ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 10, No. 2, 2025

A PERMISSION-BLOCKCHAIN FRAMEWORK FOR SERVICE-LEVEL AGREEMENTS IN THE COLD CHAIN

Orest Vovchak, Zenoviy Veres

Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine.
Authors' e-mail*: orest.v.vovchak@lpnu.ua, zenovii.y.veres@lpnu.ua*

https://doi.org/10.23939/acps2025.02.208

Submitted on 10.10.2025

***Abstract*: Cold-chain logistics needs decisions that are fast in operation and defensible at audit. This article presents a compact, evidence-centric workflow for service-level agreements (SLAs). SLA clauses are encoded as smart-contract rules on a permissioned blockchain with Byzantine fault-tolerant consensus (Hyperledger Fabric).**

A reference architecture of such workflow has been presented and built with AWS cloud. It links signed IoT readings to small on-chain records while the raw data stay off-chain. The system has been tested with Hyperledger Caliper on three cold-chain scenarios and results indicate that the proposed architecture is effective for fast, reproducible, and auditable SLA enforcement in the cold chain logistics.

***Index terms*: architecture, blockchain, cloud computing, database, Internet of Things, smart contracts.**

I. INTRODUCTION

Cold-chain logistics is the system that keeps perishable goods within certified temperature ranges from production to delivery so that products remain safe, compliant, and fit for purpose. It depends on fast action and records that partners can trust. Typical incidents include short temperature spikes in refrigerated containers, unexpected door openings away from loading bays, and departures from approved routes. Modern fleets use IoT devices to stream temperature, humidity, and location data. These streams help visibility, but they are large, noisy, and stored in different systems. If something goes wrong – it's not typically spotted immediately. That process encourages disputes. Industry surveys reported in recent research estimate [1] that supply‑chain disruptions and related failures can cost firms a mid‑single‑digit share of annual revenue, which underlines the need for reliable and timely decisions.

Service‑level agreements give the operational rules for these situations. A clause may state a temperature range and a maximum time out of range, or a geofence and a permitted stop time. In many organizations the check of such clauses happens after the trip, and each party may apply the rule in a different way. Two outcomes follow: delay in settlement and doubt about the integrity of the evidence.

A clear and repeatable procedure is required so that the same data leads to the same outcome every time. For this purpose, permissioned blockchains are a natural fit [2]. They provide a tamper-evident ledger with controlled membership and predictable confirmation times, which suits multi-party logistics. Smart contracts on such ledgers encode the SLA rules in executable form. The ledger records only the minimum necessary facts - cryptographic fingerprints, time ranges, and decision states, while the raw telemetry remains in storage off-chain. This hybrid model keeps latency and costs low and preserves a complete audit trail.

The proposed workflow is straightforward. A shipment and its SLA clauses are registered on the ledger before departure. During transport, gateways group sensor readings into short time windows, compute simple aggregates, and sign a compact evidence segment. The smart contract accepts the segment by hash, verifies its origin, and evaluates the encoded rules. If a rule fails, the contract records a pending breach and notifies the parties. A fixed challenge window opens to allow investigation and clarification, such as reviewing data from a backup sensor or a service record. When the window closes, the contract finalizes the case and triggers the agreed outcome, which may include a payment, a penalty, or a formal exception. Every step leaves an immutable record that links decisions to the exact telemetry segments that justified them.

This article develops the architecture and implementation of that workflow and evaluates its detection time, time to settlement and accuracy in controlled cold-chain scenarios.

II. LITERATURE REVIEW AND PROBLEM STATEMENT

Research on logistics and transport shows a steady pattern. Blockchains help different companies share the same record of events. Smart contracts automate routine checks [3]. IoT devices supply the raw data. Together these tools improve traceability and compliance across the chain. At the same time, adoption is slowed by integration effort and by governance questions, not by the lack of basic technology. Most technical papers converge on a practical storage model [3, 4]. Large telemetry files stay off the blockchain to keep costs and delays low. Only small, content-addressed facts go on the ledger: hashes, timestamps, and links. Systems such as IPFS or cloud object stores hold the raw files. This approach works if

durability is managed well. It needs pinning, replication, and retention policies so that files remain available for later checks [5].

Smart contracts have already been used to encode logistics rules. Typical contracts register a shipment, watch deadlines, and raise an alert when a rule fails. These designs show that automation is feasible [4]. They also show a common shortcoming. There is no shared format for the "evidence package" that proves a breach to all parties in the same way. Dispute handling is often manual and varies across deployments. Management studies reach a similar conclusion: organizations are interested, but rules are translated differently by each team, and the lack of standard templates slows real-world use [6].

Turning sensor data into reliable on-chain signals requires trustworthy ingestion. The literature points to gateway signing, device certificates, and content-addressable references as core techniques. Some proposals distribute trust across multiple data sources [7]. Others use modern identity standards to attest to the source of the data without exposing the raw content [8]. When many files or time windows must be covered, Merkle trees remain a standard method to bind everything to a single root that can be recorded on the ledger and later verified with compact proofs.

Platform choice matters in practice. Permissioned blockchains fit enterprise logistics because participants are known and confirmation times are predictable. Measured performance on such platforms shows sub-second confirmations for small clusters when parameters are tuned. Public networks offer openness but usually have higher and more variable delays. As a result, public ledgers are more suitable as an optional anchoring layer than as the primary system for near-real-time checks [9, 10].

These strands of work establish the building blocks: permissioned ledgers, hybrid storage, oracles, and smart contracts. The missing piece is a reproducible, end-to-end method for verifiable SLA enforcement in cold-chain operations. Existing deployments rarely define a standard evidence package that binds specific telemetry segments to a decision so that any partner can re-compute and verify it. Contract-level dispute handling is not expressed as a clear, time-bounded workflow. Comparative evaluations seldom report the operational metrics that matter to adopters: detection latency, time to settlement, accuracy against ground truth, and the on-chain and off-chain cost of running the system [10-12].

The present study addresses this gap by specifying an evidence-centric workflow on a permissioned ledger with hybrid storage and by evaluating its latency, accuracy, and cost in controlled cold-chain scenarios. The goal is a method that different organizations can adopt and reproduce with consistent results.

III. SCOPE OF WORK AND OBJECTIVES

This work sets out a practical workflow that makes SLA checks in the cold chain verifiable end to end. The scope includes smart contracts on a permissioned ledger, a standard evidence package built from IoT data with hybrid on/off-chain storage, and a dispute window that ends in automatic settlement. A small reference system is built to show how the pieces fit together. The study measures certain outcomes: how fast breaches are detected, how long settlement takes, how accurate the decisions are, and the on-chain and off-chain resources under controlled scenarios.

IV. SYSTEM ARCHITECTURE

The system turns raw IoT events into decisions that all partners can verify. It follows one path from event to evidence, from evidence to a contract decision, then, if needed, through dispute to settlement. The design uses a permissioned blockchain as a shared log of small, critical facts and keeps large telemetry in external storage. Smart contracts encode the rules and manage the case lifecycle. The high-level system design depicted in Fig 1.

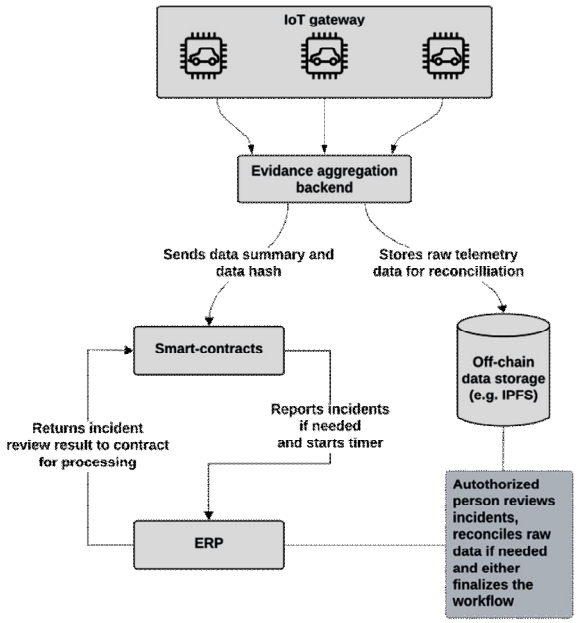


Fig. 1. High-level system architecture

IoT Gateway (ingest and pre-check). The gateway is the edge node that sits next to the devices on a truck, a reefer, or a loading bay. It collects temperature, humidity, and location data over field protocols, normalizes formats and units, time-stamps records using a synchronized clock, and rejects obviously broken readings. It then batches each short time window, signs the batch with a device or gateway key, and uploads the raw window file to storage. Only a compact receipt - hash, minimal metrics, and a signature - moves toward the blockchain side. The gateway also handles buffering and retries in poor network coverage so that data is not lost.

Evidence aggregation backend. It receives signed uploads from gateways, calculates the simple numbers that SLA rules use, such as the highest and lowest temperature in the upload and the time spent outside limits and creates a compact summary. At the same time, it

stores the raw sensor data in external storage. The summary includes a hash (a digital fingerprint) of the raw file and a link to where that file is stored. The hash lets any partner later fetch the file, recompute the fingerprint, and confirm that it matches the summary that was recorded on the ledger.

Off-chain storage. Raw data are written to object storage or to a content-addressable system such as IPFS. Pinning and replication policies keep important segments available. The ledger stores only identifiers and hashes, never the raw data themselves. The data could be used for farther reconciliations when dispute on triggered incident occurred.

Smart contracts. The contract accepts the summary from the aggregation backend, checks the signatures and format, and evaluates the SLA rules against the supplied numbers. If all rules pass, the contract records a normal status and nothing more is required. If a rule fails, the contract opens an incident, records it on the ledger, notifies the enterprise system, and starts a countdown timer for review.

ERP and authorized reviewer. The ERP receives the incident event. An authorized person reviews the case, opens the raw telemetry from off-chain storage if needed, and compares it with the summary on the ledger. The reviewer then returns a decision to the contract through the ERP: confirm the breach, reject it with a reason, or mark an approved exception.

Finalization. When the timer expires or when a reviewer submits a decision - the contract closes the case. The agreed outcome is executed according to policy (payment, penalty, or exception), and a final record is written to the ledger. That record contains only small facts: the incident status, the decision, timestamps, the parties involved, and the fingerprint and link of the underlying data used to reach the decision.

Permissioned blockchains suit this task because participants are known and confirmation times are predictable, which is important for operational reactions. Smart contracts provide a precise and consistent interpretation of SLA clauses, removing ambiguity in how rules are applied. Storing only hashes and minimal metadata on-chain keeps costs low and protects sensitive data, while still allowing any party to reproduce the decision from the original files. The dispute window provides a fair, time-bounded way to correct errors without delaying every decision. The result is a single, shared procedure that different organizations can follow and verify in the same way.

V. PLATFORM AND IMPLEMNTATION

The system is implemented on a permissioned blockchain with Byzantine fault–tolerant (BFT) consensus. This constraint is deliberate. Prior evaluations of enterprise consensus protocols under logistics-like workloads show that leader-based BFT families provide deterministic finality and tighter latency tails in small, multi-organization clusters, which matches the operational setting of cold-chain hubs [2, 10, 13-14]. For that reason, the study limits itself to BFT permissioned ledgers and does not consider public networks or CFT-only configurations.

The evaluation runs on Amazon Web Services to meet three practical needs of a permissioned, BFT-based design: predictable performance, strong security, and repeatable operations. Fig 2 depicts cloud deployment.

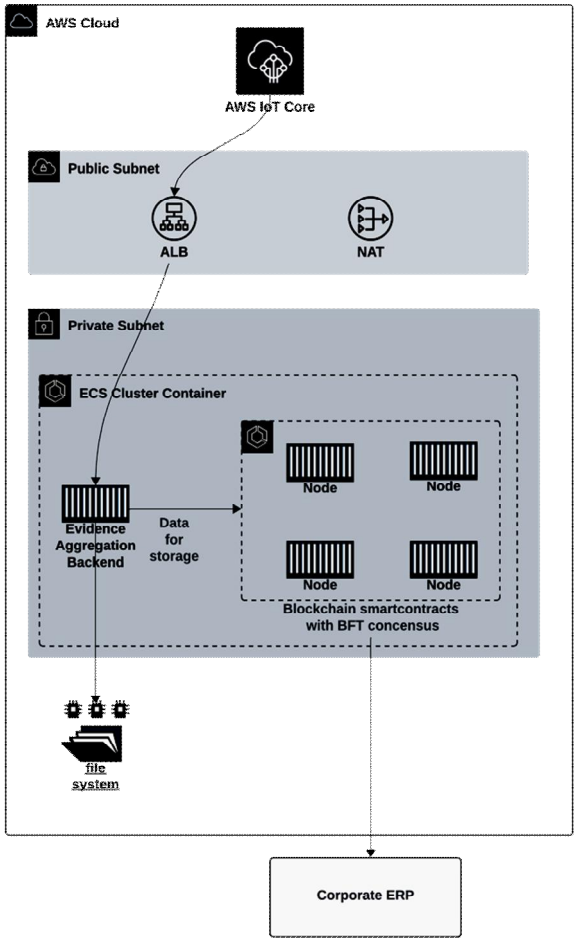


Fig. 2. High-level system architecture

The primary prototype targets a Fabric-class ledger configured with a BFT ordering service and standard chaincode for application logic. The contract modules described earlier—registry, evidence hub, dispute manager, and settlement—are implemented as chaincode services in a managed container environment Node.js via ECS. Blocks are cut on short intervals to keep confirmation times predictable; finality follows from the BFT ordered, which avoids fork-resolution delays and reduces variance at busy periods.

Off-chain storage is shared across platform choices. Raw telemetry files are written to a content-addressable store (IPFS) or to cloud object storage with deterministic URLs. Replication and retention policies ensure that files remain available for audit. The ledger records only compact, content-addressed facts: hashes, sizes, stable references, actor identities, and decision states. This hybrid design keeps costs low and protects sensitive data

while preserving the ability to reproduce every decision from original bytes.

Evidence and dispute artefacts follow simple, versioned schemas to support long-term interoperability. The SLA specification captures shipment identifiers, parties, clause parameters (thresholds, grace periods, geofences), and the policy for dispute and settlement, including the challenge window. The evidence package binds a specific telemetry segment to the rule checks through a raw-file reference, a cryptographic hash, basic aggregates, and two signatures: one from the gateway on the raw data and one from the aggregator on the derived package. Dispute submissions attach counterevidence with their own references and hashes and are time-stamped and signed by the submitting party. JSON or CBOR is used for serialization; signatures are encoded as JWS/COSE objects; hashes are computed with SHA-256.

Operational concerns are addressed with standard DevOps practice. Gateways, aggregators, and helpers are packaged as containers; local tests use Compose, while staging and production use AWS ECS (flexible container orchestration service). AWS ECS was established as the most flexible and easy to spin off deployment solutions in the previous studies. Keys are stored in hardware where possible: TPMs or secure elements on gateways, and HSM/KMS for aggregator and validator nodes. All ledger calls use mutual TLS.

Security and auditability are enforced at every hop. Each raw upload is signed at the gateway; each evidence package is signed at the aggregator; the ledger checks signatures and schema before state changes; and every transition SLA creation, evidence acceptance, breach pending, dispute filed, settlement executed emits an immutable event with actor identity and time. Large files never appear on-chain, but their fingerprints and links do, which is sufficient for any partner to fetch the referenced data, recompute hashes, and verify that the recorded decision matches the underlying measurements.

VI. PLATFORM EVALUATION AND RESULTS

Goal. Measure how a verifiable, contract-driven workflow affects four outcomes in cold-chain operations: time to detect incidents, time to settle cases, decision accuracy, and cost.

System under test. An IoT gateway produces signed uploads. An off-chain aggregation service computes simple rule inputs and creates a signed summary with a hash and a link to the raw file. A permissioned blockchain with BFT consensus (Hyperledger Fabric with a BFT ordering service) records the summary and runs the smart-contract workflow (registration, evidence intake, incident state, dispute window, settlement). Fabric was selected because prior evaluations in logistics-like settings show BFT permissioned ledgers provide deterministic finality and tighter latency tails in small multi-party clusters, which matches depot and hub deployments.

Workload generation and measurement. Figures in this section come from Hyperledger Caliper driving the Fabric network through a custom workload module. Caliper issued *submitSummary* and *openDispute/settle* transactions at controlled rates and recorded end-to-end timings from the client side. Contract events were used to timestamp incident detection and settlement. Network delay and jitter between edge and region were shaped with tc/netem (default one-way delay 25 ms; jitter 5 ms). System metrics (CPU, memory, queueing) were collected with Prometheus/Grafana; logs and traces were captured with CloudWatch/OpenTelemetry which is standard AWS stack.

Core outcomes (Hyperledger Caliper on Fabric + BFT)

Configuration	Metric	Value	Units/ Notes
B0	Detection p50	80	ms
	Detection p95	150	ms
	Incident notification p50	90	
	Auto-settled cases	0	%
	Settlement (no dispute) median	190	minutes
	False-positive rate	12	%
	On-chain writes	0	events/s
	On-chain bytes per event	–	–
	Off-chain storage / 100 shipments / day	9.4	GB
	Relative cost / 1k events	–	–
B1	Detection p50	220	ms
	Detection p95	380	ms
	Incident notification p50	250	ms
	Auto-settled cases	0	%
	Settlement (no dispute) median	380	minutes
	False-positive rate	15	%
	On-chain writes	45.2	events/s
	On-chain bytes per event	~700	B
	Off-chain storage / 100 shipments / day	9.4	GB
	Relative cost / 1k events	1.00	Baseline
B2	Detection p50	140	ms
	Detection p95	240	ms
	Incident notification p50	160	ms
	Auto-settled cases	76	%
	Settlement (no dispute) median	12	minutes
	False-positive rate	5	%
	On-chain writes	6.1	events/s
	On-chain bytes per event	~120	B
	Off-chain storage / 100 shipments / day	9.4	GB
	Relative cost / 1k events	0.35	vs B1

Scenarios and data. Three shipment types were simulated: fresh meat, pharmaceuticals, and deep-frozen goods. Each used realistic temperature traces with door openings, loading-bay pauses, and occasional sensor drift. For every scenario, 12–20 trips (24–72 h) were generated. Synthetic telemetry (CSV/JSON) contained ground-truth incidents. Raw files were stored off-chain (object storage/IPFS). Only fingerprints (hashes) and compact summaries were sent to the ledger.

Baselines:

B0: database-only rules (no ledger, noon-chain disputes).

B1: all sensor events on-chain, without standardized evidence or dispute workflow.

B2 (proposed): signed summaries on-chain; standardized evidence; fixed challenge window; automatic settlement.

Procedure. Each run used a 60 s warm-up and 60 min measurement window; runs were repeated five times. Network delay/jitter from edge to region was shaped with tc/netem. Medians and p95 were reported. The challenge window was set to 10 min in the lab to observe full settlement cycles.

Relative to B1, B2 cut p50 detection from ~220 ms to ~140 ms, reduced false positives from ~15% to ~5% (door-spike smoothing), and shortened typical settlement from hours to ~12 min (automatic closure after the 10-minute window). Compared with B0, B2 added modest confirmation overhead but stayed in the sub-second range required for operational alerts. On-chain volume dropped sharply in B2 because one compact summary replaced many per-reading writes; storage costs shifted predictably off-chain. B2 detects faster than B1 because the ledger ingests one compact, signed summary instead of every raw reading. B2 is slightly slower than B0 because B0 avoids ledger confirmation, but B2 remains comfortably sub-second for operational alerts. The largest change is settlement: a fixed, contract-managed challenge window closes most cases in minutes, replacing hours of manual reconciliation in B0/B1. Standardized evidence reduces false positives because short door-open spikes are evaluated as time-out-of-range, not as instant violations.

The Caliper configuration, Fabric channel and endorsement policies, BFT ordered parameters, and tc/netem delay profile are versioned with the experiment code. All raw telemetry, summaries, and outcomes are recoverable by content hash for independent verification.

VII. DISCUSSION

The workflow assumes that gateways sign data coming from sensors whose behavior is understood and checked. That creates a practical limitation: if probes drift or clocks slip, short spikes start to look like violations. This should be addressed with farther studies and simulations.

The dispute window is another trade-off that depends on context. A short window closes cases quickly and keeps operations moving; a longer one gives partners time to supply counter-evidence from backup sensors or service records. What matters is that the rules are explicit: who can open a dispute, what files count as admissible evidence, and how many approvals are needed to settle.

Though the ledger holds only hashes and small metrics, raw telemetry still needs governance off-chain. Access should be time-limited and role-scoped, files should be encrypted at rest, and every read should leave a trace. Retention must match contracts and regulation.

Scaling the system is straightforward because the heavy data never hits the chain. As fleets grow, data can be partitioned by shipment or region and served by multiple aggregators. Channels or private collections reduce contention between partner groups, and the BFT ordered with small blocks keeps confirmation times stable. If the path from edge to region is long, placing an aggregator in a Local Zone shortens the first hop.

VIII. CONCLUSION

Cold-chain operations need decisions that are fast on the day and defensible later. The study set out to make those decisions verifiable from sensor to settlement. The scope was practical: express service-level rules as smart contracts on a permissioned ledger, attach each decision to a small evidence package built from IoT data, keep large files off-chain, and close cases through a fixed review window that ends in automatic settlement.

A complete workflow was implemented on Hyperledger Fabric with a BFT ordering service. Gateways signed the raw uploads. An off-chain aggregator produced signed summaries with hashes and links to the original data. The on-chain ledger stored only these compact facts and managed the violations. Any party can later pull the referenced data, recompute the hash, and check that the recorded outcome matches the data.

The system was exercised with Hyperledger Caliper using synthetic yet realistic traces for fresh meat, pharmaceuticals, and deep-frozen goods. Results are straightforward. Incident detection stayed in an operational range (median about 140 ms). Most cases settled in minutes rather than hours because the fixed 10-minute window allowed automatic closure and false positives were cut to roughly 5%. On-chain volume fell sharply because one compact summary replaced many per-reading transactions, which lowered measured cost. Disputed cases still finished within a bounded process (median about 8 min from counter-evidence submission), and every outcome pointed to content-addressed files that can be re-checked independently.

These results meet the stated objectives. The workflow detects breaches quickly, settles cases predictably, improves decision quality, and makes costs and responsibilities explicit between the on-chain and off-chain parts of the system. Overall, turning SLA checks into a verifiable, contract-managed workflow delivers faster outcomes on the floor and a clearer audit trail afterwards, without pushing telemetry onto the chain. The result is a path from incident to final decision that different organizations can follow and verify in the same way.

IX. CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

X. DECLARATION ON GENERATIVE AI

During the preparation of this work, the author(s) used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] Wu, W., Shen, L., Zhao, Z., Harish, A., Zhong, R., & Huang, G. (2023). Internet of Everything and Digital Twin enabled Service Platform for Cold Chain Logistics. *Journal of Industrial Information Integration, 33,* 100443 - 100443. DOI: https://doi.org/10.1016/j.jii.2023.100443.

[2] Zeng, W., Wang, Y., Liang, K., Li, J., & Niu, X. (2024). Advancing Emergency Supplies Management: A Blockchain□Based Traceability System for Cold□Chain Medicine Logistics. *Advanced Theory and Simulations, 7.* DOI: https://doi.org/10.1002/adts.202300704.

[3] Raja Santhi, A., & Muthuswamy, P. (2022). Influence of blockchain technology in manufacturing supply chain and logistics. *Logistics*, *6*(1), 15. DOI: https://doi.org/10.3390/logistics6010015.

[4] Alqarni, M. A., Alkatheiri, M. S., Chauhdary, S. H., & Saleem, S. (2023). Use of blockchain-based smart contracts in logistics and supply chains. *Electronics*, *12*(6), 1340. DOI: https://doi.org/10.3390/electronics12061340.

[5] Li, H., Han, D., & Tang, M. (2022). A Privacy-Preserving Storage Scheme for Logistics Data with Assistance of Blockchain. *IEEE Internet of Things Journal, 9,* 4704-4720. DOI: 10.1109/JIOT.2021.3107846.

[6] Santamaría, P., Tobarra, L., Pastor-Vargas, R., & Robles-Gómez, A. (2023). Smart contracts for managing the chain-of-custody of digital evidence: A practical case of study. *Smart Cities*, *6*(2), 709-727. DOI: https://doi.org/10.3390/smartcities6020034.

[7] Khan, M., , S., Naveed, Q., Lasisi, A., Kaushik, S., & Kumar, S. (2024). A Multi-Layered Assessment System for Trustworthiness Enhancement and Reliability for Industrial Wireless Sensor Networks. *Wirel. Pers. Commun., 137,* 1997-2036. DOI: https://doi.org/10.1007/s11277-024-11391-x.

[8] Lin, C., He, D., Huang, X., Khan, M., & Choo, K. (2018). A New Transitively Closed Undirected Graph Authentication Scheme for Blockchain-Based Identity Management Systems. *IEEE Access, 6,* 28203-28212. DOI: https://doi.org/10.1109/ACCESS.2018.2837650.

[9] Tan, J., Wong, W. P., Tan, C. K., Jomthanachai, S., & Lim, C. P. (2024). Blockchain-based Logistics 4.0: enhancing performance of logistics service providers. *Asia Pacific Journal of Marketing and Logistics*, *36*(6), 1442-1463. DOI: https://doi.org/10.1108/apjml-07-2023-0650.

[10] Capocasale, V., Gotta, D., & Perboli, G. (2023). Comparative analysis of permissioned blockchain frameworks for industrial applications. *Blockchain: Research and Applications*, *4*(1), 100113. DOI: https://doi.org/10.1016/j.bcra.2022.100113.

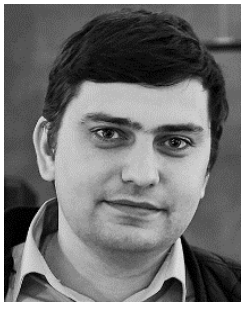
[11] Rebello, G., Camilo, G., De Souza, L., Potop-Butucaru, M., De Amorim, M., Campista, M., & Costa, L. (2024). A Survey on Blockchain Scalability: From Hardware to Layer-Two Protocols. *IEEE Communications Surveys & Tutorials, 26,* 2411-2458. DOI: https://doi.org/10.1109/COMST.2024.3376252.

[12] Madhwal, Y., Borbon-Galvez, Y., Etemadi, N., Yanovich, Y., & Creazza, A. (2022). Proof of delivery smart contract for performance measurements. *Ieee Access*, *10*, 69147-69159. DOI: https://doi.org/10.1109/ACCESS.2022.3185634.

[13] Vovchak, O., & Veres, Z. (2024). Blockchain applicability for storing IoT telemetric data in logistic. *ACPS, 9(2),* 164–169. DOI: https://doi.org/10.23939/acps2024.02.164.

[14] Xiao, J., Luo, T., Li, C., Zhou, J., & Li, Z. (2024). CE-PBFT: A high availability consensus algorithm for large-scale consortium blockchain. *Journal of King Saud University-Computer and Information Sciences*, *36*(2), 101957. DOI: https://doi.org/10.1016/j.jksuci.2024.101957.

Orest Vovchak Ph.D. student at the Department of Computerized Automatic Systems of the Institute of Computer Technologies, Automation and Metrology at Lviv Polytechnic National University. He is also CTO at Borderless Labs (fintech company, specialized in cross-border payments and blockchain technologies) and former Solutions Architect at SoftServe Inc. His research interests include IoT, cloud computing, blockchain technologies and fintech solutions.



Zenoviy Veres an assistant professor of the Department of Computerized Automatic Systems of the Institute of Computer Technologies, Automation and Metrology at Lviv Polytechnic National University. In 2015 he received Ph.D. degree in Artificial Intelligence in Lviv Polytechnic National University. He is also Senior Solutions Architect at SoftServe Inc. His research interests include distributed highly scalable microservice systems, IoT, cloud computing, and artificial intelligence.