T1

Tier 1 · Digital Backbone

Technical Specification

Block Architecture, Data Contracts, and Integration Reference

Version	1.0
Status	FINALIZED
Zone	OT Zone
Standards	ISA-95 / IEC 63278 AAS / W3C RDF
Block Count	6 blocks
Document Date	May 2026

1. Technical Overview

Tier 1 implements a six-block linear enrichment pipeline operating exclusively in the OT zone. All inter-block data follows a single enriched message contract. Each block is implemented as a Node-RED function node with optional companion nodes (sqlite, http-request, mqtt-out) as declared in the block type field.

1.1 Message Contract

Every message exiting T1 conforms to the following schema:

```
msg = {
  topic:       string,          // Original MQTT topic
  payload: {
    EquipmentProperty: { ... }, // B2MML structure from Ingress
    tagId: string,              // Resolved tag identifier
    value: any,                 // Normalized value
    eu: string,                 // Engineering unit
    quality: string,            // OPC-UA QualityCode
    timestamp: string           // ISO 8601
  },
  assetContext: {
    asset: ISA95Asset,
    parentChain: ISA95Asset[],
    isaLevel: string,
    attributes: object
  },
  tagModel: TagDefinition,      // Injected by Tag Model Manager
  context: MESContext           // Injected by MES Metadata Registry
```

1.2 Block Dependencies

Block	Dependency
Telemetry B2MML Ingress	None (pipeline entry point)
Asset Registry	Requires Telemetry B2MML Ingress upstream; SQLite database populated
MES Metadata Registry	Requires Asset Registry upstream; MES endpoint reachable
Tag Model Manager	Requires Asset Registry upstream; tag_definitions table populated
Digital Twin Sync	Requires Tag Model Manager upstream (validated tags only)
Asset Graph	Requires Asset Registry upstream; invoked on-demand by T2

2. Block Specifications

2.1 Telemetry B2MML Ingress

ID	telemetry-ingress
Type	universal-in + b2mml-wrapper
Standard	B2MML V0600 / Multi-Protocol
Input	Raw protocol payload (MQTT msg, OPC-UA DataValue, Modbus register array, HTTP body)
Output 1	B2MML EquipmentProperty normalized message
Output 2	Raw protocol-specific message (debug)
Output 3	Communication fault object { protocol, endpoint, error, timestamp }
State	Stateless — no internal buffer

B2MML Output Schema:

```
msg.payload = {
  B2MML_Header: {
    Version: "V0600",
    Protocol: "MQTT|OPC-UA|Modbus|HTTP",
    Source: cfg.endpoint
  },
  EquipmentProperty: {
    ID: sourceId,
    Value: { ValueString: rawValue, UnitOfMeasure: uom },
    QualityCode: "192",   // Good = 192 per OPC-UA spec
    Timestamp: "2026-05-15T10:00:00.000Z"
  }
}
```

2.2 Asset Registry

ID	asset-registry
Type	function + sqlite + http-request
Standard	ISA-95 Part 1 & 2
SQLite schema	assets(asset_id TEXT PK, name TEXT, isa_level TEXT, parent_id TEXT, uns_topic TEXT, attributes TEXT, status TEXT)
Cache	LRU cache, TTL = cacheTtlSec (default 300s)
REST endpoints	GET /assets/:id, GET /assets, POST /assets, PUT /assets/:id, DELETE /assets/:id
Output 1	msg.assetContext = { asset, parentChain[], isaLevel, attributes }
Output 3	msg.unresolvedPath = topic prefix that had no match

2.3 MES Metadata Registry

ID	mes-metadata-registry
Type	function + connector + cache
Standard	ISA-95 Part 2 — Canonical Data Model
Cache TTL	cacheTtl seconds (default 60)
Protocols	REST (default), GraphQL, OData v4, SQL (via node-red-node-mysql or similar)
CDM Output	msg.context = { lotId, orderId, recipeId, productSKU, operatorId, shiftId, enrichedBy }
Fault handling	Port 3 on timeout; fallbackToMock returns static mock context when true

2.4 Tag Model Manager

ID	tag-model-manager
Type	function + sqlite + schema
Standard	ISA-95 / IEC 63278 AAS semantic IDs
DB schema	tag_definitions(tag_id TEXT PK, name TEXT, data_type TEXT, eu TEXT, range_min REAL, range_max REAL, semantic_id TEXT, status TEXT, created_at INTEGER)
Validation order	1. Status check, 2. Naming regex, 3. Data type, 4. Range bounds
Violation codes	TAG_UNKNOWN, TYPE_MISMATCH, RANGE_VIOLATION, NAMING_VIOLATION, STATUS_DEPRECATED
Output 1	msg.tagModel = full tag_definitions record

2.5 Digital Twin Sync

ID	digital-twin-sync
Type	function + mqtt + http-request
Standard	IEC 63278 AAS v3 — SubmodelElement
Twin storage	In-memory Map<twinId, AASDigitalTwin> + JSON persistence at twinStorePath
Deadband formula	|new - current| / current * 100 < deadbandPct — skip update
Change history	Ring buffer, last 100 entries per twin
Bidirectional	When bidirectional=true, writes to Eclipse BaSyx AAS Server via REST

Output 1	{ twinId, changed: { semanticId, prev, next }, twinSnapshot }

2.6 Asset Graph

ID	asset-graph
Type	function + graph + http
Standard	W3C RDF / SPARQL-aligned property graph
Graph structure	Map<nodeId, { nodeId, nodeType, unsTopic, edges: Edge[] }>
Edge schema	{ type: string, targetId: UUID, weight?: number, attrs?: object }
BFS output	{ nodes[], wildcards[], impactArray[], rootId }
Impact Array	All nodes with nodeType === "ControlModule" in the BFS subtree
Persistence	JSON file at graphStorePath; reloaded on startup if autoSyncFromRegistry=true

3. Configuration Reference

Block	Parameter	Type	Default / Notes
Block	Parameter	Type	Default / Notes
Ingress	protocol	enum	MQTT | OPC-UA | Modbus TCP | HTTP REST
Ingress	b2mmlVersion	string	V0600
Ingress	defaultUoM	string	units
Ingress	scanRate	string	1000ms
Asset Registry	dbPath	path	/data/assets/registry.db
Asset Registry	cacheTtlSec	number	300
Asset Registry	httpApiPort	number	3001
MES Metadata	protocol	enum	REST | ODATA | GRAPHQL | SQL
MES Metadata	cacheTtl	number	60 (seconds)
Tag Model	namingConventionRegex	regex	^[A-Z][a-zA-Z0-9_]{2,63}$
Tag Model	autoRegister	bool	true
Tag Model	approvalRequired	bool	false
DTS	deadbandPct	number	1.0 (percent)
DTS	bidirectional	bool	false
Asset Graph	maxTraversalDepth	number	10
Asset Graph	autoSyncFromRegistry	bool	true

4. Integration Points

4.1 T1 → T2 Interface

T1 emits enriched messages to T2 Connectivity Hub via MQTT on topic twins/state-changes (published by Digital Twin Sync). T2 MQTT Bridge subscribes to this topic and forwards to the UNS publisher pipeline. The Asset Graph is invoked by T2 Action Logic Engine by injecting a traversal request message directly into the Asset Graph node input.

4.2 T1 → T3 Interface

T3 Historian Proxy subscribes to twins/state-changes (the same DTS output topic). It consumes only change events and applies its own deduplication layer before writing to storage. T3 never reads directly from the T1 SQLite databases.

4.3 External Systems

Eclipse BaSyx AAS Server	Optional. Enabled by bidirectional=true in DTS. REST API on port 4001.
CMMS / CAFM systems	Can query Asset Registry REST API on configured httpApiPort.
Any ISA-95 MES	Connected via MES Metadata Registry using REST, OData, GraphQL, or SQL.
Node-RED Dashboard	Can subscribe to twins/state-changes MQTT topic for live twin visualization.