

Review

Decentralized AI Agents and Blockchain: Architectures, Coordination Mechanisms, and Governance Frameworks

Marios Touloupou ^{1,*}  and Evgenia Kapassa ² 

¹ Department of Electrical Engineering, Computer Science and Engineering, Cyprus University of Technology, Limassol 3036, Cyprus

² Department of Management, Entrepreneurship and Digital Business, Cyprus University of Technology, Limassol 3036, Cyprus; evgenia.kapassa@cut.ac.cy

* Correspondence: marios.touloupou@cut.ac.cy

Abstract

Autonomous AI agents capable of holding digital assets, signing transactions, and executing smart contracts on public blockchain networks have moved from research prototypes to active deployment over the past two years. Despite this pace of adoption, no systematic treatment of their architecture, coordination protocols, and governance structures exists that spans the full design space. This survey addresses that gap through a systematic review of the literature from 2019 to 2026, covering 177 peer-reviewed publications and 14 system documentation sources, identified through a structured search of IEEE Xplore, the ACM Digital Library, Scopus, and arXiv. We classify deployed and proposed systems along four architectural dimensions: on-chain execution, off-chain agents with on-chain settlement, verifiable off-chain computation, and multi-agent on-chain interaction. Then, we examine the coordination mechanisms through which agents reach collective decisions, covering auction-based protocols, cooperative multi-agent reinforcement learning, token-incentive structures, and gossip-based peer-to-peer coordination. Governance is treated as a distinct dimension, analysed through a technical lens, covering on-chain parameter control, dispute resolution, and DAO structures, and an organizational one, covering accountability, incentive alignment, principal-agent dynamics, and regulatory compatibility). We survey applications across decentralized finance, supply chain, IoT, and agent marketplace domains, and identify six open research problems whose resolution is a prerequisite for broader deployment. The convergence of mechanism design and multi-agent reinforcement learning in asynchronous blockchain environments is identified as the direction of greatest near-term research value.

Keywords: decentralized AI agents; multi-agent reinforcement learning; blockchain; smart contracts; decentralized autonomous organizations; token incentive mechanisms; AI governance; verifiable computation; digital platform governance; Internet of Things



Academic Editor: Francesco Buccafurri

Received: 5 June 2026

Revised: 1 July 2026

Accepted: 2 July 2026

Published: 6 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In March 2024, an autonomous software agent deployed on a public blockchain network executed a series of decentralized exchange trades, paid gas fees from its own wallet, and distributed proceeds to token holders, with no per-step human approval at any stage. It was ordinary operation on the Autonolas protocol [1], one of several production systems in which AI agents currently act with genuine economic autonomy on-chain. The broader phenomenon is accelerating: agent frameworks with native wallet support, on-

chain task markets where agents hire other agents, and DAO governance systems where AI agents hold voting power are all live as of 2025–2026 [2–4].

The literature has not kept pace with this shift. Multi-agent systems research has a long tradition of studying autonomous agent coordination [5], but it largely treats the execution environment as either a simulation or a centralized server. Blockchain research, in turn, has examined smart contracts, consensus protocols, and token economics in depth [6], but tends to treat AI as a capability to be added on top of an existing system, not treated as a structural design variable. The governance literature on Decentralized Autonomous Organisations (DAOs) and decentralised protocols [7] has developed its own vocabulary (voting mechanisms, treasury management, on-chain constitutions) without systematic connection to the multi-agent coordination problem. Researchers and practitioners working at the intersection of multi-agent systems, blockchain, and governance have no unified reference as a result.

This survey constructs that reference. We examine how AI agents are architecturally situated relative to blockchain infrastructure, how they coordinate with each other through on-chain and off-chain mechanisms, and how their autonomous behaviour is constrained and legitimated through governance structures. We treat these three dimensions, namely architecture, coordination, and governance, as mutually dependent: an agent's coordination options depend on its execution architecture, and the governance mechanisms available depend on both.

The scope of this survey requires some clarification. We use the term *decentralized AI agent* to mean a software agent satisfying three conditions: operational autonomy (it acts across multiple steps without per-step human approval), economic autonomy (it controls digital assets and can transact independently), and some form of on-chain interaction, meaning its behaviour is mediated by or recorded on a blockchain network. This excludes simple smart contracts, which are deterministic and non-adaptive, and cloud-hosted AI systems that call blockchain APIs incidentally, which are not decentralized in any meaningful sense. The boundary cases are discussed in Section 2.

The maturation of Layer 2 infrastructure is one reason this survey is timely. By reducing transaction costs by two to three orders of magnitude compared to the Ethereum mainnet [8], these scaling solutions have made frequent on-chain agent operations economically viable. Large language model (LLM) agents with tool use, meaning agents that reason in natural language but act through blockchain transactions, have added a new architectural pattern with no clear precedent in the multi-agent systems literature [9]. And the EU AI Act [10] and the Markets in Crypto-Assets Regulation (MiCA) [11] came into force in 2024–2025, creating a concrete regulatory context for decentralized AI agents that did not previously exist. The governance dimension of this survey is motivated by the need to understand how these frameworks apply, and where they break down.

Cases from separate disciplines illustrate why this intersection demands unified treatment. In decentralised finance, autonomous liquidation agents on lending protocols such as Aave and Compound executed over \$200 million in forced collateral sales during the market dislocations of March 2020 and May 2021 [12]. No human approved individual transactions; no single legal entity was identifiable as the responsible operator. The resulting losses affected retail users who had no visibility into the agent logic that triggered their liquidation. This is a problem that sits at the boundary of computer science, financial regulation, and tort law. In UAV coordination, DAO-governed drone swarms are being proposed for post-disaster response across national borders [13], raising questions about which jurisdiction's aviation rules apply when the governing DAO has no registered domicile and the swarm's collective decisions emerge from an on-chain vote among anonymous token holders. In supply chain management, autonomous AI agents that flag

compliance violations and trigger smart contract penalties without human approval are entering production deployments [14], creating situations where a supplier's payment is withheld by an algorithm whose decision logic is opaque to both parties. Each case involves computer science, a domain discipline, and a governance framework that were designed independently and do not compose cleanly. A survey that addresses only the technical dimensions cannot answer the questions these cases raise.

The main contributions of this survey are as follows:

1. **A three-dimensional analytical framework** for decentralised AI agent systems, treating architecture, coordination, and governance as jointly determined design dimensions instead of independent concerns. The interaction structure across these three dimensions is made explicit throughout the paper: an agent's execution architecture constrains its coordination options, which in turn determine what governance mechanisms are available.
2. **A four-pattern architectural taxonomy** (Section 3) classifying systems by execution locus, verification mechanism, and learning paradigm, together with a 20-entry comparative table and three non-obvious cross-taxonomy observations about the current deployment state.
3. **A structured coordination mechanism comparison** (Section 4) across auction-based, MARL-based, token-incentivised, and gossip-based protocols, covering communication overhead, trust assumptions, incentive alignment, and latency tolerance.
4. **A governance analysis** (Section 5) spanning on-chain technical mechanisms (DAO structures, slashing, dispute resolution) and organisational dimensions (principal-agent alignment, regulatory compatibility under the EU AI Act and MiCA, inter-organisational adoption conditions). Two named analytical constructs (the governance vacuum and the dual agency problem) are introduced and grounded in the IS and mechanism design literatures.
5. **Six precisely stated open research problems** (Section 7), each accompanied by a success criterion that defines what a resolution would look like.

Several surveys address adjacent territory without covering the intersection this paper targets. Nguyen et al. [15] survey blockchain-based task offloading and resource allocation in edge and UAV networks, but their scope is restricted to off-chain MARL with on-chain settlement and does not address architectural alternatives, coordination mechanisms beyond MARL, or governance. Zheng et al. [6] survey blockchain systems broadly, treating AI as one application area among many, not as a structural design variable. Busoniu et al. [16] provide the standard MARL reference but predate blockchain deployment and do not address on-chain execution environments, token incentives, or DAO governance. On the IS side, Lumineau et al. [17] examine blockchain governance as an inter-organisational coordination mechanism, but without an agent system layer. Fritsch et al. [7] analyse voting power concentration in DAOs empirically, but do not connect this to AI agent deployment or coordination. The present survey differs from all of these by treating the three dimensions (architecture, coordination, and governance) as jointly determined, by grounding the analysis in a systematic 177-paper corpus, and by targeting the design space for AI agents that satisfy all three conditions in the scope definition in Section 2.3.

The paper is structured as follows. Section 2 establishes scope, defines terms, and describes the survey methodology. Section 3 classifies deployed and proposed systems along four architectural patterns. Section 4 examines coordination mechanisms across four protocol families. Section 5 analyses governance from both a technical and an organisational perspective. Section 6 surveys empirical evidence across four application domains. Section 7 identifies six open research problems, and Section 8 draws the principal conclusions.

2. Background and Scope

2.1. Blockchain Primitives

An agent operating in a blockchain environment has fewer degrees of freedom than one operating in a conventional cloud environment. Understanding those constraints is necessary before the architectural taxonomy in Section 3 will make sense.

Smart contracts are programs deployed at a fixed network address, executed by a distributed virtual machine; most commonly the Ethereum Virtual Machine (EVM) or a compatible equivalent. Their execution is deterministic, publicly verifiable, and, absent an upgrade mechanism, immutable. In regards to agent design, this has direct consequences. Computation is metered by gas cost, which limits how much work an agent can do in a single transaction. Persistent mutable state is held in contract storage rather than memory. Native randomness is unavailable without an external feed. On-chain execution is therefore cheap only for simple logic; agents that require inference over high-dimensional inputs must execute off-chain and commit only results to the chain.

Account abstraction, standardised on Ethereum through ERC-4337 [18], decouples the signing key from the account logic. A smart contract account can be controlled by an agent's key pair instead of a human-held private key, and the account's validation logic can be customised, to allow multi-signature approval, rate limiting, or session keys with constrained spending. This is the primary mechanism through which agents acquire persistent on-chain identity and asset custody that is not tied to a human wallet.

Oracle networks bridge the gap between on-chain execution and off-chain data. An agent executing on-chain has no native access to external prices, sensor readings, or API responses; it can only read contract state and transaction inputs. Oracle networks such as Chainlink [19], Pyth [20], and UMA [21] push signed data onto the chain at configurable intervals. For agents that must react to real-world events, oracle latency and update frequency are binding constraints. Manipulation resistance matters equally: price oracle manipulation has caused substantial losses in DeFi systems that relied on agents acting on stale or distorted data [12].

Layer 2 rollup networks have changed the cost structure of agent operation. Ethereum mainnet gas fees make frequent on-chain agent actions uneconomical for most applications. Rollup networks, such as Arbitrum, Optimism, Base, zkSync, execute transactions off-chain and post compressed proofs or state differences to Layer 1, reducing per-transaction costs by one to two orders of magnitude. Block times on these networks are typically two seconds or less, compared to twelve seconds on Ethereum mainnet. These parameters directly constrain action frequency and agent responsiveness; as a result, most production deployments favour Layer 2 over Layer 1 environments.

Token incentive structures such as fungible tokens, staking mechanisms, and slashing conditions, are the primary tool for aligning agent behaviour with system-level objectives in the absence of a trusted authority. Staking requires an agent to lock tokens as collateral, creating a financial penalty for misbehaviour. Slashing destroys some or all of that collateral when a verifiable violation is detected. Reputation tokens, which accumulate from past behaviour and are difficult to transfer, serve a similar function where slashing would be too coarse an instrument. The design of these structures is inseparable from the coordination problem examined in Section 4.

2.2. AI and Agent Primitives

The systems surveyed draw on three families of AI methods, which differ in how an agent acquires its policy, what inputs it processes, and what behavioural guarantees, if any, it provides.

Reinforcement Learning (RL) agents learn a policy through repeated interaction with an environment by maximising cumulative reward [22]. The policy maps observed state to actions; the reward function encodes the designer's objective. In blockchain environments, the state typically includes token prices, contract balances, and pending transaction data; actions include trade execution, liquidity provision, or protocol parameter updates. Deep RL algorithms such as DQN [23] and PPO [24] are the most common in this context. A known limitation is sample inefficiency: RL agents require large numbers of environment interactions to learn effective policies, which is costly when each interaction is a real blockchain transaction with gas overhead.

Multi-agent Reinforcement Learning (MARL) extends this to settings where multiple learning agents interact simultaneously [16]. As all agents learn concurrently, the environment each agent observes is non-stationary by construction. The Centralised Training, Decentralised Execution (CTDE) paradigm [25] addresses this by allowing agents to share information during training while restricting them to local observations at execution time. QMIX [26], MADDPG [27], and COMA [28] are the most widely used CTDE algorithms in the literature. The connection to blockchain-based systems follows directly from this property: CTDE maps naturally onto environments where agents act on local information but coordinate through shared contract state during a training phase. LLM-based agents use large language models as a planning and reasoning component, with external tools for action [9,29]. The ReAct pattern, which alternates between explicit reasoning steps and action calls, is the dominant design choice. These agents handle unstructured inputs and natural-language task specifications that pure RL agents cannot. Their weaknesses run in the opposite direction; inference is expensive, behaviour is non-deterministic, and the agent does not learn from interaction without fine-tuning. Hybrid architectures, in which an LLM handles high-level task decomposition and an RL or rule-based component handles execution, appear in several recent production agent frameworks and fall within the scope of this survey.

A note on terminology: this survey uses “AI agent” to mean a system in one of these three families (RL, MARL, or LLM-based) that acts with genuine policy-level autonomy. This is narrower than the common usage of “AI” to cover any machine learning system, and narrower than the use of “agent” in economic theory to mean any rational decision-maker. Systems that apply supervised or unsupervised ML to classify or predict without taking actions, and economic agents that are rational but not implemented as software, are outside the scope of this survey even where they interact with blockchain infrastructure.

2.3. What Makes an Agent Decentralised

Definitions in this space tend to be either too narrow or too broad. Some existing surveys treat any smart contract with conditional logic as an autonomous agent [30], while others restrict the scope to LLM-based agents and exclude reinforcement learning systems that constitute the majority of current deployments [31]. Industry usage, meanwhile, frequently extends the term to systems where a human approves every transaction. None of these captures what this survey is concerned with. We anchor on three conditions, each of which must hold.

The first is *operational autonomy*: the agent acts across multiple steps without per-step human approval. A system that executes a transaction only after a human clicks confirm is not operationally autonomous in the relevant sense, even if the underlying decision logic is sophisticated. The threshold matters because it determines where accountability sits and what governance mechanisms are required.

The second is *economic autonomy*: the agent controls digital assets and can transact with them independently. This separates agents that read blockchain state from agents that

write to it with real consequences. An agent that monitors a liquidity pool without the ability to rebalance it unilaterally is an observer rather than an economic actor.

The third is *on-chain interaction*: the agent's behaviour depends on blockchain state as input, commits its consequential outputs to a blockchain, or is governed by on-chain contracts, such that removing the blockchain component would fundamentally alter the agent's operating mode or accountability structure. This condition excludes systems that use a blockchain purely as an incidental audit log while acting through centralised infrastructure, and admits off-chain reasoning with on-chain settlement, which is the dominant architectural pattern in current deployments.

Several system types fall close to but outside this boundary. Standard smart contracts fail the first condition: they execute deterministically on fixed inputs and cannot adapt their behaviour across interactions without an external call. Cloud-hosted AI systems that read blockchain data but act through centralised exchanges fail the third condition in the meaningful sense, since their consequential actions leave no on-chain record. Human-supervised systems that require approval above a spending threshold occupy a boundary position; we include them where the autonomous action space is otherwise non-trivial and the approval gate is the exception, not the rule.

These three conditions are not arbitrary. Each maps onto a design dimension addressed in subsequent sections. Operational autonomy shapes the governance problem (Section 5). Economic autonomy determines which coordination mechanisms are viable (Section 4). On-chain interaction defines the execution architecture options (Section 3).

2.4. Survey Methodology

This survey is guided by three research questions that reflect its three analytical dimensions:

- **RQ1 (Architecture):** What architectural patterns have emerged for deploying AI agents on or through blockchain infrastructure, and what are their respective trade-offs in verifiability, computational expressiveness, and latency?
- **RQ2 (Coordination):** Through what mechanisms do decentralised AI agents coordinate their actions, and how do these mechanisms perform across communication overhead, incentive alignment, and trust assumptions?
- **RQ3 (Governance):** How are decentralised AI agent systems governed, and where do existing technical and regulatory mechanisms fall short?

The current survey covers publications from January 2019 to April 2026. The lower bound reflects the period when smart contract platforms with sufficient expressiveness for non-trivial agent logic became widely available; the bulk of relevant work appears from 2022 onward, coinciding with the maturation of Layer 2 infrastructure and the first production deployments of autonomous on-chain agents.

We searched four databases: IEEE Xplore, the ACM Digital Library, Scopus, and arXiv. Search strings combined *blockchain* or *distributed ledger* with *multi-agent system*, *autonomous agent*, or *AI agent*, paired with at least one of *coordination*, *governance*, *smart contract*, or *reinforcement learning*. We additionally conducted targeted searches for deployed systems by name (e.g., Autonolas, Fetch.ai, SingularityNET, Giza, Bacalhau) and included technical documentation and whitepapers where peer-reviewed coverage was absent but the system was substantive enough to warrant inclusion.

Initial search resulted in 463 records across the four databases: 51 from IEEE Xplore (two query passes), 102 from the ACM Digital Library, 307 from Scopus, and 3 from arXiv. After removing 44 duplicates, 419 records remained for title and abstract screening. The relatively low cross-database overlap reflects the fact that the four queries used substantially different syntax to work around each database's proprietary search grammar, producing

partially non-overlapping result sets despite targeting the same conceptual intersection. Papers were excluded if they used *agent* purely in the economic rationality sense without any computational agent behaviour, if blockchain and AI appeared in unrelated sections of the same paper without integration, or if they were workshop papers under four pages without a corresponding full publication. Of the 187 records assessed at full text, 177 peer-reviewed publications met all three inclusion conditions. The final corpus comprises these 177 publications and 14 system documentation sources. The PRISMA flow diagram summarising the selection process is shown in Figure 1. Full search strings, inclusion and exclusion criteria, and coding rules are provided in Appendix A.

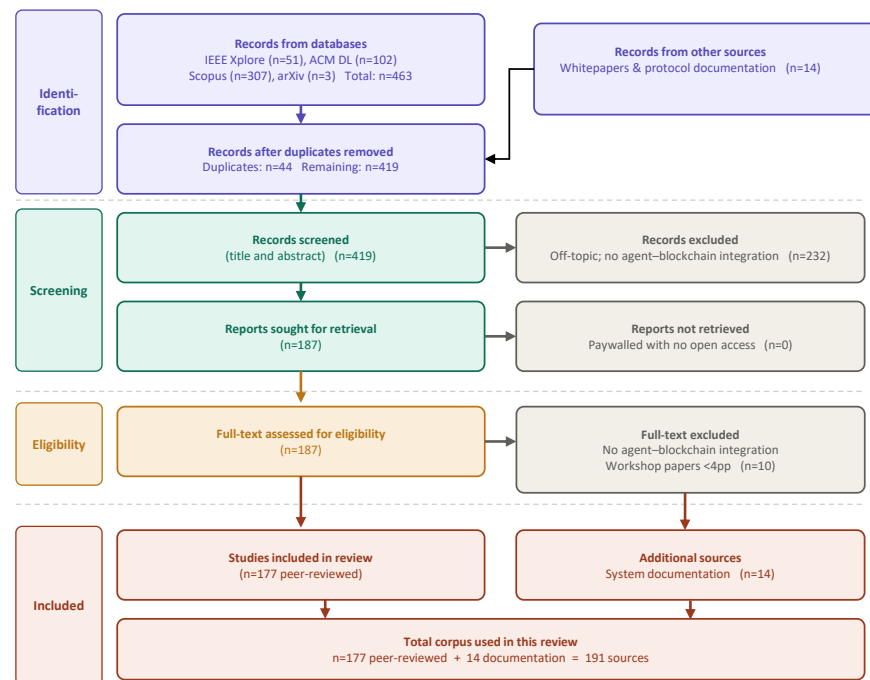


Figure 1. PRISMA 2020 [32] flow diagram. Of 463 identified records, 419 remained after removing 44 duplicates; 177 peer-reviewed publications met the inclusion criteria alongside 14 documentation sources (total: 191).

3. Architectures for Decentralized AI Agents

This section organises the architectural space along a single axis: where the agent's computation runs and what guarantees that gives about its behaviour. Four patterns emerge from the literature. On-chain execution places all agent logic inside a smart contract, making behaviour verifiable by construction but severely constrained by gas metering and the absence of floating-point arithmetic. Off-chain agents with on-chain settlement remove the computation constraint at the cost of verifiability, relying instead on economic mechanisms and reputational accountability. Verifiable off-chain computation, attempts to recover some verification guarantees without returning to on-chain execution. Multi-agent systems introduce coordination structures on top of any single-agent architecture, with shared contract state and atomic transaction pipelines as the distinguishing properties.

Figure 2 shows how these components fit together in a complete system, from external data sources through oracle and bridge layers to the AI agent core, and from there through the blockchain layer to verification, coordination, and governance.



Figure 2. High Level architecture of a decentralised AI agent system across six layers: external data, oracle/bridge, AI agent core (RL/MARL, LLM, rule-based, ERC-4337), blockchain, verification, and coordination/governance. Dashed blue arrows = data reads; solid green = transaction settlement; dashed coral = governance control.

3.1. On-Chain Execution Architectures

The simplest architectural pattern places all agent logic inside a smart contract. The agent's policy is encoded as contract code; it executes whenever triggered by a transaction or an on-chain event, reads state from the chain, and writes its output back as a state transition. No off-chain component is required.

The appeal is verifiability. Because the computation runs inside the EVM or an equivalent runtime, every step is reproducible by any node in the network. The agent's behaviour is auditable in principle by anyone with access to the contract bytecode and transaction history. In decentralised systems where counterparties cannot be assumed to trust each other, this property has real practical value that more capable off-chain architectures cannot easily replicate.

The constraints are severe. EVM computation is metered by gas, capping the complexity of logic that can run in a single transaction. As of 2024, the Ethereum mainnet block gas limit permits approximately 30 million gas units per block, sufficient for arithmetic, storage reads, and simple conditional branches, but not for floating-point inference over a neural network of any practical size [33]. Contract storage is also expensive; reading a 32-byte word costs 2100 gas cold and writing costs 20,000 gas [34]. Agents that need to maintain large learned state across interactions will find this cost structure prohibitive.

Native randomness is unavailable without an external oracle, which rules out stochastic policies unless the randomness source is brought on-chain through a verifiable random function such as Chainlink VRF [35].

The systems that operate within these constraints are not trivial, but they are rule-based, not adaptive. Automated market makers such as Uniswap v3 [36] implement concentrated liquidity algorithms entirely on-chain, adjusting price ranges and fee tiers through contract logic that responds to swap activity. Liquidation agents in lending protocols such as Aave [37] and Compound [38] monitor collateral ratios and execute liquidations when thresholds are crossed, with trigger logic encoded in the contract and execution incentivised by liquidation bonuses paid to the calling account. MakerDAO's keeper system [39] follows the same pattern for collateral auctions. These agents are operationally and economically autonomous, but their policies are fixed at deployment. On-chain execution architectures are therefore best understood as the home of threshold-based and rule-based agents. Work on bringing ML inference on-chain through zero-knowledge proofs is at an early stage and is discussed in Section 3.3.

Beyond purely financial applications, on-chain logic has been applied to blockchain network management itself. Skowroński [40] demonstrates a DRL agent that adapts mining threat mitigation in proof-of-work chains, encoding detection logic in smart contract state transitions. Chitra et al. [41] use agent-based simulation to model the incentive dynamics of blockchain consensus protocols at the network level, showing that individually rational agents can produce system-level instabilities not predicted by static game-theoretic analysis. On the permissioned side, Li et al. [42] and Zhang and Chen [43] address parameter tuning for Hyperledger Fabric and similar systems through RL agents that observe throughput and latency metrics and adjust block size, endorsement policy, and batching parameters on-chain. Halasz et al. [44] formalise the trust requirements for Smart Agents executed on-chain to coordinate autonomous systems in Smart Cities, proposing a blockchain-enforced execution substrate as the mechanism for guaranteeing coordination safety properties.

3.2. Off-Chain Agents with On-Chain Settlement

The dominant architectural pattern in deployed systems separates reasoning from settlement. The agent runs off-chain, such as on a conventional server, a containerised cloud process, or edge hardware, and commits only its outputs to the chain; a signed transaction, a service delivery attestation, a state update, or a governance vote.

This pattern removes the computation constraint. Off-chain, an agent can run inference on arbitrarily large models, maintain rich learned state, and act at sub-second latency. The cost is verifiability. No network node can confirm that the off-chain computation was performed correctly, by the declared agent, or at all. Trust in agent behaviour is currently mediated by one of three mechanisms: the deployer's reputation, a staking and slashing arrangement that makes misbehaviour financially costly, or monitoring by a third-party watchdog agent. None of these is a cryptographic guarantee.

The pattern subdivides by how the off-chain agent interfaces with the chain. In the event-driven variant, the agent subscribes to on-chain events, including a new block, a price crossing a threshold, or a task posted to a registry contract, and responds by submitting a transaction. Keeper networks, standardised by Chainlink Keepers [45] and Gelato [46], formalise this pattern: a set of registered off-chain agents compete to respond to trigger conditions, with rewards paid to the first valid responder. The competitive structure incentivises low-latency infrastructure, which suits reactive applications but tends to produce redundant computation when multiple agents race for the same trigger.

In the strategy vault variant, common in DeFi, the agent implements a long-running investment or rebalancing strategy. It reads on-chain prices and balances, computes an

allocation off-chain (using optimisation, RL, or a rule set) and executes rebalancing through on-chain transactions. Yearn Finance vaults [47] are the most widely deployed example, though their strategy logic is rule-based; several subsequent systems have replaced the rule layer with trained RL policies [48]. The agent's off-chain computation is opaque to the chain; its correctness is assessed retrospectively through the vault's on-chain performance record.

LLM-based agents with wallet signers represent a newer instance of the same pattern. The agent uses a language model to plan and reason (interpreting task descriptions, decomposing goals, selecting tools) and executes on-chain actions through a connected wallet. Autonolas [1] and Fetch.ai [2] provide production frameworks for agents of this type, with identity registered on-chain and inter-agent communication handled through a combination of on-chain registries and off-chain messaging. The language model introduces non-determinism absent from rule-based or RL agents: two invocations with the same input may produce different outputs, which complicates any form of post-hoc audit.

A concern specific to this pattern is front-running. When an agent's intended transaction is visible in the mempool before block inclusion, other agents can observe it and submit competing transactions with higher gas fees to execute first. This is the basis of the maximal extractable value (MEV) problem [49]. It affects any off-chain agent whose strategy depends on relative execution order. Mitigations include private transaction relays such as Flashbots Protect [50], commit-reveal schemes, and threshold encryption of transaction content, each with different latency and complexity trade-offs.

The off-chain agent with on-chain settlement pattern appears across a wide range of application domains in the corpus, and its prevalence is not uniform. Energy systems account for the largest single cluster. Shen et al. [51] established the foundational pattern for peer-to-peer microgrid trading: a DRL agent computes allocation decisions off-chain and commits settlement transactions to a permissioned blockchain, using the ledger for auditability and dispute resolution, not computation. Subsequent work has extended this to more complex market structures, including multi-party demand response [52] and cross-border prosumer coordination [53], but the architectural template has remained stable: off-chain policy, on-chain settlement, staking for compliance enforcement.

Resource allocation in mobile edge computing and UAV networks forms the second major cluster. The common structure is an off-chain MARL system whose joint policy is trained centrally on historical chain data, with individual agents executing at edge nodes and committing resource allocation decisions through smart contract escrow. Seid et al. [54] provide a representative example in 5G-RAN, and the same architecture recurs in multi-domain 6G networks [55] and multi-UAV task offloading [56]. The use of smart contract escrow for payment is the key difference from conventional MARL: it removes the need to trust the resource provider to execute and pay correctly, which matters when participants are from different organisations or administrative domains.

Blockchain network self-management is a smaller but analytically distinct sub-pattern. Here the agent's environment *is* the chain: Li et al. [42] deploy an RL agent that observes Hyperledger Fabric telemetry and adjusts block size and endorsement policy through governance transactions, while Xing [57] applies the same approach to permissionless sharding reconfiguration. The feedback loop between agent and environment is tighter than in any other application cluster; the agent modifies the infrastructure it runs on, which introduces stability considerations not present in external control applications.

Vehicular and IoT deployments follow the same off-chain/on-chain split, primarily for path planning, caching, and task offloading [58–60], with blockchain providing cross-organisational settlement and tamper-evident logging. The Davarakis et al. system [61] is notable for making the coupling explicit: an RL agent's state space, action space, and reward

signal are all defined by on-chain contract state, so the blockchain is the environment in which the agent operates, not just a settlement layer.

3.3. Verifiable Off-Chain Computation

The architectural gap between on-chain and off-chain patterns is a trust gap. On-chain execution is verifiable by construction; off-chain execution is trusted by necessity. Closing this gap without requiring the agent's computation to run inside the EVM has been attempted through zero-knowledge proofs, trusted hardware, and optimistic dispute mechanisms.

Zero-knowledge machine learning (ZKML) applies zero-knowledge proof systems to ML inference, producing a proof that a given model was evaluated correctly on a given input without revealing model weights or intermediate computation. The verifier, typically a smart contract, checks the proof rather than re-running the inference. Giza [4] and EZKL [62] are the main production-oriented implementations; both target the STARK proof system for its transparency and avoidance of a trusted setup. Modulus Labs [63] benchmarked proving cost across a range of model sizes and found that a 2.8-million-parameter model required approximately 2.5 min of proving time on a high-end GPU, with proof size on the order of a few hundred kilobytes. Proving time scales poorly with model size under current circuit compilation approaches, which limits ZKML to small models (e.g., classifiers, price predictors, lightweight policy networks). It is currently most plausible for verifying constrained RL policies, not LLM-based agents, where model size is orders of magnitude larger.

Trusted execution environments (TEEs) take a hardware-level approach. A TEE (e.g., Intel SGX, AMD SEV, or ARM TrustZone) is a hardware-isolated compute region where code runs in a protected enclave, shielded from the host operating system. Remote attestation allows an external verifier to confirm that a specific program is running inside a genuine TEE on unmodified hardware [64]. Applied to agent systems, an on-chain contract can receive a hardware-signed attestation that an off-chain agent ran a declared policy without interference. TEEs are computationally capable enough to run substantial ML inference, and the trust assumption shifts from the software deployer to the hardware manufacturer; a narrower surface, though not an unconditional one. Several oracle networks have adopted TEE-based computation for sensitive data feeds [19]. Known side-channel vulnerabilities in Intel SGX, including Spectre and Foreshadow, have been disclosed and partially mitigated, but the hardware trust assumption remains a point of concern for high-stakes agent deployments [65].

Optimistic verification does not attempt to prove correct computation upfront. It assumes the computation is correct and allows a challenge window during which any party can submit a fraud proof. If no valid challenge arrives before the window closes, the result is accepted. This mechanism underlies Arbitrum's dispute resolution [8] and has been proposed for general off-chain computation by Truebit [66] and for agent-specific workloads by Cartesi [67]. The latency cost is the challenge window, typically several days on current deployments. Optimistic verification is unsuitable for time-sensitive agent actions but appropriate where finality can be deferred: batch settlement, periodic re-balancing, and governance proposal evaluation.

The security of smart contracts as execution environments for agent logic has received growing attention as an adjacent verification problem. Andrijasa et al. [68] survey the application of multi-agent Deep Reinforcement Learning (DRL) to smart contract fuzzing, in which agents act as adversarial testers probing contract logic for re-entrancy and integer overflow vulnerabilities. This is a case where the blockchain is the subject of verification rather than the verifier; agents act off-chain to find failures in on-chain code. Phiri [69] formalises eight auditability axioms for agentic AI systems, including integrity, temporal

coherence, and attribution, and proposes blockchain as the natural substrate for satisfying them, since immutable transaction logs provide the permanent, tamper-evident record that auditability requires. Yoo et al. [70] address a related gap in federated learning pipelines, where off-chain agent contributions must be attributed and verified for incentive purposes, and propose a DLT-based credentialing architecture that provides verifiable agent identity without requiring full ZK proofs.

Table 1 summarises the three approaches across the dimensions most relevant to agent design.

Table 1. Comparison of verification approaches for off-chain agent computation (production deployment status as of April 2026). *Proof latency*: time to generate the verification artefact; *Settlement latency*: time until on-chain acceptance. For ZKML these two quantities can differ by orders of magnitude.

Approach	Comp. Overhead	Model Size	Proof Latency	Settlement Latency	Trust
ZKML	Very high	Small only	Seconds–minutes	Seconds	Cryptographic
TEE	Low	Moderate–large	Milliseconds	Seconds	Hardware mfr.
Optimistic	Low	Arbitrary	None (assumed)	Days	Economic

3.4. Multi-Agent Systems On-Chain

The preceding subsections treat agents in isolation. Many deployed and proposed systems involve multiple agents, interacting competitively, cooperatively, or both. The on-chain environment creates coordination properties that do not exist in conventional multi-agent systems; agents can read shared contract state atomically, commit to actions in a verifiable and binding way, and transfer value without a trusted intermediary.

Agent registries are the entry point for most multi-agent deployments. An on-chain registry records each agent’s identifier, declared capabilities, service endpoint, and staking position, giving other agents and clients a verifiable directory from which to discover counterparties. Fetch.ai’s Almanac contract [2] and the Autonolas service registry [1] are the most widely deployed examples. The registry pattern creates a first coordination layer: agents announce their capabilities, and other agents or contracts query this without relying on a centralised directory operator.

Task markets extend this by providing a protocol for agents to post work, receive bids from other agents, and settle payment on verified delivery. A requesting agent specifies a task and a payment ceiling; bidding agents submit offers; the contract awards the task and holds payment in escrow until the task output is submitted and verified. Bacalhau [71] implements this pattern for distributed computation tasks, with a network of worker agents bidding to execute jobs posted by client agents. The verification step varies by implementation: some systems use cryptographic output hashes, others use optimistic challenge periods, and some rely on reputation scores accumulated over prior interactions.

Multi-agent pipelines, where the output of one agent feeds the next, appear in DeFi sequential strategy execution and in agent service frameworks where an LLM planner delegates subtasks to specialised execution agents. On-chain, a pipeline can be encoded as a sequence of contract calls within a single transaction, giving it atomicity: either all steps complete or none do. This guarantee, unavailable in conventional multi-agent middleware without a trusted coordinator, is one of the more useful structural properties the blockchain environment provides. The cost is gas; each step adds to the transaction cost, and long pipelines require careful budgeting even on Layer 2 networks.

MARL in shared on-chain environments has properties that distinguish it from standard simulation-based MARL. The environment state, including contract balances, oracle prices, and pending transactions, is observable by all agents simultaneously and without additional cost, removing one common source of partial observability. At the same time, the block-by-block time structure means agents cannot act continuously: they submit transactions during a block and observe outcomes only after the block finalises. Chain reorganisations, where a confirmed block is replaced by a competing fork, introduce environment non-stationarity with no equivalent in simulation. The implications for MARL convergence are examined in Section 7.

The corpus contains several multi-agent architectures that extend the patterns above. Yang et al. [72] propose a general smart agent architecture on Ethereum in which agent logic is decomposed into modular on-chain components, establishing an early template for how classical MAS design patterns translate into the EVM environment. At the other end of the complexity spectrum, Chen et al. [73] build AgentChain, a blockchain-coordinated LLM multi-agent system for trustworthy question-answering, in which blockchain enforces attribution and tamper-evidence across agent contributions. The LearnChain system [74] is notable for its explicit treatment of cooperative RL on-chain: agents share learning updates via smart contracts, using blockchain both to coordinate training and to create an auditable record of each agent's contribution to the joint policy. ManuChain II [75] and the blockchained smart contract system of Leng et al. [76] demonstrate Multi-Agent Systems (MAS) coordination for manufacturing pipelines, where smart contract pyramids encode hierarchical process control with verifiable state transitions. Smart contract security in MAS contexts is addressed by Praitheeshan et al. [77], whose SolGuard system detects external call vulnerabilities in multi-agent robotic systems that use smart contracts as coordination middleware. Pakana [78] takes a semantic approach, proposing that smart contracts can serve as semantic signalling mechanisms for AI agents, enabling interoperability across heterogeneous agent platforms without requiring shared representation formats.

3.5. Summary Taxonomy

Table 2 classifies the systems discussed in Section 3 along four dimensions: execution locus, verification mechanism, learning paradigm, and coordination protocol. These dimensions are not fully independent; execution locus constrains verification options, and the learning paradigm constrains viable coordination protocols. Treating them separately, however, still supports comparison across systems that differ along multiple axes simultaneously.

The taxonomy supports a few observations. The verification column is dominated by trusted deployer and staking entries, reflecting the current gap between the verifiability ideal and the practical cost of cryptographic alternatives. Coordination protocols are also absent from most on-chain and verifiable off-chain systems; multi-agent coordination is predominantly an off-chain concern settled on-chain. Learning paradigm diversity, meanwhile, is concentrated in off-chain systems; on-chain agents are almost entirely rule-based, for the reasons given in Section 3.1.

A caveat on orthogonality: the four dimensions in Table 2 are analytically distinct but not fully independent in practice. The verification mechanism is constrained by the execution locus: on-chain execution implies chain consensus verification by construction, while off-chain execution admits the full range of verification options. The coordination protocol is constrained by both: token incentive mechanisms require on-chain settlement, while gossip-based coordination can operate independently of the chain. Several systems in the corpus occupy ambiguous positions. Fetch.ai's architecture spans off-chain execution and on-chain settlement with both staking-based and auction-based coordination components, making single-column classification an oversimplification. Autonolas similarly combines staking,

on-chain logging, and auction mechanisms in a layered stack. The entries in Table 2 reflect the dominant or most distinctive property of each system, not its full architectural description.

Table 2. Taxonomy of representative decentralised AI agent systems (selected). Rule-based systems are lower-boundary reference points. Coord. = coordination; n/a = not applicable.

System	Execution Locus	Verification	Learning	Coord.
Uniswap v3 [36]	On-chain	Chain consensus	Rule-based	n/a
Aave liquidator [37]	Off-chain + settle	Trusted deployer	Rule-based	Token incentive
Compound [38]	Off-chain + settle	Trusted deployer	Rule-based	Token incentive
Yearn vault [47]	Off-chain + settle	Trusted deployer	Rule-based/RL	Token incentive
Yang et al. [72]	On-chain	Chain consensus	Rule-based	n/a
SquirRL [79]	Off-chain + settle	Trusted deployer	RL	n/a
Nguyen et al. [15]	Off-chain + settle	Trusted deployer	MARL	Auction
Chainlink Keepers [45]	Off-chain + settle	Trusted + staking	Rule-based	Auction
Gelato [46]	Off-chain + settle	Trusted + staking	Rule-based	Auction
Fetch.ai [2]	Off-chain + settle	Staking	RL/LLM	Auction
Autonolas [1]	Off-chain + settle	Staking	LLM/hybrid	Auction + token
SingularityNET [3]	Off-chain + settle	Reputation + staking	Various	Token incentive
Bacalhau [71]	Off-chain + settle	Optimistic	Various	Auction
Giza [4]	Verifiable (ZK)	ZK proof	RL/supervised	n/a
TEE oracle [64]	Verifiable (TEE)	TEE attestation	Rule-based	Token incentive
Davarakis et al. [61]	Off-chain + settle	Trusted deployer	RL	Token incentive
Seid et al. [54]	Off-chain + settle	Staking	MARL	Auction
APOLLO [80]	Off-chain + settle	Staking + on-chain	RL	Token incentive
LearnChain [74]	Off-chain + settle	On-chain log	Cooperative RL	Token incentive
AgentChain [73]	Off-chain + settle	On-chain log	LLM multi-agent	Token incentive

4. Coordination Mechanisms

Coordinating decentralised AI agents across a shared blockchain infrastructure requires resolving a set of problems that do not arise in conventional distributed systems. Agents must allocate tasks, enforce compliance, and adapt to changing conditions, all without a trusted central authority, and within the hard constraints that on-chain execution imposes; transaction costs limit interaction frequency, block-time discretisation breaks the synchrony assumptions underlying standard coordination theory, and smart contract logic must be self-enforcing by construction. This section examines four coordination mechanisms through which deployed systems have addressed these problems: auction-based task allocation, multi-agent reinforcement learning, token incentive design, and federated or gossip-based coordination. The four are not alternatives in the strict sense; production systems typically combine elements from more than one. The analysis in the current section proceeds mechanism by mechanism while Section 4.4 draws a comparative assessment across communication overhead, trust assumptions, incentive alignment, and latency tolerance.

4.1. Auction-Based Coordination

Auctions are a natural fit for on-chain task allocation. A requesting agent posts a task specification and a payment ceiling to a contract; candidate agents bid to fulfil it; the contract clears the auction and holds payment in escrow until verified delivery. The entire protocol executes without a trusted auctioneer, which is the primary reason auction mechanisms have become the most common coordination pattern in agent task markets.

The choice of auction format has consequences beyond price discovery. First-price sealed-bid auctions, where the winning bidder pays their bid, are simple to implement but incentivise bid shading: agents submit offers below their true valuations to preserve surplus, and the resulting equilibrium is allocatively inefficient. Second-price (Vickrey) auctions recover truthful bidding as a dominant strategy, since the winner pays only the second-highest bid [81]. The Vickrey–Clarke–Groves (VCG) mechanism generalises this to multiple simultaneous items and combinatorial bidding, achieving allocative efficiency and incentive compatibility under quasi-linear preferences [82,83]. On-chain VCG implementation is constrained by gas: clearing a combinatorial VCG auction is NP-hard in general, and even polynomial-time approximations become expensive at scale. Current implementations either restrict the combinatorial structure to single-minded bidders or process auction clearing off-chain and commit only the result to the chain, trading verifiability for tractability.

English auctions (ascending price with open bids) sidestep this complexity and map naturally to the block-time structure. Bids are submitted in successive blocks; the auction closes when no higher bid arrives within a timeout. MakerDAO's collateral auctions [39] use this format. Dutch auctions (descending price) are used in contexts where settlement speed matters more than revenue maximisation, including several DeFi liquidation mechanisms.

A subtler problem in on-chain combinatorial auctions is the exposure problem [84]: an agent bidding on a bundle of complementary tasks may win only a subset, leaving it holding items of lower value than it paid. Two concrete smart-contract patterns address this, each with distinct gas and latency trade-offs.

The first is *conditional batch settlement*. All bids are collected in a commitment round spanning one or more blocks; the contract then clears all bids atomically in a single settlement transaction, either awarding the full bundle or rejecting it entirely. Because clearing logic runs on-chain, it is verifiable and tamper-evident. The cost is gas: clearing even a modestly sized combinatorial auction (e.g., 20 tasks, 10 agents) can consume 2–5 million gas units in a single transaction [85], approaching 10–20% of Ethereum's mainnet block gas limit. This restricts the mechanism to small auctions or to Layer 2 networks where gas is cheaper by one to two

orders of magnitude. The latency to settlement is bounded by the commitment round length plus one block for clearing, typically 15–60 s on current Layer 2 deployments.

The second pattern is *escrow with conditional release*. Each bid is submitted as a separate transaction locking the bid value in an escrow contract; a separate resolution contract aggregates outcomes and releases funds only when the full bundle condition is satisfied. This spreads gas cost across multiple transactions and avoids the block gas limit ceiling, but introduces a window between individual bid submission and bundle resolution during which an agent's funds are locked and unavailable. The window can span multiple blocks, creating latency of 30 s to several minutes depending on network congestion. Front-running during this window is a real attack surface: an adversary who observes partial bids in the mempool can submit competing bids to disrupt bundle formation before the resolution contract executes [86].

4.2. Multi-Agent Reinforcement Learning

MARL in blockchain environments inherits the standard challenges of the field and adds several that are specific to the execution context. Standard MARL settings assume agents observe the environment and act at a fixed frequency, the environment transitions on all agents' joint actions, and rewards are delivered at each step. None of these assumptions holds cleanly on-chain.

Block time introduces discrete, variable-length intervals: agents submit transactions during a block and observe outcomes only at finalisation, typically every 2–12 s depending on the network. Between blocks, no action is possible. This turns the MARL problem into an asynchronous repeated game with variable delays rather than a standard Markov game. Convergence proofs for CTDE algorithms such as QMIX [26] and MADDPG [27] assume synchronous step transitions; their validity under block-time discretisation is an open question addressed in Section 7.

Non-stationarity in blockchain MARL takes two distinct forms that should not be conflated. The first is *agent-induced* non-stationarity: the standard MARL problem where the environment each agent faces changes because other agents are learning concurrently. CTDE partially addresses this by conditioning on joint information during training while restricting agents to local observations at execution time. In blockchain environments, joint information is available directly from chain state, which is public and consistent across all participants, so CTDE training can use verified on-chain data rather than reconstructed joint state, a genuine advantage over partially observable simulation environments. The second form is *environment-induced* non-stationarity from chain reorganisations: recently finalised blocks are replaced by competing forks, retroactively changing the state that agents observed and acted on. This source of non-stationarity is qualitatively different from the first and is not addressed by CTDE or by any existing asynchronous MARL approach. Its implications for convergence are discussed in Section 7.

The severity of environment-induced non-stationarity depends critically on the consensus mechanism's finality model. *Nakamoto-style* consensus (used by Bitcoin and pre-Merge Ethereum) achieves only probabilistic finality: a block is considered final after a sufficient number of successor blocks build on it, but this guarantee is statistical rather than absolute, and deep reorganisations remain possible in theory. Under Nakamoto consensus, an agent must treat any recently observed state as potentially reversible, which requires either deferring action until sufficient confirmations accumulate (adding latency of minutes to hours) or accepting that some fraction of committed actions may be retroactively undone. Tendermint-style Byzantine Fault Tolerant (BFT) consensus (used by Cosmos, Hyperledger Fabric, and similar systems) achieves deterministic finality within a single round: once a block is committed, it cannot be reorganised. For MARL agents, this eliminates

environment-induced non-stationarity entirely. The state observed after block commitment is guaranteed to persist. The trade-off is throughput and decentralisation: BFT consensus requires $O(n^2)$ message complexity and a known, bounded validator set, which limits the number of participants and introduces a different trust surface. Ethereum post-Merge occupies a middle position: its PoS consensus achieves economic finality after approximately 12.8 min (two epochs), providing a much smaller reorganisation window than Nakamoto consensus but stopping short of the single-round finality guarantee of BFT protocols. System designers choosing a blockchain substrate for MARL agent deployment should treat the finality model as a primary design parameter: BFT chains are preferable where convergence guarantees are required; Nakamoto chains require agents to be designed with explicit state-reversal tolerance.

Token-denominated rewards introduce a property absent from simulation-based MARL: the reward function is on-chain and verifiable. An agent's reward is a token transfer recorded on the ledger, so reward disputes can be adjudicated by the contract rather than by the system designer. Reward hacking takes the form of on-chain manipulation instead of simulation artefacts, which means it produces real transactions with real economic consequences—a deterrent that simulation environments cannot replicate.

Credit assignment in cooperative MARL maps directly onto an economic problem in token-incentivised systems. When a group of agents jointly produces an outcome, the contract must distribute payment in proportion to each agent's marginal contribution. The Shapley value [87] provides a theoretically grounded attribution, but its exact computation is exponential in the number of agents. Practical systems use approximations or restrict contribution structures to admit simpler attribution rules, accepting some fairness loss in exchange for on-chain tractability.

The surveyed corpus contains a large and growing body of MARL deployments in blockchain-enabled distributed systems. Energy is the dominant application domain: studies address peer-to-peer microgrid trading [51,53,88,89], demand response [52,90–93], vehicle-to-grid coordination [94], and virtual power plant operation [95], with MADDPG and QMIX the most frequently used algorithms. Task offloading in mobile edge and UAV networks is the second largest cluster [15,56,96–101], sharing the architectural pattern of centralised training on historical chain data followed by decentralised execution at edge nodes. Consensus mechanism design using MARL is a smaller but distinct category: Islam et al. [102] propose MRL-PoS, a MARL-based proof-of-stake consensus algorithm in which validators learn bidding strategies, and Bappy et al. [103] apply MARL to detecting and mitigating malicious nodes in PoS chains. Cross-chain optimisation using MARL is addressed by Daliri and Jassbi [104], who propose a customised multi-agent architecture that handles heterogeneous blockchain networks with different consensus rules and latencies, providing one of the few empirical treatments of the cross-chain coordination problem identified in Section 7.

Token incentives are the primary coordination tool in decentralised agent systems where cryptographic guarantees are absent or too costly to compute. Rather than proving correct behaviour, token mechanisms impose a financial cost on misbehaviour and a financial reward on correct behaviour, relying on the assumption that agents are rational and will not deviate when the expected penalty exceeds the expected gain.

Staking and slashing form the core of this approach. An agent deposits tokens into a contract as collateral before participating in the system. If a verifiable protocol violation is subsequently detected, such as signing two conflicting messages, missing a required response window, or submitting an output that fails an on-chain check, the contract destroys some or all of the deposit. The deterrence effect depends on three parameters: the stake size relative to the gain from misbehaviour, the probability of detection, and the opportunity

cost of capital locked as collateral. Systems where misbehaviour is hard to detect, or where the gain from a single deviation exceeds the stake, will experience equilibrium violation despite the mechanism.

Reputation tokens accumulate from past behaviour and provide a longer-horizon coordination substrate. Non-transferable reputation tokens (e.g., soulbound tokens [105]) cannot be purchased, meaning they track actual agent history, not financial position. Token-Curated Registries (TCRs) [106] extend this by allowing any participant to challenge a registry entry by staking tokens on the challenge, with disputes resolved through a vote weighted by stake. The result is a decentralised quality filter in which participants with accurate judgement about registry quality are financially rewarded.

A structural tension in token mechanism design is Goodhart's Law [107]: when a measurable outcome becomes the reward target, agents optimise for the measure rather than the underlying objective. In agent systems this appears concretely. An agent rewarded for transaction volume will submit low-value transactions. One rewarded for uptime will remain online without performing useful work. Mechanism design responses, namely output-based bonding, mutual monitoring, and commit-reveal schemes that tie rewards to ex-post verified outcomes, are an active research area [108] with direct implications for how agent systems are designed in practice.

Tian [109] provides an empirical treatment of incentive-compatible mechanism design for MARL systems specifically, demonstrating that misalignment between token reward structures and agent learning objectives produces predictable exploitation patterns. Tang et al. [110] show that proof-of-stake incentive structures can be adapted to reward data collection agents in UAV-IoT systems, using staking as both a commitment device and a quality signal. Fernandez et al. [111] use agent-based simulation to analyse how initial token distribution policies in post-fair-launch blockchain networks affect long-term governance concentration; an important empirical input for the governance analysis in Section 5. Park et al. [112] extend this to show that blockchain token economics can inadvertently amplify economic inequality among network participants, with implications for the design of agent incentive structures in public protocols.

4.3. Federated and Gossip-Based Coordination

Not all coordination requires on-chain settlement. Where transaction costs or latency make frequent on-chain interaction impractical, agents coordinate through peer-to-peer messaging and commit only final outcomes to the chain.

Gossip protocols, the communication substrate of most peer-to-peer blockchain networks, propagate messages across a node network without centralised routing. Each node forwards received messages to a random subset of its peers; information reaches the full network in $\mathcal{O}(\log n)$ steps with high probability [113]. Fetch.ai's agent messaging layer [2] and the libp2p stack used by Autonolas [1] both rely on gossip for agent-to-agent communication, with on-chain contracts handling only task posting, payment escrow, and dispute resolution. This keeps the bulk of coordination traffic off-chain while preserving verifiable settlement.

State channels allow two or more agents to conduct an arbitrarily long sequence of interactions off-chain, with the blockchain serving as a court of last resort. Agents open a channel by depositing funds into a contract, exchange signed state updates off-chain at will, and close the channel by submitting the final agreed state. Any party can challenge the closing state by submitting a more recent signed update during a dispute window. The result is near-instant finality and zero gas cost for all intermediate interactions. The Lightning Network [114] established this pattern for payment channels; generalised state channels [115] extend it to arbitrary agent computation. For agent pairs that interact

frequently (e.g., a planning agent and a designated execution agent) state channels reduce coordination cost by orders of magnitude relative to on-chain settlement of each step.

Federated learning with blockchain coordinates the specific case where agents need to train a shared model without sharing raw data [116]. Each agent trains locally and submits model updates to an aggregation contract; the contract records contributions and distributes token rewards proportional to update quality. The blockchain role is audit and incentive, not computation. Byzantine fault tolerance for federated learning under adversarial agents [117] applies here, with the additional property that the on-chain reward mechanism creates a financial incentive for honest gradient submission that pure federated learning frameworks lack.

The corpus contains several substantive implementations of blockchain-federated coordination. Chen et al. [118] develop BELIEFS, a blockchain-based federated intelligence system that combines attack-tolerant aggregation with on-chain reward distribution. Liu et al. [119] address the privacy and incentive dimensions jointly in a P2P energy trading system, using MADDPG with federated training and blockchain settlement to protect prosumer data while aligning contribution incentives. Wang et al. [120] extend this to asynchronous settings, where agents train on locally cached IoT data and commit updates at variable intervals, using blockchain to manage staleness and attribution without a central aggregator. Chai et al. [121] address the edge intelligence case, coordinating decentralised edge nodes through a blockchain-incentivised federated optimisation protocol with adaptive contribution weighting. Sankaran et al. [122] apply federated MARL to the blockchain consensus problem itself, demonstrating that the training and execution separation native to FL maps naturally onto permissioned blockchain governance where nodes cannot share raw state.

Decentralised AI agents use three distinct communication modes whose timing and cost profiles differ substantially. Table 3 summarises the key properties. State reads are the most frequent operation: an agent may query chain state hundreds of times per second at zero cost. Settlement transactions are comparatively rare but are the only mode that produces binding, cryptographically final outcomes. Gossip communication is used for high-frequency peer coordination where on-chain cost would be prohibitive, but any outcome that must be binding requires a subsequent settlement transaction.

Table 3. Comparison of the three communication modes available to decentralised AI agents.

Mode	Latency	Gas Cost	Finality	Typical Use
State read (eth_call)	10–50 ms (L2: <5 ms)	None	None (read-only)	Observe prices, balances, contract state before acting
Tx settlement	L1: ~12 s/block; L2: 0.25–2 s	Yes (variable)	Cryptographic; block-time dependent	Commit bids, transfer assets, trigger contract logic
Gossip/P2P	1–200 ms	None	None; requires on-chain settlement for binding outcome	Coordinate off-chain, share observations, negotiate

4.4. Comparative Analysis

Table 4 compares the four coordination mechanisms described in the previous subsections, across five dimensions. Communication overhead refers to the volume of on-chain or off-chain messages required per coordination episode. Latency tolerance indicates how sensitive the mechanism is to block-time delays and network latency. Incentive alignment strength reflects how well the mechanism's reward structure tracks the designer's intended objective under rational agent behaviour. The ratings reflect the authors' assessment of typical behaviour based on the surveyed literature. They are not derived from formal measurement; read them as a structured qualitative summary, not as empirical findings, drawn from the deployment classes described in Sections 4.1–4.3. Specific implementations may differ depending on mechanism design choices.

Table 4. Qualitative comparison of coordination mechanisms, synthesised from the surveyed literature. Ratings reflect typical behaviour and should not be interpreted as universal properties.

Mechanism	Comm. Overhead	Trust Assumption	Incentive Alignment	Latency Tolerance
Auction	Low (on-chain only)	Contract logic	High for allocation; weak for compliance	Low (block time bound)
MARL	High (training)	Shared policy	High; adaptive over time	Medium
Token incentive	Low	Economic	Medium; Goodhart risk	High
Gossip/channel	Very low	Off-chain peer	Low; no on-chain enforcement	Very high

The comparison supports a few observations. Auction-based and token-based mechanisms are complementary: auctions handle task allocation, while token mechanisms handle compliance enforcement, and most production systems combine both. Gossip and state channel coordination substantially reduce on-chain cost but shift the trust surface to off-chain infrastructure; network partitions, messaging failures, and counterparty availability are not addressed by the on-chain components. And Multi-Agent Reinforcement Learning (MARL) stands apart as the only mechanism that produces agents capable of adapting their coordination behaviour over time; the remaining three fix the coordination protocol at design time and rely on agents to optimise within it. This distinction becomes consequential in environments where task distributions, agent populations, and resource availability shift over the system's lifetime.

What the coordination analysis does not address is the question of legitimacy: whether the outcomes produced by these mechanisms are acceptable to the participants who did not design them, and whether the agents executing them can be held accountable when they are not. Token incentives can align individual behaviour with a specified objective, but they cannot by themselves determine whether that objective is the right one, who authorised it, or how it should change when circumstances shift. Auction mechanisms allocate tasks efficiently under their stated assumptions, but those assumptions are embedded in smart contract code by developers whose interests may diverge from those of the agents or users that the contract governs. MARL agents adapt their policies over time, raising the additional question of who monitors whether the adapted policy remains within acceptable

bounds. These are governance questions, and they require a different analytical frame from the one used in Section 4. Section 5 examines that frame.

Table 5 makes the interdependencies across all three dimensions explicit. Each cell states a concrete constraint: given an architectural pattern, what coordination options are available, and what governance mechanisms are implicated? The choice of execution architecture forecloses some coordination and governance options while enabling others; the table does not assert that all combinations in a row are deployed together.

Table 5. Interaction constraints across the three analytical dimensions: how execution architecture limits available coordination mechanisms and governance options.

Architecture	Available Coordination	Verification	Governance Implications
On-chain execution	Auction, token incentive; no MARL (gas prohibits repeated inference)	Chain consensus (automatic)	Deactivation by contract upgrade; accountability clear (deployer); DAO can modify parameters on-chain
Off-chain + on-chain settlement	All four mechanisms; MARL feasible off-chain with on-chain reward	Trusted deployer or staking; ZK/TEE optional add-on	Accountability gap: deployer identifiable but policy evolution unaudited; dual agency problem arises; DAO governs parameters but not agent pipeline
Verifiable off-chain (ZK/TEE)	Auction, token; MARL possible but proof generation per step is expensive	Cryptographic (ZK) or hardware (TEE); proof covers computation, not policy identity for LLMs	Narrows accountability gap for fixed-weight models; LLM policy commitment remains open (see Section 7); regulatory auditability improved but not complete
Multi-agent on-chain	Auction, token, shared contract state; no MARL training (gas)	Chain consensus for all agents jointly; atomic composition verifiable	DAO governs shared registry and payment rules; emergent collusion visible on-chain but detection logic must be pre-specified in contract

5. Governance of Decentralized AI Agent Systems

Sections 3 and 4 established how decentralised AI agents are built and how they coordinate. This section addresses a harder question: how are they governed, and by whom. The answer is incomplete in every domain examined. Section 5.1 frames the

structural problem: the accountability gap that decentralised architectures produce by design. Sections 5.2–5.5 examine the mechanisms that attempt to fill it, namely DAO-based governance, regulatory frameworks, incentive alignment, and organisational adoption conditions. Section 5.6 states the limitations of the present survey. Section 6 traces how these governance mechanisms play out across four application domains (decentralised finance, supply chain, agent marketplaces, and IoT) where the tensions between operational autonomy and governance accountability are most visible, closing with cross-domain observations drawn from the full corpus. Governance investment lags technical deployment throughout: the architectural choices made in Section 3 determine what an agent can do, and separately, what governance mechanisms are even available to constrain it.

5.1. The Governance Problem for Autonomous Agents

Governance of AI agents in decentralised systems splits into three overlapping problems. Governance *of* agents concerns who determines an agent's objectives, constraints, and operating parameters, and who can modify or terminate its operation. Governance *by* agents concerns autonomous agents participating in governance processes, such as voting, proposing protocol changes, or managing treasuries, on behalf of token holders. Governance *accountability* concerns who bears legal and ethical responsibility when an agent causes harm, and through what mechanism redress is obtained.

Traditional software governance allocates accountability through a relatively clear chain. A developer writes code, an operator deploys it, a user interacts with it, and liability for harm flows upward through that chain according to established software and product liability frameworks [123]. Autonomous AI agents disrupt this chain on two axes simultaneously. Their behaviour is not fully predictable at deployment because a learning agent's policy evolves through interaction, and an LLM-based agent's outputs depend on prompts and contexts the deployer could not have anticipated. The deployer in a decentralised system, moreover, may not be a legal entity at all. A DAO that governs an agent fleet may have no registered legal domicile, no identifiable human controller, and no conventional enforcement surface. The code is law, but only in the sense that the code executes; it does not follow that the code is answerable.

The space between an agent's actions and any identifiable responsible party is what we term the *accountability gap*. It surfaces wherever AI systems operate with significant autonomy, so the gap is not confined to blockchain systems, but it is especially acute in decentralised environments because the architectural property that removes the trusted intermediary also removes the natural locus of accountability. Organisations that deploy AI agents through DAOs are, in effect, externalising governance risk to a mechanism that was designed to resist central control, not to provide it.

From an information systems perspective, this creates what we term a *governance vacuum*: a multi-stakeholder system in which no participant has both the authority and the information necessary to govern the whole. The concept has antecedents in adjacent literatures. Matthias [124] identified a “responsibility gap” in autonomous systems where no human agent can be held accountable for machine behaviour. Danaher [125] extended this to algorithmic governance, Zittrain [126] described governance deficits in generative systems where downstream consequences exceed the original designer's control. The governance vacuum described here is a specific instantiation of these ideas in which the gap is structurally reproduced by the architecture, not produced by regulatory lag or technological surprise: decentralisation deliberately removes the central authority that would otherwise be the natural target of governance obligations. Each stakeholder, including developers, token holders, agents, users, and regulators, holds a partial view and partial authority.

5.2. DAO-Based Governance of Agent Systems

DAOs represent the most developed attempt to fill the governance vacuum described in Section 5.1. They do so by encoding governance rules in smart contracts and distributing governance authority across token holders, removing the need for a central operator while maintaining the capacity to update system parameters. Applied to agent systems, DAO governance controls the parameters under which agents operate, the conditions for agent deactivation, and the distribution of agent-generated revenue.

The dominant mechanism is token-weighted voting; holders of a protocol's governance token submit and vote on proposals that update contract parameters, allocate treasury funds, or modify protocol rules. Token-weighted voting has three well-documented structural failure modes. Participation rates in major DAO governance votes rarely exceed 10–20% of eligible tokens [127], producing decisions made by a small active minority that may not represent the full stakeholder base. Governance token concentration among early investors and founding teams also produces outcomes that consistently favour capital holders over protocol users [128], the same plutocratic dynamics that decentralised governance was intended to escape. And flash loan governance attacks allow an adversary to borrow a large token position for the duration of a single block, vote on a proposal, and repay the loan before the block closes, bypassing the assumption that governance power correlates with long-term stake. This vector has been exploited in practice; the Beanstalk protocol [129] lost approximately \$182 million in April 2022 to an attacker who used a flash loan to pass a malicious governance proposal and drain the treasury in the same transaction.

Several governance designs attempt to address these failure modes. Quadratic voting [130] weights votes as the square root of tokens held, reducing plutocratic concentration at the cost of introducing Sybil attack vulnerability, since an adversary who can create many identities can vote more cheaply. Conviction voting accumulates voting power over time, rewarding sustained engagement over reactive position-taking, which addresses low participation but not concentration. Optimistic governance defaults to no change and activates only when a proposal receives sufficient active opposition, reducing the gas cost of routine non-decisions. For agent-specific governance, time-locked parameter changes insert a delay between proposal approval and implementation, giving affected agents and users time to respond before a change takes effect.

From an organisational theory perspective [131], these mechanisms share a common limitation: they treat governance as a voting problem, not a legitimacy problem. Legitimacy in organisational governance requires two things: that decisions follow a specified procedure, and that the procedure itself is regarded as fair by those it affects [17]. Token-weighted voting satisfies the first requirement for token holders but has no mechanism for incorporating the preferences of users, counterparties, or third parties who bear consequences but hold no tokens.

Autonolas implements a service registry where agent service parameters are governed by holders of the OLAS token; services that fail quality checks can be removed by DAO vote [1]. SingularityNET's AGI token holders vote on which AI services are listed on the platform and under what terms [3]. Neither system has faced a governance attack targeting its agent fleet specifically, but both inherit the general vulnerabilities of token-weighted governance, and neither provides a mechanism for incorporating feedback from the end users who interact with the agent services.

Dispute resolution for individual agent interactions operates below the DAO level through on-chain arbitration protocols. Kleros [132] and Aragon Court [133] use economic incentives to recruit jurors who stake tokens on correct verdicts; jurors who vote with the majority receive a share of the losing side's stake. Applied to agent disputes, such as a client claiming a task was not completed or an agent claiming payment was withheld,

these mechanisms provide decentralised adjudication without a trusted third party. Their limitation is the juror pool problem: effective arbitration of technical disputes between AI agents requires jurors with specialised knowledge, and staking mechanisms that recruit economically rational generalists may not produce the right adjudicators for complex agent-behaviour questions.

The corpus illustrates several concrete implementations of DAO-based agent governance. Cu et al. [134,135] develop a decentralised autonomous agent system for fractional asset ownership governed by a DAO, with on-chain governance managing agent parameters and dispute resolution for asset management decisions. Lee et al. [136] examine the interaction between AI policy governance and blockchain-based voting in federated learning systems, showing that DAO governance structures can extend meaningfully to FL coordination decisions where the agents are gradient-submitting nodes. Valiente and Pavón [137] provide a formal ontology for DAOs that covers the agent interaction patterns most relevant to multi-agent coordination, offering a shared conceptual vocabulary that the field currently lacks. Virovets et al. [138] survey interaction modes between autonomous economic agents in DAO environments, distinguishing competitive, cooperative, and hierarchical patterns and their governance implications. Ahmed et al. [80] develop APOLLO, an autonomous on-chain learning orchestrator in which the agent itself participates in blockchain governance (e.g., submitting proposals, casting votes, and updating its own policy parameters through the DAO) one of the few deployed systems where the governed and the governing entity are the same. Ikeda et al. [139] address the security dimension of DAO agent governance, proposing anomaly detection methods that distinguish malicious agent behaviour from legitimate protocol participation within DAO-managed crypto-asset transaction systems.

5.3. AI Governance Frameworks Applied to Decentralised Agents

The EU AI Act [10] establishes a risk-based framework for AI systems deployed in the European Union, with compliance requirements that scale with assessed risk. High-risk systems must register in an EU database, implement risk management processes, ensure human oversight, maintain technical documentation, and demonstrate accuracy and robustness. Prohibited systems include those that use subliminal techniques to manipulate behaviour.

The geographic scope of this analysis requires a caveat. The EU AI Act applies to systems placed on the EU market or whose outputs are used in the EU, regardless of where the developer is located. For decentralised agent systems deployed on global blockchains, this creates extraterritorial reach: a DeFi protocol accessible to EU users may fall within the Act's scope even if no component of its infrastructure is physically located in the EU. Outside the EU, however, the regulatory picture differs substantially. The United States has not enacted unified AI legislation as of 2026; regulatory action has proceeded through sector-specific agencies (SEC, CFTC, FTC) and executive orders instead of a unified framework. The UK's AI governance approach remains principles-based and non-statutory. China's AI regulations focus on generative AI and algorithmic recommendation, not autonomous agent systems. The three governance gaps identified below therefore apply specifically to the EU context, though analogous gaps exist in other jurisdictions under different regulatory instruments.

Mapping these requirements onto decentralised AI agents reveals structural mismatches at several points. The Act defines a “provider” as a natural or legal person that develops or places an AI system on the market. A DAO is neither; it has no legal personality in most jurisdictions. An agent fleet governed by a DAO therefore lacks an identifiable provider, and the Act's compliance obligations have no obvious bearer. The human oversight requirement is directly incompatible with operational autonomy as defined in Section 2.3: an agent requiring per-step human approval is not, by that definition,

a decentralised AI agent. The Act does not currently distinguish between systems that are autonomous by design and those deployed without oversight incidentally; this gap will require regulatory clarification as autonomous on-chain agents become more prevalent.

The NIST AI Risk Management Framework (AI RMF 1.0) [140] takes a more flexible approach, organising risk management around four functions: (a) Map (identify context and risks), (b) Measure (analyse and assess), (c) Manage (prioritise and implement responses), and (d) Govern (establish accountability structures). This framework can be applied to decentralised agent systems without presupposing a single identifiable legal entity, since it addresses processes and practices instead of compliance obligations. Its limitation is that it carries no enforcement mechanism, which reduces its practical effect on systems operating outside US regulatory reach.

Beyond the US, regulatory approaches diverge sharply. MiCA [11], the EU's markets in crypto-assets regulation, is the most immediately relevant instrument for agents operating in financial markets, though its scope is confined to the EU and firms serving EU customers. It establishes licensing requirements for crypto-asset service providers and issuers, and extends to algorithmic market-making and automated trading where an identifiable operator can be designated. However, MiCA explicitly defers fully decentralised protocols to future legislation: Article 4(3) and Recital 22 acknowledge that services provided in a fully decentralised manner without an intermediary fall outside the regulation's current perimeter, and the European Banking Authority's 2023 consultation on DeFi (Decentralised Finance) confirmed that this gap remains unresolved. An AI agent that provides liquidity to a decentralised exchange or executes algorithmic strategies on behalf of token holders therefore sits in a regulatory grey zone; it may fall within MiCA's scope if an identifiable operator or issuer can be designated, but the majority of production decentralised agent deployments deliberately eliminate that designation. Jurisdictional attribution compounds the difficulty: an agent deployed by a pseudonymous address, governed by a DAO with no registered domicile, and executing on a globally distributed network does not map cleanly onto MiCA's territorial scope regardless of how the perimeter question is resolved.

This mapping exposes three governance gaps that current frameworks do not address. Accountability without a legal entity is the first: existing frameworks presuppose an identifiable responsible party, which decentralised systems deliberately avoid. Auditability under zero-knowledge proofs is the second: a regulator can verify that an agent ran its declared policy but cannot inspect that policy for compliance, since the model weights remain private. Jurisdiction under a borderless protocol is the third: no single national regulator has clear authority over a globally distributed agent system, and regulatory arbitrage is achievable with minimal friction.

Empirical analysis of these gaps is beginning to emerge. Otalora and Themistocleous [141] analyse the regulatory challenges of LLM-integrated blockchain agents in the metaverse [142], documenting the specific points at which current frameworks, including MiCA and the AI Act, fail to map onto systems that are simultaneously AI agents, blockchain participants, and metaverse entities. Louck et al. [143] conduct a comparative security analysis of agentic AI communication protocols, identifying the trust boundaries and attack surfaces that any governance framework must address. Sekar [144] examines autonomous administrative intelligence in the context of DAO-governed organisations, arguing that existing administrative law provides partial but insufficient coverage for AI-mediated governance decisions. Vivekanand et al. [145] consider the sustainability governance dimension, proposing that agentic AI and blockchain together enable verifiable corporate sustainability reporting; a governance application where the auditability properties of the technology align naturally with regulatory intent.

5.4. Incentive Alignment and Principal–Agent Problems

The principal–agent problem describes the conflict of interest that arises when an agent acting on behalf of a principal has access to information the principal cannot observe, and objectives that may diverge from the principal’s [146]. In token-governed decentralised agent systems, this formulation undergoes a structural transformation: the principal is a distributed and heterogeneous set of token holders, each with different risk preferences, time horizons, and information access, rather than a single decision-maker. The autonomous software agent is simultaneously the agent in the economic sense and in the computational sense, but its objective function is encoded at deployment and may not reflect the preferences of the principal population that emerges after the token is distributed. Call this the *dual agency problem*: the software agent pursues its programmed objective, while the governance token holders pursue their investment and utility objectives, and neither mechanism ensures the two are aligned.

Divergence between the two shows up in several forms. Reward hacking occurs when an agent collects the reward signal without delivering the intended service [147]. In token-incentivised systems reward hacking takes the form of on-chain manipulation: submitting outputs that technically satisfy the contract’s verification condition without producing genuine value. Oracle manipulation attacks and governance exploits in DeFi confirm this is a live risk rather than a hypothetical one: economically rational agents will find and exploit the gap between the letter of a contract and its intent.

Emergent collusion arises when multiple independently deployed agents, each individually rational, converge on a cooperative equilibrium that extracts value from non-agent participants. MEV extraction by agent coalitions is the most documented instance: searcher bots that individually seek arbitrage opportunities can coordinate through private channels to divide MEV rents and exclude competitors, producing cartel-like outcomes without an explicit agreement [50]. The information systems and economics literature on algorithmic markets documents an analogous phenomenon: independent pricing agents, each optimising individually, can converge on supra-competitive equilibria through repeated interaction alone, without any communication or explicit coordination mechanism [148]. The mechanism is an emergent property of agents learning in shared environments with symmetric incentive structures, distinct from collusion in the legal sense. In shared on-chain environments the effect is amplified: agents observe the same public state, face the same incentive structure, and may independently arrive at strategies that are collectively anticompetitive.

The multi-principal problem arises when token holders have conflicting interests and the governance mechanism fails to aggregate them coherently. Platform theory [149,150] addresses multi-sided incentive conflicts in conventional platforms by having the operator set prices that internalise cross-side externalities. The DAO case lacks this operator: decisions that balance competing stakeholder interests must emerge from voting processes subject to the concentration and participation failures described in Section 5.2.

Mechanism design responses (e.g., output-based bonding, mutual monitoring, and retrospective evaluation) are partial at best. Output-based bonding requires a verifiable quality measure, reintroducing Goodhart’s Law. Mutual monitoring creates incentives for false reporting and may produce monitoring cartels. Retrospective evaluation reduces agent liquidity and may deter participation from smaller operators. The deeper difficulty is that all three responses assume a principal who can define correct behaviour and verify deviations. In distributed systems where the principal is itself fragmented, even this starting assumption fails. Token incentive design and MARL reward structure design are the same problem approached from different disciplines; the convergence of the two literatures has not yet occurred, and it represents one of the most productive directions for future work at the boundary of mechanism design and multi-agent reinforcement learning.

5.5. Organisational and Inter-Organisational Dimensions

Decentralised AI agent systems are not purely technical artefacts; they are organisational infrastructures that redistribute coordination functions across institutional boundaries, and their adoption and governance cannot be understood without attending to the organisational conditions that make them viable.

Digital platform theory characterises multi-sided platforms as intermediaries that reduce transaction costs between participant groups by maintaining the rules of interaction without directly controlling either side [149,151]. Blockchain-based agent systems fit this characterisation closely; the smart contract encodes interaction rules, the blockchain enforces them, and no single organisation controls the matching between agent service providers and consumers. This structural similarity makes platform theory a useful analytical lens, but it also exposes a significant gap. Platform governance in the conventional sense is exercised by a platform operator who can intervene, adjusting pricing, removing bad actors, or updating matching algorithms, in response to market failures. A DAO-governed agent platform can make analogous interventions only through its governance process, which is slower, more contentious, and subject to the failures described in Section 5.2. The governance capability that conventional platforms take for granted is precisely what decentralised governance sacrifices.

Cross-organisational agent pipelines represent the most direct application of decentralised agent systems to inter-organisational coordination. In supply chain management, a pipeline might span a manufacturer, a logistics provider, a customs authority, and a retailer, with each organisation operating its own agents and the blockchain providing a shared coordination layer that no single party controls. Inter-organisational information systems research has long identified trust, power asymmetries, and switching costs as the primary structural barriers to adoption in multi-party coordination settings [152,153]. Electronic data interchange deployments in the 1980s and 1990s demonstrated that technically sound coordination infrastructure fails when governance rights are perceived as asymmetrically distributed—a finding that anticipates the TradeLens outcome almost exactly [154]. Blockchain-based agent systems reduce the trust requirement by replacing institutional trust with cryptographic guarantees, but they do not eliminate it: confidence in the smart contract's correctness, the oracle network's integrity, and the governance process's fairness is required instead. These are, if anything, harder for non-specialist organisations to assess than institutional reputation [17], because they demand technical due diligence capabilities that most supply chain participants do not possess.

The transaction cost economics perspective [155] adds a further constraint: decentralised agent systems impose significant fixed costs, including development, audit, deployment, and governance participation, that are only justified when transaction volumes are high enough or trust deficits between participants are severe enough to warrant them. Empirical research on blockchain adoption in inter-organisational settings [17,156] consistently finds that adoption concentrates in niches where these conditions hold: cross-border trade finance, where correspondent banking relationships are slow and costly; energy markets with many small distributed producers who cannot individually afford bilateral settlement infrastructure; and supply chains with documented fraud risk. Outside these niches, centralised alternatives remain dominant.

Digital innovation ecosystem theory [157] provides a third lens. In ecosystem frameworks, a platform layer attracts complementors who build services on top of shared infrastructure, with value creation distributed across the ecosystem instead of captured by a single firm. Decentralised AI agent platforms aspire to this model: a base layer of coordination infrastructure, governed by a DAO, on which independent operators deploy specialised agent services. The theory predicts that ecosystem health depends on the plat-

form's ability to set and enforce quality standards for complementors while maintaining openness to entry. DAOs can enforce quality through token-curated registries and staking, but their enforcement speed and precision are limited by governance throughput. Whether DAO governance can sustain the complementor attraction and quality control that successful platform ecosystems require remains unsettled, and the answer will depend on whether the governance legitimacy problem in Section 5.2 is resolved before the ecosystem reaches the scale at which governance failures become consequential.

The corpus provides several concrete instantiations. Wang et al. [158] develop a prototype for self-owning intelligent buildings in which blockchain enables the building itself to act as an economic agent (e.g., signing service contracts, managing energy, and paying maintenance fees) an extreme case of operational autonomy where the asset and the agent are the same entity, with implications for property law and organisational liability that existing frameworks do not address. Gürpınar [159] frames the broader trajectory as a move toward Web 4.0, in which autonomous AI agents operate within decentralised enterprise coordination structures governed by smart contracts, identifying the governance design requirements for this shift to be compatible with existing organisational structures. De Gasperis et al. [160] demonstrate MAS and DAO integration for public sector tax credit tracking, showing that governance properties of DAOs extend meaningfully beyond financial applications to contexts where public accountability requirements apply. Singh et al. [161] survey the architectural spectrum from fully decentralised to semi-autonomous Web3 agent systems, providing an organisational taxonomy that complements the technical taxonomy in Table 2. Qi et al. [162] address the multi-agent trust and incentive problem directly, proposing a blockchain-driven framework for transparent and incentive-compatible collaboration in decentralised LLM multi-agent systems, treating inter-agent governance as a mechanism design challenge rather than an engineering one.

The three governance dimensions examined above, the DAO mechanisms, the regulatory frameworks, and the organisational adoption conditions, are not independent of the architectural and coordination choices as those described in Sections 3 and 4. They interact in a specific way. An agent's execution architecture determines what governance mechanisms are even available: an on-chain agent can be deactivated by contract, but an off-chain agent can only be deactivated by its deployer. A verifiable off-chain agent narrows the accountability gap by providing a cryptographic record of its behaviour, but introduces the Zero Knowledge governance problem described in open problem one. Token incentive coordination creates the dual agency problem precisely because token rewards are specified at deployment and cannot adapt when the principal population shifts. MARL-based coordination makes the principal-agent problem worse: the agent's policy evolves, which means the governance contract that was written against its initial policy may not constrain its adapted one. These dependencies mean governance cannot be treated as a layer added on top of a completed technical design. It is a design constraint that must be considered at the same time as execution architecture and coordination protocol, not after.

5.6. Survey Limitations

The findings of this survey should be interpreted with the following considerations in mind.

Publication bias. The corpus reflects what has been published in peer-reviewed venues and established preprint repositories, which over-represents successful deployments and positive experimental results. Systems that were designed but abandoned, architectures that failed to deploy, and coordination mechanisms that proved unworkable in practice are systematically underrepresented. The cross-domain observations in Section 6.5 should be read with this bias in mind: the apparent maturity of certain application domains may

reflect the publish-or-perish incentive structure of academic research rather than the actual distribution of production deployments.

Negative-result underrepresentation. Related to the above, studies reporting that a proposed coordination mechanism failed to converge, that a governance design produced perverse incentives, or that a ZKML deployment was computationally infeasible at target scale are rare in the corpus. The open problems in Section 7 partially compensate by identifying gaps that the positive literature has not closed, but the absence of systematic negative evidence limits the ability to assess which research directions are likely to remain blocked.

Reliance on system documentation for production deployments. Fourteen of the 191 corpus sources are vendor or foundation documentation for production systems (Autonolas, Fetch.ai, SingularityNET, Giza, and similar). These sources are not peer-reviewed and may reflect the system's intended rather than actual behaviour. Where peer-reviewed work on the same system is available, we have cross-referenced it; where it is not, claims about production deployment scale, performance, and governance structure carry the caveat that they rest on primary sources with an obvious interest in favourable presentation.

6. Applications and Empirical Evidence

6.1. Decentralised Finance

Decentralised Finance (DeFi) is the domain with the highest concentration of deployed autonomous agent systems and, by a wide margin, the largest real-money exposure. Total value locked in DeFi protocols peaked above \$180 billion in late 2021 and remained in the tens of billions through 2024–2025 [163], with a substantial fraction managed or directly influenced by autonomous agents.

The dominant architectural pattern is off-chain agent with on-chain settlement (Section 3.2). Liquidation bots monitor lending protocol state continuously and submit liquidation transactions the moment a collateral ratio crosses a threshold; the Aave [37] and Compound [38] ecosystems each support hundreds of competing bots. AMM (Automated Market Maker), a decentralised exchange mechanism that uses algorithmic pricing formulas, arbitrage bots detect price discrepancies between DEXs (Decentralised Exchanges) and close them through a rapid sequence of on-chain transactions. This activity keeps prices aligned across venues but its MEV dimension extracts value from ordinary traders in the same block [49].

AI-driven portfolio management is an area of active development. Yearn Finance vaults [47] introduced the pattern of automated strategy switching; later vaults incorporated RL-trained strategies [48]. The Numerai tournament [164] provides an unusual governance model: data scientists submit ML-trained models whose on-chain performance is staked with NMR tokens, and the tournament aggregates predictions into a hedge fund portfolio. It is one of the few DeFi applications where model quality is directly coupled to financial stake, producing strong incentive alignment that the staking and slashing mechanisms in Section 4.2 only approximate.

Governance of DeFi agents is weak relative to the standards discussed in Section 5. Most liquidation and arbitrage bots are deployed unilaterally by their operators with no DAO oversight and no staking requirement. The deployer is effectively the sole principal; liquidity providers and ordinary traders bear the costs of MEV extraction without formal recourse.

The corpus adds several DeFi agent deployments not covered above. Hou et al. [79] present SquirRL, which uses DRL to automate attack analysis on blockchain incentive mechanisms, demonstrating that RL agents can systematically discover selfish mining and withholding strategies that are economically suboptimal in theory but viable in practice. Li et al. [165] survey strategic mining behaviour in proof-of-work systems from a reinforcement learning perspective, showing that the incentive design of Bitcoin creates

an environment in which RL agents consistently converge on non-cooperative equilibria. Arcifa et al. [166] develop a bounded LLM orchestration system with deterministic safety gates for concentrated liquidity management in DeFi, combining LLM-based reasoning with smart contract guardrails that prevent the LLM from producing transactions that violate safety constraints. Luo et al. [167] address adversarial multi-agent dynamics in meme coin markets, using MARL with chain-of-thought reasoning to resist front-running manipulation by adversarial bots—a use case that highlights the competitive rather than cooperative face of multi-agent DeFi systems. Thomaidis et al. [168] study the coordination problem between AI workloads and Bitcoin mining, applying distributional MARL to jointly optimise computational resource allocation and mining revenue under operational constraints. The verification challenge differs substantially between fixed-weight RL agents (such as those in SquirRL and the strategic mining systems above) and LLM-based agents (such as Arcifa et al.’s bounded orchestration system); the two cases require separate technical paths, which are examined in detail in open problem 1 (Section 7).

6.2. Supply Chain and Logistics

Supply chain is the second most mature application domain, though production deployments with genuine multi-agent AI components remain rare [169]. The more common pattern is blockchain as a shared audit ledger, with rule-based agents handling trigger conditions rather than learning agents making complex decisions.

IBM Food Trust, built on Hyperledger Fabric [170], provides a provenance tracking system for food products in which participating organisations run local agents that submit supply chain events to a shared ledger. The system is federated: each organisation controls its own agents, and the shared ledger provides a consistent reference without a central operator. TradeLens [171], a similar system for shipping container tracking built jointly by Maersk and IBM, operated from 2018 until its discontinuation in November 2022. Its failure was attributed primarily to insufficient industry adoption rather than technical shortcomings, and it is a concrete example of the governance and adoption barriers discussed in Section 5.5: shipping carriers that competed with Maersk were reluctant to share data on infrastructure in which Maersk had a governance role.

More recent research has addressed smart-contract-based coordination in supply chains, where autonomous agents representing different organisations negotiate pricing, delivery terms, and quality guarantees, with agreed terms encoded in contracts that execute automatically on delivery confirmation [172]. The agent negotiation component remains largely at prototype stage; the settlement layer is production-ready.

The governance pattern in operational supply chain deployments is a consortium DAO or federated authority: a fixed set of known organisations share governance rights, with voting weighted by supply chain position or transaction volume. This is a more constrained structure than public DAO governance and reduces the flash loan attack surface described in Section 5.2, since governance token holdings are not tradeable on open markets.

The corpus extends coverage across several supply chain agent deployment types. Swain and Patra [173–175] present a series of distributed software agent frameworks for blockchain-enabled supply chain management and traceability, demonstrating progressively more capable agent coordination across supply chain actors. Kumar et al. [176] propose a blockchain-driven multi-agent framework for smart supply chain management that integrates transparency, efficiency, and resilience as co-equal design objectives, with smart contracts enforcing service-level agreements at each handoff. Giannopoulos et al. [177] address decentralised pricing in supply chains through a blockchain-enabled MARL approach in which competing agents discover equilibrium prices without a central auctioneer, showing that MARL convergence is achievable in supply chain negotiation

environments with moderate state space complexity. Velastegui et al. [178] provide a conceptual model for scheduling and control of production and logistics operations using MAS and blockchain, covering the coordination protocol design space that precedes implementation. Paruchuru et al. [14] demonstrate fully autonomous AI agents for continuous fraud monitoring in supply networks, using anomaly detection agents that operate without human supervision and commit evidence to blockchain for auditability.

6.3. Decentralised AI Agent Marketplaces

Agent marketplaces are the most architecturally ambitious application category: Fetch.ai, SingularityNET, and Autonolas have achieved meaningful deployment scale among agent marketplaces.

Fetch.ai [2] combines an agent framework, a peer-to-peer messaging network, and an on-chain registry (the Almanac) into a general-purpose coordination infrastructure. Agents register their capabilities and service endpoints on the Almanac; clients query the registry and negotiate directly over the messaging layer, with FET token staking required for registration. As of 2025, the network supports several thousand registered agents across logistics, DeFi, and smart energy applications. SingularityNET [3] focuses on AI service composition: agents representing ML models offer inference services, and client agents chain multiple services into composite workflows, with AGIX token governance determining listing conditions. Autonolas [1] takes a service-oriented approach, registering multi-agent systems with defined interfaces on a DAO-governed registry; operators stake OLAS tokens and earn rewards for uptime and correct behaviour.

From a platform economics perspective [149,151], all three systems face the standard cold-start problem: agents are not useful without clients, and clients are not attracted without useful agents. Each has addressed this through different subsidy strategies: Fetch.ai through direct grants to agent developers, SingularityNET through an internal research division that supplies initial services, and Autonolas through a developer incentive programme funded by the protocol treasury. Whether any of these strategies produces self-sustaining network effects at commercial scale remains to be demonstrated.

Beyond the three major marketplace platforms, the corpus documents a growing set of agent marketplace architectures at various stages of maturity. Yang et al. [179] develop BlkMaC, a blockchain-based MAS with an LLM communication framework in which agents discover each other through on-chain registries and coordinate using natural language interfaces backed by smart contract enforcement. Ioannides et al. [180] propose Mod-X, a modular open decentralised exchange framework for heterogeneous interoperable AI agents, addressing the inter-platform compatibility problem that currently prevents agents on different blockchain networks from composing services. Rahmani et al. [181] describe a peer-to-peer autonomous agent communication network that operates without centralised discovery infrastructure, relying on cryptographic identity and reputation for counterparty selection. Ragupathy [182] presents DAIS, a decentralised autonomous identity system for AI agents that uses distributed ledgers and trust registries to manage agent credentials at scale. Sümer et al. [183] characterise a semi-autonomous digital commerce model in which AI agents act as customers in Industry 5.0 supply chains, extending the marketplace concept to procurement contexts where agents have spending authority on behalf of human principals.

Ocean Protocol [184] represents a structurally distinct variant of the agent marketplace model. Where Fetch.ai, SingularityNET, and Autonolas facilitate markets for AI agent services, Ocean Protocol facilitates markets for the *data* that AI agents consume. Data publishers tokenise datasets as data NFTs and publish them with access conditions encoded in smart contracts; AI agents acting as buyers acquire compute access without the data ever leaving the publisher's infrastructure, through a Compute-to-Data architecture in which the

computation runs adjacent to the data rather than transferring the data to the computation. Governance of the protocol's curation, pricing, and access policies is managed by OCEAN token holders through a DAO. The analytical significance of this model for the present survey is the reversal of the standard marketplace structure: the agent sits on the demand side of a data market, not the supply side of a service market, which changes the incentive alignment problem. The publisher's interests (data confidentiality, fair compensation) must be balanced against the agent's interests (data quality, access continuity) without a centralised data broker, and the token governance mechanism must aggregate these competing preferences across a large and heterogeneous stakeholder base.

6.4. IoT and Edge-Based Agent Systems

IoT environments create a natural demand for decentralised agent coordination: large numbers of heterogeneous devices distributed across organisational boundaries, generating real-time data that must be acted upon locally while being recorded centrally for audit and billing. Placing agent logic at the network edge and committing only outcomes to the chain addresses both the latency and the bandwidth constraints of full on-chain execution.

IOTA [185] provides a directed acyclic graph ledger designed for high-throughput, feeless IoT transactions, with agent-like stream components that publish signed data sequences for downstream consumption. Fetch.ai has deployed IoT-specific agents for smart energy grid coordination, where local agents representing prosumers negotiate power allocation and commit settlement transactions on-chain [2]. The primary technical constraint in these deployments is resource asymmetry: edge devices often have limited compute and battery capacity, while blockchain interaction requires elliptic curve signature computation and sustained network connectivity. Lightweight agent designs that delegate heavy computation to nearby gateway nodes while retaining on-chain identity and settlement are the dominant architectural response [186].

SLA enforcement is a recurrent governance need in IoT agent systems. When an agent representing a sensor network commits to delivering data at a specified quality and frequency, a smart contract can hold payment in escrow and release it automatically on verified delivery, or redirect it to a backup agent if the primary fails. This is a direct application of the optimistic verification pattern described in Section 3.3, and it is one of the few IoT blockchain use cases where the on-chain settlement overhead is clearly justified by the governance benefit.

The corpus documents a large and diverse set of IoT and edge agent deployments. UAV networks represent the most active sub-area. Seid et al. study blockchain-empowered resource allocation for multi-UAV networks in 5G-RAN [54] and for multi-domain resource trading in 6G TN-NTN networks [55], consistently finding that blockchain settlement reduces the need for inter-operator trust while MARL handles the real-time allocation problem. Jia et al. [187] and He et al. [188] address trusted routing for blockchain-empowered UAV networks using MARL, demonstrating that routing decisions made by on-chain smart contracts with MARL-learned parameters can outperform centralised routing under adversarial conditions. Hafeez et al. [189] coordinate UAV fleets for post-disaster response using a consortium blockchain architecture, with smart contracts handling multi-agency access control and consensus enforcing coordination finality across competing responders. Vasalos et al. [13] develop a DAO-based swarm architecture for UAV analytics, in which governance of the swarm mission parameters is delegated to a DAO whose token holders include the operating organisations.

In vehicular networks and Internet of Vehicles (IoV), Chang et al. [59] apply distributed MARL to collaborative path planning and scheduling in blockchain-based cognitive IoV, while Kang et al. [190] address avatar migration for vehicular metaverse services using

MARL with blockchain settlement. Security applications appear throughout the corpus: Raja et al. [58] use blockchain-integrated MARL to secure cooperative adaptive cruise control against adversarial manipulation, and Chelghoum et al. [191] apply blockchain-enabled collaborative intrusion detection with RL to securing vehicle-to-everything communications in 6G networks. Wang et al. [120] address the IoT-specific problem of asynchronous federated RL for trajectory prediction, using blockchain to manage staleness and contribution attribution in a system where device availability is intermittent. Xu et al. [192] address on-chain and off-chain data management for blockchain-IoT systems using MARL, directly examining the data placement and cost trade-off that underlies most IoT blockchain deployments. Antuley et al. [193] develop SORA-ATMAS, an adaptive trust management system with multi-LLM aligned governance for smart cities, representing one of the more complete treatments of how LLM-based agents can be governed in IoT environments through a combination of on-chain trust registries and human-aligned policy constraints. Zhan et al. [194] provide the first systematic security analysis of edge agent deployments on IoT hardware, identifying attack surfaces specific to the three-tier cloud-edge-device architecture that most production IoT agent systems use.

6.5. Cross-Domain Observations

The 177-paper corpus, spanning four application domains over seven years, supports five cross-cutting observations that are not visible within any single domain.

The first is an architectural concentration that predates and persists across all domains. The expanded taxonomy in Table 2 classifies 20 representative systems; all but two fall in the off-chain agent with on-chain settlement category. On-chain execution is limited to rule-based financial contracts. Verifiable computation appears only in isolated research deployments. The distribution reflects a genuine engineering constraint rather than a sampling artefact: off-chain reasoning with on-chain settlement is the only pattern that simultaneously supports complex agent behaviour and economically viable transaction costs. The architectural alternatives described in Sections 3.1 and 3.3 will not challenge this dominance until ZKML proving costs fall by at least one order of magnitude or L2 execution environments support native ML inference.

MARL, meanwhile, dominates the research literature but is essentially absent from production deployments. The corpus contains over 50 systems that combine MARL with blockchain infrastructure in experimental or prototype settings, yet the production systems surveyed in Sections 6.1–6.4 are almost entirely rule-based or use simple single-agent RL. The gap has two explanations. MARL convergence in asynchronous on-chain environments is not guaranteed, as discussed in Section 7, which deters risk-averse deployers. And MARL systems require sustained training infrastructure that most production deployments do not maintain. The research-to-deployment gap for MARL is therefore wider in this setting than in most other applied MARL domains.

Governance maturity, for its part, is inversely related to deployment scale. DeFi, the domain with the largest deployed capital, has the weakest agent governance: most DeFi agents are unilaterally deployed with no DAO oversight and no staking requirement. Supply chain consortium deployments have stronger governance structures, since the participating organisations are identifiable and subject to contract law, but smaller deployment scale. Agent marketplaces have the most developed token governance, but remain at early commercial scale. The pattern suggests that governance investment is driven by regulatory pressure and reputational risk among identifiable institutional participants rather than by technical necessity. MiCA's current perimeter excludes fully decentralised protocols, but its progressive extension and the EU AI Act's applicability to high-risk autonomous systems

are likely to shift the DeFi governance posture substantially over the 2025–2027 period as regulatory guidance matures.

The fourth observation is that energy and UAV applications together account for roughly 40% of the corpus. Table 6 shows the distribution of the 177 peer-reviewed papers across application domains. Each paper was assigned to the domain of its primary application as stated in its abstract; papers that addressed multiple domains were assigned to the one accounting for the majority of the technical contribution. Papers where no single domain exceeded 50% of the contribution were placed in the residual “Other” category. Energy (including microgrids, P2P trading, demand response, and smart grids) and UAV/edge networks each account for approximately 20% of the corpus, and their combination exceeds DeFi, supply chain, and agent marketplaces combined.

Table 6. Distribution of the 177 peer-reviewed corpus papers by primary application domain.

Domain	Papers	Share (%)
Energy (P2P trading, microgrids, smart grids, demand response)	38	21.5
UAV, vehicular, and mobile edge computing	36	20.3
Decentralised finance and blockchain network management	27	15.3
IoT and general edge computing	26	14.7
Supply chain and manufacturing	18	10.2
Agent marketplaces and multi-agent platforms	16	9.0
Other (healthcare, governance, security, education)	16	9.0
Total	177	100

This concentration reflects the problem structure of the two leading domains rather than a bias in the literature. Energy and UAV systems share three features that map naturally onto blockchain-based MARL: many small heterogeneous actors with no prior trust relationship, short decision horizons that reward fast off-chain computation, and a genuine need for tamper-evident settlement because the underlying assets (energy, spectrum, compute) are rivalrous and contestable. Domains with fewer actors, longer horizons, or strong centralised governance alternatives (healthcare data markets, financial clearing, government services) are underrepresented for structural reasons, not bibliographic ones.

The fifth observation is that no production system in any domain currently uses ZKML or TEE verification at scale. Verifiable computation is consistently described as a necessary future capability, but the practical systems that would motivate its adoption (production agents whose behaviour must be audited by regulators or counterparties) are governed instead by staking and reputation mechanisms whose deterrence depends on economic assumptions that may not hold at high stakes. This is the dimension of the architecture-governance interaction that is most likely to be resolved by regulatory pressure rather than technical progress: when regulators require verifiable agent behaviour, the demand for ZKML or TEE verification will create the economic incentive to close the current cost gap.

7. Open Research Problems

The six problems identified below are stated precisely enough that each could form the basis of a targeted research contribution. They are not generic limitations restated as gaps; each identifies a specific missing result or mechanism whose absence currently constrains system design or deployment. A prioritized research roadmap summarising suggested first steps and estimated horizons is provided in Appendix C.

Verifiable agent behaviour under zero-knowledge constraints.

Current ZKML implementations (Giza [4] and EZKL [62]) prove that a model was evaluated correctly on a given input, but do not by default prove that the model used at inference

time is the model declared at deployment. An agent operator can swap underlying policy weights between deployments without any on-chain record of the change, provided the proof system commits only to the computation graph structure rather than the weights themselves.

For fixed-weight RL policies this gap has a tractable engineering response: committing a Merkle hash of the weight tensors at deployment and including that commitment in the ZK proof statement binds the declared policy to each inference proof. The additional circuit complexity is substantial but not prohibitive for the small models that current ZKML can handle. For constrained RL agents, then, this gap is an engineering step that production deployments have not yet taken rather than an open research problem, and its adoption will likely be driven by regulatory pressure rather than technical discovery.

The genuinely open problem lies one level up. LLM-based agents have no fixed weight set to commit: their “policy” is a prompt-plus-model combination that changes with each input, each session context, and potentially each fine-tuning cycle. No mechanism currently exists for proving that an LLM-based agent acted according to a declared prompt strategy without revealing the prompt, and without allowing the operator to substitute a different strategy between calls while presenting the same nominal identity. Weight hashing cannot address this. Progress requires either restricting the scope of verifiable LLM agents to those with immutable system prompts committed on-chain, or developing proof systems for program-level behavioural specifications that are independent of the model’s specific weights and prompt instantiation.

Success criterion: Can an on-chain verifier confirm, without ever seeing the prompt, that an LLM-based agent’s output was generated under a declared system prompt and that the prompt has not changed since a committed registration transaction? A proof system answering yes to both conditions would resolve this problem.

MARL convergence under chain-reorganisation-induced non-stationarity.

As established in Section 4.2, blockchain MARL involves two distinct forms of non-stationarity. Agent-induced non-stationarity (the standard MARL problem) is partially addressed by CTDE, and practical implementations using experience replay and target networks already handle variable block-time intervals with reasonable empirical performance. The open theoretical problem is *environment-induced* non-stationarity from chain reorganisations.

When a recently confirmed block is replaced by a competing fork, the environment state that agents observed and acted upon is retroactively changed. An agent that submitted a transaction in block k expecting reward r may find, after a reorganisation, that its transaction never occurred, that the state it acted on no longer exists, or that a different agent’s transaction was executed instead. This violates the Markov property not because other agents are learning but because the execution environment itself is non-deterministic in a way that has no equivalent in simulation. Standard CTDE convergence proofs for QMIX [26] and MADDPG [27] do not account for this. Whether CTDE algorithms converge to stable joint policies under the reorganisation frequencies observed in production blockchains [195], and at what rate convergence degrades as reorganisation frequency increases, is an open theoretical question. A simulation environment that accurately models block-time variance and reorganisation statistics would be the necessary first step, and no such environment currently exists in the MARL literature.

Success criterion: Prove or disprove convergence for a standard CTDE algorithm (QMIX or MADDPG) in a Markov game environment subject to reorganisation events at empirically calibrated frequencies. A convergence proof or a counterexample would both count, provided each is accompanied by an open-source simulation environment that reproduces the relevant chain dynamics.

Incentive-compatible agent deactivation.

Token-governed agent systems have well-designed mechanisms for activating agents (registration, staking, capability announcement) but poor mechanisms for deactivating them when their behaviour degrades. A DAO governance vote to deactivate an agent is slow, costly in governance attention, and vulnerable to blocking by the agent's deployer if they hold sufficient governance tokens. Slashing addresses individual violations but not gradual performance degradation that remains above the slashing threshold. What is needed is a mechanism that makes it individually rational for an agent or its deployer to voluntarily decommission the agent when performance falls below a system-level threshold, rather than continuing to collect reduced rewards. This is a mechanism design problem with no obvious solution: deactivation must be incentive-compatible and not simply allowed by the rules. Potential directions include time-decaying staking rewards conditioned on performance metrics, and automatic stake forfeiture below a rolling performance floor, but neither has been deployed or formally analysed in the agent context.

Success criterion: A rational agent deployer, facing the mechanism, finds it individually optimal to decommission the agent once its performance metric falls below a specified threshold, that is, decommissioning is the Nash equilibrium strategy. This would need to be shown formally or through agent-based simulation under calibrated parameter ranges.

Cross-chain agent coordination.

An agent that must act across multiple blockchain networks (for example, reading price data from one chain, executing a trade on a second, and settling payment on a third) faces fragmented state and asymmetric latency. Cross-chain messaging protocols such as the Inter-Blockchain Communication protocol (IBC) [196] and LayerZero [197] address the transport problem but not the coordination problem: an agent cannot atomically commit to actions on two chains simultaneously, because the chains finalise independently. This creates a window during which the agent has committed on one chain but not the other, exposing it to state inconsistency and front-running. The problem is analogous to the two-phase commit problem in distributed databases, but without a trusted coordinator, and with finality times that differ by orders of magnitude across chains. General solutions to this problem, whether based on ZK proofs of cross-chain state, trusted relayer networks, or optimistic challenge protocols, would substantially expand the design space for multi-chain agent systems.

Success criterion: A protocol under which an agent can atomically commit to actions on two or more chains with different finality models, with a formal proof of atomicity that holds under asynchronous networks and the failure of up to f of n relayers, deployed and tested on at least two production blockchain networks.

Regulatory attribution in autonomous agent pipelines.

When a pipeline of autonomous agents causes a harmful outcome (a biased credit decision, a market manipulation, a privacy violation), existing regulatory frameworks provide no clear mechanism for attributing liability across the pipeline. The EU AI Act [10] assigns obligations to “providers” and “deployers”, but in a pipeline where agent A was developed by one party, deployed and operated by another, and its output consumed by agent B developed and operated by third and fourth parties respectively, there is no unambiguous mapping from the Act's categories to the pipeline participants. MiCA [11] faces the same problem for financial agent pipelines. Legal scholarship on AI liability in multi-agent systems is in its early stages [123]; technical mechanisms that produce auditable records of each agent's contribution to a pipeline outcome (including the provenance of each input and the basis for each decision) would be a prerequisite for any attribution framework, whether legal or contractual. The combination of on-chain execution logs and ZK proofs of model evaluation is a plausible technical direction, but no deployed system currently provides this at pipeline scale.

Success criterion: A pipeline-level attribution architecture would take the on-chain transaction log of a multi-agent pipeline and produce a machine-readable provenance record, mapping each component of the final output to the agent and input responsible for it. The record would need to satisfy the evidentiary standard required under at least one of the EU AI Act, MiCA, or GDPR's automated decision-making provisions.

Emergent collusion detection.

Multi-agent systems operating on shared contract state may converge on collusive equilibria that are individually rational for each agent but collectively harmful to the protocol and its users. MEV coalition formation [50] is the best documented instance, but the problem is more general: any MARL system trained in a shared environment may develop cooperative strategies that optimise joint value at the expense of non-agent participants. The difficulty is that collusive behaviour is often empirically indistinguishable from coordination that is genuinely beneficial. Two agents that consistently act in sequence on the same price discrepancy may be colluding to extract MEV, or they may independently be providing liquidity that happens to be complementary. On-chain detection of collusion therefore cannot rely on behavioural signatures alone; it requires a model of counterfactual outcomes under non-collusive behaviour. No on-chain detection mechanism currently exists that can distinguish the two cases at runtime. Game-theoretic approaches to collusion detection in mechanism design [84] and empirical methods from industrial organisation offer partial tools, but translating them to the on-chain agent setting is an open problem.

Success criterion: Demonstrated in a controlled simulation environment: an on-chain or off-chain detection mechanism that keeps the false-positive rate below 5% on known-legitimate coordination patterns (e.g., complementary liquidity provision) while correctly flagging at least 80% of synthetic MEV coalition scenarios, with the detection logic publicly auditable.

8. Conclusions

The deployment of autonomous AI agents on blockchain infrastructure has outpaced the research frameworks needed to design, evaluate, and govern them. Systems that hold digital assets, execute smart contracts, and coordinate with other agents without per-step human approval are now operating at scale across decentralised finance, supply chain, IoT networks, and agent marketplace platforms. They control substantial economic value and interact with regulatory frameworks that were not designed with them in mind. This survey has imposed structure on that space across three dimensions (architecture, coordination, and governance), and the picture that emerges is uneven in a specific way: the technical dimensions are maturing faster than the governance dimension, and the governance dimension is where the practical risks are highest.

On architecture, the design space is now mapped well enough to support principled choices. The taxonomy in Table 2 distinguishes four execution patterns that differ in verifiability, computational expressiveness, and latency. The comparison in Table 1 quantifies the cost of each verification approach. Off-chain reasoning with on-chain settlement is the dominant pattern and will remain so; the 177-paper corpus confirms this is a stable equilibrium under current cost structures rather than an interim state. Practitioners now have sufficient empirical evidence to choose between execution patterns on technical grounds rather than by analogy.

On coordination, production-ready mechanisms (auctions and token incentives) are well enough understood that their failure modes can be designed around. MARL coordination is more promising and more fragile. The corpus documents substantial experimental deployment across energy, UAV, and vehicular domains, consistently showing that blockchain's public-state and atomic-commitment properties provide genuine advantages over conventional MARL environments. The open question is whether CTDE algorithms converge under block-time non-stationarity, and the stakes are real: systems deployed today on MARL policies

rest on convergence assumptions that have not been validated for this execution context. The right response is to instrument production deployments carefully and treat convergence as an empirical property to be verified rather than assumed, not to avoid MARL.

On governance, the gap between what exists and what is needed is widest. The governance vacuum described in Section 5.1 is a multi-stakeholder system in which no participant has both the authority and the information to govern the whole. It describes the operational state of most deployed decentralised agent systems today, not just an abstract theoretical concern. DAOs govern token parameters but not agent pipelines, and the dual agency problem in Section 5.4 captures why: the software agent and the governance token holders pursue different objectives, and neither the staking mechanism nor the DAO voting process provides a reliable bridge between them. This matters for the architectural and coordination choices in Sections 3 and 4 directly: an off-chain agent with no on-chain verification of its policy creates an accountability gap that no governance mechanism can close after the fact. The governance dimension constrains which architectural patterns are deployable in regulated contexts and which coordination mechanisms are viable at scale; it is not a concern that can be deferred to a later stage.

The EU AI Act and MiCA entered into force in 2024 and 2025. MiCA's current perimeter explicitly defers fully decentralised protocols, but its progressive extension and enforcement guidance will narrow the regulatory gap for DeFi agent systems over the next two to three years. The EU AI Act's risk-based framework applies to AI systems deployed in the EU regardless of the operator's location, which creates real compliance exposure for high-risk autonomous agent deployments even where MiCA does not yet reach. Regulatory attribution in autonomous agent pipelines (open problem five in Section 7) is the question that enforcement actions will force into resolution within that window, and soon.

This survey offers researchers and system designers a shared vocabulary for a field that has developed largely in parallel silos across distributed systems, mechanism design, IS governance, and AI: the three-condition definition of a decentralised AI agent, the four-dimension architectural taxonomy, the coordination mechanism comparison in Table 4, the governance vacuum and dual agency problem formulations, and the mapping of EU AI Act and MiCA onto the agent setting in Section 5.3. The most consequential open problems are at the boundaries between those silos. Whether the next significant advance comes from ZKML efficiency, MARL convergence theory, regulatory design, or digital platform governance is an open question. The answer will determine which of the six problems in Section 7 resolves first, and which ones wait for the others.

Author Contributions: Conceptualization, M.T. and E.K.; methodology, M.T. and E.K.; investigation, M.T. and E.K.; writing—original draft preparation, M.T. (Sections 1–4, 6 and 7) and E.K. (Section 5); writing—review and editing, M.T. and E.K.; visualization, M.T.; supervision, M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. The survey corpus consists of previously published, publicly available literature and system documentation cited in the References; the complete search strings, screening criteria, and coding rules used to assemble the corpus are provided in Appendix A.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MARL	Multi-Agent Reinforcement Learning
CTDE	Centralised Training, Decentralised Execution
RL	Reinforcement Learning
LLM	Large Language Model
DAO	Decentralised Autonomous Organisation
DeFi	Decentralised Finance
ZK	Zero-Knowledge
ZKML	Zero-Knowledge Machine Learning
TEE	Trusted Execution Environment
AMM	Automated Market Maker
MEV	Maximal Extractable Value
IoT	Internet of Things
IoV	Internet of Vehicles
UAV	Unmanned Aerial Vehicle
MiCA	Markets in Crypto-Assets Regulation
VCG	Vickrey–Clarke–Groves (mechanism)
SLA	Service-Level Agreement
L2	Layer 2 (blockchain scaling solution)
QMIX	Q-value Mixing Network (MARL algorithm)
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
COMA	Counterfactual Multi-Agent Policy Gradients
EVM	Ethereum Virtual Machine
MAS	Multi-Agent System
FL	Federated Learning

Appendix A. Search Protocol and Screening Criteria

Appendix A.1. Database Search Strings

The following platform-specific search strings were used. All searches were conducted between January and April 2026, covering publications from January 2019 onward.

IEEE Xplore (Command Search)—Primary pass:

```
("blockchain" OR "distributed ledger" OR "smart contract") AND ("autonomous agent" OR "multi-agent system" OR "AI agent" OR "software agent" OR "intelligent agent") AND ("coordination" OR "reinforcement learning" OR "multi-agent reinforcement learning" OR "governance" OR "incentive mechanism" OR "task allocation")
```

IEEE Xplore—MARL supplementary pass:

```
("QMIX" OR "MADDPG" OR "COMA" OR "cooperative MARL") AND ("blockchain" OR "smart contract" OR "distributed ledger")
```

ACM Digital Library:

```
("blockchain" OR "smart contract" OR "distributed ledger") AND ("autonomous agent" OR "multi-agent") AND ("coordination" OR "reinforcement learning" OR "governance" OR "incentive")
```

Scopus—Primary pass:

```
TITLE-ABS-KEY(("blockchain" OR "distributed ledger" OR "smart contract") AND ("autonomous agent" OR "multi-agent system" OR "AI agent" OR "software agent" OR "intelligent agent") AND ("coordination" OR "reinforcement learning" OR "governance" OR "incentive mechanism" OR "task allocation")) AND PUBYEAR > 2018 AND (LIMIT-TO(DOCTYPE,"ar") OR LIMIT-TO(DOCTYPE,"cp"))
```

Scopus—DAO supplementary pass:

```
TITLE-ABS-KEY(("decentralized autonomous organization" OR "DAO governance" OR "token voting" OR "on-chain governance") AND ("artificial intelligence" OR "machine learning" OR "autonomous agent" OR "automated")) AND PUBYEAR > 2018
```

arXiv (four targeted passes across cs.MA, cs.CR, cs.DC, cs.LG):

1. blockchain autonomous agent coordination reinforcement learning
2. multi-agent reinforcement learning blockchain smart contract
3. decentralized autonomous organization governance token voting AI
4. zero knowledge machine learning on-chain inference agent

*Appendix A.2. Inclusion and Exclusion Criteria***Inclusion criteria** (all three must hold):

1. The paper describes a software agent satisfying at least one of: RL/MARL-based policy, LLM-based reasoning, or rule-based automation with non-trivial adaptive logic.
2. Blockchain, a distributed ledger, or smart contracts are structural to the agent's operation (not incidental), in the sense that removing the blockchain component would fundamentally alter the agent's operating mode or accountability structure.
3. The paper was published or publicly available between January 2019 and April 2026.

Exclusion criteria (any one is sufficient):

- The word “agent” is used only in the economic rationality sense, with no computational agent behaviour present.
- Blockchain and AI appear in separate, unrelated sections of the paper without structural integration.
- The paper is a workshop paper under four pages with no corresponding full journal or conference publication.
- The paper is an editorial, column, or job-market profile.

Appendix A.3. Screening and Coding Protocol

Screening process: Title and abstract screening was performed by one reviewer (M.T.) applying the criteria above. Papers that could not be resolved from the abstract alone were advanced to full-text assessment. Full-text eligibility was assessed by both authors independently, with disagreements resolved by consensus discussion. No papers required arbitration beyond the initial consensus round.

Coding rules: Each included paper was coded on four dimensions corresponding to the taxonomy in Table 2: (i) execution locus (on-chain/off-chain + settlement/verifiable/multi-agent on-chain); (ii) verification mechanism; (iii) primary learning paradigm (RL/MARL/LLM/rule-based/hybrid); (iv) primary coordination mechanism (auction/token incentive/MARL/gossip/none). Papers with mixed or ambiguous profiles were coded by dominant property. Papers that span multiple application domains were assigned to the domain accounting for the majority of the technical contribution; multi-domain papers where no single domain exceeded 50% of the contribution were assigned to the “Other” category in Table 6.

Appendix B. Critical Synthesis of Representative Included Studies

Table A1 presents a structured synthesis of 42 representative papers from the corpus, selected to cover all four architectural patterns, all four coordination mechanisms, and all application domains. Arch. abbreviations: OC = On-chain; OS = Off-chain + settle; VR = Verifiable; MA = Multi-agent on-chain. Coord. abbreviations: AU = Auction; MR = MARL; TK = Token; GS = Gossip; – = n/a.

Table A1. Critical synthesis of 42 representative included studies. Arch.: OC = On-chain, OS = Off-chain + settle, VR = Verifiable, MA = Multi-agent. Coord.: AU = Auction, MR = MARL, TK = Token, GS = Gossip. Italicised rows denote application-domain groupings.

Reference	Year	Arch.	Coord.	Primary Contribution
<i>Decentralised Finance and Blockchain Network Management</i>				
Hou et al. [79]	2021	OS	TK	RL automates attack discovery on PoW incentive mechanisms; finds selfish mining via DQN
Thomaidis et al. [168]	2025	OS	MR	Distributional MARL jointly optimises AI workload and Bitcoin mining revenue
Arcifa et al. [166]	2026	OS	TK	Bounded LLM orchestration with smart contract safety gates for DeFi liquidity management
Li et al. [165]	2025	OS	TK	Systematic RL survey on strategic mining; RL agents consistently reach non-cooperative equilibria
Gularso & Wiputra [198]	2025	OS	TK	Structured outlook on autonomous DeFi agents; identifies governance deficit as main barrier
Luo et al. [167]	2026	OS	MR	MARL with chain-of-thought reasoning resists front-running bots in meme coin markets
Zhang & Chen [43]	2024	OS	TK	RL auto-tunes Hyperledger Fabric parameters via governance transactions
Islam et al. [102]	2024	OC	MR	MARL-based PoS consensus where validators learn bidding strategies on-chain
<i>Energy Systems and Smart Grids</i>				
Shen et al. [51]	2020	OS	MR	Foundational DRL + blockchain P2P energy trading; MADDPG with on-chain settlement
Liu et al. [119]	2025	OS	GS	Federated MADDPG for P2P energy trading; blockchain protects prosumer data
Kothuru et al. [90]	2026	OS	MR	Hierarchical MARL graph transformer for demand response in blockchain smart grids
Chen et al. [89]	2025	OS	MR	MARL-based P2P energy sharding optimisation on permissioned blockchain
Si et al. [199]	2024	OS	MR	Carbon-aware MADDPG for data centre microgrid energy management
Hassan et al. [95]	2025	OS	TK	Privacy-preserving energy trading and block mining for blockchain virtual power plants
Pan et al. [200]	2023	OS	MR	Blockchain and MARL for incentive IRS resource scheduling in reconfigurable networks
Sifakis [201]	2026	OS	MR	Global virtual prosumer framework for cross-border energy transactions using IoT and MARL
<i>UAV, Vehicular, and Mobile Edge Computing</i>				
Seid et al. [54]	2023	OS	AU	Blockchain-empowered multi-UAV 5G-RAN resource allocation via hierarchical MARL
Lu et al. [55]	2025	OS	AU	Multi-domain 6G TN-NTN resource trading using blockchain and hierarchical MARL
Jia et al. [187]	2025	OS	MR	Trusted routing for blockchain-empowered UAV networks via MARL
Wang et al. [120]	2025	OS	GS	Asynchronous federated RL for IoT trajectory prediction with blockchain attribution
Xu et al. [192]	2024	OS	MR	On-chain/off-chain data management framework for blockchain-IoT with MARL
Tang et al. [56]	2022	OS	MR	Blockchain-enabled task offloading with energy harvesting in multi-UAV IoT via MARL
Chelghoum et al. [191]	2025	OS	MR	Blockchain-enabled collaborative IDS with RL for V2X security in 6G networks
Kang et al. [190]	2024	OS	TK	UAV-assisted avatar task migration (on-demand, context-aware) for vehicular metaverse via MARL

Table A1. Cont.

Reference	Year	Arch.	Coord.	Primary Contribution
<i>Supply Chain and Manufacturing</i>				
Swain & Patra [174]	2024	OS	TK	Distributed software agent framework for blockchain-enabled supply chain management
Kumar et al. [176]	2026	OS	TK	Blockchain-driven MAS framework for smart supply chain with smart contract SLAs
Giannopoulos et al. [177]	2025	OS	AU	MARL for decentralised pricing in supply chains without a central auctioneer
Leng et al. [75]	2023	MA	TK	ManuChain II: blockchain smart contract system as digital twin of autonomous manufacturing
Paruchuru et al. [14]	2026	OS	TK	Fully autonomous AI agents for continuous fraud monitoring in supply networks
Velastegui et al. [178]	2023	OS	AU	MAS and blockchain for scheduling and control of production/logistics operations
<i>Decentralised Agent Marketplaces</i>				
Chen et al. [73]	2026	MA	TK	AgentChain: blockchain-coordinated LLM multi-agent system for trustworthy QA
Ioannides et al. [180]	2026	OS	AU	Mod-X: modular open decentralised exchange for heterogeneous interoperable AI agents
Qi et al. [162]	2026	OS	TK	Blockchain-driven transparent and incentive-compatible LLM multi-agent collaboration
Singh et al. [161]	2025	OS	TK	Architecture survey: Web3 agents from fully decentralised to semi-autonomous
Ragupathy [182]	2026	OS	TK	DAIS: decentralised autonomous identity system for AI agents using DLT
Davarakis et al. [61]	2023	OS	TK	RL with smart contracts: blockchain as the RL environment itself, beyond settlement
<i>Governance and DAO-Based Systems</i>				
Ahmed et al. [80]	2026	OS	TK	APOLLO: agent participates in its own DAO governance; self-modifying policy via on-chain vote
Cu et al. [134]	2021	MA	TK	DAO-governed autonomous agent system for fractional asset ownership management
Lee et al. [136]	2023	OS	GS	DAO governance extended to federated learning policy decisions
Valiente & Pavón [137]	2024	OC	TK	Formal ontology for DAOs covering multi-agent interaction patterns
Antuley et al. [193]	2026	OS	TK	SORA-ATMAS: adaptive trust management and multi-LLM governance for smart cities
Phiri [69]	2025	VR	TK	Eight auditability axioms for agentic AI; blockchain as the natural audit substrate

Appendix C. Prioritized Research Roadmap for Open Problems

Table A2 organises the six open problems identified in Section 7 by priority, prerequisite dependencies, and recommended first research step. Priority is assessed on two axes: the magnitude of the barrier the problem poses to current deployments (technical impact), and the degree to which progress depends on unresolved problems elsewhere (coupling). High-priority problems are both high-impact and relatively self-contained. The roadmap is intended as guidance for researchers entering the area rather than a prescriptive sequence.

Table A2. Prioritized research roadmap for the six open problems. Pri.: H = high, M = medium. Horizon: near = 1–3 yr, medium = 3–6 yr.

Problem	Pri.	Why Urgent	First Step	Horizon
MARL convergence under reorg-induced non-stationarity	H	Production MARL deployments rest on unvalidated convergence assumptions; a single reorg in a live system can invalidate learned policies without warning	Open-source simulation environment modelling block-time variance and reorg events at calibrated frequencies	Near (1–2 yr)
Verifiable LLM agent policy commitment	H	LLM-based agents cannot currently prove they acted under a declared strategy; regulatory attribution and trust require this	Commit system prompt hash on-chain at registration; design proof system for prompt-invariant behavioural specification	Near-medium (2–4 yr)
Emergent collusion detection	H	MEV coalition formation causes measurable economic harm today; existing detection relies on heuristics with high false-positive rates	Benchmark dataset of labelled collusive vs. cooperative on-chain agent interactions drawn from production MEV data	Near (1–3 yr)
Regulatory attribution in multi-agent pipelines	M	EU AI Act and MiCA enforcement will require attribution; no technical mechanism exists at pipeline scale	Pipeline provenance architecture using on-chain execution logs and ZK proofs of model evaluation	Medium (3–5 yr)
Incentive-compatible agent deactivation	M	Gradual performance degradation is unaddressed by slashing; governance votes are too slow and gameable	Formal mechanism design analysis of time-decaying stake rewards conditioned on rolling performance floor	Medium (3–5 yr)
Cross-chain atomic coordination	M	Required for multi-chain agent systems but currently unsolved without a trusted coordinator	Atomic cross-chain commitment protocol with formal atomicity proof under asynchronous networks and partial relayer failure	Medium-long (4–6 yr)

References

1. Autonolas. Autonolas Protocol Documentation. 2024. Available online: <https://docs.autonolas.network> (accessed on 1 May 2026).
2. Fetch.ai. Fetch.ai Technical Whitepaper. 2023. Available online: <https://fetch.ai/docs> (accessed on 1 May 2026).

3. SingularityNET Foundation. SingularityNET Platform Overview. 2023. Available online: <https://singularitynet.io> (accessed on 1 May 2026).
4. Giza. Giza: On-Chain ML Inference. 2024. Available online: <https://www.gizatech.xyz> (accessed on 1 July 2026).
5. Shoham, Y.; Leyton-Brown, K. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*; Cambridge University Press: Cambridge, UK, 2008.
6. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X.; Wang, H. Blockchain Challenges and Opportunities: A Survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375.
7. Fritsch, R.; Müller, M.; Wattenhofer, R. Analyzing Voting Power in Decentralized Governance: Who Controls DAOs? *arXiv* **2022**, arXiv:2204.01176.
8. Kalodner, H.; Goldfeder, S.; Chen, X.; Weinberg, S.M.; Felten, E.W. Arbitrum: Scalable, Private Smart Contracts. In *Proceedings of the 27th USENIX Security Symposium, Baltimore, MD, USA, 15–17 August 2018*; USENIX Association: Berkeley, CA, USA, 2018; pp. 1353–1370.
9. Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. A Survey on Large Language Model Based Autonomous Agents. *Front. Comput. Sci.* **2024**, *18*, 186345.
10. The Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act). *Off. J. Eur. Union* **2024**, 1–144. Available online: <https://eur-lex.europa.eu/eli/reg/2024/1689/oj> (accessed on 1 July 2026).
11. The Council of the European Union. Regulation (EU) 2023/1114 of the European Parliament and of the Council of 31 May 2023 on Markets in Crypto-Assets (MiCA). *Off. J. Eur. Union* **2023**, L150, 40–205.
12. Qin, K.; Zhou, L.; Livshits, B.; Gervais, A. Attacking the DeFi Ecosystem with Flash Loans for Fun and Profit. In *Proceedings of the 25th International Conference on Financial Cryptography and Data Security (FC 2021), Virtual Event, 1–5 March 2021*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 3–32.
13. Vasalos, A.; Economopoulos, A.; Oikonomakis, A.; Chakraborty, A.; Kourtis, M.-A.; Alexandridis, G.; Tavernier, W.; Xilouris, G.; Chochliouros, I.; Vasalos, I.; et al. A distributed UAV analytics framework for DAO-based swarm systems. In *2025 44th International Symposium on Reliable Distributed Systems (SRDS)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/SRDS69199.2025.00057>.
14. Paruchuru, V.V.K.; Jamili, L.K.; Kavuluri, H.V.R.; Syeda, R.H. Autonomous AI Agents for Continuous Monitoring and Fraud Mitigation in Supply Networks. In *Proceedings of the 2026 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICESES)*; IEEE: Piscataway, NJ, USA, 2026; pp. 1–10. <https://doi.org/10.1109/ICESES66558.2026.11479064>.
15. Nguyen, D.C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Li, J.; Poor, H.V. Cooperative Task Offloading and Block Mining in Blockchain-Based Edge Computing With Multi-Agent Deep Reinforcement Learning. *IEEE Trans. Mob. Comput.* **2023**, *22*, 2021–2037. <https://doi.org/10.1109/TMC.2021.3120050>.
16. Busoniu, L.; Babuska, R.; De Schutter, B. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. Part C* **2008**, *38*, 156–172.
17. Lumineau, F.; Wang, W.; Schilke, O. Blockchain Governance—A New Way of Organizing Collaborations? *Organ. Sci.* **2021**, *32*, 500–521. <https://doi.org/10.1287/orsc.2020.1379>.
18. Buterin, V.; Dror, Y.; Lesuisse, A.; Forshtat, D. Account Abstraction Using Alt Mempool (ERC-4337). Ethereum Improvement Proposals. 2021. Available online: <https://eips.ethereum.org/EIPS/eip-4337> (accessed on 1 May 2026).
19. Breidenbach, L.; Cachin, C.; Chan, B.; Coventry, A.; Ellis, S.; Juels, A.; Koushanfar, F.; Miller, A.; Magauran, B.; Moroz, D.; et al. *Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks*; Technical Report; Chainlink Labs: San Francisco, CA, USA, 2021. Available online: <https://research.chain.link/whitepaper-v2.pdf> (accessed on 1 May 2026).
20. Pyth Network Contributors. Pyth Network: A First-Party Financial Oracle. 2023. Available online: <https://pythdataassociation.com/whitepaper.pdf> (accessed on 1 July 2026).
21. UMA Project. UMA: Optimistic Oracle Protocol. 2023. Available online: <https://docs.uma.xyz> (accessed on 1 May 2026).
22. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
23. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-Level Control Through Deep Reinforcement Learning. *Nature* **2015**, *518*, 529–533.
24. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
25. Kraemer, L.; Banerjee, B. Multi-Agent Reinforcement Learning as a Rehearsal for Decentralized Planning. *Neurocomputing* **2016**, *190*, 82–94.
26. Rashid, T.; Samvelyan, M.; Schroeder de Witt, C.; Farquhar, G.; Foerster, J.; Whiteson, S. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018), Stockholm, Sweden, 10–15 July 2018*; JMLR: New York, NY, USA, 2018; pp. 4295–4304.

27. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 2017), Long Beach, CA, USA, 4–9 December 2017*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 6379–6390.
28. Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual Multi-Agent Policy Gradients. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, LA, USA, 2–7 February 2018*; AAAI: Washington, DC, USA, 2018; pp. 2974–2982.
29. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023), Kigali, Rwanda, 1–5 May 2023*; Curran Associates, Inc.: Red Hook, NY, USA, 2023.
30. Karim, M.M.; Van, D.H.; Khan, S.; Qu, Q.; Kholodov, Y. AI Agents Meet Blockchain: A Survey on Secure and Scalable Collaboration for Multi-Agents. *Future Internet* **2025**, *17*, 57. <https://doi.org/10.3390/fi17020057>.
31. Alqithami, S. Autonomous Agents on Blockchains: Standards, Execution Models, and Trust Boundaries. *arXiv* **2026**, arXiv:2601.04583.
32. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 Statement: An Updated Guideline for Reporting Systematic Reviews. *BMJ* **2021**, *372*, n71. <https://doi.org/10.1136/bmj.n71>.
33. Wood, G. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*; Technical Report 151; Ethereum Foundation: Zug, Switzerland, 2014. Available online: <https://ethereum.github.io/yellowpaper/paper.pdf> (accessed on 1 May 2026).
34. Buterin, V.; Swende, M. EIP-2929: Gas Cost Increases for State Access Opcodes. Ethereum Improvement Proposals. 2021. Available online: <https://eips.ethereum.org/EIPS/eip-2929> (accessed on 1 May 2026).
35. Chainlink Labs. Chainlink VRF: Verifiable Random Function. 2020. Available online: <https://docs.chain.link/vrf> (accessed on 1 May 2026).
36. Adams, H.; Zinsmeister, N.; Salem, M.; Keefer, R.; Robinson, D. *Uniswap v3 Core*; Technical Report; Uniswap Labs: New York, NY, USA, 2021. Available online: <https://uniswap.org/whitepaper-v3.pdf> (accessed on 1 May 2026).
37. Aave. *Aave Protocol Whitepaper V1.0*; Technical Report; Aave: London, UK, 2020. Available online: https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf (accessed on 1 May 2026).
38. Leshner, R.; Hayes, G. *Compound: The Money Market Protocol*; Technical Report; Compound Finance: San Francisco, CA, USA, 2019. Available online: <https://compound.finance/documents/Compound.Whitepaper.pdf> (accessed on 1 May 2026).
39. MakerDAO. The Maker Protocol: MakerDAO's Multi-Collateral Dai (MCD) System. 2021. Available online: <https://docs.makerdao.com> (accessed on 1 May 2026).
40. Skowroński, R. Adaptive Threat Mitigation in PoW Blockchains (Part II): A Deep Reinforcement Learning Approach to Countering Evasive Adversaries. *Sensors* **2026**, *26*, 1368. <https://doi.org/10.3390/s26041368>.
41. Martino, T.C.; Quaintance, M.; Haber, S. Agent-Based Simulations of Blockchain protocols illustrated via Kadena's Chainweb. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*; IEEE: Piscataway, NJ, USA, 2019; pp. 386–395. <https://doi.org/10.1109/EuroSPW.2019.00049>.
42. Li, M.; Wang, Y.; Ma, S.; Liu, C.; Huo, D.; Wang, Y.; Xu, Z. Auto-Tuning with Reinforcement Learning for Permissioned Blockchain Systems. *Proc. VLDB Endow.* **2023**, *16*, 1000–1012. <https://doi.org/10.14778/3579075.3579076>.
43. Chen, Y.Z. Apollo: Permissioned Blockchain Network Auto-Tuning Based on DRL Against Eclipse Attack. In *2024 20th International Conference on Mobility, Sensing and Networking (MSN)*; IEEE: Piscataway, NJ, USA, 2024; pp. 328–335. <https://doi.org/10.1109/MSN63567.2024.00053>.
44. Halasz, D.; Shandilya, S.; Buhnova, B. Trustworthy Execution in Untrustworthy Autonomous Systems. In *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*; IEEE: Piscataway, NJ, USA, 2023. <https://doi.org/10.1109/TrustCom60117.2023.00240>.
45. Chainlink Labs. Chainlink Automation Documentation. 2022. Available online: <https://docs.chain.link/chainlink-automation> (accessed on 1 May 2026).
46. Gelato Network. Gelato: Web3's Decentralized Backend. 2023. Available online: <https://docs.gelato.network> (accessed on 1 May 2026).
47. Cronje, A. Yearn Finance Documentation. 2020. Available online: <https://docs.yearn.fi> (accessed on 1 May 2026).
48. Xu, Y.; Yang, Y.; Li, J.; Ye, J.; Liu, X. Deep Reinforcement Learning for Cryptocurrency Trading: Practical Approach to Address Backtest Overfitting. *arXiv* **2023**, arXiv:2209.05559.
49. Daian, P.; Goldfeder, S.; Kell, T.; Li, Y.; Zhao, X.; Bentov, I.; Breidenbach, L.; Juels, A. Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability. In *Proceedings of the 41st IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 18–21 May 2020*; IEEE: Piscataway, NJ, USA, 2020; pp. 910–927.
50. Flashbots. Flashbots: Illuminating the Dark Forest. 2021. Available online: <https://docs.flashbots.net> (accessed on 1 May 2026).

51. Shen, Y.X.; Yu, L.; Bi, G.; Zhang, M. Deep Reinforcement Learning and Blockchain for Peer-to-Peer Energy Trading among Microgrids. In *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*; IEEE: Piscataway, NJ, USA, 2020; pp. 360–365. <https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00071>.
52. Singh, A.R.; Kumar, R.S.; Bajaj, M.; Kumar, B.H.; Blazek, V.; Prokop, L. A blockchain-enabled multi-agent deep reinforcement learning framework for real-time demand response in renewable energy grids. *Energy Strategy Rev.* **2025**, *62*, 101905. <https://doi.org/10.1016/j.esr.2025.101905>.
53. Gao, C.Z.; Song, M.; Wang, W. Personalized Dynamic Operating Envelope-Based P2P Energy Trading: A novel Multiagent Safe Reinforcement Learning. *IEEE Trans. Smart Grid* **2026**, *17*, 3248–3261. <https://doi.org/10.1109/TSG.2026.3660840>.
54. Seid, A.M.; Erbad, A.; Abishu, H.N.; Albaseer, A.; Abdallah, M.; Guizani, M. Blockchain-Empowered Resource Allocation in Multi-UAV-Enabled 5G-RAN: A Multi-Agent Deep Reinforcement Learning Approach. *IEEE Trans. Cogn. Commun. Netw.* **2023**, *9*, 991–1011. <https://doi.org/10.1109/TCCN.2023.3262242>.
55. Seid, A.M.; Abishu, H.N.; Rathore, R.S.; Erbad, A.; Jhaveri, R.H.; Lu, J. Blockchain-Empowered Multi-Domain Resource Trading in TN-NTN 6G Networks: A Hierarchical Multi-Agent DRL Approach. *IEEE Trans. Consum. Electron.* **2025**, *71*, 3874–3889. <https://doi.org/10.1109/TCE.2024.3512581>.
56. Seid, A.M.; Lu, J.; Abishu, H.N.; Ayall, T.A. Blockchain-Enabled Task Offloading With Energy Harvesting in Multi-UAV-Assisted IoT Networks: A Multi-Agent DRL Approach. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 3517–3532. <https://doi.org/10.1109/JSAC.2022.3213352>.
57. Xing, Z. AI-Driven Multi-Agent Deep Reinforcement Learning for Dynamic Sharding Reconfiguration in Permissionless Blockchains. In *2025 6th International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*; IEEE: Piscataway, NJ, USA, 2025; pp. 335–339. <https://doi.org/10.1109/ICBAIE66852.2025.11326629>.
58. Raja, G.; Kottursamy, K.; Dev, K.; Narayanan, R.; Raja, A.; Karthik, K.B.V. Blockchain-Integrated Multiagent Deep Reinforcement Learning for Securing Cooperative Adaptive Cruise Control. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 9630–9639. <https://doi.org/10.1109/TITS.2022.3168486>.
59. Chang, H.; Liu, Y.; Sheng, Z. Distributed Multi-Agent Reinforcement Learning for Collaborative Path Planning and Scheduling in Blockchain-Based Cognitive Internet of Vehicles. *IEEE Trans. Veh. Technol.* **2024**, *73*, 6301–6317. <https://doi.org/10.1109/TVT.2023.3344934>.
60. Alagha, A.; Kadadha, M.; Mizouni, R.; Singh, S.; Bentahar, J.; Otrók, H. UAV-assisted Internet of vehicles: A framework empowered by reinforcement learning and Blockchain. *Veh. Commun.* **2025**, *52*, 100874. <https://doi.org/10.1016/j.vehcom.2025.100874>.
61. Varvarigou, D.T.T.; Palaiokrassas, G.; Litke, A. Reinforcement learning with smart contracts on blockchains. *Future Gener. Comput. Syst.* **2023**, *148*, 550–563. <https://doi.org/10.1016/j.future.2023.06.018>.
62. Morton, J.; Pearce, J.; Eagen, L. EZKL: Make Inference Verifiable. 2023. Available online: <https://github.com/zkonduit/ezkl> (accessed on 1 May 2026).
63. Modulus Labs. The Cost of Intelligence: Proving Machine Learning Inference with Zero Knowledge. *arXiv* **2023**, arXiv:2211.09028.
64. Zhang, F.; Cecchetti, E.; Croman, K.; Juels, A.; Shi, E. Town Crier: An Authenticated Data Feed for Smart Contracts. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, Vienna, Austria, 24–28 October 2016; ACM: New York, NY, USA, 2016; pp. 270–282.
65. Van Bulck, J.; Minkin, M.; Weisse, O.; Genkin, D.; Kasikci, B.; Piessens, F.; Silberstein, M.; Wenis, T.F.; Yarom, Y.; Strackx, R. Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution. In *Proceedings of the 27th USENIX Security Symposium*, Baltimore, MD, USA, 15–17 August 2018; ACM: New York, NY, USA, 2018; pp. 991–1008.
66. Teutsch, J.; Reitwiesner, C. A Scalable Verification Solution for Blockchains. *arXiv* **2017**, arXiv:1908.04756.
67. Canesin, F.; Fragoso, D.; Fragoso, C.; Tupinamba, A. *Cartesi: Scalable Smart Contracts with Linux*; Technical Report; Cartesi Foundation: Singapore, 2022. Available online: <https://cartesi.io/cartesi-rollups.pdf> (accessed on 1 May 2026).
68. Andrijasa, M.F.; Ismail, S.A.; Ahmad, N.; Yusop, O.M. Enhancing Smart Contract Security Through Multi-Agent Deep Reinforcement Learning Fuzzing: A Survey of Approaches and Techniques. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 754–767. <https://doi.org/10.14569/IJACSA.2024.0150576>.
69. Phiri, C.C. Creating Characteristically Auditable Agentic AI Systems. In *Proceedings of the Intelligent Robotics FAIR 2025, IntRob '25*; ACM: New York, NY, USA, 2025; pp. 1–14. <https://doi.org/10.1145/3759355.3759356>.
70. Kim, S.Y.; Kim, D.Y.; Kim, D.-Y. A Procedural Architecture for Agent Trust and Credibility Verification in Distributed Ledger-Based Federated Learning. *IEEE Access* **2026**, *14*, 38361–38374. <https://doi.org/10.1109/ACCESS.2026.3671712>.
71. Expanso. Bacalhau: Compute Over Data. 2023. Available online: <https://docs.bacalhau.org> (accessed on 1 May 2026).

72. Yang, T.; Liu, Y.; Yang, X.; Kang, Y. A Blockchain based Smart Agent System Architecture. In *Proceedings of the 4th International Conference on Crowd Science and Engineering, ICCSE'19*; ACM: New York, NY, USA, 2019; pp. 33–39. <https://doi.org/10.1145/3371238.3371244>.
73. Chen, B.; Li, G.; Wu, J.; Li, J.; Chen, M.; Wang, J. AgentChain: Blockchain-empowered Multi-agent Coordination for Trustworthy LLM Question-answering Systems. *IEEE Trans. Dependable Secur. Comput.* **2026**, early access. <https://doi.org/10.1109/TDSC.2026.3685256>.
74. Sami, H.; Mizouni, R.; Otko, H.; Singh, S.; Bentahar, J.; Mourad, A. LearnChain: Transparent and cooperative reinforcement learning on Blockchain. *Future Gener. Comput. Syst.* **2024**, *150*, 255–271. <https://doi.org/10.1016/j.future.2023.09.012>.
75. Leng, J.; Zhu, X.; Huang, Z.; Xu, K.; Liu, Z.; Liu, Q.; Chen, X. ManuChain II: Blockchain Smart Contract System as the Digital Twin of Decentralized Autonomous Manufacturing Toward Resilience in Industry 5.0. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 4715–4728. <https://doi.org/10.1109/TSMC.2023.3257172>.
76. Leng, J.; Sha, W.; Lin, Z.; Jing, J.; Liu, Q.; Chen, X. Blockchain smart contract pyramid-driven multi-agent autonomous process control for resilient individualised manufacturing towards Industry 5.0. *Int. J. Prod. Res.* **2023**, *61*, 4302–4321. <https://doi.org/10.1080/00207543.2022.2089929>.
77. Praitheeshan, P.; Pan, L.; Zheng, X.; Jolfaei, A.; Doss, R. SolGuard: Preventing external call issues in smart contract-based multi-agent robotic systems. *Inf. Sci.* **2021**, *579*, 150–166. <https://doi.org/10.1016/j.ins.2021.08.007>.
78. Ge, S.; Meng, H.; Khan, S.; Alhayan, F. A Blockchain-Based Smart Contract Framework for Autonomous Sports Training Management in Multi-Agent Environment. *Trans. Emerg. Telecommun. Technol.* **2026**, *37*, e70364. <https://doi.org/10.1002/ett.70364>.
79. Hou, C.; Zhou, M.; Ji, Y.; Daian, P.; Tramèr, F.; Fanti, G.; Juels, A. SquirRL: Automating Attack Analysis on Blockchain Incentive Mechanisms with Deep Reinforcement Learning. In *Network and Distributed Systems Security (NDSS) Symposium 2021*; Internet Society (ISOC): Reston, VA, USA, 2021. <https://doi.org/10.14722/ndss.2021.24188>.
80. Ahmed, I.; Zubraj, Z.M.; Ferdous, M.S.; Nakano, T.; Tran, T.H. APOLLO: Autonomous Predictive On-Chain Learning Orchestrator for AI-Driven Blockchain Governance. *Digital* **2025**, *6*, 3. <https://doi.org/10.3390/digital6010003>.
81. Vickrey, W. Counterspeculation, Auctions, and Competitive Sealed Tenders. *J. Financ.* **1961**, *16*, 8–37.
82. Clarke, E.H. Multipart Pricing of Public Goods. *Public Choice* **1971**, *11*, 17–33.
83. Groves, T. Incentives in Teams. *Econometrica* **1973**, *41*, 617–631.
84. Cramton, P.; Shoham, Y.; Steinberg, R. (Eds.) *Combinatorial Auctions*; MIT Press: Cambridge, MA, USA, 2006.
85. Li, H.; Xue, W. A Blockchain-Based Sealed-Bid e-Auction Scheme with Smart Contract and Zero-Knowledge Proof. *Secur. Commun. Netw.* **2021**, *2021*, 5523394. <https://doi.org/10.1155/2021/5523394>.
86. Guo, J.; Du, X.; Wang, T.; Jia, W. Combinatorial Resources Auction in Decentralized Edge-Thing Systems Using Blockchain and Differential Privacy. *Inf. Sci.* **2022**, *607*, 456–473. <https://doi.org/10.1016/j.ins.2022.05.128>.
87. Shapley, L.S. A Value for n -Person Games. In *Contributions to the Theory of Games, Vol. II*; Kuhn, H.W., Tucker, A.W., Eds.; Princeton University Press: Princeton, NJ, USA, 1953; pp. 307–317.
88. Sun, Z.; Hao, R.; Lu, T.; Wang, M. Optimization of Multi-microgrid Operation Based on Blockchain Technology and Multi-agent Reinforcement Learning. In *2023 International Conference on Power Energy Systems and Applications (ICoPESA)*; IEEE: Piscataway, NJ, USA, 2023. <https://doi.org/10.1109/ICoPESA56898.2023.10140894>.
89. Chen, J.; Hu, X.; Zhu, X.; Chang, H.; Cui, Y.; Qi, J. Optimization Method for P2P Energy Trading Sharding Based on Multi-Agent Reinforcement Learning. In *2025 44th Chinese Control Conference (CCC)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.23919/CCC64809.2025.11179025>.
90. Kothuru, V.; majeti, K.; Dwivedi, S.; Lingaraj, K.; M, L.; Gupta, A.S.L.G. Energy efficient hierarchical multi-agent graph transformer architecture for decentralized demand response and energy trading in blockchain-coordinated smart grids. *Sustain. Comput. Inform. Syst.* **2026**, *50*, 101346. <https://doi.org/10.1016/j.suscom.2026.101346>.
91. Vidhya, S.; Rejini, K.; Megala, G.; Raja, S.P.; Kumaran, U. Energy efficient transformer-federated multi-agent learning for adaptive demand-side management in IoT-enabled renewable smart grids. *Sustain. Comput. Inform. Syst.* **2026**, *50*, 101328. <https://doi.org/10.1016/j.suscom.2026.101328>.
92. Shrichandran, G.V.; Indhumathi, K.; Ahmad, S.; Alsubait, T.; Alabdulkreem, E.; Qamar, S.; El Shafie, H.; Ayoobkhan, M.U.A. Energy-efficient multi-agent hybrid GNN-XGBoost framework for IoT-enabled smart grid management with blockchain-secured urban energy sustainability. *Sustain. Comput. Inform. Syst.* **2026**, *50*, 101338. <https://doi.org/10.1016/j.suscom.2026.101338>.
93. Maheshwari, A.K. Federated Multi-Agent Reinforcement Learning for Incentive-Based DRS over Blockchain enabled Microgrids. In *2024 7th International Conference on Signal Processing and Information Security (ICSPIS)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6. <https://doi.org/10.1109/ICSPIS63676.2024.10812651>.
94. Zheng, Y.; Liu, B.; Fan, K.; Guo, M.; Hu, Z. Blockchain and Multi-Agent Reinforcement Learning Framework for V2G benefit optimization. In *2025 9th International Conference on Power Energy Systems and Applications (ICoPESA)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/ICoPESA65876.2025.11234340>.

95. Ul Hassan, M.; Rehmani, M.H.; Chen, J. VPT: Privacy Preserving Energy Trading and Block Mining Mechanism for Blockchain Based Virtual Power Plants. *ACM Trans. Priv. Secur.* **2025**, *29*, 1–35. <https://doi.org/10.1145/3767163>.
96. Guizani, A.M.S.; Abishu, H.N.; Yacob, Y.H.; Ayall, T.A.; Erbad, A. Blockchain-Based Resource Trading in Multi-UAV-Assisted Industrial IoT Networks: A Multi-Agent DRL Approach. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 166–181. <https://doi.org/10.1109/TNSM.2022.3197309>.
97. Shen, Q.; Hu, B.-J. Blockchain-Enabled Multi-Agent Deep Reinforcement Learning for Task Offloading in UAV-Assisted Vehicular Edge Computing. In *2025 IEEE 11th World Forum on Internet of Things (WF-IoT)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/WF-IoT64238.2025.11270493>.
98. Xie, Z.L.; Xu, M.; Nie, J.; Kang, J.; Chen, W. NOMA-Enabled Cooperative Computation Offloading for Blockchain-Empowered Internet of Things: A Learning Approach. *IEEE Internet Things J.* **2021**, *8*, 2364–2378. <https://doi.org/10.1109/JIOT.2020.3016644>.
99. Pei, P.; Li, M.; Huang, Y.; Yang, R.; Sun, Y.; Wang, Z. Joint Optimization and Scheduling of Computing and Caching Resource for Blockchain-Empowered IoT via CDRL Approach. In *2023 IEEE 11th International Conference on Information, Communication and Networks (ICICN)*; IEEE: Piscataway, NJ, USA, 2023. <https://doi.org/10.1109/ICICN59530.2023.10393482>.
100. Peng, C.; Li, J.; Zhu, Y. Cooperative Task Offloading Strategy for Utility Optimization in Blockchain-Enabled IoT Networks. In *2025 28th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/CSCWD64889.2025.11033324>.
101. Chen, Z.; Yi, J.; Yang, Y.; Zhou, Y. Making Computational Task Offloading Decision Based on Swarm Intelligence Enabled RL in Internet-of-Things Blockchain. In *ICC 2025-IEEE International Conference on Communications*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/ICC52391.2025.11161841>.
102. Islam, T.; Bappy, F.H.; Shaila Zaman, T.; Sajid, S.M.I.; Pritom, M.A. MRL-PoS: A Multi-agent Reinforcement Learning based Proof of Stake Consensus Algorithm for Blockchain. In *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*; IEEE: Piscataway, NJ, USA, 2024. <https://doi.org/10.1109/CCWC60891.2024.10427777>.
103. Bappy, F.H.; Islam, T.; Hasan, K.; Sajid, M.S.I.; Pritom, M.M.A. Securing Proof of Stake Blockchains: Leveraging Multi-Agent Reinforcement Learning for Detecting and Mitigating malicious Nodes. In *LOBECOM 2024—2024 IEEE Global Communications Conference*; IEEE: Piscataway, NJ, USA, 2024. <https://doi.org/10.1109/GLOBECOM52923.2024.10900961>.
104. Daliri, S.; Jassbi, S.J. Distributed reinforcement learning for heterogeneous blockchain networks: A customized multi-agent architecture for cross-chain optimization. *J. Supercomput.* **2026**, *82*, 4. <https://doi.org/10.1007/s11227-025-08060-6>.
105. Weyl, E.G.; Ohlhaber, P.; Buterin, V. *Decentralized Society: Finding Web3's Soul*; SSRN Working Paper; Elsevier: Amsterdam, The Netherlands, 2022. Available online: <https://ssrn.com/abstract=4105763> (accessed on 1 May 2026).
106. Goldin, M. Token-Curated Registries 1.0. Medium. 2018. Available online: <https://medium.com/@ilovebagels/token-curated-registries-1-0-61a232f8dac7> (accessed on 1 May 2026).
107. Strathern, M. 'Improving Ratings': Audit in the British University System. *Eur. Rev.* **1997**, *5*, 305–321.
108. Buterin, V. On Nathan Schneider on the Limits of Cryptoeconomics. Vitalik Buterin's Website. 2021. Available online: <https://vitalik.eth.limo/general/2021/09/26/limits.html> (accessed on 1 July 2026).
109. Tian, K. Blockchain-enhanced incentive-compatible mechanisms for multi-agent reinforcement learning systems. *Sci. Rep.* **2025**, *15*, 42841. <https://doi.org/10.1038/s41598-025-20247-8>.
110. Tang, X.; Lan, X.; Li, L.; Zhang, Y.; Han, Z. Incentivizing Proof-of-Stake Blockchain for Secured Data Collection in UAV-Assisted IoT: A Multi-Agent Reinforcement Learning Approach. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 3470–3484. <https://doi.org/10.1109/JSAC.2022.3213360>.
111. Fernandez, J.D.; Barbereau, T.; Papageorgiou, O. Agent-Based Model of Initial Token Allocations: Simulating Distributions post Fair Launch. *ACM Trans. Manag. Inf. Syst.* **2024**, *15*, 1–22. <https://doi.org/10.1145/3649318>.
112. Park, J.; Kim, K.; Kim, S.Y. Blockchain Unpacked: An Agent-Based Model of Economic Inequality and Its Implications for Technology Governance in Blockchain. In *ICIS 2021 Proceedings*; Association for Information Systems (AIS): Atlanta, GA, USA, 2021.
113. Demers, A.; Greene, D.; Hauser, C.; Irish, W.; Larson, J.; Shenker, S.; Sturgis, H.; Swinehart, D.; Terry, D. Epidemic Algorithms for Replicated Database Maintenance. In *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing (PODC '87), Vancouver, BC, Canada, 10–12 August 1987*; ACM: New York, NY, USA, 1987; pp. 1–12.
114. Poon, J.; Dryja, T. *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments*; Technical Report; Lightning Network: San Francisco, CA, USA, 2016. Available online: <https://lightning.network/lightning-network-paper.pdf> (accessed on 1 May 2026).
115. Coleman, J.; Horne, L.; Xuanji, L. Counterfactual: Generalized State Channels. *arXiv* **2018**, arXiv:1812.00922.
116. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017), Fort Lauderdale, FL, USA, 20–22 April 2017*; JMLR: New York, NY, USA, 2017; pp. 1273–1282.

117. Blanchard, P.; El Mhamdi, E.M.; Guerraoui, R.; Stainer, J. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 2017), Long Beach, CA, USA, 4–9 December 2017*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 119–129.
118. Chen, S.; Zhang, J.; Bai, Y.; Xu, P.; Gao, T.; Jiang, H.; Gao, W.; Li, X. Blockchain Enabled Intelligence of Federated Systems (BELIEFS): An attack-tolerant trustable distributed intelligence paradigm. *Energy Rep.* **2021**, *7*, 8900–8911. <https://doi.org/10.1016/j.egy.2021.10.113>.
119. Liu, Z.; Huang, B.; Li, Y.; Zhang, C.; Li, T.; Sun, Q.; Gao, D.W. Blockchain and Federated Learning in P2P Energy Trading: Privacy Protection and Prosumer Incentives. *IEEE Trans. Ind. Inform.* **2025**, *21*, 9446–9457. <https://doi.org/10.1109/TII.2025.3598484>.
120. Liu, B.W.; Tian, Z.; Tang, F.; Pan, H.; She, W. Blockchain-Empowered Asynchronous Federated Reinforcement Learning for IoT-Based Traffic Trajectory Prediction. *IEEE Internet Things J.* **2025**, *12*, 17095–17109. <https://doi.org/10.1109/JIOT.2025.3538887>.
121. Shin, T.C.; Han, K. Blockchain-Based Decentralized Edge Intelligence Collaboration and Adaptive Incentive Mechanism Research. *IEEE Internet Things J.* **2026**, *13*, 4088–4098. <https://doi.org/10.1109/JIOT.2025.3609779>.
122. Sankaran, M.; Chaudhary, A.; Kumar, B.D.; Rajesh, N.; Surabhi, K.R. Federated Multi-Agent Reinforcement Learning for Secure and Scalable Blockchain Consensus. In *2025 2nd International Conference on Software, Systems and Information Technology (SSITCON)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/SSITCON66133.2025.11342104>.
123. Delvaux, M. *Report with Recommendations to the Commission on Civil Law Rules on Robotics*; European Parliament: Strasbourg, France, 2017. Available online: https://www.europarl.europa.eu/doceo/document/A-8-2017-0005_EN.html (accessed on 1 May 2026).
124. Matthias, A. The Responsibility Gap: Ascribing Responsibility for the Actions of Learning Automata. *Ethics Inf. Technol.* **2004**, *6*, 175–183.
125. Danaher, J. Algorithmic Governance and the Need for Consumer Empowerment in Data-Driven Markets. *Internet Policy Rev.* **2016**, *5*, 1–18.
126. Zittrain, J. *The Future of the Internet and How to Stop It*; Yale University Press: New Haven, CT, USA, 2008.
127. Faqir-Rhazoui, Y.; Arroyo, J.; Hassan, S. A Comparative Analysis of the Platforms for Decentralized Autonomous Organizations in the Ethereum Blockchain. *J. Internet Serv. Appl.* **2021**, *12*, 9.
128. Han, J.; Lee, J.; Li, T. DAO Governance. *J. Corp. Financ.* **2025**, *91*, 102734.
129. Beanstalk Farms. Beanstalk Governance Exploit: Post-Mortem Report. 2022. Available online: <https://bean.money/blog/beanstalk-governance-exploit> (accessed on 1 May 2026).
130. Lalley, S.P.; Weyl, E.G. Quadratic Voting: How Mechanism Design Can Radicalize Democracy. *AEA Pap. Proc.* **2018**, *108*, 33–37.
131. Suchman, M.C. Managing Legitimacy: Strategic and Institutional Approaches. *Acad. Manag. Rev.* **1995**, *20*, 571–610. <https://doi.org/10.5465/amr.1995.9508080331>.
132. Lesaege, C.; Ast, F.; George, W. *Kleros Short Paper v1.0.7*; Technical Report; Kleros: Paris, France, 2019. Available online: <https://kleros.io/assets/whitepaper.pdf> (accessed on 1 July 2026).
133. Aragon Association. Aragon Court: A Subjective Oracle for Decentralized Organizations. 2020. Available online: <https://github.com/aragon/aragon-court> (accessed on 1 May 2026).
134. Cu, M.; Chan, J.; Peko, G.; Sundaram, D. Blockchain-Based Governance in Fractional Ownership: Mitigating Zero-Sum Games Through Decentralized Autonomous Agents. In *Context-Aware Systems and Applications*; Lecture Notes of the Institute for Computer Sciences; Springer: Cham, Switzerland, 2021; Volume 409. https://doi.org/10.1007/978-3-030-93179-7_19.
135. Cu, M.; Peko, G.; Chan, J.; Sundaram, D. Blockchain-based Governance, Risk Management, and Compliance for Fractional Ownership: Design and Implementation of a Decentralized Autonomous Agent System. In *Proceedings of the 56th Hawaii International Conference on System Sciences (HICSS), Maui, HI, USA, 3–6 January 2023*; pp. 3870–3879. <https://doi.org/10.24251/hicss.2023.473>.
136. Lee, C.A.; Chow, K.M.; Chan, H.A.; Lun, D.P.K. Decentralized governance and artificial intelligence policy with blockchain-based voting in federated learning. *Front. Res. Metrics Anal.* **2023**, *8*, 1035123. <https://doi.org/10.3389/frma.2023.1035123>.
137. Pavón, V.M.C. Web3-DAO: An ontology for decentralized autonomous organizations. *J. Web Semant.* **2024**, *82*, 100830. <https://doi.org/10.1016/j.websem.2024.100830>.
138. Virovets, D.; Obushnyi, S.; Shtepa, O.; Hulak, H.; Vlasenko, V. Ways of Interaction of Autonomous Economic Agents in Decentralized Autonomous Organizations. *Cybersecur. Provid. Inf. Telecommun. Syst.* **2023**, *3421*, 183–190.
139. Ikeda, Y.; Hadfi, R.; Ito, T.; Fujihara, A. Anomaly detection and facilitation AI to empower decentralized autonomous organizations for secure crypto-asset transactions. *AI Soc.* **2025**, *40*, 3999–4010. <https://doi.org/10.1007/s00146-024-02166-w>.
140. National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*; Technical Report NIST AI 100-1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023.
141. Otalora, F.C.; Themistocleous, M. Regulatory Challenges of LLM-Integrated Blockchain Agents in the Metaverse. In *2025 3rd International Conference on Intelligent Metaverse Technologies & Applications (iMETA)*; IEEE: Piscataway, NJ, USA, 2025; pp. 33–38. <https://doi.org/10.1109/iMETA66706.2025.11306431>.

142. Dwivedi, Y.K.; Hughes, L.; Baabdullah, A.M.; Ribeiro-Navarrete, S.; Giannakis, M.; Al-Debei, M.M.; Dennehy, D.; Metri, B.; Buhalis, D.; Cheung, C.M.K.; et al. Metaverse Beyond the Hype: Multidisciplinary Perspectives on Emerging Challenges, Opportunities, and Agenda for Research, Practice and Policy. *Int. J. Inf. Manag.* **2022**, *66*, 102542. <https://doi.org/10.1016/j.ijinfomgt.2022.102542>.
143. Louck, Y.; Dvir, A.; Stulman, A. Security Analysis of Agentic AI Communication Protocols: A Comparative Evaluation. *ACM Trans. AI Secur. Priv.* **2026**, online ahead of print. Available online: <https://dl.acm.org/doi/abs/10.1145/3803431> (accessed on 1 July 2026).
144. Sekar, A. Autonomous Administrative Intelligence: Governing AI-Mediated Administration in Decentralized Organizations. *Adm. Sci.* **2026**, *16*, 95. <https://doi.org/10.3390/admsci16020095>.
145. Vivekanand, C.V.; Kushbu, C.S.; Kalaivani, S.; Inbamalar, T.M.; Selvam, A.J.M.; Anna, D.E. Enhancing Corporate Sustainability through Agentic AI and Blockchain Technologies. In *2025 IEEE First International Conference on Innovations in Engineering and Next-Generation Technologies for Sustainability (ICINVENTS)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/ICINVENTS64613.2025.11402608>.
146. Jensen, M.C.; Meckling, W.H. Theory of the Firm: Managerial Behavior, Agency Costs and Ownership Structure. *J. Financ. Econ.* **1976**, *3*, 305–360.
147. Krakovna, V.; Uesato, J.; Mikulik, V.; Martic, M.; Togelius, J.; Xu, Z.; Powell, J.; Beirlant, A.; Everitt, T.; Leike, J.; et al. Avoiding Side Effects in Complex Environments. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 2020), Virtual Event, 6–12 December 2020*; Curran Associates, Inc.: Red Hook, NY, USA, 2020.
148. Calvano, E.; Calzolari, G.; Denicolò, V.; Pastorello, S. Artificial Intelligence, Algorithmic Pricing, and Collusion. *Am. Econ. Rev.* **2020**, *110*, 3267–3297. <https://doi.org/10.1257/aer.20190623>.
149. Rochet, J.C.; Tirole, J. Platform Competition in Two-Sided Markets. *J. Eur. Econ. Assoc.* **2003**, *1*, 990–1029.
150. Parker, G.; Van Alstyne, M.; Choudary, S.P. *Platform Revolution: How Networked Markets Are Transforming the Economy*; W.W. Norton & Company: New York, NY, USA, 2016.
151. Eisenmann, T.; Parker, G.; Van Alstyne, M. Strategies for Two-Sided Markets. *Harv. Bus. Rev.* **2006**, *84*, 92–101.
152. Kumar, K.; Van Dissel, H.G. Sustainable Collaboration: Managing Conflict and Cooperation in Interorganizational Systems. *MIS Q.* **1996**, *20*, 279–300. <https://doi.org/10.2307/249657>.
153. Clemons, E.K. Limits to Interfirm Coordination through Information Technology: Results of a Field Study in Consumer Packaged Goods Distribution. *J. Manag. Inf. Syst.* **1993**, *10*, 73–95. <https://doi.org/10.1080/07421222.1993.11517991>.
154. Iansiti, M.; Lakhani, K.R. The Truth About Blockchain. *Harv. Bus. Rev.* **2017**, *95*, 118–127.
155. Bakos, J.Y. A Strategic Analysis of Electronic Marketplaces. *MIS Q.* **1991**, *15*, 295–310.
156. Kshetri, N. Blockchain's Roles in Meeting Key Supply Chain Management Objectives. *Int. J. Inf. Manag.* **2018**, *39*, 80–89.
157. Jacobides, M.G.; Cennamo, C.; Gawer, A. Towards a Theory of Ecosystems. *Strateg. Manag. J.* **2018**, *39*, 2255–2276.
158. Wang, H.; Hunhevicz, J.J.; Hall, D.M. From automation to agency: Prototype for self-owning intelligent buildings enabled by blockchain. *Autom. Constr.* **2025**, *177*, 106309. <https://doi.org/10.1016/j.autcon.2025.106309>.
159. Gürpınar, T. Towards web 4.0: Frameworks for autonomous AI agents and decentralized enterprise coordination. *Front. Blockchain* **2025**, *8*, 1591907. <https://doi.org/10.3389/fbloc.2025.1591907>.
160. De Gasperis, G.; Facchini, S.D.; Letteri, I. Leveraging Multi-Agent Systems and Decentralised Autonomous Organisations for Tax Credit Tracking: A Case Study of the Superbonus 110% in Italy. *Appl. Sci.* **2024**, *14*, 10622. <https://doi.org/10.3390/app142210622>.
161. Soni, K.S.; Singh, V.P.; Mahajan, A.; Singh, J.; Singh, A. The Architecture of AI Agents in Web3: Decentralized Networks to Semi-Autonomous Systems. In *2025 IEEE DELCON-International Conference on Recent Smart Technologies in Engineering for Sustainable Development*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6. <https://doi.org/10.1109/DELCON68055.2025.11400132>.
162. Qi, M.; Zhu, T.; Zhang, L.; Li, N.; Tan, Y.-a.; Zhou, W. Toward Transparent and Incentive-Compatible Collaboration in Decentralized LLM Multi-Agent Systems: A Blockchain-Driven Approach. *IEEE Trans. Netw. Sci. Eng.* **2026**, *13*, 6706–6720. <https://doi.org/10.1109/TNSE.2026.3659486>.
163. DeFiLlama. DeFi TVL Analytics. 2025. Available online: <https://defillama.com> (accessed on 1 May 2026).
164. Craib, R.; Bradway, G.; Dunn, M.; Treister, A. Numerai: A Crowd-Sourced Hedge Fund. 2020. Available online: <https://numer.ai/whitepaper.pdf> (accessed on 1 May 2026).
165. Li, J.; Xie, L.; Huang, H.; Zhou, B.; Song, B.; Zeng, W.; Deng, X.; Zhang, X. Survey on Strategic Mining in Blockchain: A Reinforcement Learning Approach. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*; ACM: New York, NY, USA, 2025. <https://doi.org/10.24963/ijcai.2025/1170>.
166. Arcifa, R.; Qiao, Y.; Lee, B.; Ye, Y. Bounded LLM Orchestration with Deterministic Safety Gates for Concentrated Liquidity Management. *IEEE Access* **2026**, *14*, 72833–72848. <https://doi.org/10.1109/ACCESS.2026.3690467>.
167. Luo, Y.; Feng, Y.; Xu, J.; Liu, Y. Resisting Manipulative Bots in Meme Coin Copy Trading: A Multi-Agent Approach with Chain-of-Thought Reasoning. In *Proceedings of the ACM Web Conference 2026, WWW '26*; ACM: New York, NY, USA, 2026; pp. 3414–3425. Available online: <https://dl.acm.org/doi/abs/10.1145/3774904.3792635> (accessed on 1 July 2026).

168. Thomaidis, I.T.; Giannopoulos, P.G.; Dasaklis, T.K. Coordinating Artificial Intelligence and Bitcoin Mining Workloads: A Risk-Aware Distributional Multi-Agent Reinforcement Learning Approach. In *2025 16th International Conference on Information, Intelligence, Systems & Applications (IISA)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/IISA66859.2025.11311340>.
169. Khan, H.U.; Khan, M.A.R.; Ali, F. Systematic Mapping Study of Blockchain Integrated Supply Chain Management. *Secur. Commun. Netw.* **2024**, 2024, 8884339. <https://doi.org/10.1155/2024/8884339>.
170. IBM. IBM Food Trust: A Smarter, Safer, More Sustainable Food System. 2020. Available online: <https://www.ibm.com/blockchain/solutions/food-trust> (accessed on 1 May 2026).
171. A.P. Moller–Maersk and IBM to Discontinue TradeLens, a Blockchain-Enabled Global Trade Platform. Maersk Press Release. 2022. Available online: <https://www.maersk.com/news/articles/2022/11/29/maersk-and-ibm-to-discontinue-tradelens> (accessed on 1 May 2026).
172. Dutta, P.; Choi, T.M.; Somani, S.; Butala, R. Blockchain Technology in Supply Chain Operations: Applications, Challenges and Research Opportunities. *Transp. Res. Part E Logist. Transp. Rev.* **2020**, 142, 102067.
173. Swain, S.; Patra, M.R. A Distributed Agent-Oriented Framework for Blockchain-Enabled Supply Chain Management. In *2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–7. <https://doi.org/10.1109/ICBDS53701.2022.9936015>.
174. Swain, S.; Patra, M.R. A Distributed Software Agent-centric Framework for Supply Chain Networks Empowered by Blockchain: Insights into Smart Contracts. *Oper. Res. Forum* **2024**, 5, 40. <https://doi.org/10.1007/s43069-024-00318-8>.
175. Swain, S.; Patra, M.R. A Distributed Software Agent-Oriented Traceability Milieu for Blockchain-Enabled Supply Chain. In *2022 5th International Conference on Computational Intelligence and Networks (CINE)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6. <https://doi.org/10.1109/CINE56307.2022.10037458>.
176. Kumar, P.S.; Pankajam, A.; Veeraiah, V.; Gharat, A.; Mamatha, G.; Roy, D.R.; Gupta, A.; Rajitha, K. A Blockchain-Driven Multi-Agent Framework for Enhancing Transparency, Efficiency, and Resilience in Smart Supply Chain Management. *J. Theor. Appl. Inf. Technol.* **2026**, 104, 319–332. <https://doi.org/10.5281/zenodo.18823863>.
177. Giannopoulos, P.G.; Malamas, V.; Dasaklis, T.K. Decentralized pricing in supply chain management: A blockchain-enabled multi-agent Reinforcement Learning approach. In *2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/CoDIT66093.2025.11321740>.
178. Velastegui, R.; Poler, R.; Díaz-Madroñero, M. Conceptual Model FOR the Scheduling and Control of Production and Logistics Operations Using Multi-Agent Robotic Systems and Blockchain. *DYNA* **2023**, 98, 307–313. <https://doi.org/10.6036/10724>.
179. Yang, R.; Hou, T.; Zhang, E.; Qi, M. BlkMaC: A Blockchain-based Multi-Agents System with LLM Communication Framework. In *Proceedings of the 2025 9th International Conference on Computer Science and Artificial Intelligence, CSAI '25*; ACM: New York, NY, USA, 2026; pp. 84–90. <https://doi.org/10.1145/3788149.3788229>.
180. Ioannides, G.; Constantinou, C.; Nija, V.; Jain, A.; Allyson, E. Mod-X: A Modular Open Decentralized eXchange Framework Proposal for Heterogeneous Interoperable Artificial Intelligence Agents. In *AI Agent for Information Retrieval: Generating and Ranking*; Communications in Computer and Information Science; Springer: Cham, Switzerland, 2026; Volume 2695. https://doi.org/10.1007/978-3-032-11477-8_7.
181. Rahmani, L.; Minarsch, D.; Ward, J. Peer-to-peer Autonomous Agent Communication Network. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '21, Online, 3–7 May 2021*; International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS): Pittsburgh, PA, USA, 2021; pp. 1037–1045.
182. Ragupathy, V. Secure Agent-Based Architectures for Decentralized AI Identity Management: The DAIS Framework. In *2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC)*; IEEE: Piscataway, NJ, USA, 2026; pp. 1–6. <https://doi.org/10.1109/ICAIC67076.2026.11395719>.
183. Sümer, M.Ş.; Özsoy, T.; Okay, F.Y.; Kök, İ. Smart Agents as Customers: A Semi-Autonomous Digital Commerce Model for Industry 5.0. In *2025 7th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (ICHORA)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/ICHORA65333.2025.11017052>.
184. Ocean Protocol Foundation. Ocean Protocol: Tools for the Web3 Data Economy. 2022. Available online: <https://oceanprotocol.com/tech-whitepaper.pdf> (accessed on 1 July 2026).
185. Popov, S. *The Tangle*; Technical Report; IOTA Foundation: Berlin, Germany, 2018. Available online: <https://whitepaper.io/coin/iota> (accessed on 1 July 2026).
186. Pan, J.; Wang, J.; Hester, A.; Alqerm, I.; Liu, Y.; Zhao, Y. EdgeChain: An Edge-IoT Framework and Prototype Based on Blockchain and Smart Contracts. *IEEE Internet Things J.* **2019**, 6, 4719–4732.
187. Jia, Z.; He, S.; Zhu, Q.; Wang, W.; Wu, Q.; Han, Z. Trusted Routing for Blockchain-Empowered UAV Networks via Multi-Agent Deep Reinforcement Learning. *IEEE Trans. Commun.* **2025**, 73, 14227–14242. <https://doi.org/10.1109/TCOMM.2025.3597655>.
188. He, S.; Jia, Z.; Zhu, Q.; Zhou, F.; Wu, Q. Trusted Routing for Blockchain-Enabled Low-Altitude Intelligent Networks. In *2025 IEEE/CIC International Conference on Communications in China (ICCC)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/ICCC65529.2025.11149011>.

189. Hafeez, S.; Cheng, R.; Mohjazi, L.; Imran, M.A.; Sun, Y. A Blockchain-Enabled Framework of UAV Coordination for Post-Disaster Networks. In *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*; IEEE: Piscataway, NJ, USA, 2024. <https://doi.org/10.1109/VTC2024-Spring62846.2024.10683259>.
190. Kang, J.; Chen, J.; Xu, M.; Xiong, Z.; Jiao, Y.; Han, L.; Niyato, D.; Tong, Y.; Xie, S. UAV-Assisted Dynamic Avatar Task Migration for Vehicular Metaverse Services: A Multi-Agent Deep Reinforcement Learning Approach. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 430–445. <https://doi.org/10.1109/JAS.2023.123993>.
191. Chelghoum, M.; Bendiab, G.; Benmohammed, M.; M.-A., L.; Bousalem, S.; Mellouk, A. Securing 6G-enabled vehicle-to-everything communications: A blockchain-enabled collaborative intrusion detection framework with reinforcement learning. *Comput. Netw.* **2025**, *270*, 111568. <https://doi.org/10.1016/j.comnet.2025.111568>.
192. Tsang, Y.P.; Lee, C.K.M.; Zhang, K.; Wu, C.H.; Ip, W.H. On-Chain and Off-Chain Data Management for Blockchain-Internet of Things: A Multi-Agent Deep Reinforcement Learning Approach. *J. Grid Comput.* **2024**, *22*, 16. <https://doi.org/10.1007/s10723-023-09739-x>.
193. Antuley, U.; Siddiqui, S.; Hameed, S.; Arif, W.; Shah, S.A. SORA-ATMAS: Adaptive trust management and multi-LLM aligned governance for future smart cities. *Knowl.-Based Syst.* **2026**, *337*, 115403. <https://doi.org/10.1016/j.knosys.2026.115403>.
194. Zhan, Z.; Li, K.; Zhang, Y.; Haddadi, H. Systems-Level Attack Surface of Edge Agent Deployments on IoT. In *Proceedings of the Sixth European Workshop on Machine Learning and Systems, EuroMLSys '26*; ACM: New York, NY, USA, 2026; pp. 99–108. <https://doi.org/10.1145/3805621.3807636>.
195. Kapengut, E.; Mizrach, B. An Event Study of the Ethereum Transition to Proof-of-Stake. *Commodities* **2023**, *2*, 96–110. <https://doi.org/10.3390/commodities2020006>.
196. Goes, C. *IBC: Inter-Blockchain Communication Protocol*; Tendermint; Interchain Foundation: Zug, Switzerland, 2020. Available online: <https://github.com/cosmos/ibc> (accessed on 1 May 2026).
197. Zarick, R.; Pellegrino, B.; Banister, C. *LayerZero: Trustless Omnichain Interoperability Protocol*; Technical Report; LayerZero Labs: Vancouver, BC, Canada, 2022. Available online: https://layerzero.network/pdf/LayerZero_Whitepaper_Release.pdf (accessed on 1 May 2026).
198. Gularso, S.N.; Wiputra, R. Are Autonomous Agents the Inevitable Future of Decentralized Finance? A Review and Forward Outlook. In *2025 17th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*; IEEE: Piscataway, NJ, USA, 2025. <https://doi.org/10.1109/IHMSC66529.2025.00038>.
199. Lu, X.S.; Wan, J.; Liu, C.; Li, L.; Wang, X. Carbon-Aware Distributed Energy Management for Data Center Microgrids Based on Blockchain. In *2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*; IEEE: Piscataway, NJ, USA, 2024; pp. 2274–2275. <https://doi.org/10.1109/ISPA63168.2024.00324>.
200. Pan, Q.; Wu, J.; Li, J.; Yang, W.; Guizani, M. Blockchain and Multi-Agent Learning Empowered Incentive IRS Resource Scheduling for Intelligent Reconfigurable Networks. *IEEE/ACM Trans. Netw.* **2023**, *32*, 943–958. <https://doi.org/10.1109/tnet.2023.3309729>.
201. Sifakis, N. Global Virtual Prosumer Framework for Secure Cross-Border Energy Transactions Using IoT, Multi-Agent Intelligence, and Blockchain Smart Contracts. *Information* **2026**, *17*, 396. <https://doi.org/10.3390/info17040396>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.