SYSTEM ARCHITECTURE / EXAMPLE 02

02

ILLUSTRATIVE CONTENT · NOT AN ACTUAL CLIENT REPORT

System Design Specification

A technical design format for architecture boundaries, data contracts, failure handling, deployment and operational acceptance.

PREPARED BY
Carlo Hennsa

DOCUMENT REFERENCE
EX-SD-002

VERSION / DATE
1.0 · 14 October 2026

CONTEXT
Aster Sync (fictional service) · Illustrative example

01

DOCUMENT SECTION

Overview and boundaries

Specify behaviours, not simply product names

Aster Sync is a fictional local-first event service for applications that must record events while disconnected and reconcile them when network access returns.

PRIMARY OUTCOME

Reliable local capture

Write accepted events locally even when the network is unavailable.

CONSISTENCY MODEL

Eventual, per aggregate

Server reconciliation is authoritative; clients observe an explicit status.

KEY DESIGN PROPERTY

Idempotent replay

Re-sent events never produce repeated business actions.

OUT OF SCOPE

Real-time global ordering

No claim of total ordering across devices.

02

DOCUMENT SECTION

Quality attributes and constraints

Show non-functional requirements alongside behaviour

ID	Requirement	Verification method
NFR-01	Local event acceptance does not depend on an internet connection.	Disconnect test while creating events.
NFR-02	Each event has a globally unique identifier and a schema version.	Schema validation and duplicate replay test.
NFR-03	Every server acknowledgement is durable before the client marks an event synchronized.	Crash and restart test.
NFR-04	A replay with the same event identifier cannot apply a business action twice.	Idempotency integration test.
NFR-05	Conflicts and invalid messages are visible to operators.	Quarantine and alert drill.

03

DOCUMENT SECTION

Logical architecture

Four components; ownership explicitly separated

01 · Client application

Captures action and status; displays pending / accepted / rejected

02 · Local durable store

Write-ahead log, device sequence, retry queue

03 · Ingestion API

Auth, schema checks, event ID deduplication and acknowledgement

04 · Reconciliation

Applies business rules and publishes outcomes

CLIENT RESPONSIBILITIES

Capture and retransmit

- Assign event IDs before local commit.
- Persist requests before showing success.
- Retry with exponential backoff.
- Show queued, accepted and rejected states.

SERVICE RESPONSIBILITIES

Validate and reconcile

- Authenticate sender and verify schema.
- Deduplicate using event ID.
- Evaluate version and business constraints.
- Publish a traceable final outcome.

04

DOCUMENT SECTION

Message and API contract

Example wire format and processing states

```
{
  "eventId": "024f39c8-1b78-4e18-85f6-3ca054d8ab9b",
  "aggregateId": "asset-204",
  "deviceId": "device-07",
  "sequence": 42,
  "schemaVersion": 1,
  "occurredAt": "2026-10-14T07:00:00Z",
  "type": "InspectionCompleted",
  "payload": { "result": "Pass", "evidenceCount": 2 }
}
```

API	Expected behaviour
`POST /v1/events`	Durably accept, reject with a reason, or report already accepted.
`GET /v1/events/{id}`	Return current ingestion and application state.
`GET /v1/sync?cursor=`	Return scoped incremental changes with a durable cursor.

CONTRACT RULE

An HTTP acknowledgement only establishes that the message was durably accepted. It does not imply that every downstream business rule succeeded.

05

DOCUMENT SECTION

Failure modes and recovery

Design for the scenarios users actually experience

Scenario	Expected system response	Evidence
Client restarts mid-upload	Persisted event remains queued or accepted; safe retry on restart.	Restart test / dedup log
Server accepts but reply is lost	Client retries; same event ID returns existing outcome.	Network fault test
Conflicting updates	Conflict is explicitly rejected or queued for resolution.	Conflict report
Schema version unsupported	Reject safely with a machine-readable error; preserve event locally.	Contract test

Scenario	Expected system response	Evidence
Device clock incorrect	Server records receipt time separately; policies define handling.	Clock skew test

06

DOCUMENT SECTION

Operations, security and rollout

Make ownership and testability part of the design

OBSERVABILITY

Queue age

Measure oldest pending event, retry count and reconciliation latency.

SECURITY

Least privilege

Per-device credentials, revocation, encryption and audit context.

RECOVERY

Restore drill

Rebuild one node, replay retained events and compare expected outcomes.

DECISION PENDING

Choose the transactional persistence technology and conflict resolution rules after load, failure and recovery tests. Product names in a diagram are not architectural approval.

07

DOCUMENT SECTION

Document control

Example technical design record

Field	Value
Author	Carlo Hennsa
Document type	System design (illustrative)
Reference	EX-SD-002
Version	1.0
Issued	14 October 2026
Decision status	Proposed; requires engineering review
Scope	Fictional local-first synchronization service