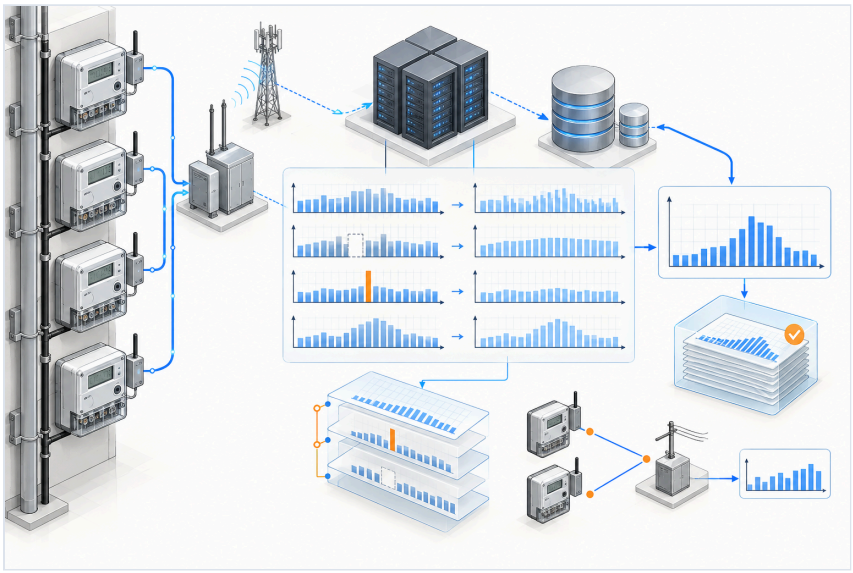
AETHERONE / AETHERMDMS

AetherMDMS

Datasheet

Meter Data Management System for effective-dated metering context, measurement versions and technical data delivery



CONCEPT ILLUSTRATION; NOT A CUSTOMER INSTALLATION OR RUNTIME EVIDENCE.

Concept illustration of an AMI data path: a column of smart meters on a riser feeding a concentrator and a cellular tower, servers and a database, interval bar strips in which a gap and a spike are handled, stacked version layers, an accepted data package and two meters linked to one point with a usage chart. It shows the work AetherMDMS does and is not a customer site or a record of measurements.

Positioning

AetherMDMS is the meter data management system of AetherONE. It sits between the head end and billing. It takes decoded readings from AetherHES, establishes which meter stood on which Metering Point at the time, keeps every version of each measurement, validates and estimates intervals, turns accepted intervals into Usage and technical Bill Determinants, and delivers data products downstream together with acknowledgement and lineage evidence. Its scope also covers technical energy balance and replay.

The table below gives the approved scope and says, row by row, what this edition offers for evaluation.

Intended readers

- **Metering-data operations and data-quality teams**, who need to see how an interval was received, normalized, validated or estimated.
- **Metering-master and meter-replacement teams**, who need the installation history that was correct at a given effective time.
- **Billing-data and integration teams**, who pass accepted Usage and Bill Determinants to downstream systems and need acknowledgement evidence when they do.
- **Technical-balance analysts**, who review energy-balance inputs, topology versions and residual evidence.
- **Architects and project teams** who place MDMS next to HES, CIS, GIS, Revenue and Dispatch.

Capabilities

Area	What AetherMDMS does	In this edition
Metering context	Keeps installations, physical and logical channels, register assignments and CT/PT/multiplier versions as effective-dated, append-only records, so a question about an earlier date is answered with the installation then in force. The metering-point list and Metering Point 360 read a compact read-only projection of the persisted metering context, with an explicit demo mode.	Local read model for evaluation: installations, channels, registers and CT/PT/multiplier versions, read through the list and Metering Point 360 views.
Replacement continuity	A deterministic replacement flow keeps old and new interval references, checks half-open installation boundaries and preserves lineage across a restart.	Local flow for evaluation. Continuity is shown between adjacent intervals and installations.
Canonical measurements	Keeps received, normalized, validated and estimated versions as separate immutable records, each with event, correlation and causation IDs, input version references and hashes.	Local component for evaluation. Each version keeps its inputs and hashes.
VEE (validation, estimation and editing)	Three rules: missing interval, spike and clock drift. Bounded linear interpolation fills a missing interval only where compatible normalized boundary readings exist.	Three rules, and interpolation between compatible boundary readings, for evaluation.
Usage and Bill Determinants	Integrates accepted 15-minute average-kW versions into kWh Usage and writes append-only provisional total-consumption Bill Determinants that carry their input lineage. An incomplete window yields no determinant.	Local component for evaluation. 15-minute average kW in; kWh Usage and provisional total-consumption determinants out.
Data delivery	Immutable data products and approved subscription versions. Async-event and batch-file datasets, plus REST-pull. Every HTTP attempt is recorded, the recipient's acknowledgement is reconciled, and retry, re-run and supersession keep their lineage.	Local component for evaluation. Async-event and batch-file datasets and REST-pull, with each attempt recorded and acknowledgements reconciled.

Area	What AetherMDMS does	In this edition
Lineage and replay	A lineage witness links HES-read references, measurement versions, VEE runs, Usage, determinants, delivery and acknowledgement. An isolated sandbox replay reports a match or a mismatch against pinned hashes and re-sends no delivery.	Local evidence for evaluation. Replay runs in an isolated sandbox and reports a match or a mismatch.
Quality and lifecycle evidence	Reports evidence consistency as match, mismatch or incomplete. Quality scores are not computed because no scoring formula is configured. Lifecycle evaluation is a local dry run that returns decisions only.	Local evidence for evaluation: consistency reporting and lifecycle dry runs.
Technical energy balance	Computes quality-aware balances from persisted Usage and immutable valid-time topology snapshots. Residual and anomaly thresholds produce evidence for investigation.	Local component for evaluation, built on persisted Usage and topology snapshots.
Standards mapping	Versioned field mapping aligned with IEC 61968-9.	For evaluation. Conformance with the standard is not validated.

Interfaces and private-writer responsibility

AetherMDMS has its own private store and is its only writer. Other products reach it through versioned contracts and projections and never read that storage directly.

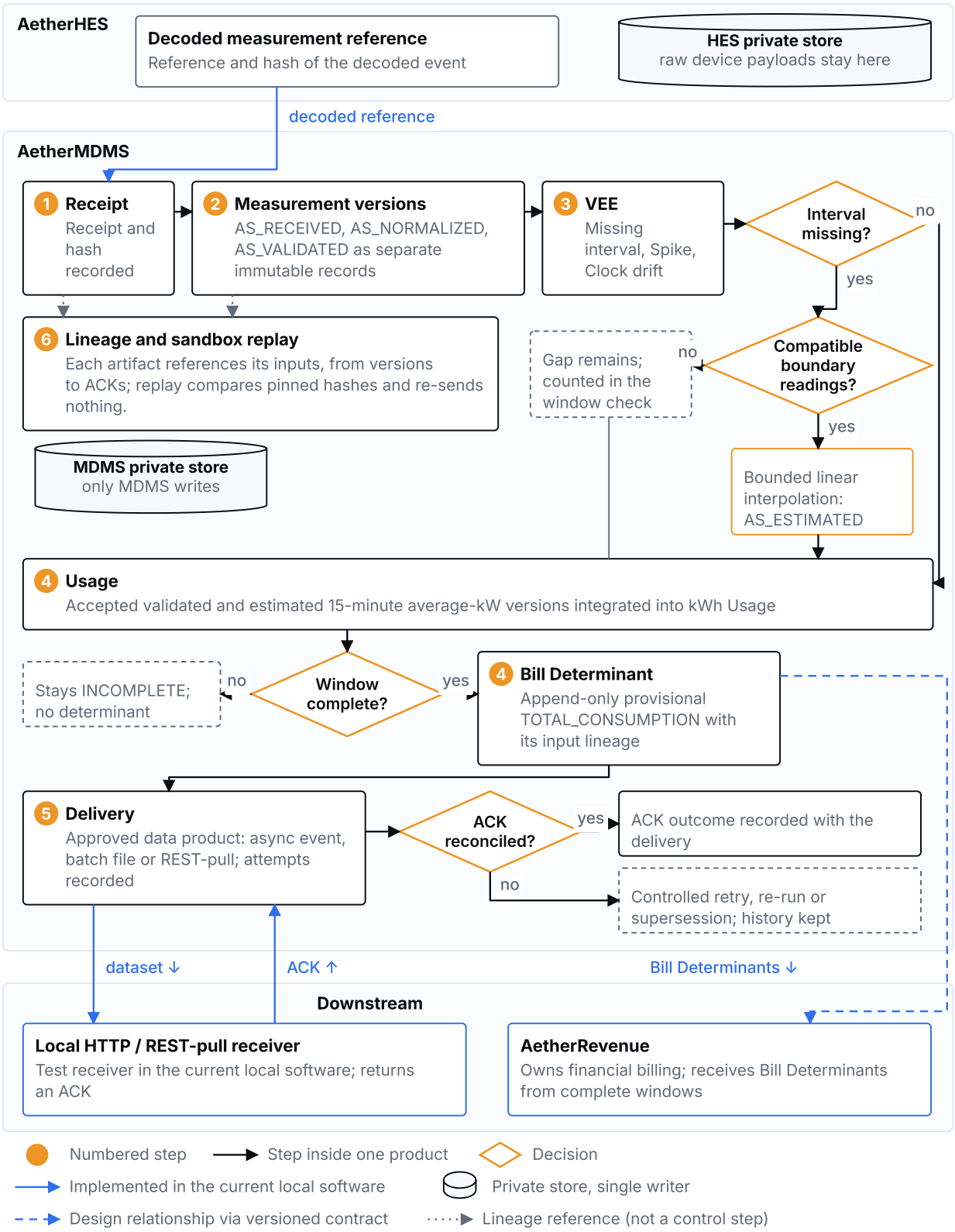
Product	Relationship
AetherHES	Supplies acquisition and decoded measurement references. MDMS has no direct access to the HES private store and does not own device sessions or device control.
AetherCIS	Owns the lifecycle of party, customer account, premise, Service Point and agreement. MDMS holds versioned references and projections. A Metering Point is an MDMS object and is distinct from a CIS Service Point. Referencing both does not merge them.
AetherGIS	Owns address, coordinate, service-area and network-topology truth. MDMS takes versioned topology snapshots as calculation input.
AetherRevenue	Owns tariffs, financial bill cycles and runs, bills and invoices, payments, debt, settlement and prepayment/STS. MDMS supplies validated Usage and Bill Determinants. A Bill Determinant is a technical billing input, not a financial bill or invoice.
AetherDispatch	Owns field-work planning and execution. After the field evidence is validated, MDMS commits the installation and the measurement continuity.
AetherUtilityConsole	Presents MDMS in the unified frontend through navigation and read-only composition. It is not a shared backend.

Billing, settlement and prepayment screens are kept as protected compatibility surfaces. The retained prepayment witness runs on deterministic test tokens with local evidence and is not production STS.

AetherMDMS data path

DESIGN WORKFLOW

Intended order of responsibility; numbered steps match the catalogue table.



DESIGN WORKFLOW Design workflow of the AetherMDMS data path: decoded measurement references, measurement versions, VEE, Usage, Bill Determinants, delivery, acknowledgement, lineage and replay. It is a conceptual design diagram of the intended order of responsibility and does not describe a production integration.

Edition and evaluation conditions

AetherMDMS has an approved design scope, and this edition makes part of it available for evaluation as bounded local components, as the table shows. A utility deployment would need these conditions confirmed. The browser editing, approval and settlement screens and the meter-replacement wizard are demonstration interfaces on synthetic data. Normalization covers the implemented cases rather than every time-zone, unit and late-data situation. Usage stops at 15-minute average kW converted to kWh, so TOU, maximum demand and similar quantities are later scope, and price and tax belong to AetherRevenue. Delivery runs locally, so production adapters, authentication and authorization still have to be configured. Evidence is local and append-only with SHA-256 digests, and no capacity or availability figure is published.

Technical specifications

- **Measurement.** 15-minute average power in kW, integrated into kWh Usage.
- **Processing.** Three validation rules, and linear interpolation limited to a missing interval between compatible boundary readings.
- **Delivery.** Async-event or batch-file datasets, or REST-pull, with every attempt recorded and acknowledgements reconciled.
- **Evaluation environment.** Most local services need Node.js 24 or later and keep their evidence in MDMS-owned local stores.
- **Document edition.** 2026.10 identifies this document. It is not a product release.