

OCETS v1.1 Executive Summary

Open Canonical Energy Telemetry Standard

Canonical Energy Telemetry Platform

Release: July 2026

Canonical Energy Telemetry Platform Standard

Document Identifier: OCETS-CORE-1.1

Status: Architecture Review Candidate

Release: July 2026

Compatibility: Reference Implementation 1.x

Document Type: Executive Summary

1. Purpose

OCETS defines a vendor-independent semantic architecture for energy telemetry at platform scale. It standardizes how observations, events, identities, metrics, time, quality, provenance, topology, ownership, security, and lifecycle are interpreted across vendors, protocols, collectors, storage systems, countries, and years of operation.

The standard is intended for photovoltaic plants, batteries, EV charging, weather systems, meters, transformers, loads, grid services, and related infrastructure. Physical implementations may change without changing canonical meaning or historical identity.

2. Scope and Architecture Constraints

OCETS standardizes telemetry and its governing metadata. It intentionally does not standardize control systems, SCADA commands, setpoints, dispatch, optimization algorithms, forecast-generation models, market communication, billing, settlement, asset management, ERP integration, vendor register maps, dashboards, or a specific software stack.

A `controls` relationship records semantic responsibility but never authorizes a write. External forecasts may be represented only with explicit origin, issue time, and validity time; model generation and revision remain outside v1.1.

3. Reference Architecture



Registry, generator, collectors, transport, time-series database, APIs, and visualization are separable components. Identical physical measurements SHALL produce identical canonical output in central, edge, or hybrid acquisition topologies.

4. Core Principles

- **Semantic before technical:** vendor fields and protocol objects are mapped before canonical storage.
- **Stable identity:** hostnames, IP addresses, register addresses, collector placement, and replaceable hardware do not define telemetry identity.
- **Canonical representation:** one physical concept has one Metric, canonical unit, and semantic definition.
- **Separation of concerns:** Organization, Portfolio, Plant, Asset, Device, Measurement Point, Metric, Observation, Event, Relationship, and Topology remain distinct.
- **Declarative configuration:** a schema-enforced Registry is authoritative; generated artifacts do not become independent truth.
- **Operational state as telemetry:** missing, stale, invalid, failed, and measured zero are distinguishable conditions.

5. Canonical Information Model

```
Organization -> Portfolio -> Plant
                        -> Asset -> Device -> Measurement Point -> Metric
                        -> Topology -> Relationship -> typed endpoints

Metric + identity + value + acquisition time + quality + provenance = Observation
event type + source + interval + lifecycle + severity + correlation = Event
```

Each Plant belongs to one Organization and may appear in several Portfolios of that Organization. Plant IDs remain globally unique, so portfolios and tenant names never become part of time-series identity.

6. Observations, Quality, and Provenance

Every Observation resolves to its Metric and catalogue version, value, canonical unit, semantic identity, acquisition timestamp, quality, and provenance. A TSDB may store these in coordinated structures, but no required property may be lost or turned into an unbounded label.

Quality uses `GOOD`, `BAD`, `UNCERTAIN`, `ESTIMATED`, `SUBSTITUTED`, `MISSING`, and `SIMULATED`. Missing data is never fabricated as zero. Rollups are `GOOD` only when every contributing Observation is `GOOD`; otherwise they propagate the most conservative quality.

Provenance uses `device`, `calculated`, `forecast`, `estimated`, `manual`, or `simulated`. It identifies the originating Device, calculation, model, or authenticated actor. Provenance describes origin; quality describes usability, and neither replaces the other.

7. Canonical Event Model

Events are separate from Metrics and samples. A canonical Event carries globally unique identity, event type, severity, typed source, begin and optional end, lifecycle state, acknowledgement history, quality, provenance, and optional correlation and causation.

Initial event types are alarm, warning, fault, maintenance, restart, firmware update, and grid failure. Event updates are idempotent. Acknowledgement records actor and time without destroying prior state.

8. Relationships and Topology

Relationships are stable, typed, time-bounded Registry objects. OCETS defines `contains`, `feeds`, `connected_to`, `measures`, `controls`, and `powered_by`.

A Topology is an immutable versioned electrical, physical, or telemetry graph for one Plant. Grid connections, transformers, busbars, inverters, batteries, loads, Devices, and Measurement Points use registered identity rather than anonymous diagram nodes. Containment is acyclic; electrical networks may be meshed.

9. Metrics, Units, and Definition Versions

Metric names are canonical by themselves. Examples include `active_power`, `energy_import`, `voltage`, `state_of_charge`, `communication_status`, and `timestamp_skew`. Units such as kW, kWh, V, A, Hz, degC, and percent are normalized before storage.

Every Metric definition records immutable `definition_version`, `introduced_in`, `deprecated_in`, `removed_in`, and `replaced_by` metadata. Published meanings are not edited in place. Incompatible meanings require a new Metric name or OCETS major version.

10. Time, Health, and Data Lifecycle

Observation timestamps represent acquisition time in UTC, not processing, transport, receive, or storage time. Forecast-origin observations additionally distinguish `issued_at` and `valid_at`.

Health covers Device, communication, collector, pipeline, freshness, synchronization, buffering, storage, mapping, and validation. OCETS also defines an optional Edge Collector Observability Profile.

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

Downsampling follows Metric aggregation semantics and propagates quality and provenance. Retention duration remains governed deployment policy.

11. Registry and Registry API

The Registry contains Organizations, Portfolios, Plants, Assets, Devices, Measurement Points, capabilities, Topologies, Relationships, Metric Definitions, Event Types, lifecycle state, and canonical labels. Schema and cross-reference validation enforce identity, tenant, graph, interval, and profile rules.

The read-only HTTP/JSON profile at `/api/ocets/registry/v1` provides resource discovery, revision-consistent snapshots, changes since a revision, cursor pagination, filtering, ETags, and stable error objects. Mutation and approval workflows remain implementation-specific.

12. Security

OCETS requires distinct identities for people, services, collectors, and Devices; authentication of non-public connections; encrypted transport; default-deny tenant-, resource-, and action-scoped authorization; managed credential rotation; least privilege; protected data at rest; and audit of security-relevant actions.

Secrets are forbidden in Registry content, telemetry labels, Events, and logs. Network location or VPN membership alone does not establish trust. Cross-Organization Registry relationships are invalid.

13. Governance

Maintainers, the Architecture Review Board, and Extension Owners govern changes. Normative changes require ownership, review, compatibility assessment, and an ADR where architectural. Released Registry, catalogue, and conformance-profile revisions remain retrievable.

Local extensions use `x-<organization>` namespaces and cannot redefine canonical names. Deprecations include replacement or rationale and remain supported for at least one later minor release. A confidential security-disclosure and remediation process is required.

14. Versioning

Version 1.x may add compatible Metrics, labels, event types, capabilities, optional Registry fields, and clarifications. Version 2.0 is required for incompatible identity, unit, timestamp, required-field, or Metric-meaning changes.

OCETS v1.1 remains an Architecture Review Candidate until the new normative models, schema, conformance tests, and published artifacts complete architecture review.

15. Conformance

Core conformance requires the canonical information and Observation models, Metrics and units, quality and provenance, timestamps, power sign rules, health, Registry validation, security baseline, and every applicable `SHALL` requirement.

Event, Topology, Multi-Tenancy, Registry API, and Edge Collector Observability are claimable profiles. A system implements all mandatory rules of each profile it claims. No specific collector, broker, database, dashboard, identity provider, or protocol is required.

16. Executive Conclusion

OCETS v1.1 is a coherent architecture standard rather than a tool-specific design. Its canonical Observation and Event contracts, versioned topology, tenant model, Registry API, security baseline, and governance rules make telemetry interpretable and interoperable from one Plant to tens of thousands of Plants and millions of series.

The remaining Architecture Review Candidate phase is therefore validation and external review, not expansion of the implementation stack or platform scope.