

Open Canonical Energy Telemetry Standard

Normative Architecture and Information Model for Vendor-Independent Energy Telemetry Platforms

Open Canonical Energy Telemetry Standard

Document: 1 of 5

Type: Normative Standard

Version: 1.1

Status: Architecture Review Candidate

Release: July 2026

Compatibility: Reference Implementation 1.x

1. Purpose

The Open Canonical Energy Telemetry Standard (OCETS) defines a vendor-independent information model for acquiring, validating, storing, and consuming telemetry from energy systems.

OCETS standardizes the semantic shape of observations and events: identities, metrics, units, labels, timestamps, quality, provenance, topology, registry requirements, security, and lifecycle behavior. It does not standardize a particular software stack or vendor integration method.

2. Scope

This document specifies:

- normative terminology;
- the canonical information model;
- identity scope and label semantics;
- canonical metric and unit rules;
- role-based power sign convention;
- controlled vocabularies;
- measurement point, asset, device, and plant modeling;
- Plant Registry requirements;
- timestamp semantics;
- telemetry health;
- downsampling semantics;
- governance and conformance rules.
- canonical observations, events, quality, and provenance;
- organizations, portfolios, relationships, and topology;
- Registry API interoperability and a normative security baseline.

This document does not specify implementation details such as collector configuration, transport topics, storage deployment, dashboard construction, vendor register maps, polling templates, or transformation scripts. Those topics belong to the Reference Implementation Guide.

OCETS v1.1 is telemetry-only. A `controls` relationship describes topology; it does not authorize or define a write operation.

2.1 Architecture Constraints

OCETS intentionally does not standardize:

- control systems, SCADA commands, setpoints, dispatch, or actuator writes;
- optimization algorithms or forecast-generation models;
- market communication, billing, settlement, or tariff calculation;
- asset-management, maintenance-management, ERP, or procurement workflows;
- vendor register maps, protocol configuration, dashboards, or a specific database.

Externally generated forecasts MAY be represented as observations with `origin=forecast`, but SHALL include both `issued_at` and `valid_at`. Their generation, model lifecycle, confidence model, and revision policy remain outside OCETS v1.1.

2.2 Reference Architecture



Every component MAY be replaced independently if it preserves the canonical contracts defined by OCETS.

3. Normative Language

The key words `SHALL`, `SHALL NOT`, `SHOULD`, `SHOULD NOT`, and `MAY` are to be interpreted as normative requirement levels.

- `SHALL` indicates a mandatory requirement.
- `SHALL NOT` indicates a mandatory prohibition.
- `SHOULD` indicates a recommended requirement that may be waived only with documented rationale.
- `MAY` indicates an optional capability.

4. Terminology

Term	Definition
Plant	A physical or operational site that produces, consumes, stores, converts, or measures energy.
Asset	A physical energy-system component within one Plant.
Device	A logical telemetry-producing or telemetry-relevant function within one Plant.
Measurement Point	A semantic location where a Metric applies.
Metric	A canonical physical or operational quantity identified by metric name only.
Capability	A declared ability of a Device or Measurement Point to provide one or more canonical Metrics.
Registry	The authoritative schema-enforced declaration of Plants, Assets, Devices, Measurement Points, Capabilities, and relationships.
Label	A bounded metadata field attached to telemetry to identify context without changing the Metric name.
Organization	A tenant and governance boundary that owns Plants and Portfolios.
Portfolio	A named grouping of Plants within one Organization.
Observation	A metric value with identity, acquisition time, quality, and provenance.
Event	A point-in-time or interval occurrence with lifecycle, severity, source, and correlation.
Relationship	A stable, typed connection between registered objects.
Topology	A versioned graph of Relationships for one Plant and one topology kind.

5. Architectural Principles

5.1 Semantic Before Technical

The platform SHALL standardize physical and operational concepts rather than protocol objects, register names, addresses, hostnames, or vendor terminology.

Vendor-specific telemetry SHALL be mapped into canonical concepts before it is stored as OCETS telemetry.

5.2 Stable Identity

Telemetry identity SHALL be independent of network addresses, collector placement, gateway topology, source file layout, and deployment topology.

Logical identities SHALL remain stable throughout the lifecycle of the platform. Replacing a collector, gateway, protocol adapter, or physical communication path SHALL NOT require a semantic identity change.

5.3 Canonical Representation

Every physical quantity SHALL have one canonical metric name, one canonical engineering unit, and one semantic definition.

The canonical metric representation is the metric name itself, such as `active_power`, `voltage`, or `state_of_charge`. OCETS telemetry SHALL NOT require or add a redundant `metric` label to restate the metric name.

Multiple canonical representations of the same physical quantity SHALL NOT coexist in the canonical model. Domain-specific metric names SHALL be used only where the domain concept changes the meaning of the quantity.

5.4 Separation of Concerns

The following concepts SHALL remain independent:

- Plant;
- Asset;
- Device;
- Measurement Point;
- Metric;
- Capability.

No concept SHALL implicitly define another. For example, `active_power` defines what is measured; the Measurement Point and labels define where and in what role it is measured.

5.5 Declarative Configuration

The platform SHALL be described by an authoritative Plant Registry.

Operational configuration SHOULD be generated from the registry. Manual duplication of plant, asset, device, measurement point, and label definitions SHOULD be avoided.

5.6 Acquisition Independence

The canonical model SHALL NOT depend on where telemetry acquisition takes place.

Central acquisition, edge acquisition, and hybrid acquisition are conformant if they emit identical canonical telemetry for identical measurements.

5.7 Operational State Is Telemetry

Operational health SHALL be represented explicitly.

Device health, communication health, collector health, timestamp synchronization, freshness, and validation health SHALL NOT be inferred only from process values.

6. Canonical Information Model

The canonical information model defines the semantic structure of telemetry:

```
Plant
  -> Asset
    -> Device
      -> Measurement Point
        -> Metric
```

The ownership and grouping model is:

```
Organization
  -> Portfolio
    -> Plant
```

The telemetry model produces two distinct records: Observations and Events. Relationships form versioned physical, electrical, and telemetry Topologies around registered objects.

6.1 Plant

A Plant is a physical or operational site that produces, consumes, stores, converts, or measures energy.

A Plant SHALL have a globally unique stable identifier.

6.2 Asset

An Asset is a physical energy-system component, such as an inverter, meter, battery system, transformer, gateway enclosure, or switchgear cabinet.

An Asset SHALL belong to exactly one Plant. An Asset identifier SHALL be unique within its Plant.

6.3 Device

A Device is a logical telemetry-producing or telemetry-relevant function.

A physical Asset MAY expose more than one Device. A Device SHALL belong to exactly one Plant and SHALL have an identifier unique within that Plant.

6.4 Measurement Point

A Measurement Point is a semantic location where a Metric applies.

Examples include:

- grid connection;
- inverter AC output;
- inverter DC input;
- PV string;
- MPPT channel;
- battery DC bus;
- site load;
- transformer secondary.

A Measurement Point SHALL belong to exactly one Plant and SHALL have an identifier unique within that Plant.

6.5 Metric

A Metric is a canonical physical or operational quantity.

Examples include `active_power`, `energy_import`, `voltage`, `current`, `frequency`, `state_of_charge`, `device_available`, `communication_status`, and `timestamp_skew`.

Raw vendor register names, dashboard aliases, protocol field names, and implementation-specific source names SHALL NOT be treated as canonical metric names.

7. Identity and Label Model

The canonical identity of a telemetry series is the combination of:

- canonical metric name;
- `plant`;
- `asset`, where applicable;
- `device`, where applicable;
- `measurement_point`, where applicable;
- additional allowed labels required to disambiguate the measurement, such as `phase`, `string`, `mppt`, or `direction`.

The `plant` label value SHALL be globally unique. `asset`, `device`, and `measurement_point` label values SHALL be unique within a Plant.

Labels SHALL use controlled values where this standard or the Plant Registry Specification defines a vocabulary. Labels SHALL NOT encode multiple concepts in one field when separate fields are defined.

Implementation labels MAY exist outside the canonical model, but they SHALL NOT be required to interpret OCETS telemetry semantics.

8. Controlled Vocabularies

OCETS v1.1 defines the following controlled vocabularies. The Plant Registry Specification may add stricter schema placement rules for these values.

8.1 Role

Allowed `role` values:

Value	Meaning
<code>grid</code>	Grid interconnection or utility-facing measurement.
<code>load</code>	Consumer or site load measurement.
<code>pv</code>	Photovoltaic generation measurement.
<code>battery</code>	Electrical storage measurement.
<code>inverter</code>	Power conversion device or measurement.
<code>meter</code>	Metering device or measurement.
<code>transformer</code>	Transformer device or measurement.
<code>busbar</code>	Electrical busbar or bus section.
<code>collector</code>	Telemetry acquisition or collection component.
<code>environment</code>	Environmental sensing component.

8.2 Energy Form

Allowed `energy_form` values:

Value	Meaning
<code>electricity_ac</code>	Alternating-current electricity.
<code>electricity_dc</code>	Direct-current electricity.
<code>thermal</code>	Thermal energy or temperature-related telemetry.
<code>environmental</code>	Environmental conditions.
<code>operational</code>	Platform or equipment health telemetry.

8.3 Status

Allowed `status` values:

Value	Meaning
<code>planned</code>	Defined but not yet commissioned.
<code>active</code>	In normal service.
<code>maintenance</code>	Temporarily under maintenance.
<code>disabled</code>	Intentionally disabled.
<code>retired</code>	No longer active but retained for history.

8.4 Phase, Direction, and Lifecycle

Allowed `phase` values are `l1`, `l2`, `l3`, `n`, and `total`.

Allowed `direction` values are `import`, `export`, `charge`, `discharge`, `generation`, `consumption`, and `bidirectional`.

Allowed metric lifecycle values are `active`, `proposed`, `deprecated`, and `removed`.

9. Power Sign Convention

OCETS v1.1 uses a role-based sign convention for instantaneous power metrics.

For `active_power`, `reactive_power`, and `apparent_power`:

Role or Measurement Context	Positive Value Means	Negative Value Means
<code>pv</code> generation or export	Generation/export from the PV source	Reverse flow into the PV source, if measured
<code>load</code> or consumer	Consumption by the load	Reverse export from the load context, if measured
<code>battery</code>	Charging into storage	Discharge/export from storage
<code>grid</code>	Import from grid into the plant	Export from plant to grid
<code>inverter</code>	Direction defined by the associated Measurement Point role	Opposite of the associated Measurement Point role

Implementations SHALL apply this convention before canonical storage. Where the physical source exposes only unsigned values, the Measurement Point role and `direction` label SHALL make the interpretation explicit.

Energy counters SHOULD use direction-specific metrics such as `energy_import`, `energy_export`, `energy_charged`, and `energy_discharged` rather than signed accumulated counters.

10. Canonical Metrics

Canonical metrics SHALL be defined in the Canonical Metric Catalogue.

Each canonical metric definition SHALL include:

- metric name;
- canonical unit;
- value type;
- aggregation semantics;
- semantic definition;
- allowed labels;
- source class;
- lifecycle status.
- immutable definition version;
- introduction, deprecation, removal, and replacement versions where applicable.

Generic electrical metrics SHALL be preferred wherever possible. For example, battery power is represented as `active_power` at a battery Measurement Point, using the role-based sign convention. Domain-specific metric names are reserved for true domain concepts such as `state_of_charge`.

11. Canonical Units

Each physical quantity SHALL use one canonical storage unit.

Unit conversion SHALL occur before canonical storage unless the source value is already in the canonical unit.

Reference canonical units include:

Quantity	Canonical Unit
Active power	kW
Reactive power	kvar
Apparent power	kVA
Energy	kWh
Voltage	V
Current	A
Frequency	Hz
Temperature	degC
Ratio	percent

Dashboards and downstream systems SHOULD NOT apply corrective unit conversion to canonical telemetry.

12. Plant Registry

The Plant Registry SHALL be the authoritative source for:

- plants;
- assets;
- devices;
- measurement points;
- capabilities;
- relationships;
- lifecycle status;
- canonical labels.

The registry SHALL be schema-enforced. Registry data SHALL be validatable before generated configuration or documentation is produced.

Collectors, dashboards, alert rules, deployment artifacts, and implementation configuration SHOULD derive from the registry.

13. Acquisition Topologies

OCETS supports three acquisition topologies:

Topology	Description
Central acquisition	Central systems acquire telemetry from plants.
Edge acquisition	Plant-local systems acquire and forward telemetry.
Hybrid acquisition	Edge and central systems share acquisition responsibilities.

All acquisition topologies SHALL emit canonical telemetry with identical identity, units, metrics, labels, and timestamp semantics for the same physical measurement.

14. Time Model

Telemetry timestamps SHALL represent acquisition time.

Processing time, transport time, receive time, storage write time, and dashboard render time SHALL NOT replace acquisition time.

All stored timestamps SHALL be represented in UTC.

Acquisition systems SHALL synchronize against a centrally managed platform time service or an equivalent redundant time source.

15. Telemetry Health

Telemetry health SHALL be modeled explicitly.

Health telemetry SHOULD include:

- device availability;
- communication status;
- collector status;
- last successful acquisition;
- timestamp skew;
- pipeline status;
- data freshness;
- validation errors.

Missing values SHALL NOT be silently converted to zero.

Stale values SHALL NOT be presented as fresh measurements.

15.1 Edge Collector Observability Profile

The Edge Collector Observability Profile is an optional OCETS conformance profile for plant-local systems that acquire, buffer, or forward telemetry. Implementations that claim this profile SHALL represent the edge collector as a stable Device with `role=collector` and `energy_form=operational`.

The profile SHALL expose:

- `device_available` for independently observed collector availability;
- `collector_status`, `pipeline_status`, and upstream `communication_status`;
- `data_freshness` as the age of the latest successful acquisition;
- `time_sync_status` and `timestamp_skew`;
- `buffer_utilization` and `buffer_oldest_age`;
- `storage_utilization` and `storage_status`;
- `transport_error_count`, `collector_restart_count`, `mapping_error_count`, and `validation_error_count`.

An implementation MAY additionally expose `cpu_utilization`, `memory_utilization`, and `host_temperature`. Where suitable power-monitoring hardware exists, it MAY expose `external_power_available`, `backup_power_status`, and `backup_power_runtime`.

Unsupported optional metrics SHALL be absent, not reported as zero or a healthy state. A required metric that cannot currently be determined SHALL use its defined `unknown` state where the metric type permits it. Hostnames, IP addresses, operating-system interface names, mount paths, and hardware serial numbers SHALL NOT form canonical identity.

Edge Collector Observability conformance SHALL NOT be required from central-only deployments or from implementations that do not claim the profile.

16. Data Lifecycle and Downsampling

Telemetry SHALL progress through the following lifecycle:

```
Acquisition -> Validation -> Canonical Mapping -> Storage
-> Downsampling -> Long-Term Retention -> Consumption
```

Each lifecycle stage SHALL preserve semantic correctness.

Downsampling SHALL respect metric semantics. The aggregation behavior for each metric SHALL be defined in the Canonical Metric Catalogue.

Missing values SHALL NOT be converted to zero during downsampling.

Retention durations are reference policy, not normative OCETS v1.1 requirements. A conforming implementation MAY choose different retention durations if it preserves the normative downsampling semantics of each metric.

17. Canonical Observation Model

An Observation is the logical, implementation-independent representation of one telemetry value. Every Observation SHALL be resolvable to:

- canonical metric name and catalogue version;
- value and the canonical unit implied by the metric definition;
- Plant and, where applicable, Asset, Device, and Measurement Point;
- `acquired_at` in UTC;
- quality;
- provenance.

The logical model MAY be stored as one record, multiple fields, or coordinated series. A storage implementation SHALL preserve or reconstruct every required property. Quality, provenance, catalogue version, calculation identifiers, and source references SHALL NOT be encoded as unbounded TSDB labels.

Forecast-origin Observations SHALL additionally contain `issued_at` and `valid_at`. Derived or calculated Observations SHOULD identify the calculation definition and source Observations or source series.

18. Quality Model

Every Observation SHALL carry or inherit exactly one quality value:

Value	Meaning
<code>GOOD</code>	Valid according to all applicable acquisition and validation rules.
<code>BAD</code>	Known to be invalid and unsuitable for normal consumption.
<code>UNCERTAIN</code>	Available, but validity cannot be established completely.
<code>ESTIMATED</code>	Produced by an estimation method instead of direct acquisition.
<code>SUBSTITUTED</code>	Replaced by an approved alternate value.
<code>MISSING</code>	An explicit gap marker; it SHALL NOT contain a fabricated numeric zero.
<code>SIMULATED</code>	Produced by a simulation rather than the physical process.

Downsampling SHALL report `GOOD` only when every contributing Observation is `GOOD`. Otherwise the conservative precedence from worst to best is `BAD`, `MISSING`, `UNCERTAIN`, `SUBSTITUTED`, `ESTIMATED`, `SIMULATED`, `GOOD`. Implementations MAY retain additional contributing-quality detail.

19. Provenance Model

Every Observation and Event SHALL declare one origin:

Origin	Meaning
device	Directly acquired from a physical or logical Device.
calculated	Deterministically calculated from other data.
forecast	Produced externally as a prediction.
estimated	Inferred because a direct value was unavailable or incomplete.
manual	Supplied or corrected by an authenticated human workflow.
simulated	Produced by a simulation or test environment.

Provenance SHALL identify the originating Device, calculation, model, or authenticated actor as applicable. Manual changes and substitutions SHALL be auditable. Provenance describes origin; quality describes confidence or usability, and neither SHALL replace the other.

20. Canonical Event Model

Events are not Metrics and SHALL NOT be modeled as ordinary time-series samples. A canonical Event SHALL include:

- globally unique `event_id`;
- canonical `event_type` and `severity`;
- typed source reference to a registered object;
- `begin_at` and optional `end_at`, in UTC;
- lifecycle state `open`, `acknowledged`, `closed`, or `cancelled`;
- quality and provenance;
- optional `correlation_id` and `causation_id`.

Severity values are `info`, `warning`, `minor`, `major`, and `critical`. Initial event types are `alarm`, `warning`, `fault`, `maintenance`, `restart`, `firmware_update`, and `grid_failure`.

Acknowledgement SHALL record actor, timestamp, and optional comment without overwriting event history. Re-delivery or update of an `event_id` SHALL be idempotent. `end_at` SHALL NOT precede `begin_at`. Event payloads MAY contain bounded type-specific attributes but SHALL NOT contain secrets.

21. Relationship and Topology Model

A Relationship SHALL have a stable ID, Plant, Topology, type, typed `from` and `to` endpoints, lifecycle status, and validity interval. Allowed relationship types are:

- `contains`;
- `feeds`;
- `connected_to`;
- `measures`;
- `controls`;
- `powered_by`.

`contains`, `feeds`, `measures`, `controls`, and `powered_by` are directed. `connected_to` is symmetric and SHALL NOT be duplicated in reverse order. A `controls` Relationship records semantic responsibility only and SHALL NOT imply command authorization or OCETS control conformance.

A Topology is an immutable versioned graph for exactly one Plant. Its kind SHALL be `electrical`, `physical`, or `telemetry`. Grid connections, transformers, busbars, inverters, batteries, loads, collectors, Devices, and Measurement Points SHALL be represented using registered objects rather than topology-only identifiers.

`contains` Relationships SHALL be acyclic. Electrical networks MAY be meshed; cycles in `feeds` and `connected_to` are permitted. Topology versions SHALL have non-overlapping validity intervals when they claim to be the active version of the same topology.

22. Multi-Tenancy

An Organization is the tenant, ownership, authorization, and governance boundary. An Organization SHALL have a globally unique stable ID. A Portfolio SHALL belong to exactly one Organization. A Plant SHALL belong to exactly one Organization and MAY belong to zero or more Portfolios of that Organization.

Canonical Plant identity remains globally unique and SHALL NOT depend on an Organization or Portfolio name. Relationships and Topologies SHALL NOT cross Organization boundaries. Cross-organization analysis MAY occur in an authorized consumer but SHALL NOT create cross-tenant registry ownership.

Organization and Plant metadata MAY declare a data residency region. Such metadata informs deployment policy but does not replace applicable law or operator policy.

23. Registry API Profile

A conforming Registry API SHALL provide authenticated, read-only HTTP/JSON access under `/api/ocets/registry/v1` to:

- service metadata and available Registry revisions;
- Organizations, Portfolios, Plants, Assets, Devices, Measurement Points;
- Relationships, Topologies, Metric Definitions, and Event Types;
- one consistent bulk snapshot;
- changes since a retained Registry revision.

Collections SHALL support opaque cursor pagination, a bounded `limit`, and filters for tenant, lifecycle status, and `updated_since`. Every response SHALL identify one immutable Registry revision. Implementations SHALL support `ETag` and `If-None-Match`; pagination across one traversal SHALL remain snapshot-consistent.

Errors SHALL use a JSON object containing stable `code`, human-readable `message`, and `request_id`. Resource absence SHALL be distinguishable from lack of authorization without leaking cross-tenant existence. Mutation, approval, and deployment APIs are outside this profile.

24. Security Model

OCETS implementations SHALL:

- use distinct identities for people, services, collectors, and Devices;
- authenticate every non-public Registry API and telemetry connection;
- encrypt telemetry and Registry traffic in transit;
- apply default-deny authorization by Organization, resource, and action;
- prevent credentials, private keys, tokens, and other secrets from entering Registry data, labels, Events, or logs;
- protect sensitive data at rest according to its classification and residency policy;
- audit Registry changes, manual provenance, Event acknowledgement, authorization changes, and security-relevant access;
- support credential and certificate rotation without semantic identity changes.

Network location alone SHALL NOT establish trust. Service and collector permissions SHALL follow least privilege. Audit records SHOULD be tamper-evident and retained under governed policy. Security incidents and credential compromise SHALL have documented revocation and recovery procedures.

25. Governance

OCETS uses semantic versioning. Governance roles are Maintainer, Architecture Review Board, and Extension Owner. Normative changes SHALL have an identified owner, review record, compatibility assessment, and, when architectural, an ADR.

Version 1.x releases MAY add canonical metrics, labels, capabilities, optional registry fields, event types, and non-breaking clarifications. Version 2.0 is required for changed identity semantics, changed canonical units, removed required fields, incompatible timestamp rules, or incompatible metric meanings.

Metric definitions SHALL record `definition_version`, `introduced_in`, `deprecated_in`, `removed_in`, and `replaced_by` where applicable. A released definition is immutable. Deprecation SHALL provide a replacement or rationale and remain supported for at least one subsequent minor release before removal.

Local extensions SHALL use `x-<organization>-` namespaces and SHALL NOT redefine canonical names or controlled values. Released catalogue and conformance-profile versions SHALL remain retrievable. Security reports SHALL have a documented confidential disclosure and remediation process.

26. Conformance

Core OCETS v1.1 conformance requires an implementation to:

- implement the canonical information and Observation models;
- use canonical metric names, units, quality, and provenance;
- use the role-based power sign convention and OCETS timestamp rules;
- expose telemetry health and enforce the Security Model;
- validate Registry data against a schema or equivalent enforceable contract;
- satisfy all mandatory `SHALL` requirements and preserve implementation separation.

Event, Topology, Multi-Tenancy, Registry API, and Edge Collector Observability are claimable conformance profiles. An implementation SHALL satisfy every mandatory rule of each profile it claims. Representing only one Organization does not waive tenant isolation requirements if the Multi-Tenancy profile is claimed.

Conformance does not require a specific software stack, control capability, or preservation of legacy and vendor-specific names.