Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/NA10.20250371
Published: June 17, 2025

Blockchain-Enabled Information as a Service (IaaS) for Configuration Tracking and Control of Smart Manufacturing Systems

Mulang Song and Jianxin (Roger) Jiao
Georgia Institute of Technology
Atlanta, USA

Abstract

Configuration data in smart manufacturing systems must be propagated and verified almost as quickly as mechanical re-tooling occurs, yet conventional MES platforms offer neither tamper-evident provenance nor cross-vendor trust at such speeds. This paper proposes a blockchain-enabled Information-as-a-Service (IaaS) framework that records cryptographic fingerprints of every PLC program, robot trajectory, and Genetic Bill of Material and Operations (GBOMO) -based product-variant descriptor on a permissioned ledger, while retaining the associated payloads in the InterPlanetary File System (IPFS) for scalable storage. A domain-specific Proof-of-Configuration (PoC) consensus replaces compute-intensive mining with two lightweight risk metrics—structural entropy and update frequency. Complementary deployment and rollback protocols ensure that validated artefacts are atomically applied across all stations or automatically quarantined. The resulting architecture furnishes a practical basis for secure, agile configuration governance, supporting both real-time digital-twin synchronization and the emergence of high-skill roles in configuration assurance and analytics.

Keywords
Blockchain, Reconfigurable Manufacturing Systems

1. Introduction
Reconfiguration in smart manufacturing systems are specifically designed to cope with highly volatile market demand and shortened product life cycles (Koren et al.2016). They enable rapid capacity and functionality adjustments, giving manufacturers exactly the production capabilities needed when required. RMSs have thus been recognized as key to competitiveness under uncertain, fast-changing conditions (Mehrabi et al. 2000).Conventional manufacturing execution systems (MES) struggle to synchronize the massive, dynamic data generated by RMS environments in real time. Legacy MES architectures tend to have siloed data and limited connectivity, relying on manual interventions that impede adaptability in fast-changing production settings (Parapalli 2025). As a result, traditional MES cannot easily scale or respond at the speed required for highly reconfigurable and data-intensive systems (Shojaeinasab et al. 2022).

1.1 Information-as-a-Service
In manufacturing, IaaS can refer to Information-as-a-Service, a data-centric architecture where entities like manufacturing data or programs are exposed as addressable services accessible on demand. Recent service-oriented manufacturing frameworks explicitly include Information-as-a-Service as a core concept alongside other "X-as-a-Service" components (Jiao et al. 2024). A major barrier to adopting information-sharing services in heterogeneous manufacturing environments is the lack of trust among independent modules and stakeholders. Without a tamper-evident log of transactions, participants cannot be sure data or records haven't been altered, and traditional databases provide limited transparency. Indeed, researchers note that the "lack of trust and security of cloud manufacturing

platforms are bottleneck problems" for adoption. An information service spanning multiple equipment or companies requires a secure, tamper-proof record of data exchanges to establish trust (Zhang et al. 2021).

1.2 Decentralized Data Management and Information Sharing

Blockchain technologies can provide an immutable ledger to record history of manufacturing data or part genealogy, ensuring any update is traceable and cannot be retroactively altered (N. Sangeeta & Nam 2023). Meanwhile, the InterPlanetary File System (IPFS) offers decentralized, peer-to-peer file storage, which is well-suited for handling large manufacturing files (CAD models, process logs, etc.) outside the blockchain while still linking to the blockchain via content hashes. Combining blockchain with IPFS yields a distributed data management approach: the blockchain stores tamper-proof hash pointers and timestamps, while the actual data resides in IPFS's decentralized storage network. This hybrid ensures both immutable provenance on-chain and scalable decentralized file storage off-chain for manufacturing information.

Centralized Industrial Internet of Things (IIoT) architectures struggle to provide secure, tamper-evident, and decentralized access to reconfiguration data, prompting proposals that couple permissioned blockchain with off-chain storage for "a secure, traceable and decentralized" manufacturing information layer (Zhang et al. 2020). Digital-twin surveys likewise warn that sharing configuration updates across interconnected manufacturing lines "raises critical concerns of data integrity and cybersecurity," demanding immutable provenance and controlled disclosure to maintain trust in reconfiguration decisions.

1.3 Blockchain-enabled Information as a Service (IaaS)

Building on the core blockchain + IPFS service stack, the paper presents architecture for the intra-factory reality of a reconfigurable manufacturing environment. The paper also proposed a domain-specific validation layer, Proof of Configuration (PoC), that anchors each block to two measurable properties of the update stream: (i) data-complexity entropy, capturing how many distinct configuration parameters and program entities change in a single event, and (ii) update frequency–volume, reflecting the cadence and byte-size of artefact uploads generated as modules switch among product variants. These metrics are computed on-chain as lightweight cryptographic digests and embedded in the block header, allowing every module or edge gateway to verify that each commit is both legitimate (complexity lies within the authorized envelope for the module type) and timely (frequency aligns with the declared reconfiguration schedule). By replacing generic Proof-of-Work or Proof-of-Authority with PoC, the platform eliminates superfluous hashing, keeps consensus latencies in the sub-second range required on the shop floor, and provides auditors with a direct, tamper-evident trace of how aggressively each station has been reconfigured over time.

The rest of the paper is organized as follows. Section 2 reviews prior work on smart manufacturing and industrial Blockchain technologies application. Section 3 presents the revised three-layer IaaS architecture—information-sharing, virtual, and infrastructure spaces—focusing on how configuration artefacts are chunked, hashed, and referenced within IPFS while their provenance is sealed on-chain. Section 4 formalizes the information requirements for product-variety and configuration management and details the PoC mechanism, including entropy and frequency digests and their smart-contract enforcement logic. Section 5 presents a simple application of connector assembly shop floor, and Section 6 concludes with practical deployment guidelines and future research directions.

2. Literature Review

2.1 Reconfiguration in Smart Manufacturing Systems

The migration from centrally controlled production lines to highly connected, cyber-physical shopfloors is a defining feature of Industry 4.0. Recent reviews emphasize that reconfigurable manufacturing systems (RMS) have become the architecture of choice for coping with volatile demand and proliferating product variants because they allow rapid structural and functional changes at module level (Müller et al. 2023). Each physical reconfiguration spawns a surge of digital entities—parameter sets, controller programs, variant-specific bills-of-materials and sensor histories—that must remain version-consistent throughout the line. A 2025 systematic survey of industrial data-management practices notes that conventional MES stacks still rely on siloed databases and manual synchronization, leaving factories with "no coherent methodology" for governing this data deluge at line speed (Freitas et al. 2025).

The resulting transparency gap is now measurable where Parhi et al. (2025) propose a quantitative traceability-to-transparency metric and show that smart-factory deployments only achieve the inflexion point when configuration data are streamed to all stakeholders in real time. Digital-twin studies reach a similar conclusion for complex assembly:

Cheng et al. demonstrate that multi-level data models and OPC UA information schemes are indispensable to keep virtual and physical configurations aligned during rapid variant changes (Cheng et al. 2025). Collectively, these findings suggest that an Information-as-a-Service (IaaS) layer—where configuration, program and manufacturing data are exposed as addressable, on-demand services—is now a prerequisite for scalable RMS operation.

2.2 Secure Information-as-a-Service via blockchain and IPFS

While service-oriented middleware simplifies discovery of reconfiguration data, it does not by itself guarantee tamper-evidence or cross-vendor trust. Permissioned blockchain platforms meet these requirements by time-stamping every configuration commit, but storing full artefacts on-chain is impractical. Reviews of blockchain/Digital-Twin integration therefore advocate a hybrid design in which on-chain smart-contract events anchor hashes, while large files reside in decentralized content-addressed stores such as IPFS; this combination elevates data integrity, traceability and availability without breaching millisecond latency budgets (Roumelitotis et al. 2024). IPFS-linked smart contracts have already been applied to complex-product supply chains, where they encrypt and index engineering files through information-storage contracts that enforce role-based access (Guo et al. 2025). In parallel, Song et al. (2025) demonstrated blockchain-IPFS IaaS viability for crowdsourced manufacturing, implementing a web-based interface with smart contracts and IPFS algorithms to securely manage product fulfillment statuses across distributed stakeholders at low trust costs.

However, the literature remains sparse on applying the same secure-IaaS pattern inside a single factory's RMS, where reconfiguration events occur every shift and controller programs must be refreshed in near real time. The present study addresses this gap by positioning blockchain-anchored IaaS as the backbone for product-variety and configuration management in an RMS environment, thereby extending earlier supply-chain-centered work to the latency-critical, intra-factory context.

3. System architecture of blockchain-enabled IaaS fulfillment system

Reconfigurable manufacturing environments depend on an uninterrupted flow of machine-configuration files, digital-twin models, process parameters, and sensor summaries. Conventional MES databases cannot guarantee immutability or cross-vendor trust, while monolithic cloud repositories generate unacceptable latency during rapid changeovers. To reconcile agility with security we recast the three-space architecture previously used for crowdsourced production into an Information-as-a-Service (IaaS) backbone that is purpose-built for in-factory configuration governance. The stack integrates a permissioned blockchain, smart-contract automation, off-chain IPFS storage, and cyber-physical production assets, yielding a digitally porous yet auditable shop floor.

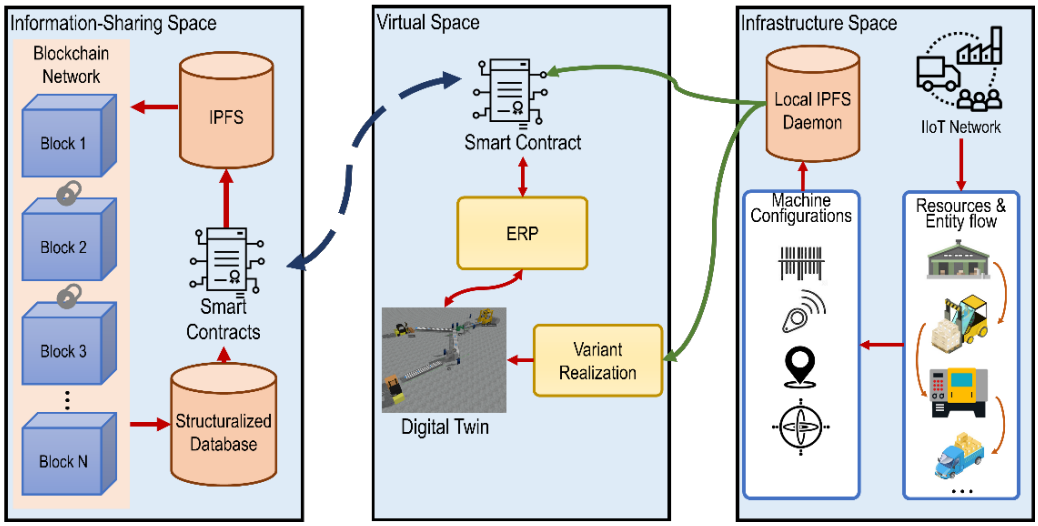

Figure 1. System architecture of the proposed IaaS fulfillment system

As shown in Figure 1, there are three layers in the proposed system architecture. At the leftmost tier a consortium blockchain hosts the canonical record of every configuration artefact and every product-variety chromosome

instantiated on the line. Two smart-contract families operate here which are Configuration-registry contracts and Access-control contracts. Configuration-registry contracts timestamp each ladder-logic file, robot program, or digital-twin snapshot via a 32-byte Keccak hash and store a pointer to the corresponding IPFS content identifier (CID). Access-control contracts embed role-based rules so that only authorized identities—e.g., controls engineers, QA engineers, or digital-twin services—can resolve a CID or widen a station's risk envelope. Because block payloads are limited to a few hundred bytes, the ledger remains compact and validates in sub-second time, yet its cryptographic consensus ensures traceability, immutability, and non-repudiation for every configuration event.

In virtual space, a digital-twin pool mirrors each reconfigurable cell. Twins subscribe to on-chain events, pull the referenced artefacts from IPFS, and hot-load them into physics-based or discrete-event models. This mechanism produces three immediate benefits. Change-impact simulation means that the proposed configuration edits are replayed in silico before PoC validation, filtering unsafe logic at the source. Variant realization is when a product-variety chromosome is committed, the twins prune the master GBOMO graph, verify routing feasibility, and hand back the derived parameter set to the edge agents. Closed-loop analytics aims for high-frequency sensor digests—pinned under a separate IPFS namespace—feed anomaly-detection and remaining-useful-life algorithms, whose advisories in turn trigger smart-contract calls that adjust risk envelopes or schedule maintenance.

The physical layer consists of station gateways (PLC edge boxes, robot IPCs, SCADA nodes) interconnected by an industrial LAN. Each gateway embeds an edge agent, a local IPFS daemon and a hardware-root-of-trust module. An agent that tokenizes local configuration files, computes PoC risk metrics and interacts with IPFS. A local IPFS daemon that caches frequently requested artefacts, delivering sub-millisecond downloads even during WAN loss. A hardware-root-of-trust module (TPM or Secure Enclave) that stores the private keys used for engineering and QA dual signatures.

4. Blockchain-Enabled IaaS Fulfilment for Configuration and Product-Variety Data Tracking

4.1 Blockchain Network, Smart Contracts and IPFS

A blockchain serves as a digital ledger recording transactions in sequential blocks. As shown in Figure 3, each block contains a header and body. The header includes: the 256-bit Previous Block Hash (virtual address of the prior block), Timestamp (mining/validation time), Nonce (variable for hash validation), and Merkle Root (aggregated transaction hashes). The body holds transaction data, constrained by block/transaction sizes. Data is encrypted during transmission, tracked via unique TX hashes, with users accessing accounts via private keys. This network forms a persistent, auditable public ledger (Zheng et al. 2017) where blocks link sequentially—except the first genesis block initiating the chain.

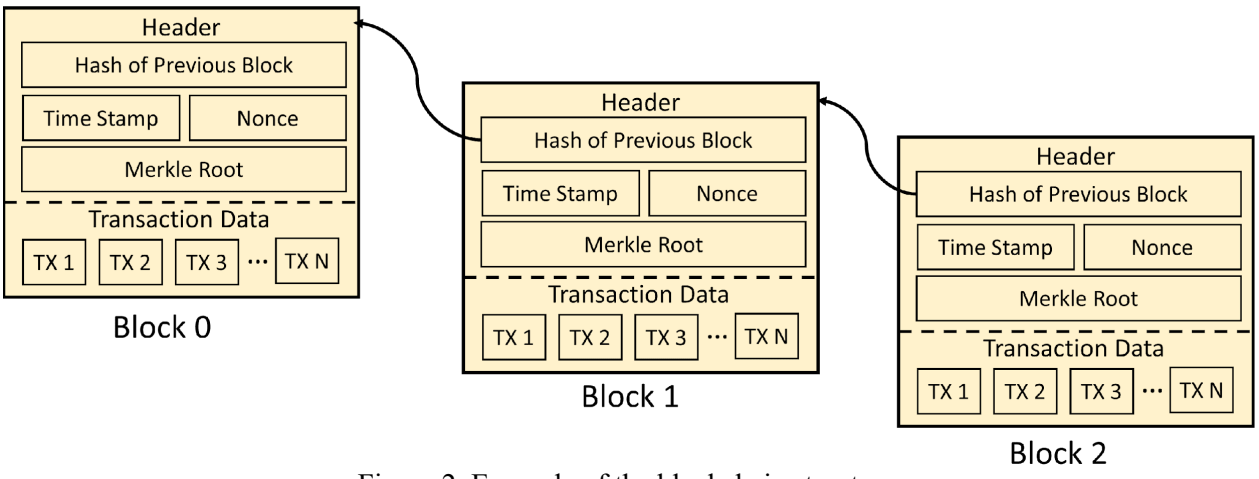

Figure 2. Example of the blockchain structure

Blockchain consensus mechanisms resolve trust issues and validate transactions, ensuring security, efficiency, and convenience. Common algorithms include Proof of Work (PoW), Proof of Stake (PoS), and Proof of Authority (PoA). PoW is trustless in which nodes compete to solve cryptographic puzzles, where the highest computational effort

reduces attack likelihood. However, it consumes significant resources. PoS is also trustless but prioritizes nodes with larger cryptocurrency holdings, as higher stakes disincentivize attacks. To prevent dominance by the wealthiest nodes, it incorporates randomization or alternative criteria (Zheng et al. 2017). PoS is more resource-efficient and secure than PoW. PoA is trust-based, suited for networks with vetted participants. Pre-approved "trusted nodes" validate transactions for untrusted nodes.

Smart contracts, essential to blockchain technology, are self-executing scripts operating under predefined rules. By embedding extensive conditions, they autonomously execute commands to influence the blockchain network. These contracts are now ubiquitous across blockchain platforms, each with distinct implementations. They exhibit three core traits which are (i) Autonomy: Execution occurs automatically once initiated, (ii) Self-sufficiency: Resources are managed within the blockchain without external control, (iii) Decentralization: Deployment spans all network nodes, aligning with blockchain's foundational principles. Additionally, their programmability enables complex service delivery and encapsulates node behaviors within the network.

The Interplanetary File System (IPFS) provides decentralized, high-throughput storage for large files, contrasting with blockchain networks. Its core features are decentralization (enabling file access without centralized control) and content addressing (where files are identified by their content). Each file uploaded to IPFS generates a unique cryptographic hash called a Content Identifier (CID), which serves as its address. Users access files by appending the CID to /ipfs/ in a browser. Any modification to the file creates a new CID, ensuring both versions remain stored and traceable within the network.

4.2 Proof-of-Configuration Consensus for Machine-level Configuration Data

This section formalizes the proposed PoC mechanism and demonstrates how it secures machine-level configuration data (PLC, robot, SCADA, MES) while retaining sub-second finality. This section defines the risk metrics used for validation, then presents the claim-generation algorithm executed at each asset and the verification algorithm embedded in the consortium smart contract. Finally, it outlines how PoC integrates with block construction and prove its key safety and liveness properties.

The proposed PoC secures every machine-level configuration update—whether it originates from a PLC, a robot controller, a SCADA node or an MES recipe server—while preserving the sub-second latency needed on the shop-floor network. We introduce the key symbols once and then use them throughout the section. A *station* $\boldsymbol{s}$ is an automation asset that issues a change set $\boldsymbol{\Delta_s}$. It is an ordered collection of configuration files $\boldsymbol{f}$; for example, a PLC project file, a RAPID program for a robot, or an XML recipe for the MES. Each file $\boldsymbol{f}$ is passed to a tokenizer whose output is the multiset $\boldsymbol{tok(f)} = \{\boldsymbol{t_1^{c_1}}, \boldsymbol{t_3^{c_3}}, \ldots\}$ Every distinct lexical element extracted from the file—such as a tag name, a motion opcode, or an XML node is called a token and is represented by the lower-case symbol $\boldsymbol{t}$. The superscript $\boldsymbol{c_i}$ records how many times that token occurs in the file. Taking the union over all files yields the set of *unique* tokens $\boldsymbol{U} = \bigcup_{\{f \in \Delta_s\}} \boldsymbol{tok(f)}$.

For each $t \in \boldsymbol{U}$, the algorithm cunts its occurrences in the entire change-set and forms the empirical probability $\boldsymbol{p_t} = \frac{\boldsymbol{count(t)}}{\sum_{u \in U} \boldsymbol{count(u)}}$. The Shannon diversity of the change-set is measured in bits which is expressed as $\boldsymbol{H_s} = -\sum_{t \in U} \boldsymbol{p_t p_t}$. The entropy score is calculated by multiplying this diversity by the breadth of the change—the cardinality $|\boldsymbol{U}|$ where $E = |\boldsymbol{U}| H_s$.

Two additional risk metrics are recorded. The volume of update is $V = \sum_{f \in \Delta_s} \boldsymbol{bytes(f)}$ and the interval since the last accepted update is $\boldsymbol{\Delta T} = \boldsymbol{T_{now}} - \boldsymbol{T_s}$. where $\boldsymbol{T_{now}}$ is the wall-clock timestamp at which the claim is being constructed and $\boldsymbol{T_s}$ is the block-timestamp of the previous successful claim from the same station. This explicit sub-script distinguishes the time variable from the token iterator used above.

Every file f is stored off-chain in IPFS, producing a content identifier $CID(f)$ that is cryptographically bound to the file's bytes. The station concatenates its identifier, the three risk metrics E, V and $\boldsymbol{\Delta T}$, ordered list of CIDs, and the current timestamp $\boldsymbol{T_{now}}$. The byte string $\langle stationID || E || V || \Delta T || CID() || \boldsymbol{T_{now}} \rangle$ is passed through the Keccak-256 hash function, written$H(\cdot)$.

Based on the byte string, the resulting tuple $(stationID, E, V, \Delta T, CID(), T_{now}, \sigma_{eng}, \sigma_{qa}, H)$ forms the PoC claim that the edge agent submits to the consortium ledger. Two digital signatures are mandatory which σ_{eng} is applied with the private key of the manufacturing or controls engineer who prepared the change, and σ_{qa} is applied with the independent private key of a quality-assurance (or safety) engineer.

In practice, engineers approve the digest through a FIDO-2 hardware token or smart-card integrated into the programming IDE. With every symbol, T_{now} for the real-time stamp, E for entropy, V for volume, ΔT for interval and H for the cryptographic digest—precisely defined and with dual human approval anchored on-chain, the PoC framework preserves millisecond-level agility while delivering regulator-grade assurances essential for life-critical industrial automation.

4.3 Proof-of-Configuration Consensus for Traceability Metric

The Generic Bill-of-Materials-and-Operations (GBOMO) merges material structure (BOM) and routing (BOO) into a single over-complete graph $\boldsymbol{G}$(Jiao et al., 2000). Each node—generic product, generic goes-into link, generic operation, or generic planning rule—shares an identical parameter vector $\boldsymbol{p} = \{\boldsymbol{p_0}, \boldsymbol{p_1}, \dots, \boldsymbol{p_{n-1}}\}$ A concrete product-and-process variant is therefore selected by a chromosome bitmap $\boldsymbol{g} = (\boldsymbol{g_0}, \boldsymbol{g_1}, \dots, \boldsymbol{g_{n-1}}) \in \{\boldsymbol{0}, \boldsymbol{1}\}^{\boldsymbol{n}}$, where bit $\boldsymbol{g_i} = \boldsymbol{1}$ activates the option associated with parameter $\boldsymbol{p_i}$. The master graph $\boldsymbol{G}$ of a product family is serialized, pinned in the factory IPFS swarm and referenced on chain by its immutable content identifier.

The Generic Bill-of-Materials-and-Operations (GBOMO) merges material structure (BOM) and routing (BOO) into a single over-complete graph $\boldsymbol{G}$(Jiao et al. 2000). The master Generic Bill-of-Materials-and-Operations (GBOMO) is stored off-chain under the immutable content hash $\boldsymbol{CID_G} = \boldsymbol{ipfs.add(G)}$, where the graph $\boldsymbol{G}$ contains four element types—generic products, generic goes-into links, generic operations and generic planning rules—each expressed with the same vector of variety parameters $\boldsymbol{p} = \{\boldsymbol{p_0}, \boldsymbol{p_1}, \dots, \boldsymbol{p_{n-1}}\}$. A concrete order for the production family (abbreviated PF) specifies a chromosome bitmap $\boldsymbol{g} = (\boldsymbol{g_0}, \boldsymbol{g_1}, \dots, \boldsymbol{g_{n-1}}) \in \{\boldsymbol{0}, \boldsymbol{1}\}^{\boldsymbol{n}}$where bit $\boldsymbol{g_i} = \boldsymbol{1}$ selects the "on" option of parameter $\boldsymbol{p_i}$. The ledger call *instantiateVariant*(PF, CIDGBOMO, g) checks the bitmap against all planning rules that reside in the GBOMO and logs a *VariantCommit* event. Every controller on the line subscribes to that event and locally prunes the graph with $\boldsymbol{g}$, producing the configuration set $\boldsymbol{\Delta_{PF}} = \{\boldsymbol{f_1}, \boldsymbol{f_2}, \dots, \boldsymbol{f_m}\}$, which contains the exact PLC project, robot path and MES recipe needed for the requested variant.

Each file $\boldsymbol{f} \in \boldsymbol{\Delta_{PF}}$ is tokenized. A token—denoted by the lower-case symbol$\boldsymbol{t}$—is one distinct lexical element such as a PLC tag or a RAPID opcode, and its multiplicity in the file is written as a superscript$\boldsymbol{c}$. Gathering all unique tokens give $\boldsymbol{U} = \bigcup_{\{f \in \Delta_{PF}\}} \boldsymbol{tok(f)}$. With $\boldsymbol{p_t}$ defined as the empirical frequency of token in $\boldsymbol{\Delta_{PF}}$, the structural entropy reflects functional complexity $\boldsymbol{E_{struct}} = |\boldsymbol{U}| - \sum_{\boldsymbol{t} \in \boldsymbol{U}} \boldsymbol{p_t} \boldsymbol{p_t}$.

The chromosome just received is compared with the previously committed bitmap $\boldsymbol{g^{old}}$. Their bitwise XOR highlights all flipped genes and the Hamming distance $\boldsymbol{k} = ||\boldsymbol{g^{old}} \oplus \boldsymbol{g^{new}}||$ counts them. Because each gene already corresponds to a validated option in GBOMO, the variant-flip entropy is intentionally mild:

$$E_{Var} = \{0, k = 0\ k\ , k \geq 1$$

A policy constant discounts this term and the complete risk score becomes $\boldsymbol{E_{tot}} = \boldsymbol{E_{struct}} + \boldsymbol{w_{var}} \boldsymbol{E_{var}}$. Two further scalars protect network traffic and cadence. The payload volume $\boldsymbol{V} = \sum_{\boldsymbol{f} \in \boldsymbol{\Delta_s}} \boldsymbol{bytes(f)}$ limits the total bytes uploaded in one claim, while the interval $\boldsymbol{\Delta T} = \boldsymbol{T_{now}} - \boldsymbol{T_{PF}}$ with $\boldsymbol{T_{now}}$ the current wall-clock time and $\boldsymbol{T_{PF}}$ the block-time of the last accepted claim for this production family, enforces a cooling-off period between commits.

If $(\boldsymbol{E_{tot}}, \boldsymbol{V}, \boldsymbol{\Delta T})$ lies inside the station envelope $\{\boldsymbol{E_{min}}, \boldsymbol{E_{max}}, \boldsymbol{V_{max}}, \boldsymbol{\Delta T_{min}}\}$ the agent hashes every file to IPFS, obtaining the list $\boldsymbol{CID}()$, and collapses it to a Merkle root. The byte string $\langle \boldsymbol{PF} || \boldsymbol{g^{new}} || \boldsymbol{E_{tot}} || \boldsymbol{V} || \boldsymbol{\Delta T} || \boldsymbol{root} || \boldsymbol{T_{now}} \rangle$ is fed to Keccak-256, giving the digest$\boldsymbol{H}$. Two credentials then sign this digest: $\boldsymbol{\sigma_{eng}}$ by the manufacturing engineer and

$\boldsymbol{\sigma_{qa}}$ by the quality-assurance engineer, thereby enforcing the four-eyes rule. The tuple $\left(\boldsymbol{PF},,\boldsymbol{E_{tot}},\boldsymbol{V},\boldsymbol{g^{new}},\boldsymbol{root},\boldsymbol{T_{now}},\boldsymbol{\sigma_{eng}},\boldsymbol{\sigma_{qa}},\boldsymbol{H}\right)$ constitutes the PoC claim.

On chain, the smart contract re-computes$\boldsymbol{H}$, authenticates both signatures, re-applies the envelope limits and, if all tests succeed, appends $\langle \boldsymbol{PF}||\boldsymbol{g^{new}}||\boldsymbol{E_{tot}}||\boldsymbol{V}||\boldsymbol{\Delta T}||\boldsymbol{root}||\boldsymbol{T_{now}}\rangle$ to the ledger. Consortium validators replay the same deterministic logic; a block is final only when every included claim passes, so the line achieves without proof-of-work. A routine order switch—say from bitmap 10011000 to 10011100—flips exactly one gene ($\boldsymbol{k=1\rightarrow E_{var}=0}$) and touches just two small files, leaving $\boldsymbol{E_{struct}\cong 7}$. With $\boldsymbol{E_{max}=25}$ the claim is accepted automatically. By contrast, introducing a new gene for RFID tagging adds 133 unique tokens to the PLC and robot programs; $\boldsymbol{E_{struct}}$ jumps to ≈ 350, exceeds $\boldsymbol{E_{max}}$ and the agent blocks the update until safety and QA widen the envelope and countersign.

Thus, every symbol—token$\boldsymbol{t}$, set $\boldsymbol{U}$, entropies $\boldsymbol{E_{struct}}$, $\boldsymbol{E_{var}}$, $\boldsymbol{E_{tot}}$ Hamming distance $\boldsymbol{k}$, volume$\boldsymbol{V}$, interval $\boldsymbol{\Delta T}$ and digest$\boldsymbol{H}$—has a single, explicit meaning and a measurable counterpart in the plant. Lightweight variant flips flow, yet any structural enlargement of the variety space is forced through cryptographically recorded human review, giving the factory both mass-customization agility and regulator-grade safety assurance—conditions that ultimately safeguard livelihoods and lives on the shop floor.

5. PoC Enabled Entity Configuration Control

Once a configuration file or product-variant chromosome has passed Proof-of-Configuration (PoC) validation, the line must apply it to the correct physical assets, verify that the change took effect and—if the next order requires a fresh variant—roll back or supersede it without loss of provenance.

Let $S=\{s_1,\dots,2_M\}$ be the set of automation stations on the line and let $\Gamma=\{\pi_1,\dots,\pi_N\}$ be the queue of PoC-approved configuration packages awaiting execution. Each package $\pi_j=\langle CID_j,h_j,g_j,T_j\rangle$ contains the Merkle root CID_j of all files, its digest h_j, the optional chromosome bitmap g_j and the block-timestamp T_j. A deployment map $\beta\colon \Gamma\times S\rightarrow\{0,1\}$ assigns each package to the stations that must consume it. The control problem is to guarantee within a finite horizon H that every s_i with $\beta\left(\pi_j,s_i\right)=1$ downloads the exact byte set described by CID_j, the station's run-time image has h_{ij}^{m} equals h_j, all state variables required interlocking or safety PLCs satisfy the predicate set $C\left(\pi_j\right)$ derived from the digital-twin model, and any deviation raises an irrevocable alarm and initiates an atomic roll-back to the last validated-and-executed packages.For each block the Controller-Dispatcher contract emits the deployment of π_j and β_j once the block reaches finality. The intent carries a hash of the deployment map $h_{\beta j}=H\left(\beta(\pi_j,.)\right)$. Stations monitor the event feed and enter the pending state if they are listed in β.

On receipt of deployment intent station $\boldsymbol{s_i}$ performs the following sequence:

Step 1 – Retrieval. The local IPFS peer resolves $\boldsymbol{CID_j}$. If a block is missing the request is forwarded to at most $\boldsymbol{K}$ remote peers before the attempt is declared failed.

Step 2 – Integrity check. A SHA-256 digest of the concatenated file bytes must equal the on-chain value $\boldsymbol{h_j}$ otherwise the deployment aborts and deployment fault is broadcast.

Step 3 – Run-time swap. Execution halts for one scan, new logic is written to the controller, a checksum of the run-time image $\boldsymbol{h_{ij}^{run}}$ is recomputed and compared with $\boldsymbol{h_j}$.

Step 4 – Commit acknowledgement. If the checksum matches, deployment acknowledgement is emitted, otherwise deployment fault is raised, and the controller rolls back to the previous version stored in its non-volatile shadow memory.

The Dispatcher contract places a time-out $\boldsymbol{H}$ on each deployment. If all stations assigned to $\boldsymbol{\pi_j}$ acknowledge within $\boldsymbol{H}$, the contract finalizes the change with *DeployCommit*$\left(\boldsymbol{\pi_j}\right)$. If at least one deployment fault is observed, or the time-out expires, the contract issues a *DeployRollback*$\left(\boldsymbol{\pi_j}\right)$ event; every station then reloads the last committed package whose digest is cached locally. Because all events are ledger-recorded, a near-miss trace is preserved for root-cause analysis.

The control layer leverages the immutable, provenance-rich record produced by PoC tracking yet enforces a deterministic execution workflow that tolerates packet loss, peer unavailability and model-detected hazards. By keeping all control-flow events on the ledger the system satisfies traceability audits, and by limiting on-chain writes to hashes and signatures it avoids the bandwidth explosion typical of naïve "store-everything-on-chain" designs.

5. Limitations and Future Work

The architecture and Proof-of-Configuration logic described in this paper are specified at design level and validated only through a narrative walk-through; no empirical shop-floor data, latency measurements, or failure-injection results are yet available. Because of that, the quantitative claims on sub-second finality and bandwidth savings remain projections rather than statistically grounded findings. Second, all risk envelopes (entropy bounds, cadence limits, payload caps) are assumed to be fixed constants derived from historical logs; in practice those thresholds drift with product mix, controller firmware and operator work patterns, so manual tuning may become a maintenance burden. Third, the deployment protocol presumes that every station can tolerate a one-scan halt and can roll back atomically; legacy PLCs without dual-bank loaders or robot controllers lacking file-swap primitives would need bespoke wrappers, complicating universal adoption. Fourth, the privacy model is coarse-grained: CIDs are publicly visible to every node in the consortium chain, which may leak the existence of confidential options even if the payload is encrypted in IPFS. Finally, the security analysis concentrates on data-integrity attacks; side-channel or denial-of-service vectors—such as flooding the chain with low-entropy commits to delay critical updates—have not been formally modelled.

The immediate next step is a controlled pilot deployment on an industrial testbed instrumented with packet sniffers and time-synchronized data loggers. That study should capture end-to-end latency, CPU and network overhead, and fault-recovery time under realistic disturbance scenarios (link loss, power cycling, signature mismatch). A second avenue is adaptive envelope learning: by mining successful and rejected commits, a reinforcement-learning agent could propose incremental adjustments to E_{min}, E_{max}, V_{max} and ΔT_{min}, reducing both false positives and human tuning effort. Third, a formal-verification layer. Fourth, the privacy gap may be addressed with selective disclosure techniques such as hash-salt commitments or zero-knowledge proofs that reveal only whether a controller possesses the correct file without exposing the CID itself. Finally, the economic impact on job creation should be quantified through socio-technical field studies, tracking how roles migrate from manual change-log maintenance to higher-skill configuration-assurance and digital-twin engineering positions once the system is in sustained operation.

Collectively, these research tasks will convert the proposed framework from a conceptually sound design into a rigorously benchmarked, industry-ready infrastructure for secure, agile configuration control in smart manufacturing systems.

6. Conclusions

Smart-factory initiatives will not realize their promised flexibility until configuration artefacts—PLC logic, robot trajectories, product-variant descriptors, sensor digests—can move across heterogeneous equipment with the same speed and integrity as physical tooling. This paper positions a permissioned-blockchain and IPFS hybrid as a practical backbone for that task, introducing an Information-as-a-Service architecture in which every file's provenance is sealed on-chain while the bulk data resides in a deduplicated, peer-to-peer store. The domain-specific Proof-of-Configuration consensus transforms abstract notions of "safe change" into two measurable signals—structural entropy and update cadence—plus a dual-signature gate that meets the four-eyes requirement of functional-safety standards. By extending the same logic to GBOMO-driven product variety, the framework reconciles lot-size-one agility with regulator-grade traceability.

The implications reach well beyond a single pilot line. For manufacturers, immutable configuration histories reduce recall exposure; for equipment vendors, content addressing eliminates version-race disputes; for the workforce, new roles emerge in configuration assurance, digital-twin analytics and smart-contract governance—high-skill jobs that anchor advanced production locally. Although empirical validation and adaptive-envelope tuning remain future work, the architecture establishes a clear technical pathway from today's siloed MES islands to a resilient, data-centric manufacturing ecosystem where every reconfiguration is both rapid and correct.

References

Cheng, X., Huang, F., Yang, Q., & Qiu, L. , A digital twin data management and process traceability method for the complex product assembly process. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 47(3),(2025). . https://doi.org/10.1007/s40430-025-05466-4

Freitas, N., Rocha, A. D., & Barata, J. , Data management in industry: concepts, systematic review and future directions. Journal of Intelligent Manufacturing,(2025). . https://doi.org/10.1007/s10845-025-02570-z

Guo, X., Zhang, G., & Zhang, Y. , Blockchain-Based Information Sharing Mechanism for Complex Product Supply Chain. Electronics, 14(9),(2025). . https://doi.org/10.3390/electronics14091780

Jiao, J., Tseng, M. M., Qinhai, N., MA, & Zou, N. Y. , Generic Bill-of-Materials-and-Operations for High-Variety Production Management. Concurrent Engineering, 8(4), 297–321,(2000).

Jiao, R., Gong, X., Song, M., Wang, S., & Helo, P. , An X-as-a-Service (XaaS) Reference Model for Smart Manufacturing Systems. In S. N. Padhi (Ed.), Trends and Applications in Mechanical Engineering, Composite Materials and Smart Manufacturing (pp. 47-68), (2024). . IGI Global. https://doi.org/10.4018/979-8-3693-1966-6.ch003

Koren, Y., Gu, X., & Guo, W. , Reconfigurable manufacturing systems: Principles, design, and future trends. Frontiers of Mechanical Engineering, 13(2), 121-136, (2017). . https://doi.org/10.1007/s11465-018-0483-0

Mehrabi, M. G., Ulsoy, A. G., & Koren, Y. , Reconfigurable manufacturing systems: Key to future manufacturing. Journal of Intelligent Manufacturing, 11(4), 403-419,(2000). . https://doi.org/10.1023/a:1008930403506

Müller, T., Caesar, B., Weiß, M., Ferhat, S., Sahlab, N., Fay, A., Oger, R., Jazdi, N., & Weyrich, M. , Reconfiguration management in manufacturing. A systematic literature review, 71(5), 330-350,(2023). . https://doi.org/doi:10.1515/auto-2022-0139

Parapalli, S. L. , Evolution of MES in Autonomous Factories: From Reactive to Predictive Systems. International journal of data science and machine learning, 05(01), 127-136,(2025). . https://doi.org/10.55640/ijdsml-05-01-15

Parhi, S., Joshi, K., Garza-Reyes, J. A., & Akarte, M. , Evaluating the transparency capability of smart manufacturing systems. Operations Management Research,(2025). . https://doi.org/10.1007/s12063-025-00547-y

Roumeliotis, C., Dasygenis, M., Lazaridis, V., & Dossis, M. ,Blockchain and Digital Twins in Smart Industry 4.0: The Use Case of Supply Chain-A Review of Integration Techniques and Applications. Designs, 8(6),(2024). . https://doi.org/10.3390/designs8060105

Sangeeta, N., & Nam, S. Y. , Blockchain and Interplanetary File System (IPFS)-Based Data Storage System for Vehicular Networks with Keyword Search Capability. Electronics, 12(7),(2023). . https://doi.org/10.3390/electronics12071545

Shojaeinasab, A., Charter, T., Jalayer, M., Khadivi, M., Ogunfowora, O., Raiyani, N., Yaghoubi, M., & Najjaran, H. , Intelligent manufacturing execution systems: A systematic review. Journal of Manufacturing Systems, 62, 503-522,(2022). . https://doi.org/10.1016/j.jmsy.2022.01.004

Song, M., Gong, X., Jiao, R. J., & Moore, R. , A blockchain-enabled information as a service (IaaS) system for crowdsourced manufacturing: A crowdsourcing case study of tank trailer manufacturing. Journal of Industrial Information Integration, 45. https://doi.org/10.1016/j.jii.2025.100844

Zhang, C., Zhou, G., Li, H., & Cao, Y. , Manufacturing Blockchain of Things for the Configuration of a Data- and Knowledge-Driven Digital Twin Manufacturing Cell. IEEE Internet of Things Journal, 7(12), 11884-11894,(2020). . https://doi.org/10.1109/jiot.2020.3005729

Zhang, Y., Zhang, L., Liu, Y., & Luo, X. , Proof of service power: A blockchain consensus for cloud manufacturing. Journal of Manufacturing Systems, 59, 1-11, (2021). . https://doi.org/10.1016/j.jmsy.2021.01.006

Biographies

Mulang Song is a Ph.D. candidate in the School of Mechanical Engineering at the Georgia Institute of Technology (USA). He is graduating in Mechanical Engineering, with research that spans both mechanical and industrial engineering domains. He received both his B.S. and M.S. degrees in Mechanical Engineering with a focus on Computer Science, also from Georgia Tech. He has led and contributed to engineering projects through direct collaboration with industry, applying research in real-world production environments. His research interests include advanced manufacturing systems, intelligent production systems, and industrial AI applications. More info about his research: https://scholar.google.com/citations?user=jLL4nycAAAAJ.

Dr. Jiao is the editor-in-chief of Journal of Engineering Design and an associate professor of mechanical and industrial engineering at Georgia Tech, USA. Prior to joining the School of Mechanical Engineering at Georgia Tech in

December 2008, he was an Assistant Professor and then Associate Professor in the School of Mechanical and Aerospace Engineering at Nanyang Technological University, Singapore. Before his career in Singapore, he was a Visiting Scholar in the Department of Industrial Engineering and Engineering Management at Hong Kong University of Science and Technology from 1998 to 1999. From 1993 to 1994, he was a Lecturer of Industrial Engineering in the School of Management at Tianjin University, China, and from 1988 to 1990, he worked as an Associate Lecturer in the Department of Industrial Design at Tianjin University of Science and Technology, China. More info about his research: https://scholar.google.com/citations?user=9yikEHAAAAAJ&hl=en&oi=ao.