

## Blockchain-Enabled Trust Management for Autonomous Multi-Robot Systems

Padmanabhan V<sup>1\*</sup>, Padmanabhan S<sup>2</sup> and Gurumurthy R<sup>3</sup>

<sup>1</sup>Department of Intelligent Systems, Vaigai Institute of Smart Systems Engineering, Taiwan Email:

[vijay.padmanabhan@vaigai.edu](mailto:vijay.padmanabhan@vaigai.edu)

ORCID: <https://orcid.org/4153-8923-5997-4134>

<sup>2</sup>Department of Computational Intelligence, Hemavathi Center for Intelligent Automation, South Korea Email:

[sneha.padmanabhan@hemavathi.edu](mailto:sneha.padmanabhan@hemavathi.edu)

ORCID: <https://orcid.org/5136-2470-3090-1645>

<sup>3</sup>Department of Intelligent Systems, Kabini Institute of Autonomous Robotics, Malaysia Email:

[ramesh.gurumurthy@kabini.edu](mailto:ramesh.gurumurthy@kabini.edu)

ORCID: <https://orcid.org/3995-2296-7172-598X>

### ABSTRACT

*Autonomous multi-robot and swarm-robotic systems must coordinate action among many physically separated units, and that coordination breaks down if any participant can misreport its state or sensed data without consequence. This article synthesizes the growing literature on using blockchain technology to make multi-robot trust management decentralized and tamper-evident rather than dependent on a single trusted coordinator. We trace the problem from pre-blockchain trust and reputation research in wireless sensor networks and mobile ad hoc networks, through the foundational distributed-systems consensus work that gives blockchain its Byzantine fault tolerance guarantees, to concrete architectures -- physically tested robot swarms, UAV-swarm consensus and authentication schemes, blockchain-based information markets, and blockchain-secured federated learning -- that now target real multi-robot deployment. Particular attention is given to the handful of studies that report precise, verifiable numbers: a blockchain-based consensus scheme that remains largely accurate with over a third of a swarm behaving adversarially, where classical consensus protocols fail outright, and UAV-swarm consensus and authentication schemes that keep the added security overhead compatible with real-time flight control. Read together, the evidence supports blockchain-enabled trust management as a coherent and increasingly well-evidenced architectural pattern for adversarial-robust multi-robot coordination, while leaving deployment at industrial fleet scale, resource cost on the smallest robotic hardware, and the security of the physical sensing layer beneath the ledger as an open research agenda.*

**Keywords:** blockchain; trust management; multi-robot systems; swarm robotics; Byzantine fault tolerance

**Citation:** Padmanabhan V, Padmanabhan S and Gurumurthy R [2022]. Blockchain-Enabled Trust Management for Autonomous Multi-Robot Systems. **DOI:** <https://doi.org/10.5281/zenodo.22843852>

**Copyright:** © 2022 by the authors. Open access under [CC BY 4.0 license](https://creativecommons.org/licenses/by/4.0/).

**Article Information:** Received: 20 January 2022 Accepted: 17 April 2022 Published: 7 July 2022

**Article Class:** Research Article

## 1. Introduction

Multi-robot systems have moved, over roughly the last decade, from a laboratory curiosity into a class of infrastructure: warehouse fleets that reallocate tasks among dozens of mobile units in real time, agricultural drone swarms that share coverage maps mid-flight, and search-and-rescue teams that must keep operating even after individual units are destroyed, jammed, or captured. What all of these deployments share is a structural vulnerability that single-robot systems mostly avoid — every one of them depends on some form of agreement among many autonomous, physically separated decision-makers, and that agreement has to survive the possibility that some of the decision-makers are wrong, broken, or actively adversarial. A robot with a corrupted sensor, a spoofed GPS signal, a compromised firmware update, or simply a hardware fault can report false state to its peers, and unless the rest of the fleet has some principled way of discounting that robot's word, the whole system's coordination logic inherits the corruption. Multi-robot coordination research has, for decades, produced sophisticated planning, task-allocation, and consensus algorithms [1,2], but a great deal of that literature quietly assumes that every robot reporting into the system is telling the truth.

Trust management is the research program that removes that assumption. It asks how a distributed system of autonomous, resource-constrained agents can maintain a shared, reasonably accurate belief about which of its members are reliable, without a single point of failure and without every agent having to independently verify every claim from scratch. The idea did not originate in robotics — trust and reputation systems were developed first for e-commerce platforms and peer-to-peer networks [3], then adapted for wireless sensor networks facing similar problems of unreliable, resource-constrained nodes [4], and mobile ad hoc networks where node mobility and intermittent connectivity make trust bookkeeping especially hard [5]. Multi-robot systems inherit essentially all of these challenges and add a further one: robots do not just report data, they act on the physical world, so a compromised or wrongly-trusted robot can cause harm that a compromised sensor reading alone cannot.

Blockchain technology entered this picture because it offers a genuinely different answer to an old distributed-systems problem: how do mutually

distrusting parties agree on a shared, tamper-evident record of events without a trusted intermediary. The mechanism — a replicated, append-only ledger maintained through a consensus protocol, secured by cryptographic hashing and, in many designs, economic incentives — was developed for financial applications but turns out to map unusually well onto the coordination problem that multi-robot trust management is trying to solve [6,7]. If every robot in a fleet writes its state reports, task claims, and votes to a shared ledger that no single robot (Byzantine or not) can unilaterally rewrite, then trust bookkeeping stops being something each robot has to reconstruct locally from noisy pairwise interactions and becomes something the whole fleet can read directly off a tamper-evident public record.

This article synthesizes the literature on that combination: blockchain-enabled trust management for autonomous multi-robot and swarm-robotic systems. We are not reporting a new experiment or a new protocol; we are tracing what a genuinely active — if still fairly young — research community has built, tested, and reported over the past several years, and being explicit about where the evidence is strong and where it remains thin. We organize the discussion around six themes: the foundations of blockchain and distributed ledger technology as they apply to distributed autonomous systems; the pre-blockchain trust management literature that motivates the problem; the architectures researchers have actually proposed for putting a blockchain inside a robot fleet's coordination loop; the consensus protocols at the center of that architecture and their measured resilience to Byzantine robots; a set of applied domains — UAV swarms, task allocation, federated learning — where the pattern has been tested; and the open challenges that, in our reading, remain substantially unresolved. We close with a conclusion that tries to state plainly what the evidence does and does not support.

It is worth being explicit about scope from the outset. Most of the quantitative results synthesized here come from simulation studies and small physical swarms of tens of robots, not large fielded deployments of hundreds or thousands of units. The strongest, most carefully measured results in this literature — reported fault-tolerance thresholds, consensus-time figures, computational-overhead numbers — come

overwhelmingly from a handful of research groups working closely with the IRIDIA laboratory's swarm-robotics program and a smaller number of UAV-security groups. That concentration is itself informative about how young this specific sub-field is, and we return to it in the Open Challenges section.

## 2. Foundations of Blockchain and Distributed Ledger Technology

A blockchain, at its technical core, is a specific way of solving an old problem in distributed computing: how a set of mutually distrusting nodes can agree on an ordered, append-only sequence of events, even when some nodes might fail arbitrarily or behave maliciously — the Byzantine fault tolerance problem. The National Institute of Standards and Technology's overview of the technology lays out the now-standard architecture plainly: transactions are grouped into blocks, each block is cryptographically linked to its predecessor via a hash pointer, the resulting chain is replicated across many nodes, and a consensus protocol determines which node gets to propose the next block and how the rest of the network validates it, so that altering any past block would require redoing the cryptographic work for every subsequent block across a majority of the replicated copies simultaneously [7]. Zheng et al.'s widely cited survey frames the resulting system along three now-conventional axes that matter directly for a robotics application: decentralization (no single node's failure or corruption compromises the ledger), persistency (once a transaction is deeply enough buried under subsequent blocks, reversing it becomes computationally infeasible), and auditability (every node can independently verify the full history) [6]. Those three properties, taken together, are precisely what a fleet of robots needs if it wants a shared record of who claimed what task, who reported what sensor reading, and who violated what protocol, that no single compromised robot can unilaterally rewrite.

The consensus mechanism is where the design space actually lives, and different mechanisms trade off very differently along axes that matter enormously for embedded, battery-powered, latency-sensitive robotic hardware. Proof-of-work mechanisms, the original Bitcoin-style design, achieve Byzantine fault tolerance through computational cost — a node earns the right to propose the next block by winning a cryptographic puzzle-solving race — but the resulting energy and

latency cost is generally incompatible with a robot that has to react to its physical environment in real time and cannot spend seconds or minutes per block on proof-of-work computation. Practical Byzantine Fault Tolerance (PBFT), introduced by Castro and Liskov, takes a different and, for robotics purposes, considerably more attractive approach: a fixed, known set of replicas exchange multiple rounds of signed messages to agree on each operation directly, without any computational puzzle, guaranteeing both safety and liveness as long as no more than one-third of the replicas are simultaneously faulty — formally, a system of  $3f+1$  replicas tolerates up to  $f$  Byzantine nodes [8]. That one-third bound recurs constantly throughout the blockchain-robotics literature discussed below, sometimes matched exactly and sometimes explicitly improved upon by protocol variants built specifically for robot swarms.

Internet-of-Things security research had already been grappling with a closely related version of this problem before robotics researchers picked it up in earnest, and that IoT literature supplies useful vocabulary and a cautionary note. Dorri, Kanhere, Jurdak, and Gauravaram's smart-home architecture was among the first to demonstrate a genuinely lightweight blockchain design for resource-constrained devices, using a hierarchical structure in which a more capable "miner" node handles the heavier consensus and cryptographic work on behalf of a cluster of lightweight IoT devices that could not run a full blockchain client themselves [9]. Kouicem, Bouabdallah, and Lakhlef's top-down survey of IoT security more broadly situates blockchain as one tool among several — alongside lightweight cryptography, intrusion detection, and access control — and is candid that no single mechanism, blockchain included, resolves the fundamental tension between security guarantees and the memory, compute, and energy budgets that constrained devices actually have [10]. Robots are, in this sense, IoT devices with actuators, and the same resource tension applies with, if anything, added urgency, because a robot that stalls waiting on a slow consensus round is not merely failing to update a dashboard, it is failing to move, sense, or act.

## 3. Trust Management Models for Multi-Robot Systems

Before blockchain entered the robotics literature, trust management for distributed autonomous systems

already had a substantial pre-history, and understanding that history clarifies exactly what problem blockchain was recruited to solve. Jøsang, Ismail, and Boyd's survey of trust and reputation systems for online service provision, though written for e-commerce platforms rather than robots, established a taxonomy that the robotics literature has largely inherited: trust can be computed from direct first-hand interaction, propagated as reputation from third-party reports, and combined using models ranging from simple weighted averages to Bayesian and subjective-logic formulations that explicitly represent uncertainty rather than collapsing it into a single trust score [3]. That last point — representing uncertainty rather than hiding it — turns out to matter for robotics specifically, because a robot with too little interaction history to judge a peer reliably is in a genuinely different epistemic position than a robot that has gathered strong evidence of misbehavior, and conflating the two leads to trust systems that are either too credulous toward new robots or too punitive toward robots that simply have not interacted much yet.

Wireless sensor network research adapted these ideas to a setting much closer to swarm robotics: many cheap, resource-constrained, often battery-powered nodes, deployed densely, with no realistic expectation of centralized administration once deployed. Khalid et al.'s comparative study surveys the resulting trust and reputation schemes and is useful for cataloguing the design axes that keep reappearing: whether trust is computed and stored centrally or fully distributed across nodes, whether it is based on direct observation, indirect (recommended) observation, or both, and how aggressively a scheme penalizes a single bad interaction versus averaging it into a longer history [4]. The paper's comparative framing is a useful reminder that "trust management" is not one algorithm but a design space with real tradeoffs — a scheme that reacts instantly to a single suspicious report is vulnerable to false accusations from a malicious minority, while a scheme that only reacts to sustained patterns is correspondingly slow to isolate a genuinely compromised node, and neither failure mode is free in a system where the nodes in question are physically capable of causing harm.

Mobile ad hoc networks (MANETs) sit closer still to multi-robot systems because they share the property that network topology itself is not fixed — nodes move,

links form and break, and any trust computation has to remain valid under constant topological churn. Govindan and Mohapatra's survey of trust computation and trust dynamics in MANETs is explicit that trust in a mobile, decentralized setting is not a static property to be computed once but a dynamic quantity that has to be continuously re-evaluated as network conditions, node behavior, and even node identity (in the presence of Sybil attacks, where one adversary presents multiple false identities) change over time [5]. Multi-robot systems inherit this dynamism directly — a ground robot fleet operating in a warehouse or a UAV swarm covering a search area is, from a networking perspective, essentially a mobile ad hoc network with actuators attached, and Sybil-style attacks (a single malicious controller spoofing many apparent robot identities to bias a fleet's collective decision) are a concern the multi-robot blockchain literature returns to directly, as discussed below.

What the pre-blockchain trust literature could not solve cleanly, and what motivated the shift toward blockchain-based designs specifically, is the single-point-of-failure problem inherent in most centralized or even loosely federated trust-aggregation schemes: if trust scores live on one coordinating node, or are computed through a protocol that a sufficiently well-positioned adversary can manipulate, then the trust system itself becomes the attack surface. Brambilla, Ferrante, Birattari, and Dorigo's influential review of swarm robotics from a swarm-engineering perspective makes a related point about the broader field: the central appeal of swarm robotics as a design paradigm is robustness through decentralization and redundancy, and any coordination mechanism — trust management very much included — that reintroduces a single point of failure undercuts the entire premise of the swarm approach [11]. Multi-robot task allocation research has converged on a similar concern from the systems-engineering side: Athira, Divya Udayan, and Subramaniam's systematic review of multi-robot task allocation surveys market-based, optimization-based, and swarm-intelligence-based allocation methods and notes that the great majority of the literature it reviews assumes cooperative, honestly-reporting robots, treating the possibility of a misreporting or malicious participant as largely out of scope [2] — a gap that trust-aware, and specifically blockchain-based, allocation mechanisms are a direct response to.

#### 4. Blockchain-Enabled Trust Architectures for Swarm and Multi-Robot Systems

The first explicit proposal to bring blockchain technology into robot swarm coordination, as far as we can trace it, is Castelló Ferrer's short framing paper, which argued that the blockchain's combination of a shared, tamper-evident ledger and programmable smart contracts could serve as a general-purpose coordination substrate for robot swarms, handling tasks from secure multi-robot voting to auditable task allocation without requiring a centralized coordinator that a swarm's decentralized design philosophy would otherwise reject [12]. The proposal was largely conceptual, but it named the specific properties — tamper-evidence, decentralization, and programmability via smart contracts — that essentially every subsequent architecture in this literature has built on.

Concrete, physically tested architectures followed within a few years. Pacheco, Strobel, and Dorigo built and demonstrated a real physical swarm of ground robots that ran blockchain consensus over an ad hoc wireless network rather than a simulated or wired testbed, which matters because it forces the architecture to confront communication realities — packet loss, variable range, intermittent connectivity — that a simulation can too easily idealize away [13]. Their system used the blockchain not merely as a passive audit log but as the actual mechanism through which robots agreed on a shared collective decision, with each robot's locally sensed evidence entered onto the chain and consensus over the aggregated evidence emerging from the blockchain's own agreement protocol rather than from a separate, blockchain-external voting round. This is an architecturally important distinction that recurs across the literature: some designs use blockchain purely as an immutable log that a separate trust-computation layer reads from, while others fold the trust computation itself into the consensus mechanism, so that reaching blockchain consensus and reaching swarm-level agreement become the same event.

Dorigo, Pacheco, Reina, and Strobel's 2024 review in *Nature Reviews Electrical Engineering* consolidates several years of this research program into a broader architectural account of what blockchain buys a mobile multi-robot system specifically, distinguishing three roles the technology can play: as a secure communication and data-sharing substrate between

robots that do not otherwise trust each other's channel, as a coordination mechanism for collective decision-making that is provably robust to a bounded number of misbehaving participants, and as an economic layer — using blockchain-native tokens or cryptocurrencies — for incentivizing honest reporting and penalizing detected misbehavior [14]. That third role, the economic layer, is worth dwelling on because it is where trust management and blockchain's origins in cryptocurrency intersect most directly. Van Calck, Pacheco, Strobel, Dorigo, and Reina built and tested a blockchain-based information market in which self-interested robots — modeled explicitly as agents that might withhold or falsify information if doing so benefited them individually — buy and sell sensed information from each other using blockchain-recorded tokens, with the market mechanism itself, rather than an external enforcement layer, creating an incentive for robots to report honestly because a robot with a track record of selling bad information loses the ability to profitably sell information at all [15]. This line of work treats trust not as a score to be computed and consulted but as an emergent property of a correctly designed incentive mechanism, which is a genuinely different engineering strategy from the reputation-scoring approaches inherited from the pre-blockchain trust literature, and the two strategies are not mutually exclusive — several of the architectures discussed here combine an explicit reputation ledger with token-based incentives layered on top.

A closely related and increasingly discussed architecture question is how far this pattern extends beyond simple voting and information-sharing toward more general distributed learning. Pacheco, De Vos, Reina, Dorigo, and Strobel's recent work on securing federated learning in robot swarms using blockchain technology extends the pattern to a setting where robots are not just sharing sensed facts but collaboratively training a shared model, and where a malicious robot could in principle poison the shared model rather than merely misreport a single fact — a strictly harder trust problem, because a poisoned model's effects are diffuse and delayed rather than immediately checkable against a single ground-truth measurement [16]. Their use of the blockchain to record and verify each robot's contributed model updates, rather than trusting a central aggregator to catch bad updates unaided, is a natural extension of the token-economy and ledger-based verification ideas

discussed above into a domain where the thing being verified is a gradient update rather than a sensor reading.

It is worth noting that architectural elegance and deployment readiness are not the same thing, and the gap between them recurs across essentially all of the papers discussed in this section. Sharma, Prajapati, and Bishnoi's recent chapter on lightweight blockchain solutions for resource-constrained multi-robot systems is explicit that most of the architectures proposed in the research literature — including several discussed above — were validated on swarms of tens of robots running for minutes to hours, not the hundreds-to-thousands-of-units, weeks-to-months deployments that industrial and field-robotics applications would actually require, and that the memory and energy overhead of even a "lightweight" blockchain client remains a real constraint on the smallest classes of robotic hardware [17]. This is a theme we return to directly in the Open Challenges section, but it is worth flagging here because it should temper how the following section's specific, well-measured numbers are read: they are honest, carefully reported results, but they are results from a still-small set of testbeds.

## 5. Consensus Protocols and Byzantine-Robot Resilience

The consensus protocol is where a blockchain-enabled trust architecture either does or does not deliver on its central promise — that the system remains correct even when some fraction of its participants are actively lying. Strobel, Castelló Ferrer, and Dorigo's *Frontiers in Robotics and AI* paper is, to our knowledge, the most carefully quantified head-to-head comparison in this literature, and it is worth walking through its actual reported numbers rather than summarizing them loosely, because the differences between the compared approaches are large and instructive. The study compared a blockchain-based consensus approach against two established distributed-consensus baselines — a Linear Consensus Protocol (LCP), which is simple but has no built-in resilience to malicious values, and the Weighted-Mean-Subsequence-Reduced (W-MSR) algorithm, which discards extreme values before averaging and so offers some resilience — in a physically-motivated simulated swarm of twenty robots reaching agreement on a shared environmental estimate [18]. With a single Byzantine robot that simply reports a

constant, incorrect value, the paper reports that the Linear Consensus Protocol fails outright: every non-Byzantine robot converges to the Byzantine robot's false value. W-MSR performs considerably better with small numbers of Byzantine robots but its accuracy degrades sharply once more than a few are present. The blockchain-based approach, by contrast, remained largely unaffected as the number of Byzantine robots increased: at seven Byzantine robots out of twenty — 35% of the swarm, above even PBFT's classical one-third bound — the paper reports an absolute estimation error of roughly 20% for the blockchain approach, compared to roughly 35% for W-MSR and roughly 75% (essentially complete failure) for the Linear Consensus Protocol under the same conditions. The paper is candid that this resilience is not free: the blockchain approach showed a measurable, if modest, increase in consensus time as the number of Byzantine robots increased, a cost the classical baselines did not pay because they simply failed rather than working harder.

**Table 1: Representative consensus and trust-verification mechanisms for multi-robot and UAV-swarm systems, with the Byzantine-resilience or performance figures each cited source reports. [8,18,21,23].**

| Mechanism                                           | Source              | Tested/Analyzed Scale                           | Reported Resilience or Performance                                                         |
|-----------------------------------------------------|---------------------|-------------------------------------------------|--------------------------------------------------------------------------------------------|
| Practical Byzantine Fault Tolerance (PBFT)          | Castro & Liskov [8] | General distributed replica set (3f+1 replicas) | Guarantees safety and liveness with up to 1/3 of replicas simultaneously faulty            |
| Linear Consensus Protocol (LCP), no fault tolerance | Strobel et al. [18] | 20-robot swarm, 7 Byzantine (35%)               | ~75% absolute estimation error -- converges to the false value of even one Byzantine robot |
| Weighted-Mean-Subsequence-Reduced (W-MSR)           | Strobel et al. [18] | 20-robot swarm, 7 Byzantine (35%)               | ~35% absolute estimation error; degrades sharply beyond ~3 Byzantine robots                |
| Blockchain-based swarm consensus                    | Strobel et al. [18] | 20-robot swarm, 7 Byzantine (35%)               | ~20% absolute estimation error; largely unaffected as Byzantine count increases            |
| DTPBFT (dynamic trust-tiered PBFT variant)          | Han et al. [21]     | UAV swarm scaled to 200 units                   | Tolerates more than 1/3 malicious nodes; 0.24 s measured consensus time at 200 UAVs        |
| Blockchain-based UAV authentication (fog-assisted)  | Du et al. [23]      | Fleet of up to 10 UAVs                          | Computational authentication overhead under 50 ms per UAV                                  |

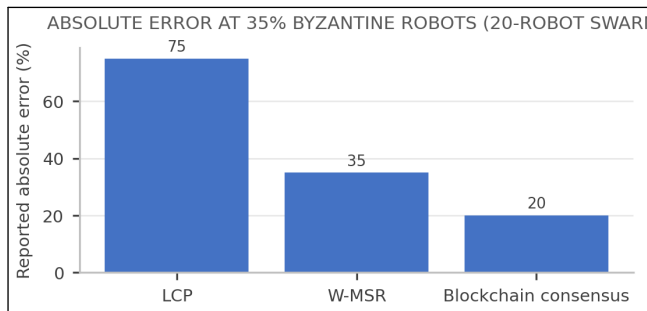


**Figure 1: Absolute estimation error of a 20-robot swarm reaching consensus with 7 of 20 robots (35%) behaving as Byzantine, comparing a Linear Consensus Protocol (LCP), the Weighted-Mean-Subsequence-Reduced (W-MSR) algorithm, and a blockchain-based consensus approach, as reported by Strobel et al. [18].**

That the blockchain-based approach in Strobel et al.'s study tolerated more Byzantine robots than PBFT's classical one-third bound is worth pausing on, because it illustrates a genuinely useful design freedom in swarm robotics that pure distributed-systems theory does not usually have available: a physical swarm can incorporate spatial or sensor-based corroborating evidence — geometric constraints, physical plausibility checks on reported values — into its consensus logic in ways that an abstract Byzantine agreement protocol, which typically assumes adversarial values are otherwise indistinguishable from honest ones, cannot. Follow-on work has pushed this idea further: recent work on Byzantine fault detection in collaborative Swarm-SLAM settings explicitly combines blockchain-recorded agreement with geometric consistency constraints between robots' reported poses and maps, using the physical implausibility of a fabricated map segment as an additional signal that a purely value-based consensus protocol would not have access to. Strobel, Pacheco, and Dorigo's follow-on Science Robotics paper extends the same research program from passive consensus toward active defense, introducing a blockchain-based token economy in which Byzantine robots are not merely outvoted but economically penalized — losing tokens for detected misbehavior — so that the swarm actively neutralizes, rather than merely tolerates, a persistently harmful minority of participants [19]. Castelló Ferrer's subsequent Frontiers in Blockchain piece takes a more skeptical, security-auditing stance on this same body of work, arguing that robot security more broadly — including the sensors, actuators, and physical layer that a blockchain's cryptographic guarantees do not by themselves protect — remains the harder and less-solved half of the problem, and that a blockchain-secured consensus layer sitting on top of an

insecure physical robot is a meaningfully incomplete solution [20].

PBFT's classical one-third bound continues to anchor comparisons even where swarm-specific protocols improve on it. Han, Wu, and Sui's DTPBFT algorithm, designed specifically for UAV swarms, reports tolerance beyond the standard one-third threshold while explicitly targeting the scalability problem that plain PBFT has at large replica counts — PBFT's message complexity grows quadratically with the number of participating nodes, which becomes a serious bottleneck once a swarm reaches into the hundreds of units. The paper reports a measured consensus time of 0.24 seconds at a swarm size of 200 UAVs, offered as evidence that a dynamically-structured trust-tiered consensus scheme can keep large-swarm consensus latency within a range compatible with real-time coordination, where plain PBFT's quadratic messaging cost would make the same swarm size considerably slower [21]. Table 1 below places several of these consensus and trust-verification approaches side by side, and Figure 1 reproduces the Strobel et al. absolute-error comparison directly, because it is, in our reading, the single clearest quantitative demonstration in this literature of what a blockchain-based consensus layer buys a robot swarm over classical alternatives.

## 6. Applications: UAV Swarms, Task Allocation, and Federated Collaboration

Unmanned aerial vehicle swarms are, among multi-robot application domains, the one where blockchain-enabled trust management has received the most sustained applied attention, for reasons that follow fairly directly from the domain's constraints: UAV fleets often operate over wireless links with no fixed infrastructure, frequently in contested or safety-critical settings (search and rescue, environmental monitoring, and explicitly, in some of this literature, military coordination) where a single spoofed or hijacked unit is a serious operational risk, and individual UAVs are resource-constrained enough that consensus latency directly competes with flight-control computation for the same limited onboard compute budget. Han, Wu, and Sui's DTPBFT work discussed above targets this domain specifically, and its reported large-swarm consensus latency is offered as evidence that trust-tiered consensus can scale to fleet sizes that matter operationally [21]. Zhou, Yang, Ma, and He's work on

secure and resilient UAV swarm networking takes a complementary angle, focusing less on the consensus algorithm itself and more on the networking layer beneath it — arguing that a blockchain-secured consensus protocol is only as resilient as the wireless mesh network carrying its messages, and that UAV-specific concerns like intermittent line-of-sight, jamming, and rapid topology change need to be designed into the blockchain's networking layer explicitly rather than assumed away [22]. Du, Tao, Yuan, Li, and Zhou's blockchain-based authentication scheme for fog-assisted UAV networks addresses a narrower but operationally critical sub-problem — verifying that a UAV requesting to join a swarm-coordination session actually is who it claims to be, before that UAV's reports enter the trust computation at all — reporting a measured computational authentication overhead of well under fifty milliseconds per UAV for fleets of up to ten units authenticating through fog-computing-assisted blockchain nodes, small enough to run alongside real-time flight control without becoming the coordination bottleneck [23]. Together, these three strands — consensus, networking resilience, and authentication — make the case that UAV-swarm trust management is not solved by any one blockchain component in isolation but by treating consensus, network security, and identity verification as a single co-designed stack.

Multi-robot task allocation is the second application domain where trust and blockchain intersect directly, and the connection is, on reflection, fairly intuitive: a task-allocation market where robots bid for tasks based on claimed capability or claimed sensor readings is only as good as the market participants' honesty, and a robot that can win tasks by lying about its capability or location degrades the whole fleet's efficiency in a way that is much harder to detect after the fact than to prevent up front. Athira, Divya Udayan, and Subramaniam's systematic review of the broader multi-robot task allocation literature documents market-based allocation mechanisms — auctions, in particular — as one of the dominant paradigms in the field, and notes, as discussed above, that trust-aware variants of these markets, where bids or capability claims are checked against a verifiable record rather than taken at face value, remain a comparatively underdeveloped corner of an otherwise mature literature [2]. The blockchain-based information market

described by Van Calck et al. is one of the more concrete responses to exactly this gap, treating task-relevant information itself as a good traded in a market whose blockchain-recorded transaction history makes a robot's track record of honest versus dishonest reporting directly auditable by every other robot in the fleet, rather than something each robot would otherwise have to reconstruct independently from its own limited pairwise interaction history [15].

A third applied thread worth noting explicitly, because it sits somewhat outside the swarm-robotics research program that dominates the rest of this section, is pandemic-response robotics. Alsamhi and Lee's review of blockchain-empowered multi-robot collaboration for pandemic response surveys proposed uses of robot fleets for contactless delivery, disinfection, and remote patient monitoring during COVID-19, and situates blockchain specifically as the mechanism for maintaining a verifiable, tamper-evident record of which robots serviced which locations or patients — a record whose integrity has real public-health consequences (contact tracing accuracy, verifiable disinfection logs) beyond the swarm-internal coordination concerns that dominate the rest of this literature [24]. The paper is more a survey of proposed architectures than a report of deployed systems, and its own framing is candid that much of what it surveys remained, at the time of writing, closer to design proposal than field-tested deployment — a pattern that, as the next section discusses, recurs across nearly the whole of this literature.

## 7. Open Challenges

Several gaps recur across the literature synthesized here clearly enough to be worth stating plainly rather than leaving implicit. First, and most consequential: the great majority of the strongest, most carefully quantified results — including the specific numbers in Figure 1 and most of Table 1 — come from a comparatively small number of research groups, testing swarms of tens of physical or simulated robots for periods of minutes to hours, not the hundreds-to-thousands-of-units, weeks-to-months deployments that industrial logistics, agricultural, or large-scale UAV applications would eventually require [18,13,17]. Aditya, Singh, Singh, and Kalla's broader survey of blockchain in robotics reaches a similar conclusion from a wider vantage point, noting that the field as a whole remains concentrated in academic testbeds and that the transition to field-scale

industrial deployment involves engineering questions — sustained network reliability over months rather than experimental sessions, integration with existing non-blockchain fleet-management software, regulatory and liability questions around an immutable but potentially erroneous ledger entry — that the research literature has barely begun to address [25].

Second, the resource-cost side of the equation remains genuinely unresolved for the smallest and most constrained classes of robotic hardware. Even the lightweight blockchain designs surveyed here, including hierarchical architectures that offload heavier consensus work to a more capable coordinating node [9], impose memory, energy, and latency costs that a full swarm-robotics deployment of very small, cheap, battery-constrained units may not be able to absorb without redesigning the swarm's hardware budget around the blockchain client rather than the other way around, a tension Sharma, Prajapati, and Bishnoi make explicit [17]. The consensus-time results reported by Strobel et al. and by Han et al. are genuinely encouraging on this front, but both studies test hardware substantially more capable than the very cheapest end of the swarm-robotics spectrum, and neither study, in our reading, fully closes the question of what happens at that cheapest tier.

Third, blockchain's cryptographic and consensus guarantees do not, by themselves, secure the physical and sensing layers that ultimately determine whether a robot's reported data is trustworthy in the first place. Castelló Ferrer's security-auditing perspective is the most direct statement of this concern in the literature synthesized here: a blockchain can guarantee that a reported value, once written, cannot be silently altered, but it cannot guarantee that the value was accurate at the moment of writing, and a sufficiently sophisticated attacker who compromises a robot's sensors or actuators directly, rather than attacking the network or the ledger, sidesteps the blockchain's protections entirely [20]. This is not a criticism unique to blockchain-based designs — no trust-management scheme, blockchain-based or otherwise, can fully solve the problem of a compromised sensor reporting confidently false data — but it does mean that claims of a blockchain-secured swarm being simply "secure" overstate what the technology actually delivers, and the more careful papers in this literature are consistently

explicit about that boundary.

Fourth, the interaction between economic incentive mechanisms — the token economies and information markets described above — and adversarial robustness is less thoroughly tested than the pure consensus-protocol results. Van Calck et al.'s information market and Strobel et al.'s token economy both report that their incentive designs successfully discourage dishonest reporting under the adversarial models they test [15,19], but an incentive mechanism's robustness is famously sensitive to exactly which adversarial strategies are modeled, and a sophisticated attacker with detailed knowledge of the market mechanism's rules — rather than the comparatively simple constant-value or random-value Byzantine robots tested in most of this literature — is a threat model the field has only begun to probe. Fifth, and finally, benchmarking standards across this literature are inconsistent: swarm sizes, Byzantine-robot fractions, communication models, and reported metrics (absolute error, consensus time, computational overhead, token-economy stability) vary enough from paper to paper that direct cross-paper comparison, beyond the qualitative pattern that blockchain-based approaches consistently outperform naive consensus baselines under adversarial conditions, is harder than it should be. Kouicem, Bouabdallah, and Lakhlef's broader observation about IoT security research — that no single mechanism resolves the fundamental tension between security guarantees and constrained-device resource budgets — applies to this literature with essentially no modification required [10]. None of these gaps, in our reading, undermine the core architectural claim that blockchain-enabled trust management is a genuinely useful pattern for multi-robot coordination; they are the honest state of a young, rapidly developing literature, and studies claiming field-deployment readiness for the pattern should be read against these gaps explicitly.

## 8. Conclusion

The literature surveyed here supports a fairly specific and, we think, well-evidenced claim: blockchain-enabled trust management — a shared, tamper-evident ledger combined with a consensus protocol that a bounded fraction of Byzantine robots cannot subvert, optionally paired with token-based economic incentives for honest reporting — is a genuinely effective architectural pattern for

coordinating autonomous multi-robot systems under adversarial or unreliable conditions, and the evidence for that claim is unusually concrete for a young sub-field: Strobel, Castelló Ferrer, and Dorigo's direct, quantified comparison shows a blockchain-based consensus approach remaining largely accurate at a Byzantine-robot fraction where classical consensus protocols fail outright or degrade sharply [18], and that basic result has since been extended, replicated in spirit, and built upon across physical swarm testbeds [13], UAV-specific consensus algorithms operating at swarm sizes in the hundreds [21], and economic mechanisms that move from merely tolerating to actively neutralizing persistently dishonest participants [19].

At the same time, the pattern inherits real constraints from both of the technologies it combines. It inherits classical distributed-systems trust management's dependence on a well-specified adversarial model, and its guarantees are correspondingly only as strong as that model's fit to whatever attacker a deployed system actually faces [3,5]. It inherits blockchain's resource cost, which — even in the deliberately lightweight architectures this literature has produced — remains a real constraint for the smallest, most constrained tier of robotic hardware [9,17]. And it does not, by itself, secure the physical sensing and actuation layers that ultimately determine whether the data entering the ledger was honest in the first place, a limitation the more security-conscious papers in this literature are careful to name directly [20]. Making the pattern deployable at the scale that logistics, agricultural, and large UAV fleets would eventually require depends on treating all of these constraints as an engineering roadmap rather than a reason to set the approach aside — continuing to push consensus latency down as swarm sizes grow [21], continuing to test economic incentive designs against more sophisticated adversaries than the literature has tried so far [15], and being honest, as the strongest papers already are, about the difference between a result demonstrated on a testbed of tens of robots and a claim about deployment readiness at industrial scale [25,14]. Read together, the consistency with which independent groups working from different starting points — swarm robotics, UAV security, IoT trust management — have converged on structurally similar blockchain-based trust architectures is, in our view, the strongest available evidence that this is a genuine and durable pattern for autonomous multi-robot coordination, and one that

merits continued, deployment-grounded empirical scrutiny rather than further purely simulated validation.

## 9. References

1. Yan, Z., Jouandeau, N., Cherif, A.A. (2013) A Survey and Analysis of Multi-Robot Coordination. *International Journal of Advanced Robotic Systems*, 10(12): 399. DOI: [10.5772/57313](https://doi.org/10.5772/57313)
2. Athira, K.A., Divya Udayan, J., Subramaniam, U. (2024) A Systematic Literature Review on Multi-Robot Task Allocation. *ACM Computing Surveys*, 57(3): 1-28. DOI: [10.1145/3700591](https://doi.org/10.1145/3700591)
3. Jøsang, A., Ismail, R., Boyd, C. (2007) A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems*, 43(2): 618-644. DOI: [10.1016/j.dss.2005.05.019](https://doi.org/10.1016/j.dss.2005.05.019)
4. Khalid, O., Khan, S.U., Madani, S.A., Hayat, K., Khan, M.I., Min-Allah, N., Kolodziej, J., Wang, L., Zeadally, S., Chen, D. (2013) Comparative Study of Trust and Reputation Systems for Wireless Sensor Networks. *Security and Communication Networks*, 6(6): 669-688. DOI: [10.1002/sec.597](https://doi.org/10.1002/sec.597)
5. Govindan, K., Mohapatra, P. (2012) Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 14(2): 279-298. DOI: [10.1109/SURV.2011.042711.00083](https://doi.org/10.1109/SURV.2011.042711.00083)
6. Zheng, Z., Xie, S., Dai, H.N., Chen, X., Wang, H. (2018) Blockchain Challenges and Opportunities: A Survey. *International Journal of Web and Grid Services*, 14(4): 352-375. DOI: [10.1504/IJWGS.2018.095647](https://doi.org/10.1504/IJWGS.2018.095647)
7. Yaga, D., Mell, P., Roby, N., Scarfone, K. (2018) Blockchain Technology Overview. *National Institute of Standards and Technology Internal Report 8202*. DOI: [10.6028/NIST.IR.8202](https://doi.org/10.6028/NIST.IR.8202)
8. Castro, M., Liskov, B. (2002) Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Transactions on Computer Systems*, 20(4): 398-461. DOI: [10.1145/571637.571640](https://doi.org/10.1145/571637.571640)
9. Dorri, A., Kanhere, S.S., Jurdak, R., Gauravaram, P. (2017) Blockchain for IoT Security and Privacy: The Case Study of a Smart Home. 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI: 618-623. DOI: [10.1109/PERCOMW.2017.7917634](https://doi.org/10.1109/PERCOMW.2017.7917634)
10. Kouicem, D.E., Bouabdallah, A., Lakhlef, H. (2018) Internet of Things Security: A Top-Down Survey. *Computer Networks*, 141: 199-221. DOI: [10.1016/j.comnet.2018.03.012](https://doi.org/10.1016/j.comnet.2018.03.012)
11. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M. (2013) Swarm Robotics: A Review from the Swarm Engineering

- Perspective. *Swarm Intelligence*, 7(1): 1-41. DOI: [10.1007/s11721-012-0075-2](https://doi.org/10.1007/s11721-012-0075-2)
12. Castelló Ferrer, E. (2016) The Blockchain: A New Framework for Robotic Swarm Systems. arXiv preprint. DOI: [10.48550/arXiv.1608.00695](https://doi.org/10.48550/arXiv.1608.00695)
  13. Pacheco, A., Strobel, V., Dorigo, M. (2020) A Blockchain-Controlled Physical Robot Swarm Communicating via an Ad-Hoc Network. *Swarm Intelligence (ANTS 2020)*, Lecture Notes in Computer Science 12421: 3-15. DOI: [10.1007/978-3-030-60376-2\\_1](https://doi.org/10.1007/978-3-030-60376-2_1)
  14. Dorigo, M., Pacheco, A., Reina, A., Strobel, V. (2024) Blockchain Technology for Mobile Multi-Robot Systems. *Nature Reviews Electrical Engineering*, 1(4): 264-274. DOI: [10.1038/s44287-024-00034-9](https://doi.org/10.1038/s44287-024-00034-9)
  15. Van Calck, L., Pacheco, A., Strobel, V., Dorigo, M., Reina, A. (2023) A Blockchain-Based Information Market to Incentivise Cooperation in Swarms of Self-Interested Robots. *Scientific Reports*, 13: 20417. DOI: [10.1038/s41598-023-46238-1](https://doi.org/10.1038/s41598-023-46238-1)
  16. Pacheco, A., De Vos, S., Reina, A., Dorigo, M., Strobel, V. (2025) Securing Federated Learning in Robot Swarms Using Blockchain Technology. *Distributed Autonomous Robotic Systems*, Springer Proceedings in Advanced Robotics: 473-488. DOI: [10.1007/978-3-032-04584-3\\_32](https://doi.org/10.1007/978-3-032-04584-3_32)
  17. Sharma, S., Prajapati, A., Bishnoi, A. (2026) Architectures for Decentralized Trust: Implementing Lightweight Blockchain Solutions in Resource-Constrained Multi-Robot Systems. *Multi-Robot Collaboration: The Enabling Technologies and Their Relevance in Today's Industrial Landscape*, Studies in Systems, Decision and Control. DOI: [10.1007/978-3-032-21150-7\\_9](https://doi.org/10.1007/978-3-032-21150-7_9)
  18. Strobel, V., Castelló Ferrer, E., Dorigo, M. (2020) Blockchain Technology Secures Robot Swarms: A Comparison of Consensus Protocols and Their Resilience to Byzantine Robots. *Frontiers in Robotics and AI*, 7: 54. DOI: [10.3389/frobt.2020.00054](https://doi.org/10.3389/frobt.2020.00054)
  19. Strobel, V., Pacheco, A., Dorigo, M. (2023) Robot Swarms Neutralize Harmful Byzantine Robots Using a Blockchain-Based Token Economy. *Science Robotics*, 8(79): eabm4636. DOI: [10.1126/scirobotics.abm4636](https://doi.org/10.1126/scirobotics.abm4636)
  20. Castelló Ferrer, E. (2023) If Blockchain Is the Solution, Robot Security Is the Problem. *Frontiers in Blockchain*, 6: 1181820. DOI: [10.3389/fbloc.2023.1181820](https://doi.org/10.3389/fbloc.2023.1181820)
  21. Han, P., Wu, X., Sui, A. (2024) DTPBFT: A Dynamic and Highly Trusted Blockchain Consensus Algorithm for UAV Swarm. *Computer Networks*, 250: 110602. DOI: [10.1016/j.comnet.2024.110602](https://doi.org/10.1016/j.comnet.2024.110602)
  22. Zhou, X., Yang, L., Ma, L., He, H. (2023) Towards Secure and Resilient Unmanned Aerial Vehicles Swarm Network Based on Blockchain. *IET Blockchain*, 4: 483-493. DOI: [10.1049/bic2.12050](https://doi.org/10.1049/bic2.12050)
  23. Du, X., Tao, S., Yuan, K., Li, Y., Zhou, Y. (2024) A Blockchain Authentication Scheme for UAV-Aided Fog Computing. *Complex & Intelligent Systems*, 10(2): 1689-1702. DOI: [10.1007/s40747-023-01232-y](https://doi.org/10.1007/s40747-023-01232-y)

24. Alsamhi, S.H., Lee, B. (2021) Blockchain-Empowered Multi-Robot Collaboration to Fight COVID-19 and Future Pandemics. IEEE Access, 9: 44173-44197. DOI: [10.1109/ACCESS.2020.3032450](https://doi.org/10.1109/ACCESS.2020.3032450)
25. Aditya, U.S.P.S., Singh, R., Singh, P.K., Kalla, A. (2021) A Survey on Blockchain in Robotics: Issues, Opportunities, Challenges and Future Directions. Journal of Network and Computer Applications, 196: 103245. DOI: [10.1016/j.jnca.2021.103245](https://doi.org/10.1016/j.jnca.2021.103245)

## 10. Declarations

### Funding

This literature synthesis received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

### Conflict of Interest

The authors declare that they have no competing interests relevant to this work.

## Data Availability Statement

This article synthesizes previously published literature and generates no new primary data; all figures and tables are drawn directly from the cited sources, which readers should consult for underlying data and code availability.

## Ethical Approval

Not applicable, as this article involves no human participants, animal subjects, or collection of primary experimental data.