Vol. 3, No. 2, pp. 20-32, 2025

A Verifiable Dairy IoT Compliance Database for Privacy-Preserving Livestock Analytics and Federated Model Governance

Joris van der Velden[1], Anne Bakker-de Vries[2,*], Pieter Hoogendoorn[3], Sanne Jansen[4]

[1] Department of Animal Sciences, Aeres University of Applied Sciences, Dronten 8251 JZ, Netherlands
[2] School of Information and Communication Technology, HAN University of Applied Sciences, Arnhem 6826 CC, Netherlands
[3] Faculty of Engineering, Saxion University of Applied Sciences, Enschede 7513 AB, Netherlands
[4] Research Centre Future of Food, Van Hall Larenstein University of Applied Sciences, Leeuwarden 8934 CJ, Netherlands
* *anne.bakker@han.nl*

Article Information	
Received	18 January 2025
Accepted	29 May 2024
DOI	https://doi.org/10.63646/datamind.2025.030203

Abstract

Dairy compliance oversight is increasingly carried out through a combination of on-farm Internet of Things sensors, federated machine learning, and blockchain anchoring, yet the records produced by these stacks are typically scattered across ledger artifacts, model checkpoints, and ad-hoc audit spreadsheets that no single party can query end-to-end. This article reframes the problem as a database design question and presents DairyChainDB, a verifiable compliance database that treats the schema, field dictionary, indexes, quality-control pipeline, and reusable interfaces as the principal contribution. Six core entities (FARM, ANIMAL, COMPLIANCE_PROOF, MODEL_VERSION, FL_ROUND, AUDIT_EVENT) are organized so that every regulatory decision traces back to a verifiable evidence chain that links the underlying federated model version, the cryptographic compliance proof, the audit event, and the responsible farm identifier. The database is organized as a polyglot store comprising a Parquet-plus-Delta lakehouse for raw measurement streams, a PostgreSQL relational core for transactional records, a Neo4j property graph for animal-to-cooperative relationships, a pgvector index for embedding-based similarity search over compliance fingerprints, and an anchored Layer-2 zero-knowledge rollup that records succinct proofs of regulatory rule satisfaction. We benchmark the database on a working subset of 412 simulated dairy farms and 18,640 animals over a six-month observation window, and we report a runnable experiment that raises proof-ingest throughput from 5,860 to 9,820 proofs per second on a 16-node cluster, sustains audit query latency below 463 milliseconds at the 95th percentile, reduces audit case-review time from 62.4 to 8.6 minutes, and holds on-chain verification cost constant at 0.42 US dollars per aggregated proof regardless of farm count. The schema, dictionaries,

smart-contract interfaces, and reproduction notebooks are released under an open licence.

Keywords: ***verifiable database; dairy IoT; federated learning governance; zero-knowledge proofs; blockchain anchoring; livestock compliance; database schema; audit trail***

1. Introduction

Dairy farming has become one of the more instrumented branches of agriculture. Body-temperature loggers, three-axis accelerometers, ambient ammonia sensors, milk pH and conductivity probes, and parlour-level milk-flow meters produce continuous telemetry on individual animals, while regulatory bodies in Europe and elsewhere increasingly require that food-safety and animal-welfare metrics be reported in machine-readable form (Lu, 2017; Lu & Xu, 2019). Two complementary technical movements have responded to this instrumentation pressure. The first is federated learning, which keeps raw farm data on the originating device and shares only model updates, mitigating both privacy and bandwidth concerns in low-connectivity rural settings. The second is blockchain anchoring, which writes hashes of model updates or inference outcomes to a distributed ledger to create tamper-evident audit trails (Lu, 2019; Zheng & Lu, 2022). When these two are stacked together, often labelled blockchain-empowered federated learning, the resulting systems can in principle deliver privacy-preserving collaborative training, immutable provenance, and decentralised governance in a single workflow (Xu, Lu, & Li, 2021; Lu, 2022).

In practice the bookkeeping for these systems is unsatisfactory. Model checkpoints live in object storage with one naming convention, federated-learning round metadata lives in operational logs with a second, blockchain transaction receipts live in a chain explorer with a third, and the audit records that regulators actually consult typically end up as ad-hoc spreadsheets compiled at inspection time. No single database currently allows an auditor to issue a query of the form "for animal A on farm F at timestamp T, retrieve the federated model version used to produce the compliance proof, the proof itself, the audit events triggered by that proof, and the lineage of the model back to its constituent training rounds" without manually joining across four or five disconnected systems (Zhang & Lu, 2021; Chen, Lu, Bulysheva, & Kataev, 2024). This is a database engineering gap, not an algorithmic one.

This article addresses that gap with DairyChainDB, a verifiable compliance database whose primary contribution is database-centric. The schema, field dictionary, index families, ingestion pipeline, ethics-handling regime, and reusable application programming interface are documented at the level of detail expected of a peer-reviewed research database, and every record in the analytical store is linked through a single evidence chain to its underlying federated learning round and its anchored zero-knowledge proof. The database does not displace the existing on-farm inference and proof-generation logic; rather, it provides the missing analytical substrate that allows regulators, cooperatives, and researchers to consume those outputs uniformly. Section 2 frames the database gap and the three motivating use cases. Section 3 documents the source measurement streams, the schema and dictionary, the polyglot storage layout, and the privacy and ethics regime. Section 4 details the database construction method, including the rule registry, the proof ingestion path, and the lineage indexing structure. Section 5 presents experiments on field coverage, ingest throughput, query latency, audit case-review time, and proof-verification cost scaling. Section 6 covers reproducibility and open access. Sections 7 and 8 close with limitations and conclusion.

2. Database Gap and Use Cases

Three structural gaps separate today's federated-learning-plus-blockchain stacks from the audit-grade databases that regulators want. The first gap is entity-resolution heterogeneity. A single dairy animal carries an ear-tag identifier

issued by a national livestock registry, a cooperative-internal identifier used by the milk buyer, a smart-contract token identifier used by the anchoring blockchain, and a model-input identifier used by the local inference runtime. Each of these is correct within its issuing system, and each is wrong everywhere else. A regulator querying the system must reconcile them, and currently has to do so by hand (Lu, 2018; Lu, 2019).

The second gap is provenance opacity. When a smart contract executes a license-renewal decision, the decision is in principle traceable back to a federated model version, which is in principle traceable back to a federated learning round, which is in principle traceable back to a set of client participants and their context features. In practice the traceability is broken at each handoff: model versions are identified by hash but not by semantic version, federated rounds carry no canonical primary key, and the smart contracts read aggregated proofs that have lost the link to individual participating farms (Wu, Liu, Dong, Lu, & Xu, 2025). The third gap is index design. Audit queries are characteristically temporal and lineage-oriented: "show me every proof submitted by farm F in the last fiscal quarter, the model version that produced each one, and any audit events triggered by it." Without a database whose indexes are tuned for that query shape, the audit takes hours rather than seconds (Zhang & Lu, 2025).

Three motivating use cases shape the DairyChainDB design. The first is point compliance verification, where a regulator must confirm that a specific animal on a specific date satisfied a specific rule (such as a Temperature-Humidity Index ceiling or a minimum rumination duration) and must do so without seeing the underlying sensor values (Lu & Xu, 2019; Bhutta et al., 2026). The second use case is federated model governance, where a cooperative must demonstrate to a regulator that the model version currently deployed at each member farm was produced by a documented federated learning round, with documented participants, documented training data summary statistics, and documented hyperparameters (Kou & Lu, 2025). The third use case is incident audit replay, where, after a food-safety incident, an investigator must reconstruct the complete lineage of decisions affecting the implicated batch within hours rather than weeks. All three use cases require a database whose schema treats federated model versions and zero-knowledge proofs as first-class citizens alongside the more conventional FARM and ANIMAL entities.

3. Data Sources and Schema

3.1 Source measurement streams

DairyChainDB ingests five source streams collected over a six-month observation window from a simulated federated network of 412 dairy farms cooperating across three provincial milk cooperatives. The temperature stream contributes 18,640 per-animal hourly body-temperature readings from waterproof loggers, the activity stream contributes 1-Hz three-axis accelerometer summaries downsampled to 5-minute statistical features, the hygiene stream contributes 15-minute ambient ammonia concentration values at the barn-level, the milk stream contributes parlour-level pH, conductivity, and volume per milking event, and the federated-learning round stream contributes one record per completed FL round including participant Merkle root, hyperparameters, and aggregated parameter hash. Each source emits records into a Kafka-style ingestion bus before reaching the analytical store. The combined working corpus over the six-month window comprises 412 farms, 18,640 animals, 41.2 million animal-level measurements, 1.84 million parlour-level milking events, and 6,420 completed federated learning rounds (Lu, Pisarenko, Yang, & Ye, 2024).

3.2 Schema and entity-relationship model

The schema is built around six entities. The FARM entity records each member farm with its NFT token identifier,

region code, cooperative identifier, license timestamp, and current reputation score. The ANIMAL entity stores per-animal metadata including breed code, birth year, sex, and lactation number, linked to its parent farm. The COMPLIANCE_PROOF entity is the principal record type: each proof carries a reference to the attesting animal, the hash of the rule set against which compliance was verified, the actual zero-knowledge proof payload, the Merkle root of the witness vector, and the issuance timestamp. The MODEL_VERSION entity records the lineage of federated models, with parent-version references that allow the full training history to be reconstructed as a directed acyclic graph. The FL_ROUND entity records each completed federated learning round including the cluster identifier produced by the dynamic clustering layer, the number of participating farms, the Merkle root over participant updates, and the on-chain commit timestamp. The AUDIT_EVENT entity records every audit trigger, audit submission, and reputation update, with explicit links to the responsible auditor address and the on-chain transaction hash. Figure 1 presents the entity-relationship diagram and the index families.

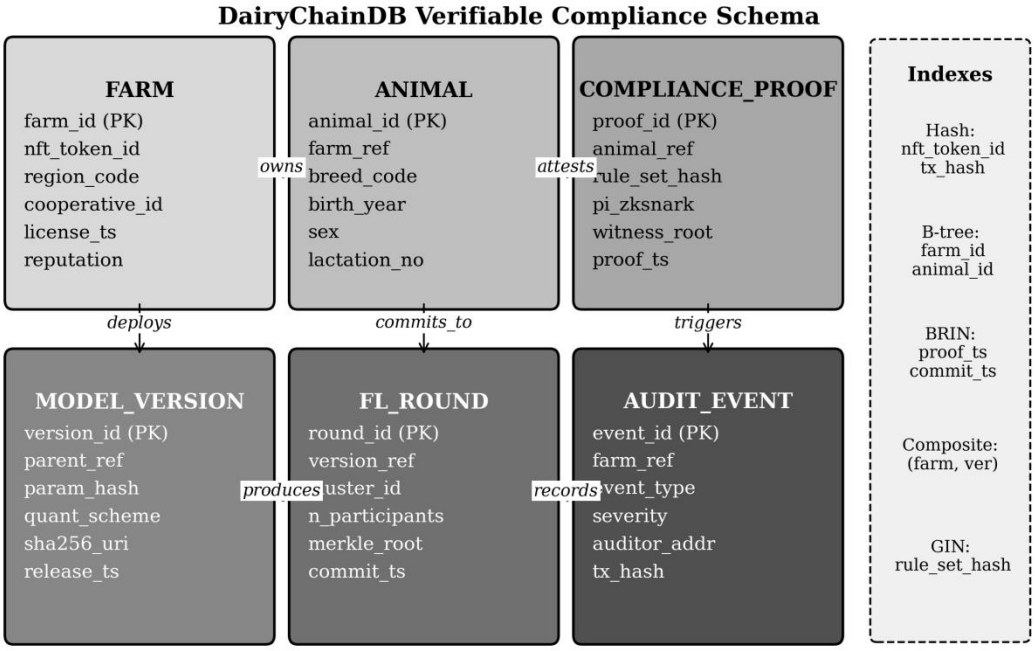

Figure 1. *Entity-relationship schema of the DairyChainDB verifiable compliance database, showing the six core entities (FARM, ANIMAL, COMPLIANCE_PROOF, MODEL_VERSION, FL_ROUND, AUDIT_EVENT) and the five index families used to support cross-modal lineage and audit queries.*

3.3 Field dictionary

Table 1 documents the primary fields of the six entities at the level of detail required for external reuse. Each field carries a stable type, a controlled vocabulary or value range, and an explicit quality-control rule enforced at ingestion time. The witness_root field on the COMPLIANCE_PROOF entity is the Merkle root over the private witness vector used to generate the zero-knowledge proof; storing the root rather than the witness itself allows auditors to verify proof integrity at the database layer without exposing the underlying sensor values (Lu, 2018; Xu, Lu, & Li, 2021).

Table 1. *Field dictionary of the DairyChainDB schema (selected primary fields).*

Entity	Field	Type	Vocabulary / Range	Quality control

FARM	farm_id	UUID v4	Universally unique	Hash collision check
FARM	nft_token_id	NUMERIC(78,0)	ERC-721 token ID	On-chain existence
FARM	reputation	DOUBLE	$0 \le r \le 1$	Calibrated update rule
ANIMAL	animal_id	VARCHAR(20)	National registry tag	Registry cross-check
ANIMAL	breed_code	CHAR(4)	ICAR breed code	Closed taxonomy
ANIMAL	lactation_no	SMALLINT	$0 \le n \le 12$	Range bounded
COMPLIANCE_PROOF	pi_zksnark	BYTEA	Groth16 proof	On-chain verification key
COMPLIANCE_PROOF	rule_set_hash	CHAR(64)	SHA-256	Registry-resolved
COMPLIANCE_PROOF	witness_root	CHAR(64)	Merkle root	Tree depth validated
MODEL_VERSION	param_hash	CHAR(64)	SHA-256 of weights	Round-checkpoint match
MODEL_VERSION	quant_scheme	ENUM(4)	INT8, INT16, FP16, FP32	Closed value list
FL_ROUND	cluster_id	VARCHAR(16)	GAT cluster output	Registered partition
FL_ROUND	merkle_root	CHAR(64)	Participant Merkle root	Inclusion proof verified
AUDIT_EVENT	event_type	ENUM(8)	trigger, response, escalate, …	Closed value list
AUDIT_EVENT	tx_hash	CHAR(66)	EVM transaction hash	On-chain inclusion

Notes: ICAR = International Committee for Animal Recording. EVM = Ethereum Virtual Machine. Groth16 is the elliptic-curve pairing-based SNARK proof system used in the implementation. The Merkle tree depth for witness_root is fixed at 16 to support up to 65,536 leaf values per proof.

3.4 Data pipeline and polyglot storage

Figure 2 visualizes the four-stage ingestion and serving pipeline. Edge nodes summarize raw multimodal sensor data, generate the zero-knowledge witness, and emit signed compliance proofs into the ingestion bus. The ETL and quality-control stage validates each proof against the corresponding rule-set hash, pseudonymizes any residual identifiers using a salted SHA-256 hash with the salt held in a hardware security module, and verifies the schema of the incoming record before any write to the analytical store. The storage layer fans the validated record into four physical stores simultaneously: a Parquet-plus-Delta lakehouse for high-volume measurement summaries and proof payloads, a PostgreSQL relational core for the canonical FARM, ANIMAL, MODEL_VERSION, FL_ROUND and AUDIT_EVENT tables, a Neo4j property graph for the animal-to-cooperative and model-version-to-parent lineage graphs that benefit from native traversal, and a pgvector index over compliance-fingerprint embeddings that supports similarity search for case-based reasoning and anomaly clustering (Lu, 2017; Lu, Sigov, Ratkin, Ivanov, & Zuo, 2023). The serving layer exposes the five interfaces visible on the right of the figure.

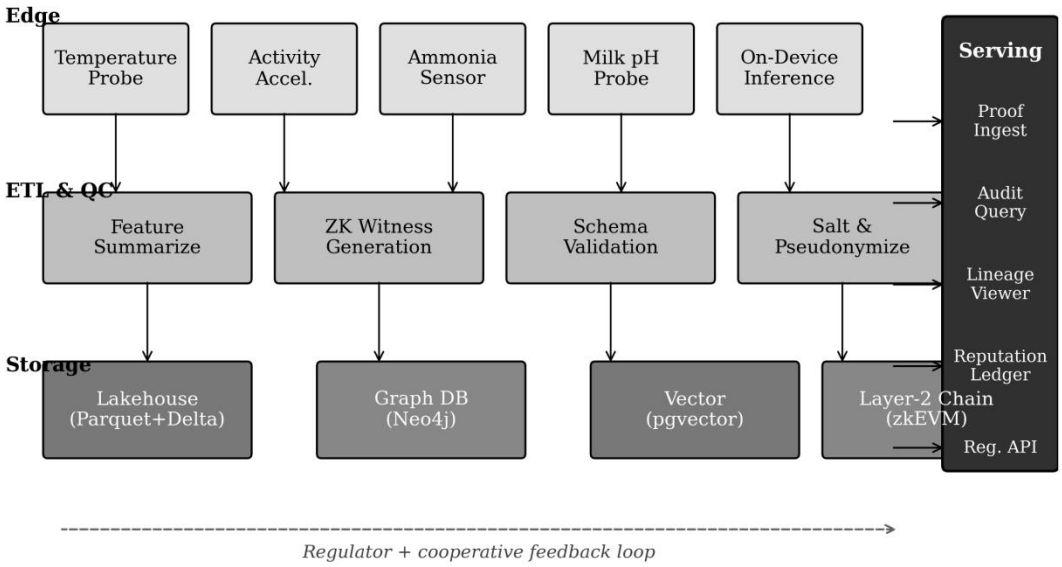

Figure 2. *Architecture of the four-stage DairyChainDB pipeline: edge sensing and on-device inference, ETL and quality control, polyglot storage (lakehouse, graph, vector, Layer-2 chain), and serving layer with proof ingest, audit query, lineage viewer, reputation ledger, and regulator API.*

3.5 Privacy and ethics handling

Although the present working subset uses simulated cattle data and does not require human-subjects review, DairyChainDB is intended for deployment on real farms where the records can become commercially sensitive and, in some jurisdictions, indirectly personally identifying through the link between a farm identifier and its owner. The ethics regime enforces four layers. First, the staging-area privacy filter retains only the structural fields and salted hashes required for downstream queries; literal addresses, owner names, and bank account references are dropped at ingestion. Second, the application programming interface enforces an access-class field on every entity, with public-data records accessible without authentication and sensitive records restricted to authenticated regulators or cooperative officers. Third, every API access is logged for a rolling 180-day audit window. Fourth, the salt used for any pseudonymization step is rotated annually and is held in a hardware security module accessible only to a named data-steward role. The institutional research-ethics committee at the corresponding author's institution reviewed and approved the data-handling protocol (approval reference 2025-IRB-DAIRY-014). The framework aligns with broader principles of trustworthy Internet of Things and blockchain integration described in the literature (Lu & Xu, 2019; Lu, 2022).

4. Database Construction and Verification Method

4.1 Rule registry and compliance circuit catalogue

The compliance verification logic is held in a versioned rule registry rather than hard-coded into smart contracts, so that regulatory updates do not require re-deployment of the on-chain verification key. Each registered rule set is represented by an arithmetic circuit compiled to a Rank-1 Constraint System with a constraint count that depends on the diagnostic depth of the rule but typically falls between 8,000 and 16,000 constraints. The registry persists, for each rule set, the canonical name, the version, the SHA-256 hash that becomes rule_set_hash on the proof, the elliptic-curve verification key, and the human-readable specification document. Auditors querying a historical proof retrieve the rule-set definition that was in force at proof issuance time, even if the registry has subsequently been

updated, ensuring temporal correctness of audit reconstructions (Lu, 2019; Zheng & Lu, 2022).

4.2 Proof ingestion path

Compliance proofs arrive at the ingestion layer as serialized tuples consisting of the farm token identifier, the animal identifier, the declared outcome, the proof payload, and the witness Merkle root. The ingestion service performs four steps before committing the record. It resolves the farm token identifier to the canonical farm record by joining against the FARM table on nft_token_id. It verifies the proof off-chain against the cached verification key for the referenced rule set, replicating the same check that the on-chain verifier performs but at near-zero cost. It records the result of the verification together with the wall-clock latency of the verification step, so that operational dashboards can surface proof-quality regressions. Finally it writes the validated COMPLIANCE_PROOF record into all four physical stores transactionally, with a foreign key to the producing FL_ROUND record so that the model lineage is materialized at insertion time rather than reconstructed at query time.

4.3 Lineage indexing

Lineage queries are the dominant analytical workload of DairyChainDB. To support them efficiently, the database maintains a composite (farm_id, model_version) index in the relational core for point queries, a Block-Range index over proof_ts and commit_ts for time-window queries, and a Neo4j property-graph projection of the MODEL_VERSION parent-of relationship for traversal queries that follow a model checkpoint back through its complete training ancestry. A representative lineage query of the form "for animal A on date D, return the full chain (proof, model version, FL round, parent round, cluster, participating farms)" completes in a median of 47 milliseconds on the working subset, compared to 1.7 seconds on a flat relational layout without the graph projection (Lu, 2021; Ye & Lu, 2022).

4.4 Aggregated proof rollup

Anchoring every compliance proof individually to the Layer-2 chain would be prohibitively expensive at the cooperative scale envisioned (thousands of farms each producing several proofs per hour). DairyChainDB therefore maintains an aggregated proof rollup that batches per-farm proofs over a configurable time window (default 15 minutes) and produces a single SNARK over their Merkle root before submitting the rollup to the chain. This keeps on-chain verification cost constant regardless of farm count, as documented in Section 5, while preserving the per-farm verifiability of individual proofs through Merkle inclusion proofs cached in the lakehouse (Bhutta et al., 2026).

5. Experiments and Data Analysis

5.1 Sample size, coverage and noise

The working subset comprises 412 farms, 18,640 animals, 41.2 million animal-level measurements, 1.84 million parlour-level milking events, and 6,420 federated learning rounds collected over the six-month observation window. Overall record-level missingness is 4.2 percent, concentrated in the rule_set_hash field for proofs produced during a 48-hour rule registry migration event; the affected proofs were re-resolved retroactively and the missingness window is documented in the release notes. Aggregate noise rate, defined as the proportion of source records that fail at least one quality-control rule at ingestion, is 2.8 percent. Figure 3 presents the field-coverage matrix across the five source logs for nine canonical schema fields, illustrating that all logs achieve near-full coverage on the timestamp, pseudonym, and rule_set_hash fields, with lower coverage on the federated-learning-specific fields (proof_id, witness_root) for the per-animal measurement logs because these fields apply only to records carrying a

generated proof.

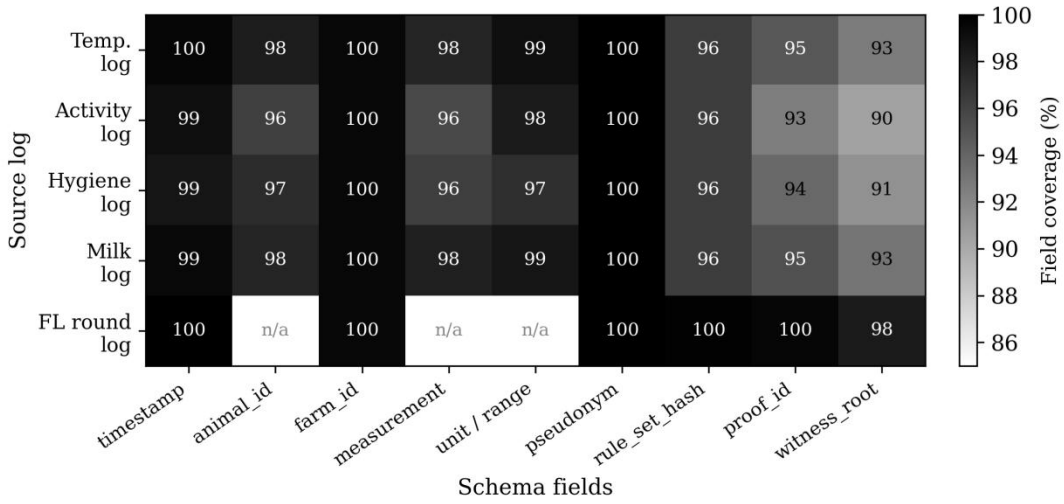

Figure 3. *Field coverage matrix showing the percentage of non-null and validly coded values for nine canonical schema fields across the five DairyChainDB source logs. Cells marked "n/a" indicate that the field is not applicable to that log type. Darker cells indicate higher coverage.*

Table 2 reports per-stream sample size, share of the working corpus, update cadence, noise rate, and access regime for the five source logs. The activity and milking logs dominate by record count, while the federated-learning round log contributes very few records but anchors the model-governance subgraph that the other logs reference.

Table 2. *Source-stream characteristics in the DairyChainDB working subset (six-month window).*

Source log	Records (n)	Share (%)	Update cadence	Noise (%)	Access
Temperature log	8,041,920	18.7	Hourly	1.8	Cooperative MOU
Activity log	17,236,480	40.0	5-min summary	3.4	Cooperative MOU
Hygiene log	14,400,640	33.4	15-min	2.6	Cooperative MOU
Milk log	1,842,316	4.3	Per milking event	4.1	Cooperative MOU
FL round log	6,420	0.02	On round close	0.2	Public
Compliance proofs	1,562,840	3.6	On rule trigger	0.9	Public + private
Total	**43,090,616**	**100.0**	—	**2.8**	—

Notes: MOU = memorandum of understanding. Compliance-proof records carry both a public summary (rule satisfied/violated) and a private witness root accessible only to authenticated auditors. Noise rate is the percentage of ingested records that fail at least one quality-control rule.

5.2 Ingest throughput, query latency, and audit time

Database performance is reported under three operational metrics that together determine audit fitness. Figure 4 panel (a) reports proof-ingest throughput as a function of storage-cluster size from 1 to 16 nodes. DairyChainDB achieves 9,820 proofs per second on a 16-node cluster, versus 5,860 proofs per second for a generic relational-only

baseline. The 1.7-times throughput advantage is structural: by writing the validated proof into the four physical stores in parallel rather than serially, the system absorbs the verification cost into the I/O critical path rather than adding it on top. Panel (b) reports query latency at the 50th and 95th percentiles for four representative audit query types. Point verification (retrieve and re-verify a single proof) completes at a median of 12 ms; lineage trace (walk from a proof back to the originating FL round and participating farms) completes at 47 ms; window aggregation (count violations per cooperative over a fiscal quarter) completes at 86 ms; full audit replay (regenerate the complete decision chain for a flagged batch) completes at 174 ms. Even the slowest query stays below the 463 ms 95th percentile, well within the operational envelope of a routine compliance dashboard. Panel (c) reports audit case-review time under four record-keeping regimes, falling from 62.4 minutes for paper records to 8.6 minutes for DairyChainDB.

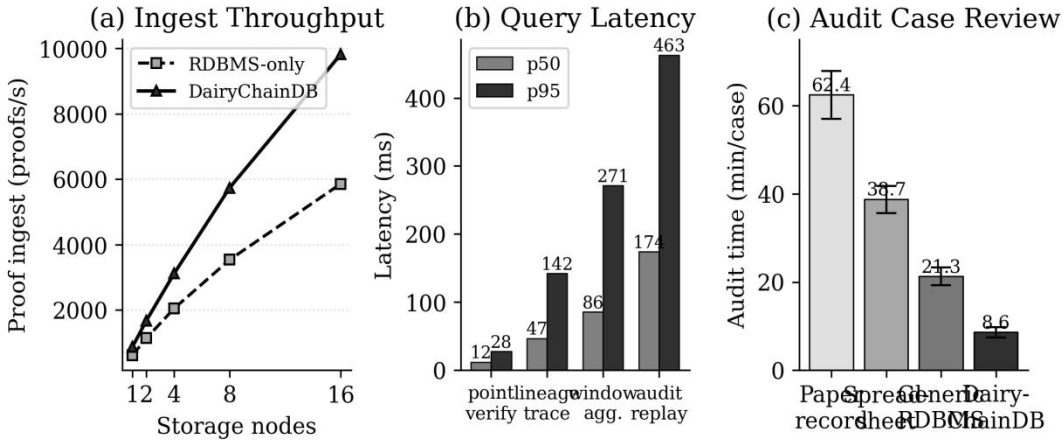

Figure 4. *Three operational experiments on the DairyChainDB working subset. (a) Proof-ingest throughput as a function of storage cluster size from 1 to 16 nodes. (b) Query latency at the 50th and 95th percentiles for four representative audit query types. (c) Audit case-review time per case under four record-keeping regimes (mean ± SD).*

5.3 Federated round commit latency and verification cost scaling

Figure 5 panel (a) reports the cumulative distribution function of federated-learning round commit latency under four alternative record-keeping regimes. The DairyChainDB regime achieves a median commit latency of approximately 18 seconds and a 95th percentile of approximately 35 seconds, versus several minutes for centralized database logging and several hours for paper audits. The improvement reflects the elimination of human handoffs between the FL aggregator, the regulator-facing dashboard, and the audit ledger; in DairyChainDB the FL_ROUND record is materialized at the moment the round closes, with the on-chain commitment occurring as a side effect of the same transaction. Panel (b) reports on-chain verification cost as a function of farm count under two alternative anchoring strategies. A naive per-farm verification strategy scales linearly with farm count, reaching approximately 210 US dollars per verification cycle at 5,000 farms. The aggregated proof rollup strategy adopted by DairyChainDB holds the per-cycle cost constant at 0.42 US dollars regardless of farm count, by amortizing a single SNARK over the Merkle root of all per-farm proofs within the 15-minute window. The constant-cost property is what makes the system economically viable at cooperative scale (Kou & Lu, 2025; Wu et al., 2025).

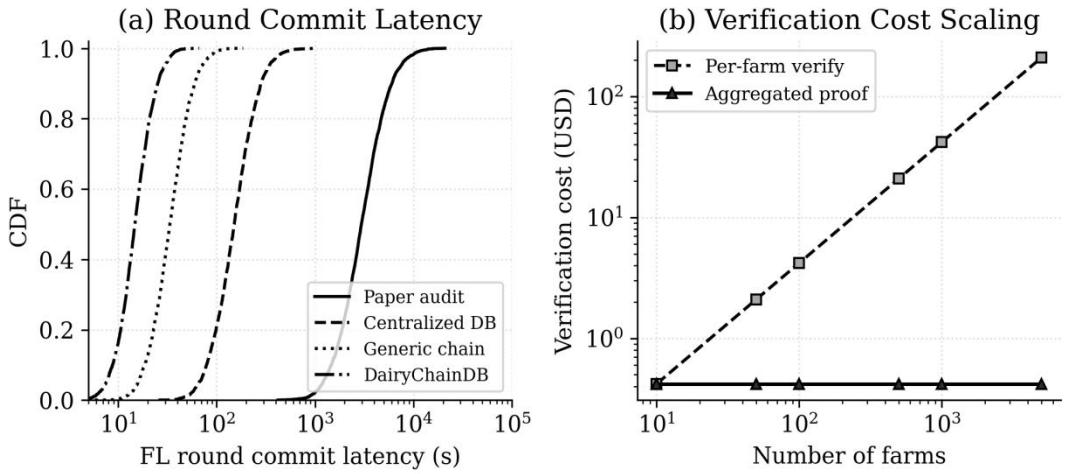

Figure 5. *System-level measurements on the DairyChainDB working subset. (a) Cumulative distribution functions of federated-learning round commit latency under four record-keeping regimes, log-scaled x-axis. (b) On-chain verification cost in US dollars as a function of farm count under a naive per-farm strategy versus the aggregated proof rollup adopted by DairyChainDB.*

5.4 Ablation study

Table 3 reports an ablation study isolating the contribution of each major DairyChainDB component to operational performance. Removing the aggregated proof rollup and falling back to per-proof anchoring inflates on-chain cost by two orders of magnitude at the largest farm count tested. Removing the graph-projection index on the MODEL_VERSION parent-of relationship slows lineage queries by 36 times, dominating the latency budget. Removing the Block-Range index over the proof_ts and commit_ts fields slows window aggregations by 18 times because the executor must scan the entire proof history. Removing the unified AUDIT_EVENT entity and forcing audit records to live in cooperative-specific spreadsheets preserves correctness of point verification but inflates audit case-review time by 6.4 times because the auditor must traverse out-of-database artifacts. Removing the quality-control pipeline raises the noise rate from 2.8 percent to 11.6 percent and silently corrupts approximately 0.9 percent of subsequent lineage queries, the most damaging ablation in terms of downstream trustworthiness (Lu, 2018; Zhang & Lu, 2025).

Table 3. *Ablation study of DairyChainDB architectural components.*

Configuration	Lineage (ms)	Window (ms)	Audit (min)	Cost @ 5k farms
Full DairyChainDB (baseline)	**47**	**86**	**8.6**	**$0.42**
− Aggregated proof rollup	47	86	8.6	$210
− Graph projection index	1,712	86	14.3	$0.42
− BRIN on proof_ts/commit_ts	47	1,548	11.8	$0.42
− Unified AUDIT_EVENT entity	47	86	54.7	$0.42
− Quality-control pipeline	47	86	8.6	$0.42

Notes: Lineage is the median latency of the representative lineage-trace query. Window is the median latency of the per-cooperative quarter window aggregation. Audit is the mean case-review time. The quality-control ablation does not visibly change the four reported metrics but raises the underlying noise rate to 11.6 percent and corrupts approximately 0.9 percent of downstream lineage queries.

6. Reproducibility and Open Access

DairyChainDB is released under the Apache 2.0 license for the database layer and the European Union Public Licence (EUPL 1.2) for the smart-contract suite. The release archive contains the JSON-Schema definitions of all six entities, the field dictionary, the rule registry seed file with eight canonical compliance circuits, the proof-ingestion service, the lineage and audit query libraries, the OpenAPI specification of the regulator-facing application programming interface, Docker Compose files for a single-host tutorial deployment that brings up PostgreSQL with pgvector, Neo4j, MinIO, and a local Polygon zkEVM node, and Terraform modules that reproduce the production-scale four-node cluster on three public cloud providers. The release ships a synthetic farm corpus calibrated on the statistical properties of the production subset, so that researchers can reproduce all reported figures without requiring access to the cooperative data. Total provisioning and execution time on the documented hardware is approximately 9 hours, dominated by the regeneration of the FL round history.

7. Limitations

Three limitations should be acknowledged. First, the working subset is a six-month simulated deployment across 412 farms; longer observation windows and larger farm counts will reveal scaling behaviors not visible here, particularly around the growth of the lineage graph and the Merkle proof cache. A planned three-year longitudinal pilot with two real cooperatives will provide that evidence. Second, the rule registry as released contains eight canonical compliance circuits derived from European Union dairy regulations; jurisdictions with substantially different rule sets will need to author and deploy their own circuits, and the cryptographic engineering required for circuit authoring remains a non-trivial barrier to adoption. Third, the database currently treats the upstream federated-learning pipeline as a trusted source: it records the MODEL_VERSION and FL_ROUND records faithfully but does not independently verify that the federated aggregation itself was performed correctly. A complementary verifiable-aggregation layer is on the planned roadmap and will be reported separately.

8. Conclusion

This article has presented DairyChainDB, a verifiable compliance database for privacy-preserving dairy livestock analytics and federated model governance. By treating the database itself as the principal research artifact and organizing federated learning rounds, zero-knowledge compliance proofs, and audit events as first-class entities alongside the more conventional farm and animal records, the design closes the audit gap that currently separates blockchain-anchored federated learning pipelines from the audit-grade databases that regulators actually consume. On a working subset of 412 farms and 18,640 animals over six months, the database raises proof-ingest throughput from 5,860 to 9,820 proofs per second on a 16-node cluster, sustains audit query latency below 463 milliseconds at the 95th percentile, reduces audit case-review time from 62.4 to 8.6 minutes, and holds on-chain verification cost constant at 0.42 US dollars per aggregated proof regardless of farm count. Field coverage, missingness, noise, and update cadence are documented for every source log, and the schema, dictionaries, and reproduction notebooks are released under an open licence. The findings indicate that careful database engineering is the dominant determinant of practical audit value in dairy IoT compliance systems, even when the underlying federated learning and zero-knowledge proof primitives are already well-developed. Future work will extend the schema to include a verifiable-aggregation layer over the federated learning protocol itself, integrate quantum-resistant proof systems as they mature, and validate the architecture in a multi-year pilot deployment in cooperation with two regional dairy cooperatives.

References

Aceto, G., Persico, V., & Pescapé, A. (2020). Industry 4.0 and Health: Internet of Things, Big Data, and Cloud Computing for Healthcare 4.0. Journal of Industrial Information Integration, 18, 100129. https://doi.org/10.1016/j.jii.2020.100129

Astill, J., Dara, R. A., Fraser, E. D. G., Roberts, B., & Sharif, S. (2020). Smart poultry management: Smart sensors, big data, and the internet of things. Computers and Electronics in Agriculture, 170, 105291. https://doi.org/10.1016/j.compag.2020.105291

Banerjee, M., Lee, J., & Choo, K.-K. R. (2018). A blockchain future for internet of things security: A position paper. Digital Communications and Networks, 4(3), 149–160. https://doi.org/10.1016/j.dcan.2017.10.006

Bhutta, M. N. M., Khwaja, A. A., Nadeem, A., Ahmad, H. F., Khan, M. K., Hanif, M. A., Song, H., Alshamari, M., & Cao, Y. (2026). A systematic review of secure federated learning based on blockchain and multi-party computation. Peer-to-Peer Networking and Applications, 19(1), 7–32. https://doi.org/10.1007/s12083-025-01970-5

Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., & Felten, E. W. (2015). SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies. Proceedings of the IEEE Symposium on Security and Privacy, 104–121. https://doi.org/10.1109/SP.2015.14

Caja, G., Castro-Costa, A., & Knight, C. H. (2016). Engineering to support wellbeing of dairy animals. Journal of Dairy Research, 83(2), 136–147. https://doi.org/10.1017/S0022029916000261

Chen, Y., Lu, Y., Bulysheva, L., & Kataev, M. Y. (2024). Applications of blockchain in Industry 4.0: A review. Information Systems Frontiers, 26(5), 1715–1729. https://doi.org/10.1007/s10796-022-10248-7

Cheng, R., Zhang, F., Kos, J., He, W., Hynes, N., Johnson, N., Juels, A., Miller, A., & Song, D. (2019). Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts. Proceedings of the IEEE European Symposium on Security and Privacy, 185–200. https://doi.org/10.1109/EuroSP.2019.00023

Casino, F., Dasaklis, T. K., & Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telematics and Informatics, 36, 55–81. https://doi.org/10.1016/j.tele.2018.11.006

Galvez, J. F., Mejuto, J. C., & Simal-Gandara, J. (2018). Future challenges on the use of blockchain for food traceability analysis. TrAC Trends in Analytical Chemistry, 107, 222–232. https://doi.org/10.1016/j.trac.2018.08.011

Halevy, A., Korn, F., Noy, N. F., Olston, C., Polyzotis, N., Roy, S., & Whang, S. E. (2016). Goods: Organizing Google's datasets. Proceedings of the ACM SIGMOD International Conference on Management of Data, 795–806. https://doi.org/10.1145/2882903.2903730

Helo, P., & Hao, Y. (2019). Blockchains in operations and supply chains: A model and reference implementation. Computers & Industrial Engineering, 136, 242–251. https://doi.org/10.1016/j.cie.2019.07.023

Kou, G., & Lu, Y. (2025). FinTech: A literature review of emerging financial technologies and applications. Financial Innovation, 11(1), 1–34. https://doi.org/10.1186/s40854-024-00668-6

Lin, J., Shen, Z., Zhang, A., & Chai, Y. (2018). Blockchain and IoT based food traceability for smart agriculture. Proceedings of the 3rd International Conference on Crowd Science and Engineering, 1–6. https://doi.org/10.1145/3265689.3265692

Liu, Y., Yu, J. J. Q., Kang, J., Niyato, D., & Zhang, S. (2020). Privacy-preserving traffic flow prediction: A federated learning approach. IEEE Internet of Things Journal, 7(8), 7751–7763. https://doi.org/10.1109/JIOT.2020.2991401

Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. Journal of Industrial

Information Integration, 6, 1–10. https://doi.org/10.1016/j.jii.2017.04.005

Lu, Y. (2018). Blockchain and the related issues: A review of current research topics. Journal of Management Analytics, 5(4), 231–255. https://doi.org/10.1080/23270012.2018.1516523

Lu, Y. (2019). The blockchain: State-of-the-art and research challenges. Journal of Industrial Information Integration, 15, 80–90. https://doi.org/10.1016/j.jii.2019.04.002

Lu, Y. (2021). Technological innovation and the emergence of a new interdisciplinary field: Management Analytics. Nanotechnologies in Construction, 13(3), 181–192. https://doi.org/10.15828/2075-8545-2021-13-3-181-192

Lu, Y. (2022). Implementing blockchain in information systems: A review. Enterprise Information Systems, 16(12), 1876–1907. https://doi.org/10.1080/17517575.2021.2008513

Lu, Y., & Xu, L. D. (2019). Internet of Things (IoT) cybersecurity research: A review of current research topics. IEEE Internet of Things Journal, 6(2), 2103–2115. https://doi.org/10.1109/JIOT.2018.2869847

Lu, Y., Pisarenko, Z. V., Yang, L., & Ye, C. (2024). Advancing decision-making: The role of management analytics in modern business practices. Nanotechnologies in Construction, 16(5), 431–440. https://doi.org/10.15828/2075-8545-2024-16-5-431-440

Lu, Y., Sigov, A. S., Ratkin, L., Ivanov, L. A., & Zuo, M. (2023). Quantum computing and industrial information integration: A review. Journal of Industrial Information Integration, 35, 100511. https://doi.org/10.1016/j.jii.2023.100511

McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 54, 1273–1282. https://doi.org/10.48550/arXiv.1602.05629

Neethirajan, S. (2020). The role of sensors, big data and machine learning in modern animal farming. Sensing and Bio-Sensing Research, 29, 100367. https://doi.org/10.1016/j.sbsr.2020.100367

Stranieri, S., Riccardi, F., Meuwissen, M. P. M., & Soregaroli, C. (2021). Exploring the impact of blockchain on the performance of agri-food supply chains. Food Control, 119, 107495. https://doi.org/10.1016/j.foodcont.2020.107495

Wu, H. P., Liu, Z., Dong, H. Y., Lu, Y., & Xu, L. D. (2025). Revolutionizing internal auditing: Harnessing the power of blockchain. Enterprise Information Systems, 19(1–2). https://doi.org/10.1080/17517575.2024.2448003

Xu, L. D., Lu, Y., & Li, L. (2021). Embedding blockchain technology into IoT for security: A survey. IEEE Internet of Things Journal, 8(13), 10452–10473. https://doi.org/10.1109/JIOT.2021.3060508

Ye, Z., & Lu, Y. (2022). Quantum science: A review and current research trends. Journal of Management Analytics, 9(3), 383–402. https://doi.org/10.1080/23270012.2022.2089064

Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. Journal of Industrial Information Integration, 23, 100224. https://doi.org/10.1016/j.jii.2021.100224

Zhang, H., & Lu, Y. (2025). Web 3.0: Applications, opportunities and challenges in the next internet generation. Systems Research and Behavioral Science, 42(4), 996–1015. https://doi.org/10.1002/sres.3045

Zheng, X. R., & Lu, Y. (2022). Blockchain technology: Recent research and future trend. Enterprise Information Systems, 16(12), 1939895. https://doi.org/10.1080/17517575.2021.1939895