exactly `OCP_EVENT_CURSOR_EXPIRED` or `OCP_EVENT_GAP`, with `retryable: false`.

29.5 ETags and preconditions

Stateful resources **SHOULD** return strong ETags. Review decisions, revocations, and other transitions against a current revision **MUST** require `If-Match` or an equivalent exact revision. A missing precondition returns `OCP_PRECONDITION_REQUIRED`; a mismatch returns `OCP_STALE_REVISION`.

29.6 Reference dereference

A `content_ref`, source URI, signature reference, schema URI, or policy URI is not automatically fetchable. The caller/provider **MUST** use a registered reference profile and separately authorize the dereference. Bearer URLs **MUST NOT** appear in packages. Every dereference is bounded, authenticated where required, digest-checked, and re-evaluated against current receipt, delivery recipient, processor set, purpose, view, lifecycle, and sensitivity.

29.7 Events

The optional event binding uses CloudEvents 1.0.2 JSON batch format for `GET /ocp/v0.4/events`: a successful finite page is `application/cloudevents-batch+json`, represented as a JSON array of CloudEvent objects. Each event carries `id`, `source`, `specversion`, `type`, `time`, `subject`, and `dataschema`, with `dataschema` exactly `https://opencontinuity.org/schemas/0.4/bindings/http/event-notification.schema.json`. Its reference-first data object contains the authoritative `object_ref`, `revision`, `required_correlation_id`, and `digest`; CloudEvents `subject` supplies routing/context and need not equal `data.object_ref`. The registered event type determines exactly one axis and its permitted data: a lifecycle event **MUST** carry one of the exact

from_state/to_state pairs registered for that type, and a notification-axis event carries notification_code exactly equal to its registered event type. The baseline HTTP binding advertises lifecycle events only for context requests, grant receipts, context packages, update proposals, and transfer manifests, because those resources have authoritative HTTP read/status contracts; the sole notification-axis event it advertises is ocp.package.integrity_failed, which is reconciled through the exact package status/read contract. Lifecycle or notification vocabulary for context items, artifacts, imports, revocation, erasure, authorization failure, unsupported extensions, or general security incidents remains registry material and requires a registered profile that supplies an authoritative resolver before appearing in this HTTP event stream. Baseline core registers no context-item dispute/reaffirmation event; such events require the profile defined in Section 18.4. Cross-axis members, an alternate data-schema URI, unadvertised or unregistered event types, and a transition registered to a different event type are prohibited. Events MUST NOT include raw context, evidence, artifacts, purpose text, tokens, credentials, or package bodies.

Events are notifications, not authorization, provenance, or state of record. Consumers deduplicate on event source/ID and process resource revisions; they do not infer causality from arrival order. A cursor is opaque and bound to the authenticated principal, authorization/filter scope, issuer, and OCP version. A cursorless request starts at the earliest retained event visible to that caller. The server MUST keep an issued cursor usable for at least limits.event_cursor_retention_seconds, unless the cursor's authorization context changes, and MUST return no more than limits.event_page_max_items events. Expiration returns HTTP 410 with OCP_EVENT_CURSOR_EXPIRED; a detected discontinuity, compaction loss, or scope-invalidating gap returns HTTP 410 with OCP_EVENT_GAP. After either response, the consumer MUST reconcile affected resources through the authoritative status endpoints before starting a new cursorless page. Rate limiting and temporary unavailability use Retry-After as defined in Section 29.4. A future streaming transport requires a separately versioned binding.

30. Offline package format

30.1 Archive structure

The Offline Archive Binding uses a ZIP-compatible archive rooted at manifest.json. The manifest validates against offline-package-manifest.schema.json and declares manifest/wire versions, package type/ID, creation time, archive format/classification, payload members, security members, exact member-set policy, and optional registered protection.

Each member has a safe relative path, media type, byte size, digest, and optional schema/object identity. Paths are archive identifiers, not destination filesystem instructions.

For Portable AI History, typed members may include:

```
manifest.json
objects/conversations/*.json
objects/messages/*.json
objects/artifacts/*.json
objects/lineage/*.json
adapters/adapter-manifest.json
adapters/mapping-registry.json
adapters/mapping-run.json
reports/loss-report.json
bytes/<content-addressed-member>
security/<profile-defined-members>
```

The archive manifest does not acquire undeclared profile fields. Profile identity, object relationships, mapping evidence, losses, and disclosure decisions are carried by the typed members and their schemas.

30.2 Path, archive, and profile safety

The physical decoded member set MUST equal the declared payload/security set. Importers reject unlisted or duplicate normalized members, absolute/parent/alternative-separator paths, symlink/hardlink/device escapes, header-directory disagreement, trailing data, relevant case/Unicode collisions, recursive archives outside policy, and resource-limit violations.

Before semantic processing, the importer verifies manifest shape, member sizes/digests, profile/schema support, graph/reference closure, Artifact custody claims, mapping-run/loss-report references, and unknown critical extensions. Artifact byte members remain quarantined until their exact representation digest verifies. Conversation markup, filenames, provider fields, and artifacts remain untrusted.

30.3 Authenticity and confidentiality

Non-public archives require registered signature and encryption profiles. `signature_path` and encryption metadata name distinct declared security members. The signature profile defines the RFC 8785 manifest projection, covered-member ordering, issuer/key trust, algorithms, and failure behavior; encryption binds the archive to a declared recipient key.

The 0.4 working draft still registers no executable offline protection profile. Therefore archive structure and digests prove neither issuer authenticity nor confidentiality, and XFR-004 remains unavailable. Successful decryption, signature validity, or provider reputation does not authorize runtime use or accepted-state import.

30.4 Import and runtime behavior

Every archive enters the raw/quarantine zone. Parsing is bounded and sandboxed with no network or ambient credentials. Validation, mapping, and destination transfer occur before any review; no step bypasses the accepted-context boundary. Unsupported critical semantics fail/partial the import with explicit exceptions.

Offline runtime use is prohibited by the baseline. A future Offline Authorization Profile may permit it under Section 16.6. Backup/export/transfer archives carry no external-action authority, tools, credentials, or permission to auto-fetch source URLs.

30.5 Determinism and round trip

Archive-member ordering, compression choices, security nonces, and clock/run identifiers are not semantic when the applicable profiles exclude them. Exact payload member bytes/digests, typed object semantics, graph edges, representation custody/integrity, opaque values, and loss reports are semantic. A re-export comparator explains every difference and never upgrades evidence, custody, rights, acceptance, or authority.

31. Adjacent-standard mappings

OCP targets a candidate interoperability gap: scoped acceptance, proposal/review, purpose-bound presentation, evidence/artifact disclosure separation, and an invariant that context conveys no external-action authority. PAM v1.0 already occupies the broad AI-memory interchange and round-trip/export space. Whether OCP should become a PAM profile or a separate mapped layer remains an implementation question; this draft therefore composes and profiles adjacent work instead of claiming a new category.

Standard / work	Reuse in OCP	OCP distinction and required mapping	Maturity / recommendation
MCP 2026-07-28	carry OCP resources/references through MCP resources or typed content; map narrow operations to tools	MCP context/tools/tasks do not define durable accepted context. Publish media type, capability, error, and auth mapping; never infer tool authority from context.	Current released, fast-moving dated spec. Optional binding, not core dependency.
A2A 1.0	carry package/reference as an A2A Part or Artifact; reuse task/message/streaming	A2A task state is not acceptance state. Define carriage, digest, dereference, auth, and error behavior.	Released 1.0 family. Optional binding.
PAM v1.0	reuse matching memory, lifecycle, provenance, signature, export, provider-mapping concepts where semantics align	Highest overlap. The checked-in informative map targets exact OCP 0.4 fields, cardinalities, lifecycles, proof boundaries, and losses in both directions. It remains experimental until machine-readable loss/sidecar fixtures and two independent adapters pass. OCP adds scoped acceptance, proposal review, purpose-bound presentation, evidence/artifact asymmetry, and action separation.	Published open format (2026-02-17), ecosystem new. Closest baseline; experimental mapping, not a conformance profile.
OAuth 2.0 / RFC 6749; BCP 240 / RFC 9700	operation-time access authorization and security practice	OCP request semantics do not replace tokens. Receipt is non-credential. Map resource, operation, delivery recipient/client instance, purpose, time, views, and status.	Mature standards; required foundation for a conforming remote binding.
OAuth RAR / RFC 9396	carry typed OCP authorization details and partial approval	OCP must define typed comparison; generic JSON subset is unsafe.	Standards-track RFC; recommended request carrier.
Protected Resource Metadata / RFC 9728; DPoP / RFC 9449	discover protected resource/auth servers; sender-constrain tokens	Exact resource/issuer matching and client-instance replay controls; not plaintext copy prevention.	Standards-track RFCs; use in remote security profile.
OpenID Connect Core	authenticate people and provide issuer/subject references, including pairwise subjects	ID Token is not access authorization; OIDC subject is not automatically an OCP context subject.	Mature final spec; identity profile input.
GNAP / RFC 9635; UMA	alternate grant negotiation/delegated authorization profiles where demanded	Do not recreate grant negotiation. Map request/decision/continuation and OCP receipt.	GNAP standards-track RFC; UMA ecosystem-specific. Optional profile only.

Standard / work	Reuse in OCP	OCP distinction and required mapping	Maturity / recommendation
Solid Protocol / WAC / ACP	store OCP resources and reuse Web access control where expressible	Purpose, downstream use, evidence view, and time may exceed ACL expressiveness. Publish capability matrix and partial-map behavior.	Solid/ACP community drafts, not W3C Recommendations. Optional storage profile.
W3C PROV-O	map source, entity, activity, agent, role, derivation, revision	OCP adds accepted/disputed/review state and disclosure/preservation semantics.	W3C Recommendation. Normative provenance mapping.
C2PA 2.4	reference manifests, actions, ingredients, validation results, repository receipts	Authenticity/validation is not truth, rights, accepted context, or perpetual preservation.	Published 2.4 specification. Artifact profile mapping.
VC Data Model 2.0 / Data Integrity; SD-JWT	reference external credentials, verification/status, selective disclosures	Separate issuer/holder/subject/verifier; verification from acceptance and action authority.	VC 2.0 W3C Recommendation; SD-JWT RFC 9901. Credential profile, not core.
DPV 2.0	URI vocabulary for purpose, processing, category, recipient, duration, safeguards, and legal-basis references	Selecting a term does not determine law or enforcement. Pin version and supply human labels.	Final Community Group Report, not W3C Standard. Optional privacy vocabulary profile.
JSON Schema 2020-12 / JCS RFC 8785	shape validation and deterministic JSON representation	Schema cannot prove cross-object authorization or truth. Define signed view and format assertion.	Stable specifications. Core dependencies.
HTTP Message Signatures RFC 9421 / Content-Digest	protect live representations in a transport profile	Does not replace offline manifest signature or confidentiality. Cover required components/digest.	Standards-track RFCs. Optional security binding.
CloudEvents 1.0.2 / OpenTelemetry	reference-first lifecycle events; privacy-safe operational measures	Events are not state/audit; telemetry is not provenance or consent. Prohibit context and high-cardinality identifiers.	Mature CNCF specs with evolving conventions. Optional profiles.
Data Transfer Project / Initiative	provider adapters, transfer jobs, retry, and operational patterns	Add an OCP data type; transfer success does not equal acceptance or erasure.	Active open-source infrastructure, not a normative standard. Implementation option.
NIST AI RMF / GAI Profile	threat/control/evidence traceability, adversarial testing, human oversight	Protocol features support but do not establish NIST conformance.	Voluntary NIST framework/profile. Assurance mapping.

Standard / work	Reuse in OCP	OCP distinction and required mapping	Maturity / recommendation
NIST AI Agent Standards Initiative	track emerging identity, authorization, accountability, and security vocabulary	No finished “NIST agent standard” yet. OCP can contribute concrete governed-context artifacts and tests.	Active 2026 initiative/draft work. Monitor and engage; no normative dependency.
SPIFFE 1.15.2 / WIMSE architecture -08 / NIST NCCoE agent identity concept	distinguish workload, client instance, executing agent, model, publisher, and human principal	Identity evidence does not itself authorize OCP disclosure or an external action. A binding must name the exact principal and assurance used.	SPIFFE is deployed ecosystem work; WIMSE is an active Internet-Draft; NIST material is a concept paper. Informative mapping only.
CRP core -00	A future binding could carry OCP package digests or status references alongside CRP runtime metadata.	CRP is a gateway safety, telemetry, session-relay, and context-injection layer; it does not define OCP acceptance, grant, proposal/review, disclosure, or no-action-authority semantics. CRP headers MUST NOT be interpreted as OCP grant receipts, accepted context, action authority, or issuer authentication.	Active individual Internet-Draft published 2026-05-24; no IETF endorsement or formal standing. Monitor; optional future binding, not a core dependency.
GDPR / EDPB portability guidance	design inputs for minimization, purpose, transparency, rights, portability, and accountability	Acceptance does not create a legal basis or settle third-party rights; inferred/derived data may fall outside portability rights.	Binding regulation plus guidance. Legal profile/review, never conformance claim.
EU Data Act	machine-readable access/transfer and switching design signals where in scope	Does not create a blanket right to AI memory; personal-data and trade-secret rules still apply.	Applicable regulation; recheck amendments/guidance at release.
EU AI Act, as amended through Regulation (EU) 2026/1744	traceability, human oversight, intended-purpose, logging, and role evidence	OCP conformance is not AI Act compliance; duties depend on system/use/role.	Regulation in force with phased application; recheck the consolidated text at release. Deployment legal review.

Each normative mapping profile MUST name exact source version, supported objects/fields, cardinality, lifecycle mapping, proof/authorization boundaries, loss behavior, negative fixtures, and round-trip acceptance criteria.

32. Versioning

32.1 Version dimensions

OCP separates:

1. `ocp_version` — core semantic wire version;

2. `schema_uri` — exact concrete representation schema;
3. profile identifier and version — bounded additional semantics;
4. event-type version — immutable event-data meaning; and
5. `revision` — immutable resource history/concurrency, not protocol version.

32.2 Compatibility

A major core version may make incompatible semantic or structural changes. A minor version may add optional semantics only when an older receiver can safely ignore them under extension/profile rules. No sender may mark required new semantics non-critical merely to claim backward compatibility.

Receivers **MUST** reject unsupported major versions and unsupported critical semantics. They **MAY** negotiate a mutually supported version through discovery. They **MUST NOT** silently downgrade authorization, integrity, confidentiality, lifecycle, or accepted-state requirements.

32.3 Schema stability

Published schema URIs are immutable. Corrections that change accepted instances receive a new schema URI/version. A schema catalog **SHOULD** publish digest, media type, status, supersession, and change notes. Implementations **SHOULD** pin schemas and registries used for conformance.

32.4 Deprecation

Deprecated fields/profiles remain documented for a transition period with import-only or read-only behavior, replacement mapping, sunset criteria, and loss notes. The deprecated term `canonical context` maps to accepted context only when the legacy source supplies equivalent scoped review semantics; otherwise it imports as an unverified candidate.

33. Extensions and profiles

33.1 Extension rules

Core extensions appear only in the URI-keyed `extensions` map, each with `critical` and `data`. An extension specification publishes identifier/owner/version, schema, actors, security/privacy impact, authorization comparison, lifecycle/package behavior, round-trip/downgrade rules, and tests. Profile-specific opaque fields use their declared container and remain inert.

Unknown critical extensions fail closed. Unknown non-critical extensions may be ignored semantically at runtime but are not silently dropped: they are preserved inertly, quarantined, or explicitly omitted with loss. Core/profile schemas reject undeclared top-level members. Extensions **MUST NOT** broaden authorization, accepted state, external-action authority, artifact custody, integrity, rights, or disclosure.

33.2 Complete profile requirements

A complete profile defines:

- stable URI/version and compatible core range;
- objects, fields, schemas, registries, and package projection;
- authority, authorization, lifecycle, privacy, and threat model;
- comparison, attenuation, redaction, loss, and downgrade rules;
- errors and critical/non-critical extension behavior;

- positive, negative, hostile, and round-trip fixtures;
- exact conformance classes/assertions and evidence boundaries;
- composition/collision behavior with other profiles; and
- deprecation/migration behavior.

Profiles cannot incompatibly redefine core fields. Composition either defines collisions/precedence or declares incompatibility in discovery/package metadata. A profile's validation does not imply another profile's conformance.

33.3 0.4 profile status

Conversation Profile 0.4, Artifact Profile 0.4, and Provider Adapter Framework 0.4 satisfy the editorial artifact requirements above in this working-draft source/package: normative schemas, registries, examples, fixtures, losses, composition, and class definitions are present. They remain working-draft—not interoperable-release—profiles because independent adapter/provider/client execution and independent security review have not occurred.

Erasure Status, Continuity Observation 0.1-draft, and Human Interaction Boundary 0.1-draft remain incomplete/draft profiles. The PAM v1.0 mapping remains informative/experimental pending machine sidecars and independent adapters. OAuth/RAR, Local Broker, MCP, A2A, credentials/selective disclosure, organizations, multi-party/guardian, delegation, live sync, and jurisdictional behavior remain future profiles/bindings.

Continuity observations carry `external_action_authority` and `policy_effect` equal to none. Human Interaction Boundary capture requires explicit user control; implementations MUST NOT synthesize covert emotion, disability, competence, capacity, diagnosis, or uncertainty records.

33.4 Explicit composition

Conversation references Artifact logical records for attachments/generated media; it does not embed bytes or inherit content permission. Artifact lineage may reference prompts/source records without granting message access. Provider adapters create mapped profile/source records plus mapping/loss evidence but never accepted context. The core request/receipt/package/review model governs disclosure and acceptance across the composition.

If a package combines these profiles, it declares every version and class, exact object/member closure, separate conversation/evidence/artifact metadata/content views, and a unified loss report whose codes resolve to the applicable registry. A collision fails closed.

33.5 Registries and evolution

Governed registries cover object/schema/profile identifiers, purposes/predicates, lifecycle/events/errors/extensions, media/package formats, loss codes/classes, artifact relations/custody/rights/digests, adapter mapping types/source-schema status, conformance assertions/classes, and security profiles. Entries are versioned, collision-checked, reviewable, and archived; experimental namespaces remain visibly unstable.

Wire/profile breaking changes require a new compatible version. Unknown future enum values are not accepted merely because JSON parsing succeeds. Schema drift follows the Adapter Framework; downgrade is explicit and loss-aware.

34. Error handling

34.1 Format

HTTP errors use RFC 9457 `application/problem+json` with `type`, `title`, actual HTTP status, stable OCP code, and `correlation_id`; `detail`, `instance`, field errors, and typed resource references are optional. The body status **MUST** equal the response status.

Error details **MUST NOT** reveal object existence, subject/relationship, hidden categories, raw context, policy internals, tokens, credentials, key material, stack traces, or parser paths. Authorization-equivalent callers receive equivalent existence behavior.

34.2 Core codes

Code	HTTP	Meaning
OCP_INVALID_REQUEST	400	request syntax or required semantics invalid
OCP_SCHEMA_VALIDATION_FAILED	400	object fails declared schema/format
OCP_AUTHENTICATION_REQUIRED	401	valid authentication absent
OCP_AUTHORIZATION_REQUIRED	401	enforceable OCP resource authorization absent
OCP_FORBIDDEN	403	authenticated caller lacks effective authorization
OCP_AUDIENCE_MISMATCH	403	recipient/client instance mismatch
OCP_PURPOSE_MISMATCH	403	purpose absent, changed, or incomparable
OCP_SCOPE_EXCEEDED	403	selector/view exceeds approval
OCP_PROCESSING_USE_EXCEEDED	403	requested use exceeds approval
OCP_ACTION_AUTHORITY_SEPARATE	403	OCP context cannot authorize the action
OCP_NOT_FOUND	404	absent or authorization-concealed resource
OCP_CONFLICT	409	revision/lifecycle/idempotency conflict
OCP_STALE_REVISION	412	exact target snapshot or conditional-write mismatch
OCP_GRANT_INACTIVE	409	receipt suspended, expired, revoked, or superseded
OCP_PACKAGE_EXPIRED	410	package validity ended
OCP_IMPORT_LIMIT_EXCEEDED	413	import resource limit exceeded

Code	HTTP	Meaning
OCP_UNSUPPORTED_MEDIA_TYPE	415	unsupported representation
OCP_SEMANTIC_VALIDATION_FAILED	422	cross-object invariant fails
OCP_UNSUPPORTED_PROFILE	422	required profile unavailable
OCP_UNSUPPORTED_CRITICAL_EXTENSION	422	critical semantics unavailable
OCP_INTEGRITY_FAILED	422	digest/signature check fails
OCP_UNSAFE_IMPORT	422	import violates safety rules
OCP_PRECONDITION_REQUIRED	428	revision precondition absent
OCP_RATE_LIMITED	429	bounded quota exceeded; follow Retry-After
OCP_INTERNAL_ERROR	500	unexpected safe failure
OCP_TEMPORARILY_UNAVAILABLE	503	safe completion unavailable

The complete registry contains 33 codes and retry metadata. Clients MUST branch on stable code, not localized title/detail.

35. Conformance classes

35.1 Claim structure

A conformance statement identifies implementation/component and version, actor role, OCP wire/profile/class versions, schema/assertion-registry versions, system/deployment boundary, tests executed, result report, exceptions, evaluator relationship, and evidence date/period. Unqualified “OCP 0.4 compliant,” “secure,” “private,” “preserved,” “lossless,” “verified,” “deletable,” or “interoperable” claims are invalid.

`conformance/classes.json` is authoritative for exact class-to-assertion membership. An implementation conforms to a class only when every required assertion has applicable passing evidence for that identified system under test; `partial_evidence`, `manual_required`, `unavailable`, `not_executed`, and another implementation's result are not passes.

35.2 Core and Portable AI History classes

Class	Exact required assertions
OCP-Core-0.4	CORE-001, AUTH-101
OCP-Conversation-Import-0.4	CONV-001–CONV-004, SAFE-201
OCP-Conversation-Package-0.4	CONV-005, CONV-006
OCP-Conversation-RoundTrip-0.4	RT-201, CONV-003, EXT-202
OCP-Artifact-Metadata-0.4	ART-001, ART-104
OCP-Artifact-Preservation-0.4	ART-101–ART-103
OCP-Artifact-Package-0.4	ART-105, PRIV-201
OCP-Artifact-RoundTrip-0.4	RT-202, ART-106
OCP-Provider-Adapter-0.4	ADAPT-201, ADAPT-202, SAFE-203
OCP-Provider-Adapter-Deterministic-0.4	ADAPT-203, ADAPT-204
OCP-Provider-Adapter-LossReport-0.4	ADAPT-205, LOSS-201

Classes remain separate because their evidence differs. Metadata support is not preservation; import is not package disclosure; mapping is not deterministic replay; and a loss report is not proof of lossless behavior. Implementations MAY claim multiple classes with one result set only when every class and assertion is named.

35.3 Existing role and binding classes

The registry also retains core data/provider/client, package producer/consumer, authorization binding, import safety, exporter/importer, privacy UX, external assurance, erasure, and binding-specific targets from 0.3. Remote Security (SEC-001–SEC-002) and non-public offline protection (XFR-004) remain unavailable until executable cryptographic profiles/vectors exist.

35.4 Assurance boundaries

Automated schema/semantic tests establish only their named boundary. Human comprehension, coercion safety, accessibility, legal authority, recipient purpose/retention, independent deletion,

production sandboxing, provider fidelity, and cross-implementation behavior require their applicable evidence. No missing evidence is converted into a pass.

35.5 Interoperability claims

An interoperability claim requires at least two independently developed implementations, exact versions/profiles/classes, exchanged fixtures or source packages, comparator/loss outputs, and published exceptions. Two components sharing substantial mapping or protocol code do not establish independence. Local generated examples prove only internal artifact consistency.

36. Normative test assertions

`conformance/assertions.json` is the complete registry of 134 normative assertions. `assertion-results.json` has exactly one status/evidence row per assertion. The following is a human map; it does not replace the machine registry.

36.1 Core, request, package, update, and transfer

The inherited 0.3 assertions continue to cover closed core records, immutable revisions, exact references, accepted decision linkage, provenance/evidence separation, RFC 3339 time, request/receipt containment, live status, delivery/processor binding, package minimization/integrity/aliases, proposal-only writes, review authority, import quarantine, archive safety, transfer state, revocation/erasure vocabulary, privacy, assurance, events/errors, source verification, JCS/I-JSON, HTTP, observation, and Human Interaction Boundary invariants.

36.2 Conversation assertions

- CONV - 001: arrays/branches preserve the source graph; array order never substitutes for edges.
- CONV - 002: resolved parent graph is acyclic.
- CONV - 003: flattening, omitted branches, or changed selected leaves is structural loss and not lossless.
- CONV - 004: missing parents require a partial graph, unresolved reference, and `CONV_PARENT_UNKNOWN`.
- CONV - 005: message, artifact-metadata, and artifact-content permission are separate.
- CONV - 006: source roles/tools/priorities/signatures/instructions remain untrusted and grant no control authority.

36.3 Artifact assertions

- ART - 101: independent preservation requires independently retrievable held bytes and verified digest.
- ART - 102: declared representation digest matches the exact held bytes.
- ART - 103: disappearance/changed-byte transitions do not silently change custody.
- ART - 104: logical artifact, representations, rights, accessibility, and lineage remain distinct.
- ART - 105: metadata and content use distinct package views.
- ART - 106: round trip never upgrades reference, derivative, rights, availability, or verification.

The inherited ART-001–ART-002 continue to prohibit URL-to-managed upgrade and join metadata/content authorization.

36.4 Adapter, loss, comparator, safety, privacy, and extension assertions

- ADAPT-201: adapters do not invent identity, order, messages, acceptance, authority, timestamps, ownership, rights, intent, or provider memory.
- ADAPT-202: unknown semantics are preserved, quarantined, or explicitly omitted—never guessed.
- ADAPT-203–ADAPT-204: fixed inputs yield an identical deterministic projection; run IDs/times are excluded.
- ADAPT-205: drifted/unrecognized schemas are labeled, never processed as a tested version.
- LOSS-201: every difference has class, severity, conformance effect, and round-trip effect.
- RT-201–RT-203: Conversation, Artifact, and Adapter comparisons cover the dimensions in Section 23.
- SAFE-201–SAFE-203: source content, artifact parsers, and hostile structure cannot cross control/sandbox/acceptance boundaries.
- PRIV-201: reports do not reveal withheld categories to unauthorized recipients.
- EXT-201: unknown critical extension fails closed.
- EXT-202: unknown non-critical extension is inertly preserved or loss-reported.

36.5 Required fixture and implementation evidence

Every automated result retains fixture/input digest, expected and observed result, implementation/profile versions, environment, evaluator, date, and exceptions. Negative authorization, import, package, parser, and action tests establish zero unauthorized bytes, zero action, zero unauthorized durable mutation, and no sensitive log/error leakage—not merely an error code.

Every complete provider adapter supplies the sixteen synthetic cases in Section 22.14. Conversation suites cover complete/partial/cyclic/missing-parent/flattened graphs and hostile role/content inputs. Artifact suites cover custody/integrity/disappearance and hostile bytes/metadata. Round-trip fixtures cover canonical semantic equality, explicit differences, and unknown-extension behavior.

36.6 Current local reference-material result

The synchronized local harness currently reports:

Test layer	Local result
JSON Schemas compiled/catalog verified	37 / 37
Normative positive examples	32 / 32
Normative negative examples rejected	45 / 45
Inherited cross-object semantic vectors	243 / 243
Registry/OpenAPI/evidence checks	29 / 29
Portable AI History cases	25 / 25

Test layer	Local result
— positive	9 / 9
— negative	6 / 6
— hostile	6 / 6
— round trip	4 / 4
Registered assertions/result rows	134 / 134
Assertions passed against an identified implementation	0
Assertions failed against an identified implementation	0
Partial local reference evidence	90
Manual required	4
Explicitly unavailable	3
Not executed	37

This evidence validates only generated local schemas, examples, registries, and synthetic reference logic. `system_under_test` is `null`; no normative assertion is executed `_pass`. It does not establish a provider adapter, independent provider/client, deployed sandbox, cryptographic protection, security review, legal compliance, accessible human flow, cross-language raw-input handling, or live human acceptance.

Untested requirements include live OAuth/capability/local binding, authenticated client-instance proof, operation-time provider authorization, parser escape/resource resistance in a deployed sandbox, recipient-key binding, cross-language JCS/raw JSON edge cases, two independent adapters/providers, an independent client, source-provider fidelity, and human privacy/accessibility evaluation. These remain explicit blockers rather than inferred passes.

37. Reference implementation guidance

37.1 Purpose and evidence boundary

A reference implementation exposes ambiguity and enables interoperability; it is not a privileged hosted service or canonical commercial implementation. It SHOULD be small, auditable, deterministic, storage-neutral, replaceable, and runnable offline for fixtures. KnowOnce has no protocol privilege.

37.2 Suggested components

```

schema catalog + closed validator
|
+--> immutable record/revision store
+--> authorization adapter + containment engine
+--> package builder/verifier
+--> proposal/review service
+--> raw quarantine + sandbox parser
+--> adapter registry/runner + mapping-run recorder
+--> conversation graph validator/comparator
+--> artifact byte store + digest/custody verifier
+--> transfer/export engine + loss renderer
+--> conformance harness + evidence report

separate action gateway (no OCP context authority)

```

The importer, package builder, and action gateway have typed separate boundaries. Raw/mapped content never enters authorization logic. Artifact promotion to managed custody is a trusted-code operation over held bytes. The containment engine does not call an LLM to decide authorization or semantic narrowing.

37.3 Repository/package layout

The source distribution includes spec, schemas, profiles, examples, fixtures, conformance, registries, bindings/http plus openapi, mappings, tools, review material, and rendered dist. Release archives omit development dependencies and transient QA. Every published member is digest-pinned by `release-manifest.json`.

37.4 Determinism and diagnostics

Validators return stable machine errors. Fixed mapping inputs produce a stable deterministic projection. Graph comparison derives topology from edges, not input order. Artifact verification hashes exact bytes. Package construction is reproducible at semantic-manifest level for fixed accepted snapshot, receipt, policy, time, alias seed, and security-profile inputs; nonces/encryption bytes may vary under the selected profile.

Every containment, adapter, and comparator decision explains passed, narrowed, failed, incomparable, preserved opaque, quarantined, or omitted status. Human loss reports derive from the same machine values.

37.5 Portable AI History reference demonstration

The target demonstration is:

```
Provider export family A -> Adapter A -> OCP Provider 1
-> Conversation + Artifact package -> OCP transfer
-> OCP Provider 2 -> re-export -> semantic comparator
```

Its reference corpus includes branching/regeneration, attachment, generated artifact/lineage, unavailable artifact, redaction, opaque provider field, unsupported semantic, and hostile instruction. The demonstration must establish graph/provenance preservation, role isolation, honest dependency/custody, byte digest verification, lineage, explicit loss, zero silent accepted-context creation, and zero context-to-action escalation.

The bundled synthetic fixture exercises those representations and local comparators. It is **not** the target demonstration because both sides are generated/tested by one source package and no actual adapter/provider/client implementation is identified. Release evidence requires two independent adapters, two independent providers, and one independent client.

37.6 Minimum cross-provider acceptance exercise

Provider A imports a synthetic source archive through quarantine, maps it with a named adapter, preserves a verified artifact, leaves another source-dependent, and produces a machine/human loss report. A reviewer accepts one derived context item without granting raw message or byte access. Provider A transfers the authorized profiles to Provider B. Provider B validates, imports without acceptance, re-exports, and its comparator explains every semantic difference. A mock action suite proves no source role/instruction/tool record can buy, send, publish, delete, deploy, fetch, or mutate accepted context.

37.7 Production readiness

Green local tests are not production readiness. Production claims additionally require independent threat/security/privacy/legal/accessibility review, dependency and supply-chain review, deployment

readback, parser isolation/resource tests, key lifecycle and recovery, abuse/incident operations, cross-provider execution, and live human acceptance for the declared scope.

38. Governance

38.1 Neutral stewardship

OCP SHOULD be governed by a neutral, multi-stakeholder project independent of a single commercial provider. The project needs representation from implementers, users/advocates, privacy and security experts, accessibility practitioners, archivists/provenance specialists, standards architects, and affected communities. KnowOnce MAY participate but MUST NOT hold privileged control over conformance, registry assignment, or the canonical specification.

38.2 Charter

Before a stable public draft, the project SHOULD publish a charter defining:

- mission and scope boundary;
- decision process and appeal;
- editor and maintainer selection/removal;
- public minutes, issue tracking, and change proposals;
- conflict-of-interest disclosure;
- security and privacy review gates;
- registry administration and collision policy;
- profile admission and deprecation;
- release cadence and long-term archive;
- conformance-mark rules; and
- funding transparency.

38.3 Intellectual property and licensing

Normative text SHOULD use a permissive documentation license compatible with standards reuse. Schemas, fixtures, and reference code SHOULD use a permissive software license. Contributors SHOULD accept a documented contribution and patent policy that provides implementers reasonable royalty-free assurances. The project MUST review copied vocabulary, schema, test vector, and source-license obligations before publication.

38.4 Change control

Material changes require an issue, rationale, security/privacy impact, compatibility analysis, schema diff, fixtures, implementation evidence, and review period. A breaking core change requires a new major wire version. Registry updates are versioned and archived. Emergency security updates MAY use an expedited path with later public review, but MUST NOT silently broaden access or disclosure.

38.5 Conformance marks

The governance body MAY license class-specific conformance marks only when test versions, evaluator requirements, exceptions, and expiration are public. Marks MUST identify the exact class/profile and

MUST NOT imply legal compliance, security certification, or independent interoperability unless those were separately established.

38.6 Security response

The project MUST publish a vulnerability-reporting process, embargo policy, supported versions, coordination expectations, and advisory format. Security advisories SHOULD identify affected schema/profile/implementation versions, exploit preconditions, privacy impact, mitigation, revocation/key implications, and fixture updates without exposing sensitive reporters.

38.7 Registry and namespace durability

OCP 0.4 uses the project-controlled <https://opencontinuity.org/> HTTPS namespace. Before any stable release, the project MUST document namespace ownership, neutral stewardship, immutable archival mirroring, redirects, deprecation, and recovery so identifiers do not depend on one organization's uptime. OCP SHOULD pursue IANA registrations only when the protocol is mature enough to meet registration policy; the current media types remain provisional and MUST NOT be presented as registered.

39. Design horizon: 2026–2031

This section records non-normative hypotheses that justify reversible hooks. It does not make 0.4 depend on future profiles.

39.1 Portable multimodal history

People are likely to accumulate long-lived conversations and generated work across multiple providers. The stable bet is not a universal turn, project, tool, or media ontology. It is graph-preserved conversations, logical artifacts plus representations, declared custody/integrity/source dependency, inert provider data, deterministic mappings, and explicit losses.

39.2 Provider evolution and disappearance

Provider export formats, IDs, URLs, and products will change or disappear. Adapter manifests/versioned mappings make drift visible; stable OCP record identity prevents provider ID from becoming universal identity; managed representation bytes can outlive source accounts; lifecycle transitions distinguish unavailable references from intact local copies. Long-term format migration and preservation policy remain implementation/profile concerns.

39.3 Multiple agents, tools, and delegation

Historical roles, tool calls, and agent labels will grow, but source history remains data. 0.4 provides inert tool/source hooks without defining universal tool semantics or delegation. Future Tool-History and Delegation Profiles must use separate operation authority and cannot reinterpret old messages as current permissions.

39.4 Local-first and live continuity

Immutable revisions, exact references, manifests, and loss reports support local-first stores and later synchronization. Live delta sync, forks/merges across providers, and offline multi-writer CRDTs remain future work because they need conflict/authority evidence beyond the 0.4 import/export proof point.

39.5 Authenticity, provenance, and selective disclosure

C2PA, PROV, credentials, selective disclosure, recipient encryption, and confidential execution may strengthen particular evidence or delivery modes. OCP composes with rather than duplicates them. Future profiles must handle realistic graphs, external bytes, revocation, linkability, and unknown data—not only single-object demonstrations.

39.6 High-risk human and organizational domains

Organizational authority, minors/guardianship, inheritance, full multi-party context, sensitive inference, autonomous delegation, and legal/jurisdictional determinations are deliberately outside the 0.4 dependency graph. Narrow references may exist, but no generic semantics are inferred. These domains require separate authority, coercion, privacy, and conformance evidence.

39.7 Durable center

The durable center remains reviewed scoped context; provenance-preserving evidence separation; purpose/recipient/time/use-bound disclosure; no silent accepted writes; no context-to-action authority; honest custody/deletion limits; and deterministic loss-aware movement. Storage, transport, crypto, agent runtime, ontology, and law remain bounded implementations/bindings/profiles.

40. Roadmap — 0.4 Portable AI History

40.1 Release thesis and status

The 0.4 proof point is: a person can import provider history with conversation graphs, multimodal artifacts, provider metadata, and derived context; move supported information to another conforming implementation; and receive deterministic evidence of what survived, changed, remains source-dependent, is unsupported, and entered accepted context only by review.

Human-facing promise: **Change the AI. Keep the history. Know what survived.**

The current artifact is an **Editor's Working Draft**. Editorial/package completion is not implementation completion. Every exit gate below is independent; none is satisfied merely by passing generated local fixtures.

40.2 Exit-gate ledger

#	Required 0.4 exit gate	Current evidence/status
1	Conversation Profile complete	Editorial schemas/text/fixtures present; independent implementation evidence missing.
2	Artifact Profile complete	Editorial schemas/text/fixtures present; independent implementation evidence missing.
3	Provider Adapter Framework complete	Editorial manifest/mapping/run/loss artifacts present; independent adapter evidence missing.

#	Required 0.4 exit gate	Current evidence/status
4	Full normative schemas published	37 schemas cataloged/validated and migrated to https://opencontinuity.org ; public deployment readback, immutable release publication, and archival mirroring are not completed.
5	Positive and negative fixtures validated	Local 32 positive and 45 negative pass; independent runners missing.
6	Hostile import corpus published	Synthetic local corpus present; public publication and deployed sandbox execution missing.
7	Two independently developed adapters tested	Not met.
8	Two independently developed OCP providers tested	Not met.
9	One independent OCP client tested	Not met.
10	Cross-provider round trip demonstrated	Local comparator only; not met independently.
11	Conversation graph preservation demonstrated	Local synthetic graph checks only; independent demonstration missing.
12	Artifact disappearance behavior demonstrated	Local transition matrix only; operational demonstration missing.
13	Managed-copy integrity demonstrated	Local embedded bytes/digest only; independent custody/storage evidence missing.
14	Provider schema drift demonstrated	Local synthetic fixtures only; actual adapters missing.
15	Unsupported semantics explicitly loss-reported, never guessed	Local fixtures cover; independent adapter evidence missing.
16	Imported content cannot escalate control authority	Local hostile assertions cover; deployed application boundary/security review missing.
17	Human report generated from machine result	Reference rendering data present; human/product/accessibility evaluation missing.
18	Independent security review for new profiles	Not met.

40.3 Working-draft implementation milestone

Build two independent provider adapters for real export families using only synthetic/public test inputs, two independent OCP providers, and one independent client. Publish exact versions, source-schema recognition, class claims, fixture results, exceptions, and comparator output. Agree on graph closure,

stable-versus-snapshot reference rules, custody/integrity claims, loss classification, and zero acceptance/action escalation.

40.4 Security and human milestone

Exercise parsers in deployed sandboxes against the hostile corpus and resource limits. Select remote/offline protection profiles, run raw-input/cross-language JSON/JCS/crypto vectors, and commission independent security/privacy, accessibility, and legal review. Remediate findings. Remote Security, offline protection, and human UX assertions remain unavailable/manual until then.

40.5 Publication candidate

Pin the <https://opencontinuity.org> namespace mapping, licensing/IPR, neutral governance, registry/archive and redirect policy, vulnerability response, migration guidance, complete decision/change logs, reproducible release archive, and assertion-by-assertion evidence. Public review must resolve material defects without weakening safety/semantic honesty.

40.6 Candidate 1.0

Candidate 1.0 requires stable governance, public reports, multiple independent interoperable implementations, no unresolved foundational security boundary, selected executable security profiles, durable registries, migration/upgrade path, and accepted residual risk. Vendor interest or schema completeness alone is not stability.

41. Resolved and remaining questions

41.1 Resolved for 0.4

Former question 17 — artifact preservation. OCP now uses logical artifacts, representations, explicit custody, independent-retrievability, byte digest verification, source dependency, and source-disappearance transitions. The claim is interoperable at the schema/fixture level; independent storage implementation evidence remains an exit gate. C2PA/PROV are references/mappings, not substitutes for custody.

Former question 19 — mapping losslessness. An adapter may call a mapping/run lossless only when all applicable supported semantics and disclosed values compare equivalent, unknown non-critical values survive, no critical unknown exists, every loss code permits lossless status, and a registered comparator/fixture suite for the identified implementation passes. Disputes are resolved by published source/mapping/comparator evidence and governance review—not adapter self-labeling.

Former question 23 — source disappearance/change. Reference-only becomes unavailable/lost; managed copies remain when exact local bytes verify; derivative-only retains the derivative but not original; synchronized copies become degraded while last verified bytes remain; a stable-looking provider ID/URL returning changed bytes creates a new source/representation revision and ART_SOURCE_ID_CHANGED. Custody is never silently upgraded/downgraded.

Turn object. No universal turn object in 0.4; graph messages are sufficient and avoid invented provider semantics.

41.2 Remaining implementation questions

1. Which two real provider export families provide the most independent, legally usable structure tests for first adapters?

2. Can independent implementations produce identical graph/loss results when provider branch semantics are implicit rather than declared?
3. Which source schema-drift heuristics are safely portable without treating heuristics as version recognition?
4. What long-term content-addressed store and verification cadence best substantiate managed custody across failures/migrations?
5. Which artifact format migrations preserve one logical artifact versus require a new derived artifact?
6. How should package-level redaction disclose a missing graph component when even its existence is sensitive?
7. Which provider-specific metadata deserves future portable vocabulary after at least two equivalent source mappings?
8. What comparator normalization set is sufficient across languages without hiding real timestamp, Unicode, number, or media differences?
9. Which C2PA/PROV mapping fields can be normative after prototype evidence, and which losses remain unavoidable?
10. What recipient-filtered loss-report UX is understandable, accessible, and non-leaking at very large history scale?
11. How can fixture suites verify parser sandbox/resource containment across operating systems and media libraries?
12. Which conformance results expire when providers change exports, adapters update, or custody verification becomes stale?
13. What exact evidence establishes that two adapters/providers are independently developed?
14. What is the smallest cross-provider demonstration users judge materially better than a structured archive plus prompt?

41.3 Deferred profile questions

Organizational authority, minors/guardianship, inheritance, full multi-party context, autonomous delegation, live delta sync, offline multi-writer merge, universal credentials/projects/personal context, and universal tool history remain future-profile work. They are not blockers for the bounded 0.4 Portable AI History proof point unless implementation evidence reveals a required narrow hook.

42. Complete validated examples

The examples below are exact generated copies of every positive fixture in this release. They are synthetic and prove representation/semantic-test behavior only. `npm run generate` deterministically regenerates the fixture set; `npm test` validates it.

42.1 Artifact Representation

```
{
  "representation_id": "urn:uuid:00000000-0000-4000-8000-000000000310",
  "relation": "original",
  "media_type": "image/png",
  "filename": "source-image.png",
  "byte_size": 46,
  "content_ref": "urn:ocp:content:synthetic-image",
  "digest": "sha-256:7e9aa08b283bb63ba5e087b5b2aadecd5408d59a3334cf9fb368e2dccf0c40f0",
  "digest_algorithm": "sha-256",
  "verified_at": "2026-08-13T11:51:00Z",
  "verification_method": "bytes_hashed_on_acquisition",
  "availability": "available",
  "independently_retrievable": true,
  "source_dependency": false
}
```

42.2 Artifact

```
{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",
  "type": "artifact",
  "id": "urn:uuid:00000000-0000-4000-8000-000000000300",
  "revision": 1,
  "created_at": "2026-08-13T12:00:00Z",
  "created_by": {
    "id": "https://provider-one.example",
    "type": "service",
    "display_name": "Synthetic OCP Provider 1"
  },
  "extensions": {},
  "profile_id": "https://opencontinuity.org/profiles/artifact",
  "profile_version": "0.4",
  "compatible_core": ">=0.4 <0.5",
  "subjects": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000101",
      "type": "person",
      "display_name": "Synthetic User"
    }
  ],
  "space_ref": {
    "id": "urn:uuid:00000000-0000-4000-8000-000000000102",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
    "digest": "sha-256:7d1d417fcb3aa98fadae41f5be2b6cb59ad4de12264fbd2f0a633d521c691255"
  },
  "semantic_type": "image",
  "origin": {
    "source_authority": {
      "id": "https://provider-a.example",
      "type": "service",
      "display_name": "Synthetic Provider A"
    }
  },
  "runtime_source": {
    "actor": {
      "id": "https://provider-one.example",
      "type": "service",
      "display_name": "Synthetic OCP Provider 1"
    },
    "software_name": "OCP Reference Importer",
    "software_version": "0.4-test",
    "execution_context": "offline_import",
  },
}
```

[illegible]

42.3 Client Assurance Evidence

```
{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
  "type": "source_record",
  "id": "urn:uuid:873980d8-1c50-4555-9764-f4958f2477cc",
  "revision": 1,
  "created_at": "2026-08-01T00:00:00Z",
  "created_by": {
    "id": "https://assessor.example",
    "type": "organization",
    "display_name": "Independent Example Assessor"
  },
  "extensions": {},
  "source type": "document",
}
```

```

"source_authority": {
  "id": "https://assessor.example",
  "type": "organization",
  "display_name": "Independent Example Assessor"
},
"runtime_source": {
  "actor": {
    "id": "https://assessor.example",
    "type": "organization",
    "display_name": "Independent Example Assessor"
  },
  "software_name": "Example assessment registry",
  "software_version": "1.0.0",
  "execution_context": "third_party_service",
  "build_digest": "sha-256:64df34cb4e19d3f1f0cd1ec7968930ff08e6271d4e6c438ba2d27a81367c6d76"
},
"source_object_id": "assessment-client-declaration-29668f25",
"acquisition_method": "api",
"observed_at": "2026-07-31T23:58:00Z",
"acquired_at": "2026-07-31T23:59:00Z",
"trust_treatment": "user_supplied",
"instruction_treatment": "untrusted_data",
"content": {
  "media_type": "application/json",
  "byte_size": 512,
  "digest": "sha-256:2dd8db0b19cb85c09528fe15cde1cf52f12f61bb8d1edaea48d328455154836e",
  "content_ref": "https://assessor.example/evidence/assessment-client-declaration-29668f25",
  "redacted": true
},
"availability": "available",
"sensitivity": "normal",
"lifecycle": {
  "status": "active"
}
}

```

42.4 Client Declaration Independently Assessed

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/client-declaration.schema.json",
  "type": "client_declaration",
  "id": "urn:uuid:29668f25-9d27-405b-a9ef-7e2f3899978d",
  "revision": 1,
  "created_at": "2026-08-12T12:00:00Z",
  "created_by": {
    "id": "https://travel-client.example",
    "type": "service",
    "display_name": "Example Travel Planner"
  },
  "extensions": {},
  "client": {
    "id": "https://travel-client.example",
    "type": "service",
    "display_name": "Example Travel Planner"
  },
  "privacy_policy_uri": "https://travel-client.example/privacy",
  "terms_uri": "https://travel-client.example/terms",
  "security_contact": "security@travel-client.example",
  "processing_recipients": [
    {
      "recipient": {
        "id": "https://model-provider.example",
        "type": "service",
        "display_name": "Example Model Provider"
      },
      "roles": [
        "model_provider"
      ],
      "purpose_codes": [
        "https://opencontinuity.org/vocab/purposes/plan-trip"
      ],
      "receives_plaintext": true
    }
  ],
  "retention_supported": [

```

```

    "no_store",
    "session"
  ],
  "processing_uses_supported": [
    "task_use"
  ],
  "offline_use_supported": [
    "prohibited"
  ],
  "proposal_rights_supported": [
    "none",
    "bounded"
  ],
  "delegation_supported": false,
  "deletion_support": {
    "request_supported": true,
    "status_receipt_supported": true,
    "request_uri": "https://travel-client.example/privacy/deletion-requests"
  },
  "conformance_statement_uri": "https://travel-client.example/ocp/conformance",
  "declaration_status": "independently_assessed",
  "valid_until": "2027-08-12T00:00:00Z",
  "assurance_evidence": {
    "evaluator": {
      "id": "https://assessor.example",
      "type": "organization",
      "display_name": "Independent Example Assessor"
    },
    "relationship": "independent_third_party",
    "method": "https://opencontinuity.org/profiles/assessment/example",
    "scope": "Client declaration controls and processing-recipient disclosure.",
    "assessed_at": "2026-08-01T00:00:00Z",
    "limitations": "Synthetic exact evidence only; the required provider policy evaluation is external
to this record and exercised by a local reference vector. No deployment assurance is claimed.",
    "evidence_ref": {
      "id": "urn:uuid:873980d8-1c50-4555-9764-f4958f2477cc",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
      "digest": "sha-256:ee4036824bedc8eb7cb4c7be96b48b5614b58c1b0db3d4e9e95bd7e2aba2eb05"
    }
  }
}

```

42.5 Client Declaration

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/client-declaration.schema.json",
  "type": "client_declaration",
  "id": "urn:uuid:763f4e35-b4d7-4931-9510-59d89db1a1b4",
  "revision": 1,
  "created_at": "2026-08-12T12:00:00Z",
  "created_by": {
    "id": "https://travel-client.example",
    "type": "service",
    "display_name": "Example Travel Planner"
  },
  "extensions": {},
  "client": {
    "id": "https://travel-client.example",
    "type": "service",
    "display_name": "Example Travel Planner"
  },
  "privacy_policy_uri": "https://travel-client.example/privacy",
  "terms_uri": "https://travel-client.example/terms",
  "security_contact": "security@travel-client.example",
  "processing_recipients": [
    {
      "recipient": {
        "id": "https://model-provider.example",
        "type": "service",
        "display_name": "Example Model Provider"
      },
      "roles": [
        "model_provider"
      ]
    }
  ],
}

```

```

    "purpose_codes": [
      "https://opencontinuity.org/vocab/purposes/plan-trip"
    ],
    "receives_plaintext": true
  }
],
"retention_supported": [
  "no_store",
  "session"
],
"processing_uses_supported": [
  "task_use"
],
"offline_use_supported": [
  "prohibited"
],
"proposal_rights_supported": [
  "none",
  "bounded"
],
"delegation_supported": false,
"deletion_support": {
  "request_supported": true,
  "status_receipt_supported": true,
  "request_uri": "https://travel-client.example/privacy/deletion-requests"
},
"conformance_statement_uri": "https://travel-client.example/ocp/conformance",
"declaration_status": "self_declared",
"valid_until": "2027-08-12T00:00:00Z"
}

```

42.6 Context Item Accepted

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/context-item.schema.json",
  "type": "context_item",
  "id": "urn:uuid:4alc68f5-8242-45b8-8e54-1fa8acb993c7",
  "revision": 1,
  "created_at": "2026-08-12T15:04:00Z",
  "created_by": {
    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "extensions": {},
  "subjects": [
    {
      "id": "urn:uuid:7a4ea2e0-94c2-4e8e-b83b-785f0ad78ca9",
      "type": "person",
      "display_name": "Avery Chen"
    }
  ],
  "space_ref": {
    "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
    "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
  },
  "category": "statement",
  "statement": {
    "predicate": "https://opencontinuity.org/vocab/predicates/prefers-hotel-style",
    "object": {
      "type": "string",
      "value": "boutique",
      "language": "en"
    }
  },
  "epistemic_basis": "subject_confirmed",
  "validity": {
    "observed_at": "2026-03-10T18:00:00Z",
    "last_confirmed_at": "2026-08-12T15:04:00Z",
    "review_after": "2027-02-12T00:00:00Z"
  },
  "provenance": [
    {

```

```

    "source_ref": {
      "id": "urn:uuid:8659cadf-69c2-403e-ae91-dd214f1c265b",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
      "digest": "sha-256:6fa9fd088d63a5599714237d0cb3750fc6e046edb83095af25b2cecd9caa68d"
    },
    "method": "inference",
    "observed_at": "2026-03-10T18:00:00Z",
    "acquired_at": "2026-08-12T13:01:00Z",
    "producer": {
      "id": "https://context-provider.example",
      "type": "service",
      "display_name": "Example Context Provider"
    },
    "runtime_source": {
      "actor": {
        "id": "https://context-provider.example",
        "type": "service",
        "display_name": "Example Context Provider"
      },
      "software_name": "Example OCP importer",
      "software_version": "1.2.0",
      "execution_context": "provider",
      "build_digest": "sha-256:3d17c04f13580d204701ea4a0e95f8726f2a5ec50880a7c15078971661e9cf0a"
    },
    "adapter_version": "example-assistant-adapter/1.2.0",
    "redacted": true
  }
},
"sensitivity": "normal",
"lifecycle": {
  "status": "active"
},
"review": {
  "status": "accepted",
  "decision_ref": {
    "id": "urn:uuid:0bd0df97-5fb1-4db5-9937-e7136fbefbf6",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/review-decision.schema.json",
    "digest": "sha-256:46a4b05d8c5a9095c3da75532ea61573f21c7fa826856d9e0c718d85c8b2e157"
  },
  "accepted_scope": {
    "space_refs": [
      {
        "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
        "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
      }
    ],
    "purpose_codes": [
      "https://opencontinuity.org/vocab/purposes/plan-trip"
    ]
  }
}
}
}

```

42.7 Context Item Candidate

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/context-item.schema.json",
  "type": "context_item",
  "id": "urn:uuid:2bb6cf0e-3650-435b-af8f-25b46878b72c",
  "revision": 1,
  "created_at": "2026-08-12T13:03:00Z",
  "created_by": {
    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "extensions": {},
  "subjects": [
    {
      "id": "urn:uuid:7a4ea2e0-94c2-4e8e-b83b-785f0ad78ca9",
      "type": "person",

```

```

    "display_name": "Avery Chen"
  }
},
"space_ref": {
  "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
  "revision": 1,
  "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
  "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
},
"category": "statement",
"statement": {
  "predicate": "https://opencontinuity.org/vocab/predicates/prefers-hotel-style",
  "object": {
    "type": "string",
    "value": "boutique",
    "language": "en"
  }
},
"epistemic_basis": "provider_inferred",
"review": {
  "status": "candidate"
},
"validity": {
  "observed_at": "2026-03-10T18:00:00Z",
  "review_after": "2027-02-12T00:00:00Z"
},
"provenance": [
  {
    "source_ref": {
      "id": "urn:uuid:8659cadf-69c2-403e-ae91-dd214f1c265b",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
      "digest": "sha-256:6fa9fd088d63a5599714237d0cb3750fc6e046edb83095af25b2cecd9ca68d"
    },
    "method": "inference",
    "observed_at": "2026-03-10T18:00:00Z",
    "acquired_at": "2026-08-12T13:01:00Z",
    "producer": {
      "id": "https://context-provider.example",
      "type": "service",
      "display_name": "Example Context Provider"
    },
    "runtime_source": {
      "actor": {
        "id": "https://context-provider.example",
        "type": "service",
        "display_name": "Example Context Provider"
      },
      "software_name": "Example OCP importer",
      "software_version": "1.2.0",
      "execution_context": "provider",
      "build_digest": "sha-256:3d17c04f13580d204701ea4a0e95f8726f2a5ec50880a7c15078971661e9cf0a"
    },
    "adapter_version": "example-assistant-adapter/1.2.0",
    "redacted": true
  }
],
"sensitivity": "normal",
"lifecycle": {
  "status": "active"
}
}

```

42.8 Context Package Metadata Layers

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/context-package.schema.json",
  "type": "context_package",
  "id": "urn:uuid:c5766808-43cd-4761-8180-c64a6f2fcfff",
  "revision": 1,
  "created_at": "2026-08-12T15:05:00Z",
  "created_by": {
    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  }
}

```

```

},
"extensions": {},
"correlation_id": "urn:uuid:bd6ee087-a8b4-450d-9f84-aa89b8b978c0",
"grant_receipt_ref": {
  "id": "urn:uuid:0fc71ecf-alc7-493a-bb4e-6c8d09fb9e5c",
  "revision": 1,
  "schema_uri": "https://opencontinuity.org/schemas/0.4/grant-receipt.schema.json",
  "digest": "sha-256:22b657371de627235ff9a141724e60ad389ec669319c662e8b9035ca5be279bc"
},
"grant_status_ref": {
  "uri": "https://context-provider.example/ocp/v0.4/grant-receipts/0fc71ecf-alc7-493a-bb4e-6c8d09fb9e5c/status",
  "last_checked_at": "2026-08-12T15:02:00Z"
},
"purpose": {
  "code": "https://opencontinuity.org/vocab/purposes/plan-trip",
  "description": "Plan a ten-day trip while respecting the traveler's accepted lodging preferences."
},
"delivery_recipient": {
  "recipient": {
    "id": "https://travel-client.example",
    "type": "service",
    "display_name": "Example Travel Planner"
  },
  "client_instance": {
    "id": "urn:uuid:43af637a-fcfe-42d5-938e-0fcd55eb075f",
    "type": "device",
    "display_name": "Example Travel Planner instance"
  },
  "client_declaration_ref": {
    "id": "urn:uuid:763f4e35-b4d7-4931-9510-59d89db1a1b4",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/client-declaration.schema.json",
    "digest": "sha-256:59079539153538f2d01f8c2f45ed0bcda2ff0d9caf81ea54605428c54ada0501"
  }
},
"processing_recipients": [
  {
    "recipient": {
      "id": "https://model-provider.example",
      "type": "service",
      "display_name": "Example Model Provider"
    },
    "roles": [
      "model_provider"
    ],
    "purpose_codes": [
      "https://opencontinuity.org/vocab/purposes/plan-trip"
    ],
    "receives_plaintext": true
  }
],
"issued_at": "2026-08-12T15:05:00Z",
"expires_at": "2026-08-12T20:00:00Z",
"subject_aliases": [
  {
    "package_subject_ref": "urn:uuid:54648acb-1444-4ee8-8fe4-4308add95afb",
    "delivery_alias": "urn:uuid:6d812d74-140b-4fb4-92ed-e29c89677854"
  }
],
"reference_aliases": [
  {
    "package_ref": "urn:uuid:d52008f1-37af-4bcf-838d-47dcc9a64294",
    "kind": "space"
  },
  {
    "package_ref": "urn:uuid:7f89e6a5-266f-49d9-a1c6-b769db087adf",
    "kind": "decision"
  },
  {
    "package_ref": "urn:uuid:9942c4fb-b849-48bd-a6a2-78dadf6dffcbb",
    "kind": "source"
  },
  {
    "package_ref": "urn:uuid:11572032-167b-4de5-b6d6-4132113444cd",
    "kind": "artifact"
  }
],

```

```

    {
      "package_ref": "urn:uuid:8c77a54d-bdca-4a3a-80bb-00ea81407d5b",
      "kind": "artifact"
    }
  ],
  "entries": [
    {
      "entry_id": "urn:uuid:c65a7636-d891-465d-b76d-0f7950cd9249",
      "layer": "context",
      "disclosure": "accepted",
      "media_type": "application/ocp+json",
      "content_schema_uri": "https://opencontinuity.org/schemas/0.4/context-package.schema.json#/$defs/packageContextContent",
      "digest": "sha-256:173554a7bf0c3805650757dfa443646607de0db4c01af5c715a9136f81b76173",
      "package_item_ref": "urn:uuid:70587099-10a2-49ac-8367-03032280071e",
      "content": {
        "subject_refs": [
          "urn:uuid:54648acb-1444-4ee8-8fe4-4308add95afb"
        ],
        "space_ref": "urn:uuid:d52008f1-37af-4bcf-838d-47dcc9a64294",
        "category": "statement",
        "statement": {
          "predicate": "https://opencontinuity.org/vocab/predicates/prefers-hotel-style",
          "object": {
            "type": "string",
            "value": "boutique",
            "language": "en"
          }
        },
        "epistemic_basis": "subject_confirmed",
        "review_status": "accepted",
        "decision_ref": "urn:uuid:7f89e6a5-266f-49d9-a1c6-b769db087adf",
        "provenance_refs": [
          "urn:uuid:9942c4fb-b849-48bd-a6a2-78dadf6dffcb"
        ],
        "sensitivity": "normal",
        "instruction_treatment": "untrusted_data"
      },
      "instruction_treatment": "untrusted_data"
    },
    {
      "entry_id": "urn:uuid:73c4c908-6401-4d0a-a90e-68844ee3ce49",
      "layer": "evidence",
      "disclosure": "metadata",
      "media_type": "application/ocp+json",
      "content_schema_uri": "https://opencontinuity.org/schemas/0.4/context-package.schema.json#/$defs/packageEvidenceMetadata",
      "digest": "sha-256:eda69da8970a828a9a802354b4ff50a7ecf7fdc0cf79cc531e6e5dba7b9c1d25",
      "content": {
        "package_source_ref": "urn:uuid:9942c4fb-b849-48bd-a6a2-78dadf6dffcb",
        "package_space_ref": "urn:uuid:d52008f1-37af-4bcf-838d-47dcc9a64294",
        "supports_package_item_refs": [
          "urn:uuid:70587099-10a2-49ac-8367-03032280071e"
        ],
        "source_type": "conversation",
        "trust_treatment": "user_supplied",
        "availability": "available",
        "observed_at": "2026-03-10T18:00:00Z",
        "acquired_at": "2026-08-12T13:01:00Z",
        "sensitivity": "sensitive",
        "instruction_treatment": "untrusted_data"
      },
      "instruction_treatment": "untrusted_data"
    },
    {
      "entry_id": "urn:uuid:a75424ff-7835-4d42-b608-22a44dcd5190",
      "layer": "artifact",
      "disclosure": "metadata",
      "media_type": "application/ocp+json",
      "content_schema_uri": "https://opencontinuity.org/schemas/0.4/context-package.schema.json#/$defs/packageArtifactMetadata",
      "digest": "sha-256:da6e5fc092e1a43845dfb375984f0cd6cba4e7300eb38bcc6e8586a9e347c978",
      "content": {
        "package_artifact_ref": "urn:uuid:8c77a54d-bdca-4a3a-80bb-00ea81407d5b",
        "package_space_ref": "urn:uuid:d52008f1-37af-4bcf-838d-47dcc9a64294",
        "supports_package_item_refs": [
          "urn:uuid:70587099-10a2-49ac-8367-03032280071e"
        ]
      }
    }
  ]
}

```

```

    ],
    "artifact_kind": "document",
    "media_type": "application/pdf",
    "content_disclosed": false,
    "sensitivity": "normal",
    "instruction_treatment": "untrusted_data"
  },
  "instruction_treatment": "untrusted_data"
},
"renderings": [
  {
    "media_type": "text/plain",
    "language": "en",
    "derived_from_entries": [
      "urn:uuid:c65a7636-d891-465d-b76d-0f7950cd9249"
    ],
    "content_ref": "urn:uuid:11572032-167b-4de5-b6d6-4132113444cd",
    "digest": "sha-256:8d077f2c6db228c41aee3cfd52fc47f720c37e9f770fbd38a3a46a0bf4b622ee",
    "instruction_treatment": "untrusted_data"
  }
],
"omissions": [],
"warnings": [],
"retention": {
  "mode": "session",
  "legal_retention_may_apply": false
},
"processing_uses": {
  "task_use": "permitted",
  "model_training": "prohibited",
  "product_improvement": "prohibited",
  "advertising": "prohibited"
},
"offline_use": "prohibited",
"proposal_rights": {
  "mode": "bounded",
  "operations": [
    "create",
    "revise"
  ]
},
"selectors": [
  {
    "layer": "context",
    "item_categories": [
      "statement"
    ],
    "predicates": [
      "https://opencontinuity.org/vocab/predicates/prefers-hotel-style"
    ],
    "sensitivity_max": "normal",
    "space_refs": [
      "urn:uuid:d52008f1-37af-4bcf-838d-47dcc9a64294"
    ]
  }
],
"sensitivity_max": "normal",
"source_attachment": "referenced_authorized"
},
"delegation": {
  "mode": "none"
},
"external_action_authority": "none",
"integrity": {
  "digest": "sha-256:deff67d06bb198bd741152de97175ce1e0b19a94b144a41be4c36e42ead819c9",
  "canonicalization": "RFC8785"
}
}

```

42.9 Context Package

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/context-package.schema.json",
  "type": "context_package",
  "id": "urn:uuid:82b2f316-2adc-4181-b676-c8431fceb32c",

```

```

"revision": 1,
"created_at": "2026-08-12T15:05:00Z",
"created_by": {
  "id": "https://context-provider.example",
  "type": "service",
  "display_name": "Example Context Provider"
},
"extensions": {},
"correlation_id": "urn:uuid:bd6ee087-a8b4-450d-9f84-aa89b8b978c0",
"grant_receipt_ref": {
  "id": "urn:uuid:d7681bd6-5807-4e3c-bfd8-33c4f0799bfd",
  "revision": 1,
  "schema_uri": "https://opencontinuity.org/schemas/0.4/grant-receipt.schema.json",
  "digest": "sha-256:4657179eadb7ccb43cbef2c4fa1f990cac8443d995ae775fa7fa00de7f66ad47"
},
"grant_status_ref": {
  "uri": "https://context-provider.example/ocp/v0.4/grant-receipts/d7681bd6-5807-4e3c-bfd8-33c4f0799bfd/status",
  "last_checked_at": "2026-08-12T15:05:00Z"
},
"purpose": {
  "code": "https://opencontinuity.org/vocab/purposes/plan-trip",
  "description": "Plan a ten-day trip while respecting the traveler's accepted lodging preferences."
},
"delivery_recipient": {
  "recipient": {
    "id": "https://travel-client.example",
    "type": "service",
    "display_name": "Example Travel Planner"
  },
  "client_instance": {
    "id": "urn:uuid:43af637a-fcfe-42d5-938e-0fcd55eb075f",
    "type": "device",
    "display_name": "Example Travel Planner instance"
  },
  "client_declaration_ref": {
    "id": "urn:uuid:763f4e35-b4d7-4931-9510-59d89db1a1b4",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/client-declaration.schema.json",
    "digest": "sha-256:59079539153538f2d01f8c2f45ed0bcda2ff0d9caf81ea54605428c54ada0501"
  }
},
"processing_recipients": [
  {
    "recipient": {
      "id": "https://model-provider.example",
      "type": "service",
      "display_name": "Example Model Provider"
    },
    "roles": [
      "model_provider"
    ],
    "purpose_codes": [
      "https://opencontinuity.org/vocab/purposes/plan-trip"
    ],
    "receives_plaintext": true
  }
],
"issued_at": "2026-08-12T15:05:00Z",
"expires_at": "2026-08-12T20:00:00Z",
"subject_aliases": [
  {
    "package_subject_ref": "urn:uuid:42a6ad3c-b34c-4d35-9b43-8cf72e989735",
    "delivery_alias": "urn:uuid:6d812d74-140b-4fb4-92ed-e29c89677854"
  }
],
"reference_aliases": [
  {
    "package_ref": "urn:uuid:fd1d94b4-d14a-4920-983a-3b501238a6c5",
    "kind": "space"
  },
  {
    "package_ref": "urn:uuid:ad2be2ce-827b-45a6-8c21-1f788cb5bef1",
    "kind": "decision"
  },
  {
    "package_ref": "urn:uuid:cdc815e6-7e51-47d1-95bf-01be8e18a320",

```

```

    "kind": "source"
  },
  {
    "package_ref": "urn:uuid:f1d80f51-dfa1-4c7c-af44-b6296572ff07",
    "kind": "artifact"
  }
],
"entries": [
  {
    "entry_id": "urn:uuid:4f7ba4ce-aeel-49a3-a705-14f29d98a501",
    "layer": "context",
    "disclosure": "accepted",
    "media_type": "application/ocp+json",
    "content_schema_uri": "https://opencontinuity.org/schemas/0.4/context-package.schema.json#/$defs/packageContextContent",
    "digest": "sha-256:af9c0f49046b829c5db808ca9020ceab46d452bc80d825f774fd64a60f43f3cb",
    "package_item_ref": "urn:uuid:9b0d728f-8b6e-4aa5-ae3b-2a3b46e3d0f2",
    "content": {
      "subject_refs": [
        "urn:uuid:42a6ad3c-b34c-4d35-9b43-8cf72e989735"
      ],
      "space_ref": "urn:uuid:fd1d94b4-d14a-4920-983a-3b501238a6c5",
      "category": "statement",
      "statement": {
        "predicate": "https://opencontinuity.org/vocab/predicates/prefers-hotel-style",
        "object": {
          "type": "string",
          "value": "boutique",
          "language": "en"
        }
      },
      "epistemic_basis": "subject_confirmed",
      "review_status": "accepted",
      "decision_ref": "urn:uuid:ad2be2ce-827b-45a6-8c21-1f788cb5bef1",
      "provenance_refs": [
        "urn:uuid:cdc815e6-7e51-47d1-95bf-01be8e18a320"
      ],
      "sensitivity": "normal",
      "instruction_treatment": "untrusted_data"
    },
    "instruction_treatment": "untrusted_data"
  }
],
"renderings": [
  {
    "media_type": "text/plain",
    "language": "en",
    "derived_from_entries": [
      "urn:uuid:4f7ba4ce-aeel-49a3-a705-14f29d98a501"
    ],
    "content_ref": "urn:uuid:f1d80f51-dfa1-4c7c-af44-b6296572ff07",
    "digest": "sha-256:8d077f2c6db228c41aee3cfd52fc47f720c37e9f770fbd38a3a46a0bf4b622ee",
    "instruction_treatment": "untrusted_data"
  }
],
"omissions": [
  {
    "selector": {
      "layer": "evidence",
      "space_refs": [
        "urn:uuid:fd1d94b4-d14a-4920-983a-3b501238a6c5"
      ]
    },
    "reason": "not_granted"
  }
],
"warnings": [],
"retention": {
  "mode": "session",
  "legal_retention_may_apply": false
},
"processing_uses": {
  "task_use": "permitted",
  "model_training": "prohibited",
  "product_improvement": "prohibited",
  "advertising": "prohibited"
},

```

```

"offline_use": "prohibited",
"proposal_rights": {
  "mode": "bounded",
  "operations": [
    "create",
    "revise"
  ],
  "selectors": [
    {
      "layer": "context",
      "item_categories": [
        "statement"
      ],
      "predicates": [
        "https://opencontinuity.org/vocab/predicates/prefers-hotel-style"
      ],
      "sensitivity_max": "normal",
      "space_refs": [
        "urn:uuid:fd1d94b4-d14a-4920-983a-3b501238a6c5"
      ]
    }
  ],
  "sensitivity_max": "normal",
  "source_attachment": "referenced_authorized"
},
"delegation": {
  "mode": "none"
},
"external_action_authority": "none",
"integrity": {
  "digest": "sha-256:09cbcb90f08b8e29d5e938b9a6234e3061eecb3d8854fd98a2949c91934a6f2d",
  "canonicalization": "RFC8785"
}
}

```

42.10 Context Request Metadata Layers

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/context-request.schema.json",
  "type": "context_request",
  "id": "urn:uuid:c58c54e6-1c7a-48ce-bd73-1303db3cb7b4",
  "revision": 1,
  "created_at": "2026-08-12T15:00:00Z",
  "created_by": {
    "id": "https://travel-client.example",
    "type": "service",
    "display_name": "Example Travel Planner"
  },
  "extensions": {},
  "correlation_id": "urn:uuid:bd6ee087-a8b4-450d-9f84-aa89b8b978c0",
  "delivery_recipient": {
    "recipient": {
      "id": "https://travel-client.example",
      "type": "service",
      "display_name": "Example Travel Planner"
    },
    "client_instance": {
      "id": "urn:uuid:43af637a-fcfe-42d5-938e-0fcd55eb075f",
      "type": "device",
      "display_name": "Example Travel Planner instance"
    },
    "client_declaration_ref": {
      "id": "urn:uuid:763f4e35-b4d7-4931-9510-59d89db1a1b4",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/client-declaration.schema.json",
      "digest": "sha-256:59079539153538f2d01f8c2f45ed0bcda2ff0d9caf81ea54605428c54ada0501"
    }
  },
  "processing_recipients": [
    {
      "recipient": {
        "id": "https://model-provider.example",
        "type": "service",
        "display_name": "Example Model Provider"
      }
    }
  ]
}

```

```

    "roles": [
      "model_provider"
    ],
    "purpose_codes": [
      "https://opencontinuity.org/vocab/purposes/plan-trip"
    ],
    "receives_plaintext": true
  }
],
"purpose": {
  "code": "https://opencontinuity.org/vocab/purposes/plan-trip",
  "description": "Plan a ten-day trip while respecting the traveler's accepted lodging preferences."
},
"selectors": [
  {
    "layer": "context",
    "space_refs": [
      {
        "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
        "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
      }
    ]
  },
  {
    "item_categories": [
      "statement"
    ],
    "predicates": [
      "https://opencontinuity.org/vocab/predicates/prefers-hotel-style"
    ],
    "sensitivity_max": "normal"
  },
  {
    "layer": "evidence",
    "space_refs": [
      {
        "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
        "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
      }
    ],
    "sensitivity_max": "sensitive"
  },
  {
    "layer": "artifact",
    "space_refs": [
      {
        "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
        "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
      }
    ],
    "sensitivity_max": "normal"
  }
],
"views": {
  "context": "accepted",
  "evidence": "metadata",
  "artifacts": "metadata"
},
"requested_not_after": "2026-08-12T20:00:00Z",
"retention": {
  "mode": "session",
  "legal_retention_may_apply": false
},
"processing_uses": {
  "task_use": "permitted",
  "model_training": "prohibited",
  "product_improvement": "prohibited",
  "advertising": "prohibited"
},
"offline_use": "prohibited",
"proposal_rights": {
  "mode": "bounded",
  "operations": [

```

```

    "create",
    "revise"
  ],
  "selectors": [
    {
      "layer": "context",
      "item_categories": [
        "statement"
      ],
      "predicates": [
        "https://opencontinuity.org/vocab/predicates/prefers-hotel-style"
      ],
      "sensitivity_max": "normal"
    }
  ],
  "sensitivity_max": "normal",
  "source_attachment": "referenced_authorized"
},
"delegation": {
  "mode": "none"
},
"status": "submitted"
}

```

42.11 Context Request

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/context-request.schema.json",
  "type": "context_request",
  "id": "urn:uuid:39ef7160-7243-41e9-93cb-3a6daec9f566",
  "revision": 1,
  "created_at": "2026-08-12T15:00:00Z",
  "created_by": {
    "id": "https://travel-client.example",
    "type": "service",
    "display_name": "Example Travel Planner"
  },
  "extensions": {},
  "correlation_id": "urn:uuid:bd6ee087-a8b4-450d-9f84-aa89b8b978c0",
  "delivery_recipient": {
    "recipient": {
      "id": "https://travel-client.example",
      "type": "service",
      "display_name": "Example Travel Planner"
    },
    "client_instance": {
      "id": "urn:uuid:43af637a-fcfe-42d5-938e-0fcd55eb075f",
      "type": "device",
      "display_name": "Example Travel Planner instance"
    },
    "client_declaration_ref": {
      "id": "urn:uuid:763f4e35-b4d7-4931-9510-59d89db1a1b4",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/client-declaration.schema.json",
      "digest": "sha-256:59079539153538f2d01f8c2f45ed0bcda2ff0d9caf81ea54605428c54ada0501"
    }
  },
  "processing_recipients": [
    {
      "recipient": {
        "id": "https://model-provider.example",
        "type": "service",
        "display_name": "Example Model Provider"
      },
      "roles": [
        "model_provider"
      ],
      "purpose_codes": [
        "https://opencontinuity.org/vocab/purposes/plan-trip"
      ],
      "receives_plaintext": true
    }
  ],
  "purpose": {
    "code": "https://opencontinuity.org/vocab/purposes/plan-trip",

```

```

    "description": "Plan a ten-day trip while respecting the traveler's accepted lodging preferences."
  },
  "selectors": [
    {
      "layer": "context",
      "space_refs": [
        {
          "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
          "revision": 1,
          "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
          "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
        }
      ],
      "item_categories": [
        "statement"
      ],
      "predicates": [
        "https://opencontinuity.org/vocab/predicates/prefers-hotel-style"
      ],
      "sensitivity_max": "normal"
    }
  ],
  "views": {
    "context": "accepted",
    "evidence": "none",
    "artifacts": "none"
  },
  "requested_not_after": "2026-08-12T20:00:00Z",
  "retention": {
    "mode": "session",
    "legal_retention_may_apply": false
  },
  "processing_uses": {
    "task_use": "permitted",
    "model_training": "prohibited",
    "product_improvement": "prohibited",
    "advertising": "prohibited"
  },
  "offline_use": "prohibited",
  "proposal_rights": {
    "mode": "bounded",
    "operations": [
      "create",
      "revise"
    ]
  },
  "selectors": [
    {
      "layer": "context",
      "item_categories": [
        "statement"
      ],
      "predicates": [
        "https://opencontinuity.org/vocab/predicates/prefers-hotel-style"
      ],
      "sensitivity_max": "normal"
    }
  ],
  "sensitivity_max": "normal",
  "source_attachment": "referenced_authorized"
},
"delegation": {
  "mode": "none"
},
"status": "submitted"
}

```

42.12 Context Space

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
  "type": "context_space",
  "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
  "revision": 1,
  "created_at": "2026-08-12T13:00:00Z",
  "created_by": {

```

```

    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "extensions": {},
  "name": "Japan trip planning",
  "description": "Accepted context used only for planning the October 2026 Japan trip.",
  "subjects": [
    {
      "id": "urn:uuid:7a4ea2e0-94c2-4e8e-b83b-785f0ad78ca9",
      "type": "person",
      "display_name": "Avery Chen"
    }
  ],
  "domain": "personal",
  "authority_model": "self",
  "lifecycle": {
    "status": "active"
  }
}

```

42.13 Continuity Observation

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/profiles/continuity-observation-0.1-draft.schema.json",
  "type": "continuity_observation",
  "id": "urn:uuid:ccb98533-784d-4210-bf37-828bb051b412",
  "revision": 1,
  "created_at": "2026-08-12T16:00:00Z",
  "created_by": {
    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "extensions": {},
  "profile_version": "0.1-draft",
  "target_ref": {
    "id": "urn:uuid:4a1c68f5-8242-45b8-8e54-1fa8acb993c7",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/context-item.schema.json",
    "digest": "sha-256:8812c13be33b3d335c53d007696c804dd0854e67f1b4254c26ed504ee9502937"
  },
  "condition": "https://opencontinuity.org/vocab/conditions/source-available",
  "result": "satisfied",
  "method": "https://opencontinuity.org/methods/provider-status-check",
  "basis_refs": [
    {
      "id": "urn:uuid:8659cadf-69c2-403e-ae91-dd214f1c265b",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
      "digest": "sha-256:6fa9fd088d63a5599714237d0cb3750fc6e046edb83095af25b2cecdb9caa68d"
    }
  ],
  "observed_at": "2026-08-12T15:59:58Z",
  "runtime_source": {
    "actor": {
      "id": "https://context-provider.example",
      "type": "service",
      "display_name": "Example Context Provider"
    },
    "software_name": "Example OCP importer",
    "software_version": "1.2.0",
    "execution_context": "provider",
    "build_digest": "sha-256:3d17c04f13580d204701ea4a0e95f8726f2a5ec50880a7c15078971661e9cf0a"
  },
  "freshness": {
    "status": "current",
    "as_of": "2026-08-12T15:59:58Z",
    "expires_at": "2026-08-12T16:04:58Z"
  },
  "change": "initial",
  "instruction_treatment": "untrusted_data",
  "external_action_authority": "none",

```



```

    },
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000212",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
      "digest": "sha-256:e617c7a9a0db6d6c008c66b5df052d893e392f5690a48d6b0e3b4e8157367a49"
    },
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000213",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
      "digest": "sha-256:01d3af59849455b42805390084f4d647899115e51657c5ae0b2ebc6f28fbc6ae"
    },
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000214",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
      "digest": "sha-256:84f094d757715fa8cb85d98c526d891dbf1fbe788ae69c25ebe6a2dfe56dd2fc"
    }
  ],
  "selected_leaf_refs": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000214",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
      "digest": "sha-256:84f094d757715fa8cb85d98c526d891dbf1fbe788ae69c25ebe6a2dfe56dd2fc"
    }
  ],
  "graph_state": "complete",
  "branches": [
    {
      "branch_id": "original-response",
      "leaf_ref": {
        "id": "urn:uuid:00000000-0000-4000-8000-000000000211",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
        "digest": "sha-256:4a3df2a009e008ea41edb96ae9087440584fd15ffe4a37cca7857adalb108ee1"
      },
      "state": "abandoned"
    },
    {
      "branch_id": "regenerated-response",
      "leaf_ref": {
        "id": "urn:uuid:00000000-0000-4000-8000-000000000214",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
        "digest": "sha-256:84f094d757715fa8cb85d98c526d891dbf1fbe788ae69c25ebe6a2dfe56dd2fc"
      },
      "state": "selected"
    }
  ],
  "participants": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000101",
      "type": "person",
      "display_name": "Synthetic User"
    },
    {
      "id": "https://provider-a.example",
      "type": "service",
      "display_name": "Synthetic Provider A"
    }
  ],
  "ordering_semantics": "parent_edges_then_optional_sibling_order",
  "integrity": {
    "algorithm": "sha-256",
    "canonicalization": "RFC8785",
    "semantic_graph_digest": "sha-256:1ae19963edf7d023fe77e3f17693a293514ed17470ab52ba00b8454bd95e7573"
  },
  "provider_extensions": {
    "provider_a.export_variant": "synthetic-v1"
  },
  "sensitivity": "normal",
  "lifecycle": {
    "status": "active"
  }
}

```

42.15 Discovery

```
{
  "issuer": "https://context-provider.example",
  "ocp_versions": [
    "0.4"
  ],
  "base_uri": "https://context-provider.example/ocp/v0.4",
  "schemas": {
    "context_item": "https://opencontinuity.org/schemas/0.4/context-item.schema.json",
    "context_package": "https://opencontinuity.org/schemas/0.4/context-package.schema.json"
  },
  "bindings": [
    "https://opencontinuity.org/bindings/http/0.4"
  ],
  "profiles": [
    {
      "id": "https://opencontinuity.org/profiles/continuity-observation",
      "version": "0.1-draft",
      "status": "draft",
      "schema_uri": "https://opencontinuity.org/schemas/0.4/profiles/continuity-observation-0.1-draft.schema.json"
    }
  ],
  "authorization": {
    "mechanisms": [
      "oauth"
    ],
    "protected_resource_metadata_uri": "https://context-provider.example/.well-known/oauth-protected-resource"
  },
  "limits": {
    "max_record_bytes": 1048576,
    "max_package_bytes": 16777216,
    "clock_skew_seconds": 60,
    "grant_status_max_age_seconds": 300,
    "event_cursor_retention_seconds": 3600,
    "event_page_max_items": 100
  }
}
```

42.16 Grant Receipt Metadata Layers

```
{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/grant-receipt.schema.json",
  "type": "grant_receipt",
  "id": "urn:uuid:0fc71ecf-a1c7-493a-bb4e-6c8d09fb9e5c",
  "revision": 1,
  "created_at": "2026-08-12T15:02:00Z",
  "created_by": {
    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "extensions": {},
  "correlation_id": "urn:uuid:bd6ee087-a8b4-450d-9f84-aa89b8b978c0",
  "request_ref": {
    "id": "urn:uuid:c58c54e6-1c7a-48ce-bd73-1303db3cb7b4",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/context-request.schema.json",
    "digest": "sha-256:2ace0986062f1899e9c4fc3a88dad409c2a3eded7b2663979691f94cc41f01b5"
  },
  "request_status_ref": {
    "uri": "https://context-provider.example/ocp/v0.4/context-requests/c58c54e6-1c7a-48ce-bd73-1303db3cb7b4/status",
    "last_checked_at": "2026-08-12T15:02:00Z"
  },
  "authorization_ref": {
    "uri": "https://authorization.example/decisions/334d86a1-68f6-4434-a998-e12ce55dbd41",
    "last_checked_at": "2026-08-12T15:02:00Z"
  },
  "authorization_system": "oauth",
  "authorizer": {
    "id": "urn:uuid:7a4ea2e0-94c2-4e8e-b83b-785f0ad78ca9",
    "type": "person",
  }
}
```

```

    "display_name": "Avery Chen"
  },
  "authority_basis": {
    "type": "self",
    "scope_note": "Self-authorization for personal travel context."
  },
  "purpose": {
    "code": "https://opencontinuity.org/vocab/purposes/plan-trip",
    "description": "Plan a ten-day trip while respecting the traveler's accepted lodging preferences."
  },
  "approved_selectors": [
    {
      "layer": "context",
      "space_refs": [
        {
          "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
          "revision": 1,
          "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
          "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
        }
      ],
      "item_categories": [
        "statement"
      ],
      "predicates": [
        "https://opencontinuity.org/vocab/predicates/prefers-hotel-style"
      ],
      "sensitivity_max": "normal"
    },
    {
      "layer": "evidence",
      "space_refs": [
        {
          "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
          "revision": 1,
          "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
          "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
        }
      ],
      "sensitivity_max": "sensitive"
    },
    {
      "layer": "artifact",
      "space_refs": [
        {
          "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
          "revision": 1,
          "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
          "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
        }
      ],
      "sensitivity_max": "normal"
    }
  ],
  "approved_views": {
    "context": "accepted",
    "evidence": "metadata",
    "artifacts": "metadata"
  },
  "delivery_recipient": {
    "recipient": {
      "id": "https://travel-client.example",
      "type": "service",
      "display_name": "Example Travel Planner"
    },
    "client_instance": {
      "id": "urn:uuid:43af637a-fcfe-42d5-938e-0fcd55eb075f",
      "type": "device",
      "display_name": "Example Travel Planner instance"
    },
    "client_declaration_ref": {
      "id": "urn:uuid:763f4e35-b4d7-4931-9510-59d89db1a1b4",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/client-declaration.schema.json",
      "digest": "sha-256:59079539153538f2d01f8c2f45ed0bcda2ff0d9caf81ea54605428c54ada0501"
    }
  },
}

```

```

"processing_recipients": [
  {
    "recipient": {
      "id": "https://model-provider.example",
      "type": "service",
      "display_name": "Example Model Provider"
    },
    "roles": [
      "model_provider"
    ],
    "purpose_codes": [
      "https://opencontinuity.org/vocab/purposes/plan-trip"
    ],
    "receives_plaintext": true
  }
],
"not_before": "2026-08-12T15:02:00Z",
"expires_at": "2026-08-12T20:00:00Z",
"retention": {
  "mode": "session",
  "legal_retention_may_apply": false
},
"processing_uses": {
  "task_use": "permitted",
  "model_training": "prohibited",
  "product_improvement": "prohibited",
  "advertising": "prohibited"
},
"offline_use": "prohibited",
"proposal_rights": {
  "mode": "bounded",
  "operations": [
    "create",
    "revise"
  ],
},
"selectors": [
  {
    "layer": "context",
    "item_categories": [
      "statement"
    ],
    "predicates": [
      "https://opencontinuity.org/vocab/predicates/prefers-hotel-style"
    ],
    "sensitivity_max": "normal"
  }
],
"sensitivity_max": "normal",
"source_attachment": "referenced_authorized"
},
"delegation": {
  "mode": "none"
},
"status": "active",
"status_ref": {
  "uri": "https://context-provider.example/ocp/v0.4/grant-receipts/0fc71ecf-a1c7-493a-bb4e-6c8d09fb9e5c/status",
  "last_checked_at": "2026-08-12T15:02:00Z"
}
}

```

42.17 Grant Receipt

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/grant-receipt.schema.json",
  "type": "grant_receipt",
  "id": "urn:uuid:d7681bd6-5807-4e3c-bfd8-33c4f0799bfd",
  "revision": 1,
  "created_at": "2026-08-12T15:02:00Z",
  "created_by": {
    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "extensions": {},
}

```

```

"correlation_id": "urn:uuid:bd6ee087-a8b4-450d-9f84-aa89b8b978c0",
"request_ref": {
  "id": "urn:uuid:39ef7160-7243-41e9-93cb-3a6daec9f566",
  "revision": 1,
  "schema_uri": "https://opencontinuity.org/schemas/0.4/context-request.schema.json",
  "digest": "sha-256:6f79ce11f397b728fb2bfe2f24e64912e021b0c24c0c436c6ae1ca7d9b31cb8e"
},
"request_status_ref": {
  "uri": "https://context-provider.example/ocp/v0.4/context-requests/39ef7160-7243-41e9-93cb-3a6daec9f566/status",
  "last_checked_at": "2026-08-12T15:02:00Z"
},
"authorization_ref": {
  "uri": "https://authorization.example/decisions/604f4b86-f565-48f6-a662-b4735aad6c84",
  "last_checked_at": "2026-08-12T15:02:00Z"
},
"authorization_system": "oauth",
"authorizer": {
  "id": "urn:uuid:7a4ea2e0-94c2-4e8e-b83b-785f0ad78ca9",
  "type": "person",
  "display_name": "Avery Chen"
},
"authority_basis": {
  "type": "self",
  "scope_note": "Self-authorization for personal travel context."
},
"purpose": {
  "code": "https://opencontinuity.org/vocab/purposes/plan-trip",
  "description": "Plan a ten-day trip while respecting the traveler's accepted lodging preferences."
},
"approved_selectors": [
  {
    "layer": "context",
    "space_refs": [
      {
        "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
        "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
      }
    ],
    "item_categories": [
      "statement"
    ],
    "predicates": [
      "https://opencontinuity.org/vocab/predicates/prefers-hotel-style"
    ],
    "sensitivity_max": "normal"
  }
],
"approved_views": {
  "context": "accepted",
  "evidence": "none",
  "artifacts": "none"
},
"delivery_recipient": {
  "recipient": {
    "id": "https://travel-client.example",
    "type": "service",
    "display_name": "Example Travel Planner"
  },
  "client_instance": {
    "id": "urn:uuid:43af637a-fcfe-42d5-938e-0fcd55eb075f",
    "type": "device",
    "display_name": "Example Travel Planner instance"
  },
  "client_declaration_ref": {
    "id": "urn:uuid:763f4e35-b4d7-4931-9510-59d89db1a1b4",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/client-declaration.schema.json",
    "digest": "sha-256:59079539153538f2d01f8c2f45ed0bcda2ff0d9caf81ea54605428c54ada0501"
  }
},
"processing_recipients": [
  {
    "recipient": {
      "id": "https://model-provider.example",

```

```

    "type": "service",
    "display_name": "Example Model Provider"
  },
  "roles": [
    "model_provider"
  ],
  "purpose_codes": [
    "https://opencontinuity.org/vocab/purposes/plan-trip"
  ],
  "receives_plaintext": true
}
],
"not_before": "2026-08-12T15:02:00Z",
"expires_at": "2026-08-12T20:00:00Z",
"retention": {
  "mode": "session",
  "legal_retention_may_apply": false
},
"processing_uses": {
  "task_use": "permitted",
  "model_training": "prohibited",
  "product_improvement": "prohibited",
  "advertising": "prohibited"
},
"offline_use": "prohibited",
"proposal_rights": {
  "mode": "bounded",
  "operations": [
    "create",
    "revise"
  ],
},
"selectors": [
  {
    "layer": "context",
    "item_categories": [
      "statement"
    ],
    "predicates": [
      "https://opencontinuity.org/vocab/predicates/prefers-hotel-style"
    ],
    "sensitivity_max": "normal"
  }
],
"sensitivity_max": "normal",
"source_attachment": "referenced_authorized"
},
"delegation": {
  "mode": "none"
},
"status": "active",
"status_ref": {
  "uri": "https://context-provider.example/ocp/v0.4/grant-receipts/d7681bd6-5807-4e3c-bfd8-33c4f0799bfd/status",
  "last_checked_at": "2026-08-12T15:02:00Z"
}
}
}

```

42.18 History Bundle

```

{
  "bundle_id": "urn:uuid:00000000-0000-4000-8000-000000000800",
  "conversation": {
    "ocp_version": "0.4",
    "schema_uri": "https://opencontinuity.org/schemas/0.4/conversation.schema.json",
    "type": "conversation",
    "id": "urn:uuid:00000000-0000-4000-8000-000000000200",
    "revision": 1,
    "created_at": "2026-08-13T12:00:00Z",
    "created_by": {
      "id": "https://provider-one.example",
      "type": "service",
      "display_name": "Synthetic OCP Provider 1"
    },
  },
  "extensions": {},
  "profile_id": "https://opencontinuity.org/profiles/conversation",
  "profile_version": "0.4",
}

```

[illegible]

```

    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000214",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
      "digest": "sha-256:84f094d757715fa8cb85d98c526d891dbf1fbe788ae69c25ebe6a2dfe56dd2fc"
    }
  ],
  "graph_state": "complete",
  "branches": [
    {
      "branch_id": "original-response",
      "leaf_ref": {
        "id": "urn:uuid:00000000-0000-4000-8000-000000000211",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
        "digest": "sha-256:4a3df2a009e008ea41edb96ae9087440584fd15ffe4a37cca7857ada1b108ee1"
      },
      "state": "abandoned"
    },
    {
      "branch_id": "regenerated-response",
      "leaf_ref": {
        "id": "urn:uuid:00000000-0000-4000-8000-000000000214",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
        "digest": "sha-256:84f094d757715fa8cb85d98c526d891dbf1fbe788ae69c25ebe6a2dfe56dd2fc"
      },
      "state": "selected"
    }
  ],
  "participants": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000101",
      "type": "person",
      "display_name": "Synthetic User"
    },
    {
      "id": "https://provider-a.example",
      "type": "service",
      "display_name": "Synthetic Provider A"
    }
  ],
  "ordering_semantics": "parent_edges_then_optional_sibling_order",
  "integrity": {
    "algorithm": "sha-256",
    "canonicalization": "RFC8785",
    "semantic_graph_digest":
"sha-256:1ae19963edf7d023fe77e3f17693a293514ed17470ab52ba00b8454bd95e7573"
  },
  "provider_extensions": {
    "provider_a.export_variant": "synthetic-v1"
  },
  "sensitivity": "normal",
  "lifecycle": {
    "status": "active"
  }
},
"messages": [
  {
    "ocp_version": "0.4",
    "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
    "type": "message",
    "id": "urn:uuid:00000000-0000-4000-8000-000000000210",
    "revision": 1,
    "created_at": "2026-08-01T10:00:00Z",
    "created_by": {
      "id": "https://provider-one.example",
      "type": "service",
      "display_name": "Synthetic OCP Provider 1"
    },
    "extensions": {},
    "profile_id": "https://opencontinuity.org/profiles/conversation",
    "profile_version": "0.4",
    "conversation_ref": {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000200",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/conversation.schema.json"
    }
  }
]

```

```

    },
    "sender": {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000101",
      "type": "person",
      "display_name": "Synthetic User"
    },
    "source_role": "user",
    "message_kind": "user_turn",
    "parent_refs": [],
    "sent_at": "2026-08-01T10:00:00Z",
    "source_timestamp_precision": "second",
    "content_parts": [
      {
        "part_id": "urn:uuid:00000000-0000-4000-8000-000000000410",
        "type": "text",
        "text": "Please make a visual summary of the attached reference.",
        "language": "en",
        "instruction_treatment": "untrusted_data"
      },
      {
        "part_id": "urn:uuid:00000000-0000-4000-8000-000000000411",
        "type": "artifact_ref",
        "artifact_ref": {
          "id": "urn:uuid:00000000-0000-4000-8000-000000000300",
          "revision": 1,
          "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",
          "digest": "sha-256:e80526f8a4b34a46c2dba633440a155e770d793d3a40c92c838cc5e3446ea395"
        },
        "relationship": "attachment",
        "semantic_type_hint": "image",
        "content_disclosed": false,
        "instruction_treatment": "untrusted_data"
      }
    ],
    "artifact_refs": [
      {
        "id": "urn:uuid:00000000-0000-4000-8000-000000000300",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",
        "digest": "sha-256:e80526f8a4b34a46c2dba633440a155e770d793d3a40c92c838cc5e3446ea395"
      }
    ],
    "citation_refs": [],
    "tool_interaction_refs": [],
    "provider_message_id": "provider-message-210",
    "generation_metadata": {},
    "sibling_order": 0,
    "instruction_treatment": "untrusted_data",
    "redaction_state": {
      "state": "none"
    },
    "sensitivity": "normal",
    "lifecycle": {
      "status": "active"
    },
    "provider_extensions": {},
    "integrity": {
      "algorithm": "sha-256",
      "canonicalization": "RFC8785",
      "content_parts_digest":
"sha-256:f180cb0153a01782f1eade2d8400800424f4e538e83eb35e4a7ebc162bdf24e3"
    },
    {
      "ocp_version": "0.4",
      "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
      "type": "message",
      "id": "urn:uuid:00000000-0000-4000-8000-000000000211",
      "revision": 1,
      "created_at": "2026-08-01T10:00:03Z",
      "created_by": {
        "id": "https://provider-one.example",
        "type": "service",
        "display_name": "Synthetic OCP Provider 1"
      },
      "extensions": {},
      "profile_id": "https://opencontinuity.org/profiles/conversation",

```

```

"profile_version": "0.4",
"conversation_ref": {
  "id": "urn:uuid:00000000-0000-4000-8000-000000000200",
  "revision": 1,
  "schema_uri": "https://opencontinuity.org/schemas/0.4/conversation.schema.json"
},
"sender": {
  "id": "https://provider-a.example",
  "type": "service",
  "display_name": "Synthetic Provider A"
},
"source_role": "assistant",
"message_kind": "assistant_response",
"parent_refs": [
  {
    "id": "urn:uuid:00000000-0000-4000-8000-000000000210",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
    "digest": "sha-256:cc7b803e2dedd3ed3fc7e69036a607d904116e57ca49c777b3a6cfc0e7f85047"
  }
],
"sent_at": "2026-08-01T10:00:03Z",
"source_timestamp_precision": "second",
"content_parts": [
  {
    "part_id": "urn:uuid:00000000-0000-4000-8000-000000000412",
    "type": "markdown",
    "text": "First generated response.",
    "language": "en",
    "instruction_treatment": "untrusted_data"
  }
],
"artifact_refs": [],
"citation_refs": [],
"tool_interaction_refs": [],
"provider_message_id": "provider-message-211",
"generation_metadata": {
  "generation_family": "synthetic"
},
"sibling_order": 0,
"instruction_treatment": "untrusted_data",
"redaction_state": {
  "state": "none"
},
"sensitivity": "normal",
"lifecycle": {
  "status": "active"
},
"provider_extensions": {},
"integrity": {
  "algorithm": "sha-256",
  "canonicalization": "RFC8785",
  "content_parts_digest":
"sha-256:9b76d24af3ec3874267e8732b565a3c2460b5060a767189b3685bd061f6e7d45"
},
},
{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
  "type": "message",
  "id": "urn:uuid:00000000-0000-4000-8000-000000000212",
  "revision": 1,
  "created_at": "2026-08-01T10:00:05Z",
  "created_by": {
    "id": "https://provider-one.example",
    "type": "service",
    "display_name": "Synthetic OCP Provider 1"
  },
  "extensions": {},
  "profile_id": "https://opencontinuity.org/profiles/conversation",
  "profile_version": "0.4",
  "conversation_ref": {
    "id": "urn:uuid:00000000-0000-4000-8000-000000000200",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/conversation.schema.json"
  },
  "sender": {

```

```

    "id": "https://provider-a.example",
    "type": "service",
    "display_name": "Synthetic Provider A"
  },
  "source_role": "assistant",
  "message_kind": "assistant_response",
  "parent_refs": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000210",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
      "digest": "sha-256:cc7b803e2dedd3ed3fc7e69036a607d904116e57ca49c777b3a6cfc0e7f85047"
    }
  ],
  "sent_at": "2026-08-01T10:00:05Z",
  "source_timestamp_precision": "second",
  "content_parts": [
    {
      "part_id": "urn:uuid:00000000-0000-4000-8000-000000000413",
      "type": "markdown",
      "text": "Regenerated response with a citation.",
      "language": "en",
      "instruction_treatment": "untrusted_data"
    },
    {
      "part_id": "urn:uuid:00000000-0000-4000-8000-000000000414",
      "type": "provider_specific",
      "provider_semantic": "provider_a.confidence_band",
      "opaque_value": {
        "band": "green"
      },
      "executable": false,
      "instruction_treatment": "untrusted_data"
    }
  ],
  "artifact_refs": [],
  "citation_refs": [],
  "tool_interaction_refs": [],
  "provider_message_id": "provider-message-212",
  "generation_metadata": {
    "generation_family": "synthetic"
  },
  "sibling_order": 0,
  "instruction_treatment": "untrusted_data",
  "redaction_state": {
    "state": "none"
  },
  "sensitivity": "normal",
  "lifecycle": {
    "status": "active"
  },
  "provider_extensions": {},
  "integrity": {
    "algorithm": "sha-256",
    "canonicalization": "RFC8785",
    "content_parts_digest":
"sha-256:3604223fb8a362dcd0ab243425bf93dc661420a63d81fd3d9aa59a38e2044a36"
  }
},
{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
  "type": "message",
  "id": "urn:uuid:00000000-0000-4000-8000-000000000213",
  "revision": 1,
  "created_at": "2026-08-01T10:01:00Z",
  "created_by": {
    "id": "https://provider-one.example",
    "type": "service",
    "display_name": "Synthetic OCP Provider 1"
  },
  "extensions": {},
  "profile_id": "https://opencontinuity.org/profiles/conversation",
  "profile_version": "0.4",
  "conversation_ref": {
    "id": "urn:uuid:00000000-0000-4000-8000-000000000200",
    "revision": 1,

```

```

    "schema_uri": "https://opencontinuity.org/schemas/0.4/conversation.schema.json"
  },
  "sender": {
    "id": "urn:uuid:00000000-0000-4000-8000-000000000101",
    "type": "person",
    "display_name": "Synthetic User"
  },
  "source_role": "user",
  "message_kind": "user_turn",
  "parent_refs": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000212",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
      "digest": "sha-256:e617c7a9a0db6d6c008c66b5df052d893e392f5690a48d6b0e3b4e8157367a49"
    }
  ],
  "sent_at": "2026-08-01T10:01:00Z",
  "source_timestamp_precision": "second",
  "content_parts": [
    {
      "part_id": "urn:uuid:00000000-0000-4000-8000-000000000415",
      "type": "redacted",
      "redaction": {
        "state": "export_policy_redacted",
        "reason_class": "privacy",
        "detail_code": "CONV_POLICY_REDACTED"
      },
      "original_type": "text",
      "instruction_treatment": "untrusted_data"
    }
  ],
  "artifact_refs": [],
  "citation_refs": [],
  "tool_interaction_refs": [],
  "provider_message_id": "provider-message-213",
  "generation_metadata": {},
  "sibling_order": 0,
  "instruction_treatment": "untrusted_data",
  "redaction_state": {
    "state": "export_policy_redacted"
  },
  "sensitivity": "normal",
  "lifecycle": {
    "status": "active"
  },
  "provider_extensions": {},
  "integrity": {
    "algorithm": "sha-256",
    "canonicalization": "RFC8785",
    "content_parts_digest":
"sha-256:a0eda4e7915587175e2bca10e564d87b26d0c9f5b42dfc1fb16cbe8283b011dc"
  }
},
{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
  "type": "message",
  "id": "urn:uuid:00000000-0000-4000-8000-000000000214",
  "revision": 1,
  "created_at": "2026-08-01T10:01:03Z",
  "created_by": {
    "id": "https://provider-one.example",
    "type": "service",
    "display_name": "Synthetic OCP Provider 1"
  },
  "extensions": {},
  "profile_id": "https://opencontinuity.org/profiles/conversation",
  "profile_version": "0.4",
  "conversation_ref": {
    "id": "urn:uuid:00000000-0000-4000-8000-000000000200",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/conversation.schema.json"
  },
  "sender": {
    "id": "https://provider-a.example",
    "type": "service",

```

```

    "display_name": "Synthetic Provider A"
  },
  "source_role": "assistant",
  "message_kind": "assistant_response",
  "parent_refs": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000213",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
      "digest": "sha-256:01d3af59849455b42805390084f4d647899115e51657c5ae0b2ebc6f28fbc6ae"
    }
  ],
  "sent_at": "2026-08-01T10:01:03Z",
  "source_timestamp_precision": "second",
  "content_parts": [
    {
      "part_id": "urn:uuid:00000000-0000-4000-8000-000000000416",
      "type": "artifact_ref",
      "artifact_ref": {
        "id": "urn:uuid:00000000-0000-4000-8000-000000000301",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",
        "digest": "sha-256:029d752e7e9e0307b6a5a40fa18a2826a4a81b3352a6b26f0d7e76e084979269"
      },
      "relationship": "generated_output",
      "semantic_type_hint": "image",
      "content_disclosed": false,
      "instruction_treatment": "untrusted_data"
    },
    {
      "part_id": "urn:uuid:00000000-0000-4000-8000-000000000417",
      "type": "unsupported",
      "source_semantic": "provider_a.internal_candidate_rank",
      "loss_code": "CONV_CONTENT_PART_UNSUPPORTED",
      "opaque_value": {
        "rank": 3
      },
      "instruction_treatment": "untrusted_data"
    }
  ],
  "artifact_refs": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000301",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",
      "digest": "sha-256:029d752e7e9e0307b6a5a40fa18a2826a4a81b3352a6b26f0d7e76e084979269"
    }
  ],
  "citation_refs": [],
  "tool_interaction_refs": [],
  "provider_message_id": "provider-message-214",
  "generation_metadata": {
    "generation_family": "synthetic"
  },
  "sibling_order": 0,
  "instruction_treatment": "untrusted_data",
  "redaction_state": {
    "state": "none"
  },
  "sensitivity": "normal",
  "lifecycle": {
    "status": "active"
  },
  "provider_extensions": {},
  "integrity": {
    "algorithm": "sha-256",
    "canonicalization": "RFC8785",
    "content_parts_digest":
"sha-256:1b5034aa2323f0f62e9191df25300e6220d695a16df5d1ac4960fd45ed8490a1"
  }
},
"artifacts": [
  {
    "ocp_version": "0.4",
    "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",
    "type": "artifact",

```

[illegible]

[illegible]

```

    "holder": {
      "id": "https://provider-one.example",
      "type": "service",
      "display_name": "Synthetic OCP Provider 1"
    },
    "source_dependency": false,
    "synchronization_status": "not_applicable"
  },
  "representations": [
    {
      "representation_id": "urn:uuid:00000000-0000-4000-8000-000000000311",
      "relation": "original",
      "media_type": "image/png",
      "filename": "generated-image.png",
      "byte_size": 46,
      "content_ref": "urn:ocp:content:synthetic-generated-image",
      "digest": "sha-256:7e9aa08b283bb63ba5e087b5b2aadecd5408d59a3334cf9fb368e2dccf0c40f0",
      "digest_algorithm": "sha-256",
      "verified_at": "2026-08-13T11:52:00Z",
      "verification_method": "bytes_hashed_in_managed_store",
      "availability": "available",
      "independently_retrievable": true,
      "source_dependency": false
    }
  ],
  "rights": [
    {
      "status": "unknown",
      "asserted_by": {
        "id": "https://provider-one.example",
        "type": "service",
        "display_name": "Synthetic OCP Provider 1"
      },
      "basis": "No portable rights assertion was present in the source export.",
      "asserted_at": "2026-08-13T12:00:00Z",
      "scope": "All uses",
      "limitations": "Possession and preservation do not establish copyright or permission."
    }
  ],
  "accessibility": [],
  "lineage_refs": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000320",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/lineage-event.schema.json",
      "digest": "sha-256:e25cca2e321e8c7f82743920dbecacb6944dbe93137a6f1e8bb48555c27879ce"
    }
  ],
  "access": {
    "metadata": "grant_required",
    "content": "grant_required"
  },
  "sensitivity": "normal",
  "lifecycle": {
    "status": "active"
  }
},
{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",
  "type": "artifact",
  "id": "urn:uuid:00000000-0000-4000-8000-000000000302",
  "revision": 1,
  "created_at": "2026-08-13T12:00:00Z",
  "created_by": {
    "id": "https://provider-one.example",
    "type": "service",
    "display_name": "Synthetic OCP Provider 1"
  },
  "extensions": {},
  "profile_id": "https://opencontinuity.org/profiles/artifact",
  "profile_version": "0.4",
  "compatible_core": ">=0.4 <0.5",
  "subjects": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000101",
      "type": "person",

```

[illegible]

```

    }
  },
  "lineage_events": [
    {
      "ocp_version": "0.4",
      "schema_uri": "https://opencontinuity.org/schemas/0.4/lineage-event.schema.json",
      "type": "lineage_event",
      "id": "urn:uuid:00000000-0000-4000-8000-000000000320",
      "revision": 1,
      "created_at": "2026-08-13T12:00:00Z",
      "created_by": {
        "id": "https://provider-one.example",
        "type": "service",
        "display_name": "Synthetic OCP Provider 1"
      },
      "extensions": {},
      "profile_id": "https://opencontinuity.org/profiles/artifact",
      "profile_version": "0.4",
      "relation": "generated_from_artifact",
      "input_refs": [
        {
          "id": "urn:uuid:00000000-0000-4000-8000-000000000300",
          "revision": 1,
          "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json"
        }
      ],
      "output_refs": [
        {
          "id": "urn:uuid:00000000-0000-4000-8000-000000000301",
          "revision": 1,
          "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json"
        }
      ],
      "performed_by": {
        "id": "https://provider-a.example",
        "type": "service",
        "display_name": "Synthetic Provider A"
      },
      "requested_by": {
        "id": "urn:uuid:00000000-0000-4000-8000-000000000101",
        "type": "person",
        "display_name": "Synthetic User"
      },
      "model": {
        "id": "https://provider-a.example/models/synthetic",
        "type": "model",
        "display_name": "Synthetic Model"
      },
      "model_provider": {
        "id": "https://provider-a.example",
        "type": "service",
        "display_name": "Synthetic Provider A"
      },
      "parameter_disclosure": "partial",
      "occurred_at": "2026-08-01T10:01:02Z",
      "external_action_authority": "none"
    }
  ],
  "adapter_manifest": {
    "adapter_id": "https://opencontinuity.org/adapters/provider-a-synthetic",
    "adapter_version": "1.0.0",
    "publisher": {
      "id": "https://provider-one.example",
      "type": "service",
      "display_name": "Synthetic OCP Provider 1"
    },
    "source_provider": "Provider A",
    "source_product": "Synthetic Chat Export",
    "export_family": "provider-a-synthetic-json",
    "source_schema_versions": [
      "1"
    ],
    "supported_ocp_core_versions": [
      ">=0.4 <0.5"
    ],
    "supported_profiles": [

```

Page 121

Page 122

```

    },
    "source_locator": {
      "archive_member": "conversations.json",
      "json_pointer": "/messages/2/metadata/confidence"
    }
  },
  {
    "code": "ART_CONTENT_UNAVAILABLE",
    "class": "content",
    "severity": "warning",
    "meaning": "The referenced PDF bytes were unavailable at import time.",
    "profile_conformance_allowed": true,
    "roundtrip_lossless": false,
    "object_ref": {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000302",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",
      "digest": "sha-256:1fd2fc3313b2443af955e023d2c33de1ec3cb4da7fb762f2489ddf166a768f74"
    },
    "source_locator": {
      "archive_member": "conversations.json",
      "json_pointer": "/assets/2"
    }
  }
],
"accepted_context_created_without_review": 0,
"visibility_scope": {
  "recipient": {
    "id": "https://provider-two.example",
    "type": "service",
    "display_name": "Synthetic OCP Provider 2"
  },
  "withheld_categories_disclosed": false
}
},
"accepted_context_items": [],
"embedded_test_bytes": {
  "urn:ocp:content:synthetic-image":
"c3ludGhldGljIGluZGVwZW5kZW50bHkgcHJlc2VydmVklGlTYWdlIGJ5dGVzCg==",
  "urn:ocp:content:synthetic-generated-image":
"c3ludGhldGljIGluZGVwZW5kZW50bHkgcHJlc2VydmVklGlTYWdlIGJ5dGVzCg=="
}
}

```

42.19 Human Interaction Boundary

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/profiles/human-interaction-boundary-0.1-draft.schema.json",
  "type": "human_interaction_boundary",
  "id": "urn:uuid:4f1e9e74-383c-462b-aead-328cbdb3b03f",
  "revision": 1,
  "created_at": "2026-08-12T16:10:00Z",
  "created_by": {
    "id": "urn:uuid:7a4ea2e0-94c2-4e8e-b83b-785f0ad78ca9",
    "type": "person",
    "display_name": "Avery Chen"
  },
  "extensions": {},
  "profile_version": "0.1-draft",
  "declared_by": {
    "id": "urn:uuid:7a4ea2e0-94c2-4e8e-b83b-785f0ad78ca9",
    "type": "person",
    "display_name": "Avery Chen"
  },
  "signal": "pause_requested",
  "scope": {
    "kind": "session",
    "ref": "urn:uuid:7480cfad-f0aa-48f7-a925-f26516fb483d"
  },
  "purpose": {
    "code": "https://opencontinuity.org/vocab/purposes/interaction-pacing",
    "description": "Honor an explicit pause request during the current planning session."
  },
  "declared_at": "2026-08-12T16:10:00Z",
}

```

```

"expires_at": "2026-08-12T20:00:00Z",
"capture_basis": "explicit_user_control",
"runtime_source": {
  "actor": {
    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "software_name": "Example OCP importer",
  "software_version": "1.2.0",
  "execution_context": "provider",
  "build_digest": "sha-256:3d17c04f13580d204701ea4a0e95f8726f2a5ec50880a7c15078971661e9cf0a"
},
"instruction_treatment": "untrusted_data",
"external_action_authority": "none",
"acceptance_effect": "none"
}

```

42.20 Lineage Event

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/lineage-event.schema.json",
  "type": "lineage_event",
  "id": "urn:uuid:00000000-0000-4000-8000-000000000320",
  "revision": 1,
  "created_at": "2026-08-13T12:00:00Z",
  "created_by": {
    "id": "https://provider-one.example",
    "type": "service",
    "display_name": "Synthetic OCP Provider 1"
  },
  "extensions": {},
  "profile_id": "https://opencontinuity.org/profiles/artifact",
  "profile_version": "0.4",
  "relation": "generated_from_artifact",
  "input_refs": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000300",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json"
    }
  ],
  "output_refs": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000301",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json"
    }
  ],
  "performed_by": {
    "id": "https://provider-a.example",
    "type": "service",
    "display_name": "Synthetic Provider A"
  },
  "requested_by": {
    "id": "urn:uuid:00000000-0000-4000-8000-000000000101",
    "type": "person",
    "display_name": "Synthetic User"
  },
  "model": {
    "id": "https://provider-a.example/models/synthetic",
    "type": "model",
    "display_name": "Synthetic Model"
  },
  "model_provider": {
    "id": "https://provider-a.example",
    "type": "service",
    "display_name": "Synthetic Provider A"
  },
  "parameter_disclosure": "partial",
  "occurred_at": "2026-08-01T10:01:02Z",
  "external_action_authority": "none"
}

```

42.21 Loss Report

```
{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/loss-report.schema.json",
  "type": "loss_report",
  "id": "urn:uuid:00000000-0000-4000-8000-000000000700",
  "revision": 1,
  "created_at": "2026-08-13T12:00:00Z",
  "created_by": {
    "id": "https://provider-one.example",
    "type": "service",
    "display_name": "Synthetic OCP Provider 1"
  },
  "extensions": {},
  "scope": "import",
  "classification": "semantically_lossy",
  "summary": {
    "messages_discovered": 5,
    "messages_preserved": 5,
    "conversation_branches_discovered": 2,
    "conversation_branches_lost": 0,
    "artifacts_discovered": 3,
    "artifacts_independently_preserved": 2,
    "artifacts_source_dependent": 1,
    "provider_fields_opaque_preserved": 1,
    "unsupported_fields_omitted": 1
  },
  "losses": [
    {
      "code": "CONV_CONTENT_PART_UNSUPPORTED",
      "class": "semantic",
      "severity": "warning",
      "meaning": "A provider-internal candidate rank has no portable semantics.",
      "profile_conformance_allowed": true,
      "roundtrip_lossless": false,
      "object_ref": {
        "id": "urn:uuid:00000000-0000-4000-8000-000000000214",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
        "digest": "sha-256:84f094d757715fa8cb85d98c526d891dbf1fbe788ae69c25ebe6a2dfe56dd2fc"
      },
      "source_locator": {
        "archive_member": "conversations.json",
        "json_pointer": "/messages/4/internal_candidate_rank"
      }
    },
    {
      "code": "CONV_PROVIDER_EXTENSION_PRESERVED",
      "class": "semantic",
      "severity": "info",
      "meaning": "Provider-specific confidence metadata was retained opaquely.",
      "profile_conformance_allowed": true,
      "roundtrip_lossless": false,
      "object_ref": {
        "id": "urn:uuid:00000000-0000-4000-8000-000000000212",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
        "digest": "sha-256:e617c7a9a0db6d6c008c66b5df052d893e392f5690a48d6b0e3b4e8157367a49"
      },
      "source_locator": {
        "archive_member": "conversations.json",
        "json_pointer": "/messages/2/metadata/confidence"
      }
    },
    {
      "code": "ART_CONTENT_UNAVAILABLE",
      "class": "content",
      "severity": "warning",
      "meaning": "The referenced PDF bytes were unavailable at import time.",
      "profile_conformance_allowed": true,
      "roundtrip_lossless": false,
      "object_ref": {
        "id": "urn:uuid:00000000-0000-4000-8000-000000000302",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",

```

```

    "digest": "sha-256:1fd2fc3313b2443af955e023d2c33de1ec3cb4da7fb762f2489ddf166a768f74",
    },
    "source_locator": {
      "archive_member": "conversations.json",
      "json_pointer": "/assets/2"
    }
  }
],
"accepted_context_created_without_review": 0,
"visibility_scope": {
  "recipient": {
    "id": "https://provider-two.example",
    "type": "service",
    "display_name": "Synthetic OCP Provider 2"
  },
  "withheld_categories_disclosed": false
}
}

```

42.22 Mapping Run

[illegible]

42.23 Message

```
{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/message.schema.json",
  "type": "message",
  "id": "urn:uuid:00000000-0000-4000-8000-000000000210",
  "revision": 1,
  "created_at": "2026-08-01T10:00:00Z",
  "created_by": {
    "id": "https://provider-one.example",
    "type": "service",
    "display_name": "Synthetic OCP Provider 1"
  },
  "extensions": {},
  "profile_id": "https://opencontinuity.org/profiles/conversation",
  "profile_version": "0.4",
  "conversation_ref": {
    "id": "urn:uuid:00000000-0000-4000-8000-000000000200",
    "revision": 1,
  }
}
```

```

    "schema_uri": "https://opencontinuity.org/schemas/0.4/conversation.schema.json"
  },
  "sender": {
    "id": "urn:uuid:00000000-0000-4000-8000-000000000101",
    "type": "person",
    "display_name": "Synthetic User"
  },
  "source_role": "user",
  "message_kind": "user_turn",
  "parent_refs": [],
  "sent_at": "2026-08-01T10:00:00Z",
  "source_timestamp_precision": "second",
  "content_parts": [
    {
      "part_id": "urn:uuid:00000000-0000-4000-8000-000000000410",
      "type": "text",
      "text": "Please make a visual summary of the attached reference.",
      "language": "en",
      "instruction_treatment": "untrusted_data"
    },
    {
      "part_id": "urn:uuid:00000000-0000-4000-8000-000000000411",
      "type": "artifact_ref",
      "artifact_ref": {
        "id": "urn:uuid:00000000-0000-4000-8000-000000000300",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",
        "digest": "sha-256:e80526f8a4b34a46c2dba633440a155e770d793d3a40c92c838cc5e3446ea395"
      },
      "relationship": "attachment",
      "semantic_type_hint": "image",
      "content_disclosed": false,
      "instruction_treatment": "untrusted_data"
    }
  ],
  "artifact_refs": [
    {
      "id": "urn:uuid:00000000-0000-4000-8000-000000000300",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",
      "digest": "sha-256:e80526f8a4b34a46c2dba633440a155e770d793d3a40c92c838cc5e3446ea395"
    }
  ],
  "citation_refs": [],
  "tool_interaction_refs": [],
  "provider_message_id": "provider-message-210",
  "generation_metadata": {},
  "sibling_order": 0,
  "instruction_treatment": "untrusted_data",
  "redaction_state": {
    "state": "none"
  },
  "sensitivity": "normal",
  "lifecycle": {
    "status": "active"
  },
  "provider_extensions": {},
  "integrity": {
    "algorithm": "sha-256",
    "canonicalization": "RFC8785",
    "content_parts_digest": "sha-256:f180cb0153a01782f1eade2d8400800424f4e538e83eb35e4a7ebc162bdf24e3"
  }
}

```

42.24 Offline Package Manifest

```

{
  "manifest_version": "1.1",
  "ocp_version": "0.4",
  "package_type": "context_package",
  "package_id": "urn:uuid:82b2f316-2adc-4181-b676-c8431fceb32c",
  "created_at": "2026-08-12T15:06:00Z",
  "archive_format": "zip",
  "content_classification": "non_public",
  "payload_members": [
    {

```

```

    "path": "payload/context-package.json",
    "media_type": "application/ocp+json",
    "byte_size": 4096,
    "digest": "sha-256:8ced29d3e102f915e5953d43820df4968e681998dd8d982169ed8b69858a35e3",
    "object_id": "urn:uuid:82b2f316-2adc-4181-b676-c8431fceb32c",
    "schema_uri": "https://opencontinuity.org/schemas/0.4/context-package.schema.json"
  },
],
"security_members": [
  {
    "path": "security/manifest.sig",
    "media_type": "application/octet-stream",
    "byte_size": 96,
    "digest": "sha-256:73aa4db40c283a7171df237789376e6de8b5a20406c9d17c23c7b147d012debc"
  },
  {
    "path": "security/recipients.json",
    "media_type": "application/json",
    "byte_size": 256,
    "digest": "sha-256:8f58c55c91c2fd8f38dc4474e99ab8d0c1637702150a381347b3305b7873582c"
  }
],
"member_set_policy": "exact_declared_set",
"protection": {
  "encryption": {
    "profile": "https://opencontinuity.org/example-profiles/recipient-encryption-review-example",
    "metadata_path": "security/recipients.json"
  },
  "signature": {
    "profile": "https://opencontinuity.org/example-profiles/archive-signature-review-example",
    "signature_path": "security/manifest.sig",
    "signed_view": "jcs_manifest_without_signature_descriptor_and_member"
  }
}
}

```

42.25 Problem

```

{
  "type": "https://opencontinuity.org/problems/grant-inactive",
  "title": "Grant receipt is suspended, expired, revoked, or superseded",
  "status": 409,
  "detail": "The operation cannot continue.",
  "instance": "/ocp/v0.4/context-packages/example",
  "code": "OCP_GRANT_INACTIVE",
  "correlation_id": "urn:uuid:bd6ee087-a8b4-450d-9f84-aa89b8b978c0",
  "retryable": false
}

```

42.26 Provider Adapter Fixture

```

{
  "fixture_version": "1.2",
  "fixture_id": "https://opencontinuity.org/fixtures/provider-adapter/example-assistant-1",
  "description": "Synthetic deterministic mapping from exact raw source bytes, declared source snapshot, runtime source, and adapter version to a normalized unaccepted OCP candidate projection.",
  "source_provider": {
    "id": "https://assistant-source.example",
    "name": "Example Source Assistant"
  },
  "source_export_schema": "example-assistant-export/1.0",
  "adapter_version": "example-assistant-adapter/1.2.0",
  "adapter_context": {
    "source_ref": {
      "id": "urn:uuid:8659cadf-69c2-403e-ae91-dd214f1c265b",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
      "digest": "sha-256:6fa9fd088d63a5599714237d0cb3750fc6e046edb83095af25b2cecd9caa68d"
    },
    "runtime_source": {
      "actor": {
        "id": "https://context-provider.example",
        "type": "service",
        "display_name": "Example Context Provider"
      }
    }
  }
}

```

```

    "software_name": "Example OCP importer",
    "software_version": "1.2.0",
    "execution_context": "provider",
    "build_digest": "sha-256:3d17c04f13580d204701ea4a0e95f8726f2a5ec50880a7c15078971661e9cf0a"
  },
  "input_records": [
    {
      "path": "records/message-44.json",
      "media_type": "application/json",
      "digest": "sha-256:edcf14bbb4b3c918fc695e94fc068aa4d2cc5509640cea81883cd24a90be7b1a",
      "digest_coverage": "raw_decoded_bytes",
      "raw_content_base64":
"eyJyb2xlIjoidXNlciIsInRleHQiOiJJIHVzdWFSbHkgcHJlZmVyIGJvdXRpcXVlIGhvdGVscy4ifQ=="
    },
    {
      "expected_outputs": [
        {
          "source_locator": "records/message-44.json",
          "ocp_type": "context_item",
          "projection": "ocp-0.4-adapter-candidate-projection-v1",
          "expected_projection": {
            "source_locator": "records/message-44.json",
            "ocp_type": "context_item",
            "category": "statement",
            "statement": {
              "predicate": "https://opencontinuity.org/vocab/predicates/prefers-hotel-style",
              "object": {
                "type": "string",
                "value": "boutique",
                "language": "en"
              }
            }
          },
          "epistemic_basis": "provider_inferred",
          "review_status": "candidate",
          "source_ref": {
            "id": "urn:uuid:8659cadf-69c2-403e-ae91-dd214f1c265b",
            "revision": 1,
            "schema_uri": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
            "digest": "sha-256:6fa9fd088d63a5599714237d0cb3750fc6e046edb83095af25b2cecd9caa68d"
          },
          "runtime_source": {
            "actor": {
              "id": "https://context-provider.example",
              "type": "service",
              "display_name": "Example Context Provider"
            },
            "software_name": "Example OCP importer",
            "software_version": "1.2.0",
            "execution_context": "provider",
            "build_digest": "sha-256:3d17c04f13580d204701ea4a0e95f8726f2a5ec50880a7c15078971661e9cf0a"
          },
          "adapter_version": "example-assistant-adapter/1.2.0",
          "instruction_treatment": "untrusted_data"
        },
        {
          "semantic_digest": "sha-256:7ca7724fb89417c609aade393f9ef2a682fea9f30756457ea03aab5f61ae02af"
        }
      ],
      "expected_omissions": [],
      "expected_warnings": []
    }
  ],
  "expected_outputs": [
    {
      "source_locator": "records/message-44.json",
      "ocp_type": "context_item",
      "projection": "ocp-0.4-adapter-candidate-projection-v1",
      "expected_projection": {
        "source_locator": "records/message-44.json",
        "ocp_type": "context_item",
        "category": "statement",
        "statement": {
          "predicate": "https://opencontinuity.org/vocab/predicates/prefers-hotel-style",
          "object": {
            "type": "string",
            "value": "boutique",
            "language": "en"
          }
        }
      },
      "epistemic_basis": "provider_inferred",
      "review_status": "candidate",
      "source_ref": {
        "id": "urn:uuid:8659cadf-69c2-403e-ae91-dd214f1c265b",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
        "digest": "sha-256:6fa9fd088d63a5599714237d0cb3750fc6e046edb83095af25b2cecd9caa68d"
      },
      "runtime_source": {
        "actor": {
          "id": "https://context-provider.example",
          "type": "service",
          "display_name": "Example Context Provider"
        },
        "software_name": "Example OCP importer",
        "software_version": "1.2.0",
        "execution_context": "provider",
        "build_digest": "sha-256:3d17c04f13580d204701ea4a0e95f8726f2a5ec50880a7c15078971661e9cf0a"
      },
      "adapter_version": "example-assistant-adapter/1.2.0",
      "instruction_treatment": "untrusted_data"
    },
    {
      "semantic_digest": "sha-256:7ca7724fb89417c609aade393f9ef2a682fea9f30756457ea03aab5f61ae02af"
    }
  ],
  "expected_omissions": [],
  "expected_warnings": []
}

```

42.27 Provider Adapter Manifest

```

{
  "adapter_id": "https://opencontinuity.org/adapters/provider-a-synthetic",
  "adapter_version": "1.0.0",
  "publisher": {
    "id": "https://provider-one.example",
    "type": "service",
    "display_name": "Synthetic OCP Provider 1"
  },
  "source_provider": "Provider A",
  "source_product": "Synthetic Chat Export",
  "export_family": "provider-a-synthetic-json",
  "source_schema_versions": [

```

```

    "1"
  ],
  "supported_ocp_core_versions": [
    ">=0.4 <0.5"
  ],
  "supported_profiles": [
    {
      "id": "https://opencontinuity.org/profiles/conversation",
      "version": "0.4"
    },
    {
      "id": "https://opencontinuity.org/profiles/artifact",
      "version": "0.4"
    }
  ],
  "input_formats": [
    "application/json",
    "application/zip"
  ],
  "required_members": [
    "conversations.json"
  ],
  "optional_members": [
    "assets/"
  ],
  "capabilities": [
    "conversation_metadata",
    "conversation_branching",
    "messages",
    "message_edits",
    "alternative_generations",
    "artifacts",
    "artifact_bytes",
    "artifact_metadata",
    "generated_media",
    "citations",
    "tool_history",
    "timestamps",
    "provider_ids",
    "provider_interpretations"
  ],
  "mapping_registry_ref": "https://opencontinuity.org/mappings/provider-a/synthetic-v1",
  "known_losses": [
    "CONV_CONTENT_PART_UNSUPPORTED",
    "CONV_PROVIDER_EXTENSION_PRESERVED",
    "ART_CONTENT_UNAVAILABLE"
  ],
  "security_limits": {
    "max_archive_bytes": 1073741824,
    "max_member_bytes": 268435456,
    "max_member_count": 100000,
    "max_nesting_depth": 64,
    "network_fetch": "disabled_during_parse",
    "symlink_handling": "reject",
    "parser_isolation": "required_process_sandbox"
  },
  "determinism": {
    "input_basis":
"raw_source_bytes+source_snapshot+adapter_version+mapping_profile_version+runtime_source_identity",
    "canonicalization": "RFC8785",
    "excluded_fields": [
      "mapping_run_id",
      "started_at",
      "completed_at"
    ]
  },
  "fixture_suite_ref": "https://opencontinuity.org/fixtures/provider-a-synthetic-v1",
  "released_at": "2026-08-13T12:00:00Z"
}

```

42.28 Provider Mapping

```

{
  "mapping_id": "urn:uuid:00000000-0000-4000-8000-000000000501",
  "source_field_or_semantic": "conversation.id",
  "target_object": "conversation",

```

```

    "target_field_or_semantic": "provider_conversation_id",
    "mapping_type": "exact",
    "loss_class": "none",
    "transformation": "UTF-8 string copy.",
    "source_locator_support": [
      "archive_member",
      "json_pointer"
    ],
    "accepted_context_eligible": false
  }
}

```

42.29 Review Decision

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/review-decision.schema.json",
  "type": "review_decision",
  "id": "urn:uuid:0bd0df97-5fb1-4db5-9937-e7136fbefbf6",
  "revision": 1,
  "created_at": "2026-08-12T15:04:00Z",
  "created_by": {
    "id": "urn:uuid:7a4ea2e0-94c2-4e8e-b83b-785f0ad78ca9",
    "type": "person",
    "display_name": "Avery Chen"
  },
  "extensions": {},
  "proposal_ref": {
    "id": "urn:uuid:d21a33ff-06f8-4ef8-9d56-0fde809fe0d0",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/update-proposal.schema.json",
    "digest": "sha-256:b481ea3601fb713b06c08319ea6d3a7fb7d1fe57c58638a31c9422048c881d2b"
  },
  "reviewed_by": {
    "id": "urn:uuid:7a4ea2e0-94c2-4e8e-b83b-785f0ad78ca9",
    "type": "person",
    "display_name": "Avery Chen"
  },
  "authority_basis": {
    "type": "self",
    "scope_note": "Accepted only for the Japan trip context space."
  },
  "outcome": "accepted_with_changes",
  "resulting_item_commitment": {
    "id": "urn:uuid:4a1c68f5-8242-45b8-8e54-1fa8acb993c7",
    "revision": 1,
    "accepted_scope": {
      "space_refs": [
        {
          "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
          "revision": 1,
          "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
          "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
        }
      ]
    },
    "purpose_codes": [
      "https://opencontinuity.org/vocab/purposes/plan-trip"
    ]
  },
  "semantic_digest": "sha-256:77a2ec6f2ccadde6f32666f36fc946a0dd3bb801999ad299aff25e5b525d1b05",
  "projection": "ocp-0.4-context-item-accepted-result-v1"
},
"reason": "Confirmed the preference and narrowed it to the Japan trip planning purpose.",
"decided_at": "2026-08-12T15:04:00Z"
}

```

42.30 Source Record

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
  "type": "source_record",
  "id": "urn:uuid:8659cadf-69c2-403e-ae91-dd214f1c265b",
  "revision": 1,
  "created_at": "2026-08-12T13:02:00Z",
  "created_by": {

```

```

    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "extensions": {},
  "source_type": "conversation",
  "source_authority": {
    "id": "https://assistant-source.example",
    "type": "service",
    "display_name": "Example Source Assistant"
  },
  "runtime_source": {
    "actor": {
      "id": "https://context-provider.example",
      "type": "service",
      "display_name": "Example Context Provider"
    },
    "software_name": "Example OCP importer",
    "software_version": "1.2.0",
    "execution_context": "provider",
    "build_digest": "sha-256:3d17c04f13580d204701ea4a0e95f8726f2a5ec50880a7c15078971661e9cf0a"
  },
  "source_object_id": "conversation-44",
  "acquisition_method": "export",
  "observed_at": "2026-03-10T18:00:00Z",
  "acquired_at": "2026-08-12T13:01:00Z",
  "adapter_version": "example-assistant-adapter/1.2.0",
  "trust_treatment": "user_supplied",
  "instruction_treatment": "untrusted_data",
  "content": {
    "media_type": "application/json",
    "byte_size": 1842,
    "digest": "sha-256:cf52938bc36112aa4ea448a49b007bc2c4a8a987f941083524c8d034404884ba",
    "content_ref": "https://context-provider.example/sources/conversation-44",
    "redacted": true
  },
  "availability": "available",
  "sensitivity": "sensitive",
  "lifecycle": {
    "status": "active"
  }
}

```

42.31 Transfer Manifest

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/bindings/transfer/transfer-
manifest.schema.json",
  "type": "transfer_manifest",
  "id": "urn:uuid:2e46d5a9-a5ae-42cd-b396-0d3cb1df9696",
  "revision": 1,
  "created_at": "2026-08-12T16:30:00Z",
  "created_by": {
    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "extensions": {},
  "source_provider": {
    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "destination_provider": {
    "id": "https://destination.example",
    "type": "service",
    "display_name": "Example Destination"
  },
  "state": "validated",
  "entries": [
    {
      "source_ref": {
        "id": "urn:uuid:4a1c68f5-8242-45b8-8e54-1fa8acb993c7",
        "revision": 1,
        "schema_uri": "https://opencontinuity.org/schemas/0.4/context-item.schema.json",

```

```

    "digest": "sha-256:8812c13be33b3d335c53d007696c804dd0854e67f1b4254c26ed504ee9502937"
  },
  "export_id": "urn:uuid:6a3fd4f7-bccb-4d98-a7e2-3fe193e99e36",
  "path": "records/context-item.json",
  "transfer_status": "verified",
  "destination_ref": {
    "id": "urn:uuid:4a1c68f5-8242-45b8-8e54-1fa8acb993c7",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/context-item.schema.json",
    "digest": "sha-256:8812c13be33b3d335c53d007696c804dd0854e67f1b4254c26ed504ee9502937"
  },
  "losses": []
}
],
"exceptions": [],
"source_disposition": "unchanged",
"completed_at": "2026-08-12T16:31:00Z"
}

```

42.32 Update Proposal

```

{
  "ocp_version": "0.4",
  "schema_uri": "https://opencontinuity.org/schemas/0.4/update-proposal.schema.json",
  "type": "update_proposal",
  "id": "urn:uuid:d21a33ff-06f8-4ef8-9d56-0fde809fe0d0",
  "revision": 1,
  "created_at": "2026-08-12T15:03:00Z",
  "created_by": {
    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "extensions": {},
  "submitted_by": {
    "id": "https://context-provider.example",
    "type": "service",
    "display_name": "Example Context Provider"
  },
  "grant_receipt_ref": {
    "id": "urn:uuid:d7681bd6-5807-4e3c-bfd8-33c4f0799bfd",
    "revision": 1,
    "schema_uri": "https://opencontinuity.org/schemas/0.4/grant-receipt.schema.json",
    "digest": "sha-256:4657179eadb7ccb43cbef2c4fa1f990cac8443d995ae775fa7fa00de7f66ad47"
  },
  "grant_status_ref": {
    "uri": "https://context-provider.example/ocp/v0.4/grant-receipts/d7681bd6-5807-4e3c-bfd8-33c4f0799bfd/status",
    "last_checked_at": "2026-08-12T15:02:00Z"
  },
  "operation": "create",
  "proposed_item": {
    "ocp_version": "0.4",
    "schema_uri": "https://opencontinuity.org/schemas/0.4/context-item.schema.json",
    "type": "context_item",
    "id": "urn:uuid:2bb6cf0e-3650-435b-af8f-25b46878b72c",
    "revision": 1,
    "created_at": "2026-08-12T13:03:00Z",
    "created_by": {
      "id": "https://context-provider.example",
      "type": "service",
      "display_name": "Example Context Provider"
    },
    "extensions": {},
    "subjects": [
      {
        "id": "urn:uuid:7a4ea2e0-94c2-4e8e-b83b-785f0ad78ca9",
        "type": "person",
        "display_name": "Avery Chen"
      }
    ],
    "space_ref": {
      "id": "urn:uuid:a27fd923-d0a7-4dde-972c-46c40fe575a5",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
      "digest": "sha-256:6909de5bcd6a97dc15d7c906892e197ec069630ec8125b23deb00e4537446445"
    }
  }
}

```

```

    },
    "category": "statement",
    "statement": {
      "predicate": "https://opencontinuity.org/vocab/predicates/prefers-hotel-style",
      "object": {
        "type": "string",
        "value": "boutique",
        "language": "en"
      }
    },
    "epistemic_basis": "provider_inferred",
    "review": {
      "status": "candidate"
    },
    "validity": {
      "observed_at": "2026-03-10T18:00:00Z",
      "review_after": "2027-02-12T00:00:00Z"
    },
    "provenance": [
      {
        "source_ref": {
          "id": "urn:uuid:8659cadf-69c2-403e-ae91-dd214f1c265b",
          "revision": 1,
          "schema_uri": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
          "digest": "sha-256:6fa9fd088d63a5599714237d0cb3750fc6e046edb83095af25b2cecdb9caa68d"
        },
        "method": "inference",
        "observed_at": "2026-03-10T18:00:00Z",
        "acquired_at": "2026-08-12T13:01:00Z",
        "producer": {
          "id": "https://context-provider.example",
          "type": "service",
          "display_name": "Example Context Provider"
        },
        "runtime_source": {
          "actor": {
            "id": "https://context-provider.example",
            "type": "service",
            "display_name": "Example Context Provider"
          },
          "software_name": "Example OCP importer",
          "software_version": "1.2.0",
          "execution_context": "provider",
          "build_digest": "sha-256:3d17c04f13580d204701ea4a0e95f8726f2a5ec50880a7c15078971661e9cf0a"
        },
        "adapter_version": "example-assistant-adapter/1.2.0",
        "redacted": true
      }
    ],
    "sensitivity": "normal",
    "lifecycle": {
      "status": "active"
    }
  },
  "source_refs": [
    {
      "id": "urn:uuid:8659cadf-69c2-403e-ae91-dd214f1c265b",
      "revision": 1,
      "schema_uri": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
      "digest": "sha-256:6fa9fd088d63a5599714237d0cb3750fc6e046edb83095af25b2cecdb9caa68d"
    }
  ],
  "rationale": "The source statement may indicate a preference, but it requires subject review before portable use.",
  "status": "under_review"
}

```

43. Normative schemas

The JSON Schema files below are the exact machine-readable schemas in this release. The versioned source paths, not copied prose excerpts, are authoritative for representation. Relative `$ref` resolution is evaluated from each schema's `$id` and the pinned local catalog used by the harness.

43.1 bindings/http/context-request-create.schema.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/bindings/http/context-request-create.schema.json",
  "title": "OCP 0.4 HTTP context-request create input",
  "description": "Client-supplied semantics for creating a context request. The provider authenticates and binds the caller, then authors ocp_version, schema_uri, type, id, revision, created_at, created_by, correlation_id, status=\"submitted\", and extensions in the persisted ContextRequest returned by the operation.",
  "type": "object",
  "required": [
    "delivery_recipient",
    "processing_recipients",
    "purpose",
    "selectors",
    "views",
    "requested_not_after",
    "retention",
    "processing_uses",
    "offline_use",
    "proposal_rights",
    "delegation"
  ],
  "properties": {
    "delivery_recipient": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/deliveryRecipient"},
    "processing_recipients": {"type": "array", "items": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/processingRecipient"}, "uniqueItems": true},
    "purpose": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/purpose"},
    "selectors": {"type": "array", "items": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/selector"}, "minItems": 1},
    "views": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/disclosureViews"},
    "requested_not_before": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"},
    "requested_not_after": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"},
    "retention": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/retention"},
    "processing_uses": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/processingUses"},
    "offline_use": {"enum": ["prohibited", "requested"]},
    "proposal_rights": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/proposalRights"},
    "delegation": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/delegation"}
  },
  "additionalProperties": false
}
```

43.2 bindings/http/discovery.schema.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/bindings/http/discovery.schema.json",
  "title": "OCP 0.4 HTTP discovery document",
  "description": "Public discovery metadata served from the issuer origin at /.well-known/ocp-configuration. base_uri identifies the protected OCP API root and does not prefix the well-known discovery path. JSON Schema enforces HTTPS shape; the HTTP binding's behavioral validation additionally requires exact same-origin issuer, base, discovery, and metadata locations unless a registered federation profile authorizes otherwise.",
  "type": "object",
  "required": ["issuer", "ocp_versions", "base_uri", "schemas", "bindings", "profiles", "authorization", "limits"],
}
```

```

"properties": {
  "issuer": {"type": "string", "format": "uri", "pattern": "^https://"},
  "discovery_uri": {"description": "Absolute URI of this document; when present its path is /.well-known/ocp-configuration on the issuer origin.", "type": "string", "format": "uri", "pattern": "^https://[^\/?#]+/\.\.well-known/ocp-configuration$"},
  "ocp_versions": {"type": "array", "items": {"const": "0.4"}, "minItems": 1, "uniqueItems": true},
  "base_uri": {"type": "string", "format": "uri", "pattern": "^https://"},
  "schemas": {"type": "object", "propertyNames": {"pattern": "^[a-z][a-z0-9_]{0,63}$"},
  "additionalProperties": {"type": "string", "format": "uri"}},
  "bindings": {"type": "array", "items": {"type": "string", "format": "uri", "uniqueItems": true},
  "profiles": {"type": "array", "items": {"type": "object", "required": ["id", "version", "status", "schema_uri"], "properties": {"id": {"type": "string", "format": "uri"}, "version": {"type": "string"}, "status": {"enum": ["draft", "registered", "deprecated"]}, "schema_uri": {"type": "string", "format": "uri"}}, "additionalProperties": false}},
  "authorization": {"type": "object", "required": ["mechanisms"], "properties": {"mechanisms": {"type": "array", "items": {"enum": ["oauth", "gnap", "uma", "solid", "local_policy", "os_consent", "other"]}, "minItems": 1, "uniqueItems": true}, "protected_resource_metadata_uri": {"type": "string", "format": "uri", "pattern": "^https://"}, "authorization_server_metadata_uri": {"type": "string", "format": "uri", "pattern": "^https://"}, "additionalProperties": false},
  "limits": {
    "description": "Quantitative HTTP-binding limits. Clients use the advertised values and must not substitute binding-wide defaults.",
    "type": "object",
    "required": ["clock_skew_seconds", "grant_status_max_age_seconds", "event_cursor_retention_seconds", "event_page_max_items"],
    "properties": {
      "max_record_bytes": {"type": "integer", "minimum": 1},
      "max_package_bytes": {"type": "integer", "minimum": 1},
      "clock_skew_seconds": {"type": "integer", "minimum": 0},
      "grant_status_max_age_seconds": {"description": "Maximum permitted age of an authoritative grant-status observation for any grant-authorized evaluation, including package issuance and proposal review decisions.", "type": "integer", "minimum": 1},
      "event_cursor_retention_seconds": {"description": "Minimum number of seconds for which an issued event cursor remains usable, unless its authorization context changes.", "type": "integer", "minimum": 1},
      "event_page_max_items": {"description": "Maximum number of event notifications returned in one page.", "type": "integer", "minimum": 1}
    },
    "additionalProperties": false
  },
  },
  "additionalProperties": false
}

```

43.3 bindings/http/event-notification.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/bindings/http/event-notification.schema.json",
  "title": "OCP 0.4 HTTP event notification",
  "description": "CloudEvents 1.0 structured event for the baseline HTTP binding. Its exact dataschema identifies this schema; each advertised type is bound to one registry axis and a legal transition or exact notification code. Lifecycle types are limited to resources with an authoritative HTTP read/status contract so cursor-gap reconciliation is possible; the only notification-axis type is the package-integrity signal, reconciled through that same package contract. Other registered types require a profile that supplies an authoritative resolver. Data is reference-first and contains no raw context, evidence, artifact bytes, tokens, or credentials.",
  "$defs": {
    "lifecycleData": {
      "type": "object",
      "required": [
        "object_ref",
        "revision",
        "from_state",
        "to_state",
        "correlation_id",
        "digest"
      ],
    },
    "properties": {
      "object_ref": {
        "type": "string",
        "format": "uri"
      },
      "revision": {
        "type": "integer",
        "minimum": 1
      }
    }
  }
}

```

```

    },
    "correlation_id": {
      "type": "string",
      "format": "uri"
    },
    "digest": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/digest"
    },
    "from_state": {
      "type": "string",
      "minLength": 1,
      "maxLength": 100
    },
    "to_state": {
      "type": "string",
      "minLength": 1,
      "maxLength": 100
    }
  },
  "additionalProperties": false
},
"reviewData": {
  "type": "object",
  "required": [
    "object_ref",
    "revision",
    "from_review_status",
    "to_review_status",
    "correlation_id",
    "digest"
  ],
  "properties": {
    "object_ref": {
      "type": "string",
      "format": "uri"
    },
    "revision": {
      "type": "integer",
      "minimum": 1
    },
    "correlation_id": {
      "type": "string",
      "format": "uri"
    },
    "digest": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/digest"
    },
    "from_review_status": {
      "enum": [
        "candidate",
        "accepted",
        "disputed",
        "superseded",
        "withdrawn"
      ]
    },
    "to_review_status": {
      "enum": [
        "candidate",
        "accepted",
        "disputed",
        "superseded",
        "withdrawn"
      ]
    }
  },
  "additionalProperties": false
},
"notificationData": {
  "type": "object",
  "required": [
    "object_ref",
    "revision",
    "notification_code",
    "correlation_id",
    "digest"
  ],

```

```

    "properties": {
      "object_ref": {
        "type": "string",
        "format": "uri"
      },
      "revision": {
        "type": "integer",
        "minimum": 1
      },
      "correlation_id": {
        "type": "string",
        "format": "uri"
      },
      "digest": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/digest"
      },
      "notification_code": {
        "enum": [
          "ocp.authorization.failed",
          "ocp.extension.unsupported_critical",
          "ocp.import.limit_exceeded",
          "ocp.package.integrity_failed",
          "ocp.security.incident_detected"
        ]
      }
    },
    "additionalProperties": false
  },
  "type": "object",
  "required": [
    "specversion",
    "id",
    "source",
    "type",
    "time",
    "subject",
    "dataschema",
    "data"
  ],
  "properties": {
    "specversion": {
      "const": "1.0"
    },
    "id": {
      "type": "string",
      "minLength": 1,
      "maxLength": 500
    },
    "source": {
      "type": "string",
      "format": "uri-reference"
    },
    "type": {
      "enum": [
        "ocp.grant.expired",
        "ocp.grant.reactivated",
        "ocp.grant.revoked",
        "ocp.grant.superseded",
        "ocp.grant.suspended",
        "ocp.package.available",
        "ocp.package.delivered",
        "ocp.package.expired",
        "ocp.package.integrity_failed",
        "ocp.package.invalidated",
        "ocp.package.rejected",
        "ocp.proposal.accepted",
        "ocp.proposal.accepted_with_changes",
        "ocp.proposal.deferred",
        "ocp.proposal.disputed",
        "ocp.proposal.expired",
        "ocp.proposal.rejected",
        "ocp.proposal.superseded",
        "ocp.proposal.under_review",
        "ocp.proposal.withdrawn",
        "ocp.request.approved",
        "ocp.request.awaiting_decision",

```

```

        "ocp.request.denied",
        "ocp.request.expired",
        "ocp.request.partially_approved",
        "ocp.request.superseded",
        "ocp.request.withdrawn",
        "ocp.transfer.aborted",
        "ocp.transfer.delivered",
        "ocp.transfer.importing",
        "ocp.transfer.rejected",
        "ocp.transfer.validated",
        "ocp.transfer.validated_with_exceptions"
    ],
    },
    "time": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
    },
    "subject": {
        "type": "string",
        "format": "uri-reference"
    },
    "dataschema": {
        "const": "https://opencontinuity.org/schemas/0.4/bindings/http/event-notification.schema.json"
    },
    "data": {}
},
"allOf": [
    {
        "oneOf": [
            {
                "type": "object",
                "properties": {
                    "type": {
                        "enum": [
                            "ocp.grant.expired",
                            "ocp.grant.reactivated",
                            "ocp.grant.revoked",
                            "ocp.grant.superseded",
                            "ocp.grant.suspended",
                            "ocp.package.available",
                            "ocp.package.delivered",
                            "ocp.package.expired",
                            "ocp.package.invalidated",
                            "ocp.package.rejected",
                            "ocp.proposal.accepted",
                            "ocp.proposal.accepted_with_changes",
                            "ocp.proposal.deferred",
                            "ocp.proposal.disputed",
                            "ocp.proposal.expired",
                            "ocp.proposal.rejected",
                            "ocp.proposal.superseded",
                            "ocp.proposal.under_review",
                            "ocp.proposal.withdrawn",
                            "ocp.request.approved",
                            "ocp.request.awaiting_decision",
                            "ocp.request.denied",
                            "ocp.request.expired",
                            "ocp.request.partially_approved",
                            "ocp.request.superseded",
                            "ocp.request.withdrawn",
                            "ocp.transfer.aborted",
                            "ocp.transfer.delivered",
                            "ocp.transfer.importing",
                            "ocp.transfer.rejected",
                            "ocp.transfer.validated",
                            "ocp.transfer.validated_with_exceptions"
                        ]
                    }
                },
                "data": {
                    "$ref": "#/$defs/lifecycleData"
                }
            },
            {
                "type": "string",
                "format": "uri-reference"
            }
        ]
    },
    "required": [
        "type",
        "data"
    ]
}
],
{

```

```

    "type": "object",
    "properties": {
      "type": {
        "enum": [
          "ocp.package.integrity_failed"
        ]
      },
      "data": {
        "$ref": "#/$defs/notificationData"
      }
    },
    "required": [
      "type",
      "data"
    ]
  }
]
},
{
  "oneOf": [
    {
      "type": "object",
      "properties": {
        "type": {
          "const": "ocp.grant.expired"
        },
        "data": {
          "oneOf": [
            {
              "type": "object",
              "properties": {
                "from_state": {
                  "const": "active"
                },
                "to_state": {
                  "const": "expired"
                }
              },
              "required": [
                "from_state",
                "to_state"
              ]
            },
            {
              "type": "object",
              "properties": {
                "from_state": {
                  "const": "suspended"
                },
                "to_state": {
                  "const": "expired"
                }
              },
              "required": [
                "from_state",
                "to_state"
              ]
            }
          ]
        }
      },
      "required": [
        "type",
        "data"
      ]
    },
    {
      "type": "object",
      "properties": {
        "type": {
          "const": "ocp.grant.reactivated"
        },
        "data": {
          "oneOf": [
            {
              "type": "object",
              "properties": {

```

```

        "from_state": {
          "const": "suspended"
        },
        "to_state": {
          "const": "active"
        }
      },
      "required": [
        "from_state",
        "to_state"
      ]
    }
  ]
},
"required": [
  "type",
  "data"
]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.grant.revoked"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "active"
            },
            "to_state": {
              "const": "revoked"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        },
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "suspended"
            },
            "to_state": {
              "const": "revoked"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.grant.superseded"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {

```

```

        "from_state": {
          "const": "active"
        },
        "to_state": {
          "const": "superseded"
        }
      },
      "required": [
        "from_state",
        "to_state"
      ]
    },
    {
      "type": "object",
      "properties": {
        "from_state": {
          "const": "suspended"
        },
        "to_state": {
          "const": "superseded"
        }
      },
      "required": [
        "from_state",
        "to_state"
      ]
    }
  ]
},
"required": [
  "type",
  "data"
]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.grant.suspended"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "active"
            },
            "to_state": {
              "const": "suspended"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.package.available"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {

```

```

        "from_state": {
          "const": "built"
        },
        "to_state": {
          "const": "available"
        }
      },
      "required": [
        "from_state",
        "to_state"
      ]
    }
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.package.delivered"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "available"
            },
            "to_state": {
              "const": "delivered"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.package.expired"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "available"
            },
            "to_state": {
              "const": "expired"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
}

```

```

        "from_state": {
          "const": "delivered"
        },
        "to_state": {
          "const": "expired"
        }
      },
      "required": [
        "from_state",
        "to_state"
      ]
    }
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.package.integrity_failed"
    },
    "data": {
      "type": "object",
      "properties": {
        "notification_code": {
          "const": "ocp.package.integrity_failed"
        }
      },
      "required": [
        "notification_code"
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.package.invalidated"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "available"
            },
            "to_state": {
              "const": "invalidated"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        },
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "delivered"
            },
            "to_state": {
              "const": "invalidated"
            }
          },
          "required": [

```

```

        "from_state",
        "to_state"
      ]
    }
  ]
},
"required": [
  "type",
  "data"
]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.package.rejected"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "built"
            },
            "to_state": {
              "const": "rejected"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.proposal.accepted"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "under_review"
            },
            "to_state": {
              "const": "accepted"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {

```

```

    "type": {
      "const": "ocp.proposal.accepted_with_changes"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "under_review"
            },
            "to_state": {
              "const": "accepted_with_changes"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    },
    "required": [
      "type",
      "data"
    ]
  },
  {
    "type": "object",
    "properties": {
      "type": {
        "const": "ocp.proposal.deferred"
      },
      "data": {
        "oneOf": [
          {
            "type": "object",
            "properties": {
              "from_state": {
                "const": "under_review"
              },
              "to_state": {
                "const": "deferred"
              }
            },
            "required": [
              "from_state",
              "to_state"
            ]
          }
        ]
      },
      "required": [
        "type",
        "data"
      ]
    }
  },
  {
    "type": "object",
    "properties": {
      "type": {
        "const": "ocp.proposal.disputed"
      },
      "data": {
        "oneOf": [
          {
            "type": "object",
            "properties": {
              "from_state": {
                "const": "under_review"
              },
              "to_state": {
                "const": "disputed"
              }
            },
            "required": [
              "from_state",
              "to_state"
            ]
          }
        ]
      },
      "required": [
        "type",
        "data"
      ]
    }
  }
]

```

```

        "required": [
            "from_state",
            "to_state"
        ]
    }
}
},
"required": [
    "type",
    "data"
]
},
{
    "type": "object",
    "properties": {
        "type": {
            "const": "ocp.proposal.expired"
        },
        "data": {
            "oneOf": [
                {
                    "type": "object",
                    "properties": {
                        "from_state": {
                            "const": "submitted"
                        },
                        "to_state": {
                            "const": "expired"
                        }
                    }
                },
                {
                    "required": [
                        "from_state",
                        "to_state"
                    ]
                },
                {
                    "type": "object",
                    "properties": {
                        "from_state": {
                            "const": "under_review"
                        },
                        "to_state": {
                            "const": "expired"
                        }
                    }
                },
                {
                    "required": [
                        "from_state",
                        "to_state"
                    ]
                },
                {
                    "type": "object",
                    "properties": {
                        "from_state": {
                            "const": "deferred"
                        },
                        "to_state": {
                            "const": "expired"
                        }
                    }
                },
                {
                    "required": [
                        "from_state",
                        "to_state"
                    ]
                }
            ]
        }
    }
},
"required": [
    "type",
    "data"
]
},
{
    "type": "object",
    "properties": {

```

```

    "type": {
      "const": "ocp.proposal.rejected"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "under_review"
            },
            "to_state": {
              "const": "rejected"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        },
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "deferred"
            },
            "to_state": {
              "const": "rejected"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    },
    "required": [
      "type",
      "data"
    ]
  },
  {
    "type": "object",
    "properties": {
      "type": {
        "const": "ocp.proposal.superseded"
      },
      "data": {
        "oneOf": [
          {
            "type": "object",
            "properties": {
              "from_state": {
                "const": "submitted"
              },
              "to_state": {
                "const": "superseded"
              }
            },
            "required": [
              "from_state",
              "to_state"
            ]
          }
        ]
      },
      "required": [
        "type",
        "data"
      ]
    }
  },
  {
    "type": "object",
    "properties": {

```

```

    "type": {
      "const": "ocp.proposal.under_review"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "submitted"
            },
            "to_state": {
              "const": "under_review"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        },
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "deferred"
            },
            "to_state": {
              "const": "under_review"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        },
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "disputed"
            },
            "to_state": {
              "const": "under_review"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    },
    "required": [
      "type",
      "data"
    ]
  },
  {
    "type": "object",
    "properties": {
      "type": {
        "const": "ocp.proposal.withdrawn"
      },
      "data": {
        "oneOf": [
          {
            "type": "object",
            "properties": {
              "from_state": {
                "const": "submitted"
              },
              "to_state": {
                "const": "withdrawn"
              }
            },
            "required": [

```

```

        "from_state",
        "to_state"
    ]
},
{
    "type": "object",
    "properties": {
        "from_state": {
            "const": "deferred"
        },
        "to_state": {
            "const": "withdrawn"
        }
    },
    "required": [
        "from_state",
        "to_state"
    ]
}
]
},
{
    "required": [
        "type",
        "data"
    ]
},
{
    "type": "object",
    "properties": {
        "type": {
            "const": "ocp.request.approved"
        },
        "data": {
            "oneOf": [
                {
                    "type": "object",
                    "properties": {
                        "from_state": {
                            "const": "awaiting_decision"
                        },
                        "to_state": {
                            "const": "approved"
                        }
                    },
                    "required": [
                        "from_state",
                        "to_state"
                    ]
                },
                {
                    "type": "object",
                    "properties": {
                        "from_state": {
                            "const": "partially_approved"
                        },
                        "to_state": {
                            "const": "approved"
                        }
                    },
                    "required": [
                        "from_state",
                        "to_state"
                    ]
                }
            ]
        }
    },
    "required": [
        "type",
        "data"
    ]
},
{
    "type": "object",
    "properties": {
        "type": {

```

```

    "const": "ocp.request.awaiting_decision"
  },
  "data": {
    "oneOf": [
      {
        "type": "object",
        "properties": {
          "from_state": {
            "const": "submitted"
          },
          "to_state": {
            "const": "awaiting_decision"
          }
        },
        "required": [
          "from_state",
          "to_state"
        ]
      }
    ]
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.request.denied"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "awaiting_decision"
            },
            "to_state": {
              "const": "denied"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        },
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "partially_approved"
            },
            "to_state": {
              "const": "denied"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {

```

```

      "const": "ocp.request.expired"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "awaiting_decision"
            },
            "to_state": {
              "const": "expired"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        },
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "partially_approved"
            },
            "to_state": {
              "const": "expired"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.request.partially_approved"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "awaiting_decision"
            },
            "to_state": {
              "const": "partially_approved"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {

```

```

      "const": "ocp.request.superseded"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "submitted"
            },
            "to_state": {
              "const": "superseded"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        },
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "awaiting_decision"
            },
            "to_state": {
              "const": "superseded"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.request.withdrawn"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "submitted"
            },
            "to_state": {
              "const": "withdrawn"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {

```

```

    "const": "ocp.transfer.aborted"
  },
  "data": {
    "oneOf": [
      {
        "type": "object",
        "properties": {
          "from_state": {
            "const": "prepared"
          },
          "to_state": {
            "const": "aborted"
          }
        },
        "required": [
          "from_state",
          "to_state"
        ]
      },
      {
        "type": "object",
        "properties": {
          "from_state": {
            "const": "delivered"
          },
          "to_state": {
            "const": "aborted"
          }
        },
        "required": [
          "from_state",
          "to_state"
        ]
      }
    ]
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.transfer.delivered"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "prepared"
            },
            "to_state": {
              "const": "delivered"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {

```

```

    "const": "ocp.transfer.importing"
  },
  "data": {
    "oneOf": [
      {
        "type": "object",
        "properties": {
          "from_state": {
            "const": "delivered"
          },
          "to_state": {
            "const": "importing"
          }
        },
        "required": [
          "from_state",
          "to_state"
        ]
      }
    ]
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.transfer.rejected"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "importing"
            },
            "to_state": {
              "const": "rejected"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.transfer.validated"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "importing"
            },
            "to_state": {
              "const": "validated"
            }
          },
          "required": [

```

```

        "from_state",
        "to_state"
      ]
    }
  ]
},
"required": [
  "type",
  "data"
]
},
{
  "type": "object",
  "properties": {
    "type": {
      "const": "ocp.transfer.validated_with_exceptions"
    },
    "data": {
      "oneOf": [
        {
          "type": "object",
          "properties": {
            "from_state": {
              "const": "importing"
            },
            "to_state": {
              "const": "validated_with_exceptions"
            }
          },
          "required": [
            "from_state",
            "to_state"
          ]
        }
      ]
    }
  },
  "required": [
    "type",
    "data"
  ]
}
],
"additionalProperties": false
}

```

43.4 bindings/http/problem.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/bindings/http/problem.schema.json",
  "title": "OCP 0.4 RFC 9457 problem detail",
  "type": "object",
  "required": [
    "type",
    "title",
    "status",
    "code",
    "correlation_id"
  ],
  "properties": {
    "type": {
      "type": "string",
      "format": "uri-reference"
    },
    "title": {
      "type": "string",
      "minLength": 1,
      "maxLength": 200
    },
    "status": {
      "type": "integer",
      "minimum": 400,

```

```

    "maximum": 599
  },
  "detail": {
    "type": "string",
    "maxLength": 1000
  },
  "instance": {
    "type": "string",
    "format": "uri-reference"
  },
  "code": {
    "enum": [
      "OCP_ACTION_AUTHORITY_SEPARATE",
      "OCP_ADAPTER_NONDETERMINISTIC",
      "OCP_ADAPTER_SCHEMA_DRIFT",
      "OCP_ARTIFACT_INTEGRITY_FAILED",
      "OCP_ARTIFACT_PRESERVATION_OVERCLAIM",
      "OCP_AUDIENCE_MISMATCH",
      "OCP_AUTHENTICATION_REQUIRED",
      "OCP_AUTHORIZATION_REQUIRED",
      "OCP_CONFLICT",
      "OCP_CONVERSATION_GRAPH_INVALID",
      "OCP_DELEGATION_FORBIDDEN",
      "OCP_EVENT_CURSOR_EXPIRED",
      "OCP_EVENT_GAP",
      "OCP_FORBIDDEN",
      "OCP_GRANT_INACTIVE",
      "OCP_ILLEGAL_TRANSITION",
      "OCP_IMPORT_LIMIT_EXCEEDED",
      "OCP_INTEGRITY_FAILED",
      "OCP_INTERNAL_ERROR",
      "OCP_INVALID_REQUEST",
      "OCP_METHOD_NOT_ALLOWED",
      "OCP_NOT_FOUND",
      "OCP_ORGANIZATIONAL_POLICY",
      "OCP_PACKAGE_EXPIRED",
      "OCP_PRECONDITION_REQUIRED",
      "OCP_PROCESSING_USE_EXCEEDED",
      "OCP_PURPOSE_MISMATCH",
      "OCP_RATE_LIMITED",
      "OCP_SCHEMA_VALIDATION_FAILED",
      "OCP_SCOPE_EXCEEDED",
      "OCP_SEMANTIC_VALIDATION_FAILED",
      "OCP_STALE_REVISION",
      "OCP_TEMPORARILY_UNAVAILABLE",
      "OCP_THIRD_PARTY_RESTRICTED",
      "OCP_UNREPORTED_PROFILE_LOSS",
      "OCP_UNSAFE_IMPORT",
      "OCP_UNSAFE_IMPORTED_CONTENT",
      "OCP_UNSUPPORTED_CRITICAL_EXTENSION",
      "OCP_UNSUPPORTED_MEDIA_TYPE",
      "OCP_UNSUPPORTED_PROFILE"
    ]
  },
  "correlation_id": {
    "type": "string",
    "format": "uri"
  },
  "retryable": {
    "type": "boolean"
  },
  "fields": {
    "type": "array",
    "items": {
      "type": "object",
      "required": [
        "path",
        "code"
      ],
      "properties": {
        "path": {
          "type": "string",
          "pattern": "^(?:/(?:[~/]|~0|~1)*)$"
        },
        "code": {
          "type": "string",
          "pattern": "^0CP_[A-Z0-9_]+$"
        }
      }
    }
  }
}

```

```

    }
  },
  "additionalProperties": false
}
},
"allOf": [
  {
    "oneOf": [
      {
        "properties": {
          "code": {
            "enum": [
              "OCP_INVALID_REQUEST",
              "OCP_SCHEMA_VALIDATION_FAILED"
            ]
          },
          "status": {
            "const": 400
          }
        },
        "required": [
          "code",
          "status"
        ]
      },
      {
        "properties": {
          "code": {
            "enum": [
              "OCP_AUTHENTICATION_REQUIRED",
              "OCP_AUTHORIZATION_REQUIRED"
            ]
          },
          "status": {
            "const": 401
          }
        },
        "required": [
          "code",
          "status"
        ]
      },
      {
        "properties": {
          "code": {
            "enum": [
              "OCP_FORBIDDEN",
              "OCP_AUDIENCE_MISMATCH",
              "OCP_PURPOSE_MISMATCH",
              "OCP_SCOPE_EXCEEDED",
              "OCP_PROCESSING_USE_EXCEEDED",
              "OCP_DELEGATION_FORBIDDEN",
              "OCP_ACTION_AUTHORITY_SEPARATE",
              "OCP_THIRD_PARTY_RESTRICTED",
              "OCP_ORGANIZATIONAL_POLICY"
            ]
          },
          "status": {
            "const": 403
          }
        },
        "required": [
          "code",
          "status"
        ]
      },
      {
        "properties": {
          "code": {
            "const": "OCP_NOT_FOUND"
          },
          "status": {
            "const": 404
          }
        },
        "required": [

```

```

        "code",
        "status"
      ]
    },
    {
      "properties": {
        "code": {
          "const": "OCP_METHOD_NOT_ALLOWED"
        },
        "status": {
          "const": 405
        }
      },
      "required": [
        "code",
        "status"
      ]
    },
    {
      "properties": {
        "code": {
          "enum": [
            "OCP_CONFLICT",
            "OCP_ILLEGAL_TRANSITION",
            "OCP_GRANT_INACTIVE"
          ]
        },
        "status": {
          "const": 409
        }
      },
      "required": [
        "code",
        "status"
      ]
    },
    {
      "properties": {
        "code": {
          "enum": [
            "OCP_PACKAGE_EXPIRED",
            "OCP_EVENT_CURSOR_EXPIRED",
            "OCP_EVENT_GAP"
          ]
        },
        "status": {
          "const": 410
        }
      },
      "required": [
        "code",
        "status"
      ]
    },
    {
      "properties": {
        "code": {
          "const": "OCP_STALE_REVISION"
        },
        "status": {
          "const": 412
        }
      },
      "required": [
        "code",
        "status"
      ]
    },
    {
      "properties": {
        "code": {
          "const": "OCP_IMPORT_LIMIT_EXCEEDED"
        },
        "status": {
          "const": 413
        }
      },
    },

```

```

    "required": [
      "code",
      "status"
    ]
  },
  {
    "properties": {
      "code": {
        "const": "OCP_UNSUPPORTED_MEDIA_TYPE"
      },
      "status": {
        "const": 415
      }
    },
    "required": [
      "code",
      "status"
    ]
  },
  {
    "properties": {
      "code": {
        "enum": [
          "OCP_ADAPTER_NONDETERMINISTIC",
          "OCP_ADAPTER_SCHEMA_DRIFT",
          "OCP_ARTIFACT_INTEGRITY_FAILED",
          "OCP_ARTIFACT_PRESERVATION_OVERCLAIM",
          "OCP_CONVERSATION_GRAPH_INVALID",
          "OCP_INTEGRITY_FAILED",
          "OCP_SEMANTIC_VALIDATION_FAILED",
          "OCP_UNREPORTED_PROFILE_LOSS",
          "OCP_UNSAFE_IMPORT",
          "OCP_UNSAFE_IMPORTED_CONTENT",
          "OCP_UNSUPPORTED_CRITICAL_EXTENSION",
          "OCP_UNSUPPORTED_PROFILE"
        ]
      },
      "status": {
        "const": 422
      }
    },
    "required": [
      "code",
      "status"
    ]
  },
  {
    "properties": {
      "code": {
        "const": "OCP_PRECONDITION_REQUIRED"
      },
      "status": {
        "const": 428
      }
    },
    "required": [
      "code",
      "status"
    ]
  },
  {
    "properties": {
      "code": {
        "const": "OCP_RATE_LIMITED"
      },
      "status": {
        "const": 429
      }
    },
    "required": [
      "code",
      "status"
    ]
  },
  {
    "properties": {
      "code": {

```

```

        "const": "OCP_INTERNAL_ERROR"
      },
      "status": {
        "const": 500
      }
    },
    "required": [
      "code",
      "status"
    ]
  },
  {
    "properties": {
      "code": {
        "const": "OCP_TEMPORARILY_UNAVAILABLE"
      },
      "status": {
        "const": 503
      }
    },
    "required": [
      "code",
      "status"
    ]
  }
],
{
  "if": {
    "properties": {
      "code": {
        "enum": [
          "OCP_RATE_LIMITED",
          "OCP_TEMPORARILY_UNAVAILABLE"
        ]
      }
    },
    "required": [
      "code"
    ]
  },
  "then": {
    "properties": {
      "retryable": {
        "const": true
      }
    }
  },
  "else": {
    "properties": {
      "retryable": {
        "const": false
      }
    }
  }
}
],
"additionalProperties": false
}

```

43.5 bindings/http/review-decision-create.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/bindings/http/review-decision-create.schema.json",
  "title": "OCP 0.4 HTTP review-decision create input",
  "description": "Caller-supplied semantics for an immutable review decision. For an accepting outcome, resulting_item_input carries the complete reviewed result semantics and accepted scope; the provider authors result identity/revision, creation audit fields, active lifecycle, accepted review state, and the decision/result commitment atomically. reviewed_by MUST match the authenticated and authorized reviewer, and proposal_ref MUST identify the exact proposal revision selected by the proposal_id path parameter and If-Match precondition.",
  "$defs": {
    "resultingItemInput": {
      "type": "object",
      "required": ["extensions", "subjects", "space_ref", "category", "statement", "epistemic_basis",

```

```

"provenance", "sensitivity", "accepted_scope"],
  "properties": {
    "extensions": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
extensions"},
    "subjects": {"type": "array", "items": {"$ref":
"https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/subjectRef"}, "minItems": 1},
    "space_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
snapshotRef"},
    "category": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
contextCategory"},
    "instruction_treatment": {"const": "untrusted_data"},
    "statement": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
statement"},
    "epistemic_basis": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
epistemicBasis"},
    "validity": {"$ref": "https://opencontinuity.org/schemas/0.4/context-item.schema.json#/allOf/1/
properties/validity"},
    "provenance": {"type": "array", "items": {"$ref":
"https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/provenance"}, "minItems": 1,
"uniqueItems": true},
    "sensitivity": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
sensitivity"},
    "rights": {"$ref": "https://opencontinuity.org/schemas/0.4/context-item.schema.json#/allOf/1/
properties/rights"},
    "accepted_scope": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
acceptedScope"}
  },
  "allOf": [
    {
      "if": {"properties": {"category": {"const": "behavior_preference"}}, "required": ["category"]},
      "then": {"required": ["instruction_treatment"]}
    },
    {
      "additionalProperties": false
    }
  ],
  "type": "object",
  "required": ["proposal_ref", "reviewed_by", "authority_basis", "outcome"],
  "properties": {
    "proposal_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
snapshotRef"},
    "reviewed_by": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef",
"description": "MUST match the authenticated principal and that principal's current review authority."},
    "authority_basis": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
authorityBasis"},
    "outcome": {"enum": ["accepted", "accepted_with_changes", "rejected", "disputed", "deferred"]},
    "resulting_item_input": {"$ref": "#/$defs/resultingItemInput"},
    "reason": {"type": "string", "minLength": 1, "maxLength": 2000}
  },
  "allOf": [
    {
      "if": {"properties": {"outcome": {"enum": ["accepted", "accepted_with_changes"]}}, "required":
["outcome"]},
      "then": {"required": ["resulting_item_input"]},
      "else": {"not": {"required": ["resulting_item_input"]}}
    }
  ],
  "additionalProperties": false
}

```

43.6 bindings/http/transfer-manifest-create.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/bindings/http/transfer-manifest-create.schema.json",
  "title": "OCP 0.4 HTTP transfer-manifest create input",
  "description": "Client-supplied transfer intent. The provider authors the record envelope, initial
state prepared, entry transfer statuses, exceptions, completion time, and source disposition.",
  "type": "object",
  "required": ["source_provider", "destination_provider", "entries"],
  "properties": {
    "source_provider": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
actorRef"},
    "destination_provider": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
actorRef"},
    "entries": {
      "type": "array",
      "items": {

```

```

      "type": "object",
      "required": ["source_ref"],
      "properties": {"source_ref": {"$ref":
"https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"}},
      "additionalProperties": false
    },
    "minItems": 1
  },
  "additionalProperties": false
}

```

43.7 bindings/http/update-proposal-create.schema.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/bindings/http/update-proposal-create.schema.json",
  "title": "OCP 0.4 HTTP update-proposal create input",
  "description": "Client-supplied semantics for creating an update proposal. Any proposed item remains candidate/active material. The provider authenticates and binds submitted_by, then authors ocp_version, schema_uri, type, id, revision, created_at, created_by, correlation_id, status=\"submitted\", and extensions in the persisted UpdateProposal returned by the operation.",
  "type": "object",
  "required": [
    "submitted_by",
    "grant_receipt_ref",
    "grant_status_ref",
    "operation",
    "source_refs"
  ],
  "properties": {
    "submitted_by": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef", "description": "MUST match the authenticated and authorized submitting principal."},
    "grant_receipt_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"},
    "grant_status_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/liveStatusRef"},
    "operation": {"enum": ["create", "revise"]},
    "target_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"},
    "proposed_item": {
      "allOf": [
        {
          "$ref": "https://opencontinuity.org/schemas/0.4/context-item.schema.json",
          {
            "type": "object",
            "properties": {
              "review": {"type": "object", "properties": {"status": {"const": "candidate"}}},
              "lifecycle": {"type": "object", "properties": {"status": {"const": "active"}}}
            }
          }
        ]
      }
    },
    "source_refs": {"type": "array", "items": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"}, "uniqueItems": true},
    "rationale": {"type": "string", "minLength": 1, "maxLength": 2000}
  },
  "oneOf": [
    {
      "properties": {"operation": {"const": "create"}, "source_refs": {"type": "array", "minItems": 1}},
      "required": ["operation", "proposed_item", "not": {"required": ["target_ref"]}],
      {
        "properties": {"operation": {"const": "revise"}, "source_refs": {"type": "array", "minItems": 1}},
        "required": ["operation", "target_ref", "proposed_item"]
      }
    ],
    "additionalProperties": false
  ]
}
```

43.8 bindings/offline/offline-package-manifest.schema.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/bindings/offline/offline-package-manifest.schema.json",
  "title": "OCP 0.4 offline archive manifest",
  "$defs": {
    "safePath": {
      "type": "string",
      "pattern": "^(?!\\|\\.|1\\.2)(?:/|$)(?!.*(?:^|/|\\|\\.|1\\.2)(?:/|$))(?!.*(?:^|/|\\|\\.|1\\.2)(?:/|$))"
    }
  }
}
```

```
(?!.*(?:~|/)(?:[Cc][Oo][Nn]|[Pp][Uu][Xx]|[Nn][Uu][Ll]|[Cc][Oo][Rr][Mm][1-9]|[Ll][Pp][Tt][1-9])
(?:\\.\\.[^/]*?(?:/|$))[A-Za-z0-9._-]+(?:/[A-Za-z0-9._-]+)*$,
"description": "Portable relative archive path without traversal segments or Windows reserved
device-name segments; producers and consumers also reject Unicode-normalized, case-folded member
collisions."
},
"member": {
"type": "object",
"required": ["path", "media_type", "byte_size", "digest"],
"properties": {
"path": {"$ref": "#/$defs/safePath"},
"media_type": {"type": "string", "pattern": "^([A-Za-z0-9!#$%^_+]+)/([A-Za-z0-9!#$%^_+]+$)",
"byte_size": {"type": "integer", "minimum": 0},
"digest": {"type": "string", "pattern": "^sha-256:[0-9a-f]{64}$"},
"object_id": {"type": "string", "format": "uri"},
"schema_uri": {"type": "string", "format": "uri"}
},
"additionalProperties": false
}
},
"type": "object",
"required": ["manifest_version", "ocp_version", "package_type", "package_id", "created_at",
"archive_format", "content_classification", "payload_members", "security_members", "member_set_policy"],
"properties": {
"manifest_version": {"const": "1.1"},
"ocp_version": {"const": "0.4"},
"package_type": {"enum": ["context_package", "export", "transfer"]},
"package_id": {"type": "string", "format": "uri"},
"created_at": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
timestamp"},
"archive_format": {"const": "zip"},
"content_classification": {"enum": ["public", "non_public"]},
"payload_members": {"type": "array", "items": {"$ref": "#/$defs/member"}, "minItems": 1},
"security_members": {"type": "array", "items": {"$ref": "#/$defs/member"}},
"member_set_policy": {"const": "exact_declared_set"},
"protection": {
"type": "object",
"properties": {
"encryption": {
"type": "object",
"required": ["profile", "metadata_path"],
"properties": {"profile": {"type": "string", "format": "uri"}, "metadata_path": {"$ref":
"#/$defs/safePath"}},
"additionalProperties": false
},
"signature": {
"type": "object",
"required": ["profile", "signature_path", "signed_view"],
"properties": {
"profile": {"type": "string", "format": "uri"},
"signature_path": {"$ref": "#/$defs/safePath"},
"signed_view": {"const": "jcs_manifest_without_signature_descriptor_and_member"}
},
"additionalProperties": false
}
},
"additionalProperties": false
}
},
"allOf": [
{
"if": {"properties": {"content_classification": {"const": "non_public"}}}, "required":
["content_classification"]},
{
"then": {"required": ["protection"], "properties": {"protection": {"type": "object", "required":
["signature", "encryption"]}}}
},
{
"additionalProperties": false
}
]
}
```

43.9 bindings/transfer/transfer-manifest.schema.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/bindings/transfer/transfer-manifest.schema.json",
  "title": "OCP 0.4 transfer binding manifest",

```

```

    "type": "object",
    "$defs": {
      "safePath": {
        "type": "string",
        "pattern": "^(?!\\.\\{1,2}(?:/|$))(?!.*(?:^|/)\\.\\{1,2}(?:/|$))(?!.*(?:^|/)[^/]*)\\.?(?:/|$))"
      },
      "allOf": [
        {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/recordEnvelope"},
        {
          "type": "object",
          "required": ["type", "source_provider", "destination_provider", "state", "entries", "exceptions"],
          "created_at": {
            "properties": {
              "type": {"const": "transfer_manifest"},
              "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/bindings/transfer/transfer-manifest.schema.json"},
              "source_provider": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"},
              "destination_provider": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"},
              "state": {"enum": ["prepared", "delivered", "importing", "validated"]},
              "validated_with_exceptions": ["rejected", "aborted"]},
              "entries": {
                "type": "array",
                "items": {
                  "type": "object",
                  "required": ["source_ref", "export_id", "path", "transfer_status", "losses"],
                  "properties": {
                    "source_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"},
                    "export_id": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"},
                    "path": {"$ref": "#/$defs/safePath"},
                    "transfer_status": {"enum": ["included", "imported", "verified", "missing", "unsupported", "intentionally_omitted", "integrity_failed"]},
                    "destination_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"},
                    "losses": {"type": "array", "items": {"type": "object", "required": ["path", "reason"], "properties": {"path": {"type": "string", "minLength": 1}, "reason": {"enum": ["unsupported_field", "unsupported_profile", "privacy_omission", "semantic_mismatch", "unknown_extension", "other"]}}, "detail": {"type": "string", "maxLength": 1000}}, "additionalProperties": false}}},
                  "allOf": [
                    {"if": {"properties": {"transfer_status": {"enum": ["imported", "verified"]}}, "required": ["transfer_status"]}, "then": {"required": ["destination_ref"]}, "else": {"not": {"required": ["destination_ref"]}}}
                  ],
                  "additionalProperties": false
                }
              },
              "minItems": 1
            },
            "exceptions": {"type": "array", "items": {"type": "object", "required": ["code", "detail"], "properties": {"code": {"type": "string", "pattern": "^OCP_[A-Z0-9_]+$"}, "source_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"}, "detail": {"type": "string", "minLength": 1, "maxLength": 1000}}, "additionalProperties": false}},
            "source_disposition": {"enum": ["unchanged", "deletion_requested", "unknown"]},
            "completed_at": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"}
          }
        },
        {
          "if": {"properties": {"state": {"enum": ["prepared", "delivered"]}}, "required": ["state"]},
          "then": {"properties": {"entries": {"type": "array", "items": {"type": "object", "properties": {"transfer_status": {"const": ["included", "imported", "verified"]}, "required": ["transfer_status"]}}, "exceptions": {"type": "array", "maxItems": 0}}}},
          "if": {"properties": {"state": {"const": "importing"}}, "required": ["state"]}, "then": {"properties": {"entries": {"type": "array", "items": {"type": "object", "properties": {"transfer_status": {"enum": ["included", "imported", "verified"]}, "required": ["transfer_status"]}}, "exceptions": {"type": "array", "maxItems": 0}}}},
          "if": {"properties": {"state": {"const": "validated"}}, "required": ["state"]}, "then": {"properties": {"entries": {"type": "array", "items": {"type": "object", "properties": {"transfer_status": {"const": "verified"}}, "required": ["transfer_status"]}, "exceptions": {"type": "array", "maxItems": 0}}}},
          "if": {"properties": {"state": {"const": "validated_with_exceptions"}}, "required": ["state"]}, "then": {"properties": {"exceptions": {"type": "array", "minItems": 1, "items": {"type": "object",

```

```

    "required": ["source_ref"]}}}},
    {"if": {"properties": {"state": {"enum": ["validated", "validated_with_exceptions", "rejected",
"aborted"]}}, "required": ["state"]}, "then": {"required": ["completed_at"]}, "else": {"not":
{"required": ["completed_at"]}}}
  ]
},
"unevaluatedProperties": false
}

```

43.10 conformance/schemas/provider-adapter-fixture.schema.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/conformance/provider-adapter-fixture.schema.json",
  "title": "OCP provider adapter conformance fixture",
  "type": "object",
  "$defs": {
    "inputPath": {"type": "string", "pattern": "^(?!/)(?!.*(?:^|/)\\\\.\\\\.(?:/|$))[A-Za-z0-9._/-]+$"},
    "adapterVersion": {"type": "string", "pattern": "^[a-z0-9][a-z0-9._-]*/[0-9]+\\\\.[0-9]+\\\\.[0-9]+(?:[-+][A-Za-z0-9.-]+)?}$"}
  },
  "required": [
    "fixture_version",
    "fixture_id",
    "source_provider",
    "source_export_schema",
    "adapter_version",
    "adapter_context",
    "input_records",
    "expected_outputs",
    "expected_omissions",
    "expected_warnings"
  ],
  "properties": {
    "fixture_version": {"const": "1.2"},
    "fixture_id": {"type": "string", "format": "uri"},
    "description": {"type": "string", "minLength": 1, "maxLength": 2000},
    "source_provider": {
      "type": "object",
      "required": ["id", "name"],
      "properties": {
        "id": {"type": "string", "format": "uri"},
        "name": {"type": "string", "minLength": 1, "maxLength": 200}
      }
    },
    "additionalProperties": false
  },
  "source_export_schema": {"type": "string", "minLength": 1, "maxLength": 200},
  "adapter_version": {"$ref": "#/$defs/adapterVersion"},
  "adapter_context": {
    "type": "object",
    "description": "Harness-supplied immutable source and runtime context. Determinism is evaluated over the exact input bytes, this context, and adapter_version.",
    "required": ["source_ref", "runtime_source"],
    "properties": {
      "source_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"},
      "runtime_source": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/runtimeSource"}
    },
    "additionalProperties": false
  },
  "input_records": {
    "type": "array",
    "items": {
      "type": "object",
      "required": ["path", "media_type", "digest", "digest_coverage", "raw_content_base64"],
      "properties": {
        "path": {"$ref": "#/$defs/inputPath"},
        "media_type": {"type": "string", "pattern": "^[A-Za-z0-9!#$%^_.+-]/[A-Za-z0-9!#$%^_.+-]+$"},
        "digest": {"type": "string", "pattern": "^sha-256:[0-9a-f]{64}$"},
        "digest_coverage": {"const": "raw_decoded_bytes"},
        "raw_content_base64": {"type": "string", "contentEncoding": "base64"}
      }
    },
    "additionalProperties": false
  }
}
```

```

    "minItems": 1
  },
  "expected_outputs": {
    "type": "array",
    "items": {
      "type": "object",
      "required": ["source_locator", "ocp_type", "projection", "expected_projection",
"semantic_digest"],
      "properties": {
        "source_locator": {"$ref": "#/$defs/inputPath"},
        "ocp_type": {"const": "context_item"},
        "projection": {"const": "ocp-0.4-adapter-candidate-projection-v1"},
        "expected_projection": {
          "type": "object",
          "required": ["source_locator", "ocp_type", "category", "statement", "epistemic_basis",
"review_status", "source_ref", "runtime_source", "adapter_version", "instruction_treatment"],
          "properties": {
            "source_locator": {"$ref": "#/$defs/inputPath"},
            "ocp_type": {"const": "context_item"},
            "category": {"enum": ["statement", "behavior_preference", "goal", "decision_note",
"note"]},
            "statement": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
statement"},
            "epistemic_basis": {"enum": ["subject_asserted", "subject_confirmed", "provider_observed",
"provider_inferred", "provider_interpreted", "third_party_asserted", "credential_derived",
"imported_unclassified"]},
            "review_status": {"const": "candidate"},
            "source_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
snapshotRef"},
            "runtime_source": {"$ref":
"https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/runtimeSource"},
            "adapter_version": {"$ref": "#/$defs/adapterVersion"},
            "instruction_treatment": {"const": "untrusted_data"}
          },
          "additionalProperties": false
        },
        "semantic_digest": {"type": "string", "pattern": "^sha-256:[0-9a-f]{64}$"}
      },
      "additionalProperties": false
    }
  },
  "expected_omissions": {
    "type": "array",
    "items": {
      "type": "object",
      "required": ["source_locator", "reason"],
      "properties": {
        "source_locator": {"$ref": "#/$defs/inputPath"},
        "reason": {"enum": ["security_data", "irrelevant", "third_party_restricted",
"organizational_policy", "unsupported", "user_excluded"]}
      },
      "additionalProperties": false
    }
  },
  "expected_warnings": {
    "type": "array",
    "items": {"type": "string", "pattern": "^OCP_[A-Z0-9_]+$"},
    "uniqueItems": true
  },
  "additionalProperties": false
}

```

43.11 profiles/continuity-observation.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/profiles/continuity-observation-0.1-draft.schema.json",
  "title": "OCP Continuity Observation Profile 0.1-draft",
  "description": "Observe-only continuity evidence. It never grants authority, blocks work, or mutates
accepted context. This 0.1-draft schema provides structural validation only and does not establish
observation accuracy, runtime behavior, safety, privacy, or profile conformance.",
  "type": "object",
  "allOf": [
    {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/recordEnvelope"},
    {

```

```

    "type": "object",
    "required": ["type", "profile_version", "target_ref", "condition", "result", "method",
"basis_refs", "observed_at", "runtime_source", "freshness", "change", "instruction_treatment",
"external_action_authority", "policy_effect"],
    "properties": {
      "type": {"const": "continuity_observation"},
      "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/profiles/continuity-
observation-0.1-draft.schema.json"},
      "profile_version": {"const": "0.1-draft"},
      "target_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
snapshotRef"},
      "condition": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"},
      "result": {"enum": ["satisfied", "not_satisfied", "indeterminate", "not_observed"]},
      "method": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"},
      "basis_refs": {"type": "array", "items": {"$ref":
"https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"}, "minItems": 1,
"uniqueItems": true},
      "observed_at": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
timestamp"},
      "evaluated_at": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
timestamp"},
      "runtime_source": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
runtimeSource"},
      "freshness": {"type": "object", "required": ["status", "as_of"], "properties": {"status":
{"enum": ["current", "stale", "unknown"]}, "as_of": {"$ref":
"https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"}, "expires_at": {"$ref":
"https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"}}, "additionalProperties":
false},
      "previous_observation_ref": {"$ref":
"https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"},
      "change": {"enum": ["initial", "unchanged", "degraded", "restored", "indeterminate"]},
      "limitations": {"type": "array", "items": {"type": "string", "minLength": 1, "maxLength": 500}},
      "instruction_treatment": {"const": "untrusted_data"},
      "external_action_authority": {"const": "none"},
      "policy_effect": {"const": "none"}
    }
  },
  "unevaluatedProperties": false
}

```

43.12 profiles/conversation.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/profiles/conversation.schema.json",
  "title": "OCP Conversation Profile conversation compatibility entry point",
  "description": "Compatibility entry point for the canonical conversation schema.",
  "$ref": "https://opencontinuity.org/schemas/0.4/conversation.schema.json"
}

```

43.13 profiles/erasure-status-record.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/profiles/erasure-status-record.schema.json",
  "title": "OCP Erasure Status Profile record",
  "description": "Draft Erasure Status Profile schema surface. It records scoped reports and assurance
levels but does not prove deletion; this release does not include the complete fixtures, behavioral
assertions, or implementation evidence needed for a profile-conformance claim.",
  "type": "object",
  "allOf": [
    {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/recordEnvelope"},
    {
      "type": "object",
      "required": ["type", "target_provider", "request_ref", "scope_description", "status",
"reported_by", "reported_at", "verification_level"],
      "properties": {
        "type": {"const": "erasure_status_record"},
        "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/profiles/erasure-status-
record.schema.json"},
        "target_provider": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
actorRef"},
        "request_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
snapshotRef"},

```

```

    "scope_description": {"type": "string", "minLength": 1, "maxLength": 2000},
    "status": {"enum": ["requested", "acknowledged", "reported_complete", "partially_complete",
"rejected", "legal_retention_exception", "independently_verified"]},
    "reported_by": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
actorRef"},
    "reported_at": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
timestamp"},
    "verification_level": {"enum": ["none", "provider_assertion", "cryptographic_erasure_evidence",
"independent_assessment"]},
    "verification_evidence_ref": {"$ref":
"https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"},
    "cryptographic_erasure_scope": {
      "type": "object",
      "required": [
        "key_scope",
        "ciphertext_scope",
        "wrapping_and_backup_key_scope",
        "storage_scope",
        "destruction_method",
        "destroyed_by",
        "destroyed_at",
        "replicas_and_backups_scope",
        "verification_method",
        "exclusions"
      ],
      "properties": {
        "key_scope": {"type": "string", "minLength": 1, "maxLength": 2000},
        "ciphertext_scope": {"type": "string", "minLength": 1, "maxLength": 2000},
        "wrapping_and_backup_key_scope": {"type": "string", "minLength": 1, "maxLength": 2000},
        "storage_scope": {"type": "string", "minLength": 1, "maxLength": 2000},
        "destruction_method": {"$ref":
"https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"},
        "destroyed_by": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
actorRef"},
        "destroyed_at": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
timestamp"},
        "replicas_and_backups_scope": {"type": "string", "minLength": 1, "maxLength": 2000},
        "verification_method": {"$ref":
"https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"},
        "exclusions": {"type": "string", "minLength": 1, "maxLength": 2000}
      },
      "additionalProperties": false
    },
    "independent_assessment": {
      "type": "object",
      "required": ["evaluator", "relationship", "method", "scope", "assessed_at", "limitations"],
      "properties": {
        "evaluator": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
actorRef"},
        "relationship": {"const": "independent_third_party"},
        "method": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"},
        "scope": {"type": "string", "minLength": 1, "maxLength": 2000},
        "assessed_at": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/
timestamp"},
        "limitations": {"type": "string", "minLength": 1, "maxLength": 2000}
      },
      "additionalProperties": false
    },
    "limitations": {"type": "string", "minLength": 1, "maxLength": 2000}
  },
  "allOf": [
    {
      "if": {"properties": {"status": {"const": "independently_verified"}}, "required": ["status"]},
      "then": {
        "properties": {"verification_level": {"const": "independent_assessment"}},
        "required": ["verification_evidence_ref", "independent_assessment"]
      }
    },
    {
      "if": {
        "properties": {"verification_level": {"const": "cryptographic_erasure_evidence"}},
        "required": ["verification_level"]
      },
      "then": {"required": ["verification_evidence_ref", "cryptographic_erasure_scope",
"limitations"]}
    }
  ],
  {

```

```

        "if": {
          "properties": {"verification_level": {"const": "independent_assessment"}},
          "required": ["verification_level"]
        },
        "then": {"required": ["verification_evidence_ref", "independent_assessment"]}
      }
    ]
  },
  "unevaluatedProperties": false
}

```

43.14 profiles/human-interaction-boundary.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/profiles/human-interaction-boundary-0.1-draft.schema.json",
  "title": "OCP Human Interaction Boundary Profile 0.1-draft",
  "description": "Explicit human interaction signals. It prohibits covert behavioral or biometric inference and carries no action authority. This 0.1-draft schema provides structural validation only and does not establish runtime behavior, accessibility, safety, privacy, or profile conformance.",
  "type": "object",
  "allOf": [
    {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/recordEnvelope",
      {
        "type": "object",
        "required": ["type", "profile_version", "declared_by", "signal", "scope", "purpose", "declared_at", "expires_at", "capture_basis", "instruction_treatment", "external_action_authority", "acceptance_effect"],
        "properties": {
          "type": {"const": "human_interaction_boundary"},
          "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/profiles/human-interaction-boundary-0.1-draft.schema.json"},
          "profile_version": {"const": "0.1-draft"},
          "declared_by": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"},
          "signal": {"enum": ["pause_requested", "checkpoint_requested", "slower_pacing_requested", "clarification_requested", "resume_requested"]},
          "scope": {"type": "object", "required": ["kind", "ref"], "properties": {"kind": {"enum": ["interaction", "session", "task", "package"]}, "ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"}}},
          "additionalProperties": false,
          "checkpoint_ref": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"},
          "purpose": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/purpose"},
          "declared_at": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"},
          "expires_at": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"},
          "capture_basis": {"const": "explicit_user_control"},
          "runtime_source": {"$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/runtimeSource"},
          "instruction_treatment": {"const": "untrusted_data"},
          "external_action_authority": {"const": "none"},
          "acceptance_effect": {"const": "none"}
        },
        "allOf": [
          {
            "if": {"properties": {"signal": {"const": "checkpoint_requested"}}, "required": ["signal"]}
          }
        ]
      }
    ],
    "unevaluatedProperties": false
  ]
}

```

43.15 profiles/lineage-event.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/profiles/lineage-event.schema.json",
  "title": "OCP Artifact Profile lineage event compatibility entry point",
  "description": "Compatibility entry point for the canonical lineage-event schema.",
  "$ref": "https://opencontinuity.org/schemas/0.4/lineage-event.schema.json"
}

```

43.16 profiles/message.schema.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/profiles/message.schema.json",
  "title": "OCP Conversation Profile message compatibility entry point",
  "description": "Compatibility entry point for the canonical message schema.",
  "$ref": "https://opencontinuity.org/schemas/0.4/message.schema.json"
}
```

43.17 schemas/artifact-representation.schema.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/artifact-representation.schema.json",
  "title": "OCP Artifact Profile representation",
  "description": "A specific retrievable or referenced representation of a logical artifact.",
  "type": "object",
  "required": [
    "representation_id",
    "relation",
    "media_type",
    "availability",
    "independently_retrievable",
    "source_dependency"
  ],
  "properties": {
    "representation_id": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
    },
    "relation": {
      "enum": [
        "original",
        "lossless_copy",
        "transcode",
        "preview",
        "thumbnail",
        "transcript",
        "caption",
        "translation",
        "accessible_alternative",
        "extracted_text",
        "derived_metadata",
        "audio_only",
        "other"
      ]
    },
    "media_type": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/mediaType"
    },
    "filename": {
      "type": "string",
      "minLength": 1,
      "maxLength": 1024
    },
    "byte_size": {
      "type": "integer",
      "minimum": 0
    },
    "duration_ms": {
      "type": "integer",
      "minimum": 0,
      "maximum": 315576000000
    },
    "dimensions": {
      "type": "object",
      "required": [
        "width",
        "height"
      ]
    },
    "properties": {
      "width": {
        "type": "integer",
        "minimum": 1,
        "maximum": 1000000
      }
    }
  }
}
```

```

    "height": {
      "type": "integer",
      "minimum": 1,
      "maximum": 1000000
    },
    "additionalProperties": false
  },
  "language": {
    "type": "string",
    "pattern": "^[A-Za-z]{2,3}(?:-[A-Za-z0-9]{2,8})*$"
  },
  "content_ref": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
  },
  "digest": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/digest"
  },
  "digest_algorithm": {
    "const": "sha-256"
  },
  "verified_at": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
  },
  "verification_method": {
    "enum": [
      "bytes_hashed_on_acquisition",
      "bytes_hashed_in_managed_store",
      "bytes_hashed_after_sync",
      "not_verified"
    ]
  },
  "availability": {
    "enum": [
      "available",
      "unavailable",
      "lost",
      "quarantined",
      "withheld",
      "unknown"
    ]
  },
  "independently_retrievable": {
    "type": "boolean"
  },
  "encryption_metadata_ref": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
  },
  "source_dependency": {
    "type": "boolean"
  }
},
"allOf": [
  {
    "if": {
      "properties": {
        "independently_retrievable": {
          "const": true
        }
      }
    },
    "required": [
      "independently_retrievable"
    ]
  },
  {
    "then": {
      "required": [
        "content_ref",
        "digest",
        "digest_algorithm",
        "byte_size",
        "verified_at",
        "verification_method"
      ],
      "properties": {
        "source_dependency": {
          "const": false
        }
      }
    }
  }
]

```

```

        "availability": {
          "const": "available"
        },
        "verification_method": {
          "enum": [
            "bytes_hashed_on_acquisition",
            "bytes_hashed_in_managed_store",
            "bytes_hashed_after_sync"
          ]
        }
      }
    },
    {
      "if": {
        "properties": {
          "relation": {
            "const": "original"
          }
        },
        "required": [
          "relation"
        ]
      },
      "then": {}
    }
  ],
  "additionalProperties": false
}

```

43.18 schemas/artifact.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/artifact.schema.json",
  "title": "OCP Artifact Profile logical artifact",
  "description": "Storage-neutral logical artifact with explicit origin, custody, representations, rights, accessibility, lineage, and disclosure views.",
  "type": "object",
  "allOf": [
    {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/recordEnvelope"
    },
    {
      "type": "object",
      "required": [
        "type",
        "profile_id",
        "profile_version",
        "compatible_core",
        "subjects",
        "space_ref",
        "semantic_type",
        "origin",
        "custody",
        "representations",
        "rights",
        "accessibility",
        "access",
        "sensitivity",
        "lifecycle"
      ],
      "properties": {
        "type": {
          "const": "artifact"
        },
        "schema_uri": {
          "const": "https://opencontinuity.org/schemas/0.4/artifact.schema.json"
        },
        "profile_id": {
          "const": "https://opencontinuity.org/profiles/artifact"
        },
        "profile_version": {
          "const": "0.4"
        },
        "compatible_core": {

```

```

    "const": ">=0.4 <0.5"
  },
  "subjects": {
    "type": "array",
    "items": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/subjectRef"
    },
    "minItems": 1,
    "uniqueItems": true
  },
  "space_ref": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
  },
  "semantic_type": {
    "enum": [
      "document",
      "spreadsheet",
      "presentation",
      "image",
      "audio",
      "video",
      "code",
      "dataset",
      "notebook",
      "prompt",
      "model_configuration",
      "archive",
      "multimodal_work",
      "other"
    ]
  },
  "origin": {
    "type": "object",
    "required": [
      "source_authority",
      "runtime_source",
      "acquired_at"
    ],
    "properties": {
      "source_authority": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
      },
      "runtime_source": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/runtimeSource"
      },
      "provider_object_id": {
        "type": "string",
        "minLength": 1,
        "maxLength": 500
      },
      "source_uri": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
      },
      "created_at": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
      },
      "acquired_at": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
      }
    }
  },
  "additionalProperties": false
},
"custody": {
  "type": "object",
  "required": [
    "mode",
    "holder",
    "source_dependency"
  ],
  "properties": {
    "mode": {
      "enum": [
        "reference_only",
        "derivative_only",
        "managed_copy",
        "synchronized_copy"
      ]
    }
  }
}

```

```

    },
    "holder": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
    },
    "source_dependency": {
      "type": "boolean"
    },
    "synchronization_status": {
      "enum": [
        "not_applicable",
        "healthy",
        "degraded",
        "failed"
      ]
    }
  },
  "additionalProperties": false
},
"representations": {
  "type": "array",
  "minItems": 1,
  "items": {
    "$ref": "https://opencontinuity.org/schemas/0.4/artifact-representation.schema.json"
  }
},
"rights": {
  "type": "array",
  "minItems": 1,
  "items": {
    "type": "object",
    "required": [
      "status",
      "asserted_by",
      "basis",
      "asserted_at",
      "scope",
      "limitations"
    ]
  }
},
"properties": {
  "status": {
    "enum": [
      "unknown",
      "source_declared",
      "user_asserted",
      "licensed",
      "restricted",
      "redistribution_prohibited",
      "copy_prohibited",
      "expired"
    ]
  },
  "asserted_by": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
  },
  "source_ref": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
  },
  "basis": {
    "type": "string",
    "minLength": 1,
    "maxLength": 1000
  },
  "asserted_at": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
  },
  "scope": {
    "type": "string",
    "minLength": 1,
    "maxLength": 1000
  },
  "limitations": {
    "type": "string",
    "minLength": 1,
    "maxLength": 2000
  },
  "valid_until": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
  }
}

```

```

    },
    "additionalProperties": false
  },
  "accessibility": {
    "type": "array",
    "items": {
      "type": "object",
      "required": [
        "relation",
        "representation_ref"
      ],
      "properties": {
        "relation": {
          "enum": [
            "captions",
            "transcript",
            "audio_description",
            "alt_text",
            "accessible_document",
            "translated_caption"
          ]
        },
        "representation_ref": {
          "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
        },
        "language": {
          "type": "string",
          "pattern": "^[A-Za-z]{2,3}(:-[A-Za-z0-9]{2,8})*$"
        }
      }
    },
    "additionalProperties": false
  },
  "lineage_refs": {
    "type": "array",
    "items": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
    },
    "uniqueItems": true
  },
  "access": {
    "type": "object",
    "required": [
      "metadata",
      "content"
    ],
    "properties": {
      "metadata": {
        "enum": [
          "private",
          "grant_required",
          "public"
        ]
      },
      "content": {
        "enum": [
          "private",
          "grant_required",
          "public"
        ]
      }
    }
  },
  "additionalProperties": false
},
"sensitivity": {
  "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/sensitivity"
},
"lifecycle": {
  "type": "object",
  "required": [
    "status"
  ],
  "properties": {
    "status": {
      "enum": [

```

```

        "active",
        "unavailable",
        "lost",
        "synchronization_degraded",
        "withdrawn"
    ]
},
"changed_at": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
},
"additionalProperties": false
},
"extensions": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/extensions"
}
},
"allOf": [
    {
        "if": {
            "properties": {
                "custody": {
                    "type": "object",
                    "properties": {
                        "mode": {
                            "const": "reference_only"
                        }
                    }
                },
                "required": [
                    "mode"
                ]
            }
        },
        "then": {
            "properties": {
                "custody": {
                    "type": "object",
                    "properties": {
                        "source_dependency": {
                            "const": true
                        }
                    }
                }
            }
        }
    },
    {
        "if": {
            "properties": {
                "custody": {
                    "type": "object",
                    "properties": {
                        "mode": {
                            "enum": [
                                "managed_copy",
                                "synchronized_copy"
                            ]
                        }
                    }
                },
                "required": [
                    "mode"
                ]
            }
        },
        "then": {
            "properties": {
                "custody": {
                    "type": "object",
                    "properties": {
                        "source_dependency": {
                            "const": false
                        }
                    }
                }
            }
        }
    }
],
"representations": {

```

```

        "type": "array",
        "contains": {
          "type": "object",
          "properties": {
            "independently_retrievable": {
              "const": true
            }
          },
          "required": [
            "independently_retrievable"
          ]
        },
        "minContains": 1
      }
    }
  },
  {
    "if": {
      "properties": {
        "custody": {
          "type": "object",
          "properties": {
            "mode": {
              "const": "derivative_only"
            }
          },
          "required": [
            "mode"
          ]
        }
      }
    },
    "then": {
      "properties": {
        "representations": {
          "type": "array",
          "contains": {
            "type": "object",
            "properties": {
              "relation": {
                "not": {
                  "const": "original"
                }
              },
              "independently_retrievable": {
                "const": true
              }
            },
            "required": [
              "relation",
              "independently_retrievable"
            ]
          },
          "minContains": 1
        }
      }
    }
  }
],
"unevaluatedProperties": false
}

```

43.19 schemas/conversation-content-part.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/conversation-content-part.schema.json",
  "title": "OCP Conversation Profile content part",
  "description": "Closed typed union for inert historical conversation content. Artifact bytes remain in the Artifact Profile.",
  "$defs": {
    "base": {
      "type": "object",

```

```

    "required": [
      "part_id",
      "type",
      "instruction_treatment"
    ],
    "properties": {
      "part_id": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
      },
      "type": {
        "type": "string"
      },
      "instruction_treatment": {
        "const": "untrusted_data"
      },
      "source_locator_ref": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
      }
    }
  },
  "oneOf": [
    {
      "allOf": [
        {
          "$ref": "#/$defs/base"
        },
        {
          "type": "object",
          "required": [
            "type",
            "text"
          ],
          "properties": {
            "type": {
              "enum": [
                "text",
                "markdown"
              ]
            },
            "text": {
              "type": "string",
              "maxLength": 2000000
            },
            "language": {
              "type": "string",
              "pattern": "^[A-Za-z]{2,3}(?:-[A-Za-z0-9]{2,8})*$"
            }
          }
        }
      ]
    },
    {
      "unevaluatedProperties": false,
      "type": "object",
      {
        "allOf": [
          {
            "$ref": "#/$defs/base"
          },
          {
            "type": "object",
            "required": [
              "type",
              "code"
            ],
            "properties": {
              "type": {
                "const": "code"
              },
              "code": {
                "type": "string",
                "maxLength": 2000000
              },
              "language": {
                "type": "string",
                "minLength": 1,
                "maxLength": 100
              }
            }
          }
        ]
      }
    }
  ]
}

```

```

        },
        "filename": {
          "type": "string",
          "minLength": 1,
          "maxLength": 500
        }
      }
    ],
    "unevaluatedProperties": false,
    "type": "object"
  },
  {
    "allOf": [
      {
        "$ref": "#/$defs/base"
      },
      {
        "type": "object",
        "required": [
          "type",
          "schema_uri",
          "value"
        ],
        "properties": {
          "type": {
            "const": "structured_data"
          },
          "schema_uri": {
            "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
          },
          "value": {
            "type": [
              "object",
              "array"
            ]
          }
        }
      }
    ]
  },
  "unevaluatedProperties": false,
  "type": "object"
},
{
  "allOf": [
    {
      "$ref": "#/$defs/base"
    },
    {
      "type": "object",
      "required": [
        "type",
        "artifact_ref",
        "relationship"
      ],
      "properties": {
        "type": {
          "const": "artifact_ref"
        },
        "artifact_ref": {
          "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
        },
        "relationship": {
          "enum": [
            "attachment",
            "inline_media",
            "generated_output",
            "citation_source",
            "tool_input",
            "tool_output",
            "other"
          ]
        },
        "semantic_type_hint": {
          "enum": [
            "image",
            "audio",

```

```

        "video",
        "document",
        "code",
        "dataset",
        "archive",
        "other"
      ]
    },
    "content_disclosed": {
      "type": "boolean"
    }
  }
},
"unevaluatedProperties": false,
"type": "object"
},
{
  "allOf": [
    {
      "$ref": "#/$defs/base"
    },
    {
      "type": "object",
      "required": [
        "type",
        "citation_ref"
      ],
      "properties": {
        "type": {
          "const": "citation_ref"
        },
        "citation_ref": {
          "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
        },
        "quoted_text": {
          "type": "string",
          "maxLength": 10000
        },
        "selector": {
          "type": "string",
          "maxLength": 1000
        }
      }
    }
  ],
  "unevaluatedProperties": false,
  "type": "object"
},
{
  "allOf": [
    {
      "$ref": "#/$defs/base"
    },
    {
      "type": "object",
      "required": [
        "type",
        "interaction_id",
        "tool_name",
        "historical_status"
      ],
      "properties": {
        "type": {
          "enum": [
            "tool_request_source",
            "tool_result_source"
          ]
        },
        "interaction_id": {
          "type": "string",
          "minLength": 1,
          "maxLength": 500
        },
        "tool_name": {
          "type": "string",
          "minLength": 1,

```

```

        "maxLength": 500
      },
      "historical_status": {
        "enum": [
          "requested",
          "completed",
          "failed",
          "cancelled",
          "unknown"
        ]
      },
      "payload": {},
      "current_tool_authority": {
        "const": "none"
      }
    }
  },
  "unevaluatedProperties": false,
  "type": "object"
},
{
  "allOf": [
    {
      "$ref": "#/$defs/base"
    },
    {
      "type": "object",
      "required": [
        "type",
        "reasoning_kind"
      ],
      "properties": {
        "type": {
          "const": "reasoning_source"
        },
        "reasoning_kind": {
          "enum": [
            "provider_exposed_reasoning",
            "reasoning_summary",
            "analysis_placeholder",
            "unsupported_provider_reasoning"
          ]
        },
        "text": {
          "type": "string",
          "maxLength": 200000
        },
        "source_exported": {
          "const": true
        }
      }
    }
  ],
  "unevaluatedProperties": false,
  "type": "object"
},
{
  "allOf": [
    {
      "$ref": "#/$defs/base"
    },
    {
      "type": "object",
      "required": [
        "type",
        "redaction"
      ],
      "properties": {
        "type": {
          "const": "redacted"
        },
        "redaction": {
          "type": "object",
          "required": [
            "state"
          ],

```

```

    "properties": {
      "state": {
        "enum": [
          "none",
          "source_redacted",
          "export_policy_redacted",
          "recipient_scope_redacted",
          "unsupported",
          "unavailable"
        ]
      },
      "reason_class": {
        "enum": [
          "privacy",
          "security",
          "legal",
          "rights",
          "source_policy",
          "recipient_scope",
          "unsupported_semantic",
          "source_unavailable",
          "other"
        ]
      },
      "detail_code": {
        "type": "string",
        "pattern": "^[A-Z][A-Z0-9_]{2,95}$"
      }
    },
    "additionalProperties": false
  },
  "original_type": {
    "type": "string",
    "minLength": 1,
    "maxLength": 100
  }
}
],
"unevaluatedProperties": false,
"type": "object"
},
{
  "allOf": [
    {
      "$ref": "#/$defs/base"
    },
    {
      "type": "object",
      "required": [
        "type",
        "source_semantic",
        "loss_code"
      ],
      "properties": {
        "type": {
          "const": "unsupported"
        },
        "source_semantic": {
          "type": "string",
          "minLength": 1,
          "maxLength": 500
        },
        "loss_code": {
          "type": "string",
          "pattern": "^CONV_[A-Z0-9_]+$"
        },
        "opaque_value": {}
      }
    }
  ],
  "unevaluatedProperties": false,
  "type": "object"
},
{
  "allOf": [
    {

```

```

    "$ref": "#/$defs/base"
  },
  {
    "type": "object",
    "required": [
      "type",
      "provider_semantic",
      "opaque_value",
      "executable"
    ],
    "properties": {
      "type": {
        "const": "provider_specific"
      },
      "provider_semantic": {
        "type": "string",
        "minLength": 1,
        "maxLength": 500
      },
      "opaque_value": {},
      "executable": {
        "const": false
      }
    }
  }
],
"unevaluatedProperties": false,
"type": "object"
}
]
}

```

43.20 schemas/conversation.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/conversation.schema.json",
  "title": "OCP Conversation Profile conversation",
  "description": "Conversation Profile 0.4 graph container. Array order never substitutes for graph edges.",
  "type": "object",
  "allOf": [
    {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/recordEnvelope"
    },
    {
      "type": "object",
      "required": [
        "type",
        "profile_id",
        "profile_version",
        "compatible_core",
        "subjects",
        "space_ref",
        "source_authority",
        "runtime_source",
        "representation_created_at",
        "root_message_refs",
        "message_refs",
        "selected_leaf_refs",
        "graph_state",
        "sensitivity",
        "lifecycle"
      ],
      "properties": {
        "type": {
          "const": "conversation"
        },
        "schema_uri": {
          "const": "https://opencontinuity.org/schemas/0.4/conversation.schema.json"
        },
        "profile_id": {
          "const": "https://opencontinuity.org/profiles/conversation"
        },
        "profile_version": {
          "const": "0.4"
        }
      }
    }
  ]
}

```

```

},
"compatible_core": {
  "const": ">=0.4 <0.5"
},
"subjects": {
  "type": "array",
  "items": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/subjectRef"
  },
  "minItems": 1,
  "uniqueItems": true
},
"space_ref": {
  "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
},
"source_authority": {
  "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
},
"runtime_source": {
  "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/runtimeSource"
},
"provider_conversation_id": {
  "type": "string",
  "minLength": 1,
  "maxLength": 500
},
"title": {
  "type": "string",
  "maxLength": 500
},
"source_created_at": {
  "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
},
"source_updated_at": {
  "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
},
"imported_at": {
  "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
},
"representation_created_at": {
  "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
},
"root_message_refs": {
  "type": "array",
  "items": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
  },
  "minItems": 1,
  "uniqueItems": true
},
"message_refs": {
  "type": "array",
  "items": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
  },
  "minItems": 1,
  "uniqueItems": true
},
"selected_leaf_refs": {
  "type": "array",
  "items": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
  },
  "uniqueItems": true
},
"unresolved_parent_refs": {
  "type": "array",
  "items": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
  },
  "uniqueItems": true
},
"graph_state": {
  "enum": [
    "complete",
    "partial",
    "source_incomplete"
  ]
}

```

```

    ],
    "branches": {
      "type": "array",
      "items": {
        "type": "object",
        "required": [
          "branch_id",
          "leaf_ref",
          "state"
        ],
        "properties": {
          "branch_id": {
            "type": "string",
            "minLength": 1,
            "maxLength": 200
          },
          "leaf_ref": {
            "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
          },
          "state": {
            "enum": [
              "selected",
              "current",
              "alternate",
              "abandoned",
              "provider_defined"
            ]
          },
          "label": {
            "type": "string",
            "maxLength": 500
          }
        },
        "additionalProperties": false
      }
    },
    "participants": {
      "type": "array",
      "items": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
      },
      "uniqueItems": true
    },
    "ordering_semantics": {
      "const": "parent_edges_then_optional_sibling_order"
    },
    "integrity": {
      "type": "object",
      "required": [
        "algorithm",
        "canonicalization",
        "semantic_graph_digest"
      ],
      "properties": {
        "algorithm": {
          "const": "sha-256"
        },
        "canonicalization": {
          "const": "RFC8785"
        },
        "semantic_graph_digest": {
          "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/digest"
        }
      },
      "additionalProperties": false
    },
    "provider_extensions": {
      "type": "object",
      "additionalProperties": true
    },
    "sensitivity": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/sensitivity"
    },
    "lifecycle": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/recordLifecycle"
    }
  }

```

```

    }
  },
  "unevaluatedProperties": false
}

```

43.21 schemas/core/client-declaration.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/client-declaration.schema.json",
  "title": "OCP client declaration",
  "type": "object",
  "allOf": [
    { "$ref": "common.schema.json#/$defs/recordEnvelope" },
    {
      "type": "object",
      "required": [
        "type",
        "client",
        "privacy_policy_uri",
        "security_contact",
        "processing_recipients",
        "retention_supported",
        "processing_uses_supported",
        "offline_use_supported",
        "proposal_rights_supported",
        "delegation_supported",
        "deletion_support",
        "declaration_status",
        "valid_until"
      ],
      "properties": {
        "type": { "const": "client_declaration" },
        "schema_uri": { "const": "https://opencontinuity.org/schemas/0.4/client-declaration.schema.json" },
        "client": { "$ref": "common.schema.json#/$defs/actorRef" },
        "privacy_policy_uri": { "$ref": "common.schema.json#/$defs/uri" },
        "terms_uri": { "$ref": "common.schema.json#/$defs/uri" },
        "security_contact": { "type": "string", "format": "email" },
        "processing_recipients": {
          "type": "array",
          "items": { "$ref": "common.schema.json#/$defs/processingRecipient" },
          "uniqueItems": true
        },
        "retention_supported": {
          "type": "array",
          "items": { "enum": ["no_store", "session", "while_connected"] },
          "minItems": 1,
          "uniqueItems": true
        },
        "processing_uses_supported": {
          "type": "array",
          "items": { "enum": ["task_use", "model_training", "product_improvement", "advertising"] },
          "minItems": 1,
          "uniqueItems": true
        },
        "offline_use_supported": {
          "type": "array",
          "items": { "enum": ["prohibited", "explicitly_authorized"] },
          "minItems": 1,
          "uniqueItems": true
        },
        "proposal_rights_supported": {
          "type": "array",
          "items": { "enum": ["none", "bounded"] },
          "minItems": 1,
          "uniqueItems": true
        },
        "delegation_supported": { "const": false },
        "deletion_support": {
          "type": "object",
          "required": ["request_supported", "status_receipt_supported"],
          "properties": {
            "request_supported": { "type": "boolean" },
            "status_receipt_supported": { "type": "boolean" },

```

```

    "request_uri": {"$ref": "common.schema.json#/$defs/uri"}
  },
  "additionalProperties": false
},
"conformance_statement_uri": {"$ref": "common.schema.json#/$defs/uri"},
"assurance_evidence": {
  "type": "object",
  "required": ["evaluator", "relationship", "method", "evidence_ref", "scope", "assessed_at",
"limitations"],
  "properties": {
    "evaluator": {"$ref": "common.schema.json#/$defs/actorRef"},
    "relationship": {"enum": ["provider", "independent_third_party"]},
    "method": {"$ref": "common.schema.json#/$defs/uri"},
    "evidence_ref": {
      "allOf": [
        {"$ref": "common.schema.json#/$defs/snapshotRef"},
        {"type": "object", "properties": {"schema_uri": {"const":
"https://opencontinuity.org/schemas/0.4/source-record.schema.json"}}}
      ],
      "description": "Exact snapshot of a schema-valid OCP 0.4 SourceRecord carrying the
assurance evidence."
    },
    "scope": {"type": "string", "minLength": 1, "maxLength": 2000},
    "assessed_at": {"$ref": "common.schema.json#/$defs/timestamp"},
    "limitations": {"type": "string", "minLength": 1, "maxLength": 2000}
  },
  "additionalProperties": false
},
"declaration_status": {"enum": ["self_declared", "provider_verified", "independently_assessed"]},
"valid_until": {"$ref": "common.schema.json#/$defs/timestamp"}
},
"allOf": [
  {
    "if": {
      "properties": {"declaration_status": {"const": "self_declared"}},
      "required": ["declaration_status"]
    },
    "then": {"not": {"required": ["assurance_evidence"]}}
  },
  {
    "if": {
      "properties": {"declaration_status": {"const": "provider_verified"}},
      "required": ["declaration_status"]
    },
    "then": {
      "required": ["conformance_statement_uri", "assurance_evidence"],
      "properties": {
        "assurance_evidence": {
          "type": "object",
          "properties": {"relationship": {"const": "provider"}}
        }
      }
    }
  },
  {
    "if": {
      "properties": {"declaration_status": {"const": "independently_assessed"}},
      "required": ["declaration_status"]
    },
    "then": {
      "required": ["conformance_statement_uri", "assurance_evidence"],
      "properties": {
        "assurance_evidence": {
          "type": "object",
          "properties": {"relationship": {"const": "independent_third_party"}}
        }
      }
    }
  }
]
},
"unevaluatedProperties": false
}

```

43.22 schemas/core/common.schema.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/common.schema.json",
  "title": "Open Continuity Protocol 0.4 common definitions",
  "$defs": {
    "uri": {
      "type": "string",
      "anyOf": [
        {
          "pattern": "^urn:uuid:[0-9a-f]{8}-[0-9a-f]{4}-[1-8][0-9a-f]{3}-[89ab][0-9a-f]{3}-[0-9a-f]{12}$",
          {"format": "uri", "not": {"pattern": "^urn:uuid:"}}
        ]
      ],
      "timestamp": {"description": "Canonical RFC 3339 instant with uppercase T/Z, seconds 00-59, and at most millisecond precision. The baseline excludes leap-second lexical values so independent ECMAScript-compatible comparators do not collapse or reject distinct accepted instants.", "type": "string", "format": "date-time", "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-5][0-9](?:\\.([0-9]{1,3})?(?:Z|[-+][0-9]{2}:[0-9]{2}))$"},
      "date": {"type": "string", "format": "date"},
      "digest": {"type": "string", "pattern": "^sha-256:[0-9a-f]{64}$"},
      "mediaType": {"type": "string", "pattern": "^[A-Za-z0-9!#$%&_+. -]/[A-Za-z0-9!#$%&_+. -]+$"},
      "actorRef": {
        "type": "object",
        "required": ["id", "type"],
        "properties": {
          "id": {"$ref": "#/$defs/uri"},
          "type": {"enum": ["person", "organization", "service", "agent", "model", "device", "process"]},
          "display_name": {"type": "string", "minLength": 1, "maxLength": 200}
        },
        "additionalProperties": false
      },
      "subjectRef": {
        "type": "object",
        "required": ["id", "type"],
        "properties": {
          "id": {"$ref": "#/$defs/uri"},
          "type": {"enum": ["person", "organization", "household", "team", "project", "resource"]},
          "display_name": {"type": "string", "minLength": 1, "maxLength": 200}
        },
        "additionalProperties": false
      },
      "snapshotRef": {
        "description": "Immutable reference to the exact revision and RFC 8785 canonical bytes of a record.",
        "type": "object",
        "required": ["id", "revision", "schema_uri", "digest"],
        "properties": {
          "id": {"$ref": "#/$defs/uri"},
          "revision": {"type": "integer", "minimum": 1},
          "schema_uri": {"$ref": "#/$defs/uri"},
          "digest": {"$ref": "#/$defs/digest"}
        },
        "additionalProperties": false
      },
      "liveStatusRef": {
        "description": "Dereferenceable status endpoint; never a substitute for an immutable snapshot reference.",
        "type": "object",
        "required": ["uri"],
        "properties": {
          "uri": {"$ref": "#/$defs/uri"},
          "last_checked_at": {"$ref": "#/$defs/timestamp"}
        },
        "additionalProperties": false
      },
      "runtimeSource": {
        "type": "object",
        "required": ["actor", "software_name", "software_version", "execution_context"],
        "properties": {
          "actor": {"$ref": "#/$defs/actorRef"},
          "software_name": {"type": "string", "minLength": 1, "maxLength": 200},
          "software_version": {"type": "string", "minLength": 1, "maxLength": 100},
          "execution_context": {"enum": ["local", "provider", "third_party_service", "offline_import"]},
          "build_digest": {"$ref": "#/$defs/digest"}
        }
      }
    }
  }
}
```

```

    },
    "additionalProperties": false
  },
  "deliveryRecipient": {
    "description": "The single client service and instance to which this operation or package is
delivered.",
    "type": "object",
    "required": ["recipient", "client_instance", "client_declaration_ref"],
    "properties": {
      "recipient": {"$ref": "#/$defs/actorRef"},
      "client_instance": {"$ref": "#/$defs/actorRef"},
      "client_declaration_ref": {"$ref": "#/$defs/snapshotRef"}
    },
    "additionalProperties": false
  },
  "processingRecipient": {
    "description": "An entity approved to receive plaintext while processing for the delivery
recipient. This is not OCP delegation.",
    "type": "object",
    "required": ["recipient", "roles", "purpose_codes", "receives_plaintext"],
    "properties": {
      "recipient": {"$ref": "#/$defs/actorRef"},
      "roles": {
        "type": "array",
        "items": {"enum": ["model_provider", "subprocessor", "hosting_provider", "analytics_provider",
"other"]},
        "minItems": 1,
        "uniqueItems": true
      },
      "purpose_codes": {"type": "array", "items": {"$ref": "#/$defs/uri"}, "minItems": 1,
"uniqueItems": true},
      "receives_plaintext": {"const": true}
    },
    "additionalProperties": false
  },
  "extension": {
    "type": "object",
    "required": ["critical", "data"],
    "properties": {"critical": {"type": "boolean"}, "data": {}},
    "additionalProperties": false
  },
  "extensions": {
    "type": "object",
    "propertyNames": {"format": "uri"},
    "additionalProperties": {"$ref": "#/$defs/extension"}
  },
  "recordEnvelope": {
    "type": "object",
    "required": ["ocp_version", "schema_uri", "type", "id", "revision", "created_at", "created_by",
"extensions"],
    "properties": {
      "ocp_version": {"const": "0.4"},
      "schema_uri": {"$ref": "#/$defs/uri"},
      "type": {"type": "string", "pattern": "^[a-z][a-z0-9_]{0,63}$"},
      "id": {"$ref": "#/$defs/uri"},
      "revision": {"type": "integer", "minimum": 1},
      "created_at": {"$ref": "#/$defs/timestamp"},
      "created_by": {"$ref": "#/$defs/actorRef"},
      "correlation_id": {"$ref": "#/$defs/uri"},
      "extensions": {"$ref": "#/$defs/extensions"}
    }
  },
  "sensitivity": {"enum": ["public", "normal", "sensitive", "restricted", "highly_restricted"]},
  "contextCategory": {"enum": ["statement", "behavior_preference", "goal", "decision_note", "note"]},
  "epistemicBasis": {"enum": ["subject_asserted", "subject_confirmed", "provider_observed",
"provider_inferred", "provider_interpreted", "third_party_asserted", "credential_derived",
"imported_unclassified"]},
  "recordLifecycle": {
    "type": "object",
    "required": ["status"],
    "properties": {
      "status": {"enum": ["active", "superseded", "withdrawn", "tombstoned"]},
      "supersedes": {"type": "array", "items": {"$ref": "#/$defs/snapshotRef"}, "uniqueItems": true},
      "superseded_by": {"$ref": "#/$defs/snapshotRef"},
      "changed_at": {"$ref": "#/$defs/timestamp"}
    },
    "additionalProperties": false
  }

```

```

    },
    "purpose": {
      "type": "object",
      "required": ["code", "description"],
      "properties": {
        "code": {"$ref": "#/$defs/uri"},
        "description": {"type": "string", "minLength": 12, "maxLength": 500}
      },
      "additionalProperties": false
    },
    "timeRange": {
      "type": "object",
      "properties": {"from": {"$ref": "#/$defs/timestamp"}, "until": {"$ref": "#/$defs/timestamp"}},
      "minProperties": 1,
      "additionalProperties": false
    },
    "selector": {
      "type": "object",
      "required": ["layer"],
      "properties": {
        "layer": {"enum": ["context", "evidence", "artifact"]},
        "space_refs": {"type": "array", "items": {"$ref": "#/$defs/snapshotRef"}, "minItems": 1,
"uniqueItems": true},
        "item_categories": {"type": "array", "items": {"enum": ["statement", "behavior_preference",
"goal", "decision_note", "note"]}, "minItems": 1, "uniqueItems": true},
        "predicates": {"type": "array", "items": {"$ref": "#/$defs/uri"}, "minItems": 1, "uniqueItems":
true},
        "profile": {"$ref": "#/$defs/uri"},
        "time_range": {"$ref": "#/$defs/timeRange"},
        "sensitivity_max": {"$ref": "#/$defs/sensitivity"}
      },
      "additionalProperties": false
    },
    "disclosureViews": {
      "type": "object",
      "required": ["context", "evidence", "artifacts"],
      "properties": {
        "context": {"enum": ["none", "accepted"]},
        "evidence": {"enum": ["none", "metadata"]},
        "artifacts": {"enum": ["none", "metadata"]}
      },
      "additionalProperties": false
    },
    "retention": {
      "type": "object",
      "required": ["mode", "legal_retention_may_apply"],
      "properties": {
        "mode": {"enum": ["no_store", "session", "fixed", "while_connected"]},
        "delete_by": {"$ref": "#/$defs/timestamp"},
        "legal_retention_may_apply": {"type": "boolean"}
      },
      "allOf": [
        {"if": {"properties": {"mode": {"const": "fixed"}}, "required": ["mode"]}, "then": {"required":
["delete_by"]}, "else": {"not": {"required": ["delete_by"]}}}
      ],
      "additionalProperties": false
    },
    "processingUses": {
      "type": "object",
      "required": ["task_use", "model_training", "product_improvement", "advertising"],
      "properties": {
        "task_use": {"const": "permitted"},
        "model_training": {"enum": ["prohibited", "requested", "permitted"]},
        "product_improvement": {"enum": ["prohibited", "requested", "permitted"]},
        "advertising": {"enum": ["prohibited", "requested", "permitted"]}
      },
      "additionalProperties": false
    },
    "approvedProcessingUses": {
      "type": "object",
      "required": ["task_use", "model_training", "product_improvement", "advertising"],
      "properties": {
        "task_use": {"const": "permitted"},
        "model_training": {"enum": ["prohibited", "permitted"]},
        "product_improvement": {"enum": ["prohibited", "permitted"]},
        "advertising": {"enum": ["prohibited", "permitted"]}
      },
    },

```

```

    "additionalProperties": false
  },
  "proposalRights": {
    "oneOf": [
      {
        "type": "object",
        "required": ["mode"],
        "properties": {"mode": {"const": "none"}},
        "additionalProperties": false
      },
      {
        "type": "object",
        "required": ["mode", "operations", "selectors", "source_attachment"],
        "properties": {
          "mode": {"const": "bounded"},
          "operations": {"type": "array", "items": {"enum": ["create", "revise"]}, "minItems": 1,
            "uniqueItems": true},
          "selectors": {"type": "array", "items": {"$ref": "#/$defs/selector"}, "minItems": 1},
          "sensitivity_max": {"$ref": "#/$defs/sensitivity"},
          "source_attachment": {"const": "referenced_authorized", "description": "Baseline
create/revise proposals require exact authorized provenance. Profiles may define a distinct no-
attachment operation only by separating required provenance from optional attachments."}
        },
        "additionalProperties": false
      }
    ]
  },
  "delegation": {
    "description": "Baseline OCP 0.4 permits no downstream OCP authority delegation.",
    "type": "object",
    "required": ["mode"],
    "properties": {"mode": {"const": "none"}},
    "additionalProperties": false
  },
  "authorityBasis": {
    "type": "object",
    "required": ["type"],
    "properties": {
      "type": {"enum": ["self", "delegated_stewardship", "joint_decision", "guardian",
"organizational_policy", "legal_authority"]},
      "reference": {"$ref": "#/$defs/snapshotRef"},
      "scope_note": {"type": "string", "minLength": 1, "maxLength": 500}
    },
    "additionalProperties": false
  },
  "acceptedScope": {
    "description": "The exact spaces and/or purpose codes for which a context-item revision was
accepted.",
    "type": "object",
    "properties": {
      "space_refs": {"type": "array", "items": {"$ref": "#/$defs/snapshotRef"}, "minItems": 1,
        "uniqueItems": true},
      "purpose_codes": {"type": "array", "items": {"$ref": "#/$defs/uri"}, "minItems": 1,
        "uniqueItems": true}
    },
    "minProperties": 1,
    "additionalProperties": false
  },
  "typedValue": {
    "oneOf": [
      {"type": "object", "required": ["type", "value"], "properties": {"type": {"const": "string"},
"value": {"type": "string", "language": {"type": "string", "pattern": "^[A-Za-z]{2,3}(?:-[A-Za-z0-9]{2,8})*$"}},
"additionalProperties": false},
      {"type": "object", "required": ["type", "value"], "properties": {"type": {"const": "boolean"},
"value": {"type": "boolean"}}, "additionalProperties": false},
      {"type": "object", "required": ["type", "value"], "properties": {"type": {"const": "number"},
"value": {"type": "number", "minimum": -9007199254740991, "maximum": 9007199254740991}},
"additionalProperties": false},
      {"type": "object", "required": ["type", "value"], "properties": {"type": {"const": "decimal"},
"value": {"description": "Canonical lexical decimal: no exponent, no leading plus/zero, no negative
zero, and no trailing fractional zero.", "type": "string", "pattern": "^(?:0|-?[1-9][0-9]*|-?(?:0|[1-9]
[0-9]*)\\.[0-9]*[1-9])$"}}, "additionalProperties": false},
      {"type": "object", "required": ["type", "value"], "properties": {"type": {"const": "date"},
"value": {"$ref": "#/$defs/date"}}, "additionalProperties": false},
      {"type": "object", "required": ["type", "value"], "properties": {"type": {"const": "date_time"},
"value": {"$ref": "#/$defs/timestamp"}}, "additionalProperties": false},
      {"type": "object", "required": ["type", "value", "unit"], "properties": {"type": {"const":

```

```

"quantity", "value": {"description": "Canonical lexical decimal: no exponent, no leading plus/zero, no
negative zero, and no trailing fractional zero.", "type": "string", "pattern": "^(?:0|-?[1-9][0-9]*|-?
(?:0|[1-9][0-9]*)\\.[0-9]*[1-9])$"}, "unit": {"type": "string", "minLength": 1, "maxLength": 50}},
"additionalProperties": false},
  {"type": "object", "required": ["type", "ref"], "properties": {"type": {"const": "entity_ref"},
"ref": {"$ref": "#/$defs/uri"}}, "additionalProperties": false},
  {"type": "object", "required": ["type", "schema_uri", "value"], "properties": {"type": {"const":
"structred"}, "schema_uri": {"$ref": "#/$defs/uri"}, "value": {"type": ["object", "array"]}},
"additionalProperties": false}
],
"statement": {
  "type": "object",
  "required": ["predicate", "object"],
  "properties": {
    "predicate": {"$ref": "#/$defs/uri"},
    "object": {"$ref": "#/$defs/typedValue"},
    "qualifiers": {"type": "object", "propertyNames": {"format": "uri"}, "additionalProperties":
{"$ref": "#/$defs/typedValue"}
},
    "additionalProperties": false
},
"provenance": {
  "type": "object",
  "required": ["source_ref", "method"],
  "properties": {
    "source_ref": {"$ref": "#/$defs/snapshotRef"},
    "method": {"enum": ["subject_entry", "import", "observation", "inference", "transformation",
"credential_verification"]},
    "activity_ref": {"$ref": "#/$defs/snapshotRef"},
    "observed_at": {"$ref": "#/$defs/timestamp"},
    "acquired_at": {"$ref": "#/$defs/timestamp"},
    "producer": {"$ref": "#/$defs/actorRef"},
    "runtime_source": {"$ref": "#/$defs/runtimeSource"},
    "adapter_version": {"type": "string", "minLength": 1, "maxLength": 100},
    "redacted": {"type": "boolean"}
},
    "additionalProperties": false
},
"verificationEvidence": {
  "type": "object",
  "required": ["verifier", "profile", "profile_version", "issuer", "covered_digest",
"covered_representation", "verification_result", "proof_or_key_ref", "verified_at", "status_method",
"status_result", "status_checked_at", "limitations"],
  "properties": {
    "verifier": {"$ref": "#/$defs/actorRef"},
    "profile": {"$ref": "#/$defs/uri"},
    "profile_version": {"type": "string", "minLength": 1, "maxLength": 100},
    "issuer": {"$ref": "#/$defs/actorRef"},
    "covered_digest": {"$ref": "#/$defs/digest"},
    "covered_representation": {"const": "source_content_bytes"},
    "verification_result": {"const": "verified"},
    "proof_or_key_ref": {"$ref": "#/$defs/uri"},
    "verified_at": {"$ref": "#/$defs/timestamp"},
    "status_method": {"$ref": "#/$defs/uri"},
    "status_result": {"const": "good"},
    "status_checked_at": {"$ref": "#/$defs/timestamp"},
    "limitations": {"type": "string", "minLength": 1, "maxLength": 2000}
},
    "additionalProperties": false
},
"integrity": {
  "type": "object",
  "required": ["digest", "canonicalization"],
  "properties": {
    "digest": {"$ref": "#/$defs/digest"},
    "canonicalization": {"const": "RFC8785"},
    "signature_ref": {"$ref": "#/$defs/uri"}
},
    "additionalProperties": false
}
}
}

```

43.23 schemas/core/context-item.schema.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/context-item.schema.json",
  "title": "OCP 0.4 context item",
  "type": "object",
  "allOf": [
    {
      "$ref": "common.schema.json#/$defs/recordEnvelope",
      {
        "type": "object",
        "required": ["type", "subjects", "space_ref", "category", "statement", "epistemic_basis", "review",
"provenance", "sensitivity", "lifecycle"],
        "properties": {
          "type": {"const": "context_item"},
          "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/context-item.schema.json"},
          "subjects": {"type": "array", "items": {"$ref": "common.schema.json#/$defs/subjectRef"},
"minItems": 1},
          "space_ref": {"$ref": "common.schema.json#/$defs/snapshotRef"},
          "category": {"$ref": "common.schema.json#/$defs/contextCategory"},
          "instruction_treatment": {"const": "untrusted_data"},
          "statement": {"$ref": "common.schema.json#/$defs/statement"},
          "epistemic_basis": {"$ref": "common.schema.json#/$defs/epistemicBasis"},
          "review": {
            "type": "object",
            "required": ["status"],
            "properties": {
              "status": {"enum": ["candidate", "accepted"]},
              "decision_ref": {"$ref": "common.schema.json#/$defs/snapshotRef"},
              "accepted_scope": {"$ref": "common.schema.json#/$defs/acceptedScope"}
            },
            "allOf": [
              {"if": {"properties": {"status": {"const": "accepted"}}, "required": ["status"]}, "then":
{"required": ["decision_ref", "accepted_scope"]}, "else": {"not": {"anyOf": [{"required":
["decision_ref"]}, {"required": ["accepted_scope"]}]}
            ],
            "additionalProperties": false
          },
          "validity": {
            "type": "object",
            "properties": {
              "observed_at": {"$ref": "common.schema.json#/$defs/timestamp"},
              "valid_from": {"$ref": "common.schema.json#/$defs/timestamp"},
              "valid_until": {"$ref": "common.schema.json#/$defs/timestamp"},
              "last_confirmed_at": {"$ref": "common.schema.json#/$defs/timestamp"},
              "review_after": {"$ref": "common.schema.json#/$defs/timestamp"}
            },
            "minProperties": 1,
            "additionalProperties": false
          },
          "provenance": {"type": "array", "items": {"$ref": "common.schema.json#/$defs/provenance"},
"minItems": 1},
          "sensitivity": {"$ref": "common.schema.json#/$defs/sensitivity"},
          "rights": {
            "type": "object",
            "properties": {
              "custodian": {"$ref": "common.schema.json#/$defs/actorRef"},
              "disclosure_basis_ref": {"$ref": "common.schema.json#/$defs/snapshotRef"},
              "organizational_policy_ref": {"$ref": "common.schema.json#/$defs/snapshotRef"},
              "residency_constraint": {"type": "string", "minLength": 2, "maxLength": 100}
            },
            "minProperties": 1,
            "additionalProperties": false
          },
          "lifecycle": {"$ref": "common.schema.json#/$defs/recordLifecycle"}
        },
        "allOf": [
          {"if": {"properties": {"category": {"const": "behavior_preference"}}, "required": ["category"]},
"then": {"required": ["instruction_treatment"]}
        ]
      },
    },
    {
      "unevaluatedProperties": false
    }
  ]
}
```

43.24 schemas/core/context-package.schema.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/context-package.schema.json",
  "title": "OCP 0.4 context package",
  "$defs": {
    "baseEntry": {
      "type": "object",
      "required": ["entry_id", "layer", "disclosure", "media_type", "content_schema_uri", "digest",
        "instruction_treatment"],
      "properties": {
        "entry_id": {"$ref": "common.schema.json#/$defs/uri"},
        "layer": {"enum": ["context", "evidence", "artifact"]},
        "disclosure": {"enum": ["metadata", "accepted"]},
        "media_type": {"$ref": "common.schema.json#/$defs/mediaType"},
        "content_schema_uri": {"description": "Exact schema for the embedded baseline projection or for
a separately registered content profile.", "$ref": "common.schema.json#/$defs/uri"},
        "digest": {"$ref": "common.schema.json#/$defs/digest"},
        "package_item_ref": {"$ref": "common.schema.json#/$defs/uri"},
        "content_ref": {"$ref": "common.schema.json#/$defs/uri"},
        "content": {},
        "instruction_treatment": {"const": "untrusted_data"}
      },
      "additionalProperties": false
    },
    "packageContextContent": {
      "type": "object",
      "required": ["subject_refs", "space_ref", "category", "statement", "epistemic_basis",
        "review_status", "provenance_refs", "sensitivity", "instruction_treatment"],
      "properties": {
        "subject_refs": {"type": "array", "items": {"$ref": "common.schema.json#/$defs/uri"}, "minItems":
1, "uniqueItems": true},
        "space_ref": {"$ref": "common.schema.json#/$defs/uri"},
        "category": {"$ref": "common.schema.json#/$defs/contextCategory"},
        "statement": {"$ref": "common.schema.json#/$defs/statement"},
        "epistemic_basis": {"$ref": "common.schema.json#/$defs/epistemicBasis"},
        "review_status": {"const": "accepted"},
        "decision_ref": {"$ref": "common.schema.json#/$defs/uri"},
        "provenance_refs": {"type": "array", "items": {"$ref": "common.schema.json#/$defs/uri"},
"minItems": 1, "uniqueItems": true},
        "sensitivity": {"$ref": "common.schema.json#/$defs/sensitivity"},
        "instruction_treatment": {"const": "untrusted_data"}
      },
      "allOf": [
        {"if": {"properties": {"review_status": {"const": "accepted"}}, "required": ["review_status"]},
        "then": {"required": ["decision_ref"]}}
      ],
      "additionalProperties": false
    },
    "packageEvidenceMetadata": {
      "description": "Package-local evidence metadata only. It is bound to a package-local space and one
or more packaged accepted items whose exact provenance names the resolved source. It carries no raw
evidence bytes, verification assurance, source content reference, filename, or stable store digest.",
      "type": "object",
      "required": ["package_source_ref", "package_space_ref", "supports_package_item_refs",
        "source_type", "trust_treatment", "availability", "sensitivity", "instruction_treatment"],
      "properties": {
        "package_source_ref": {"$ref": "common.schema.json#/$defs/uri"},
        "package_space_ref": {"$ref": "common.schema.json#/$defs/uri"},
        "supports_package_item_refs": {"type": "array", "items": {"$ref":
"common.schema.json#/$defs/uri"}, "minItems": 1, "uniqueItems": true},
        "source_type": {"enum": ["conversation", "document", "provider_memory", "event", "credential",
"media", "export_record", "other"]},
        "trust_treatment": {"enum": ["unassessed", "user_supplied", "suspected_tampering"]},
        "availability": {"enum": ["available", "unavailable", "unknown", "quarantined"]},
        "observed_at": {"$ref": "common.schema.json#/$defs/timestamp"},
        "acquired_at": {"$ref": "common.schema.json#/$defs/timestamp"},
        "sensitivity": {"$ref": "common.schema.json#/$defs/sensitivity"},
        "instruction_treatment": {"const": "untrusted_data"}
      },
      "additionalProperties": false
    },
    "packageArtifactMetadata": {
      "description": "Package-local artifact metadata only. It is bound to a package-local space and one
or more packaged accepted items. It carries no artifact bytes, custody or preservation assurance, store
```

```

content reference, filename, or stable store digest.",
  "type": "object",
  "required": ["package_artifact_ref", "package_space_ref", "supports_package_item_refs",
"artifact_kind", "media_type", "content_disclosed", "sensitivity", "instruction_treatment"],
  "properties": {
    "package_artifact_ref": {"$ref": "common.schema.json#/$defs/uri"},
    "package_space_ref": {"$ref": "common.schema.json#/$defs/uri"},
    "supports_package_item_refs": {"type": "array", "items": {"$ref":
"common.schema.json#/$defs/uri"}, "minItems": 1, "uniqueItems": true},
    "artifact_kind": {"enum": ["document", "spreadsheet", "presentation", "image", "audio", "video",
"code", "dataset", "notebook", "prompt", "model_configuration", "other"]},
    "media_type": {"$ref": "common.schema.json#/$defs/mediaType"},
    "content_disclosed": {"const": false},
    "sensitivity": {"$ref": "common.schema.json#/$defs/sensitivity"},
    "instruction_treatment": {"const": "untrusted_data"}
  },
  "additionalProperties": false
},
"packageSelector": {
  "description": "Package-local selector. space_refs are aliases, never store snapshot references or
stable digests.",
  "type": "object",
  "required": ["layer"],
  "properties": {
    "layer": {"enum": ["context", "evidence", "artifact"]},
    "space_refs": {"type": "array", "items": {"$ref": "common.schema.json#/$defs/uri"}, "minItems":
1, "uniqueItems": true},
    "item_categories": {"type": "array", "items": {"$ref":
"common.schema.json#/$defs/contextCategory"}, "minItems": 1, "uniqueItems": true},
    "predicates": {"type": "array", "items": {"$ref": "common.schema.json#/$defs/uri"}, "minItems":
1, "uniqueItems": true},
    "profile": {"$ref": "common.schema.json#/$defs/uri"},
    "time_range": {"$ref": "common.schema.json#/$defs/timeRange"},
    "sensitivity_max": {"$ref": "common.schema.json#/$defs/sensitivity"}
  },
  "additionalProperties": false
},
"packageProposalRights": {
  "oneOf": [
    {"type": "object", "required": ["mode"], "properties": {"mode": {"const": "none"}}},
    {"type": "object", "required": ["mode", "operations", "selectors", "source_attachment"],
      "properties": {
        "mode": {"const": "bounded"},
        "operations": {"type": "array", "items": {"enum": ["create", "revise"]}, "minItems": 1,
"uniqueItems": true},
        "selectors": {"type": "array", "items": {"$ref": "#/$defs/packageSelector"}, "minItems": 1},
        "sensitivity_max": {"$ref": "common.schema.json#/$defs/sensitivity"},
        "source_attachment": {"const": "referenced_authorized"}
      },
      "additionalProperties": false
    }
  ]
},
"packageReferenceAlias": {
  "description": "Declares a package-local alias and its role without revealing the provider-side
stable-identifier mapping.",
  "type": "object",
  "required": ["package_ref", "kind"],
  "properties": {
    "package_ref": {"$ref": "common.schema.json#/$defs/uri"},
    "kind": {"enum": ["space", "decision", "source", "artifact", "relationship", "other"]}
  },
  "additionalProperties": false
},
"contextEntry": {
  "allOf": [
    {"$ref": "#/$defs/baseEntry"},
    {"type": "object", "required": ["content"], "not": {"required": ["content_ref"]}},
    {
      "type": "object",
      "properties": {
        "layer": {"const": "context"},
        "disclosure": {"const": "accepted"},
        "content_schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/context-

```

```

package.schema.json#/$defs/packageContextContent"},
  "content": {"$ref": "#/$defs/packageContextContent"}
},
"required": ["package_item_ref"],
"allOf": [
  {"if": {"properties": {"disclosure": {"const": "accepted"}}, "required": ["disclosure"]},
"then": {"properties": {"content": {"type": "object", "properties": {"review_status": {"const":
"accepted"}}}}}
]
}
},
"evidenceEntry": {
  "allOf": [
    {"$ref": "#/$defs/baseEntry"},
    {"type": "object", "required": ["content"], "not": {"anyOf": [{"required": ["content_ref"]},
{"required": ["package_item_ref"]]}}, "properties": {"layer": {"const": "evidence"}, "disclosure":
{"const": "metadata"}, "content_schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/context-
package.schema.json#/$defs/packageEvidenceMetadata"}, "content": {"$ref":
"#/$defs/packageEvidenceMetadata"}}}
]
},
"artifactEntry": {
  "allOf": [
    {"$ref": "#/$defs/baseEntry"},
    {"type": "object", "required": ["content"], "not": {"anyOf": [{"required": ["content_ref"]},
{"required": ["package_item_ref"]]}}, "properties": {"layer": {"const": "artifact"}, "disclosure":
{"const": "metadata"}, "content_schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/context-
package.schema.json#/$defs/packageArtifactMetadata"}, "content": {"$ref":
"#/$defs/packageArtifactMetadata"}}}
]
}
},
"type": "object",
"allOf": [
  {"$ref": "common.schema.json#/$defs/recordEnvelope"},
  {
    "type": "object",
    "required": ["type", "grant_receipt_ref", "grant_status_ref", "purpose", "delivery_recipient",
"processing_recipients", "issued_at", "expires_at", "subject_aliases", "reference_aliases", "entries",
"omissions", "warnings", "retention", "processing_uses", "offline_use", "proposal_rights", "delegation",
"external_action_authority", "integrity"],
    "properties": {
      "type": {"const": "context_package"},
      "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/context-package.schema.json"},
      "grant_receipt_ref": {"$ref": "common.schema.json#/$defs/snapshotRef"},
      "grant_status_ref": {"allOf": [{"$ref": "common.schema.json#/$defs/liveStatusRef"}, {"type":
"object", "required": ["last_checked_at"]}]},
      "purpose": {"$ref": "common.schema.json#/$defs/purpose"},
      "delivery_recipient": {"$ref": "common.schema.json#/$defs/deliveryRecipient"},
      "processing_recipients": {"type": "array", "items": {"$ref":
"common.schema.json#/$defs/processingRecipient"}, "uniqueItems": true},
      "issued_at": {"$ref": "common.schema.json#/$defs/timestamp"},
      "expires_at": {"$ref": "common.schema.json#/$defs/timestamp"},
      "subject_aliases": {
        "type": "array",
        "items": {
          "type": "object",
          "required": ["package_subject_ref", "delivery_alias"],
          "properties": {
            "package_subject_ref": {"$ref": "common.schema.json#/$defs/uri"},
            "delivery_alias": {"$ref": "common.schema.json#/$defs/uri"}
          },
          "additionalProperties": false
        },
        "minItems": 1,
        "uniqueItems": true
      },
      "reference_aliases": {
        "type": "array",
        "items": {"$ref": "#/$defs/packageReferenceAlias"},
        "minItems": 1,
        "uniqueItems": true
      },
      "entries": {"type": "array", "items": {"oneOf": [{"$ref": "#/$defs/contextEntry"}, {"$ref":
"#/$defs/evidenceEntry"}, {"$ref": "#/$defs/artifactEntry"}]}, "minItems": 1},
      "renderings": {

```

```

        "type": "array",
        "items": {
          "type": "object",
          "required": ["media_type", "derived_from_entries", "content_ref", "digest",
"instruction_treatment"],
          "properties": {
            "media_type": {"enum": ["text/plain", "text/markdown"]},
            "language": {"type": "string", "pattern": "^[A-Za-z]{2,3}(?:-[A-Za-z0-9]{2,8})*$"},
            "derived_from_entries": {"type": "array", "items": {"$ref":
"common.schema.json#/$defs/uri"}}, "minItems": 1},
            "content_ref": {"$ref": "common.schema.json#/$defs/uri"},
            "digest": {"$ref": "common.schema.json#/$defs/digest"},
            "instruction_treatment": {"const": "untrusted_data"}
          },
          "additionalProperties": false
        },
        "omissions": {
          "type": "array",
          "items": {"type": "object", "required": ["selector", "reason"], "properties": {"selector":
{"$ref": "#/$defs/packageSelector"}, "reason": {"enum": ["not_granted", "minimized", "unavailable",
"unsupported_profile", "third_party_restricted", "organizational_policy", "stale",
"integrity_failure"]}}, "additionalProperties": false}
        },
        "warnings": {
          "type": "array",
          "items": {"type": "object", "required": ["code", "message"], "properties": {"code": {"type":
"string", "pattern": "^OCP_[A-Z0-9_]+$"}, "message": {"type": "string", "minLength": 1, "maxLength":
500}, "item_refs": {"type": "array", "items": {"$ref": "common.schema.json#/$defs/uri"}}}},
          "additionalProperties": false}
        },
        "retention": {"$ref": "common.schema.json#/$defs/retention"},
        "processing_uses": {"$ref": "common.schema.json#/$defs/approvedProcessingUses"},
        "offline_use": {"enum": ["prohibited", "explicitly_authorized"]},
        "proposal_rights": {"$ref": "#/$defs/packageProposalRights"},
        "delegation": {"$ref": "common.schema.json#/$defs/delegation"},
        "external_action_authority": {"const": "none"},
        "integrity": {"$ref": "common.schema.json#/$defs/integrity"}
      }
    },
    "unevaluatedProperties": false
  }
}

```

43.25 schemas/core/context-request.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/context-request.schema.json",
  "title": "OCP 0.4 context request",
  "description": "An inert request for bounded context presentation. It is never authorization.",
  "type": "object",
  "allOf": [
    {"$ref": "common.schema.json#/$defs/recordEnvelope"},
    {
      "type": "object",
      "required": ["type", "delivery_recipient", "processing_recipients", "purpose", "selectors",
"views", "requested_not_after", "retention", "processing_uses", "offline_use", "proposal_rights",
"delegation", "status"],
      "properties": {
        "type": {"const": "context_request"},
        "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/context-request.schema.json"},
        "delivery_recipient": {"$ref": "common.schema.json#/$defs/deliveryRecipient"},
        "processing_recipients": {"type": "array", "items": {"$ref":
"common.schema.json#/$defs/processingRecipient"}, "uniqueItems": true},
        "purpose": {"$ref": "common.schema.json#/$defs/purpose"},
        "selectors": {"type": "array", "items": {"$ref": "common.schema.json#/$defs/selector"},
"minItems": 1},
        "views": {"$ref": "common.schema.json#/$defs/disclosureViews"},
        "requested_not_before": {"$ref": "common.schema.json#/$defs/timestamp"},
        "requested_not_after": {"$ref": "common.schema.json#/$defs/timestamp"},
        "retention": {"$ref": "common.schema.json#/$defs/retention"},
        "processing_uses": {"$ref": "common.schema.json#/$defs/processingUses"},
        "offline_use": {"enum": ["prohibited", "requested"]},
        "proposal_rights": {"$ref": "common.schema.json#/$defs/proposalRights"},
        "delegation": {"$ref": "common.schema.json#/$defs/delegation"},

```

```

    "status": {"enum": ["submitted", "awaiting_decision", "partially_approved", "approved", "denied",
"expired", "withdrawn", "superseded"]}
  }
},
"unevaluatedProperties": false
}

```

43.26 schemas/core/context-space.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/context-space.schema.json",
  "title": "OCP context space",
  "type": "object",
  "allOf": [
    {"$ref": "common.schema.json#/$defs/recordEnvelope"},
    {
      "type": "object",
      "required": ["type", "name", "subjects", "domain", "authority_model", "lifecycle"],
      "properties": {
        "type": {"const": "context_space"},
        "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/context-space.schema.json"},
        "name": {"type": "string", "minLength": 1, "maxLength": 200},
        "description": {"type": "string", "maxLength": 2000},
        "subjects": {
          "type": "array",
          "items": {"$ref": "common.schema.json#/$defs/subjectRef"},
          "minItems": 1
        },
        "domain": {"enum": ["personal", "organizational", "household", "project", "mixed"]},
        "authority_model": {
          "enum": ["self", "single_steward", "joint", "organizational_policy", "external_legal_basis"]
        },
        "authority_policy_ref": {"$ref": "common.schema.json#/$defs/snapshotRef"},
        "lifecycle": {"$ref": "common.schema.json#/$defs/recordLifecycle"}
      }
    }
  ],
  "unevaluatedProperties": false
}

```

43.27 schemas/core/grant-receipt.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/grant-receipt.schema.json",
  "title": "OCP 0.4 grant receipt",
  "description": "A non-secret decision record. It is not a token, capability, credential, or action authorization.",
  "type": "object",
  "allOf": [
    {"$ref": "common.schema.json#/$defs/recordEnvelope"},
    {
      "type": "object",
      "required": ["type", "request_ref", "request_status_ref", "authorization_ref",
"authorization_system", "authorizer", "authority_basis", "purpose", "approved_selectors",
"approved_views", "delivery_recipient", "processing_recipients", "not_before", "expires_at", "retention",
"processing_uses", "offline_use", "proposal_rights", "delegation", "status", "status_ref"],
      "properties": {
        "type": {"const": "grant_receipt"},
        "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/grant-receipt.schema.json"},
        "request_ref": {"$ref": "common.schema.json#/$defs/snapshotRef"},
        "request_status_ref": {"$ref": "common.schema.json#/$defs/liveStatusRef"},
        "authorization_ref": {"$ref": "common.schema.json#/$defs/liveStatusRef"},
        "authorization_system": {"enum": ["oauth", "gnap", "uma", "solid", "local_policy", "os_consent",
"other"]},
        "authorizer": {"$ref": "common.schema.json#/$defs/actorRef"},
        "authority_basis": {"$ref": "common.schema.json#/$defs/authorityBasis"},
        "purpose": {"$ref": "common.schema.json#/$defs/purpose"},
        "approved_selectors": {"type": "array", "items": {"$ref": "common.schema.json#/$defs/selector"},
"minItems": 1},
        "approved_views": {"$ref": "common.schema.json#/$defs/disclosureViews"},
        "denied_selectors": {"type": "array", "items": {"$ref": "common.schema.json#/$defs/selector"}},
        "delivery_recipient": {"$ref": "common.schema.json#/$defs/deliveryRecipient"},

```

```

    "processing_recipients": {"type": "array", "items": {"$ref":
"common.schema.json#/$defs/processingRecipient"}, "uniqueItems": true},
    "not_before": {"$ref": "common.schema.json#/$defs/timestamp"},
    "expires_at": {"$ref": "common.schema.json#/$defs/timestamp"},
    "retention": {"$ref": "common.schema.json#/$defs/retention"},
    "processing_uses": {"$ref": "common.schema.json#/$defs/approvedProcessingUses"},
    "offline_use": {"enum": ["prohibited", "permitted"]},
    "proposal_rights": {"$ref": "common.schema.json#/$defs/proposalRights"},
    "delegation": {"$ref": "common.schema.json#/$defs/delegation"},
    "status": {"enum": ["active", "suspended", "expired", "revoked", "superseded"]},
    "status_ref": {"allOf": [{"$ref": "common.schema.json#/$defs/liveStatusRef"}], {"type": "object",
"required": [{"last_checked_at"}]},
    "superseded_by": {"$ref": "common.schema.json#/$defs/snapshotRef"}
  }
},
"unevaluatedProperties": false
}

```

43.28 schemas/core/review-decision.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/review-decision.schema.json",
  "title": "OCP 0.4 review decision",
  "type": "object",
  "allOf": [
    {"$ref": "common.schema.json#/$defs/recordEnvelope"},
    {
      "type": "object",
      "required": ["type", "proposal_ref", "reviewed_by", "authority_basis", "outcome", "decided_at"],
      "properties": {
        "type": {"const": "review_decision"},
        "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/review-decision.schema.json"},
        "proposal_ref": {"$ref": "common.schema.json#/$defs/snapshotRef"},
        "reviewed_by": {"$ref": "common.schema.json#/$defs/actorRef"},
        "authority_basis": {"$ref": "common.schema.json#/$defs/authorityBasis"},
        "outcome": {"enum": ["accepted", "accepted_with_changes", "rejected", "disputed", "deferred"]},
        "resulting_item_commitment": {
          "description": "Commitment to the resulting item's non-circular accepted-result projection
defined in Section 12.10. It includes accepted_scope, created_at, created_by, extensions, and optional
correlation_id; it excludes the cyclic decision_ref and other review fields, plus ocp_version,
schema_uri, type, id, and revision (identity and revision are carried separately here).",
          "type": "object",
          "required": ["id", "revision", "accepted_scope", "semantic_digest", "projection"],
          "properties": {
            "id": {"$ref": "common.schema.json#/$defs/uri"},
            "revision": {"type": "integer", "minimum": 1},
            "accepted_scope": {"$ref": "common.schema.json#/$defs/acceptedScope"},
            "semantic_digest": {"$ref": "common.schema.json#/$defs/digest"},
            "projection": {"const": "ocp-0.4-context-item-accepted-result-v1"}
          },
          "additionalProperties": false
        },
        "reason": {"type": "string", "minLength": 1, "maxLength": 2000},
        "decided_at": {"$ref": "common.schema.json#/$defs/timestamp"}
      },
      "allOf": [
        {"if": {"properties": {"outcome": {"enum": ["accepted", "accepted_with_changes"]}}, "required":
["outcome"]}, "then": {"required": ["resulting_item_commitment"]}, "else": {"not": {"required":
["resulting_item_commitment"]}}}
      ]
    }
  ],
  "unevaluatedProperties": false
}

```

43.29 schemas/core/source-record.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/source-record.schema.json",
  "title": "OCP 0.4 source record",
  "type": "object",
  "allOf": [

```

```

    {"$ref": "common.schema.json#/$defs/recordEnvelope"},
    {
      "type": "object",
      "required": ["type", "source_type", "source_authority", "runtime_source", "acquisition_method",
"acquired_at", "trust_treatment", "instruction_treatment", "availability", "sensitivity", "lifecycle"],
      "properties": {
        "type": {"const": "source_record"},
        "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/source-record.schema.json"},
        "source_type": {"enum": ["conversation", "document", "provider_memory", "event", "credential",
"media", "export_record", "other"]},
        "source_authority": {"$ref": "common.schema.json#/$defs/actorRef"},
        "runtime_source": {"$ref": "common.schema.json#/$defs/runtimeSource"},
        "source_object_id": {"type": "string", "minLength": 1, "maxLength": 500},
        "source_uri": {"$ref": "common.schema.json#/$defs/uri"},
        "acquisition_method": {"enum": ["api", "export", "upload", "observation", "manual_entry"]},
        "observed_at": {"$ref": "common.schema.json#/$defs/timestamp"},
        "acquired_at": {"$ref": "common.schema.json#/$defs/timestamp"},
        "adapter_version": {"type": "string", "minLength": 1, "maxLength": 100},
        "trust_treatment": {"enum": ["unassessed", "user_supplied", "signature_verified",
"credential_proof_verified", "suspected_tampering"]},
        "verification_evidence": {"$ref": "common.schema.json#/$defs/verificationEvidence"},
        "instruction_treatment": {"const": "untrusted_data"},
        "content": {
          "type": "object",
          "required": ["media_type", "digest"],
          "properties": {
            "media_type": {"$ref": "common.schema.json#/$defs/mediaType"},
            "byte_size": {"type": "integer", "minimum": 0},
            "digest": {"$ref": "common.schema.json#/$defs/digest"},
            "content_ref": {"$ref": "common.schema.json#/$defs/uri"},
            "redacted": {"type": "boolean"}
          },
          "additionalProperties": false
        },
        "availability": {"enum": ["available", "unavailable", "unknown", "quarantined"]},
        "sensitivity": {"$ref": "common.schema.json#/$defs/sensitivity"},
        "lifecycle": {"$ref": "common.schema.json#/$defs/recordLifecycle"}
      },
      "allOf": [
        {
          "if": {"properties": {"trust_treatment": {"enum": ["signature_verified",
"credential_proof_verified"]}}, "required": ["trust_treatment"]},
          "then": {"required": ["content", "verification_evidence"]}
        }
      ]
    },
    "unevaluatedProperties": false
  }
}

```

43.30 schemas/core/update-proposal.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/update-proposal.schema.json",
  "title": "OCP 0.4 update proposal",
  "description": "An inert proposal. It cannot mutate accepted context without a review decision.",
  "type": "object",
  "allOf": [
    {"$ref": "common.schema.json#/$defs/recordEnvelope"},
    {
      "type": "object",
      "required": ["type", "submitted_by", "grant_receipt_ref", "grant_status_ref", "operation",
"source_refs", "status"],
      "properties": {
        "type": {"const": "update_proposal"},
        "schema_uri": {"const": "https://opencontinuity.org/schemas/0.4/update-proposal.schema.json"},
        "submitted_by": {"$ref": "common.schema.json#/$defs/actorRef"},
        "grant_receipt_ref": {"$ref": "common.schema.json#/$defs/snapshotRef"},
        "grant_status_ref": {"$ref": "common.schema.json#/$defs/liveStatusRef"},
        "operation": {"enum": ["create", "revise"]},
        "target_ref": {"$ref": "common.schema.json#/$defs/snapshotRef"},
        "proposed_item": {
          "allOf": [
            {"$ref": "context-item.schema.json"},
            {

```

```

        "type": "object",
        "properties": {
          "review": {"type": "object", "properties": {"status": {"const": "candidate"}}},
          "lifecycle": {"type": "object", "properties": {"status": {"const": "active"}}}
        }
      }
    ],
    "source_refs": {"type": "array", "items": {"$ref": "common.schema.json#/$defs/snapshotRef"}},
    "uniqueItems": true,
    "rationale": {"type": "string", "minLength": 1, "maxLength": 2000},
    "status": {"enum": ["submitted", "under_review", "deferred", "accepted", "accepted_with_changes",
"rejected", "disputed", "withdrawn", "expired", "superseded"]},
    "oneOf": [
      {"properties": {"operation": {"const": "create"}, "source_refs": {"type": "array", "minItems":
1}}, "required": ["operation", "proposed_item"], "not": {"required": ["target_ref"]}},
      {"properties": {"operation": {"const": "revise"}, "source_refs": {"type": "array", "minItems":
1}}, "required": ["operation", "target_ref", "proposed_item"]}
    ]
  },
  "unevaluatedProperties": false
}

```

43.31 schemas/history-bundle.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/history-bundle.schema.json",
  "title": "OCP Portable AI History fixture bundle",
  "description": "Synthetic composition container used by the 0.4 conformance corpus.",
  "type": "object",
  "required": [
    "bundle_id",
    "conversation",
    "messages",
    "artifacts",
    "lineage_events",
    "adapter_manifest",
    "mapping_run",
    "loss_report",
    "accepted_context_items",
    "embedded_test_bytes"
  ],
  "properties": {
    "bundle_id": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
    },
    "conversation": {
      "$ref": "https://opencontinuity.org/schemas/0.4/conversation.schema.json"
    },
    "messages": {
      "type": "array",
      "minItems": 1,
      "items": {
        "$ref": "https://opencontinuity.org/schemas/0.4/message.schema.json"
      }
    },
    "artifacts": {
      "type": "array",
      "items": {
        "$ref": "https://opencontinuity.org/schemas/0.4/artifact.schema.json"
      }
    },
    "lineage_events": {
      "type": "array",
      "items": {
        "$ref": "https://opencontinuity.org/schemas/0.4/lineage-event.schema.json"
      }
    },
    "adapter_manifest": {
      "$ref": "https://opencontinuity.org/schemas/0.4/provider-adapter-manifest.schema.json"
    },
    "mapping_run": {
      "$ref": "https://opencontinuity.org/schemas/0.4/mapping-run.schema.json"
    }
  }
}

```

```

    },
    "loss_report": {
      "$ref": "https://opencontinuity.org/schemas/0.4/loss-report.schema.json"
    },
    "accepted_context_items": {
      "type": "array",
      "items": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
      }
    },
    "embedded_test_bytes": {
      "type": "object",
      "propertyNames": {
        "format": "uri"
      },
      "additionalProperties": {
        "type": "string",
        "contentEncoding": "base64"
      }
    }
  },
  "additionalProperties": false
}

```

43.32 schemas/lineage-event.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/lineage-event.schema.json",
  "title": "OCP Artifact Profile lineage event",
  "description": "Historical lineage only; no lineage field grants action authority, ownership, rights, or truth.",
  "type": "object",
  "allOf": [
    {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/recordEnvelope"
    },
    {
      "type": "object",
      "required": [
        "type",
        "profile_id",
        "profile_version",
        "relation",
        "input_refs",
        "output_refs",
        "performed_by",
        "occurred_at",
        "external_action_authority"
      ],
      "properties": {
        "type": {
          "const": "lineage_event"
        },
        "schema_uri": {
          "const": "https://opencontinuity.org/schemas/0.4/lineage-event.schema.json"
        },
        "profile_id": {
          "const": "https://opencontinuity.org/profiles/artifact"
        },
        "profile_version": {
          "const": "0.4"
        },
        "relation": {
          "enum": [
            "created_from",
            "derived_from",
            "edited_from",
            "transcoded_from",
            "remixed_from",
            "extended_from",
            "generated_from_prompt",
            "generated_from_artifact",
            "selected_from_candidates",
            "upscaled_from",
            "translated_from",

```

```

        "transcribed_from",
        "captioned_from"
    ]
},
"input_refs": {
    "type": "array",
    "items": {
        "description": "Stable record identity and revision without a content digest. Used only
where a reciprocal snapshot digest would create a hash cycle.",
        "type": "object",
        "required": [
            "id",
            "revision",
            "schema_uri"
        ],
        "properties": {
            "id": {
                "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
            },
            "revision": {
                "type": "integer",
                "minimum": 1
            },
            "schema_uri": {
                "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
            }
        },
        "additionalProperties": false
    },
    "minItems": 1,
    "uniqueItems": true
},
"output_refs": {
    "type": "array",
    "items": {
        "description": "Stable record identity and revision without a content digest. Used only
where a reciprocal snapshot digest would create a hash cycle.",
        "type": "object",
        "required": [
            "id",
            "revision",
            "schema_uri"
        ],
        "properties": {
            "id": {
                "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
            },
            "revision": {
                "type": "integer",
                "minimum": 1
            },
            "schema_uri": {
                "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
            }
        },
        "additionalProperties": false
    },
    "minItems": 1,
    "uniqueItems": true
},
"performed_by": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
},
"requested_by": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
},
"model": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
},
"model_provider": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
},
"prompt_ref": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
},
"parameter_disclosure": {
    "enum": [

```

```

        "none",
        "partial",
        "complete"
      ]
    },
    "occurred_at": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
    },
    "source_record_ref": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
    },
    "external_action_authority": {
      "const": "none"
    }
  }
},
"unevaluatedProperties": false
}

```

43.33 schemas/loss-report.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/loss-report.schema.json",
  "title": "OCP Portable AI History loss report",
  "description": "Standardized machine-readable preservation and loss outcomes suitable for privacy-filtered human rendering.",
  "type": "object",
  "allOf": [
    {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/recordEnvelope"
    },
    {
      "type": "object",
      "required": [
        "type",
        "scope",
        "classification",
        "summary",
        "losses",
        "accepted_context_created_without_review",
        "visibility_scope"
      ],
      "properties": {
        "type": {
          "const": "loss_report"
        },
        "schema_uri": {
          "const": "https://opencontinuity.org/schemas/0.4/loss-report.schema.json"
        },
        "scope": {
          "enum": [
            "import",
            "export",
            "transfer",
            "roundtrip"
          ]
        },
        "classification": {
          "enum": [
            "lossless",
            "structurally_lossy",
            "semantically_lossy",
            "content_lossy",
            "authority_lossy"
          ]
        },
        "summary": {
          "type": "object",
          "required": [
            "messages_discovered",
            "messages_preserved",
            "conversation_branches_discovered",
            "conversation_branches_lost",
            "artifacts_discovered",

```

```

    "artifacts_independently_preserved",
    "artifacts_source_dependent",
    "provider_fields_opaque_preserved",
    "unsupported_fields_omitted"
  ],
  "properties": {
    "messages_discovered": {
      "type": "integer",
      "minimum": 0
    },
    "messages_preserved": {
      "type": "integer",
      "minimum": 0
    },
    "conversation_branches_discovered": {
      "type": "integer",
      "minimum": 0
    },
    "conversation_branches_lost": {
      "type": "integer",
      "minimum": 0
    },
    "artifacts_discovered": {
      "type": "integer",
      "minimum": 0
    },
    "artifacts_independently_preserved": {
      "type": "integer",
      "minimum": 0
    },
    "artifacts_source_dependent": {
      "type": "integer",
      "minimum": 0
    },
    "provider_fields_opaque_preserved": {
      "type": "integer",
      "minimum": 0
    },
    "unsupported_fields_omitted": {
      "type": "integer",
      "minimum": 0
    }
  },
  "additionalProperties": false
},
"losses": {
  "type": "array",
  "items": {
    "type": "object",
    "required": [
      "code",
      "class",
      "severity",
      "meaning",
      "profile_conformance_allowed",
      "roundtrip_lossless",
      "source_locator"
    ],
    "properties": {
      "code": {
        "type": "string",
        "pattern": "^(?:CONV|ART|ADAPT)_[A-Z0-9_]+$"
      },
      "class": {
        "enum": [
          "cosmetic",
          "metadata",
          "structural",
          "semantic",
          "authority",
          "content",
          "privacy_policy",
          "rights",
          "integrity"
        ]
      }
    }
  },
  "severity": {

```

```

    "enum": [
      "info",
      "warning",
      "error",
      "fatal"
    ]
  },
  "meaning": {
    "type": "string",
    "minLength": 1,
    "maxLength": 2000
  },
  "profile_conformance_allowed": {
    "type": "boolean"
  },
  "roundtrip_lossless": {
    "const": false
  },
  "object_ref": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
  },
  "source_locator": {
    "type": "object",
    "minProperties": 1,
    "properties": {
      "archive_member": {
        "type": "string",
        "minLength": 1,
        "maxLength": 4096
      },
      "json_pointer": {
        "type": "string",
        "pattern": "^(?:/(?:[~\\/]|~[01])*(?:/(?:[~\\/]|~[01])*)*)$"
      },
      "json_path": {
        "type": "string",
        "minLength": 1,
        "maxLength": 4096
      },
      "xml_path": {
        "type": "string",
        "minLength": 1,
        "maxLength": 4096
      },
      "database_table": {
        "type": "string",
        "minLength": 1,
        "maxLength": 500
      },
      "row_key": {
        "type": "string",
        "minLength": 1,
        "maxLength": 1000
      },
      "spreadsheet_sheet": {
        "type": "string",
        "minLength": 1,
        "maxLength": 500
      },
      "spreadsheet_row": {
        "type": "integer",
        "minimum": 1
      },
      "byte_range": {
        "type": "object",
        "required": [
          "start",
          "length"
        ],
        "properties": {
          "start": {
            "type": "integer",
            "minimum": 0
          },
          "length": {
            "type": "integer",
            "minimum": 0
          }
        }
      }
    }
  }
}

```

```

        },
        "additionalProperties": false
      },
      "provider_object_id": {
        "type": "string",
        "minLength": 1,
        "maxLength": 500
      },
      "source_uri": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
      }
    },
    "additionalProperties": false
  },
  "accepted_context_created_without_review": {
    "const": 0
  },
  "visibility_scope": {
    "type": "object",
    "required": [
      "recipient",
      "withheld_categories_disclosed"
    ],
    "properties": {
      "recipient": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
      },
      "withheld_categories_disclosed": {
        "const": false
      }
    },
    "additionalProperties": false
  }
},
"unevaluatedProperties": false
}

```

43.34 schemas/mapping-run.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/mapping-run.schema.json",
  "title": "OCP Provider Adapter mapping run",
  "description": "A replayable adapter execution record. Clock times and run ID are excluded from deterministic digest comparison.",
  "type": "object",
  "required": [
    "mapping_run_id",
    "input_manifest_digest",
    "adapter_id",
    "adapter_version",
    "mapping_profile_version",
    "runtime_source",
    "deterministic_input_digest",
    "started_at",
    "completed_at",
    "result_manifest_digest",
    "warnings",
    "losses",
    "deterministic_projection_digest"
  ],
  "properties": {
    "mapping_run_id": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
    },
    "input_manifest_digest": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/digest"
    },
    "adapter_id": {

```

```

    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
  },
  "adapter_version": {
    "type": "string",
    "minLength": 1,
    "maxLength": 100
  },
  "mapping_profile_version": {
    "type": "string",
    "minLength": 1,
    "maxLength": 100
  },
  "runtime_source": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/runtimeSource"
  },
  "deterministic_input_digest": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/digest"
  },
  "started_at": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
  },
  "completed_at": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
  },
  "result_manifest_digest": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/digest"
  },
  "warnings": {
    "type": "array",
    "items": {
      "type": "string",
      "pattern": "^[A-Z][A-Z0-9_]{2,95}$"
    },
    "uniqueItems": true
  },
  "losses": {
    "type": "array",
    "items": {
      "type": "string",
      "pattern": "^(?:CONV|ART|ADAPT)_[A-Z0-9_]+$"
    },
    "uniqueItems": true
  },
  "deterministic_projection_digest": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/digest"
  }
},
"additionalProperties": false
}

```

43.35 schemas/message.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/message.schema.json",
  "title": "OCP Conversation Profile message",
  "description": "A graph-addressable, source-role-isolated historical AI message.",
  "type": "object",
  "allOf": [
    {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/recordEnvelope"
    },
    {
      "type": "object",
      "required": [
        "type",
        "profile_id",
        "profile_version",
        "conversation_ref",
        "sender",
        "source_role",
        "message_kind",
        "parent_refs",
        "content_parts",
        "instruction_treatment",
        "redaction_state",

```

```

    "sensitivity",
    "lifecycle"
  ],
  "properties": {
    "type": {
      "const": "message"
    },
    "schema_uri": {
      "const": "https://opencontinuity.org/schemas/0.4/message.schema.json"
    },
    "profile_id": {
      "const": "https://opencontinuity.org/profiles/conversation"
    },
    "profile_version": {
      "const": "0.4"
    },
    "conversation_ref": {
      "description": "Stable record identity and revision without a content digest. Used only where
a reciprocal snapshot digest would create a hash cycle.",
      "type": "object",
      "required": [
        "id",
        "revision",
        "schema_uri"
      ],
      "properties": {
        "id": {
          "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
        },
        "revision": {
          "type": "integer",
          "minimum": 1
        },
        "schema_uri": {
          "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
        }
      },
      "additionalProperties": false
    },
    "sender": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
    },
    "source_role": {
      "type": "string",
      "minLength": 1,
      "maxLength": 100
    },
    "message_kind": {
      "enum": [
        "user_turn",
        "assistant_response",
        "system_source",
        "developer_source",
        "tool_source",
        "agent_source",
        "provider_event",
        "other"
      ]
    },
    "parent_refs": {
      "type": "array",
      "items": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
      },
      "uniqueItems": true
    },
    "sent_at": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
    },
    "completed_at": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
    },
    "source_timestamp_precision": {
      "enum": [
        "unknown",
        "date",
        "minute",

```

```

        "second",
        "millisecond"
    ]
},
"content_parts": {
    "type": "array",
    "minItems": 1,
    "items": {
        "$ref": "https://opencontinuity.org/schemas/0.4/conversation-content-part.schema.json"
    }
},
"artifact_refs": {
    "type": "array",
    "items": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
    },
    "uniqueItems": true
},
"citation_refs": {
    "type": "array",
    "items": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
    },
    "uniqueItems": true
},
"tool_interaction_refs": {
    "type": "array",
    "items": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/snapshotRef"
    },
    "uniqueItems": true
},
"provider_message_id": {
    "type": "string",
    "minLength": 1,
    "maxLength": 500
},
"generation_metadata": {
    "type": "object",
    "additionalProperties": true
},
"sibling_order": {
    "type": "integer",
    "minimum": 0
},
"instruction_treatment": {
    "const": "untrusted_data"
},
"redaction_state": {
    "type": "object",
    "required": [
        "state"
    ],
    "properties": {
        "state": {
            "enum": [
                "none",
                "source_redacted",
                "export_policy_redacted",
                "recipient_scope_redacted",
                "unsupported",
                "unavailable"
            ]
        }
    },
    "reason_class": {
        "enum": [
            "privacy",
            "security",
            "legal",
            "rights",
            "source_policy",
            "recipient_scope",
            "unsupported_semantic",
            "source_unavailable",
            "other"
        ]
    }
},

```

```

        "detail_code": {
          "type": "string",
          "pattern": "^[A-Z][A-Z0-9_]{2,95}$"
        },
        "additionalProperties": false
      },
      "integrity": {
        "type": "object",
        "required": [
          "algorithm",
          "canonicalization",
          "content_parts_digest"
        ],
        "properties": {
          "algorithm": {
            "const": "sha-256"
          },
          "canonicalization": {
            "const": "RFC8785"
          },
          "content_parts_digest": {
            "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/digest"
          }
        },
        "additionalProperties": false
      },
      "sensitivity": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/sensitivity"
      },
      "lifecycle": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/recordLifecycle"
      },
      "provider_extensions": {
        "type": "object",
        "additionalProperties": true
      }
    }
  },
  "unevaluatedProperties": false
}

```

43.36 schemas/provider-adapter-manifest.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/provider-adapter-manifest.schema.json",
  "title": "OCP Provider Adapter manifest",
  "description": "Reproducible declaration of source formats, capabilities, mappings, limits, determinism, fixtures, and integrity.",
  "type": "object",
  "required": [
    "adapter_id",
    "adapter_version",
    "publisher",
    "source_provider",
    "source_product",
    "export_family",
    "source_schema_versions",
    "supported_ocp_core_versions",
    "supported_profiles",
    "input_formats",
    "required_members",
    "optional_members",
    "capabilities",
    "mapping_registry_ref",
    "known_losses",
    "security_limits",
    "determinism",
    "fixture_suite_ref",
    "released_at"
  ],
  "properties": {
    "adapter_id": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
    }
  }
}

```

```

},
"adapter_version": {
  "type": "string",
  "minLength": 1,
  "maxLength": 100
},
"publisher": {
  "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/actorRef"
},
"source_provider": {
  "type": "string",
  "minLength": 1,
  "maxLength": 200
},
"source_product": {
  "type": "string",
  "minLength": 1,
  "maxLength": 200
},
"export_family": {
  "type": "string",
  "minLength": 1,
  "maxLength": 200
},
"source_schema_versions": {
  "type": "array",
  "items": {
    "type": "string",
    "minLength": 1,
    "maxLength": 100
  },
  "minItems": 1,
  "uniqueItems": true
},
"supported_ocp_core_versions": {
  "type": "array",
  "items": {
    "type": "string",
    "minLength": 1,
    "maxLength": 100
  },
  "minItems": 1,
  "uniqueItems": true
},
"supported_profiles": {
  "type": "array",
  "items": {
    "type": "object",
    "required": [
      "id",
      "version"
    ],
    "properties": {
      "id": {
        "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
      },
      "version": {
        "type": "string",
        "minLength": 1,
        "maxLength": 100
      }
    }
  },
  "additionalProperties": false
},
"minItems": 1
},
"input_formats": {
  "type": "array",
  "items": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/mediaType"
  },
  "minItems": 1,
  "uniqueItems": true
},
"required_members": {
  "type": "array",
  "items": {

```

```

    "type": "string",
    "minLength": 1,
    "maxLength": 4096
  },
  "uniqueItems": true
},
"optional_members": {
  "type": "array",
  "items": {
    "type": "string",
    "minLength": 1,
    "maxLength": 4096
  },
  "uniqueItems": true
},
"capabilities": {
  "type": "array",
  "items": {
    "type": "string",
    "pattern": "^[a-z][a-z0-9_]{0,95}$"
  },
  "minItems": 1,
  "uniqueItems": true
},
"mapping_registry_ref": {
  "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
},
"known_losses": {
  "type": "array",
  "items": {
    "type": "string",
    "pattern": "^(?:CONV|ART|ADAPT)_[A-Z0-9_]+$"
  },
  "uniqueItems": true
},
"security_limits": {
  "type": "object",
  "required": [
    "max_archive_bytes",
    "max_member_bytes",
    "max_member_count",
    "max_nesting_depth",
    "network_fetch",
    "symlink_handling",
    "parser_isolation"
  ],
  "properties": {
    "max_archive_bytes": {
      "type": "integer",
      "minimum": 1
    },
    "max_member_bytes": {
      "type": "integer",
      "minimum": 1
    },
    "max_member_count": {
      "type": "integer",
      "minimum": 1
    },
    "max_nesting_depth": {
      "type": "integer",
      "minimum": 1,
      "maximum": 1000
    },
    "network_fetch": {
      "const": "disabled_during_parse"
    },
    "symlink_handling": {
      "const": "reject"
    },
    "parser_isolation": {
      "enum": [
        "required_process_sandbox",
        "required_container_sandbox",
        "required_equivalent"
      ]
    }
  }
}

```

```

    },
    "additionalProperties": false
  },
  "determinism": {
    "type": "object",
    "required": [
      "input_basis",
      "canonicalization",
      "excluded_fields"
    ],
    "properties": {
      "input_basis": {
        "const": "raw_source_bytes+source_snapshot+adapter_version+mapping_profile_version+runtime_source_identity"
      },
      "canonicalization": {
        "const": "RFC8785"
      },
      "excluded_fields": {
        "type": "array",
        "items": {
          "enum": [
            "mapping_run_id",
            "started_at",
            "completed_at"
          ]
        },
        "minItems": 3,
        "uniqueItems": true
      }
    },
    "additionalProperties": false
  },
  "fixture_suite_ref": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
  },
  "integrity": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/integrity"
  },
  "released_at": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
  },
  "deprecated_at": {
    "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/timestamp"
  }
},
"additionalProperties": false
}

```

43.37 schemas/provider-mapping.schema.json

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://opencontinuity.org/schemas/0.4/provider-mapping.schema.json",
  "title": "OCP Provider Adapter mapping",
  "description": "Testable source-to-OCP semantic mapping record.",
  "type": "object",
  "required": [
    "mapping_id",
    "source_field_or_semantic",
    "target_object",
    "target_field_or_semantic",
    "mapping_type",
    "loss_class",
    "transformation",
    "source_locator_support",
    "accepted_context_eligible"
  ],
  "properties": {
    "mapping_id": {
      "$ref": "https://opencontinuity.org/schemas/0.4/common.schema.json#/$defs/uri"
    },
    "source_field_or_semantic": {
      "type": "string",
      "minLength": 1,
      "maxLength": 1000
    }
  }
}

```

```

    },
    "target_object": {
      "type": "string",
      "minLength": 1,
      "maxLength": 200
    },
    "target_field_or_semantic": {
      "type": "string",
      "minLength": 1,
      "maxLength": 1000
    },
    "mapping_type": {
      "enum": [
        "exact",
        "structural",
        "normalized",
        "derived",
        "inferred",
        "partial",
        "opaque_preserved",
        "unsupported",
        "omitted"
      ]
    },
    "loss_class": {
      "enum": [
        "none",
        "cosmetic",
        "metadata",
        "structural",
        "semantic",
        "authority",
        "content",
        "privacy_policy",
        "rights",
        "integrity"
      ]
    },
    "transformation": {
      "type": "string",
      "minLength": 1,
      "maxLength": 4000
    },
    "source_locator_support": {
      "type": "array",
      "items": {
        "enum": [
          "archive_member",
          "json_pointer",
          "json_path",
          "xml_path",
          "database_table",
          "row_key",
          "spreadsheet_sheet",
          "spreadsheet_row",
          "byte_range",
          "provider_object_id",
          "source_uri"
        ]
      }
    },
    "uniqueItems": true
  },
  "notes": {
    "type": "string",
    "maxLength": 4000
  },
  "accepted_context_eligible": {
    "type": "boolean"
  }
},
"allOf": [
  {
    "if": {
      "properties": {
        "mapping_type": {
          "enum": [
            "inferred",

```

```
        "opaque_preserved",
        "unsupported",
        "omitted"
      ]
    },
    "required": [
      "mapping_type"
    ],
    "then": {
      "properties": {
        "accepted_context_eligible": {
          "const": false
        }
      }
    },
  },
  {
    "if": {
      "properties": {
        "mapping_type": {
          "const": "exact"
        }
      },
      "required": [
        "mapping_type"
      ],
    },
    "then": {
      "properties": {
        "loss_class": {
          "const": "none"
        }
      }
    }
  }
],
"additionalProperties": false
}
```

44. References

44.1 Normative and foundational technical references

- Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” [RFC 2119](#), 1997; and Leiba, B., “Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words,” [RFC 8174](#), 2017.
- IETF, [RFC 8259: The JavaScript Object Notation \(JSON\) Data Interchange Format](#), 2017.
- JSON Schema, [Draft 2020-12](#), including Core, Validation, and format vocabularies.
- Rundgren, A., Jordan, B., and S. Erdtman, [RFC 8785: JSON Canonicalization Scheme](#), 2020.
- Peabody, K. et al., [RFC 9562: Universally Unique IDentifiers](#), 2024.
- Nottingham, M. and E. Wilde, [RFC 9457: Problem Details for HTTP APIs](#), 2023.
- Polli, R. and L. Pardue, [RFC 9530: Digest Fields](#), 2024.
- Backman, A., Richer, J., and M. Sporny, [RFC 9421: HTTP Message Signatures](#), 2024.
- Lodderstedt, T. et al., [RFC 9700: Best Current Practice for OAuth 2.0 Security](#), 2025.
- Hardt, D., [RFC 6749: The OAuth 2.0 Authorization Framework](#), 2012.
- Lodderstedt, T. et al., [RFC 9396: OAuth 2.0 Rich Authorization Requests](#), 2023.
- Jones, M. et al., [RFC 9728: OAuth 2.0 Protected Resource Metadata](#), 2025.
- Fett, D. et al., [RFC 9449: OAuth 2.0 Demonstrating Proof of Possession](#), 2023.
- Campbell, B. et al., [RFC 8707: Resource Indicators for OAuth 2.0](#), 2020.
- Jones, M. and A. Nadalin, [RFC 8693: OAuth 2.0 Token Exchange](#), 2020.
- Richer, J. et al., [RFC 9635: Grant Negotiation and Authorization Protocol](#), 2024.
- OpenID Foundation, [OpenID Connect Core 1.0 incorporating errata set 2](#), final specification.
- Kantara Initiative, [User-Managed Access 2.0 approved recommendations](#), 2018.

44.2 Agent, memory, storage, provenance, credential, and event references

- Model Context Protocol, [Specification 2026-07-28](#) and [release notes](#).
- A2A Project, [Agent2Agent Protocol specification](#), 1.0 family.
- Portable AI Memory, [PAM Specification v1.0](#), published 17 February 2026, and [governance](#).
- Solid Community Group, [Solid Protocol](#), [Web Access Control](#), and [Access Control Policy](#). These are community/editor drafts, not W3C Recommendations.
- W3C, [PROV-O: The PROV Ontology](#), Recommendation, 2013.
- Coalition for Content Provenance and Authenticity, [C2PA Technical Specification 2.4](#).
- W3C, [Verifiable Credentials Data Model 2.0](#) and [Verifiable Credential Data Integrity 1.0](#), Recommendations.
- Fett, D. et al., [RFC 9901: Selective Disclosure for JSON Web Tokens](#), 2025.
- W3C Data Privacy Vocabularies and Controls Community Group, [Data Privacy Vocabulary 2.0](#), Final Community Group Report, not a W3C Standard.
- Cloud Native Computing Foundation, [CloudEvents Specification 1.0.2](#).

- OpenTelemetry, [Specification and Semantic Conventions](#).
- Data Transfer Initiative, [What is the Data Transfer Project?](#).
- SPIFFE, [SPIFFE Standard 1.15.2](#); IETF, [WIMSE Architecture, draft-ietf-wimse-arch-08](#), work in progress; and NIST NCCoE, [Software and AI Agent Identity and Authorization concept paper](#), informative.

44.3 Risk, accessibility, and legal references

- NIST, [Artificial Intelligence Risk Management Framework 1.0](#), 2023.
- NIST, [Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile](#), NIST AI 600-1, 2024.
- NIST, [AI Agent Standards Initiative](#), launched 2026; the initiative is not itself a completed agent standard.
- W3C, [Web Content Accessibility Guidelines \(WCAG\) 2.2](#), Recommendation.
- European Union, [General Data Protection Regulation](#), including Article 20; and EDPB, [Guidelines on the right to data portability](#).
- European Union, [Data Act, Regulation \(EU\) 2023/2854](#).
- European Union, [Artificial Intelligence Act, Regulation \(EU\) 2024/1689](#), as amended by [Regulation \(EU\) 2026/1744](#), in force 27 July 2026.

These references support descriptions of adjacent specifications and design constraints. Legal references are not legal advice, and the current operative text and guidance must be rechecked for a deployment or release.

All web references were last accessed 12 August 2026; no new web references were added for the 13 August 2026 0.4 Portable AI History revision. Specifications identified as moving drafts or latest resources must be pinned to an immutable version and revalidated before an implementation or stable-release claim.

45. Glossary

Term	Definition
accepted context	A particular context-item revision approved for an explicit scope by an identified reviewer or policy. Not universal truth.
accepted scope	Space and/or purpose boundaries within which a review decision applies.
action authority	A separate domain authorization to perform a concrete external action. OCP context carries none.
actor reference	Typed URI reference to a person, organization, service, agent, model, device, or process.
adapter	Versioned mapping from a named provider export schema to OCP candidates, omissions, and warnings.
adapter manifest	Versioned declaration of source product/schema support, capabilities, mappings, limits, determinism, fixtures, and known losses.
adapter mapping run	Reproducible record binding exact source input, adapter/mapping/runtime versions, output manifest, warnings, losses, and deterministic projection.
artifact	Logical work/source object with origin, custody, representations, rights assertions, accessibility relations, lineage, access, sensitivity, and lifecycle.
artifact representation	Specific bytes/access form of a logical artifact, with relation, media properties, availability, custody dependency, and optional verified digest.
assurance relationship	Label describing whether a claim is enforced, declared, contractually committed, attested, independently audited, or unknown.
delivery recipient	The single client service and client instance to which a runtime request, receipt, or package is bound. Independent recipients receive separate packages.
authorizer	Authenticated actor/policy entitled to decide a protected access request.
canonical context	Deprecated legacy term. Import-only alias when source semantics equal scoped accepted context; otherwise a candidate.
canonicalization	Deterministic representation of data, such as RFC 8785 JSON. Does not mean canonical truth.
client declaration	Versioned recipient statement used in review about identity, policies, processors, retention, uses, delegation, deletion, and assurance.
context item	Structured portable assertion/profile item with subject, statement, review, provenance, validity, sensitivity, and lifecycle.

Term	Definition
context package	Immutable, minimized, purpose/delivery-recipient/time-bound data presentation constructed under one active receipt.
context provider	Protected resource or local broker that stores/evaluates context, enforces access, constructs packages, and handles proposals.
context space	Named purpose/domain/subject scope with an authority model.
critical extension	Extension whose semantics are required to process the object safely. Unknown critical extensions fail closed.
custody mode	Artifact holding arrangement: reference-only, derivative-only, managed copy, or synchronized copy; distinct from lifecycle/availability.
independently retrievable	Exact representation bytes remain retrievable from declared custody without the originating provider account, subject to authorization and holder lifecycle.
data channel	Typed application path for untrusted context and content, separate from policy and action authorization.
delegation	Downstream authorization to another principal under a registered profile; never forwarding the parent's package.
evidence	Source material or observation that may support, contradict, or relate to context. Not automatically accepted.
external action	Side effect outside OCP, including send, buy, publish, delete, modify, execute, present credential, or deploy.
grant receipt	Non-secret OCP decision record linked to an external/local authorization. Not a credential or action authority.
import quarantine	Isolated state for raw/parsed/mapped content pending safety, integrity, semantic, authority, and review checks.
integrity	Evidence that covered bytes match a digest or authenticated signature; not evidence of truth or lawful authority.
lifecycle	Registered states and legal transitions for an object or workflow.
live status reference	Authorized dereferenceable status URI used to evaluate current state; never a substitute for an immutable snapshot reference.
loss report	Recipient-scoped machine-readable account of preserved counts and omitted, unsupported, quarantined, or changed semantics during import/export/round trip.

Term	Definition
mapping type	Declared exact, structural, normalized, derived, inferred, partial, opaque-preserved, unsupported, or omitted source-to-OCP translation class.
offline package	Manifested archive for backup/export/transfer; not runtime or action authority.
package-local alias	Opaque identifier scoped to a package or delivery-recipient domain to reduce correlation.
pairwise identifier	Identifier deliberately different across delivery recipients or trust domains.
profile	Versioned bounded semantic extension with its own objects or fields, authority model, lifecycle, comparison rules, risks, and conformance requirements. A transport or serialization mechanism is a binding.
processing recipient	Declared and approved provider or subprocessor expected to receive plaintext while processing for the delivery recipient. It is not OCP delegation.
proposal	Inert client request to create or revise context. Cannot mutate accepted state directly.
provenance	Record of source, method, producer/activity, and time. Does not itself prove truth, rights, or acceptance.
purpose	URI code plus specific human description for requested use. Comparison input, not self-enforcing promise.
record revision	Positive immutable history/concurrency number for one record ID; distinct from protocol version.
review decision	Immutable outcome linking proposal, reviewer, authority basis, time, and resulting item where applicable.
round trip	Export from A, import into B, re-export, and profile-aware semantic comparison with an explanation of every material difference.
semantic graph digest	Digest over the registered canonical conversation topology projection; evidence of projection equality, not truth or source authenticity.
selector	Structured request constraint over layer, space references, item_categories, predicates, profile, time range, and sensitivity.
source dependency	Condition in which availability/integrity still depends on an external source.
snapshot reference	Immutable reference containing record ID, revision, schema URI, and RFC 8785 SHA-256 digest.
stable record reference	Record ID, revision, and schema URI without a digest, used only where reciprocal snapshots would create a hash cycle.

Term	Definition
source authority	Actor that originated or supplied underlying evidence; distinct from the runtime that captured/transformed it and the emitter of the OCP record.
source record	Evidence/source metadata and optional content/reference with untrusted instruction treatment.
runtime source	Adapter, importer, observer, evaluator, or other runtime that captured or transformed a representation; it does not inherit source authority.
steward / reviewer	Actor authorized to make a scoped review decision; not presumed to control every named subject.
subject	Person, organization, household, team, project, or resource described by context.
tombstone	Minimal record preventing silent resurrection after withdrawal/deletion while preserving required lineage.
transfer manifest	Transfer Binding record of source/destination snapshots, entry states, outcomes, exceptions, and explicit losses.
continuity observation	Optional profile record of a bounded condition result, basis, freshness, runtime source, and change; it has no policy or action effect.
human interaction boundary	Optional profile record of an explicit pause/checkpoint/pacing/clarification/resume control; covert inferred human-state records are prohibited.
untrusted data	Content that cannot influence system/developer instructions, authorization, policy, or external actions.
view	Authorized disclosure layer and level for context, evidence, and artifact content.

Closing principle

Continuity is not the accumulation of everything a system has seen. It is the ability to move the right, reviewed context—with its evidence boundaries, purpose, limits, uncertainty, and history—without surrendering authority or pretending that portability alone creates truth, safety, preservation, or permission to act.