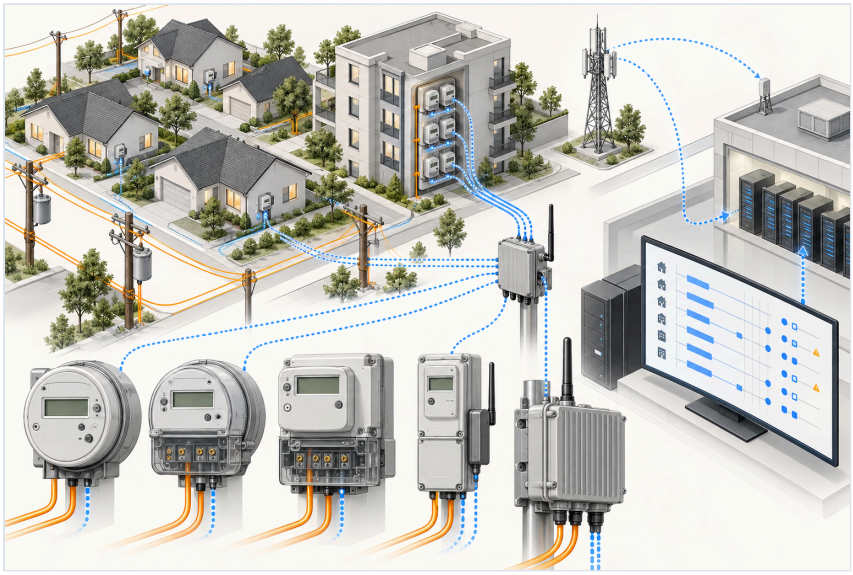
AETHERONE / AETHERHES

AetherHES Datasheet

Head End System (HES) for device communication, collection and command execution · Document edition 2026.10



CONCEPT ILLUSTRATION; NOT A CUSTOMER INSTALLATION OR RUNTIME EVIDENCE.

Concept illustration of an AMI acquisition path: smart meters in houses and an apartment-block meter cabinet, meters of several designs, pole-mounted data concentrators with antennas and a cellular tower, with blue communication paths leading to head-end server racks and a collection monitor. It shows the path AetherHES serves and is not a customer installation or a record of operation.

Selection summary

AetherHES is the head end of AetherONE: the product that holds the conversation with smart meters, data concentrators (DCU) and gateways. For each device it keeps a communication identity and protocol session, runs reads and collection, records what happened to every command, tracks link telemetry and device events, delivers firmware and configuration, and stores the raw communication record (Raw Chronicle) in a private store. What leaves HES is a typed, decoded measurement event with references and hashes. The raw bytes stay behind.

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

Intended readers

- **Integration engineers** who follow a reading from a device frame to a decoded event and an MDMS receipt.
- **Operations engineers** who must tell a device acknowledgement from a matching read-back, and a read-back from a final business outcome.
- **Communication-network and fleet operators** who evaluate fleet, link and firmware views.
- **Solution architects** who need to see where HES ends and where MDMS, GIS, Revenue and Dispatch begin.

Capabilities

Capability	What AetherHES does	In this edition
Adapter and session gateway	Accepts device frames through a local TCP adapter using newline-delimited JSON. Keeps each session open or closed, bound to its device, in an HES-owned SQLite WAL store, and resumes after a restart.	Available for evaluation as a single-host local component.
Collection and Raw Chronicle	Writes every received frame to a transactional raw chronicle with a durable outbox. Produces a versioned decoded measurement event with a computed raw-frame hash, a payload hash and references. Publishes through a separate broker spool with replay-safe delivery and persisted acknowledgement.	Available for evaluation on a single host. The decoded payload is a bounded kW reading, not a general OBIS or load-profile model.
Read-only live view	Serves the Fleet and Collection pages from a loopback, GET-only API with a server-side field allow-list. Deterministic demonstration data appears only when explicitly requested and never replaces a failed live read.	Local evaluation view, not an authenticated production service.
Command Execution Queue	A resident single-instance pump reads requests from one spool and appends ordered status evidence to another. It consumes each request idempotently, binds the target to the observed session and records transport acknowledgement, device acknowledgement and matching read-back.	Available for evaluation as a test-mode token-delivery flow on a deterministic adapter, with local evidence only.
Operations overview, network, firmware, events and alarms	Dashboard, topology and link quality, firmware repository, campaigns and configuration drift, event stream, alarm inbox and raw-frame viewer.	Demonstration interfaces on synthetic data that show the intended operating picture.

How a command is judged

A command moves through eight stages: requested, authorized, queued, dispatched, transport-acknowledged, device-acknowledged, read-back verified and business outcome confirmed. The chain is built around the gap between the sixth stage and the eighth. A device that acknowledged a command has not necessarily changed state, and a changed state is not yet a business result, so device acknowledgement is never a final state. In the test-mode token flow, HES reports execution evidence and only AetherMDMS projects a completed business outcome, after a matching read-back.

Integration and private-writer responsibility

HES writes only to its own store and neither reads nor writes the AetherMDMS private database. Whatever crosses a product boundary travels as a versioned contract or a read-only projection.

Counterpart	What crosses the boundary
AetherMDMS	Decoded measurement events through hes.measurement.decoded.v1: typed fields, including a bounded kW value, plus references and hashes. In the test-mode token exchange, MDMS sends mdms.token.delivery-requested.v1 and HES answers with hes.token.delivery-status-changed.v1. Raw frame text, private event JSON, acknowledgement JSON and token material stay inside HES and appear in neither browser-facing projections nor MDMS.
AetherGIS	Owns spatial truth. HES network and topology views are projections of it.
AetherRevenue	Owns financial billing and settlement. HES reports execution evidence only and carries no financial authority.
AetherDispatch	Owns utility field work. The Command Execution Queue is the device-facing execution surface of HES, not field dispatch.
AetherUtilityConsole	Presents HES inside the unified frontend through navigation and read-only composition. It is not a shared backend.

Edition and evaluation conditions

AetherHES has an approved design scope, and this edition makes part of it available for controlled local evaluation. The components provided are built for one host and a deterministic simulated adapter: the session gateway, the Raw Chronicle with decoded events and a broker spool, read-only Fleet and Collection views, and a test-mode command flow. The operations-center views are demonstration interfaces that show the intended operating picture. Production DLMS, IDIS and SMGW protocol packs, vendor certification, operator authentication and approval, high availability and live field integration are outside this edition, and the command flow is a test-mode token delivery, not production STS. No throughput, latency or capacity figure is published, and the Raw Chronicle is a transactional local store that is not presented as tamper-proof.

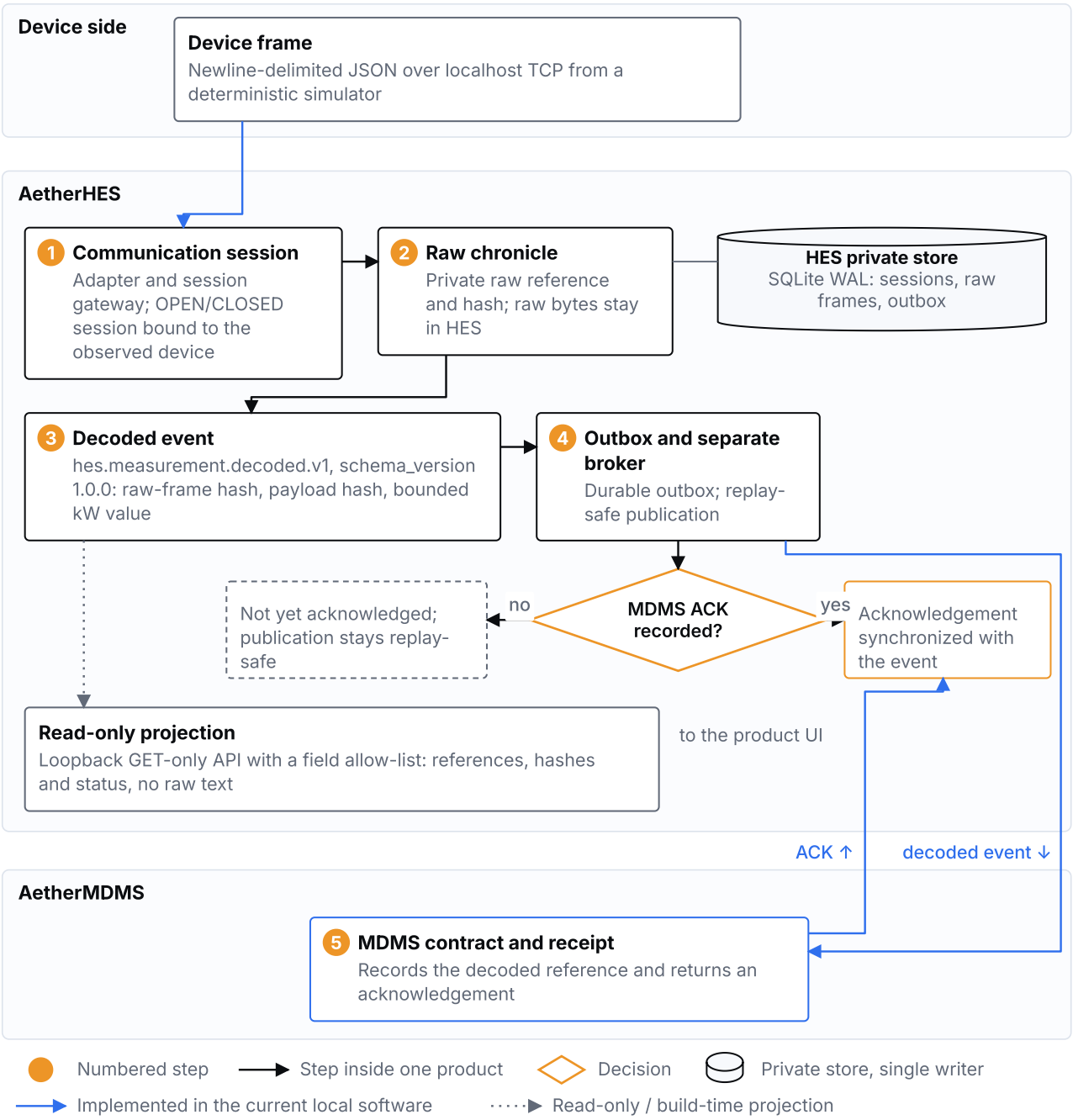
Technical specifications

- **Contracts.** hes.measurement.decoded.v1 (schema_version 1.0.0); mdms.token.delivery-requested.v1; hes.token.delivery-status-changed.v1.
- **Packages and runtime.** The web package is private, version 0.1.0, and is not a commercial release. The web interface builds on Node.js 20.19 or later in the 20 line, or on 22.12 or later. The measurement service needs Node.js 24 or later.
- **Command pump.** Enabling it requires an explicit shared broker path and paired token request and status spool settings.
- **Document edition.** 2026.10 identifies this document. It is not a product release.

AetherHES: one reading, from session to MDMS receipt

DESIGN WORKFLOW

Current local slice: one host, one adapter, a deterministic frame.



DESIGN WORKFLOW Design workflow: communication session, private raw reference and hash, decoded event, separate broker, MDMS contract and receipt. This is a conceptual design diagram of the intended path of one reading and does not describe a production integration.