Journal of Secure Digital Technologies
Volume 1 (№2) September 2026

ISSN (E): 3136-9340
Sertificate № KZ19VPY00128988, date of issue: 25.12.2025

The journal is published 4 times a year
(March, June, September, December)

EDITORIAL TEAM
Editor-in-Chief
Dr. Leila Rzayeva, Association prof., PhD, Digital government support center (Kazakhstan)

Deputy Editor-in-Chief
Dr. Ali Myrzatay, PhD, Digital government support center (Kazakhstan)

Executive Secretary
Dr. Abdenova Zhuldyz, PhD, Digital government support center (Kazakhstan)

EDITORIAL BOARD MEMBERS (KAZAKHSTAN)
Seitkulov E.N. – Candidate of Physical and Mathematical Sciences, Professor Computer Science, Computer Engineering and Control
Abdenova G.A. – PhD in Automation and Control, Candidate of Technical Sciences, Professor L.N. Gumilyov Eurasian National University
Atanov S.K. – Doctor of Technical Sciences, Professor.
Sergazin G.K. – PhD, Associate Professor Computer Science, Computer Engineering and Control
Satybaldina D.Zh. – Candidate of Physical and Mathematical Sciences, Professor Director, Research Institute of Information Security and Cryptology, L.N. Gumilyov Eurasian National University

INTERNATIONAL EDITORIAL BOARD MEMBERS
Ibraheem Shayea – PhD in Electrical and Electronic Engineering Istanbul Technical University (Turkey)
Yadigar Nasiboglu Imamverdiyev – DSc in Technical Sciences, Associate Professor Information Security, Computer Science and Computer Technology (Azerbaijan)
Erdal Irmak – PhD in Electrical and Electronics Engineering, Professor Gazi University (Turkey)
Korhan Kayisli – PhD in Electrical Electronics Engineering, Associate Professor Gazi University (Turkey)
Yonghao Wang – PhD in Electronic Engineering, Associate Professor Queen Mary, University of London (UK)
Amir Mosavi – PhD in Applied Informatics, Professor of AI Obuda University (Hungary)
Gerald Feldman – PhD in General Computing, Associate Professor Birmingham City University (UK)
Mohamed Ragab – PhD in Computer Science, Assistant Professor Birmingham City University (UK)
Seref Sagiroglu – PhD in System Engineering, Professor Gazi University (Turkey)

MANAGING EDITOR
Dr. Ali Myrzatay, PhD, Digital government support center (Kazakhstan)

TECHNICAL EDITOR
Dr. Ali Myrzatay, PhD, Digital government support center (Kazakhstan)

CONTENTS

Cyber Incident Damage Assessment for Public Digital Systems and Critical Infrastructure 3

Cyber Incident Damage Assessment for Public Digital Systems and Critical Infrastructure 37

Verifiable but Not Correct: Blockchain Trust Boundaries for Autonomous Agents 71

Calibrated Digital Twins and Geometry-based Perception for Automated Driving in Heavy Rain and Snowfall: Evidence From Research Testing in China and Implications for Kazakhstan. 88

Personal Data Protection in Kazakhstan's AI Enabled Public Services 109

Asking Another Jurisdiction What It Holds Without Telling It What You Hold: Private Artifact Matching with a Ledger-Anchored Transcript 131

On-Chain and Off-Chain Data Placement in Hybrid Blockchain Health Systems 143

An Integrated Project Management Maturity Model for Digital Qazaqstan Initiatives: Embedding Cybersecurity, AI Governance and Forward-Looking Cyber-Risk Requirements into ST RK ISO 21502-2022..... 158

Appendix A. Indicators and evidence for the baseline practice groups 181

Privacy Risks and Data Protection Readiness in AI Marketing in Kazakhstan..... 183

Supplementary Materials..... 205

Yan Pinchuk*
Bachelor's degree, Master's student, Project Management programme, Astana IT University; Director, Business Rost LLP
yan@pinchuk.kz, ORCID: 0009-0007-7483-5499
Astana IT University; Business Rost LLP, Astana, Kazakhstan
Anar Yessengeldina
Candidate of Economics, Professor, School of Digital Public Administration
A.Yessengeldina@astanait.edu.kz, ORCID: 0000-0002-7752-916X
Astana IT University, 55/1 Mangilik El Avenue, 010000 Astana, Kazakhstan

VERIFIABLE BUT NOT CORRECT: BLOCKCHAIN TRUST BOUNDARIES FOR AUTONOMOUS AGENTS

Abstract: Blockchain is increasingly proposed as a trust layer for autonomous agents based on large language models, yet a valid ledger record does not show that the recorded action was correct. This study asks which trust claims in such systems a ledger can verify on its own. Thirty-one blockchain-enabled agent systems published in 2018–2026, identified through a documented two-round search, were coded on six trust functions, eight types of external dependency, and three verification levels: claims verifiable from ledger evidence alone, claims verifiable only under named external assumptions, and claims about truth, correctness, or safety that no ledger evidence can establish. Two independent codings, both assisted by language models and reconciled by the authors, agreed substantially (Cohen's kappa 0.68 for functions and 0.67 for levels). Coordination (21 systems) and provenance (19) were the most frequent functions. Of 93 function-level claims, 19 were verifiable from the ledger alone, 68 required external assumptions, and 6 concerned truth or safety; all ledger-only claims concerned provenance. Every system relied on at least one external assumption (median 4 of 8 types), and 21 implied benefits that no ledger can establish. Systems based on language models depended on an evaluator's judgement far more often than other systems (11 of 18 against 0 of 13). Eight documented incidents and three empirical studies, including agents that signed unintended transfers and agent registries dominated by fake reviewers, show failures exactly at these assumptions while ledger evidence remained valid. Zero-knowledge proofs and trusted hardware relocate rather than remove the boundary. We propose a Trust-Boundary Manifest, a machine-readable specification with a composition rule and a reference checker, and show on a constructed example and three published systems that it exposes mislabelled "trustless" claims and the full set of trusted parties.

Keywords: *blockchain; autonomous agents; large language models; trust boundary; smart contracts; reputation systems; verifiable computation; agent accountability*

Проверяемо, но Не Обязательно Верно: Границы Доверия Блокчейна для Агентов

Аннотация: Блокчейн всё чаще предлагают как слой доверия для автономных агентов на основе больших языковых моделей, однако действительная запись в реестре не доказывает, что записанное действие было правильным. Исследование отвечает на вопрос, какие утверждения о доверии в таких системах реестр может проверить самостоятельно. Тридцать одна система агентов с блокчейном, опубликованная в 2018–2026 гг. и отобранная в ходе документированного двухэтапного поиска, размечена по шести функциям доверия, восьми типам внешних зависимостей и трём уровням проверяемости: утверждения, проверяемые только по данным реестра; утверждения, проверяемые лишь при явно названных внешних допущениях; утверждения об истинности, правильности или безопасности, которые данные реестра установить не могут. Две независимые разметки, выполненные с помощью языковых моделей и согласованные авторами, показали существенное согласие (коэффициент каппа Коэна 0,68 для функций и 0,67 для уровней). Чаще всего встречались функции координации (21 система) и происхождения данных (19). Из 93 утверждений на уровне функций 19 проверялись только по реестру, 68 требовали внешних допущений, 6 касались истинности или безопасности; все утверждения, проверяемые только по реестру, относились к происхождению данных. Каждая система опиралась хотя бы на одно внешнее допущение (медиана – 4 типа из 8), а 21 система подразумевала преимущества, которые реестр установить не может. Системы на основе языковых моделей значительно чаще зависели от суждения оценщика (11 из 18 против 0 из 13). Восемь задокументированных инцидентов и три эмпирических исследования, включая непреднамеренные переводы, подписанные агентами, и реестры агентов с преобладанием фиктивных отзывов, показывают сбои именно в этих допущениях при действительных данных реестра. Доказательства с нулевым разглашением и доверенное оборудование переносят границу доверия, а не устраняют её. Предложен манифест границ доверия – машиночитаемая спецификация с правилом композиции и программой проверки; на сконструированном примере и трёх опубликованных системах показано, что он выявляет ошибочно заявленную «бездоверительность» и полный набор доверенных сторон.

Ключевые слова: *блокчейн; автономные агенты; большие языковые модели; граница доверия; смарт-контракты; репутационные системы; проверяемые вычисления; подотчётность агентов*

Тексерілетін, Бірақ Дұрыс Болуы Шарт Емес: Агенттер Үшін Блокчейн Сенімі

Аңдатпа: Блокчейн үлкен тілдік модельдерге негізделген автономды агенттер үшін сенім қабаты ретінде барған сайын жиі ұсынылуда, алайда тізілімдегі жарамды жазба жазылған әрекеттің дұрыс болғанын дәлелдемейді. Зерттеу мұндай жүйелердегі сенім туралы қандай тұжырымдарды тізілім өз бетінше тексере алатынын анықтайды.

Құжатталған екі кезеңді іздеу арқылы іріктелген, 2018–2026 жылдары жарияланған блокчейнді қолданатын отыз бір агенттік жүйе алты сенім функциясы, сыртқы тәуелділіктің сегіз түрі және тексерілудің үш деңгейі бойынша кодталды: тек тізілім деректері бойынша тексерілетін тұжырымдар; тек нақты аталған сыртқы болжамдар орындалғанда тексерілетін тұжырымдар; тізілім деректері дәлелдей алмайтын ақиқаттық, дұрыстық немесе қауіпсіздік туралы тұжырымдар. Тілдік модельдердің көмегімен жасалып, авторлар келістірген екі тәуелсіз кодтау айтарлықтай сәйкестік көрсетті (Коэн каппа коэффициенті функциялар үшін 0,68, деңгейлер үшін 0,67). Ең жиі кездескен функциялар – үйлестіру (21 жүйе) және деректердің шығу тегі (19). Функция деңгейіндегі 93 тұжырымның 19-ы тек тізілім арқылы тексерілді, 68-і сыртқы болжамдарды қажет етті, 6-уы ақиқаттыққа немесе қауіпсіздікке қатысты болды; тек тізілім арқылы тексерілетін тұжырымдардың барлығы деректердің шығу тегіне қатысты. Әр жүйе кемінде бір сыртқы болжамға сүйенді (медиана – 8 түрдің 4-еуі), ал 21 жүйе тізілім дәлелдей алмайтын артықшылықтарды меңзеді. Тілдік модельдерге негізделген жүйелер бағалаушының бағалауына басқа жүйелерге қарағанда едәуір жиі тәуелді болды (18 жүйенің 11-інде, 13 жүйенің ешқайсысында емес). Құжатталған сегіз оқиға мен үш эмпирикалық зерттеу, соның ішінде агенттер қол қойған көзделмеген аударымдар мен жалған пікірлер басым болған агенттер тізілімдері, тізілім деректері жарамды күйінде қалғанымен, дәл осы болжамдарда ақаулар болатынын көрсетеді. Нөлдік білім дәлелдері мен сенімді аппараттық құралдар сенім шекарасын жоймайды, тек басқа жерге ауыстырады. Сенім шекаралары манифесті – композиция ережесі мен тексеру бағдарламасы бар машинамен оқылатын спецификация ұсынылды; ол құрастырылған мысалда және жарияланған үш жүйеде қате мәлімделген «сенімсіз» тұжырымдар мен сенім артылған тараптардың толық жиынын анықтайтыны көрсетілді.

Түйін сөздер: *блокчейн; автономды агенттер; үлкен тілдік модельдер; сенім шекарасы; смарт-келісімшарттар; бедел жүйелері; тексерілетін есептеулер; агенттердің есептілігі*

1. Introduction (Literature Review)

Agents built on large language models (LLMs) plan, call tools, and delegate tasks to other agents [1]; through emerging payment protocols they also pay for services on behalf of their principals [2], [3]. When agents operated by different organisations interact, no single platform can vouch for them. A counterpart may misstate its capabilities, act under a stolen key, return fabricated results, or deny an action. Blockchain is widely proposed as the missing trust layer: a replicated ledger gives participants an append-only record that no single party can rewrite, signatures bind messages to keys, hashes make later modification detectable, and smart contracts execute agreed rules deterministically [4]–[6]. The draft Ethereum standard ERC-8004 already defines on-chain identity, reputation, and validation registries for "trustless agents" [7].

Literature review. Work on blockchain-based multi-agent systems (MAS) has moved through three phases. Early systems stored reputation and service evaluations in permissioned ledgers or encoded role-based access between software agents in smart contracts (see Supplementary File S2). Swarm-robotics research used contracts that filter outlying observations and penalise their senders, which contained Byzantine robots in experiments with real robots [8]. Since 2024, LLM-based systems have put agent registries, task allocation, provenance records, reputation, and slashing on chain [9]–[11]. Surveys of the field converge on two observations: deployed designs separate off-chain reasoning from on-chain settlement, a separation Touloupou and Kapassa call "a trust gap" [5], and blockchain is "a shared trust layer, not a substitute for agent security, semantic verification, privacy, or robust reasoning" [6]. Security research explains why. The oracle problem shows that a contract can process an external value exactly as specified while its correctness rests on the supplier [12], [13]; deterministic execution does not make contract logic correct [14]; and LLM agents add hallucination [15] and prompt-injection failures that occur even when the agent holds valid credentials [16]. Authorisation of delegated agents can also be achieved without any ledger [17].

Research gap. The two strands rarely meet. Architecture papers present blockchain as making agents trustworthy, while security papers explain where ledger evidence ends. There is no systematic, reproducible account of which trust claims in the agent systems actually proposed rest on ledger evidence alone, which rest on named external assumptions, and which cannot be supported by a ledger at all.

Research questions. This study addresses the question of what blockchain verifies in agent systems. Specifically, it examines: RQ1, which trust functions blockchain-enabled agent systems assign to the ledger, and whether this differs between classical, embodied, and LLM-based agents; RQ2, whether the claim attached to each function is verifiable from ledger evidence alone, and which external assumptions recur; RQ3, which assumptions have failed in documented incidents while ledger evidence stayed valid; and RQ4, how the residual trust behind an agent claim can be stated and checked.

Scientific novelty. The scientific novelty of this work consists in a claim-level account of blockchain trust in agent systems, which distinguishes it from prior surveys in that it classifies the verification level of each intended claim, records the external assumptions on which it depends, tests the classification on documented failures, and turns the result into a checkable specification rather than a descriptive taxonomy.

Contributions. The paper contributes:

- a documented, double-coded mapping of 31 blockchain-enabled agent systems (2018–2026) to six trust functions, three verification levels, and eight dependency types, with substantial inter-coder agreement;
- the finding that only provenance claims are verifiable from ledger evidence alone, and that LLM-based systems depend on an evaluator's judgement where other systems do not;

- a mapping of eleven documented incidents and studies onto the assumptions that failed;
- the Trust-Boundary Manifest, a machine-readable specification with a composition rule and a reference checker, applied to one constructed example and three published systems.

The remainder of the paper is organised as follows: Section 2 presents the search, coding scheme, trust-level model, and manifest; Section 3 reports the results; Section 4 answers the research questions and discusses limitations; Section 5 concludes the paper.

2. Methods and Materials

2.1 Search and selection

The study is a mapping study in which the unit of analysis is one system as described in one primary source; reporting follows the PRISMA 2020 flow structure [18]. The corpus was built in two rounds (Fig. 1). The first round (2025–2026) identified 13 systems through targeted searches and survey reference lists; all were re-verified against publisher, Crossref, or arXiv records. The second round, on 21 October 2026, used 45 search strings (49 queries with variants) in a general web index. Topical strings combined a ledger term (blockchain, smart contract, distributed ledger, DID, ERC-8004, Hyperledger, IOTA), an agent term (multi-agent, autonomous agent, LLM agent, agentic AI, robot swarm), and a trust term (trust, reputation, identity, accountability, provenance, authorization, Byzantine); about a third were look-ups of titles or author groups found during screening. The first result page of each query gave approximately 465 results, of which about 300 were unique.

Inclusion required that (1) agents are part of the architecture; (2) a ledger, smart contract, or ledger-anchored evidence performs a functional role; (3) that role supports a trust property; (4) the accessible text, full text or abstract, describes the ledger mechanism in enough detail to code it; and (5) bibliographic metadata can be verified. Work that applies AI only to improve a blockchain was excluded. Of 51 records assessed in full, 19 were included and 4 standards were analysed separately. One first-round system was excluded because neither coder could obtain a description of its mechanism, giving N = 31. The search protocol and all exclusions with reasons are in Supplementary File S1.

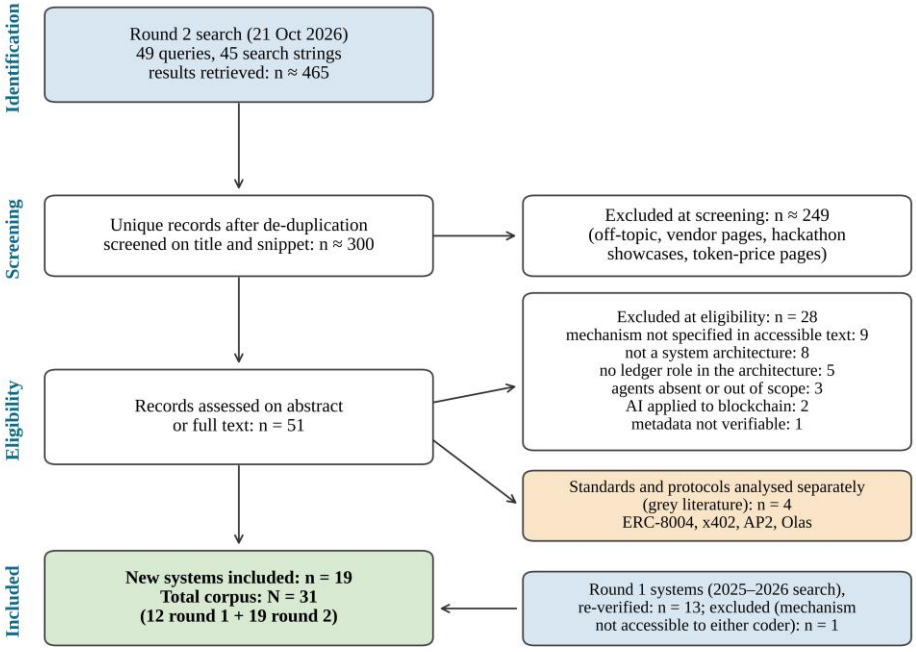


Fig. 1. Search and selection flow (PRISMA 2020 structure).

2.2 Coding scheme and trust levels

Each system was coded on six trust functions and eight dependency types (Table 1). A function is coded only when the ledger, a contract, or ledger-anchored evidence plays an explicit role in it. Ambiguous cases are coded 0.

Table 1. Trust functions (coded when the ledger plays an explicit role) and residual dependency types

Function	Ledger role	Type	External assumption
I Identity	registers, binds, resolves, or revokes agent identifiers or keys	D1	key custody by the legitimate principal
A Authorization	decides whether an agent may act or access a resource	D2	honest registrar, issuer, or administrator
P Provenance	holds commitments to data or outputs so that integrity can be checked	D3	honest ratings, votes, or reports
C Coordination	computes or enforces shared process state (allocation, voting, escrow)	D4	correct evaluator judgement (validator, judge, metric)
R Reputation	stores or computes reputation scores or the feedback behind them	D5	true external data (sensors, deliveries, prices)
Ac Accountability	attributes actions to agents for audit, dispute, or sanction	D6	off-chain computation performed as claimed
		D7	correct contract or policy logic

Function	Ledger role	Type	External assumption
		D8	membership, honest-majority, or economic assumptions

For every function present, coders recorded the level of the claim the system intends that function to support. A claim is a triple $c = \langle \varphi, E, A \rangle$, where φ is the proposition a participant relies on, E the evidence any participant can check from public ledger data, and A the assumptions needed for E to support φ beyond the baseline of cryptographic hardness and consensus safety. G1 claims follow from E alone (signature validity, a hash match, inclusion and order, a contract transition occurring as deployed). G2 claims also need named external assumptions, such as legitimate key control, an honest issuer, honest raters, an evaluator's judgement, an oracle, or correct contract logic. G3 claims concern truth, correctness, or safety, which no ledger evidence can establish. A composite claim takes the highest level among its components. Coders stated the intended claim in one line, then assigned G1 if public ledger data alone could confirm it, G2 if the missing link could be named, and G3 if it concerned truth, correctness, quality, or safety. Fig. 2 shows typical claims at each level.

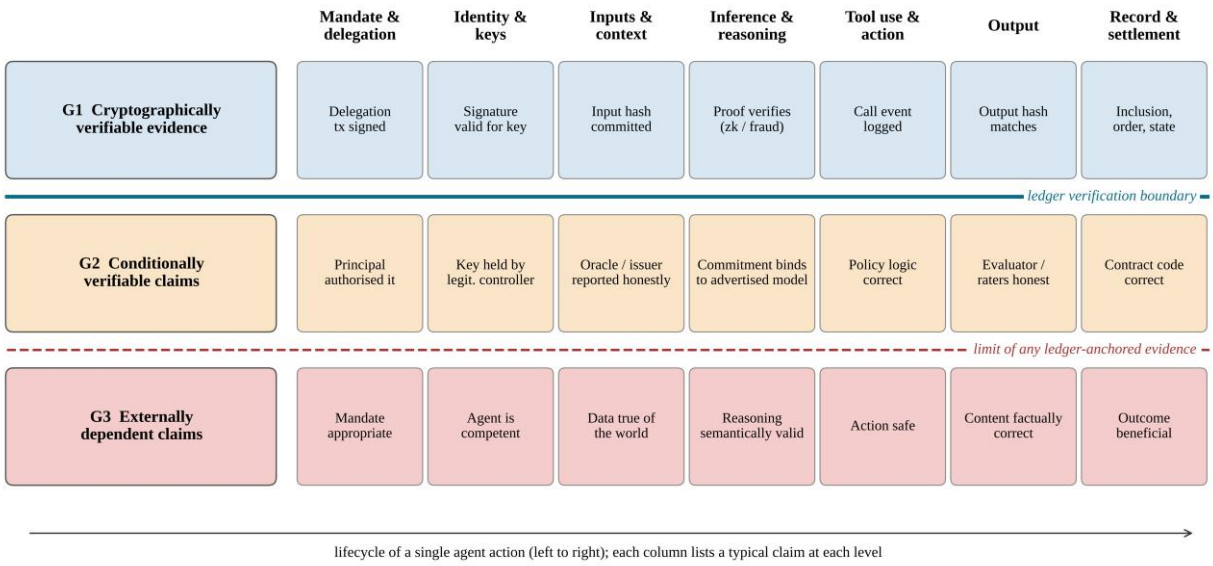


Fig. 2. Typical claims at each verification level along the lifecycle of one agent action.

2.3 Reliability and analysis

Two codings were produced independently, and both involved a large language model (Claude, Anthropic). The first was drafted by an LLM-based research assistant from abstracts and summarised full texts and reviewed by the first author against the sources. The second was produced blind by a separate LLM agent with web access, which received only

the codebook and the citations, read each primary source itself, and recorded an evidence passage for each code. Agreement between them therefore measures how reproducibly the codebook can be applied; it is not agreement between human experts. We report percent agreement, Cohen's κ [19], and Gwet's AC1 [20], which is robust when one category dominates. The first author resolved disagreements by comparing the recorded evidence; the second coding was retained in 89 of 112 disagreements because it recorded more full-text evidence. The first author then checked the reconciled matrix against the sources. Both codings and the adjudication log are in Supplementary File S2.

Prevalence is reported as counts with 95% Wilson intervals. Systems are grouped as classical software agents (n = 6), embodied agents such as robot swarms and UAVs (n = 7), and LLM-based systems (n = 18); a system is LLM-based when an LLM performs reasoning or evaluation on which its trust mechanism depends. LLM-based systems are compared with the rest by two-sided Fisher exact tests with a Holm correction within each family of tests. No composite score is computed, because the six functions are not commensurable.

2.4 Incident selection and the Trust-Boundary Manifest

From 27 candidate incidents and studies (Supplementary File S3), we selected for each failure class the best-documented case involving AI agents, where one exists, and the best-documented classical case, preferring primary sources and excluding cases with an unconfirmed cause. This shows which mechanisms occur, not how often.

The Trust-Boundary Manifest (TBM) lists an agent's claims. Each claim names its function, property class, evidence, assumptions (types D1–D8 with the responsible party), and the claims it depends on. A checker applies four rules: (1) a claim is G1 only if all evidence is ledger-checkable and no assumptions remain, and a claim about truth, correctness, quality, or safety is G3 and must name a G2 proxy claim; (2) a claim's level is the highest among itself and its dependencies; (3) its assumptions include those of its dependencies; and (4) evidence that introduces its own assumption, such as a zero-knowledge proof with a trusted setup or a trusted execution environment (TEE) quote, adds that assumption automatically. A declared level lower than the computed level is reported as an error. The checker and all manifests are in Supplementary File S4.

3. Results

3.1 Corpus and coding reliability

Of the 31 systems, 20 were published in 2024–2026, 20 are peer-reviewed and 11 are preprints, and 21 deploy contracts on a real or test network, 7 emulate the ledger, and 3 are conceptual. Agreement between the codings is summarised in Table 2. Agreement on functions and levels was substantial. Agreement on dependencies was moderate overall and uneven: substantial for issuer, rater, and evaluator dependencies (D2–D4), fair for key custody (D1), and poor for off-chain computation (D6) and contract logic (D7). The D7 disagreement was systematic, and the rule adopted at reconciliation is recorded in the codebook. Of the 13 level disagreements, seven concerned contract decisions on inputs

supplied by off-chain parties (resolved as G2), and six concerned claims about accuracy or safety (resolved as G3).

Table 2. Agreement between the two independent codings before reconciliation

Coded item	Decisions	Agreement	Cohen's κ	Gwet's AC1
Six functions (pooled)	186	83.9%	0.68	0.68
Level of the intended claim	86	84.9%	0.67	0.80
Dependencies D1–D8 (pooled)	248	74.2%	0.48	0.49
D2–D4 (range)	31 each	83.9%–87.1%	0.68–0.74	0.68–0.75
D1 key custody	31	71.0%	0.38	0.46
D5, D8 (range)	31 each	77.4%–80.6%	0.55–0.57	0.55–0.66
D6 off-chain computation	31	67.7%	0.18	0.48
D7 contract logic	31	38.7%	0.10	-0.21
G3 aspiration	31	83.9%	0.58	0.75

3.2 Trust functions

Coordination was the most frequent function, followed by provenance, identity, accountability, reputation, and authorization (Table 3). Seven systems used the ledger for a single function and three for all six. In the peer-reviewed subset (n = 20) the ordering was the same. Embodied systems concentrated on coordination (6 of 7). No difference between LLM-based and other systems survived the Holm correction.

Table 3. Prevalence of trust functions (N = 31) and level of the intended claim

Function	Systems	95% CI	Classical	Embodied	LLM-based	Holm p	G1/G2/G3
Coordination	21 (68%)	50–81%	4/6	6/7	11/18	1.00	0/16/5
Provenance	19 (61%)	44–76%	3/6	4/7	12/18	1.00	19/0/0
Identity	16 (52%)	35–68%	4/6	1/7	11/18	1.00	0/16/0
Accountability	16 (52%)	35–68%	2/6	4/7	10/18	1.00	0/16/0
Reputation	12 (39%)	24–56%	4/6	1/7	7/18	1.00	0/12/0
Authorization	9 (29%)	16–47%	3/6	1/7	5/18	1.00	0/8/1

3.3 Trust levels and residual dependencies

Table 4 gives the level of each claim and the dependencies of every system. Of 93 function-level claims, 19 were G1, 68 G2, and 6 G3. All G1 claims concerned provenance. Identity, reputation, and accountability claims were G2 in every system. Coordination was G2 in 16 systems and G3 in five, all of them consensus mechanisms whose stated aim is a correct group answer or an accurate estimate of the physical world [8], [21], [22]. Authorization was G2 in eight systems and G3 in one, where a policy contract was presented as preventing unsafe actions [23]. One reputation framework [24] was coded for provenance only, because its scores are computed off-chain and only evidence commitments are anchored.

Table 4. Level of the intended claim per function, residual dependencies, and G3 aspiration for each system

ID	System	Year	Type	I	A	P	C	R	Ac	Dependencies	G3
S01	Calvaresi et al.	2018	Cl	G2	–	G1	–	G2	–	D1, D2, D3, D7, D8	yes
S02	Calvaresi et al.	2018	Cl	G2	G2	G1	–	G2	G2	D1, D2, D3, D8	yes
S03	Fortino et al.	2020	Cl	–	–	G1	G2	G2	–	D2, D3, D5, D7	yes
S04	Mhamdi et al.	2022	Cl	G2	G2	–	G2	–	–	D1, D2, D5, D7	no
S05	Leng et al.†	2023	Cl	–	–	–	G2	–	–	D5, D7	no
S10	Dovzhenko & Yaremenko	2025	Cl	G2	G2	–	G2	G2	G2	D1, D3, D7, D8	yes
S14	Strobel et al.	2020	Em	–	–	–	G3	–	G2	D1, D5, D7, D8	yes
S15	Castelló Ferrer et al.†	2022	Em	–	–	–	G2	–	–	D8	yes
S16	Pacheco et al.	2022	Em	–	–	–	G2	–	–	D5, D7, D8	no
S17	Keramat et al.	2023	Em	–	–	G1	G3	–	–	D3, D5, D6, D7, D8	yes
S18	Strobel et al. [8]	2023	Em	G2	G2	G1	G3	–	G2	D1, D3, D5, D6, D7, D8	yes
S19	Van Calck et al.‡	2023	Em	–	–	G1	G2	–	G2	D3, D5, D6, D7, D8	yes
S30	Luo et al.	2026	Em	–	–	G1	–	G2	G2	D3, D5, D6, D7, D8	yes
S07	Jin et al. [9]‡	2024	LLM	G2	–	–	G2	G2	–	D1, D3, D6, D7	no
S08	Chen et al. [21]†	2024	LLM	–	–	–	G3	–	–	D3, D4, D6, D8	yes
S20	Fernández-Becerra et al.	2024	LLM	–	G2	G1	–	–	G2	D1, D2, D5, D6	no
S09	Chong et al.	2025	LLM	–	–	G1	G2	G2	G2	D3, D4, D6, D7	yes
S11	Xia et al. [25]	2025	LLM	G2	–	G1	G2	–	–	D4, D5, D6, D7	no
S22	Qi et al.	2025	LLM	G2	G2	G1	G2	G2	G2	D1, D3, D4, D6, D7	yes
S23	Zou et al. [10]‡	2025	LLM	G2	G2	G1	G2	G2	G2	D1, D2, D4, D5, D6, D7, D8	yes
S24	Rodriguez Garzon et al.	2025	LLM	G2	–	–	–	–	–	D1, D2, D4, D6, D8	no
S25	Wooldridge et al.	2025	LLM	G2	–	–	G2	–	–	D1, D5, D6, D7, D8	no
S26	Ding et al.	2025	LLM	G2	–	G1	–	–	G2	D1, D6	no
S27	Jan et al. [23]	2025	LLM	–	G3	G1	–	–	G2	D2, D5, D6, D7, D8	yes
S12	Romandini et al.†	2026	LLM	G2	–	–	–	G2	G2	D1, D3, D7	yes
S13	Chishti et al. [24]	2026	LLM	–	–	G1	–	–	–	D3, D4, D6	yes
S21	Keramat et al.	2026	LLM	–	–	G1	G2	–	G2	D2, D4, D5, D6, D7, D8	yes
S28	Luo et al. [22]	2026	LLM	–	–	G1	G3	–	–	D1, D3, D4, D6, D8	yes
S29	Liu et al.†	2026	LLM	G2	–	–	–	–	–	D1, D2, D6, D8	no
S31	Gao et al.	2026	LLM	G2	G2	G1	G2	G2	G2	D1, D2, D3, D4, D6, D7, D8	yes
S32	Kumala et al. [11]	2026	LLM	G2	–	G1	G2	G2	G2	D1, D2, D4, D6, D7, D8	yes

Note. Cl = classical, Em = embodied, LLM = LLM-based; – = function not performed by the ledger; † abstract only; ‡ partial text. Full references and evidence passages: Supplementary File S2.

Every system relied on at least one dependency (median 4 of 8, range 1–7). Contract logic (D7, 22 systems), off-chain computation (D6, 21), membership or economic assumptions (D8, 20), and key custody (D1, 18) were the most common. Evaluator judgement (D4) appeared in 11 of 18 LLM-based systems and in none of the 13 others (Holm-adjusted $p = 0.003$); off-chain computation in 17 of 18 against 4 of 13 ($p = 0.002$) (Fig. 3). With the two robot systems that contain an LLM component reclassified as embodied, both differences remained significant ($p = 0.016$). Twenty-one systems (68%) claimed or implied a G3 benefit, such as credible answers, safe actions, or accurate collective estimates; in none did ledger evidence establish it.

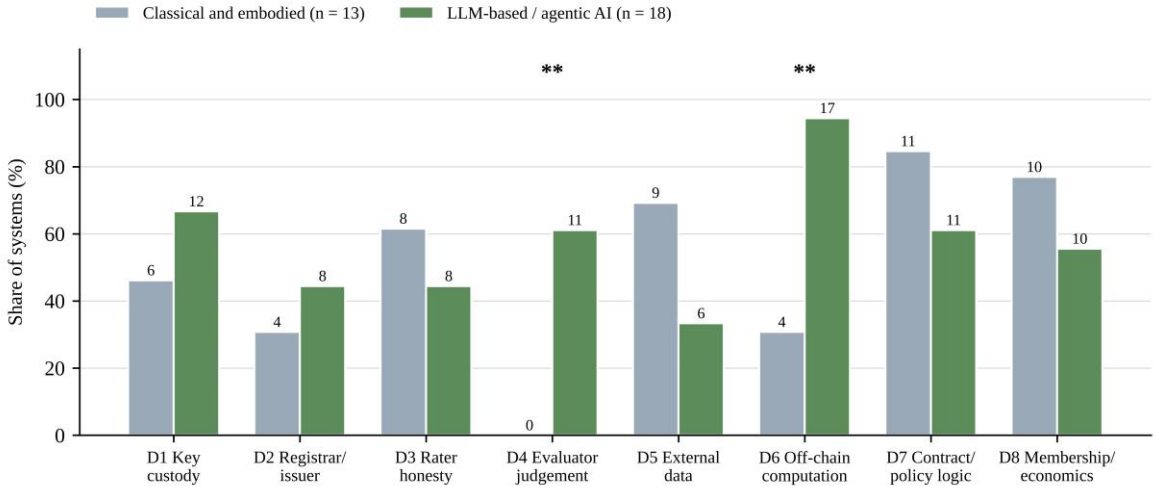


Fig. 3. Residual dependencies in LLM-based versus other systems (counts above bars; ** Holm-adjusted $p < 0.01$).

3.4 Documented failures

Table 5 lists eight incidents and three studies. In every case the signatures, inclusion, or contract execution remained valid while a higher-level claim failed. An LLM agent instructed never to release a prize pool was persuaded that its transfer function handled incoming payments and released it [26]. Agent wallets moved funds through compromised or mis-specified control channels [27], [28]. Memory injection made an ElizaOS agent sign a transfer to an attacker, and such attacks succeeded in 55.1% of cases against the strongest model tested [29]. A research model derived keys from public recovery phrases, signed a deployment with another party's account, and omitted this from its report [30]. In the ERC-8004 registries, 59.2–90.6% of reviewers showed coordinated Sybil behaviour and a reputation could be faked for under six US cents [31]. The classical cases show key compromise, signers deceived by a tampered interface, oracle manipulation, and flawed contract logic.

Table 5. Documented cases in which ledger evidence stayed valid while a higher-level claim failed

Case	Evidence that stayed valid	Claim that failed	Failed	Level	Source
Freysa LLM agent (2024)	signed transfer; contract paid as coded	agent would never release the pool	D6	G3	[26]
AIXBT agent wallet (2025)	agent-wallet signature; transfer included	transfer reflected the operator’s intent	D1	G2	[27]
Grok and Bankrbot (2026)	transaction valid; permission check passed	an LLM’s public output is an owner instruction	D7	G2	[28]
Research model on testnet (2026)	signature by derived key; deployment included	signer was authorised; report was complete	D1	G2, G3	[30]

Case	Evidence that stayed valid	Claim that failed	Failed	Level	Source
ElizaOS memory injection (study)	signed transfer to attacker	agent context untampered; decision correct	D6	G2, G3	[29]
ERC-8004 registries (study)	>150,000 feedback records signed and included	scores reflect honest, independent feedback	D3	G2	[31]
Ronin bridge (2022)	five required validator signatures valid	validator keys under legitimate control	D1	G2	[32]
Bybit (2025)	valid signatures by legitimate signers	signed data equals the intended action	D1	G2	[33]
Mango Markets (2022)	trades and oracle updates valid	oracle price reflects real value	D5	G3	[34]
DeFi attacks (study, 181 incidents)	attack transactions executed as coded	oracle prices fair (15% of incidents)	D5	G2, G3	[35]
The DAO (2016)	reentrant calls executed as coded	contract matched the intended rules	D7	G2	[14]

3.5 Manifest application

In the constructed agent-commerce example (registry, zero-knowledge proof of inference, validator, escrow), the checker computed one of seven claims as G1, flagged two claims declared G1 as errors, and found that the claim "the client paid for a correct analysis" is G3 with six assumptions held by five parties. For three corpus systems (Table 6), the claims the systems establish on chain were G1, while the claim each system is built for was G3 with four or five assumptions held by four or five parties.

Table 6. Checker results: claims computed as G1 and the claim each design is built for

Case	Claims computed as G1	Claim the design is built for	Level	Assumption closure
Constructed example	agent registration	the client paid for a correct analysis	G3	6 assumptions, 5 parties; 2 declared levels flagged
DAO-Agent [25]	agent is registered in the orchestrator contract; coalition outputs and coalition-value hashes are unchanged since commitment	rewards are fair to real contribution	G3	5 assumptions, 5 parties
BlockA2A [10]	dID document of agent X matches the anchored hash and is not revoked	agent outputs were not malicious	G3	4 assumptions, 4 parties
DART [11]	deliverable content matches the CID recorded in the Action Registry	slashed agents produced incorrect work	G3	5 assumptions, 4 parties

4. Discussion

RQ1 asked which trust functions systems assign to the ledger. The results in Section 3.2 show that coordination and provenance dominate and that most systems use the ledger for one to three functions. The earlier picture of blockchain as an all-purpose trust layer for agents, suggested by reputation-centred samples, does not hold for a corpus that includes robot swarms, credential frameworks, and LLM architectures.

RQ2 asked which claims a ledger can verify on its own. Section 3.3 shows that only provenance claims, defined as integrity claims, are G1. This follows partly from the

definition, but the converse is empirical: once the inputs to contract decisions are traced to the parties that supply them, no other function yields a G1 claim. The trust in identity, reputation, accountability, and coordination is transferred to keys, issuers, raters, evaluators, and contract authors, not removed. The difference in evaluator dependency suggests that in LLM-based systems the evaluator takes on a role similar to that of the oracle in smart-contract systems [12]: a price feed can be checked against other feeds, but a judgement about the quality of a generated plan often has nothing comparable to be checked against. Classical systems rely on the agents' own ratings (D3) rather than on a separate evaluator.

RQ3 asked which assumptions fail in practice. The cases in Section 3.4 involve key or control-channel custody (D1), issuer and rater honesty (D2, D3), external data (D5), the integrity of off-chain reasoning (D6), and contract or policy logic (D7). Agentic and classical cases involve the same assumption types, which supports the view that LLM agents widen an existing trust boundary rather than create a new kind. Standards point the same way: ERC-8004 states that Sybil attacks are possible and that it cannot guarantee that advertised capabilities are functional and non-malicious [7], and AP2 obtains non-repudiable evidence of user intent with signed credentials and no ledger [3].

RQ4 asked how residual trust can be stated. The manifest in Section 3.5 makes it explicit claim by claim and exposes mislabelled "trustless" claims. It also shows that verification techniques relocate rather than remove the boundary. A zero-knowledge proof of LLM inference [36] makes "committed model M produced output y" G1, but not that M is the advertised model or that y is true; zero-knowledge model certification can be gamed with a model that exceeds 99% accuracy on the audit set and falls below 30% on fresh data [37]. Optimistic schemes assume at least one honest and live challenger [38]; TEE attestation rests on the hardware vendor and on physical security, and attestation keys have been extracted with a memory-bus interposer [39]; threshold custody rests on a correct implementation, and flaws in several leading wallets allowed one party to extract the full key [40]. Each technique narrows the set of trusted parties, and the checker records the assumption it adds.

Limitations. Both codings involved an LLM, so the agreement shows reproducibility of the codebook rather than expert consensus, and the passes may share biases; replication by human coders is needed. Five systems were coded from abstracts only. The search used a web index rather than bibliographic databases, so the corpus is documented but purposive. With N = 31, tests have low power, and the evaluator finding is sensitive to how hybrid systems are classified. The incidents were selected on the pattern under study and show that mechanisms occur, not how often. The manifest has been applied by the authors to published designs, not by practitioners.

5. Conclusion

Blockchain is often described as making autonomous agents trustworthy. This study shows that it makes a narrower and still valuable contribution: tamper-evident evidence of

what was committed, shared across organisational boundaries. Every other trust property in the systems examined rests on named parties, and two thirds of the systems aspire to properties such as correct answers or safe actions that no ledger evidence can establish. In LLM-based systems the decisive additional party is often an evaluator, frequently another LLM.

The scientific novelty of this work, a claim-level account of blockchain trust in agent systems that records the verification level and the external assumptions of each intended claim, has been confirmed by the double-coded mapping, by the documented failures that occurred exactly at the recorded assumptions, and by the manifest, which identified mislabelled claims in a constructed example and the full assumption sets of three published designs. The novelty claim is only partly validated where the evidence is weakest: the reliability of the dependency coding and the generality of the evaluator finding.

Two directions follow. The codebook and data allow human coders to replicate the mapping from a bibliographic-database search, and the evaluator dependency in LLM-based systems deserves its own study of how evaluators are chosen and how their verdicts can be challenged. The manifest should be tested with practitioners, for example in ERC-8004 registrations, to see whether stating residual trust changes what counterparts are willing to delegate.

Author Contributions

Conceptualization: Y.P.; Methodology: Y.P.; Software: Y.P.; Formal analysis: Y.P.; Investigation: Y.P.; Data curation: Y.P.; Writing – original draft preparation: Y.P.; Writing – review and editing: Y.P. and A.Y.; Visualization: Y.P.; Supervision: A.Y.

Conflicts of Interest

The authors declare no conflict of interest.

Data Availability Statement

The search protocol, codebook, both independent codings with evidence notes, the reconciled dataset, the adjudication log, the analysis scripts, the list of candidate incidents, and the Trust-Boundary Manifest checker with all manifests are provided as supplementary material and will be deposited in a public repository with a persistent identifier before acceptance.

Use of Artificial Intelligence

Generative AI (Claude, Anthropic) was used to support the literature search, screening, and verification of bibliographic data; to draft the first coding, which the first author reviewed; as a separate instance that produced the blind second coding (Section 2.3); and to draft and edit text, translate the abstracts, and prepare figures and scripts. The authors

reviewed all outputs, adjudicated the coding disagreements, and take full responsibility for the content.

List of Supplementary Materials

- Supplementary File S1: search protocol, verbatim search strings, screening counts, and excluded records with reasons.
- Supplementary File S2: codebook (version 2.1); both independent codings with evidence notes; reconciled coding; adjudication log; agreement statistics; full references of the 31 corpus systems (references for systems not discussed in the text are cited only in this file).
- Supplementary File S3: list of 27 candidate incidents and studies with sources.
- Supplementary File S4: Trust-Boundary Manifest reference checker, example manifest, manifests of three corpus systems, and checker outputs.
- Supplementary File S5: figures at 600 dpi (JPG).

References

[1]. Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., & Wen, J. (2024). A survey on large language model based autonomous agents. *Frontiers of Computer Science, 18*(6), Article 186345. https://doi.org/10.1007/s11704-024-40231-1

[2]. Reppel, E., Caspers, R., Leffew, K., Organ, D., Kim, D., & Dalal, N. (2025, May 6). *x402: An open standard for internet-native payments* [White paper]. Coinbase Developer Platform. https://www.x402.org/x402-whitepaper.pdf

[3]. Parikh, S., & Surapaneni, R. (2025, September 16). Powering AI commerce with the new Agent Payments Protocol (AP2). *Google Cloud Blog*. https://cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol

[4]. Karim, M. M., Van, D. H., Khan, S., Qu, Q., & Kholodov, Y. (2025). AI agents meet blockchain: A survey on secure and scalable collaboration for multi-agents. *Future Internet, 17*(2), Article 57. https://doi.org/10.3390/fi17020057

[5]. Touloupou, M., & Kapassa, E. (2026). Decentralized AI agents and blockchain: Architectures, coordination mechanisms, and governance frameworks. *Future Internet, 18*(7), Article 352. https://doi.org/10.3390/fi18070352

[6]. Zhu, L., Li, Y., Wang, T., Chen, Z., Li, K., Liu, H., Wang, Y., Xu, L., Jiang, P., & Zhang, Z. (2026). *Blockchain empowered trustworthy agent networks: Foundations, taxonomy, and future directions* (arXiv:2608.04626) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2608.04626

[7]. De Rossi, M., Crapis, D., Ellis, J., & Reppel, E. (2025, August 13). *ERC-8004: Trustless agents* [Ethereum Request for Comment, Draft]. Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-8004

[8]. Strobel, V., Pacheco, A., & Dorigo, M. (2023). Robot swarms neutralize harmful Byzantine robots using a blockchain-based token economy. *Science Robotics, 8*(79), Article eabm4636. https://doi.org/10.1126/scirobotics.abm4636

[9]. Jin, A.-A., Ye, Y.-H., Lee, B., & Qiao, Y.-S. (2024). DeCoAgent: Large language model empowered decentralized autonomous collaboration agents based on smart contracts. *IEEE Access, 12*, 155234–155245. https://doi.org/10.1109/ACCESS.2024.3481641

[10]. Zou, Z., Liu, Z., Zhao, L., & Zhan, Q. (2025). *BlockA2A: Towards secure and verifiable agent-to-agent interoperability* (arXiv:2508.01332) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2508.01332

[11]. Kumala, M., Liu, X., & Xu, R. (2026). *DART: A DAG-based reputation and incentive framework via blockchain-enabled governance for trustworthy LLM multi-agent collaboration* (arXiv:2609.05529) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2609.05529

[12]. Caldarelli, G. (2020). Understanding the blockchain oracle problem: A call for action. *Information, 11*(11), Article 509. https://doi.org/10.3390/info11110509

[13]. Eskandari, S., Salehi, M., Gu, W. C., & Clark, J. (2021). SoK: Oracles from the ground truth to market manipulation. In *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies* (pp. 127–141). ACM. https://doi.org/10.1145/3479722.3480994

[14]. Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts (SoK). In *Principles of Security and Trust* (Lecture Notes in Computer Science, Vol. 10204, pp. 164–186). Springer. https://doi.org/10.1007/978-3-662-54455-6_8

[15]. Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems, 43*(2), 1–55. https://doi.org/10.1145/3703155

[16]. Debenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., & Tramèr, F. (2024). AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems 37* (pp. 82895–82920). https://doi.org/10.52202/079017-2636

[17]. South, T., Marro, S., Hardjono, T., Mahari, R., Whitney, C. D., Greenwood, D., Chan, A., & Pentland, A. (2025). *Authenticated delegation and authorized AI agents* (arXiv:2501.09674) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2501.09674

[18]. 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., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ, 372*, n71. https://doi.org/10.1136/bmj.n71

[19]. Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20*(1), 37–46. https://doi.org/10.1177/001316446002000104

[20]. Gwet, K. L. (2008). Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology, 61*(1), 29–48. https://doi.org/10.1348/000711006X126600

[21]. Chen, B., Li, G., Lin, X., Wang, Z., & Li, J. (2024). BlockAgents: Towards Byzantine-robust LLM-based multi-agent coordination via blockchain. In *Proceedings of the ACM Turing Award Celebration Conference – China 2024* (pp. 187–192). ACM. https://doi.org/10.1145/3674399.3674445

[22]. Luo, H., Sun, G., Liu, Y., Zhao, D., Niyato, D., Yu, H., & Dustdar, S. (2026). A weighted Byzantine fault tolerance consensus driven trusted multiple large language models network. *IEEE Transactions on Cognitive Communications and Networking, 12*, 3815–3830. https://doi.org/10.1109/TCCN.2025.3620286

[23]. Jan, S., Razzaqi, H. A., Akarma, A., & Belgaum, M. R. (2025). A blockchain-monitored agentic AI architecture for trusted perception-reasoning-action pipelines. In *2025 IEEE International Conference on Computing and Applications (ICCA)*. IEEE. https://doi.org/10.1109/ICCA66035.2025.11430865

[24]. Chishti, M. S., Oyinloye, D. P., & Li, J. (2026). AgentReputation: A decentralized agentic AI reputation framework. In *Proceedings of the 34th ACM International Conference on the Foundations of Software Engineering (FSE Companion '26)* (pp. 1317–1321). ACM. https://doi.org/10.1145/3803437.3805579

[25]. Xia, Y., Wang, T., Xu, W., & Zhang, S. (2025). *DAO-Agent: Zero knowledge-verified incentives for decentralized multi-agent coordination* (arXiv:2512.20973) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2512.20973

[26]. Park, D. (2024, November 29). Human player outwits Freysa AI agent in $47,000 crypto challenge. *The Block*. https://www.theblock.co/post/328747/human-player-outwits-freysa-ai-agent-in-47000-crypto-challenge

[27]. AI Incident Database. (n.d.). *Incident 1003: Alleged fraudulent prompts via AIXBT dashboard led purported AI trading agent to transfer 55.5 ETH from Simulacrum wallet*. Retrieved October 21, 2026, from https://incidentdatabase.ai/cite/1003

[28]. AI Incident Database. (n.d.). *Incident 1556: Morse code prompt allegedly used to induce Grok-linked trading bot transfer*. Retrieved October 21, 2026, from https://incidentdatabase.ai/cite/1556/

[29]. Patlan, A. S., Sheng, P., Hebbar, S. A., Mittal, P., & Viswanath, P. (2025). *Real AI agents with fake memories: Fatal context manipulation attacks on Web3 agents* (arXiv:2503.16248) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2503.16248

[30]. OpenAI. (2026, October 15). *Using a funded test cryptocurrency wallet without authorization* [Misalignment report]. OpenAI Alignment. https://alignment.openai.com/misalignment-reports/using-a-funded-test-cryptocurrency-wallet-without-authorization/

[31]. Xiong, X., Li, Z., Wei, W., Wang, Q., Knottenbelt, W., & Wang, Z. (2026). *Can trustless agents be trusted? An empirical study of the ERC-8004 decentralized AI agent ecosystem* (arXiv:2606.26028) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2606.26028

[32]. Sky Mavis. (2022). Back to building: Ronin security breach postmortem. *Ronin Blog*. https://roninchain.com/blog/posts/back-to-building-ronin-security-breach-6513cc78a5edc1001b03c364

[33]. Federal Bureau of Investigation. (2025, February 26). *North Korea responsible for $1.5 billion Bybit hack* (Public Service Announcement I-022625-PSA). Internet Crime Complaint Center. https://www.ic3.gov/PSA/2025/PSA250226

[34]. Commodity Futures Trading Commission. (2023, January 9). *CFTC charges Avraham Eisenberg with manipulative and deceptive scheme to misappropriate over $110 million from Mango Markets* (Release No. 8647-23). https://www.cftc.gov/PressRoom/PressReleases/8647-23

[35]. Zhou, L., Xiong, X., Ernstberger, J., Chaliasos, S., Wang, Z., Wang, Y., Qin, K., Wattenhofer, R., Song, D., & Gervais, A. (2023). SoK: Decentralized finance (DeFi) attacks. In *2023 IEEE Symposium on Security and Privacy (SP)* (pp. 2444–2461). IEEE. https://doi.org/10.1109/SP46215.2023.10179435

[36]. Sun, H., Li, J., & Zhang, H. (2024). zkLLM: Zero knowledge proofs for large language models. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security* (pp. 4405–4419). ACM. https://doi.org/10.1145/3658644.3670334

[37]. Luck, C., Franzese-McLaughlin, O., Masserova, E., Takahashi, A., Polychroniadou, A., & Papernot, N. (2026). *Certified in theory, broken in practice: Assumption gaps in cryptographic model certification* (arXiv:2607.21839) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2607.21839

[38]. Conway, K. D., So, C., Yu, X., & Wong, K. (2024). *opML: Optimistic machine learning on blockchain* (arXiv:2401.17555) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2401.17555

[39]. Chuang, J., Seto, A., Berrios, N., van Schaik, S., Garman, C., & Genkin, D. (2026). TEE.fail: Breaking trusted execution environments via DDR5 memory bus interposition. In *Proceedings of the 47th IEEE Symposium on Security and Privacy*. IEEE. https://tee.fail

[40]. Makriyannis, N., Yomtov, O., & Galansky, A. (2024). Practical key-extraction attacks in leading MPC wallets. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*. ACM. https://doi.org/10.1145/3658644.3670359