T3

Tier 3 · Enterprise Intelligence & Trust

User Manual

Operating Historian Writes, Operational Analytics, and Security Controls

Version	1.0
Status	PRODUCTION
Zone	IT Zone
Standards	InfluxDB 2.x / ISO 22400 OEE / CEF SIEM / OAuth2 RBAC
Block Count	5 blocks
Document Date	May 2026

1. Introduction

Tier 3 is the intelligence and trust layer of the IIoT Framework. It operates in the IT zone and performs three distinct functions: durable time-series storage via the Historian Proxy, operational analytics (OEE, Anomaly Detection, Trend Analysis), and security enforcement (Audit Logger, IAM/RBAC Engine).

T3 consumes T1 Digital Twin Sync state-change events from the twins/state-changes MQTT topic. It never reads raw field data — only clean, validated, deduplicated data from T1 reaches T3 storage.

1.1 Design Principles

- Validated writes only — Historian Proxy writes exclusively from DTS state-change events.
- Deduplication at ingress — no identical value/timestamp pair is written twice.
- Statistical analytics — OEE and anomaly detection use proven statistical methods, no heavy ML dependency.
- Closed-loop feedback — anomaly events from T3 flow to T2 Action Logic Engine.
- CEF-compliant audit trail — every event is logged in a SIEM-compatible format.
- Flow-level access control — RBAC Engine guards all API and MQTT access.

2. Historian Proxy

The Historian Proxy is the only T3 block that writes to persistent storage. It acts as a smart buffer, batching writes and suppressing duplicates before committing to InfluxDB 2.x or PostgreSQL.

2.1 InfluxDB Configuration

1. Set influxUrl to your InfluxDB 2.x endpoint (e.g., http://influx:8086).
2. Create a bucket in InfluxDB and set influxBucket and influxOrg to match.
3. Generate an API token in InfluxDB and store it securely (use Node-RED credentials, not plain text).
4. Set batchSize (default 100 points) and flushIntervalMs (default 5000ms).

2.2 PostgreSQL Configuration

5. Set pgConnStr to your PostgreSQL connection string.
6. Create the historian schema: CREATE TABLE tag_values (asset_id TEXT, semantic_id TEXT, value DOUBLE PRECISION, prev DOUBLE PRECISION, ts BIGINT);
7. Both InfluxDB and PostgreSQL can be active simultaneously — useful for dual write to time-series and relational stores.

2.3 Deduplication Window

The deduplicateWindow parameter (default 1s) defines the sliding window within which identical values from the same tag are suppressed. If a tag publishes the same value twice within 1 second, only the first write reaches storage.

3. OEE Calculator

Calculates Overall Equipment Effectiveness per the ISA-95 / ISO 22400 standard on a configurable tumbling window. OEE = Availability × Performance × Quality.

3.1 Input Data Requirements

The OEE Calculator requires the following fields in windowData, populated by aggregation over the window period:

downtime	Total planned downtime in seconds within the window
actualOutput	Units produced during the window
targetOutput	Target production rate × window duration (from MES recipe)
defectCount	Number of rejected or reworked units
totalProduced	Total units produced including defects
start	Window start timestamp (ISO 8601)

3.2 Alert Configuration

When OEE drops below oeeThreshold (default 0.75), Port 2 fires with an alert object. Wire this to the T2 Action Logic Engine to trigger automatic corrective actions, or to a notification node for operator alerts.

4. Trend & Anomaly Detection

Detects anomalous tag values using Z-Score statistics and linear regression trend analysis. Anomaly events emitted on Port 2 feed directly into the T2 Action Logic Engine to close the control loop.

4.1 Algorithm Selection

z-score (default)	Best for normally distributed sensor data (temperature, pressure, speed). Flags values more than zThreshold standard deviations from the rolling mean.
iqr	Better for skewed distributions or data with outliers. Uses interquartile range — less sensitive to extreme values in the training window.
trend	Flags abnormal rates-of-change using linear regression slope. Useful for detecting slow drift before an outright fault.

4.2 Window Sizing

- windowSize (default 60): number of samples in the rolling statistics window. Larger windows are more stable but slower to adapt.
- minDataPoints (default 30): the block passes all messages through without anomaly checking until this many samples are collected. Prevents false positives during startup.
- trendWindowSize (default 20): number of recent samples used for the linear regression slope calculation.

4.3 Connecting to the Action Logic Engine

Wire Trend & Anomaly Detection Port 2 (anomaly output) to T2 Action Logic Engine input. Ensure the anomaly message includes impactArray from T1 Asset Graph — typically injected by a preceding Asset Graph lookup node in the flow.

5. Audit Logger

Every security-relevant event — authentication decisions, MQTT publishes, configuration changes, anomalies, and command issuances — should flow through the Audit Logger before reaching its destination.

5.1 CEF Format

All events are written in CEF (Common Event Format), compatible with Splunk, IBM QRadar, and Elastic SIEM:

```
CEF:0|NodeRED-IIoT|NodeRED|1.0|AUTH_FAIL|Login Failure|8|
src=192.168.1.42 act=LOGIN outcome=failure msg=Invalid credentials
rt=1747300000000
cs1=<SHA256 of this entry combined with previous entry hash>
```

5.2 Log Rotation

Set rotateDaily=true to rotate the log file at midnight. Archived files are named audit.log.YYYY-MM-DD. Ensure the log directory has sufficient disk space for your retention period.

5.3 SIEM Integration

The Audit Logger publishes every CEF event to the auditTopic MQTT topic (default security/audit). Configure your SIEM to subscribe to this topic. Alternatively, tail the log file using a syslog forwarder.

6. IAM / RBAC Engine

The RBAC Engine enforces role-based access control on every API call and MQTT operation. Wire it as the first node after any HTTP-in or MQTT-in node.

6.1 Role Store Configuration

Create /data/security/roles.json with users and ACL entries:

```
{
  "users": {
    "alice": { "role": "engineer" },
    "bob":   { "role": "operator" }
  },
  "acl": {
    "engineer": [
      { "pattern": "factory/#", "actions": ["read","write","execute"] }
    ],
    "operator": [
      { "pattern": "factory/+/+/+/+/+", "actions": ["read"] }
    ],
    "readonly": [
      { "pattern": "factory/#", "actions": ["read"] }
    ]
  }
}
```

6.2 JWT Configuration

Set jwtSecret to a strong random secret (minimum 32 characters). Clients must include a valid JWT Bearer token in the Authorization header of all API requests. The JWT sub claim is used as the userId for ACL lookups.

6.3 Denial Logging

Set auditDenials=true to forward all denied requests to the Audit Logger automatically. This is enabled by default and should not be disabled in production.

7. Troubleshooting

Symptom	Cause	Check	Fix
Symptom	Likely Cause	Check	Fix
Historian writes stop	InfluxDB token expired or bucket full	InfluxDB UI / logs	Regenerate token; configure retention policy
All values show as duplicates	deduplicateWindow too large	Actual tag update rate vs window	Reduce deduplicateWindow to match tag rate
OEE always 0 or 1	windowData not populated	Aggregation node upstream	Add aggregation node to collect downtime/output data
No anomalies detected	minDataPoints not reached	Window fill progress	Reduce minDataPoints or wait for warm-up period
RBAC denies all requests	JWT secret mismatch	Token signing vs jwtSecret	Ensure token issuer uses same secret
Audit log not growing	logPath permission denied	File system permissions	chmod 755 /data/logs directory