CLOUDNIX TECHNICAL REPORT · CNX-TR-2026-01

Designing IoT Systems for the Field: Monitoring, Control and Identity

Tejaswini, Mohammed Irfan, Avinash KS, Pranoy G N and Avinash Nishant

Cloudnix Software Labs, Bengaluru, India

CNX-TR-2026-01 · Version 1.1 · November 2026

Abstract

Connected systems rarely fail because of the sensor; they fail between the device and the decision. This report describes the engineering patterns we use to close that gap, drawn from systems designed, built and operated by our team: environmental and cold-chain monitoring, confirmed remote control of pumps and valves, water-level control, NFC authentication for people, documents and products, offline-first capture on phones, fleet compliance and process calculations in an iron foundry. We present a five-layer reference architecture, a minimal line-oriented device protocol, a command lifecycle with device confirmation and explicit failure diagnosis, an excursion-based alerting model, a validation chain for NFC taps, and a per-row synchronisation scheme for intermittent connectivity. We discuss security threats and mitigations, sizing and operations, and conclude with an anonymised case study, an implementation roadmap and a 20-point checklist.

Keywords: *Internet of Things; telemetry; remote control; command confirmation; alerting; water-level control; NFC; authentication; offline synchronisation; fleet compliance; cold chain*

Contents

1. Executive summary
2. Why IoT projects struggle in the field
3. Reference architecture
4. Device connectivity and the message protocol
5. Device registry and lifecycle
6. Command and control with confirmation
7. Telemetry, safe bands and alerting
8. Water-level and pump control
9. NFC authentication
10. Offline-first capture and synchronisation
11. Fleet and compliance at the source
12. Process calculations at the point of work
13. Security model
14. Operations, sizing and reliability
15. Case study: student transport safety
16. Implementation roadmap
17. Checklist: 20 questions for your IoT programme
18. Glossary
19. References

1 Executive summary

Connected systems rarely fail because of the sensor. They fail in the space between the device and the decision: an alert nobody trusts, a command nobody can confirm, a reader that loses signal at the exact place the work happens, or a tap that is accepted without checking who made it. This paper describes the patterns we use to close that space, drawn from systems we have designed, built and operated: environmental and cold-chain monitoring, remote control of pumps and valves, NFC authentication for people, documents and products, fleet and fuel capture, and process calculations in a foundry.

Five principles run through every chapter.

1. **Confirm, don't assume.** A reading is confirmed by the platform; a command is confirmed by the device.
2. **One alert per problem.** Alerts follow excursions, not readings.
3. **Quiet devices are visible devices.** Every device has a last-seen time and an online window; quiet devices are flagged and locked out of control.
4. **Validate identity before it counts.** Every tap is checked against validity, context and the reader that made it, and refused taps are logged.
5. **Evidence by default.** Every reading, command, tap and refusal is recorded so that reports for auditors, insurers and parents are a query, not a project.

The chapters that follow go from architecture to protocol, device lifecycle, control, alerting, two domain patterns (water level, NFC), offline capture, compliance, security and operations, followed by an anonymised case study, a roadmap and a checklist.

2 Why IoT projects struggle in the field

Failure mode	What it looks like	Pattern that addresses it
Alert floods	A drifting sensor alerts on every reading; within a day alerts are muted.	Safe bands with excursion tracking (ch. 7)
Unconfirmed commands	The screen says “off”; nobody knows whether the pump heard it.	Command queue with device confirmation (ch. 6)
Connectivity gaps	Readers and devices lose signal where the work happens.	Offline-first capture and sync (ch. 10)
Unverifiable identity	A card is accepted without checking validity or context.	NFC validation chain (ch. 9)
Data without evidence	Readings exist but cannot be produced cleanly for an audit.	Event logging and scheduled reports (ch. 7, 14)
Silent devices	A dead sensor looks like a healthy one reporting nothing unusual.	Device registry, last-seen and online window (ch. 5)

3 Reference architecture

Every system in this paper follows the same five-layer shape. Keeping the layers distinct is what lets a temperature sensor, a pump controller and an NFC reader share one platform without sharing failure modes.

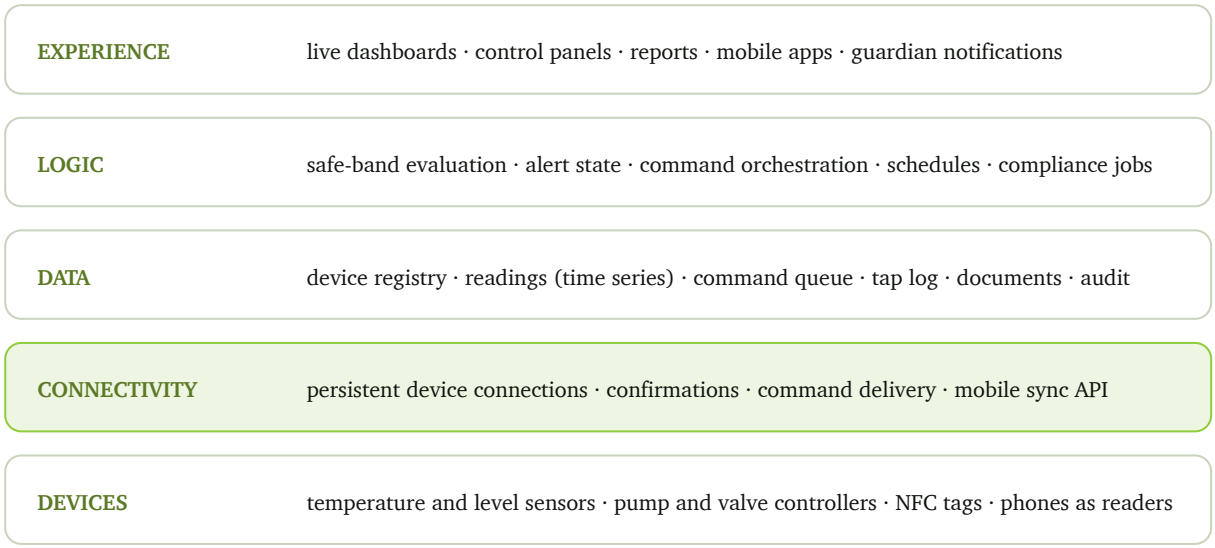


Figure 1. The five layers. Connectivity is the layer most often underdesigned, so it gets its own chapter.

3.1 *Responsibilities by layer*

Layer	Owns	Never does
Devices	Sensing, actuation, local safety interlocks	Decide who gets alerted
Connectivity	Framing, confirmation, delivery, connection health	Business rules
Data	Durable record of every reading, command and tap	Silent overwrites; every change is appended or audited
Logic	Bands, alert state, command lifecycle, compliance rules	Talk to devices directly; it goes through connectivity
Experience	Views and actions by role	Assume a command succeeded before confirmation

3.2 *Deployment shape*

A typical deployment runs the connection service and the web platform on separate processes on the same or separate servers, with a relational database for registry, readings and queues, scheduled jobs for alerts and reports, and a mobile sync API for phone readers. Device traffic arrives on a dedicated port; the web panel is served separately over HTTPS. Backups, monitoring and patching are part of the deployment, not an afterthought (chapter 14).

4 Device connectivity and the message protocol

Low-cost controllers have little memory and intermittent links. The protocol we use for always-connected devices is deliberately small [1, 7]: a persistent TCP connection, one message per line, and comma-separated `key:value` pairs. Every inbound line gets a reply on the same connection, and the same connection carries commands down.

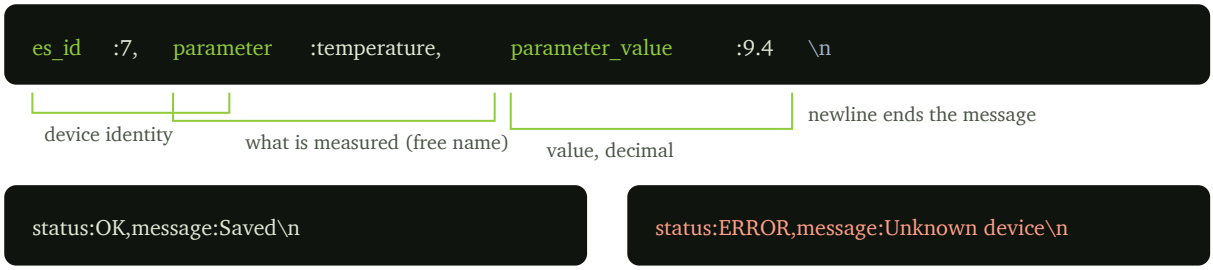


Figure 2. Anatomy of an uplink reading and the two possible replies.

4.1 *Message types*

Direction	Message	Fields	Reply
Device → platform	Reading	device id, parameter, value	status OK, or ERROR with reason
Platform → device	Command	device id, action, response id	device sends a confirmation
Device → platform	Confirmation	response id, status	command marked confirmed
Either	Close	—	connection closed cleanly

```
# a session, as seen on the wire
device   → es_id:7,parameter:temperature,parameter_value:9.4
platform → status:OK,message:Saved
device   → es_id:3,parameter:level,parameter_value:18
platform → status:OK,message:Saved
platform → es_id:12,action:PUMP_START,response_id:912
device   → response_id:912,status:OK
```

Listing 1. Example exchange or schema.

4.2 *Errors the device can act on*

Storage-layer failures are translated into messages firmware can act on, rather than generic errors: an unknown device identity (the device is not registered), or a duplicate reading. A device receiving “unknown device” should stop sending and signal for provisioning, instead of retrying forever.

4.3 *How the connection service is built*

A single process multiplexes all device connections with an event loop (a select-style wait with a short timeout). On each pass it reads complete lines from devices that sent data, writes replies, and checks the command queue for work to deliver. This keeps memory per connection tiny and needs no threads; for larger fleets the same design scales out by sharding devices across several connection services behind a TCP load balancer.

4.4 *Choosing a transport*

Option	Best when	Trade-offs
Persistent TCP line protocol	Mains-powered controllers that must take commands quickly	Simple and fast; needs TLS and device credentials added for public networks
MQTT [1]	Many devices, publish/subscribe, brokers already in place	Rich ecosystem; broker to run and secure
HTTPS polling	Battery devices, readings every few minutes, no inbound commands	Simple through firewalls; commands wait for the next poll
LoRaWAN / NB-IoT [9]	Wide areas, low power, small payloads	Low bandwidth; network operator or gateways required

Hardening for public networks. On untrusted networks, wrap device connections in TLS [7], give each device its own credential issued at provisioning [2, 4], and restrict administrative messages (such as closing the service) to authenticated operators, never to devices.

5 Device registry and lifecycle

Every device is registered before it can report. The registry is the single source of truth for what exists, where it is, what limits apply and who is told.

Field	Purpose
Identity	The id the device sends in every message
Location and physical location	Site, room, tank or vehicle; shown on every alert
Safe minimum and maximum	The band for the measured parameter
Alert list	Who is told about excursions
Status	Enabled, disabled or fault
Last seen	Time of the last message from the device

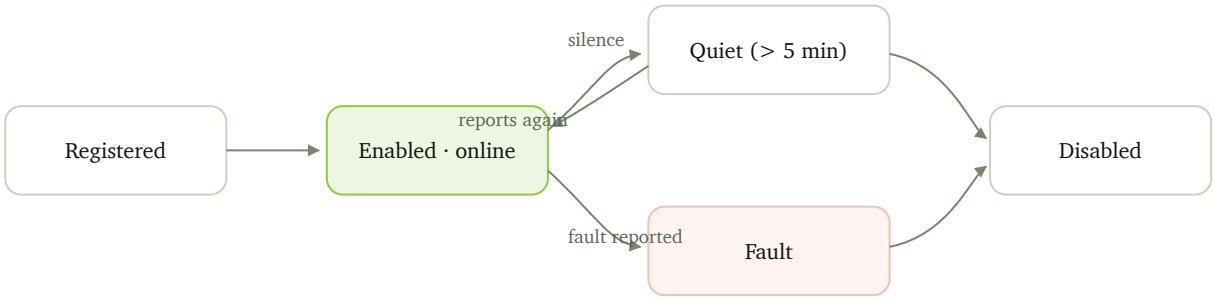


Figure 3. Device lifecycle. “Quiet” is not a stored state but derived from last-seen and the online window.

5.1 *The online window*

A device counts as online if it has reported within the online window (five minutes in our deployments). The panel shows last-seen on every row; quiet devices are highlighted and their controls are locked, so nobody sends a command to equipment that cannot answer.

5.2 *Provisioning steps*

1. Create the device in the registry with location, band and alert list.
2. Issue its identity (and, on public networks, its credential).
3. Install and power on; confirm the first reading appears on the panel.
4. Run a test command and confirm the device's acknowledgement.
5. Enable alerts.

6 Command and control with confirmation

A command is a row in a queue with a lifecycle, not a fire-and-forget packet. The operator sees exactly where it is, and the system never shows equipment in a state the equipment has not confirmed.

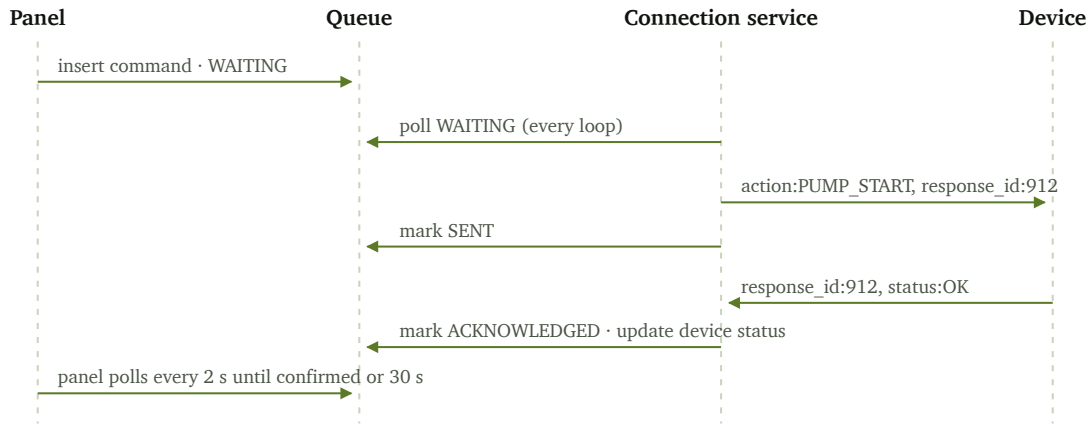


Figure 4. Command sequence. The panel only shows the new state after the device's confirmation is recorded.

6.1 *States and timings*

State	Meaning	Next
WAITING	Queued by an operator, not yet delivered	SENT when written to the device's connection
SENT	Delivered; awaiting the device's confirmation	ACKNOWLEDGED on confirmation
ACKNOWLEDGED	The device confirmed; equipment state updated	Final

Timer	Value	Why
Panel poll	2 s	Feels live without loading the server
Give-up	30 s	Long enough for slow links; short enough for an operator to act
Online window	5 min	Devices quieter than this cannot be commanded

6.2 *Telling failures apart*

At 30 s the command is…	Operator sees	Likely cause
Still WAITING	“No response from the connection service”	Service down, or device not connected
SENT, not acknowledged	“No response from the device”	Device received but did not act, or lost power

6.3 *Good practice*

- **Idempotent actions.** Use absolute actions (“set pump ON”) rather than toggles, so a repeated command is harmless.
- **Expire stale commands.** A command not delivered within a set time should expire rather than execute hours later when the device reconnects.
- **Role checks.** Only roles with control rights can queue commands; viewers see state only.
- **Audit.** Every command records who requested it, when, and each state change.
- **Local interlocks.** Safety-critical limits (for example, a dry-run cut-out on a pump) belong in the device as well as the platform.

7 Telemetry, safe bands and alerting

7.1 *Storing readings*

Readings are stored in a narrow table: device, parameter, value and timestamp. A narrow shape lets one table hold temperature, level, pressure or any new parameter without schema changes. Index by device and time; for long retention, roll raw readings into hourly and daily minimum, maximum and average.

```
reading(device_id, parameter, value DECIMAL(10,2), recorded_at)
INDEX (device_id, parameter, recorded_at)
```

Listing 2. Example exchange or schema.

7.2 *From readings to excursions*

A reading is in band when `minimum ≤ value ≤ maximum`. The alerting logic tracks excursions, not readings, so a sensor that stays out of band for an hour produces one alert, not hundreds.

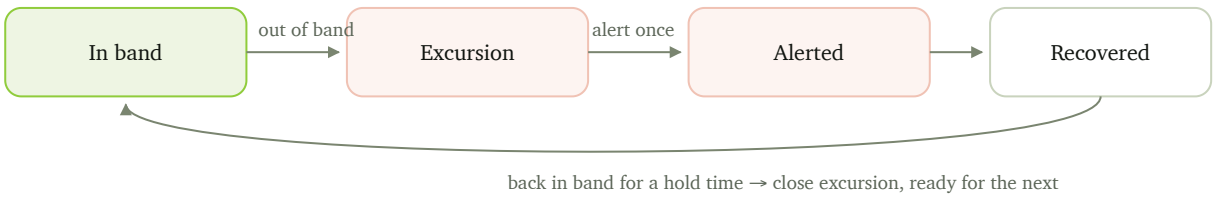


Figure 5. One alert per excursion. A short hold time before recovery avoids flapping around the limit.

7.3 *Alert content*

An alert names the device and its physical location, the value, the band, how long it has been out of band, and who else has been told. It goes by email by default; SMS and chat are added per site. A recovery message closes the loop.

7.4 *Evidence*

Scheduled jobs render activity reports as PDF by device, site and period, with minimum, maximum and every excursion, and email them to administrators. For cold chain and pharma, keep the raw readings and excursion log [8] for the regulator's retention period.

Common pitfalls. Comparing against the band with the limits reversed (so every reading alerts); storing readings with a uniqueness rule that rejects a repeated identical value; and alerting on every reading. Each is easy to test for and worth a unit test.

8 Water-level and pump control

Overhead tanks and sumps are the most common control problem we see: keep the tank full, never run the pump dry, never overflow, and know when the pump has stopped responding.

8.1 *Sensing options*

Sensor	Strength	Watch out for
Ultrasonic	Non-contact; reports level as a continuous percentage	Foam, condensation and tank geometry
Pressure transducer	Accurate in deep tanks and sumps	Calibration and sediment
Float switches	Cheap, robust discrete marks	Only tells you “above / below”

8.2 *Control logic*

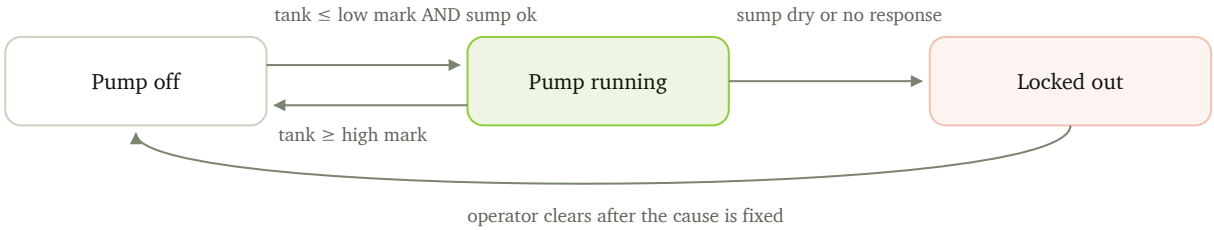


Figure 6. Automatic mode with hysteresis between the low and high marks, and a lock-out for dry-run and unresponsive pumps.

```
# evaluated on every level reading (automatic mode)
if pump.locked: return
if sump.level < sump.dry_mark:            stop(pump); lock(pump, "sump dry");   alert()
elif tank.level <= tank.low_mark  and pump.off:  command(pump, "START")
elif tank.level >= tank.high_mark and pump.on:   command(pump, "STOP")
if tank.level >= tank.overflow_mark:      alert("overflow risk")
# every START/STOP goes through the confirmed command queue (chapter 6);
# no confirmation in 30 s → alert "pump not responding" and lock automatic mode
```

Listing 3. Pseudocode.

8.3 *Worth designing in*

- Hysteresis: a gap between low and high marks so the pump does not short-cycle.
- Minimum run and rest times to protect the motor.
- Manual override with a reason, recorded in the audit trail.
- Daily summaries: pump run hours, starts, and litres moved if flow is metered.

9 NFC authentication

An NFC card or tag carries an identity; an approved phone is the reader. The same mechanism authenticates people at gates and on transport, confirms who collected a child, and proves that a certificate or product is genuine.

9.1 *Where it is used*

Use	Who taps	What is checked	Result
Campus and lab attendance	Staff, students, researchers	card valid, person active, reader approved	check-in / check-out with time and place
Transport boarding	Students	card valid, vehicle active, crew assigned	boarding record, guardian notified
Guardian handover	Registered guardian	guardian linked to the child	handover recorded
Certificate verification	Verifier	certificate exists, issuer approved	issuer, holder and details shown
Product and asset authenticity	Inspector or customer	tag linked to a genuine record	genuine / not found

9.2 *The validation chain*

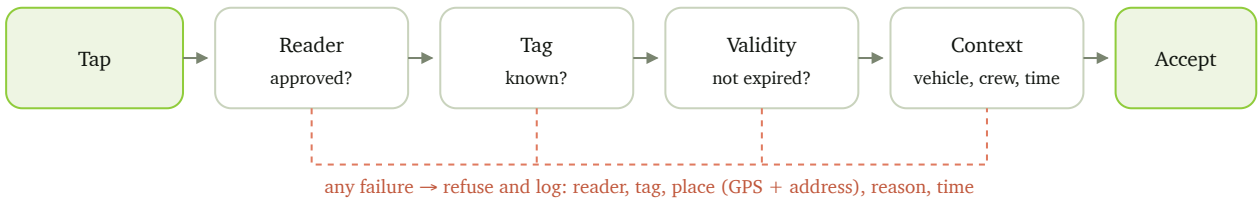


Figure 7. Every tap passes four checks before it counts. Refusals are never silent.

9.3 *Rules that make it reliable*

- **Sessions.** Taps before noon belong to the morning session, after noon to the evening session.
- **Debounce.** A second tap within three minutes is ignored; a later one is a check-out.
- **Photo policy.** Capture a photo at always, every 2nd, 4th or 8th tap, or never, to balance assurance and speed.
- **Offline first.** Readers keep a local database and sync when signal returns (chapter 10).

9.4 *Choosing tags, and resisting cloning*

Identifying a card by its serial number (UID) alone is simple and fast but can be copied by determined attackers. For higher assurance, use tags that generate a fresh cryptographic code on each tap (for example NTAG 424 DNA with secure unique messaging [5]), or store a signed payload on the tag [6] and verify the signature on the reader. Combine either with reader approval and context checks.

Assurance	Approach	Good for
Basic	Tag UID mapped to a record	Attendance with supervision, asset tags
Medium	Signed payload on the tag	Certificates, licences
High	Dynamic per-tap codes from secure tags	Product authenticity, high-value assets

10 Offline-first capture and synchronisation

Phones used as readers and data-capture tools work where signal is worst: buses, basements, plants and rural sites. The pattern below keeps them working without signal and makes synchronisation safe.

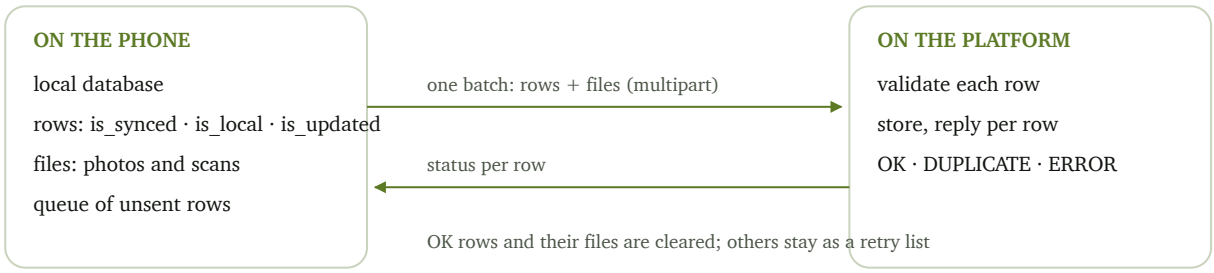


Figure 8. Per-row status makes partial success safe: nothing is lost and nothing is sent twice.

10.1 *Design rules*

- Every record created offline carries a client-generated identifier, so a retry can be recognised as the same record.
- Where several sites create records independently and later merge, give each site its own identifier range so keys never collide.
- Sync in a fixed order (create before update; parents before children) and keep a watermark per table and direction.
- Write conflicts to a conflict log for review rather than overwriting silently.
- Show the user what failed as a checklist they can retry.

11 Fleet and compliance at the source

Compliance data is most accurate when captured where the event happens. For fleets, that is the fuel pump and the vehicle.

Captured	Where	How it is identified
Fuel quantity and station	At the pump	Vehicle NFC tag
Odometer	At fill and at trip start/end	Vehicle tag + photo
Trip sheet and crew	Start of trip	Driver and supervisor tags
Job cards	Workshop	Vehicle tag

From these, kilometres between fills, consumption trends and anomalies follow directly. Documents (driver licence, insurance, pollution certificate, permits) carry expiry dates; a nightly job deactivates drivers and vehicles whose documents have expired and notifies the fleet manager, so a non-compliant trip cannot start.

12 Process calculations at the point of work

Some of the most valuable IoT is a calculation placed where the decision is made [10]. In an iron foundry, the metallurgist enters the spectrometer reading at the furnace; the tablet returns corrections in kilograms.

Quantity	Formula
Carbon equivalent	`CE = C + Si/3 + P/3 − 0.027·Mn + 0.4·S`
Liquidus temperature	`TL = (14.45 − CE) / 0.0089` °C
Band per element	target ± tolerance (C 0.10, Si 0.05, Mn 0.10, P 0.01, S 0.01, CE 0.03)
Addition	`kg = shortfall% × (metal + steel kg) / 100 / recovery%`

12.1 *Worked example*

Input	Value	Output	Value
Charge	1,000 kg	CE	4.05
Recovery, carbon raiser	30%	Liquidus	≈ 1,169 °C
Recovery, ferro-silicon	65%	Out of band	Si (low)

The app works offline, is licensed to one device, and keeps each iteration of the melt so the trend across a shift is visible.

13 Security model [2, 3, 4]

Threat	Example	Mitigation
Spoofed device	A laptop sends fake readings as device 7	Per-device credentials at provisioning; TLS; reject unregistered identities
Replay	An old “valve open” is resent	Response ids are single-use; commands expire; absolute actions
Unauthorised command	A viewer switches off a pump	Role-based control rights; audit of every command
Service disruption	A client tells the connection service to stop	Administrative actions only from authenticated operators, never from device sockets
Cloned tag	A copied student card	Approved readers; validity and context checks; dynamic-code tags for high assurance
Lost phone	A reader phone is stolen	Device approval revoked centrally; local data scoped to the signed-in user
Tampering with records	Editing a reading after the fact	Append-only readings; audit trail on every change

13.1 *Roles*

Role	Can
Administrator	Register devices, set bands and alert lists, manage users, control equipment
Power user	Control equipment, acknowledge alerts, run reports
Viewer	See dashboards and reports

14 Operations, sizing and reliability

14.1 *What we monitor*

- Connection service: up, connections open, messages per minute, error rate.
- Devices: count online versus registered; quiet devices by site.
- Commands: share confirmed, median time to confirmation, timeouts.
- Alerts: excursions open, time to acknowledge.
- Sync: pending rows per reader, failed rows.

14.2 *Sizing, back of the envelope*

Readings per day = devices × (86,400 ÷ reporting interval in seconds). 200 sensors reporting every 30 seconds produce 576,000 readings a day; at roughly 50 bytes a row with index, that is about 30 MB a day before roll-up. A single event-loop connection service handles hundreds to low thousands of mostly idle connections comfortably; beyond that, shard devices across several services.

Devices	Interval	Readings / day	Approx. storage / day
50	60 s	72,000	4 MB
200	30 s	576,000	30 MB
1,000	10 s	8,640,000	430 MB

14.3 *Reliability practices*

- Run the connection service under a supervisor that restarts it, and alert if it is down.
- Nightly backups of registry, readings and logs, with a regular restore drill.
- Roll raw readings into hourly and daily aggregates; keep raw data for the retention period required.
- Patch servers on a schedule; critical fixes within 48 hours.
- Keep a staging copy to test firmware and platform changes together.

15 Case study: student transport safety

Anonymised. A school transport operation running morning and evening routes.

15.1 *The challenge*

Parents wanted to know when their child boarded. The operator needed proof of who travelled on which vehicle, and a guarantee that no run started with an unassigned driver or an inactive vehicle. Coverage along routes was unreliable, and supervisors had minutes at each stop.

15.2 *What was built*

- An NFC card for every student, and the supervisor's phone as the reader, approved by device.
- Checks before a tap counts: card valid and not expired, vehicle active, driver and supervisor assigned.
- Morning and evening sessions tracked separately; the first tap checks a child in and notifies the guardian; a later tap, at least three minutes on, checks them out.
- Guardian handover taps recorded per session.
- Every refused tap logged with the reader, location, address and reason.
- A configurable photo policy at boarding.
- Offline recording on the phone, synchronised when signal returns.

- Automatic deactivation of drivers whose licence or engagement has expired.

15.3 *A run, step by step*

Moment	What happens
Before departure	Vehicle, driver and supervisor validated; the reader is approved.
At each stop	Students tap; valid taps check in and notify guardians; refused taps are logged with the reason.
On arrival	The headcount on board is known; anyone expected but missing stands out.
Evening	Drop-offs recorded as check-outs; handovers to guardians logged.

15.4 *What it changed*

Guardians are informed at the moment of boarding instead of calling the school; the operator has a per-run record of who travelled with whom; and compliance gaps, such as an expired licence, stop a run before it starts rather than being found afterwards.

16 Implementation roadmap

Phase	Outcome	Key activities
1. Discovery and site survey	Agreed device list, bands, roles and alert paths	Walk the site; check power and coverage; list parameters and limits; map who acts on what
2. Pilot	One site live with real devices	Provision 5–20 devices; panel, alerts and reports; test commands and failure cases
3. Scale	All sites onboarded	Provisioning runbook; per-site dashboards; training; role set-up
4. Operate and improve	Stable, measured service	Monitoring; monthly reviews of alerts, quiet devices and command success; firmware updates

17 Checklist: 20 questions for your IoT programme

1	Is every device registered with a location, band and owner before it can report?
2	Does every reading get a confirmation, and does the device act on errors?
3	Do you get one alert per excursion, with a recovery message?
4	Are band comparisons unit-tested (including reversed limits)?
5	Is every command complete only when the device confirms it?
6	Can an operator tell “never delivered” from “device didn't answer”?
7	Are quiet devices visible and locked out of control?
8	Are commands absolute, single-use and set to expire?
9	Do pumps have hysteresis, minimum run times and dry-run lock-out?
10	Do device connections use TLS and per-device credentials on public networks?
11	Can only authenticated operators perform administrative actions?
12	Is every identity tap validated for reader, tag, validity and context?
13	Are refused taps logged with place and reason?
14	Does your tag choice match the assurance you need?
15	Do readers and apps work offline, with per-row sync status?
16	Are documents and certifications tracked with expiry and acted on automatically?
17	Can you produce a clean report for any device and period?
18	Do you roll up raw readings and keep them for the required retention?
19	Are backups restored in a drill, regularly?
20	Who runs it after launch: patches, monitoring, support?

18 Glossary

Band	The safe minimum and maximum for a measured parameter.
Excursion	A period during which readings are outside the band.
Online window	How recently a device must have reported to count as online.
Confirmation	A reply that proves a message or command was received and acted on.
Hysteresis	A gap between switching thresholds that prevents rapid on-off cycling.
NFC	Near-field communication: short-range radio used by cards, tags and phones.
UID	A tag's serial number.
Carbon equivalent (CE)	A single figure summarising how an iron melt's chemistry affects solidification.
Liquidus	The temperature at which a melt begins to solidify.
Recovery rate	The share of an addition that ends up in the melt.

19 References

1. OASIS. *MQTT Version 5.0*. OASIS Standard, 2019.
2. M. Fagan et al. *Foundational Cybersecurity Activities for IoT Device Manufacturers*. NIST Interagency Report 8259, 2020.
3. OWASP Foundation. *OWASP Internet of Things Top 10*, 2018.
4. ETSI. *EN 303 645: Cyber Security for Consumer Internet of Things: Baseline Requirements*, v2.1.1, 2020.
5. NXP Semiconductors. *NTAG 424 DNA: Secure Unique NFC message (SUN) tag*, product data sheet.
6. NFC Forum. *NFC Data Exchange Format (NDEF) Technical Specification*.
7. E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. IETF RFC 8446, 2018.
8. World Health Organization. *Model guidance for the storage and transport of time- and temperature-sensitive pharmaceutical products*. WHO Technical Report Series 961, Annex 9, 2011.
9. LoRa Alliance. *LoRaWAN L2 1.0.4 Specification*, 2020; 3GPP TS 36 series, NB-IoT (Release 13 onward).
10. ASM International. *ASM Handbook, Volume 15: Casting*, 2008 (background on carbon equivalent and liquidus in cast irons).

About this report. Prepared by the Cloudnix engineering team from systems we have designed, built and operated. Client and product names are withheld; illustrative values are marked as examples. Cloudnix Software Labs, Bengaluru, India · cloudnix.com. This report may be shared freely in its entirety.

Document control

Report	CNX-TR-2026-01	Version	1.1
Issued	November 2026	Status	Published
File	cloudnix-iot-field-systems-v1.1.pdf		