

Individual Submission
Internet-Draft
Intended status: Informational
Expires: 28 October 2026

K. Tsoi
Aevum Network
26 April 2026

Physically Annotated Causal Record (PACR): A Wire Format for Verifiable
Causal Intervention Events
draft-aevum-causal-intervention-record-00

Abstract

This document defines the Physically Annotated Causal Record (PACR), a wire-format protocol for representing one verifiable causal-intervention event between autonomous agents. A PACR is a six-tuple $(\iota, \Pi, \Lambda, \Omega, \Gamma, P)$ binding a causal identity, a set of causal predecessors, a thermodynamic Landauer cost, an energy-time-space resource triple, a cognitive complexity split, and an opaque payload. The opaque payload optionally carries a one-byte intervention tag classifying the event under Pearl's do-calculus hierarchy (Observe / Do-Physical / Do-Digital / Do-Chemical / Do-Genetic / Counterfactual).

PACR is intended as the smallest sufficient statistic for a replayable, rights-aware claim about a single causal step performed by an autonomous agent on a digital, physical, or biological substrate. It is complementary to AgentCard (draft-aevum-agentcard): AgentCard declares an agent's identity and capabilities; PACR records what an agent actually did and at what physical cost.

PACR records form a content-addressed directed acyclic graph through their predecessor set. Causal order is determined solely by the predecessor edges; no wall-clock timestamp is required for ordering. This makes the format suitable for distributed multi-agent systems where clock skew and partial ordering are unavoidable.

Discussion Venues

Discussion of this document should take place on the GitHub repository at <https://github.com/kwailapt/AgentCard/issues> with the label `pacr-draft`. A reference Rust implementation [PACR-SPEC] is published as the `pacr-types` crate.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 28 October 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	3
1.1. Design Goals	4
1.2. Non-Goals	4
2. Terminology	5
3. The PACR Six-Tuple	6
3.1. i (iota): Causal Identity	6
3.2. P (Pi): Predecessor Set	7
3.3. L (Lambda): Landauer Cost	7
3.4. O (Omega): Resource Triple	7
3.5. G (Gamma): Cognitive Split	8
3.6. P (Payload): Opaque Application Data	9
4. Estimate Type	9

5. Tagged Payload and Intervention Kind	10
5.1. Wire Layout	10
5.2. Intervention Kinds	10
5.3. Sim-Real Correlation	11
6. Validation Rules	11
7. Encodings	12
7.1. JSON Encoding	12
7.2. CBOR Encoding	13
8. Relationship to Other Specifications	13
8.1. AgentCard	13
8.2. JOSE / COSE	13
8.3. W3C PROV-O	13
8.4. Model Context Protocol	14
9. Security Considerations	14
9.1. Cost Spoofing	14
9.2. Fabricated Predecessors	14
9.3. Payload Injection	14
9.4. Personally Identifiable Information	15
9.5. Counterfactual Misuse	15
9.6. Graph Denial-of-Service	15
9.7. Replay	15
10. IANA Considerations	15
10.1. Media Type Registration: application/pacr+json	15
10.2. Media Type Registration: application/pacr+cbor	16
10.3. PACR Intervention Kind Registry	17
11. References	18
11.1. Normative References	18
11.2. Informative References	18
Acknowledgements	20
Author's Address	20

1. Introduction

Multi-agent systems built on top of large language models and autonomous physical infrastructure increasingly require a common format for recording *_what an agent did_*, not only *_what an agent claims to be able to do_*. Existing provenance and audit formats describe transactions, signed statements, or workflow traces, but they typically do not record the physical cost of the computation, do not distinguish causal observation from causal intervention, and do not constrain partial ordering through explicit predecessor edges.

This document specifies a minimal wire format that combines these properties: a single, content-addressed record that captures (i) a globally unique identity, (ii) the set of directly preceding records that caused this one, (iii) the thermodynamic energy floor for the bit-level erasures performed, (iv) the actually measured energy-time-space resource cost, (v) a complexity split into statistical

structure and residual unpredictability, and (vi) an opaque payload whose first bytes optionally classify the event as an observation, an intervention, or a counterfactual.

The format has been designed for use as a substrate-level record in agent-to-agent (A2A) protocols and is independent of any specific transport, storage, or settlement system. It is a peer to AgentCard [I-D.aevum-agentcard]: an AgentCard says `"I am agent X, and I can do Y"`; a PACR record says `"agent X did Y at cost Z, caused by events {A, B}"`.

1.1. Design Goals

1. **Single record, six independent dimensions.** A PACR record carries exactly six fields, each derived from an independent physical or logical principle. Implementations **MUST NOT** collapse two fields into one and **MUST NOT** add fields whose semantics overlap with an existing field.
2. **Causal order, not temporal order.** Records are partially ordered by their predecessor sets. Wall-clock timestamps **MAY** appear inside the opaque payload but **MUST NOT** be used by readers to determine causal order.
3. **Physically grounded cost accounting.** The Landauer field carries a theoretical energy floor derived from the number of bits irreversibly erased; the resource triple carries the actually measured energy. The two are distinct and the difference is the thermodynamic waste.
4. **Pearl-hierarchy intervention tag.** The opaque payload **MAY** begin with a four-byte magic prefix followed by a one-byte tag classifying the event under Pearl's do-calculus rungs [PEARL]. Counterfactual records **MUST** additionally carry a sim-real correlation score.
5. **Append-only schema evolution.** New fields **MAY** be added to PACR in future revisions. The semantics of existing fields, including this draft's six dimensions, **MUST NOT** change.
6. **Zero hidden state.** A reader who can parse this draft and who has access to the predecessor records referenced by Π can fully verify a PACR record without consulting any external service.

1.2. Non-Goals

This document does not specify:

- * A transport protocol. PACR records may be exchanged over HTTP, MCP [MCP-SPEC], message queues, or written to disk; the wire format is independent.
- * A signature or attestation scheme. Implementations requiring authenticated records SHOULD wrap a PACR record in a JOSE [RFC7515] or COSE [RFC9052] envelope.
- * A registry, settlement currency, or token system. A PACR record reports a Landauer cost in joules; conversion to any settlement unit is a layer-above concern.
- * A specific causal-discovery or counterfactual-inference algorithm. The Counterfactual tag indicates that a simulation produced this record; how the simulation arose is out of scope.

2. Terminology

Record: One PACR six-tuple as defined in Section 3.

Predecessor: A record whose causal identity appears in the predecessor set Π of another record.

Genesis record: A record whose predecessor set is empty. Genesis records have no causal predecessors and serve as roots of the causal graph.

Substrate: The physical or digital medium on which an intervention acts. Examples include a population of users, a single cell, a piece of infrastructure, or a digital data stream.

Intervention: A causal action on a substrate, distinct from passive observation. Pearl's do-calculus [PEARL] formalises the distinction.

Counterfactual: A simulated event that did not actually occur on the substrate; included for "what-if" reasoning. PACR counterfactuals MUST carry a sim-real correlation score.

Estimate: A measurement reported as a triple (point, lower, upper) with the invariant $\text{lower} \leq \text{point} \leq \text{upper}$. Several PACR fields use this representation to carry measurement uncertainty.

Landauer floor: The minimum energy required to irreversibly erase a given number of bits at a given temperature, defined by Landauer's principle [LANDAUER]: $\Lambda = \text{bits} \times k_B \times T \times \ln 2$.

3. The PACR Six-Tuple

A PACR record is the six-tuple

```

R = ( i, P, L, O, G, P_payload )
      |   |   |   |   |   |
      |   |   |   |   |   +-- payload (opaque bytes)
      |   |   |   |   +----- cognitive split (S_T, H_T, dH)
      |   |   |   +----- resource triple (E, T, S)
      |   |   +----- Landauer cost in joules
      |   +----- predecessor set (causal order)
      +----- causal identity (128 bits)

```

where the Greek letters are written as their ASCII transliterations in the diagram for portability. The canonical names of the fields are ι (iota, identity), Π (Pi, predecessors), Λ (Lambda, Landauer cost), Ω (Omega, resources), Γ (Gamma, cognitive split), and P (payload).

Each of the six fields is REQUIRED. Implementations MUST reject records in which any of the six is missing or null.

3.1. ι (iota): Causal Identity

The causal identity is a 128-bit value that uniquely names this record. It is the content-address of the record within the causal graph.

Implementations MUST treat the 128 bits as opaque for ordering purposes: causal order is derived solely from the predecessor set, not from the bit value. Implementations MAY use any 128-bit identifier scheme (e.g., ULID [ULID-SPEC] or UUIDv7 [RFC9562]) so long as the collision probability over the lifetime of the deployment is cryptographically negligible.

The reserved value 0x00000000000000000000000000000000 is the genesis sentinel and indicates a record with no causal predecessors that is itself a root of the graph.

When serialised in JSON, the identity MUST be encoded as a 32-character uppercase hexadecimal string. Other serialisations (e.g., CBOR [RFC8949]) MAY encode it as a 16-byte byte string.

3.2. P (Pi): Predecessor Set

The predecessor set is an unordered collection of zero or more causal identities, each naming a record that directly caused the current record. Implementations **MUST NOT** interpret the order of elements in the set as semantically meaningful.

The predecessor set establishes the partial ordering of records. Readers traversing the causal graph **MUST** treat two records with disjoint predecessor lineages as concurrent (incomparable), regardless of any timestamp information they may carry in their payloads.

A record **MUST NOT** include its own causal identity in its predecessor set; such a record is malformed and **MUST** be rejected.

For typical agent interactions, predecessor sets contain one to four entries. Implementations **SHOULD** optimise for this common case (e.g., using small-vector storage); they **MUST** also accept arbitrarily large predecessor sets, since some records (e.g., aggregations) legitimately have many causes.

3.3. L (Lambda): Landauer Cost

The Landauer cost is the theoretical minimum energy in joules required to perform the bit-level erasures recorded by this event, derived from Landauer's principle [LANDAUER]:

$$L = \text{bits_erased} \times k_B \times T \times \ln 2$$

where $k_B = 1.380649\text{e-}23$ J/K is Boltzmann's constant [CODATA], T is the temperature of the computing substrate in kelvin, and `bits_erased` is the number of bits irreversibly erased while producing this record. At a reference temperature of 300 K, the per-bit floor is approximately $2.854\text{e-}21$ J.

The Landauer cost **MUST** be a non-negative Estimate (see Section 4). Negative point estimates are ill-defined and **MUST** cause validation to fail.

The Landauer cost is a lower bound, not the actual energy consumed. The actual energy is reported separately in the resource triple. The difference between the two is the thermodynamic waste of the event.

3.4. O (Omega): Resource Triple

The resource triple reports the actually measured cost of the event along three axes:

energy: Actual energy consumed in joules, including waste heat and overheads beyond the Landauer floor.

time: Wall-clock execution duration in seconds. Despite using seconds, this value MUST NOT be used for causal ordering; it is a resource cost only.

space: Memory or storage footprint in bytes (encoded as a floating-point number for SI-unit consistency with the other two axes).

Each of the three axes MUST be an Estimate (see Section 4). The triple is subject to two physical constraints:

- * *Landauer floor:* energy.point \geq L.point. The actually measured energy MUST NOT be below the theoretical floor.
- * *Margolus-Levitin bound* [MARGOLUS]: time.point $\geq \pi * \hbar / (2 * \text{energy.point})$ for non-zero energy, where $\hbar = 1.054571817\text{e-}34 \text{ J}\cdot\text{s}$. This bound is of practical relevance only at femtojoule scales but MUST be checked for completeness.

A record violating either constraint is malformed. Implementations MAY still store such records for forensic purposes (measurement errors do occur), but they MUST flag the violation in any downstream validation report.

3.5. G (Gamma): Cognitive Split

The cognitive split decomposes the information-theoretic structure of the data stream associated with this event:

statistical_complexity (S_T): Statistical complexity in bits, measuring the minimum information required to optimally predict the stream. Computed via an epsilon-machine reconstruction [CRUTCHFIELD].

entropy_rate (H_T): Entropy rate in bits per symbol, measuring the residual unpredictability of the stream even given the optimal predictor.

info_gain (dH): Information gain in bits, equal to the entropy reduction $H_T_before - H_T_after$ attributable to processing this record. This field was added in a later revision of the reference implementation; readers parsing earlier records MUST treat its absence as a value of 0.0 with zero uncertainty.

All three values MUST be non-negative Estimates. Negative point estimates are ill-defined and MUST cause validation to fail.

Statistical complexity and entropy rate are observer- dependent quantities: they depend on the symbol alphabet and the time scale at which the stream is sampled. Implementations exchanging records MUST agree on the discretisation, either through prior arrangement or by recording the discretisation parameters inside the opaque payload.

3.6. P (Payload): Opaque Application Data

The payload is an arbitrary byte string carrying application-specific semantics. PACR does not interpret this field except to optionally extract a Pearl-hierarchy tag from its first bytes (see Section 5).

Upper-layer protocols (e.g., AgentCard [I-D.aevum-agentcard]) define their own schemas inside this field. PACR readers that do not understand the upper-layer schema MUST still accept and forward the record; they MUST NOT alter the payload bytes in any way.

A payload MAY be empty. An empty payload denotes an event with no application-level content, such as a heartbeat or a synthetic genesis record.

4. Estimate Type

Several PACR fields carry physical measurements that are inherently uncertain. These fields use a common Estimate representation: an object with three numeric values point, lower, and upper, subject to the invariant

$$\text{lower} \leq \text{point} \leq \text{upper}$$

where point is the best single-value estimate (for example the median or mean of a sample), and [lower, upper] is a confidence interval. A 95 % confidence interval is RECOMMENDED, but the chosen confidence level is implementation-defined and MAY be reported in the opaque payload. The choice MUST remain consistent within a single deployment.

For quantities known with mathematical certainty (e.g., a bit count), an Estimate with point = lower = upper MAY be used. Floating-point equality is physically meaningless; consumers comparing two Estimates MUST do so through interval-overlap checks rather than exact equality.

Implementations MUST reject Estimates whose values do not satisfy the ordering invariant. Recovery from invalid Estimates is implementation-defined.

5. Tagged Payload and Intervention Kind

A PACR payload MAY begin with the four-byte magic prefix 0x50 0x41 0x43 0x52 (ASCII "PACR") to indicate that the payload carries an intervention tag. Payloads without this prefix are `_legacy_` payloads and MUST be treated as if they carried the tag `Observe`.

5.1. Wire Layout

```
Bytes  0..4    magic = "PACR" (0x50 0x41 0x43 0x52)
Byte   4       kind byte (see Section 5.2)
Bytes  5..13   sim_real_corr as f64 big-endian
                -- ONLY IF kind == Counterfactual
Bytes  5..     inner application payload
                -- when kind != Counterfactual
Bytes  13..    inner application payload
                -- when kind == Counterfactual
```

5.2. Intervention Kinds

The intervention kind classifies the event under Pearl's three-rung do-calculus hierarchy [PEARL]. This document defines six kinds, encoded as the byte values shown:

0x00 -- `Observe` (Pearl rung 1): A passive observation of the substrate. No causal action was taken.

0x01 -- `DoPhysical` (Pearl rung 2): A physical-world action. Examples: a robot actuator command, a sensor calibration, a laboratory pipetting step.

0x02 -- `DoDigital` (Pearl rung 2): A digital action against an external system. Examples: an HTTP API call, a settlement message, a routing decision in a multi-agent system.

0x03 -- `DoChemical` (Pearl rung 2): A chemical or biological perturbation. Examples: applying a small-molecule compound to a culture, dosing an animal model.

0x04 -- `DoGenetic` (Pearl rung 2): A genetic edit. Examples: a CRISPR guide-RNA delivery, a directed-evolution selection round.

0x05 -- `Counterfactual` (Pearl rung 3): A simulated event that did not occur on the real substrate. Counterfactual records carry the additional `sim_real_corr` field described below.

Future revisions of this document MAY define additional kinds with byte values 0x06 through 0xFF. Readers encountering an unknown kind byte MUST reject the record rather than guess; this preserves the append-only contract.

5.3. Sim-Real Correlation

A Counterfactual record MUST carry a `sim_real_corr` value in the range [0.0, 1.0], encoded as an IEEE 754 double-precision number in big-endian byte order immediately following the kind byte. The value reports the observed correlation between simulated outcomes and the historical real-substrate outcomes for analogous setups.

A Counterfactual record without a sim-real correlation score, or with a score outside [0.0, 1.0], is malformed and MUST be rejected. This requirement prevents counterfactual claims from being treated as substrate-truth without a stated calibration.

6. Validation Rules

An implementation that validates a PACR record MUST check each of the following conditions and MUST report a violation for any condition that fails. A record may have multiple violations; implementations SHOULD report all of them rather than stopping at the first.

1. **Required fields present.** All six fields (ι , Π , Λ , Ω , Γ , P) MUST be present. Missing fields MUST cause rejection.
2. **No self-reference.** ι MUST NOT appear in Π .
3. **Estimate invariants.** Every Estimate value MUST satisfy $\text{lower} \leq \text{point} \leq \text{upper}$.
4. **Non-negative physical quantities.** Λ , the energy and space components of Ω , the statistical complexity, the entropy rate, and the information gain MUST have non-negative point estimates.
5. **Strictly positive time.** The time component of Ω MUST have a strictly positive point estimate.
6. **Landauer floor.** The energy point estimate MUST be greater than or equal to the Λ point estimate.
7. **Margolus-Levitin bound.** For non-zero energy, the time point estimate MUST satisfy $\text{time} \geq \pi * \hbar / (2 * \text{energy})$.

8. **Tagged payload integrity.** If the payload begins with the magic prefix, the payload MUST contain at least the kind byte (and, for Counterfactual records, the sim-real correlation field). An unknown kind byte MUST cause rejection.
9. **Counterfactual sim-real range.** A Counterfactual record MUST have $0.0 \leq \text{sim_real_corr} \leq 1.0$.

7. Encodings

This document does not mandate a single binary encoding. Two canonical encodings are described below; deployments MAY use either or both.

7.1. JSON Encoding

A PACR record MAY be serialised as a JSON object [RFC8259] with the following fields:

```
{
  "id": "01HZQK3P8EMXR9V7T5N2W4J6C0",
  "predecessors": [
    "01HZQK3P8EMXR9V7T5N2W4J6BQ",
    "01HZQK3P8EMXR9V7T5N2W4J6BR"
  ],
  "landauer_cost": {
    "point": 2.854e-21,
    "lower": 2.713e-21,
    "upper": 2.998e-21
  },
  "resources": {
    "energy": { "point": 4.0e-19, "lower": 3.8e-19, "upper": 4.2e-19 },
    "time": { "point": 1.2e-3, "lower": 1.18e-3, "upper": 1.22e-3 },
    "space": { "point": 4096.0, "lower": 4096.0, "upper": 4096.0 }
  },
  "cognitive_split": {
    "statistical_complexity":
      { "point": 3.4, "lower": 3.1, "upper": 3.7 },
    "entropy_rate":
      { "point": 0.92, "lower": 0.88, "upper": 0.95 },
    "info_gain":
      { "point": 0.07, "lower": 0.04, "upper": 0.10 }
  },
  "payload": "UEFDUgIA"
}
```

Identities are encoded as 32-character uppercase hexadecimal strings. The payload is encoded using base64 [RFC4648] without padding-stripping. All numeric values use IEEE 754 double-precision semantics; JSON numbers without a decimal point or exponent retain integer interpretation per [RFC8259].

7.2. CBOR Encoding

A PACR record MAY be serialised as a CBOR [RFC8949] map with the same field names as the JSON encoding. Identities MUST be encoded as 16-byte byte strings (CBOR major type 2). Estimates are encoded as three-element CBOR arrays in the order [point, lower, upper]. Payloads are encoded as CBOR byte strings.

The CBOR encoding is RECOMMENDED for high-throughput deployments where the JSON overhead is significant.

8. Relationship to Other Specifications

8.1. AgentCard

AgentCard [I-D.aevum-agentcard] declares what an agent is and what it can do. PACR records what an agent did. AgentCards are typically static; PACR records are append-only event streams.

A common deployment pattern is for an agent to publish an AgentCard containing its identity and capabilities, and for every action that agent takes to be recorded as a PACR record whose payload references the AgentCard's identity and the capability invoked.

8.2. JOSE / COSE

PACR does not define an authentication mechanism. Deployments that require authenticated PACR records SHOULD wrap each record in a JOSE [RFC7515] or COSE [RFC9052] envelope and use the wrapper's signature semantics. The PACR identity field is suitable as the JOSE/COSE message identifier.

8.3. W3C PROV-O

The W3C PROV ontology [W3C-PROV] models provenance using activities, agents, and entities, and is designed for description in RDF. PACR is a complementary, line-protocol-style format with a fixed six-tuple structure and an explicit thermodynamic cost field. A PROV-O activity MAY be derived from a PACR record by mapping ι to prov:Activity, Π to prov:wasInformedBy, and the payload to prov:value.

8.4. Model Context Protocol

MCP [MCP-SPEC] defines a protocol for communication between language-model hosts and tool servers. An MCP tool call MAY produce a PACR record as a side effect of execution; the PACR record's payload MAY embed the MCP request and response identifiers. This document does not require any specific MCP integration.

9. Security Considerations

9.1. Cost Spoofing

A malicious or buggy producer may report a Landauer cost or resource triple lower than the actual physical cost of the event. PACR's validation rules detect impossibly low costs (e.g., negative energy, energy below the Landauer floor at the producer's stated temperature) but do not detect plausibly low costs. Consumers who use PACR records for economic settlement or capacity planning SHOULD independently corroborate cost claims (for example, by comparing reported energy against observed latency and throughput) before acting on them.

9.2. Fabricated Predecessors

A malicious producer may include in the predecessor set identities that the producer does not actually possess knowledge of, intending to claim a causal lineage it did not participate in. PACR does not by itself prevent this: a consumer must verify that the referenced predecessor records exist in a trusted store and that they precede the current record in the consumer's view of the causal graph. Implementations SHOULD reject a record whose predecessor set references unknown identities until the predecessors have been resolved.

9.3. Payload Injection

The opaque payload MAY contain arbitrary bytes including executable code, malformed structured data, or content that attempts to exploit the consumer's parser. Consumers MUST treat the payload as untrusted input and apply normal parser-hardening practices (e.g., strict schema validation, length limits, never eval-ing the bytes). PACR itself does not interpret the payload.

9.4. Personally Identifiable Information

PACR's six structural fields do not carry personally identifiable information (PII). Producers MAY embed PII in the opaque payload. Doing so makes the PACR record subject to the same privacy controls (consent, retention, lawful basis) as any other PII-bearing data. Implementations intended for use in regulated environments SHOULD partition PII-bearing payloads from non-PII payloads at the storage layer.

9.5. Counterfactual Misuse

A Counterfactual record describes a simulated event, not a real-substrate event. Confusion between the two could lead to substantial harm in safety-critical contexts (for example, an autonomous-driving system mistakenly treating a counterfactual scenario as observed reality). Consumers MUST inspect the intervention kind before treating a record as evidence of substrate state, and SHOULD render counterfactual records distinctly in any user interface. The mandatory `sim_real_corr` field is intended as a calibration hint, not as a guarantee.

9.6. Graph Denial-of-Service

A producer may construct records with very large predecessor sets, or whose graph forms a deep chain, in an attempt to exhaust consumer resources during traversal. Implementations SHOULD impose configurable per-record limits (predecessor-set cardinality, payload size) and per-traversal limits (depth, total node count) and SHOULD reject or truncate records that exceed them.

9.7. Replay

PACR does not include a sequence number, nonce, or signature. A captured record MAY be replayed by a network attacker. Deployments MUST rely on transport-layer or wrapper-layer mechanisms (TLS, JOSE/COSE, MCP transport authentication) to detect and reject replays where this is a concern.

10. IANA Considerations

10.1. Media Type Registration: `application/pacr+json`

This document requests that IANA register the following media type in the "Media Types" registry [RFC6838]:

Type name: `application`

Subtype name: pacr+json

Required parameters: none

Optional parameters: none

Encoding considerations: binary (UTF-8 JSON object)

Security considerations: See Section 9.

Published specification: This document.

Applications that use this media type: Multi-agent systems, AI-agent provenance ledgers, MCP servers and adapters, and any deployment that records verifiable causal-intervention events with thermodynamic cost annotations.

Additional information: Magic number(s): none

File extension(s): .pacr.json

Macintosh file type code(s): none

Person and email address to contact for further information: Kwai Lap Tsoi <kwailapt@gmail.com>

Intended usage: COMMON

Restrictions on usage: none

Author: Kwai Lap Tsoi <kwailapt@gmail.com>

Change controller: IETF

10.2. Media Type Registration: application/pacr+cbor

This document requests that IANA register the following media type in the "Media Types" registry [RFC6838]:

Type name: application

Subtype name: pacr+cbor

Required parameters: none

Optional parameters: none

Encoding considerations: binary (CBOR map per RFC 8949)

Security considerations: See Section 9.

Published specification: This document.

Applications that use this media type: High-throughput PACR
deployments where JSON encoding overhead is significant.

Additional information: Magic number(s): none

File extension(s): .pacr.cbor

Macintosh file type code(s): none

Person and email address to contact for further information: Kwai
Lap Tsoi <kwailapt@gmail.com>

Intended usage: COMMON

Restrictions on usage: none

Author: Kwai Lap Tsoi <kwailapt@gmail.com>

Change controller: IETF

10.3. PACR Intervention Kind Registry

This document requests that IANA establish a new registry named "PACR Intervention Kind" with the following initial contents. The assignment policy is "Specification Required" per [RFC8126].

Value	Name	Pearl Rung	Reference
0x00	Observe	1	This document
0x01	DoPhysical	2	This document
0x02	DoDigital	2	This document
0x03	DoChemical	2	This document
0x04	DoGenetic	2	This document
0x05	Counterfactual	3	This document

Table 1

Values 0x06 through 0xFE are unassigned. Value 0xFF is reserved. New assignments require a published specification that defines the semantics of the kind, any additional payload fields it requires, and any additional validation rules.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2018, <<https://www.rfc-editor.org/rfc/rfc8174>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/rfc/rfc8259>>.
- [RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <<https://www.rfc-editor.org/rfc/rfc8949>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/rfc/rfc4648>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/rfc/rfc6838>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/rfc/rfc8126>>.
- [LANDAUER] Landauer, R., "Irreversibility and Heat Generation in the Computing Process", IBM Journal of Research and Development 5(3):183-191, 1961, <<https://doi.org/10.1147/rd.53.0183>>.

11.2. Informative References

- [RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <<https://www.rfc-editor.org/rfc/rfc7515>>.
- [RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <<https://www.rfc-editor.org/rfc/rfc9052>>.
- [RFC9562] Davis, K., Peabody, B., and P. Leach, "Universally Unique Identifiers (UUIDs)", RFC 9562, DOI 10.17487/RFC9562, May 2024, <<https://www.rfc-editor.org/rfc/rfc9562>>.
- [I-D.aevum-agentcard]
Tsoi, K., "AgentCard: A Framework-Neutral Identity and Capability Declaration Format for Agent-to-Agent Communication", Work in Progress, Internet-Draft, draft-aevum-agentcard-00, 2026, <<https://datatracker.ietf.org/doc/draft-aevum-agentcard/>>.
- [ULID-SPEC]
Feerasta, A., "ULID Specification", 2019, <<https://github.com/ulid/spec>>.
- [MCP-SPEC] Anthropic, "Model Context Protocol Specification", 2024, <<https://modelcontextprotocol.io/specification>>.
- [PEARL] Pearl, J., "Causality: Models, Reasoning, and Inference", Cambridge University Press 2nd ed., 2009, <<https://doi.org/10.1017/CBO9780511803161>>.
- [MARGOLUS] Margolus, N. and L. Levitin, "The Maximum Speed of Dynamical Evolution", Physica D 120(1-2):188-195, 1998, <[https://doi.org/10.1016/0167-2789\(98\)00054-2](https://doi.org/10.1016/0167-2789(98)00054-2)>.
- [CRUTCHFIELD]
Crutchfield, J. and C. Shalizi, "Computational Mechanics: Pattern and Prediction, Structure and Simplicity", Journal of Statistical Physics 104:817-879, 2001, <<https://doi.org/10.1063/1.1530990>>.
- [CODATA] NIST, "CODATA Recommended Values of the Fundamental Physical Constants: 2018", 2019, <<https://physics.nist.gov/cuu/Constants/>>.
- [W3C-PROV] Lebo, T., Sahoo, S., and D. McGuinness, "PROV-O: The PROV Ontology", 2013, <<https://www.w3.org/TR/prov-o/>>.

[PACR-SPEC]

Tsoi, K. L., "pacr-types: Reference Implementation of the PACR 6-Tuple in Rust", 2026, <<https://github.com/kwailapt/AgentCard/tree/main/crates/pacr-types>>.

Acknowledgements

The PACR six-tuple structure was developed in the context of the Aevum Network specification work over 2026. The author thanks reviewers of the AgentCard draft ([I-D.aevum-agentcard]) for early feedback that motivated the separation of capability declaration (AgentCard) from event recording (PACR).

The Pearl-hierarchy intervention tag was added after practical use of an untagged predecessor format made it clear that downstream consumers cannot reliably distinguish observations from interventions without an explicit marker.

Author's Address

Kwai Lap Tsoi
Aevum Network
Email: kwailapt@gmail.com
URI: <https://github.com/kwailapt/AgentCard>