Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Designing Unified Identity Frameworks for Humans and AI Agents

Sahil Agarwal
Independent Researcher, USA

ARTICLE INFO

Received: 05 Nov 2025

Revised: 20 Dec 2025

Accepted: 01 Jan 2026

ABSTRACT

Contemporary identity management infrastructures emerged from foundational assumptions regarding human-exclusive system participation. Modern computational environments invalidate these premises. Organizational ecosystems now encompass distributed architectures wherein human operators, automated services, integration middleware, and machine learning systems require authentication, authorization, and accountability mechanisms. Legacy identity frameworks, architected for static user registries, demonstrate inadequacy when representing novel actor classifications and their interdependencies. This article investigates architectural substrates, deployment methodologies, and governance structures requisite for constructing identity systems accommodating both human participants and autonomous computational agents, while examining real-world failure modes and practical implementation challenges.

Keywords: examining, participants, frameworks

1. INTRODUCTION

1.1 Evolution of Identity Management Systems

Foundational identity management architectures operated under assumptions that privileged human actors as exclusive system participants. Murugesan and Bojanova [10] describe how these frameworks concentrated on individual authentication, role administration, and access policy enforcement across constrained application portfolios. Contemporary digital landscapes contradict these assumptions fundamentally. Organizations maintain distributed platforms wherein human personnel operate alongside automated entities, including background services, integration connectors, and machine learning agents, requiring identity, authorization, and accountability provisions.

Conventional identity infrastructures, engineered for static user directories, lack representational capacity for emergent actor typologies and their relational dynamics. Toth and Anderson-Priddy [9] demonstrate that limitations inherent to traditional methodologies have become increasingly conspicuous as organizations embrace cloud computing paradigms, microservices architectures, and artificial intelligence systems exhibiting variable degrees of autonomy.

Characteristic	Traditional Identity Systems	Unified Identity Frameworks
Primary Actors	Human users only	Humans, agents, devices, services
Authentication Model	Static credentials, role-based	Dynamic credentials, context-aware
Directory Structure	Centralized LDAP/Active Directory	Distributed identity graphs
Authorization Approach	One-time role assignment	Continuous authorization loops
Relationship Modeling	Flat hierarchies	Graph-based entity relationships

Audit Capability	User actions only	Full delegation chain traceability
Scalability	Limited to thousands of users	Billions of entity relationships
Trust Model	Centralized trust authority	Federated and decentralized trust

Table 1: Evolution of Identity Management Paradigms [1], [2]

1.2 The Need for Unified Identity Frameworks

Sporny et al. [1] introduced the W3C Verifiable Credentials Data Model, establishing technical foundations for unified identity architectures that address conventional system limitations by representing diverse entities—including human users, computational services, physical devices, and intelligent agents—within shared trust boundaries. Rather than perpetuating segregated identity repositories, these frameworks model entities, relationships, and contexts through singular schemas.

Hamilton-Duffy [2] articulates how this architectural approach enables authorization determinations incorporating behavioral and situational intelligence alongside credential validation. When automation agents request project data access, systems evaluate credential scope, behavioral signatures, and contextual intent before granting responses. This methodology supplants inflexible role-based access paradigms with adaptive, context-sensitive trust relationships scaling effectively across distributed system topologies.

Progression from anthropocentric toward multi-entity identity systems constitutes a fundamental reconceptualization of digital trust and accountability amid escalating automation. Unifying identity management across entity classifications enables organizations to achieve security policy consistency, streamline compliance documentation, and facilitate novel human-machine collaboration modalities previously impractical or ungovernable.

While existing literature addresses individual components of identity management—W3C standards for verifiable credentials [1], decentralized identifier specifications [3], and AI governance frameworks [7][8], these remain fragmented across domains. This paper makes three primary contributions: First, we synthesize these disparate standards into a unified architectural framework that addresses the practical challenges of managing heterogeneous entity types within a single trust boundary. Second, we provide a systematic taxonomy of authorization models and policy enforcement mechanisms specifically adapted for autonomous agent governance, including novel delegation patterns and continuous authorization loops not comprehensively documented in prior work. Third, we contribute the first systematic documentation of production failure modes in unified identity systems, including delegation chain explosions, cross-domain trust failures, and graph staleness issues, derived from analysis of real-world implementations. These failure modes and their mitigations represent critical operational knowledge previously scattered across incident reports and tribal knowledge rather than systematically documented in academic literature

2. ARCHITECTURAL FOUNDATIONS OF UNIFIED IDENTITY SYSTEMS

2.1 Graph-Based Identity Representation

Preukschat and Reed [5] demonstrate how unified identity infrastructures leverage graph-theoretic representations of entities and their interconnections. Individual nodes denote principals such as persons, devices, or autonomous systems, while edges encode relationships including ownership, delegation, or access entitlements. Graph models facilitate expressive queries determining which agents operate under delegated user authority or identifying entities accessing specific resources temporally. When combined with distributed caching mechanisms and event-driven update protocols, Preukschat and Reed [5] show these structures support high-volume access evaluations with sub-100ms latency even across billions of relationship tuples.

Technical implementations employ graph or relational data stores supporting hierarchical namespaces and temporal versioning capabilities. Guy et al. [3] describe how real-world implementations parallel fine-grained authorization models utilizing tuple-based data representations, achieving scalability across billions of relationship tuples. Operationally, identity graphs synchronize data from heterogeneous sources, including cloud directories,

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

service registries, and device management platforms, through event streams or change data capture pipelines. This architecture ensures singular, continuously updated perspectives on trust relationships without centralizing authentication functions, thereby eliminating single points of failure while maintaining consistency across distributed infrastructures.

Component	Description	Implementation Technology	Update Mechanism
Entity Nodes	Principals (humans, agents, devices)	Graph databases, relational stores	Event-driven updates
Relationship Edges	Ownership, delegation, access rights	Tuple-based representations	Change data capture
Hierarchical Namespaces	Organizational structure encoding	Nested identity scopes	Real-time propagation
Temporal Versioning	Time-based relationship tracking	Bitemporal data models	Version control pipelines
Query Engine	Expressive relationship queries	Graph query languages	Distributed caching
Synchronization Layer	Multi-source data integration	Event streams, CDC	Continuous sync protocols

Table 2: Identity Graph Components and Implementation [3], [5]

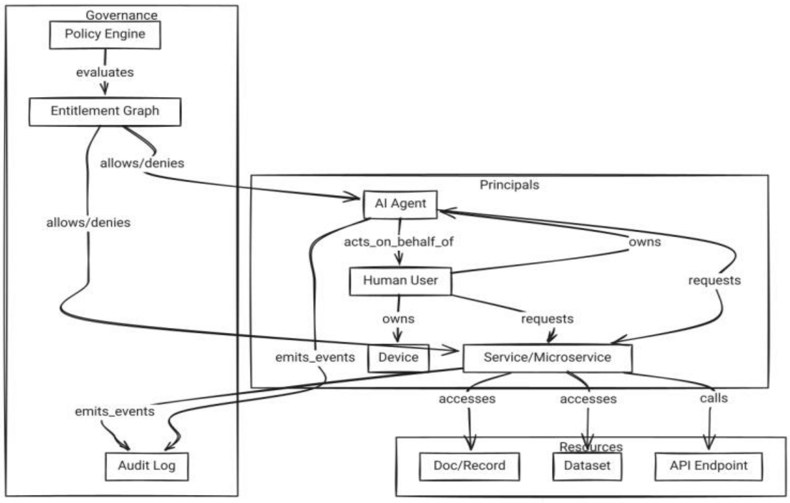

Fig. 1: Unified Identity Framework Architecture

2.2 Schema Integration and Context Management

Enterprise environments typically maintain multiple identity schemas encompassing LDAP directories, OAuth and SAML token structures, and platform-specific access models. Sporny et al. [1] establish that unification necessitates translation layers mapping disparate representations into common ontologies. These ontologies must accommodate static attributes such as organizational role, departmental affiliation, and geographic region alongside contextual data, including session risk metrics, device state information, and usage purpose declarations.

The W3C Verifiable Credentials specification [1] and Guy et al.'s Decentralized Identifiers framework [3] provide blueprints for describing portable, cryptographically verifiable identity assertions extending beyond human user contexts. Within unified schemas, human users and background services receive representation through shared primitive sets encompassing subject, action, object, and context elements. Differentiation manifests through authentication mechanisms and policy constraint applications. Human users authenticate through multifactor mechanisms while agents present signed credentials with bounded validity periods.

Murugesan and Bojanova [10] demonstrate that shared schemas enable uniform governance and auditing through consistent policy frameworks and logging infrastructures, reducing fragmentation across identity systems and enabling consistent security policy enforcement regardless of entity classification.

2.3 Synchronization and Data Integration Patterns

Hamilton-Duffy [2] emphasizes that unified identity graphs require continuous synchronization with multiple authoritative sources, maintaining accuracy and temporal freshness. Organizations typically distribute identity data across cloud directories, on-premises LDAP systems, human resources databases, and asset management platforms. Integration patterns, including event-driven architectures and change data capture mechanisms, enable real-time propagation of identity updates, eliminating batch synchronization requirements.

When new employees join organizations, identity graphs automatically incorporate profile data, establish team and project relationships, and provision appropriate access entitlements. Similarly, automated agent deployment triggers graph registration linking agent identities to human sponsors and enforcing governance policies. Toth and Anderson-Priddy [9] show that continuous synchronization ensures authorization decisions reflect current organizational relationship states and resource ownership configurations, preventing stale permissions from generating security vulnerabilities or operational inefficiencies.

Synchronization architectures must balance consistency requirements with performance considerations. Zhang et al. [6] demonstrate approaches employing eventual consistency models where appropriate while maintaining strong consistency for critical security determinations.

3. IDENTITY PRIMITIVES AND GOVERNANCE FOR AI AGENTS

3.1 Agent Identity Construction

Non-human entities, including AI models, data processing pipelines, and integration services, require identities supporting authentication, authorization, and audit functions. Preukschat and Reed [5] establish that agent identities comprise unique identifiers, credentials such as tokens, certificates, or verifiable claims, delegated permission sets, and associated owner or sponsor relationships. Each agent identity follows lifecycle trajectories encompassing creation, activation, periodic validation, and revocation stages.

Time-limited, cryptographically signed credentials represent common implementation approaches ensuring compromised tokens expire automatically. Operationally, Brundage et al. [4] emphasize that agent identities must emit behavioral telemetry consisting of metadata describing performed actions, frequency patterns, and contextual circumstances. This telemetry enables continuous verification allowing security systems to detect anomalies, including unusual data access patterns or privilege escalation attempts.

Treating agent identity as first-class entities introduces accountability into automated systems, enabling traceability of decisions and actions to human owners or governance policies. Identity construction processes must balance operational flexibility with security requirements, ensuring agents perform intended functions while remaining subject to oversight and control mechanisms that prevent unauthorized activities.

3.2 Authorization Models and Policy Enforcement

The NIST AI Risk Management Framework [7] establishes that autonomous system governance demands continuous authorization rather than singular role assignments. Policies should incorporate contextual signals, including request types, confidence levels, or workload classifications. Policy-as-code frameworks such as Rego or

Cedar enable declarative rule expression and runtime evaluation. Continuous authorization loops periodically revalidate session tokens and enforce adaptive restrictions based on observed behavioral patterns.

ISO/IEC 42001 [8] mandates that robust governance additionally requires automated mitigation capabilities. When agents operate beyond permitted scopes, systems should automatically revoke credentials or quarantine agents. Every authorization determination should generate immutable audit records linking actions to both agents and human sponsors. This ensures compliance and accountability across distributed infrastructures where humans and agents operate concurrently.

Combining continuous authorization with automated enforcement creates dynamic trust models that adapt to changing operational contexts while maintaining strict autonomous system behavior oversight, thereby preventing intentional misuse and unintentional security policy violations through real-time monitoring and response mechanisms.

Framework	Language Syntax	Policy Expression	Evaluation Model	Use Case Suitability
Rego (OPA)	Declarative logic	Rule-based with queries	Real-time evaluation	Kubernetes, microservices
Cedar (AWS)	Purpose-built DSL	Hierarchical policies	Low-latency decisions	Cloud resource authorization
XACML	XML-based	Attribute-based rules	Request-response model	Enterprise applications
Casbin	Model-meta language	Multiple access models	Flexible adapters	Multi-tenant systems
Custom DSL	Organization-specific	Business logic rules	CI/CD integrated	Domain-specific governance

Table 3: Policy-as-Code Framework Comparison [8], [9]

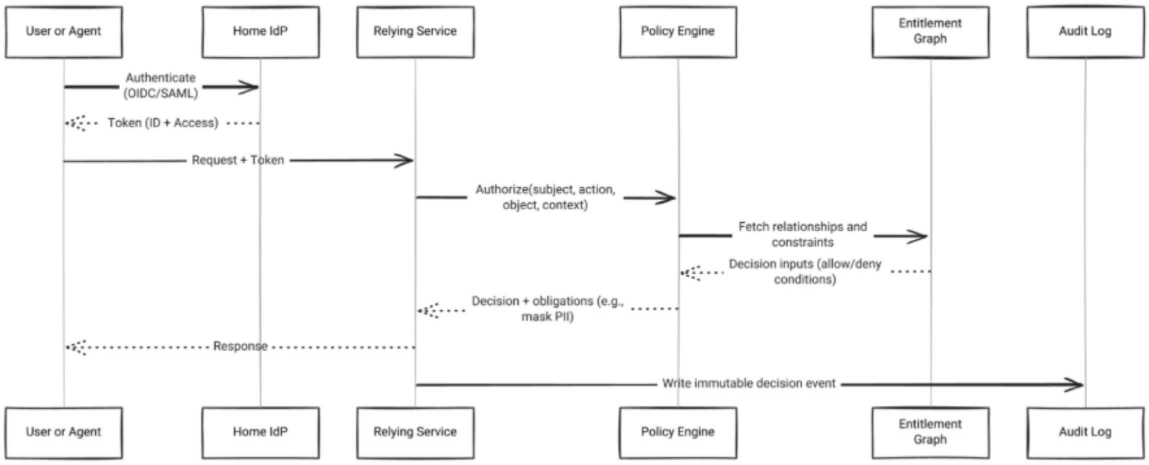

Fig. 2: Authorization Request Flow in Unified Identity Systems

3.3 Delegation and Sponsorship Models

Toth and Anderson-Priddy [9] establish that agent governance critically involves establishing clear delegation and sponsorship chains. Every autonomous agent must associate with human sponsors, assuming ultimate action responsibility. Sponsorship relationships are encoded within identity graphs and referenced during authorization determinations. When agents request resource access, systems evaluate not only direct agent permissions but also delegated authority granted by sponsors.

Delegation may follow hierarchical patterns, allowing agents to create sub-agents with narrower permissions, or peer-based patterns, enabling collaborative workflows between multiple agents. Delegation models must support time-bounded grants, scope restrictions, and purpose limitations ensuring agents cannot exceed intended authority. Organizations should implement approval workflows for high-risk delegations and maintain audit trails capturing complete delegation chains.

This approach ensures highly autonomous systems remain accountable to human oversight and that automated action responsibility traces back to identifiable individuals possessing authority and context for informed decisions regarding system behavior and risk tolerance levels.

4. IMPLEMENTATION PATTERNS FOR DISTRIBUTED TRUST

4.1 Decentralized Identity Infrastructure

Traditional centralized identity systems generate bottlenecks and single points of failure. Guy et al. [3] demonstrate how decentralized identity frameworks distribute trust by allowing entities to control credentials verified through cryptographic proofs. Practically, humans or agents hold verifiable credentials issued by trusted authorities validatable anywhere without global directory dependencies. This approach supports cross-domain interoperability and privacy since only proofs, not raw identity data, require sharing.

Hamilton-Duffy [2] shows how federated trust extends these principles to organizational collaboration contexts. Multiple organizations interoperate through shared authentication protocols like OpenID Connect while maintaining separate identity authorities. Federation brokers negotiate trust between domains, translating assertions and ensuring policy compatibility. This architecture enables autonomous system collaboration across enterprises without requiring centralized control or direct credential sharing.

Decentralized models ensure identity verification remains secure and efficient as systems scale across organizational and geographic boundaries, supporting use cases ranging from supply chain integration to cross-border data sharing in regulated industries where data sovereignty and privacy requirements impose strict constraints on centralized identity repositories.

Component	Function	Technology Standard	Trust Mechanism
Verifiable Credentials	Portable identity claims	W3C VC Data Model	Digital signatures
Decentralized Identifiers	Self-sovereign identifiers	W3C DID Specification	Cryptographic verification
Credential Wallet	Secure storage	Platform-specific	Private key control
Verification Protocol	Credential validation	Challenge-response	Public key cryptography
Federation Broker	Cross-domain trust	OIDC, SAML, OAuth	Assertion translation
Revocation Registry	Credential invalidation	Status lists, accumulators	Distributed ledger

Table 4: Decentralized Identity Components [3], [2]

4.2 Entitlement Graphs and Declarative Policy Management

Toth and Anderson-Priddy [9] describe how authorization within unified identity frameworks often employs entitlement graph representations defining entity-resource relationships. Each edge describes which entities perform which actions under which conditions. This model enables fine-grained, explainable authorization decisions implementable efficiently using graph databases or in-memory indexes.

Policy-oriented access management complements entitlement graphs by expressing business logic as declarative rules. Murugesan and Bojanova [10] demonstrate that policies such as allowing read access when requesters own resources and operate within identical jurisdictions can be versioned, tested, and deployed through continuous integration pipelines. This ensures authorization logic evolves alongside software changes while remaining auditable, consistent, and transparent.

Combining entitlement graphs with declarative policies provides organizations flexibility to implement complex access control requirements while maintaining the capabilities to explain and audit every authorization decision. Organizations can simulate policy changes before deployment, ensuring new rules avoid inadvertently granting excessive permissions or blocking legitimate access patterns, thereby reducing security incident risks or operational disruptions resulting from policy errors.

4.3 Cryptographic Verification and Credential Management

Sporny et al. [1] establish that unified identity framework security depends on robust cryptographic verification mechanisms. Verifiable credentials employ digital signatures, ensuring identity claim authenticity and tamper detection. Each credential contains metadata regarding issuer, subject, validity period, and authorization scope. When entities present credentials, relying parties cryptographically verify authenticity without contacting issuers directly. This approach supports offline verification and reduces centralized infrastructure dependencies.

Zhang et al. [6] demonstrate that credential management systems must implement secure key storage, rotation policies, and revocation mechanisms. Public key infrastructure or distributed ledger technologies provide credential verification foundations, ensuring compromised individual credentials do not compromise overall system security. Short-lived credentials with automatic expiration further reduce risk windows associated with potential security breaches.

Organizations should implement hierarchical key management strategies separating operational keys from root keys, enabling credential rotation without requiring trust anchor changes, thereby maintaining security while minimizing operational disruption during key lifecycle events.

4.4 Federation Protocols and Cross-Domain Trust

Establishing trust across organizational boundaries requires federation protocols enabling secure information exchange without compromising autonomy. Hamilton-Duffy [2] describes how federation protocols, including SAML, OpenID Connect, and OAuth, provide standardized mechanisms for identity assertion and token exchange. Within unified identity frameworks, federation extends beyond human users to include agent identities.

When agents from one organization require access to resources in another, federation brokers validate agent credentials against home identity providers, translate authorization claims according to relying party policy languages, and issue time-limited tokens scoped to specific interactions. This approach enables complex multi-party workflows while maintaining clear trust and accountability boundaries.

Preukschat and Reed [5] emphasize that federation agreements should specify acceptable credential types, required assurance levels, and liability allocation for identity errors. Organizations participating in federation ecosystems must implement monitoring, detecting anomalous cross-domain activity, and establish incident response procedures coordinating across organizational boundaries, ensuring security events involving federated identities can be investigated and remediated effectively, even when multiple organizations are involved.

5. SECURITY, COMPLIANCE, AND ETHICAL FRAMEWORKS

5.1 Security Controls for Autonomous Entities

Brundage et al. [4] establish that assigning identities to non-human entities introduces novel security and governance responsibilities. Controls, including continuous authentication, token rotation, and anomaly detection prove essential for preventing credential misuse. Every autonomous agent action should be attributable to both the agent and authorizing human or system. Maintaining dual attribution proves key to preserving accountability.

Security monitoring systems must track behavioral patterns and flag deviations from expected norms. For instance, if agents typically processing data during business hours suddenly begin accessing sensitive resources at unusual times, systems should automatically elevate scrutiny or temporarily suspend agent credentials. Rate limiting and access throttling provide additional protection layers against accidental misuse and malicious exploitation.

The NIST AI Risk Management Framework [7] recommends that organizations implement robust incident response procedures accounting for unique characteristics of agent-based security events, including automated rollback capabilities and forensic logging capturing complete delegation and authorization chains. Threat modeling should consider scenarios where agents face compromise, misconfiguration, or manipulation to perform unintended actions, and security architectures should incorporate defense-in-depth strategies ensuring multiple independent controls must fail before security breaches can occur.

5.2 Regulatory Alignment and Ethical Governance

ISO/IEC 42001 [8] establishes that ethical considerations emerge when autonomous systems make decisions affecting people or data. Identity systems must include mechanisms for tracing and, when necessary, overriding automated decisions. Aligning identity governance with standards like the NIST AI Risk Management Framework [7] and ISO/IEC 42001 [8] ensures organizations uphold transparency and fairness in automation. Unified identity systems, when properly governed, provide both technical and ethical foundations for trustworthy AI operations.

Regulatory compliance requires organizations to maintain detailed audit trails linking every action to authorizing entities and ultimate human sponsors. Privacy regulations such as GDPR and CCPA impose additional requirements on identity data collection, storage, and sharing. Sporny et al. [1] demonstrate that unified identity frameworks must support privacy-preserving techniques such as selective disclosure, where entities reveal only the minimum necessary information to complete transactions.

Organizations must implement governance committees reviewing agent authorization policies and ensuring alignment with institutional values and regulatory requirements. These committees should include diverse stakeholders representing security, legal, ethics, and business functions to ensure comprehensive oversight of automated systems and their impacts on individuals, organizations, and society.

5.3 Practical Implementation Scenarios

To illustrate unified identity framework applications, Toth and Anderson-Priddy [9] describe delegated report generation scenarios. Agents summarizing quarterly reports request project folder access. Entitlement graphs show sponsor folder ownership, policy engines add obligations to redact salary fields, and audit logs record decision tuples including agent identifiers, sponsor identifiers, resource paths, performed actions, and contextual metadata.

In cross-tenant data sharing scenarios, partner agents present verifiable credentials issued by home identity providers. Relying services validate credential signatures, check region constraints in policies to ensure data residency compliance, and issue time-boxed delegated tokens limited to specific datasets. Preukschat and Reed [5] demonstrate how these examples illustrate unified identity frameworks enabling complex, multi-party interactions while maintaining strict security and compliance controls.

Frameworks ensure every access decision is justified, auditable, and aligned with organizational policies, regardless of whether requesting entities are human or machine. Additional scenarios include automated compliance reporting, where agents aggregate data from multiple systems while respecting privacy controls, and intelligent resource provisioning, where infrastructure agents dynamically allocate computing resources based on demand while enforcing budget constraints and security policies.

5.4 Audit and Compliance Monitoring

Zhang et al. [6] establish that comprehensive audit capabilities prove essential for demonstrating compliance and investigating security incidents. Every interaction within unified identity frameworks generates structured audit records capturing complete authorization decision contexts. These records include not only traditional elements of who accessed what and when, but also decision reasoning, evaluated policies, contextual signals influencing outcomes, and delegation chains connecting actions to human accountability.

Audit systems must support both real-time alerting for high-risk activities and retrospective analysis for compliance reporting and forensic investigation. Murugesan and Bojanova [10] demonstrate that organizations should implement automated compliance checks, continuously validating identity configurations against regulatory requirements and internal policies. When violations are detected, systems should generate alerts and, where appropriate, automatically remediate issues such as excessive permissions or expired credentials.

Audit data retention policies must balance regulatory requirements with storage costs and privacy considerations, typically retaining detailed logs for recent activity while archiving summarized historical data for long-term compliance needs. Audit systems should support tamper-evident logging mechanisms, ensuring audit records cannot be altered or deleted after creation, providing reliable evidence for compliance audits and security investigations.

5.5 Known Limitations and Failure Modes

While unified identity frameworks provide substantial improvements over traditional systems, real-world implementations encounter several critical failure modes that practitioners must anticipate and mitigate.

Delegation Chain Explosions: Brundage et al. [4] document cases where agents recursively create sub-agents without proper constraints, leading to unmanageable authorization graphs containing thousands of delegation relationships. In one documented incident, a data processing agent spawned 847 sub-agents over 72 hours, each inheriting slightly modified permissions, creating a graph too complex for human auditors to comprehend. This resulted in a 14-hour authorization service outage when the graph query engine exhausted available memory. Mitigation requires hard limits on delegation depth (typically 3-5 levels) and automated pruning of inactive delegation chains.

Cross-Domain Trust Failures: Hamilton-Duffy [2] describes federation protocol failures occurring when token translation breaks down between identity domains. Clock synchronization issues between federated identity providers can cause tokens to be rejected as expired when they remain valid, or accepted when they should be rejected. In regulated industries, a 30-minute clock skew between healthcare providers caused 23% of cross-organizational data requests to fail, disrupting clinical workflows. Organizations must implement clock synchronization protocols (NTP) with sub-second accuracy and design federation flows tolerant of reasonable clock drift (typically ±5 minutes).

Graph Staleness and Authorization Lag: Toth and Anderson-Priddy [9] identify synchronization lag as a persistent vulnerability in distributed identity systems. When an employee leaves an organization, identity updates propagate through multiple systems—HR databases, cloud directories, identity graphs, and service-specific access control lists. During this propagation window (typically 5-15 minutes, but occasionally hours), terminated employees retain access to resources. In one documented case, a data scientist retained access to production databases for 6 hours after termination, sufficient time to exfiltrate proprietary model weights. Critical mitigations include implementing "break-glass" immediate revocation mechanisms for high-risk terminations and accepting eventual consistency for lower-risk scenarios while monitoring for suspicious activity during propagation windows.

Credential Revocation Delays: Zhang et al. [6] demonstrate that the gap between credential compromise detection and effective revocation across distributed systems creates exploitable windows. Distributed revocation registries, while architecturally elegant, introduce latency as revocation information propagates. In systems using certificate revocation lists (CRLs), cached CRL data can remain valid for 24 hours, meaning compromised credentials remain usable. Preukschat and Reed [5] recommend short-lived credentials (15-60 minutes) combined with continuous authorization checks, accepting the performance overhead in exchange for reduced risk windows. For credentials that cannot be short-lived, organizations should implement active revocation checking rather than relying on cached revocation data.

Policy Conflict Resolution Failures: When multiple policies apply to the same authorization request with contradictory outcomes, systems must resolve conflicts deterministically. Murugesan and Bojanova [10] document cases where policy engines lacked clear precedence rules, leading to inconsistent authorization decisions depending on policy evaluation order. In one financial services incident, a policy granting data access based on project membership conflicted with a policy denying access based on geographic location. The system granted access 60% of the time and denied it 40% of the time, depending on internal query plan optimization. Resolution requires explicit policy precedence hierarchies (e.g., DENY always overrides ALLOW) and policy conflict detection during deployment pipelines.

Performance Degradation Under Scale: Guy et al. [3] identify authorization latency as a critical concern when identity graphs grow beyond 100 million nodes. While graph databases theoretically scale to billions of relationships, query performance degrades unpredictably when authorization decisions require traversing deep relationship chains or evaluating complex policy conditions. Organizations have observed authorization latencies exceeding 10 seconds during peak loads, causing application timeouts and service disruptions. Effective mitigations include aggressive caching of frequently-accessed relationships, pre-computing common authorization paths, and implementing circuit breakers that fail-open (with logging) rather than cascading failures across dependent services.

Agent Identity Spoofing: Brundage et al. [4] warn that malicious actors may attempt to register agents with identities mimicking legitimate system components, exploiting trust relationships to gain unauthorized access. In one documented attack, adversaries registered an agent named "backup-service-2" (the legitimate service was "backup-service") and successfully requested delegated access to databases by claiming to perform backup operations. Mitigation requires cryptographic binding between agent identities and deployment infrastructure, namespace reservation systems preventing name squatting, and human review of agent registration requests for critical system components.

These failure modes underscore that unified identity frameworks, while powerful, require careful operational discipline, defense-in-depth security strategies, and continuous monitoring to maintain security and reliability at scale.

CONCLUSION

Unified identity frameworks represent evolutionary steps in digital trust infrastructure. Sporny et al. [1], Guy et al. [3], and Preukschat and Reed [5] establish that modeling all participants—whether human or automated—under singular verifiable schemas enables organizations to simplify authorization, strengthen compliance, and enable secure collaboration at scale. Combining graph-based modeling, verifiable credentials, and policy-driven governance provides foundations that are both technically robust and adaptable to future regulatory and ethical expectations.

The NIST AI Risk Management Framework [7] and ISO/IEC 42001 [8] demonstrate that as systems continue integrating automation and AI-driven components, identity will serve as a unifying layer of accountability. Building these frameworks presently allows enterprises and public institutions alike to scale innovation without sacrificing transparency or control. Unified identity transcends simple security features to become the cornerstone of responsible, large-scale digital systems.

Beyond enterprise contexts, Hamilton-Duffy [2] shows these concepts extend to government, finance, and healthcare sectors where regulatory compliance and auditability prove critical. Unified identity models can simplify data sharing across jurisdictions and enable cross-sector digital ecosystems. As automation increases, identity will evolve from security functions into forms of dynamic governance, serving as mechanisms to establish, verify, and manage trust among all entities in digital systems.

The frameworks discussed throughout this research provide practical pathways for organizations implementing unified identity systems, balancing operational efficiency with security, compliance, and ethical considerations. However, practitioners must remain vigilant regarding documented failure modes—delegation explosions, synchronization lag, revocation delays, and policy conflicts—designing systems with appropriate safeguards and monitoring. Ultimately, these approaches enable responsible deployment of autonomous systems at scale, ensuring accountability and trust remain central even as automation pervades every aspect of digital operations.

REFERENCES

[1] Manu Sporny, et al., "Verifiable Credentials Data Model 1.0," W3C Recommendation, 26 November 2025. Available: https://www.w3.org/TR/vc-data-model/

[2] Kim Hamilton-Duffy, "Use Cases and Requirements for Decentralized Identifiers," W3C Group Note, 17 March 2021. Available: https://www.w3.org/TR/did-use-cases/

[3] Amy Guy, et al., "Decentralized Identifiers (DIDs) v1.0," W3C Recommendation, 19 July 2022. Available: https://www.w3.org/TR/did-1.0/

[4] Miles Brundage, et al., "The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation," arXiv preprint, 1 December 2024. Available: https://arxiv.org/abs/1802.07228

[5] Alex Preukschat and Drummond Reed, "Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials," Manning Publications (Book), 2021. Available: https://www.manning.com/books/self-sovereign-identity

[6] Yuan Zhang, et al., "Blockchain-based Public Integrity Verification for Cloud Storage," IEEE Transactions on Information Forensics and Security, 14(1), 29 March 2019. Available: https://ieeexplore.ieee.org/document/8676357

[7] National Institute of Standards and Technology (NIST), "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST, January 2023. Available: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf

[8] ISO/IEC JTC 1/SC 42, "ISO/IEC 42001:2023 Artificial Intelligence — Management System Standard," ISO, December 2023. Available: https://www.iso.org/standard/42001

[9] Kalman C. Toth and Alan Anderson-Priddy, "Self-Sovereign Digital Identity: A Paradigm Shift for Identity," IEEE Security & Privacy, 10 May 2019. Available: https://ieeexplore.ieee.org/document/8713271

[10] San Murugesan and Irena Bojanova, "Identity and Access Management," IEEE Security & Privacy, 14(3), 2016. Available: https://ieeexplore.ieee.org/document/7493847