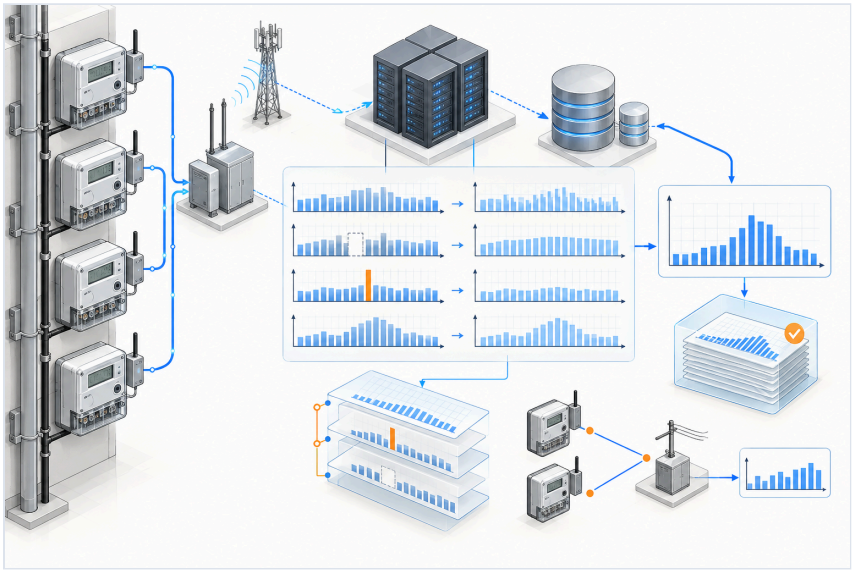
AETHERONE / AETHERMDMS

# AetherMDMS

## Product Catalogue

### From a meter reading to an accepted, traceable data product



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.

## Why metering data needs its own system

A reading that arrives from the head end is not yet something a utility can bill, balance or hand to another system. Someone has to establish which meter was installed at which point at that moment, whether the interval is complete, whether a spike or a clock error has distorted it, and which version of the value went downstream. When those decisions live in spreadsheets and integration scripts, a plain question such as "which reading produced this determinant?" turns into an investigation.

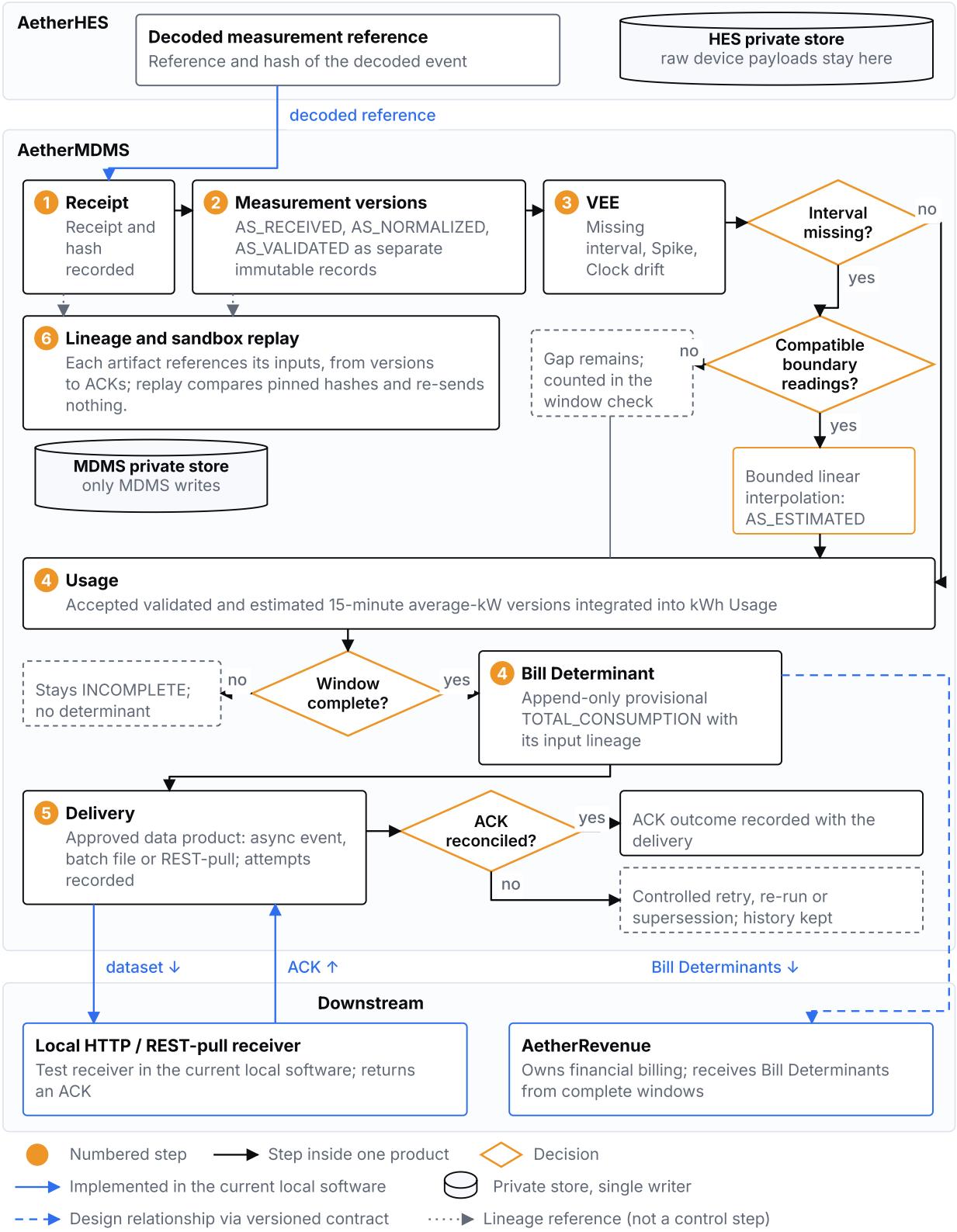
AetherMDMS is the Meter Data Management System of AetherONE, and it answers that question by design. It owns Metering Points and their effective-dated installations, channels and registers, canonical Measurements, VEE (validation, estimation and editing), Usage, Bill Determinants, technical energy balance, lineage, replay and downstream delivery. Device sessions belong to AetherHES. Customer and account identity belongs to AetherCIS, spatial truth to AetherGIS, tariffs, invoices and payments to AetherRevenue, and field-work execution to AetherDispatch.

# The workflow in one view

## 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.

| Step | Object | What happens |
|---|---|---|
| 1 | Decoded measurement reference from HES | MDMS records the receipt and the hash of the decoded event. Raw device payloads stay inside HES. |
| 2 | Measurement versions | Received, normalized, validated and estimated versions are stored as separate immutable records with their input references. |
| 3 | VEE | Missing-interval, spike and clock-drift checks run, and bounded linear interpolation can estimate a missing interval. |
| 4 | Usage and Bill Determinant | Accepted 15-minute average-kW versions are integrated into kWh Usage and provisional total-consumption determinants. |
| 5 | Delivery and acknowledgement | An approved data product goes to a subscriber, and the recipient's acknowledgement is reconciled. |
| 6 | Lineage and replay | Each artifact points back to its inputs, and a sandbox replay compares the result with pinned hashes. |

## Scenario one: replacing a meter without breaking the story

A meter swap changes the installation but not the Metering Point. The metering-master and meter-replacement team needs to see what changed at the effective time of the swap: the old and new installations, the interval references on each side, and the evidence that the two connect.

AetherMDMS keeps installations as effective-dated records, with channels, register assignments and CT/PT/multiplier versions attached. Its deterministic replacement flow keeps the old and new interval references and checks that installation boundaries are half-open. Measurement, VEE, Usage and determinant lineage survives a restart. The Metering Point 360 view gathers this context for one point.

The first excerpt places the Metering Point beside the premise and service point it relates to, which remain separate objects owned by AetherCIS. The second keeps the ratio history with its effective-from dates, so the ratio that applied on a given date can be read back. In the evaluation edition the continuity shown is adjacent interval and installation evidence. It does not prove cumulative-register kWh offset continuity. AetherDispatch owns the field work, and AetherMDMS commits the validated installation and the measurement continuity once that work has been validated.

1 Identity and installation of one metering point

**Identity and installation**

| | |
|---|---|
| Metering point | MP-100200 |
| Premise | PRM-55000 |
| Service point | SP-70100 |
| Current device | KEP88091044 |
| Grid path | SS-Gwangju-01 / FD-Gwangju-01-F2 / TX-GW-1 |
| CT / PT | 200/5 / 1/1 (multiplier 40) |
| Tariff | RES-TOU-A |
| Channels | 2 channels: ch1, ch2 |

2 CT/PT ratio history with effective-from dates

| EFFECTIVE FROM | CT RATIO | PT RATIO | MULTIPLIER |
|---|---|---|---|
| 2023-03-14 | 100/5 | 1/1 | 20 |
| 2024-06-30 | 200/5 | 1/1 | 40 |

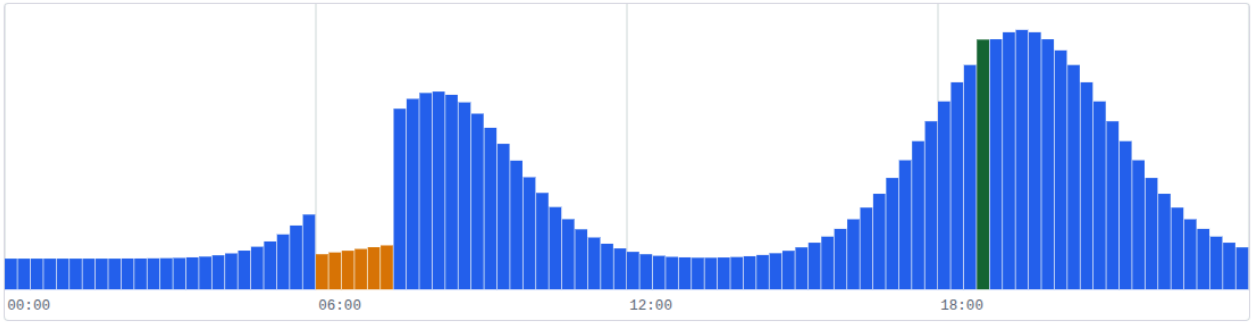
PRODUCT SCREEN · DEMONSTRATION DATA Evaluation interface with demonstration data: Metering Point 360 for a synthetic point. The first excerpt shows the identity and installation of one metering point, with premise, service point and tariff references, current device, grid path, CT/PT and channels. The second shows the CT/PT ratio history with effective-from dates. All values are synthetic.

## Scenario two: following a suspect interval through its versions

When a value looks wrong, the meter-data operations and quality team starts from that one interval, its versions and its rule results. They follow its received, normalized, validated and estimated provenance and read the missing-interval, spike and clock-drift evidence.

Every version of an interval is kept with event, correlation and causation IDs, the versions it was built from, and a hash. An analyst can see what arrived, what normalization made of it, which rule fired, and whether the final value was measured or estimated. Three rules are implemented: missing interval, spike and clock drift. Estimation is bounded linear interpolation, applied only where compatible boundary readings exist.

1 One day of intervals by version layer, with an estimated block and one corrected interval




PRODUCT SCREEN · DEMONSTRATION DATA Evaluation interface with demonstration data: the Measurement Explorer showing one day of intervals, with an estimated block and one corrected interval highlighted. The interval values are synthetic.

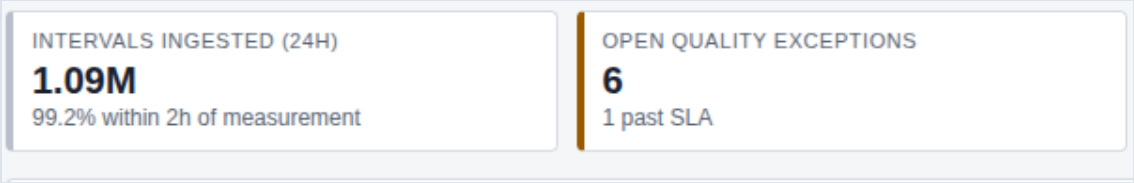
The excerpt shows a day of intervals in which a block has been estimated and a single interval corrected. In the product these layers are separate versions that sit side by side, and none overwrites another. The corrected interval in the demonstration is sample data. Editing, approval and settlement are demonstration controls in this edition, and rule administration and manual editing with four-eyes approval are part of the approved scope but not of this edition.

## Scenario three: handing accepted data downstream and reading the answer

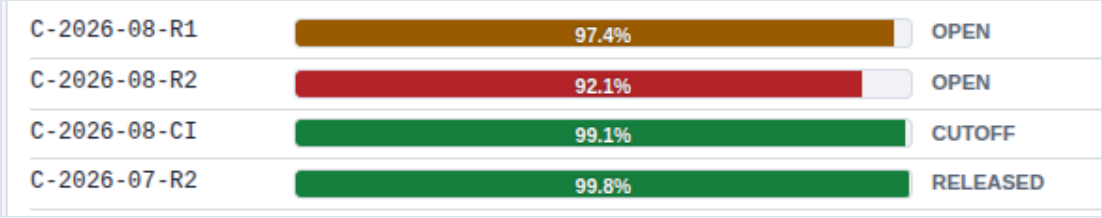
The billing-data and integration team cares about what left AetherMDMS and what came back. They handle accepted Usage, provisional Bill Determinants, deliveries and acknowledgements, tracing technical inputs through dataset delivery and reconciling the recipient's response.

The usage service turns accepted 15-minute average-kW versions into kWh Usage and writes append-only provisional total-consumption determinants, each with its exact input versions. A window that is incomplete produces no determinant. The delivery service then stores immutable data products and approved subscription versions, sends filtered async-event or batch-file datasets (REST-pull is also supported), records each bounded HTTP attempt and reconciles the external acknowledgement. Retries, re-runs and supersessions keep their lineage.

1 Intervals ingested and open quality exceptions




2 Billing readiness by cycle




3 Attention queue




PRODUCT SCREEN · DEMONSTRATION DATA Evaluation interface with demonstration data: the operations summary. The excerpts show intervals ingested and open quality exceptions, billing readiness by billing cycle, and a queue of items that need attention. All figures are synthetic and are not product performance.

Two boundaries matter for this team. First, AetherRevenue keeps tariffs, financial bill cycles and runs, bills and invoices, payments and debt, so MDMS hands over technical inputs and no financial results, and a Bill Determinant is never a bill. Second, the delivery path in the evaluation edition is a local component.

## Scenario four: reviewing a technical energy balance

The technical-balance analyst reviews the assumptions and anomaly thresholds behind a balance, starting from Usage IDs, a topology version and residual evidence. A bounded service computes quality-aware technical energy balances from persisted Usage and immutable valid-time topology snapshots. Each result keeps its input Usage IDs, the topology versions and a result hash, and configured residual or anomaly thresholds produce evidence for later investigation.

Treat this as an evaluation scenario: the balance runs on local Usage and topology snapshots, without production GIS topology ingestion or calibrated loss models, and no loss reduction is claimed.

## How AetherMDMS fits with the other products

HES supplies acquisition and decoded references. CIS supplies the customer account, premise and service point relationships that MDMS refers to. GIS supplies versioned topology. Revenue consumes validated Usage and Bill Determinants for financial billing. Dispatch supplies validated field completion, and MDMS commits the installation and its continuity. Each product keeps its own private store, and they meet through versioned contracts and projections.

## Availability and evaluation edition

AetherMDMS has an approved design scope, and this edition makes part of it available for evaluation as bounded local components: the effective-dated metering read model, replacement continuity, immutable measurement versions, three VEE rules with bounded interpolation, Usage and provisional determinants, local delivery with acknowledgement reconciliation, a lineage witness with sandbox replay and a technical energy balance service.

The browser editing, approval and settlement screens are demonstration interfaces, and the retained prepayment witness uses deterministic test tokens rather than production STS. A utility deployment would need exact contract versions, adapters, authentication and deployment conditions confirmed, and no capacity or availability figure is published. Document edition 2026.10 identifies this document and is not a commercial release.

To request the datasheet or speak with the ARCHAI team, use the download and contact links that accompany this catalogue.