

A Digital Chain-of-Custody Architecture for Freight Documents: Event-Driven Capture, Transformation, Delivery, and Retrieval

Prasanna Sundaravaradan

Submitted:01/04/2022

Revised:19/05/2022

Accepted:29/05/2022

Abstract—There are several systems in play for Bills of Lading, Proof of Delivery records, cargo documents, receipts, photos, trip signoffs, and exception reports before they reach operations teams and customers. These are all cycles in which documents can be delayed, duplicated without proper tracking, be modified without proper traceability, sent to the wrong recipient, or not be associated with the trip or driver for which they were created. In this paper, a digital chain-of-custody architecture for transportation documents is introduced which records each of the following independently auditable events: capture, metadata validation, content transfer, format conversion, archival, customer notification, retrieval and retention. It features content hashes, an immutable processing history, claim-check messaging, role-based access control, delivery receipts, and policy-governed retention, and contrasts with paper-and-e-mail workflows, electronic data interchange and blockchain-only custody schemes on parameters of tamper evidence, delivery reliability, format interoperability, access granularity, retrieval speed, and deployment simplicity. A method is proposed for evaluating representative capture-to-retrieval workflows that incorporates a series of metrics, including custody completeness, delivery latency, duplicate suppression, retrieval precision and recall, and recovery time. The architecture provides a design pattern that can be reused to build an auditable digital freight-document infrastructure that doesn't require consensus overhead like that of public blockchain designs, while still providing the fundamental tamper-evidence guarantee.

Keywords—*chain of custody; digital freight documents; event-driven architecture; proof of delivery; document capture; content hashing; claim-check pattern; role-based access control; retention policy; audit trail; distributed ledger; transportation logistics*

I. INTRODUCTION

Each freight movement in networks is supported by bills of lading, proof-of-delivery documentation, cargo receipts, trip photographs, exception reports and signoff forms. They are generated at pickup or delivery, often on a handheld device used by a driver, then they pass through capture software, transformation workflows, storage systems, notification systems, long-term archives and then to dispatchers, billing systems, auditors and customers. Traditional document workflows that rely on paper handoffs, facsimile transmission and unstructured e-mail attachments break this lifecycle, with a document potentially delayed at a handoff point, duplicated across inboxes, being converted to another incompatible format without records of changes, sent to an unintended recipient or lost from the trip, driver,

*Lead Developer,
Anveta Inc*

and shipment context in which it was created. In the event of a dispute regarding delivery time, condition of cargo, or authenticity of documents, there is not an account of the capturing, altering and transmission of the document which can be verified and it becomes a material business and legal obligation.

This paper presents a digital chain-of-custody model for freight documents that takes each phase of the document life cycle (document capture, transformation, delivery, and retrieval) as an independent auditable event. Instead of having a single monolithic transaction that is used to store the history of a document, the architecture models the history of a document as a sequence of events that are recorded as distinct events, each sent to an event bus with a timestamp, and each event is cryptographically linked to the event that precedes it. The design is inspired by four streams of practice: structured business-document exchange: electronic data interchange (EDI); tamper-evident record keeping: distributed-ledger and

blockchain-based custody research; event-driven microservice architecture: asynchronous, loosely coupled service communication; role-based access control and role mining: fine-grained access governance [1].

II. BACKGROUND AND RELATED CONCEPTS

Freight-document handling has gone through some paradigms. For decades, paper and facsimile records were the primary means of communication and had legal familiarity, but no means of proving a document was not altered since it was signed. Reduces manual re-entry of data and speeds up processing by introducing structured, machine-readable message formats that are used to exchange shipping instructions, invoices and delivery confirmations between trading partners, but schemas were rigid, batch-oriented and required a bilateral trading-partner agreement to be flexible when a shipment involves parties that are not part of an existing EDI relationship [2]. The next step was to switch from text to images and from e-mail to PDF, which added flexibility and

decreased the reliance on point-to-point integration, but created a new recordability problem: an e-mailed PDF lacks any built-in record of its transformation, when, and from where. Since then, the emergence of the Internet of Things (IoT) sensing and mobile telematics has allowed the collection of trip events to be captured in real time from the vehicle, or handheld device, in real time and with location, providing even more raw material for custody records and increasing the amount of risk that supply-chain participants must deal with. More recently, proposals for distributed ledger or blockchain systems have focused directly on issues of providing tamper-proof evidence and managing custody, in which cryptographic hash chaining, consensus mechanisms and smart contracts have been used to ensure that altering a record is a computationally detectable process, with the forensics-chain designs extending this concept specifically to the tracking of custody [3]. These paradigms are compared in Table I in terms of the various dimensions related to freight-document handling.

TABLE I COMPARISON OF FREIGHT-DOCUMENT HANDLING PARADIGMS

Dimension	Paper / E-mail	EDI-Based	Blockchain-Only	Proposed Architecture
Tamper evidence	None	Low	High (consensus)	High (hash-linked)
Integration flexibility	High (ad hoc)	Low (rigid schema)	Moderate	High (event-driven)
Delivery reliability	Low	Moderate–High	Moderate	High (pub/sub)
Retrieval speed	Low (manual search)	Moderate	Low (ledger scan)	High (indexed)
Deployment overhead	Low	Moderate	High (consensus)	Moderate

This proposed architecture is not public blockchain, consensus blockchain, but rather a different approach. In the operations management context, distributed-ledger literature confirms that a public consensus mechanism would add significant transaction costs, latency, and governance overhead when the group of carriers responsible for the custody of the document is the same in every instance or a relatively small and known group of trading partners. Instead, it takes the fundamental unit of custody used in blockchain systems — the append-only event record — and adds two other sets of practices. Event-driven microservice architecture provides the message backbone for the movement of documents and

metadata between capture, transformation, delivery, and retrieval services as message events on a publish/subscribe bus, with at least once or exactly once guarantees for the delivery of each event. Permitting access to captured documents is structured based on role-based access control and role-mining research, meaning only the custody events and content visible to drivers, dispatchers, customers and auditors [4].

III. PROPOSED ARCHITECTURE

The proposed architecture consists of four layers, capture, transformation, delivery and retrieve and retain, each of which writes to the immutable

processing history, but is prohibited from modifying or deleting from the processing history. All of the layers are of a layered nature and the ledger is cut across the layers, as shown in Fig. 1.

A. Capture Layer

Documents are captured in the capture layer using a mobile capture application at pickup or delivery, capturing the image of the document or the electronic signature, along with a time stamp, geolocational

information and the identifiers of the trip, driver and shipment captured [5]. An event publisher will publish a document. Captured event into the event bus after a metadata validator has validated the captured document to ensure it includes the minimum metadata needed to process downstream. Rather than reject documents that are only partially captured downstream, rejecting them at this boundary ensures that the documents don't propagate through the rest of the pipeline [6].

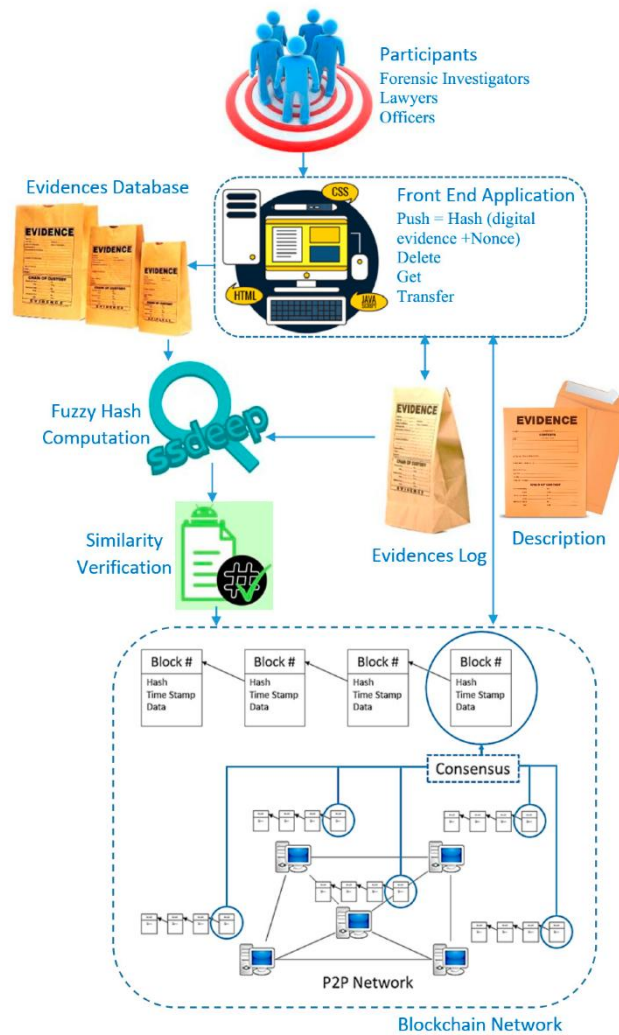


Fig. 1. A Procedure for Tracing Chain of Custody in Digital Image Forensics: A Paradigm Based on Grey Hash and Blockchain (MDPI, 2021)

B. Transformation Layer

The transformation processing is triggered by an event captured by a document. A format converter transforms photos, scanned images, or native mobile-signature format data to a long-term-preservation guaranteed format like PDF/A. A hash-and-sign service calculates a cryptographic “content hash” of the transformed document and attaches it to the identity of the “transforming service” and the time of

the transformation; the content hash is not the content of the document itself, but rather it is the service provider's specific implementation of the transformation that is being referenced by subsequent “custody events”. Following the claim-check messaging pattern designed to maintain a clear pointer to where a large payload is kept, but to remove it from the message bus, the transformed binary is stored in an

object store, and only a short object reference — the claim check — is sent on subsequent bus messages [7].

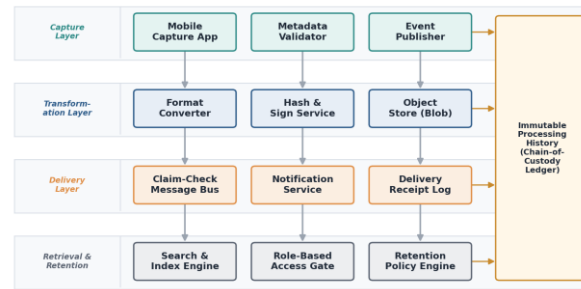


Fig. 1. Layered architecture: capture, transformation, and delivery layers feed a retrieval and retention layer, with every stage writing to a cross-cutting immutable custody ledger.

C. Delivery Layer

The delivery layer provides access to the parties who have a right to receive a transformed, hashed document. The claim-check message bus carries lightweight delivery events that refer to the object in storage, a notification service notifies the appropriate delivery dispatcher, billing system or customer portal, and a delivery-receipt log produces an acknowledgment event when the recipient system or user confirms receipt. This decoupling of the heavy weight document object and the lightweight event referencing the object enables the delivery layer to be scaled to accommodate document size and enables the delivery layer to retry failed notifications without re-transmitting the content of the document [8].

D. Retrieval and Retention Layer

The retrieval and retention layer exposes captured documents, which are transformed, to operational and customer search [9][10]. A search and index engine indexes metadata of documents such as trip, driver, document type, capture time, and delivery status, allowing for fast query without a full scan of the object store. A role-based access gate filters requests for all retrievals by the requester's role and a retention policy engine controls the duration of which a document and associated custody events are retrievable prior to archival or deletion based on regulatory and contractual requirements of the carrier.

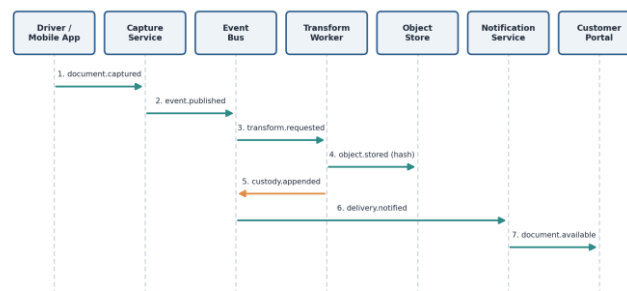


Fig. 2. Event-driven document lifecycle: seven ordered events carry a single document from mobile capture through to availability in the customer portal.

Fig. 2 shows the sequence of events that occur as one document traverses all four layers: from the document. Captured event raised by the mobile application, to the document. Available event that

indicates document availability in the customer portal. Table II lists the main types of events that appear in the pipeline and Table III lists the metadata schema attached to each custody record [11].

TABLE II PRINCIPAL EVENT TYPES IN THE CUSTODY PIPELINE

Event Type	Emitted By	Description
document.captured	Mobile capture app	Raw document/signature recorded with timestamp, geolocation, and trip context
event.published	Event publisher	Validated capture event placed on the event bus
transform.requested	Event bus	Signal to convert and hash the captured document
object.stored	Object store	Transformed binary persisted; content hash recorded
custody.appended	Hash-and-sign service	New hash-linked record appended to the immutable ledger
delivery.notified	Notification service	Recipient alerted that a document is available
document.available	Customer portal	Document confirmed retrievable by the intended recipient

TABLE III**PROPOSED CUSTODY-RECORD METADATA SCHEMA**

Field	Description
document_id	Unique identifier for the document instance
trip_id / shipment_id	Reference to the associated trip and shipment
actor_id, actor_role	Identity and role of the party generating the event
event_type	One of the event types listed in Table II
capture_timestamp	Time the underlying document activity occurred
geolocation	Coordinates recorded at capture, where applicable
this_hash / prev_hash	Content hash of this event and of its predecessor
signature	Digital signature binding the event to the actor's credential

IV. CHAIN-OF-CUSTODY AND SECURITY MECHANISMS

The proposed architecture is based on the integrity guarantee, where each custody event is hash-linked to the previous custody event, as shown in Fig. 3. Each record in the immutable processing history contains the cryptographic hash of the previous event in the processing history (prev_hash), the cryptographic hash of the event it contains (this_hash), a copy of the identity and role of the party that processed the event and a timestamp. Each event's hash depends on both the content of the event itself and the hash of the previous event: if you change or reorder any historical event, then the hash of that event will change, which will get a pass during a simple chain verification as a warning sign of the possibility of silent tampering, without needing a distributed consensus protocol to detect [12][13]. This design intentionally sacrifices Byzantine fault tolerance of public blockchain consensus for the lower latency, cost and operational

complexity that is suitable to a custody chain that stretches across a single carrier's systems and a limited set of known trading partners, while distributed-ledger research also warns against implementing full consensus mechanisms when transaction volume or trust boundaries do not justify its overhead.

Digital signatures are added to hash-linking to make each event the signature of the service account or authenticated user that created it, which makes the forging of the actor field after the fact to invalidate the signature [14]. Role-based access control is implemented as per the role of the freight organization, which includes the following: Driver, Dispatcher, Billing Clerk, Compliance Auditor and Customer (from outside the organization), each having a unique combination of access to captured documents, view, transform, export and audit permissions, as summarized in Table IV. The permission model is

structured around roles not around the individual, as is the case with role-based models for access-control design, which makes it easier to handle the model as the organisation's customer base and staff evolve over time.

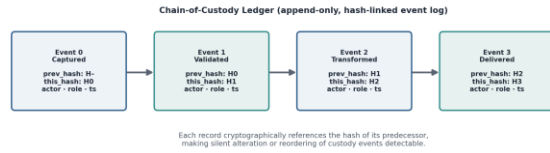


Fig. 3. Hash-linked chain-of-custody ledger. Each record's prev_hash field references the this_hash of its predecessor, making silent alteration or reordering detectable.

Freight documents often include personally identifiable information, such as driver names, signatures and delivery-address information, and the retention policy engine is designed to adhere to the principles of data-minimization by limiting the time personally identifiable fields are queryable after the document is added to the ledger, and by supporting redaction workflows that remove sensitive information from the document itself while retaining the hash record that the removal occurred. Retention policies also need to consider cross-border freight, as the retention time and acceptable storage jurisdiction for a shipment may vary from the origin to the destination, which means that the retention policy engine must be able to trigger jurisdiction-aware rules, instead of a single global retention period [15].

TABLE IV ROLE-BASED ACCESS CONTROL MATRIX

Role	Capture	View	Transform	Export	Audit
Driver	Yes	Own trips	No	No	No
Dispatcher	No	Assigned trips	No	Limited	No
Billing clerk	No	Delivery docs	No	Yes	No
Compliance auditor	No	All (read-only)	No	Yes	Yes
External customer	No	Own shipments	No	Own shipments	No

V. EVALUATION METHODOLOGY

This section provides metrics and method for evaluating the architecture; as the architecture is proposed as a design and not an already completed deployment, it does not report on results of

deployment. Five measures are suggested. The completeness of the chain-of-custody is the percentage of all expected events that are recorded and hash-verified in a document lifecycle, C:

$$C = N_{\text{verified}} / N_{\text{expected}} \quad (1)$$

N_{verified} is the number of custody events for which the hash chain is verified to be correct, and N_{expected} is the number of events for that document type that the pipeline definition specifies. The time period from document.captured event to document.available event is called delivery latency, L. Duplicate-suppression rate, D, is the ratio of redundant

capture or transformation events properly identified and removed from the object store due to idempotency-key matching before storing them in the object store. The effectiveness of retrieval is measured through the use of standard precision and recall measures for a set of search queries made against the index:

$$P = TP_r / (TP_r + FP_r) \quad (2)$$

$$R = TP_r / (TP_r + FN_r) \quad (3)$$

where TP_r , FP_r , and FN_r denote, respectively, correctly retrieved, incorrectly retrieved, and missed documents relative to a query's intended result set. Recovery time, T_r , measures the interval between a processing-stage failure and successful reprocessing through the pipeline's retry and dead-letter mechanisms.

Table V lists proposed design targets for each metric, expressed as objectives the architecture is intended to satisfy rather than measurements obtained from a running deployment [16].

Evaluation would be done by instrumenting each service in the pipeline to log timing and result information with each custody event that passes

through, replaying representative workloads of capture, transform and delivery, and calculating the above metrics from the event log. Failure-injection testing – purposely interrupting the transformation or delivery layer during processing – would also test the

recovery-time metric and ensure that partially processed documents do not become visible as available before the transformation has finished or that documents don't become lost if a downstream service goes temporarily down [17].

TABLE V PROPOSED EVALUATION METRICS AND DESIGN TARGETS

Metric	Formula	Design Target
Custody completeness (C)	Eq. (1)	Approaching 100% verified events
Delivery latency (L)	$t(\text{available}) - t(\text{captured})$	Low, bounded by SLA
Duplicate suppression (D)	Suppressed / Total duplicates	High, near-complete suppression
Retrieval precision / recall	Eq. (2)–(3)	Balanced, both maximized
Recovery time (T_r)	$t(\text{reprocessed}) - t(\text{failure})$	Minimized via automated retry

VI. COMPARATIVE ANALYSIS AND DISCUSSION

Table VI and Fig. 4 depict the proposed architecture against three other architectures on six qualitative capability dimensions: (1) tamper evidence, (2) delivery reliability, (3) format interoperability, (4) access granularity, (5) retrieval speed and (6) deployment simplicity. Paper and e-mail workflows are still the easiest to implement, but provide virtually no tamper proofing, detailed access control, or quick retrieval, because finding a particular document often necessitates a manual search across unindexed archives [18]. While Pure EDI integration enhances format interoperability and delivery reliability between existing trading partners, it still does not include a built-in method for guaranteeing that the received document is identical to the one

initially captured, and its schema inflexibility makes integration costs soar when a new trading partner or document type is added. The pure blockchain-only approach is maximally tampering proof, but has the downside of retrieval speed, deployment ease and, in the case of frequent capture events like in freight operations, transaction throughput, as consensus finality and on-chain storage costs grow linearly with capture count. The proposed architecture is designed to offer a significant improvement in retrieval speed, access granularity and deployment complexity compared to a consensus-based ledger, while maintaining most of the tamper-evidence properties that hash chaining has to offer [18].

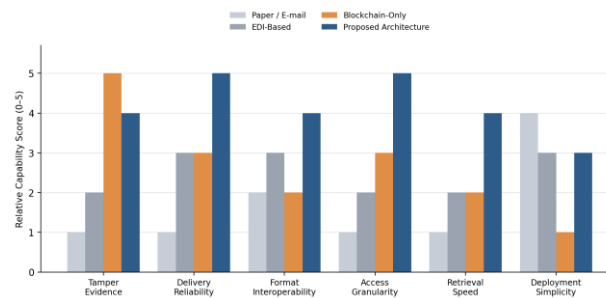


Fig. 4. Qualitative capability comparison across paper/e-mail, EDI-based, blockchain-only, and the proposed architecture, scored on a relative 0–5 design-assessment scale.

There are a number of operational issues to consider. A concern with architecture that must interoperate with legacy EDI feeds that predate the proposed metadata schema is schema drift across trading partners, in which a mapping layer must be provided to map between the two schemas. Key management for the hash-and-sign service is a single point of operational risk: if the signing key is compromised, then a valid-looking custody event can

easily be created by an attacker, highlighting the need for hardware-backed key storage and key rotation policies. When network connectivity is intermittent, the capture layer must be able to store captures in its local buffer until network connectivity is reestablished; otherwise, the ordering guarantees that are required for the hash chain will be lost. Each of these architectural components is traced to its conceptual roots, and the proposed design is placed in

the context of the literature on distributed ledgers, electronic data interchange, IoT supply chains, event-driven microservices, and access-control governance [19][20].

Lastly, in the context of cargo claims and insurance coverage, chain-of-custody documentation that is

provided in the manner proposed here has evidentiary value, and therefore an increased duty to ensure that the content of the ledgers themselves, which could be used to selectively delete content that might be adverse, are themselves documented, logged and audited [21].

TABLE VI FAILURE MODES AND RECOVERY MECHANISMS

Failure Mode	Detection	Recovery Mechanism
Capture app loses connectivity	Local queue age threshold	Local event queue with deferred publish on reconnect
Transformation worker crash	Missing transform.requested ack	Retry with idempotency key; dead-letter after N attempts
Duplicate capture submission	Repeated idempotency key	Duplicate discarded before object store write
Notification delivery failure	Missing delivery receipt	Exponential-backoff retry; escalation alert
Hash-chain break detected	Chain-verification pass	Event flagged; investigation triggered; chain not auto-repaired

VII. CONCLUSION AND FUTURE WORK

This paper has proposed a digital chain-of-custody architecture for freight documents that captures, transforms, delivers, and retrieves the document as independent, auditable, hash-linked events, using ideas derived from distributed-ledger custody research, electronic data interchange, event-driven microservice messaging, and role-based access governance [22][23]. Its architecture aims to eliminate the cost, latency and governance burden of public blockchain consensus while creating a digital document pipeline that is auditable and can replace paper or e-mail-based document flow, addressing the challenge of replacing current inefficiencies. The metrics that were proposed for evaluation serve as a basis for assessing an implementation against the design objectives set forth in Section V, with each measurement being related to a specific objective.

The present design is based on one organization or small trading partner trust boundary. Implementing a verifiable custody hand-off with multiple distrustful, but interdependent, carriers without bringing the full consensus overhead is cited as a major direction for future research, along with closer integration with IoT telematics for event trigger geofencing, and the creation of formally verifiable retention and redaction policies that can prove regulatory compliance without revealing document contents.

REFERENCES

- [1] L. Ahmad, S. Khanji, F. Iqbal, and F. Kamoun, "Blockchain-based chain of custody: Towards real-time tamper-proof evidence management," in Proc. 15th Int. Conf. Availability, Reliability and Security, 2020, Art. no. 48, pp. 1–8.
- [2] H. Amine and A. Mesnaoui, "Electronic Data Interchange (EDI) in the supply chain: Impact on customer satisfaction," *IJRDO – J. Bus. Manage.*, vol. 4, no. 6, pp. 15–22, 2018.
- [3] V. Astarita, V. P. Giofrè, G. Mirabelli, and V. Solina, "A review of blockchain-based systems in transportation," *Information*, vol. 11, no. 1, Art. no. 21, 2020.
- [4] V. Babich and G. Hilary, "Distributed ledgers and operations: What operations management researchers should know about blockchain technology," *Manuf. Serv. Oper. Manage.*, vol. 22, no. 2, pp. 223–240, 2020.
- [5] H. S. Birkel and E. Hartmann, "Internet of Things – the future of managing supply chain risks," *Supply Chain Manage., Int. J.*, vol. 25, no. 3, pp. 229–245, 2020.
- [6] S. E. Chang and Y. Chen, "When blockchain meets supply chain: A systematic literature review on current development and potential applications," *IEEE Access*, vol. 8, pp. 62478–62494, 2020.

- [7] S. E. Chang, Y. C. Chen, and M. F. Lu, "Supply chain re-engineering using blockchain technology: A case of smart contract-based tracking process," *Technol. Forecast. Soc. Change*, vol. 144, pp. 1–11, 2019.
- [8] S. E. Chang, H. L. Luo, and Y. Chen, "Blockchain-enabled trade finance innovation: A potential paradigm shift on using letter of credit," *Sustainability*, vol. 12, no. 1, Art. no. 188, 2020.
- [9] A. Chavan, "Exploring event-driven architecture in microservices: Patterns, pitfalls and best practices," *Int. J. Sci. Res. Archive*, vol. 4, no. 1, pp. 229–249, 2021.
- [10] R. Cole, M. Stevenson, and J. Aitken, "Blockchain technology: Implications for operations and supply chain management," *Supply Chain Manage., Int. J.*, vol. 24, no. 4, pp. 469–483, 2019.
- [11] D. Coquil and H. Kosch, "On reliability in publish/subscribe systems: A survey," *Int. J. Parallel Emergent Distrib. Syst.*, vol. 27, no. 5, pp. 369–386, 2012.
- [12] G. M. Hastig and M. S. Sodhi, "Blockchain for supply chain traceability: Business requirements and critical success factors," *Prod. Oper. Manage.*, vol. 29, no. 4, pp. 935–954, 2020.
- [13] N. Kshetri, "Blockchain's roles in meeting key supply chain management objectives," *Int. J. Inf. Manage.*, vol. 39, pp. 80–89, 2018.
- [14] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, vol. 107, pp. 841–853, 2020.
- [15] A. H. Lone and R. N. Mir, "Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer," *Digit. Investig.*, vol. 28, pp. 44–55, 2019.
- [16] B. Mitra, S. Sural, J. Vaidya, and V. Atluri, "A survey of role mining," *ACM Comput. Surv.*, vol. 48, no. 4, Art. no. 50, 2016.
- [17] S. Narayanan, A. S. Marucheck, and R. B. Handfield, "Electronic data interchange: Research review and future directions," *Decis. Sci.*, vol. 40, no. 1, pp. 121–163, 2009.
- [18] M. Pournader, Y. Shi, S. Seuring, and S. C. L. Koh, "Blockchain applications in supply chains, transport and logistics: A systematic review of the literature," *Int. J. Prod. Res.*, vol. 58, no. 7, pp. 2063–2081, 2020.
- [19] D. Roeck, H. Sternberg, and E. Hofmann, "Distributed ledger technology in supply chains: A transaction cost perspective," *Int. J. Prod. Res.*, vol. 57, no. 7, pp. 2117–2135, 2019.
- [20] S. Villamil, C. Hernández, and G. Tarazona, "An overview of the internet of things," *TELKOMNIKA*, vol. 18, no. 5, pp. 2320–2327, 2020.
- [21] Y. Wang, J. H. Han, and P. Beynon-Davies, "Understanding blockchain technology for future supply chains: A systematic literature review and research agenda," *Supply Chain Manage., Int. J.*, vol. 24, no. 1, pp. 62–84, 2019.
- [22] M. Zhao, "Blockchain technology improves supply chain: A literature review," *Acad. J. Comput. Inf. Sci.*, vol. 3, no. 4, pp. 73–85, 2020.
- [23] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Serv.*, vol. 14, no. 4, pp. 352–375, 2018.