AETHERONE / AETHERCIS

AetherCIS

Catalogue

Customer and service relationships

A household is moving out of one house and into another. The person is the same, but the electricity service location is not: one location is being closed out while another is opened, and a smart meter sits at each. Someone at the utility has to know who is served, at which location, under which account and agreement, and what has been asked of them. AetherCIS is the AetherONE component responsible for that customer side of an AMI utility, and for the request that starts any follow-up.



CONCEPT ILLUSTRATION; NOT A CUSTOMER INSTALLATION OR RUNTIME EVIDENCE.

Concept illustration of customer, account and service-location relationships in an AMI utility.

Concept illustration: a household moving between two electricity service locations, each with its own smart meter, with customer, account and service-request cards linked to them. It is not a customer photograph or a named site.

Six objects, one chain of responsibility

The approved design makes AetherCIS responsible for the identity and lifecycle of six objects, from the person to the request. That is a statement of responsibility, not a description of finished business functions.

A **Party** is a person or organization acting in a role. A **Customer Account** is the customer's account inside CIS, neither a financial ledger nor a portal login. A **Premise** is the site, building or unit where service is delivered, and a **Service Point** is the CIS identity of electricity service at that premise: the place where responsibility for service starts and ends when a household moves. A **Service Agreement** records the service relationship and who is responsible for what. A **Service Request** is where new work begins; it is received and routed, and it keeps its own identity from then on.

The smart meter sits beside this picture, not inside it. A Service Point is not a Metering Point. The Metering Point, with its installations and measurements, belongs to AetherMDMS, and AetherCIS may keep versioned references and compact projections of them, not the measurements themselves. Referencing both does not merge them or move writer authority. In the same way a Service Request is not a work order: CIS owns the request, AetherDispatch owns the field work that follows, AetherGIS owns the normalized address and spatial truth, and AetherRevenue owns the financial bill.

Three situations

A household moves

The person stays the same Party. What changes is who is responsible for service at each location: at the old Service Point that responsibility ends, and at the new one it begins, under a Service Agreement and a Customer Account that AetherCIS is designed to keep with effective dates. The two locations are two Service Points at two Premises.

Everything around that change belongs elsewhere. The metering facts for each location, meaning Metering Points, installations and measurements, are AetherMDMS's, so the final reading is a measurement question for AetherMDMS, and any final bill is AetherRevenue's. If someone has to visit, the field work would be a Dispatch work order raised from a CIS request. A service-administration team can walk through these steps in the Move-in/Move-out Wizard and the agreement wizards, using read-only demonstration values. The wizards show the sequence and the responsibilities; they complete no move, which would still need effective-dated lifecycle persistence and approved integration.

A reading is questioned

A customer calls to say the smart-meter reading looks wrong. The agent starts from the customer's side: who is calling, which account, which Premise and Service Point, and which Service Agreement defines the service. Whether the reading is right is a question of measurements, VEE and usage, and AetherMDMS owns those. If a site visit is needed, the field work would be a Dispatch work order, created after CIS has recorded the originating request.

In the evaluation edition that starting point can be rehearsed on the linked synthetic records and the request and detail screens. Handling a real customer inquiry through to a field visit would need actual authenticated CIS data and explicit metering and field projections.

A meter replacement is requested

A customer-service agent takes a meter-replacement inquiry. AetherCIS owns the request: its identity, intake and routing. A versioned design contract describes that hand-off: the request's identity, type, origin, priority, status and creation time.

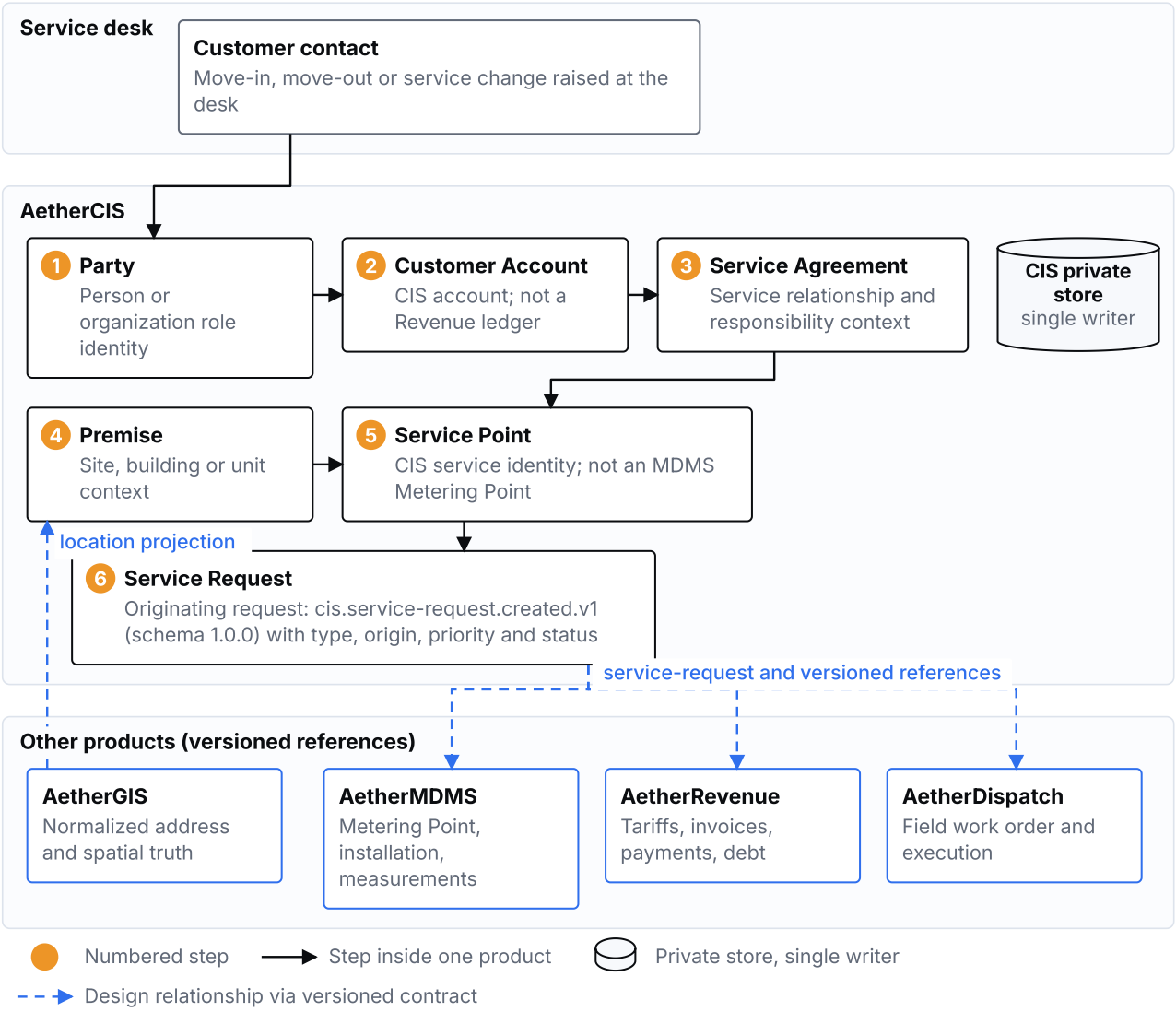
From there, in the approved design, the responsibilities divide. AetherDispatch owns the work order, planning, crew or contractor assignment and the field evidence. After validated completion, AetherMDMS commits the effective-dated meter installation and its continuity. An integration architect or business analyst can use the ownership map in the evaluation edition to check that every fact has exactly one owner. The event contract is a design; no producer or consumer has been demonstrated, so this is an ownership map, not an end-to-end integration.

How the pieces fit

AetherCIS: customer, account and service identity

DESIGN WORKFLOW

Design workflow from service intake to versioned references for other products.



DESIGN WORKFLOW Design workflow of AetherCIS identity and request relationships with the private-writer boundaries of AetherMDMS, AetherGIS, AetherRevenue and AetherDispatch.

Design workflow: the relationships between the CIS objects and the other products as designed, with each product keeping its own private store and writer. These are design relationships, not running integrations.

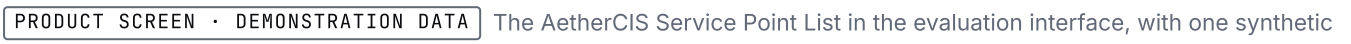
AetherCIS neither reads nor writes another product's private database. AetherUtilityConsole is the unified frontend shell and has no shared backend, and the shared contract and coverage control plane carries contracts, coverage and evidence rather than business records. AetherRevenue keeps its own financial-context account references, and this edition does not state how they relate to a CIS Customer Account.

Supplier switching is the one optional screen. It appears only in selected demonstration profiles, makes no statement about the law or market rules of any jurisdiction and connects to no supplier system.

What you can look at today

1 A service point record and the product that owns it

OBJECT	REFERENCE	OWNERSHIP
Service point	SERVICE-POINT-DEMO-001	CIS



The AetherCIS Service Point List in the evaluation interface, with one synthetic record shown in detail.

The Service Point List in the evaluation interface, showing one synthetic record with its object type, reference and owner (CIS). The wider workspace covers customer and party search, customer 360, party roles, duplicate resolution, customer accounts, premises, service address validation, agreements, move-in and move-out, and service requests.

The responsibilities above are the approved product design. What is available today is a read-only evaluation interface running on synthetic demonstration data. It does not perform live CIS business operations: it has no backend connection, authentication or persistence, and its forms and actions only name the integration they would need. By default it shows no records and says that live data is unavailable, and synthetic data appears only when demonstration mode is chosen explicitly. Values from the other products are typed demonstration references.

A real operational deployment would additionally need a CIS backend, authentication and authorization, a private store with persistence, and approved integration contracts. Document edition 2026.10 is not a product release, and no production deployment, customer installation or measured capacity is claimed.

Contact the ARCHAI team to discuss AetherCIS.