GPT29 Cold Chain Tracking for Freezers

Engineering and Compliance White Paper (Public Edition)

Hardware-led visibility with data integration - no proprietary SaaS platform required

Prepared by Eelink | Version 1.0 | January 2026

This document describes a reference technical approach for deploying the Eelink GPT29 tracker in freezer and cold-chain use cases. Final specifications, supported radios, and regional compliance requirements depend on the purchased configuration and the deployment geography.

Website: eelinktech.com | Contact: info@eelinktech.com

Document Control

Field	Value
Document ID	WP-GPT29-CC-EN-001
Title	GPT29 Cold Chain Tracking for Freezers - Engineering and Compliance White Paper
Edition	Public (external)
Version	1.0
Date	2026-01-22
Audience	Engineering, Operations, Quality/Compliance, and Solution Architects
Product scope	Eelink GPT29 hardware tracker and integration guidance (no Eelink SaaS platform)
Contact	info@eelinktech.com | eelinktech.com

Important: Eelink does not provide an end-user tracking platform in this offer. The integration target is the customer's existing system (IoT platform, TMS, WMS, or a third-party portal). This white paper therefore focuses on device behavior, data contracts, and deployment/acceptance practices.

Table of Contents

1. Executive Summary

Freezer and reefer logistics across long ocean legs and multi-modal handoffs create a predictable set of engineering and compliance challenges: signal attenuation in metal enclosures, long mission duration, low-temperature battery behavior, and the need for audit-ready evidence when temperature excursions, suspected door openings, or handling shocks occur.

Eelink GPT29 is designed as a cold-chain data acquisition device that can be integrated into an existing customer system. It captures location and key environmental signals (temperature, humidity, light exposure events, and shock/vibration events), and transmits them using a configurable reporting policy that balances near real-time visibility with multi-week battery life objectives.

Key design goals addressed in this white paper:

- Support mission profiles that require an operational window of at least 60 days (for example, LATAM to North America/Europe round trips), using configurable reporting and exception-driven escalation.
- Provide near real-time location and condition data without assuming an Eelink-hosted platform; enable direct ingestion into a customer or third-party system through a defined data contract.
- Enable audit and claims workflows by producing time-stamped measurements, event records, and integrity-related metadata (for example, sequence counters, acknowledgements, and optional signing patterns).
- Provide deployment guidance for freezer installations, including sensor placement, RF constraints, and validation/acceptance testing.

This document is intentionally implementation-oriented. It describes reference architectures, recommended data models, and practical deployment checklists. Where numeric parameters vary by configuration or route (for example, radio technology, sampling/reporting intervals, or battery capacity), we provide methods and engineering approaches rather than fixed claims.

2. Scope, Assumptions, and Terminology

In scope

- GPT29 device behavior in cold-chain/freezer monitoring use cases.
- Reference connectivity and data transport patterns (MQTT and HTTPS) over TLS.
- Reference data contract (payload schema, units, timestamps, ordering/deduplication).
- Edge event detection patterns for temperature excursions, light exposure events, and shock events.
- Power and battery-life engineering for long missions.
- Installation and operational procedures, including pilot design and acceptance testing.

Out of scope

- A proprietary Eelink tracking platform (not provided in this offer).
- Customer-specific dashboards and workflows (implemented in the customer's system).
- Regulatory legal advice. This document discusses common compliance patterns but does not replace legal guidance.

Terminology used throughout this document

Term	Meaning (in this document)
Near real-time	Configurable periodic reporting, typically in minutes to tens of minutes. Not continuous second-level streaming.
Mission window	The required operational duration of a deployment without maintenance (for example, battery replacement).
Sampling vs reporting	Sampling is local sensor measurement; reporting is uplink transmission to the backend.
Exception escalation	A policy where reporting increases when an event is detected (for example, temperature excursion).
Store-and-forward	Local buffering of measurements and events when connectivity is unavailable, followed by later upload.

Important clarification - satellite positioning vs satellite communications: GPT29 uses GNSS satellites for positioning. Data transmission is typically via cellular networks. Satellite communications require a different hardware configuration and commercial model.

3. Cold Chain Risk Model and Required Signals

A freezer monitoring deployment is successful when it provides the right evidence at the right time. In practice, most cold-chain failures fall into a small number of patterns. The following table maps common risk patterns to signals that GPT29 can capture and how those signals are typically used in operations and compliance workflows.

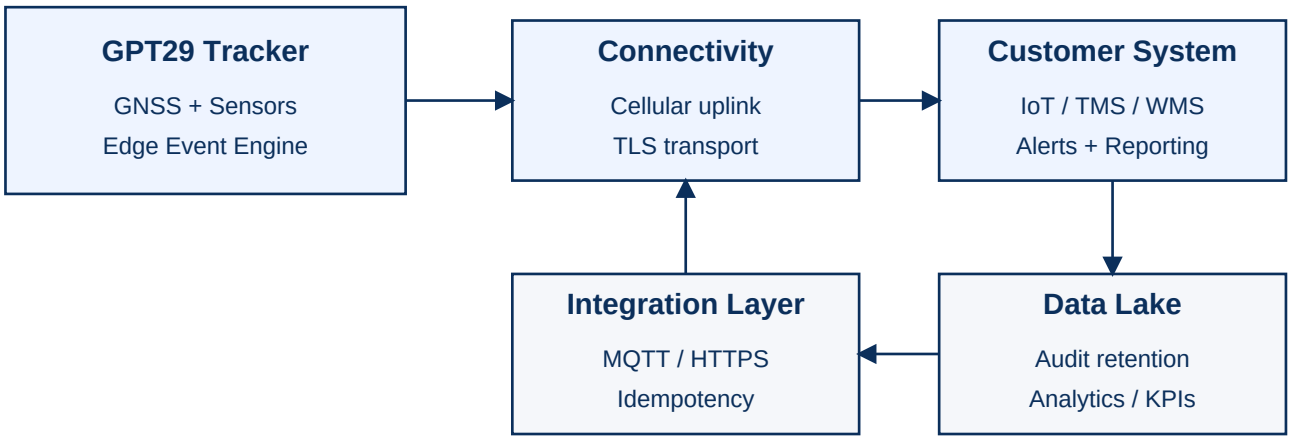
Risk pattern	Signals to capture	How the data is used
Temperature excursion (high/low)	Temperature samples; duration above/below thresholds; optional rate-of-change	Real-time alerts for intervention; post-trip audit of excursion duration; claims support
Suspected door opening / exposure	Light exposure event (threshold + duration); temperature trend correlation; optional shock context	Investigate unauthorized access; correlate exposure to subsequent temperature drift; compliance evidence
Handling shock / drop / impact	Acceleration peaks; shock event count and magnitude; time stamps	Identify abnormal handling; support root-cause analysis and liability discussions
Route deviation / unexpected dwell	GNSS positions; geofence events; dwell time calculations	Detect unplanned stops; investigate cold-chain breaks at ports/terminals
Connectivity gaps	Store-and-forward backlog size; upload latency; radio status metadata	Distinguish 'no event' from 'no data'; maintain audit completeness
Device health degradation	Battery level trends; sensor self-check flags; reboot counters	Preventive maintenance; spares planning; quality assurance

A core principle is separating **measurement** from **reporting**. Sampling may be frequent (for example, every 1-5 minutes for temperature stability analysis), while reporting can be less frequent during stable conditions and increase only when exceptions occur. This separation is key to achieving long mission windows while still producing high-quality evidence.

4. Reference System Architecture (No Proprietary Platform)

Because this offer does not include an Eelink-hosted tracking platform, the architecture assumes direct integration into an existing customer system. The device is responsible for acquiring, validating, and packaging data. The customer system is responsible for visualization, alert routing, ticketing, reporting, and retention policies.

4.1 High-level data flow



Reference architecture - no proprietary SaaS platform required.

Integration objective: provide a stable, auditable data stream that the customer can ingest with minimal custom logic, while still allowing configurable policies (reporting intervals, event thresholds, escalation rules) suitable for cold-chain missions.

4.2 Major components and responsibilities

- **GPT29 device:** sensor sampling, GNSS fixes, edge validation, event detection, payload building, buffering, secure transport.
- **Connectivity:** cellular uplink as configured for the deployment geography; transport secured with TLS; retry policies for intermittent coverage.
- **Customer/third-party backend:** ingestion endpoint (MQTT broker or HTTPS API), decoding and storage, alert routing, dashboards, reports, retention and access control.

In integration projects, the most common causes of delays are not the sensors, but the **data contract**: time stamps, units, ordering/deduplication, and alert semantics. Sections 6 and 10 provide detailed guidance and acceptance criteria.

5. Device Technical Design (GPT29)

This section describes a reference internal architecture for how a cold-chain tracker like GPT29 is typically implemented. Exact component selection varies by product configuration and region; the focus here is on the engineering mechanisms that enable reliable data collection in freezer environments and long-duration missions.

5.1 Functional blocks

- **Positioning:** GNSS engine for periodic fixes; configurable fix cadence and timeout policies.
- **Sensors:** temperature, humidity, light exposure, and acceleration (shock/vibration).
- **Edge compute:** event engine to detect excursions and encode events; local filtering and plausibility checks.
- **Storage:** non-volatile buffering for store-and-forward and audit completeness.
- **Connectivity:** cellular radio (technology depends on configuration); TLS-secured MQTT or HTTPS.
- **Power management:** sleep scheduling, sensor duty-cycling, and escalation logic to meet mission window targets.

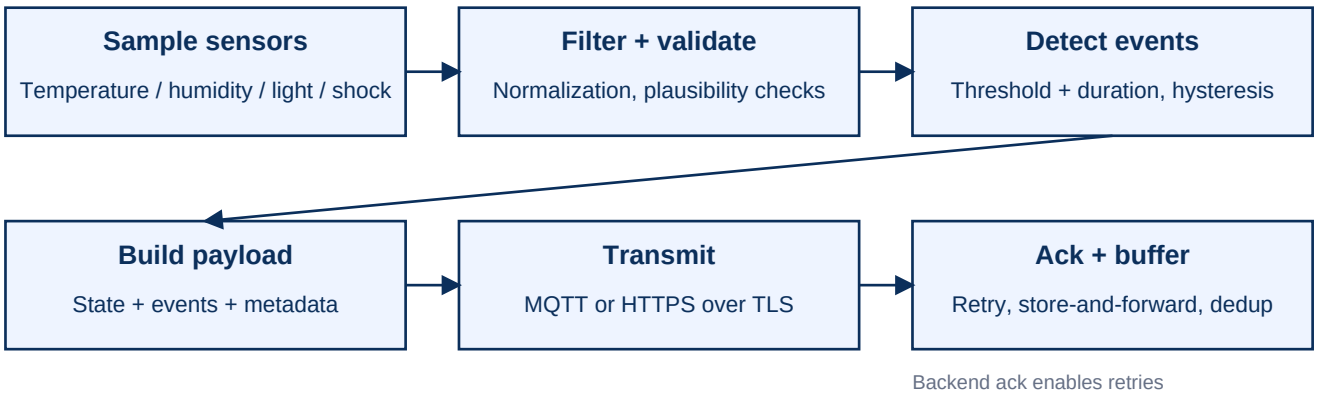
5.2 Sampling, reporting, and event escalation

A cold-chain tracker must reconcile two competing requirements: frequent measurements (to capture transient excursions) and long battery life. A robust implementation therefore uses three concurrent loops:

- **Sampling loop:** measure sensors on a schedule (for example, temperature every N minutes).
- **Event loop:** evaluate measured values against rules (threshold + duration, hysteresis, rate-of-change).
- **Reporting loop:** transmit either periodic status reports or event-driven packets, with escalation during anomalies.

The reporting loop can be configured to shift from low-frequency (for example, hourly) to high-frequency (for example, every 5-15 minutes) for a limited time window when an exception occurs. This limits average power draw while still producing actionable near real-time data during risk periods.

5.3 Reference edge event engine flow



Edge event engine - reference flow for reliability and audit-ready data.

The event engine should be deterministic and auditable. For each event type, define: (1) trigger condition, (2) required persistence duration, (3) hysteresis/reset condition, (4) minimum time between events (debounce), and (5) payload fields required for audit.

5.4 Practical event detection patterns

Temperature excursion

A robust pattern uses threshold plus duration rather than single-sample triggers. A typical rule is: 'temperature above T_high continuously for D_high minutes' or 'temperature below T_low continuously for D_low minutes'. Hysteresis prevents rapid toggling around the threshold.

```
Reference pseudo-logic (illustrative):
if temp >= T_high:
    high_counter += sample_interval
else:
    high_counter = max(0, high_counter - sample_interval)

if high_counter >= D_high and not event_active('TEMP_HIGH'):
    raise_event('TEMP_HIGH', temp, duration=high_counter)

if temp <= (T_high - HYST) and event_active('TEMP_HIGH'):
    clear_event('TEMP_HIGH')
```

Light exposure (suspected opening)

Light exposure is commonly used as an indicator for door opening or container exposure. Because light sensors are sensitive to placement and stray reflections, a good implementation uses both a threshold and a minimum duration, plus optional correlation with temperature slope. In audits, it is often useful to store the raw peak or integrated light value and the duration.

Shock / vibration

For handling risk, trackers often monitor acceleration magnitude and detect spikes beyond a configured G threshold. To avoid over-reporting, a common approach is to record summary features: peak magnitude, event count, and a short 'pre/post' context window. In battery-constrained scenarios, store only peak magnitude and time stamp, and escalate reporting for a short window after a major shock.

5.5 Engineering specification checklist (project SOW)

Cold-chain projects typically require a specification package beyond marketing claims. The table below lists the key items that engineering and compliance teams should confirm during the statement-of-work (SOW) phase. Exact numeric values depend on the purchased configuration and regional certifications.

Category	Specification item to confirm	Notes / verification method
Sensors - temperature	Range, accuracy, resolution, response time	Validate in a temperature chamber and spot-check against a calibrated reference logger
Sensors - humidity	Range and accuracy; condensation behavior	Validate at representative RH and temperature; ensure sensor placement avoids icing
Sensors - light	Units and threshold behavior; baseline in darkness	Perform installation baseline measurement; configure duration threshold to avoid leakage false positives
Sensors - shock	Measurement range; event threshold logic; peak capture method	Pilot-based tuning; document threshold selection and false alert rate
GNSS positioning	Supported constellations; fix strategy; performance in metal environments	Confirm GNSS module capabilities; validate from intended mounting location
Connectivity	Radio technology/bands for target regions; roaming strategy; SIM/eSIM	Confirm by region; test on representative routes and carriers
Battery and power	Battery capacity; replaceable vs sealed; low-temperature performance; mission window under policy	Power budget analysis + route pilot validation; avoid aggressive retries in poor coverage
Environmental	Operating/storage temperature; ingress protection (IP) rating (if applicable)	Confirm with qualification reports or supplier declarations
Security	TLS support; device identity method; credential rotation process (if required)	Align with customer security policies; define provisioning and revocation workflow
Data contract	Payload schema versioning; time stamps (UTC); sequence counters; acknowledgements	Integration test with backend; enforce idempotency and ordering rules
Maintenance	Battery replacement procedure; calibration policy; RMA process	Define spares strategy and turnaround targets for critical routes

6. Reliability Engineering: Store-and-Forward, Ordering, and Deduplication

Long ocean legs and port operations frequently produce gaps in cellular connectivity. A freezer tracker must therefore treat connectivity as intermittent and design for eventual delivery, while still supporting near real-time reporting when coverage is available.

6.1 Store-and-forward buffering model

A reference implementation uses a persistent ring buffer in non-volatile storage. Each record is appended with a monotonically increasing sequence counter and a UTC time stamp. When connectivity is available, the device uploads records in order and waits for acknowledgement. When connectivity is unavailable, the buffer continues to grow until its configured capacity is reached.

- **Record identity:** (device_id, seq) uniquely identifies a measurement or event record.
- **Acknowledgement:** backend returns highest contiguous seq accepted, or an explicit ack list for batched uploads.
- **Retry:** exponential backoff with jitter; avoid synchronized re-connect storms.
- **Overflow behavior:** define a policy (stop sampling, drop oldest, or compress summaries). For compliance use cases, 'drop oldest' should be avoided if possible.

6.2 Ordering and idempotency (backend expectations)

Backend systems must assume that records can arrive out of order or be retransmitted. Therefore, the ingestion contract should be idempotent: processing the same (device_id, seq) twice should not create duplicates.

Recommended backend approach:

- Use a uniqueness constraint on (device_id, seq) in the time-series storage layer.
- Treat device-provided timestamps as the primary event time; store server receive time separately for latency analytics.
- When records arrive out of order, preserve their original timestamps and sequence numbers; do not reorder by receive time.
- If a batch contains gaps, accept what you can and request retransmission of missing seq ranges.

6.3 Practical upload batching

Batching reduces overhead and improves power efficiency. A common pattern is to upload a batch of N records or a maximum payload size limit, whichever is reached first. Batches should include both periodic status samples and event records, but event records should be prioritized for faster delivery when possible.

7. Security and Data Integrity for Audit-Ready Evidence

Cold-chain deployments often support claims and compliance workflows, where the integrity of records matters. The security model should protect both data-in-transit and the authenticity of the device.

7.1 Baseline security controls (recommended)

- **Unique device identity:** each device provisioned with unique credentials (for example, client certificate, token, or pre-shared key).
- **TLS transport:** use TLS for MQTT or HTTPS. Validate server certificates to prevent man-in-the-middle attacks.
- **Least privilege:** device credentials should allow only publish/upload actions, not administrative operations.
- **Secure provisioning:** device identity bound during manufacturing or secure onboarding; avoid shared credentials across a fleet.

7.2 Optional integrity enhancements (for higher assurance)

When audit requirements are strict, additional integrity metadata can be used to strengthen chain-of-custody. The following patterns are common in IoT evidence systems and can be implemented on the backend even when the device performs only basic signing.

- **Sequence counters:** include a monotonically increasing seq to prove ordering and detect missing data.
- **Hash per record:** include a SHA-256 hash of the canonicalized payload to detect tampering.
- **Hash chain:** include hash_prev and hash_curr to create a forward-linked chain of records (tampering breaks the chain).
- **Digital signatures:** sign payload hashes with device private keys (for example, ECDSA). Backend verifies signatures using registered public keys.
- **Time-stamp anchoring:** backend can periodically anchor record hashes to a trusted time-stamp service (implementation choice of the customer system).

These enhancements should be designed with power and compute constraints in mind. For example, computing hashes is typically cheap, while public-key signatures are more expensive. A common compromise is to sign only event records (temperature excursions, suspected opening, major shocks) and use hash chaining for the full stream.

7.3 Data privacy and access control (backend responsibilities)

- Define role-based access control for location data and sensitive customer identifiers.
- Implement retention policies aligned to internal audit and regulatory needs.
- Log all access and modifications to alarm thresholds and geofences; treat them as configuration evidence.

8. Power and Battery-Life Engineering for 60+ Day Missions

Battery life is the most frequently misunderstood requirement in cold-chain tracking. In long-haul routes, a device cannot be both continuously online and maintenance-free for months without careful policy design. The correct approach is to treat battery life as an engineering outcome of a mission profile: sampling cadence, reporting cadence, GNSS fix strategy, radio conditions, and exception escalation rules.

8.1 Mission profile decomposition

A mission profile can be expressed as the sum of energy costs for recurring activities. The device's total energy consumption over a mission window is approximately:

```
E_total ~= E_base + N_fix * E_gnss_fix + N_reports * E_report + N_events * E_event
Battery_life_days ~= Battery_capacity / (E_total_per_day)

Where:
E_base     = quiescent sleep + periodic wake overhead
E_gnss_fix = energy per GNSS acquisition (depends on time-to-first-fix and sky view)
E_report   = energy per uplink transmission (depends on radio technology and signal)
E_event    = additional energy from escalation, extra fixes, or repeated transmissions
```

Rather than hard-coding one interval, the recommended approach is a **two-tier policy**: a low-frequency baseline that meets the mission window, and an exception tier that increases visibility when risk is detected.

8.2 Baseline + exception escalation policy (recommended)

- **Baseline reporting:** transmit a compact status message at a low frequency during stable conditions.
- **Exception triggers:** temperature excursion, suspected opening (light), major shock, geofence exit, prolonged dwell.
- **Escalation behavior:** temporarily increase reporting frequency and/or GNSS fix rate for a bounded time window (for example, 2-12 hours), then return to baseline.
- **Adaptive backoff:** if the radio is in poor coverage, reduce retry aggressiveness to avoid power drain; rely on store-and-forward.

8.3 Low-temperature considerations

Freezer environments can reduce available battery capacity and increase internal resistance, affecting peak current delivery. Practical mitigations include:

- Use battery chemistries and cell designs suitable for low-temperature discharge (specific selection depends on device configuration).
- Avoid repeated high-power radio transmissions in rapid succession when temperature is extremely low; rely on buffering and batch upload.
- Design the reporting policy around the coldest expected soak temperature rather than ambient conditions.

8.4 Example configuration worksheet (illustrative)

Parameter	Baseline (stable)	Exception (during risk)
Sensor sampling interval	e.g., 1-5 min temperature; 5-15 min humidity	same or increased for diagnostics

Parameter	Baseline (stable)	Exception (during risk)
GNSS fix interval	e.g., 30-120 min	e.g., 5-15 min (bounded window)
Uplink report interval	e.g., 30-120 min	e.g., 5-15 min (bounded window)
Retry policy	conservative, backoff + jitter	prioritize event packets; do not drain battery with aggressive retries
Batch size	N records or max bytes	smaller batches for faster event delivery
Event rules	threshold + duration + hysteresis	additional context fields enabled

A pilot should validate battery life under real route conditions. The acceptance section provides recommended KPIs and how to measure them (for example, data completeness and average report energy).

9. Data Integration Without an Eelink Platform

In this offer, the customer (or a third-party) provides the tracking application. Eelink supplies the hardware and supports integration by defining a clear data contract. The goal is to ensure that engineering teams can ingest and operational teams can trust the data stream.

9.1 Integration patterns

Two common ingestion patterns are supported in modern IoT backends. The choice depends on the customer's infrastructure and security policies.

- **MQTT over TLS:** device publishes to customer broker; best for high-throughput streaming and low overhead; supports retained last-known-state if desired.
- **HTTPS over TLS:** device posts batches to a customer API endpoint; simple to integrate with REST-oriented systems; easier to place behind API gateways and WAFs.

9.2 Topic / endpoint conventions (reference)

The following naming conventions reduce ambiguity. Final conventions should be agreed during the integration kickoff.

Transport	Reference path/topic	Notes
MQTT	eelink/gpt29/{device_id}/telemetry	Periodic status samples and derived summaries
MQTT	eelink/gpt29/{device_id}/event	Event records (excursion, light, shock, geofence, etc.)
MQTT	eelink/gpt29/{device_id}/lwt	Optional last-will topic for unexpected disconnect (broker feature)
HTTPS	POST /iot/eelink/gpt29/ingest	Batch ingestion endpoint; returns acknowledgement
HTTPS	POST /iot/eelink/gpt29/ingest/events	Optional separate endpoint for events

9.3 Payload schema (reference JSON)

The payload should be self-describing, unit-consistent, and stable over time. We recommend JSON for readability during pilots. For high-scale deployments, a binary format can be used, but the semantic field dictionary should remain identical.

```
{
  "schema_version": "1.0",
  "device_id": "GPT29-XXXXXX",
  "seq": 123456,
  "timestamp_utc": "2026-01-22T12:34:56Z",
  "received_hint": {
    "timezone": "UTC"
  },
  "position": {
    "lat": 37.7749,
    "lon": -122.4194,
    "source": "GNSS",
    "hdop": 1.2,
    "fix_age_s": 0
  },
  "power": {
    "battery_pct": 78,
    "voltage_v": 3.62
```

```
  },
  "sensors": {
    "temperature_c": -18.7,
    "humidity_rh_pct": 62.1,
    "light_level": 0.02,
    "accel_peak_g": 0.8
  },
  "flags": {
    "tamper_suspected": false,
    "in_buffered_mode": false
  },
  "meta": {
    "fw_version": "x.y.z",
    "report_policy": "baseline"
  }
}
```

Notes: The numeric values above are illustrative only. Field presence and radio metadata may vary by configuration. For audit, the most important fields are device_id, seq, timestamp_utc, and the measured values with units.

9.4 Field dictionary (recommended)

Field	Type	Unit / format	Purpose / notes
schema_version	string	semver-like	Allows evolution without breaking decoders
device_id	string	identifier	Unique device identifier used in provisioning
seq	int	monotonic counter	Ordering, deduplication, missing record detection
timestamp_utc	string	ISO 8601 UTC	Primary event time; do not use local time on device
position.lat/lon	float	degrees	Location when available
position.source	string	GNSS / assisted	Position source classification
sensors.temperature_c	float	degrees C	Primary cold-chain measurement
sensors.humidity_rh_pct	float	%RH	Environmental context
sensors.light_level	float	normalized	Supports suspected opening exposure logic
sensors.accel_peak_g	float	g	Shock context; may be a peak in a window
events[]	array	list	Event records included in the payload (see next section)
power.battery_pct	int	%	Operational planning and health monitoring
meta.fw_version	string	x.y.z	Traceability in pilots and audits

9.5 Event records: structure and semantics

Events should be explicit records rather than implicit interpretations. The backend can then apply business logic (for example, whether a light event constitutes a door opening) without losing the raw facts.

```
{
  "schema_version": "1.0",
  "device_id": "GPT29-XXXXXX",
  "seq": 123500,
  "timestamp_utc": "2026-01-22T14:08:00Z",
  "event": {
    "type": "TEMP_HIGH",
    "severity": "critical",
    "threshold_c": -15.0,
    "duration_s": 1800,
    "max_temp_c": -12.4,
    "context": {
      "policy": "exception",
      "light_peak": 0.0,
      "shock_peak_g": 0.7
    }
  },
  "power": {
    "battery_pct": 77
  },
  "meta": {
    "fw_version": "x.y.z"
  }
}
```

Recommended event types for freezer deployments include: TEMP_HIGH, TEMP_LOW, LIGHT_EXPOSURE, SHOCK, GEOFENCE_EXIT, DWELL_TOO_LONG, LOW_BATTERY, DEVICE_REBOOT, and CONNECTIVITY_GAP.

9.6 Acknowledgement contract (MQTT or HTTPS)

The acknowledgement contract defines reliability. Without it, devices may over-retry (wasting battery) or drop data (breaking audits). A practical pattern is: backend returns the highest contiguous seq ingested successfully.

```
{
  "device_id": "GPT29-XXXXXX",
  "ack": {
    "max_contiguous_seq": 123500,
    "missing_ranges": [
      [
        123450,
        123460
      ]
    ]
  },
  "server_time_utc": "2026-01-22T14:08:05Z"
}
```

When missing ranges exist, the device should prioritize retransmission of those ranges before uploading newer records, unless the backend explicitly supports out-of-order ingestion with gap tolerance.

10. Installation and Deployment for Freezers and Metal Enclosures

Installation quality determines data quality. In freezer and reefer environments, the most common issues are: RF attenuation due to metal, inaccurate temperature measurement due to poor probe placement, and false light/shock events due to mechanical mounting choices.

10.1 Mounting principles

- **Prioritize sensor accuracy:** temperature measurement should reflect the monitored volume or product zone. If a probe is used, place it where it captures the relevant thermal mass.
- **Prioritize connectivity second:** if the enclosure severely blocks RF, consider mounting the device body where it has better sky/cellular view while routing the temperature sensing element to the measurement point.
- **Mechanical robustness:** use fastening methods appropriate for vibration, moisture, and low temperature (industrial-grade mounting, strain relief for cables).
- **Tamper evidence:** use seal/tamper labels or physical covers if a chain-of-custody process requires it.

10.2 Sensor placement guidance

Temperature

Avoid placing the sensor directly against metal walls that track ambient exterior temperature rather than internal freezer temperature. Avoid direct airflow jets that may create local minima/maxima. Prefer a location representative of product temperature. If using a probe, provide cable routing that avoids door seals and moving parts.

Light exposure

Light sensors are sensitive to orientation and gaps. During installation, validate the baseline 'dark' level and configure thresholds to avoid false triggers from minor light leakage. Light events should be interpreted together with temperature trends when used for compliance evidence.

Shock/vibration

Shock detection is strongly affected by how rigidly the device is mounted. A loose mount may amplify vibration, while an overly damped mount may hide impact peaks. Select thresholds during pilot based on observed distributions rather than assumptions.

10.3 Commissioning workflow (recommended)

- Assign device_id to an asset record in the customer system (freezer ID / container ID / shipment ID).
- Configure reporting policy: baseline interval, escalation interval, escalation duration, and retry backoff.
- Configure event thresholds: temperature limits and persistence duration, light exposure duration, shock threshold.
- Perform a cold start test: verify GNSS fix and uplink from the intended mounting position.
- Perform a sensor sanity test: compare temperature reading to a calibrated reference (spot check).
- Lock configuration and record a commissioning report (for audit).

10.4 Installation checklist (field use)

Checklist item	Pass criteria / notes
Mounting secure	No movement under vibration; cables strain-relieved
Temperature measurement validated	Reading consistent with reference measurement and placement rationale documented
RF verification	At least one successful uplink from final mounting location (baseline network conditions)
Light baseline verified	Dark baseline measured; thresholds set to avoid leakage false positives
Shock threshold tuned (pilot)	Threshold selected based on observed distribution; avoid noisy false alerts
Configuration recorded	Policy and thresholds saved with versioning in the customer system

11. Operations, Maintenance, and Fleet Health

After deployment, operational excellence depends on monitoring both the freezer and the device. A typical workflow integrates device telemetry into operations dashboards and ticketing systems.

11.1 Recommended operational telemetry

- **Data completeness:** expected vs received records per device and per route segment.
- **Latency:** event-time to receive-time distribution; track the tail (p95/p99) during ports and ocean legs.
- **Battery trend:** battery_pct trajectory; detect abnormal drain (often caused by repeated retries in poor coverage).
- **Connectivity state:** buffered mode flag, backlog size, last successful uplink time.
- **Reboots and resets:** counters to detect instability.

11.2 Maintenance planning

Maintenance intervals should be based on mission windows and observed battery depletion, not on fixed calendar periods. For high-value cargo, maintaining a small pool of spare devices and probes is recommended.

- Define an end-of-mission procedure: stop tracking, upload remaining buffer, close out the shipment record, and archive reports.
- Perform spot calibration checks for temperature measurement if the use case requires strict traceability.
- Use RMA workflows for devices with abnormal reboot rates, sensor anomalies, or physical damage.

11.3 RMA and spares (recommended approach)

A pragmatic approach is to keep a spare ratio aligned to operational criticality. Spare planning should consider transit times for replacement and the cost of losing visibility during a mission.

12. Pilot Design and Acceptance Criteria

A pilot should be treated as an engineering validation, not only a commercial trial. The objective is to confirm: (1) data completeness under real route conditions, (2) event detection performance, and (3) mission window feasibility for 60+ day requirements.

12.1 Pilot phases (recommended)

- **Phase A - Lab/yard validation:** confirm sensor readings, reporting, and buffering using controlled tests.
- **Phase B - Short route validation:** run a 3-10 day route to validate network behavior and installation.
- **Phase C - Long mission validation:** run a full route representative of production (including ocean leg and ports).

12.2 KPIs to measure

KPI	How to measure	Why it matters
Data completeness	received unique (device_id, seq) / expected seq range	Proof that 'no alert' means stable conditions, not missing data
Event delivery latency	server_receive_time - timestamp_utc (p50/p95/p99)	Determines whether operations can respond in time
Temperature accuracy (spot check)	compare to reference at commissioning and after mission	Supports quality assurance; avoid false excursions
Excursion detection performance	event count + duration vs reference logger (if available)	Reduces false positives/negatives in compliance workflows
Battery drain rate	delta battery_pct per day under route conditions	Validates 60+ day mission feasibility
Buffer upload effectiveness	backlog size vs time; missing range closure	Validates store-and-forward and reliability

12.3 Acceptance checklist (engineering sign-off)

- Integration: backend ingests payloads, enforces idempotency on (device_id, seq), and stores event-time and receive-time.
- Security: TLS configured; device credentials are unique; backend rejects unauthorized devices.
- Data correctness: temperature values are in degrees C; timestamps are UTC ISO 8601; units are consistent across devices.
- Reliability: buffered records upload after coverage returns; missing ranges close within an agreed window.
- Policy: baseline reporting and exception escalation produce the desired trade-off between visibility and battery.
- Documentation: installation placement, thresholds, and policy versions are recorded per mission for audit.

13. FAQ and Clarifications (Engineering/Compliance)

Q: Does GPT29 provide 'real-time' tracking?

A: GPT29 supports near real-time reporting with configurable intervals. The design target is to provide actionable updates in minutes when needed, while preserving long mission windows by using baseline + exception escalation.

Q: Can the solution track temperature, humidity, light, and shock at the same time?

A: Yes - GPT29 is intended for multi-signal acquisition. Sampling and event rules are configured so that critical exceptions trigger escalation while routine sampling remains power efficient.

Q: What battery life can be achieved?

A: Battery life depends on the mission profile (sampling cadence, GNSS fix rate, report interval, coverage quality, and exception frequency). For long ocean missions, the recommended approach is a low-frequency baseline plus bounded escalation during events to target 60+ day windows. Battery-life targets should be validated with a route pilot under representative freezer temperatures.

Q: Do you provide a tracking software platform?

A: Not in this offer. GPT29 integrates with the customer's existing system or a third-party portal through a defined data contract. Eelink can provide integration documentation and engineering support for MQTT/HTTPS ingestion.

Q: How many trackers can be monitored?

A: The practical limit is set by the customer's backend capacity (ingestion throughput, storage, dashboards) rather than by GPT29. Sizing should be based on your fleet size and reporting policy (messages per device per day).

Q: Do we need on-site software training or to send staff to learn the system?

A: Because the tracking application is provided by the customer/third party, training focuses on integration and operations workflows. Typical delivery is remote (workshops, documentation, and integration test support). On-site support can be arranged for larger rollouts if required.

Q: Is the documentation available in English and Spanish?

A: This white paper is in English. Engineering deliverables (data dictionary, interface guide, and pilot checklist) are typically provided in English; Spanish versions can be prepared as a project deliverable upon request.

Q: What shipping lines or carriers are using this solution?

A: Carrier and customer references are often subject to confidentiality. If you require references for procurement, Eelink can share relevant cases under NDA when available.

Q: Is the positioning based on BeiDou or foreign satellites?

A: Positioning uses GNSS satellites. Multi-constellation support (including BeiDou) depends on the GNSS module configuration of the purchased device. Eelink can confirm the exact constellation support for your bill of materials and region.

Q: Is the device communicating via satellite?

A: GNSS is used for positioning. Data uplink is typically cellular. If your route requires satellite communications in areas without cellular coverage, that is a different configuration and should be evaluated separately.

Q: How is maintenance and repair handled?

A: A typical approach is: track battery health, keep a small spare pool for critical routes, and use an RMA process for devices with physical damage or abnormal behavior. Project-specific maintenance procedures should be defined in the SOW (battery replacement, calibration policy, and turnaround time).

14. Appendices

14.1 Recommended alarm definition template

Alarm definitions should be documented in a way that is both operationally clear and auditable. The following template can be implemented in the customer system and version-controlled:

```
Alarm name: TEMP_HIGH
Purpose: Detect freezer temperature above allowed limit
Trigger: temperature_c >= T_high for D_high seconds (continuous)
Hysteresis: clear when temperature_c <= (T_high - HYST)
Escalation: report_interval = R_high for W_high minutes
Payload fields required: device_id, seq, timestamp_utc, temperature_c, duration_s, max_temp_c
Notification routing: Ops team (email/SMS/ticket)
Audit retention: retain raw samples for N days; retain excursion events for M months
```

14.2 Glossary

Term	Definition
GNSS	Global Navigation Satellite System (for example, GPS, Galileo, GLONASS, BeiDou)
Geofence	A virtual boundary used to trigger entry/exit events
Hysteresis	A margin used to prevent rapid toggling of an event near a threshold
Idempotency	A property where repeating the same operation does not change the result beyond the first application
Store-and-forward	Buffering data locally and forwarding later when connectivity resumes

Contact & Next Steps

For overseas deployments, integration kits, pilot planning, and engineering/compliance discussions, please contact:

Eelink

Website: **eelinktech.com**

Email: **info@eelinktech.com**

What to send us for a fast technical assessment

- Route type (ocean/rail/truck), and expected mission duration (days).
- Expected reporting needs (baseline interval and escalation interval).
- Temperature thresholds and compliance requirements (if any).
- Integration preference: MQTT or HTTPS; target backend system details.
- Physical installation constraints (internal mounting, probe routing, tamper requirements).

This white paper is a public technical reference. A project-specific specification package (interface details, configuration profiles, and pilot acceptance plan) can be provided upon request.