# Identity Governance and Security Automation: A Technical Framework for Enterprise Access Management

**Suneel Kumar Rawat**
Independent Researcher, USA

**Abstract**

Enterprise landscapes need identity and access management systems that can go beyond perimeter-based authentication models. Identity governance and security automation are the converging disciplines of enforcing least privilege, continuous verification, and policy-based runtime decisioning across distributed environments, enabling such requirements to be met. IAM products in modern usage separate runtime-based authentication from identity lifecycle provisioning. This separation exists due to the challenge of balancing governance requirements against the need for real-time access enforcement in a world of on-premises, cloud and hybrid deployments. Customary role-based access control systems also face 'role explosion' issues where the number of role definitions required can exceed the number of employees. Attribute-Based Access Control (ABAC) uses subject attributes, object attributes, environment attributes and policy rules at access decision time to enable dynamic removal or granting of access decisions that are not predefined assignments of static roles. Identity Lifecycle Management (ILM) automates Joiner-Mover-Leaver workflows to sync to authoritative Human Resource systems for access entitlements. Separation of Duties prevents conflicting entitlements from being assigned to identities, whilst Just-in-Time Access refers to the provision of time-limited elevation of privileges upon request and policy validation, adhering to Zero Trust principles through assumed breach and continuous verification of all access requests for resources. Policy-as-Code is a model-driven specification of IAM policies in a domain-specific language, enabling IAM policies to be controlled in source version control and specified declaratively in a CI/CD pipeline. Identity Fabric is a federated authentication and authorization architecture between domains using open standards protocols.

**Keywords:** Identity Governance, Access Control Models, Zero Trust Architecture, Privileged Access Management, Policy Automation

## 1. Introduction

As Identity and Access Management evolved from perimeter-oriented static authentication, access control at each entry point was based on manually assigned, coarse-grained roles through governance frameworks providing continuous authorization decisions across the distributed environment [1][2]. The results were excessive privileges and excess management of entitlements as organizations grew in size and complexity. Modern IAM systems separate two operational domains: Access Management provides runtime authentication and session management over the user's use of resources using multi-factor authentication and contextual policy evaluation, while Identity Governance provisions changes and runs periodic certifications and attestations over heterogeneous identity repositories [1].

Today organizations are required to implement governance policy mandates by implementing on-demand access control on cloud, on-premises, and hybrid IT infrastructures. Role-based access control implementations today face the challenge of role explosion, in which the number of roles required to support variations of permissions outstrips an organization's population. According to a survey of 199 ABAC research papers, role explosion is a known problem with role-based access control where an organization has more roles than employees to provide certain entitlement combinations. This administrative burden is further increased by the presence of legacy entitlements as employees move through different roles in an organization. This leads to privilege creep, as manually deprovisioning does not remove all access from legacy entitlements.

The challenge of non-human identity management is further complicated by the existence of service accounts, API tokens, and other machine identities that exist outside human-resource-based identity lifecycles. Customary identity platforms, designed around human employee provisioning workflows, cannot easily manage machine identities that lack standardized provisioning across multiple applications and cloud services. Although authentication and authorization can be performed in a federated or centralized way when suitably configured, customary identity-based approaches become tedious when applied to all user-to-resource relationships across organizations [2].

Regulations themselves often have controls to be enforced by a policy engine. For example, the Sarbanes-Oxley Act requires publicly traded corporations to document and certify internal controls over financial reporting, of which separation of duties is one of the most basic. To ensure effective compliance with SOX, conflicting entitlements must not be assigned to a single identity. For example, a user must not be allowed to be both a requester and an approver for financial transactions. Automated conflict detection must compare entitlement assignments against policy rules during both provisioning and periodic review activities [1].

Modern access control systems translate compliance requirements into machine-enforceable runtime policies on heterogeneous technology stacks. Policy engines evaluate access requests against dynamic sets of policy statements based on subject characteristics (e.g., attributes), resource characteristics, environmental conditions, and temporal conditions (e.g., time of day or week). The use of rules in place of manual management of access lists means organizations can scale identity governance to enforce least privilege and continuous verification across the digital perimeters formed by cloud-enabled environments, edge computing resources, and data sharing agreements between organizations.

## 2. Related Work and Methodology

Modern identity governance models build further on access control models while improving their scalability and flexibility. Role-Based Access Control (RBAC) tries to reduce administrative overhead in security provisioning by assigning permissions to roles instead of subjects, but it suffers from the role explosion problem, when organizations end up with more roles than employees because of the need to meet the variability in permissions. According to Enterprise implementations, role mining processes with minimum support thresholds yield role sets that cover more users with lower administration costs than direct allocative assignment of users to permissions. Another approach to access control is Attribute-Based Access Control (ABAC), which considers the attributes of subjects, objects and environmental conditions, together with a set of rules that govern the access control decision process. Hybrid role-based and attribute-based provisioning and governance are common enterprise technical implementations. Role-based provisioning of birthright access provides fundamental organizational entitlements, while attribute-based governance of fine-grained entitlements requires complex, multi-dimensional contextualization. A benefit of the Zero Trust Service Function Chaining architectures is that dynamic privilege management has been shown to measurably improve performance. Dynamic management of privileges reduces service access latency through selective application of security functions, unlike customary architectures wherein all traffic must traverse a service function chain. Risk-Adaptive Access Control (RAAC) models are based on multiple risk assessment functions. The different functions consider multiple attribute dimensions, such as user and device credentials, device trust level, and connection security conditions, as well as the user's previous access behavior. Global and local situational risk factors comprise the system-wide security status and user-related access situation, respectively.

## 3. Access Control Model Evolution

Role-Based Access Control (RBAC) defines a mechanism for inheritance of permissions whereby roles are granted to those identities that hold resource-related permissions in organizational hierarchies [3][4]. RBAC is intended to minimize security administration of an organization by assigning permissions to roles instead of individual users and then assigning users to roles based on their job functions. The relationships organizations using RBAC need to maintain between users, roles and permissions are at most $|Ur| + |Pr|$ for a role and less than $|Ur| \cdot |Pr|$ for most role cardinalities greater than 2 (the number of relationships required to correlate users directly to permissions, without roles as intermediaries) [3]. The scalability of RBAC suffers from role explosion phenomena [3][4]: large organizations create more roles than employees to provide the specific variations in permissions required by user groups.

Empirical studies of enterprise access control implementations found a strong effect of administrative cost on the adoption of RBAC. In a domain with 4743 users and 2907 permissions, 42 roles over 88% of users with the 23 most populous permissions reduced administration cost by 59% compared to assigning permissions directly to users. The roles available were generated through role mining algorithms with a minimum support threshold of 4.2% [3]. The cost function $f{=}\alpha\ |UA|{+}\beta\ |PA|{+}\gamma\ |ROLES|{+}\delta\Sigma c(r)$ captures the contribution of user-role assignment, permission-role assignment, the total number of roles, and other cost components towards the desired minimization of role and permission assignment, leading to the optimal role-set while completely assigning roles [3]. In organizations keeping separate roles for distinct permission combinations, the number of roles may be prohibitive in administrative effort and may weaken RBAC's promise of access control simplification as roles approach or exceed user counts [3][4].

The Attribute-Based Access Control (ABAC) model is an alternative that considers subject and object attributes, environmental conditions and policy rules at the access control decision time, allowing a dynamic authorization process without a predefined static assignment of roles [4]. ABAC-based methods use Boolean matrix decomposition, which decomposes the user-permission assignment matrix UPA (m, n) into a user-role matrix UA (m, k) and a role-permission matrix PA (k, n) through Boolean multiplication, minimizing k and minimizing the error bound [4]. The Role Mining Problem was defined and proved NP-complete through reduction from the Set Basis Problem, establishing a theoretical complexity that excludes the use of polynomial time algorithms for NP-complete problems unless P=NP [4]. Experimentation with 100 to 2000 users and 100 to 2000 permissions indicates that approximate solutions achieving 95% coverage with $\delta$= 5% create about 80% as many roles as exact match solutions, considerably reducing administration costs while maintaining accuracy [4].

The programmability of ABAC predicate expressions provides a solution. It allows access to resources based on subject properties, resource properties, time and environment to be evaluated at runtime, thereby not needing to enumerate all possible subject-object pairs. As a result, hybrid systems combining RBAC with ABAC have become the dominant access control model in enterprise systems, combining the organizational-birthright access control of RBAC with the fine-grain access control of ABAC based on multiple contextual factors. On synthetic datasets, the algorithm scales with increasing population size as $O(n^2)$ for candidate generation using pairwise user intersection patterns. Worst-case complexity can be $O(n^3)$ if iterative refinement requires taking the entire candidate set as input for n roles. Hybrid model organizations must juggle the ease of administration provided by role-based provisioning based on typical access patterns and the enforcement of complex authorization policies, e.g., where resources are protected by attribute-driven authorization based on more than just static role memberships but also temporal, geographic and behavioral attributes [3][4].

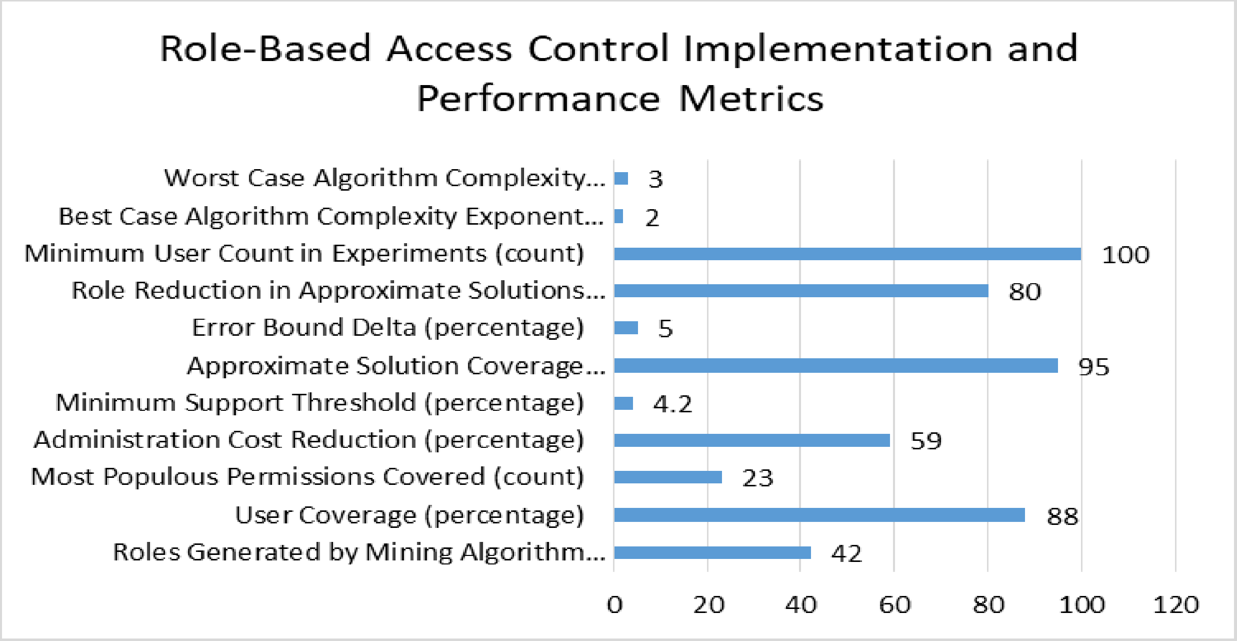

**Figure 1:** Role-Based Access Control Implementation and Performance Metrics [3, 4]

### 4. Governance and lifecycle management

Identity Governance implements business policies of the organization by provisioning, certifying users' access, and analyzing access rights across the enterprise. Modern identity management architectures distinguish centralized governance from decentralized governance with differing deployment characteristics. Centralized identity management systems under optimal enterprise conditions may achieve response times under 200 ms for up to 10,000 queries per second, but single points of failure still exist with reported downtime rates of nearly 1% per year [5]. In fact, the 2021 Microsoft Azure AD outage was a catastrophic failure where several hours of downtime made millions of dependent applications unable to authenticate users [5].

Identity Lifecycle Management orchestrates Joiner-Mover-Leaver workflows. Their purpose is to provision day one access based on authoritative HR systems and de-provision upon termination [5]. Centralized identity and access management systems such as LDAP can provide high availability through multi-master replication, where the backup servers take over as primary servers within seconds of a server fault [5]. Mover processes that remove entitlement upon role transfer can leave behind permissions. This occurs in the absence of manual deprovisioning, when historical access is not revoked, leading to privilege creep. Decentralized identity systems using Byzantine Fault Tolerance protocols are more resilient to failure than centralized systems, withstanding up to 33% node failure. Implementations of Hyperledger Indy using Redundant Byzantine Fault Tolerance can obtain throughput of 300 transactions per second (tps) with full agreement of peers and distributed validators [5].

Separation of Duties (SoD) policies do not allow the same identity to hold mutually exclusive entitlements [6]. Commonly found in financial transaction workflows, SoD controls block users from approving their own payments. Entitlement assignments are checked by automation through the use of conflict matrices and during provisioning events and certification campaigns. The ABACα model can be used to express classical access control models (discretionary access control, mandatory access control, role-based access control, etc.) in a unified attribute-based way [6]. The policies are authorization policies represented by boolean functions over the attributes of subjects and the attributes of objects. They are evaluated as true or false based on the attribute values and set membership operations [6]. The constraint functions that assign subject attributes or update object attributes can enforce security policies on create or update operations and prevent unauthorized privilege escalation by testing the value of the attribute against the security policy [6]. Entitlement governance involves managers attesting to their subordinates' entitlements periodically and identity governance platforms collecting summaries of anomalous, unused or sensitive entitlements. Least Privilege only requires sufficient entitlement for a specific task. However, excessive cloud IAM policy wildcards, accounts with legacy entitlements, and service accounts without just-in-time elevation violate the principle. Attribute-based access control models support atomic-valued or set-valued attribute functions that are evaluated on partially ordered or unordered ranges of attribute values [6]. Security engineers use formal specification languages to declaratively specify authorization policies that are composable through conjunction, disjunction, negation, and quantification over the domain of attribute values. This enables complex organizational policies to be specified while maintaining formal verification [6].

| Component | Function | Key Characteristics |
|---|---|---|
| Identity Lifecycle Management | Orchestrates Joiner-Mover-Leaver workflows synchronizing access rights with HR systems | Automates day-one provisioning and termination deprovisioning; addresses privilege creep through mover processes |
| Separation of Duties Controls | Prevents mutually exclusive entitlement combinations within single identities | Enforces conflict matrices during provisioning; blocks users from approving own transactions |
| Least Privilege Enforcement | Restricts identities to minimum permissions required for specific tasks | Challenges include excessive IAM wildcards, dormant accounts, over-permissioned service accounts |
| Certification Processes | Requires periodic manager attestation of subordinate access rights | Governance platforms flag anomalous, unused, or high-risk entitlements for investigation |
| Attribute-Based Policy Framework | Evaluates authorization through boolean functions over subject and object attributes | Supports atomic and set-valued attributes with formal specification languages enabling complex policy composition |

**Table 1:** Identity Governance and Lifecycle Management Framework [5, 6]

## 5. Security Automation and Privileged Access Controls

Just-in-Time Access removes standing privileges by granting time-limited elevation only when requested after completion of an approval workflow and after policies have been checked. Recent developments in Zero Trust architecture have also shown that dynamic permission management is superior to customary static permission management. Zero Trust Service Function Chaining results in 30% performance improvement (with respect to baseline Zero Trust architectures that require a static security function chaining for each traffic flow) in terms of the service access latency [7]. For instance, with high load (140 clients), the architecture achieves 973 accesses per second and 184 ms average service access latency, in comparison with the baseline architectures [7]. JIT mechanisms can automatically revoke elevated privileges when a session expires while capturing justification metadata for audit purposes. This aligns with the Zero Trust model, which assumes that the network is already compromised and continuously verifies the identity, rather than implicitly trusting a user based on their location and permanent credentials.

Some evaluations in enterprise distributed environments have shown that with selective application of security functions, the CPU load of monitoring servers in a Zero Trust environment is reduced by about 25% for some workloads [7]. For example, with 140 concurrent access requests, the load on logging servers is reduced while maintaining full security monitoring. On infrastructure servers with Zero Trust configurations, the observed context switch rate was 50% lower than on conventional architectures where all traffic traversed static security control points. The measurement result was attributed in part to risk-adaptive traffic routing, in which high-trust access requests were forwarded directly to resources without a resource-intensive security inspection, while low-trust traffic was routed through dynamically assembled service function chains.

Privileged Access Management technology provides credential vaulting, session proxying, risk-adaptive access control, and automated credential rotation to secure administrative accounts. Risk-Adaptive Access Control technology uses composite risk evaluation functions that obtain their inputs from more than a single dimension (such as user credentials, device trust level, connection security properties, and historical access patterns). [8] Access decisions in the attribute-based model also consider operational need, quantify security risk within a specific context, and assess situational awareness of the environment. [8] PAM implementations are responsible for discovering privileged accounts within varied environments, controlling access via approval workflows, and recording administrative sessions for forensic analysis. These identity governance solutions can enable PAM with business-relevant contexts and policy-driven privilege management.

ZTA coordinates, merges, and analyzes identity verification, device posture, and behavior analytics into runtime access decisions. Each policy decision point considers subject properties, resource properties, environmental conditions, and time-based conditions against dynamic rule sets [8]. It distinguishes global situational factors of the system's security posture from local situational factors of the user or other access context [8]. Next, enforcement points grant and deny decisions based on authorization decisions. Authorization takes into account risk scores, or trustworthiness, of users, devices, connections, and purposes of operation. The access history repositories should also provide information about authorization decisions and attributes. This should support heuristics to improve access control decisions by learning from patterns of compromise and correlations of security incidents [8].

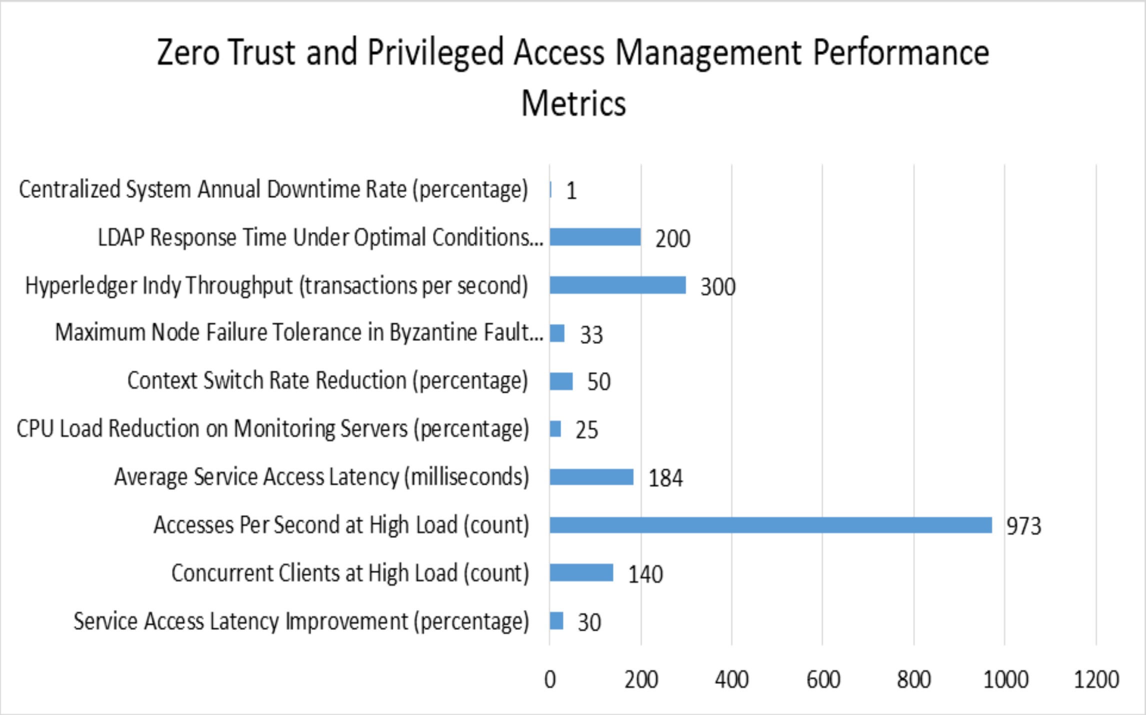

**Figure 2:** Zero Trust and Privileged Access Management Performance Metrics [7, 8]

### 6. Automation Primitives and Integration Architecture

Policy-as-Code frameworks define IAM policies using a domain-specific language. This enables them to be defined declaratively, version controlled, automatically tested and programmatically deployed. Other access control designs may address computational limitations such as those imposed by certain Big Data environments, where access control over hundreds of millions of records would be resource prohibitive. Generally speaking, the fine-grained access control mechanisms for distributed Big Data systems vary in terms of performance overhead as a function of policy granularities and architectural design. GuardMR, a fine-grained access control framework for Hadoop MapReduce, performs with a medium to high efficiency metric as it performs a filter-based access control prior to calling the Map function by checking the key-value pairs [9]. Query rewriting techniques applied to document-oriented datastores are efficient only if they enforce document-level policies, as opposed to field-level policies. Field-by-field authorization evaluation incurs a considerably higher computational overhead since they must be enforced whenever the view is generated [9].

However, platform-level authorization mechanisms can vary across integration methods, with the K-VAC wide-column datastore permission model achieving high efficiency scores when integrated directly with Cassandra source code, as compared to external libraries with strict binding compatibility standards across various platforms [9]. Authorization filter mechanisms control access to individual key-value pairs. The Vigiles framework for Hadoop deployments uses platform-specific APIs and control flow primitives without source code changes to intercept and enforce access control checks on data access requests [9]. The Mem framework for MongoDB provides a very efficient implementation of purpose-based access control through proxy-based message interception and query rewriting, done at document granularity. The ConfinedMem extension of Mem provides field-level policies but at medium to low efficiency because the authorization checking process is more complex. [9]. Performance assessments across different types of Big Data platforms show that the overhead increases with the complexity of policy, data granularity and architectural integration.

Identity Fabric architectures federate authentication and authorization services across organizational boundaries using standardized protocols. To express such capabilities formally, Hierarchical Group and Attribute-Based Access Control (HGABAC) models express policies in a directed acyclic graph structure that supports attribute inheritance across hierarchies of users and objects [10]. For hierarchical group approaches, a total of 3U + O + 9 assignments are required for Case 1 in a multi-course, multi-department system. For non-hierarchical attribute-based approaches, the total number of assignments required is 4U + 2O. In these equations 'U' is the number of users and 'O' is the number of objects [10]. In more complicated cases, the relative advantage improves, with case 2 requiring 4U + O + 11 relative to 5U + 2O as in the standard performance measures [10].

SAML provides browser-based single sign-on. OpenID Connect provides for newer token-based authentication flows. SCIM provides automated provisioning of identities across disparate, but connected, identity stores. RESTful SCIM defines HTTP calls to create, read, update or delete user accounts based on standard schema attributes. Identity providers use SCIM to synchronize users with target applications. Formal attribute-based models support strongly typed attribute structures, where each attribute can be of varied types, such as integers, floating-point numbers, strings or set-valued collections. This ensures that the implementation of a policy does not create type errors or ambiguities during the authorization process [10]. In policy languages that use the boolean rule-based evaluation model with ternary logic, the possible outcomes of an authorization request are TRUE, FALSE and UNDEFINED, the latter being the case of access denial [10].

Event-driven automation leverages SIEM alerts, webhooks and SOAR platforms to implement governance actions such as emergency access revocation and anomaly-based re-certification. User and Entity Behavior Analytics (UEBA) uses machine learning to look at both real and expected user access patterns. It then feeds this information into automated risk scoring and adaptive authentication workflows. In hierarchical attribute models, the consolidation mapping functions for homonymous attributes combine value sets of attributes with the same name from different inheritance paths, allowing for efficient computation of effective attribute sets [10]. Authorization decision functions apply policies to sets of attributes: user, object, environment (system state), connection (session-level attributes), and administration. The latter can include attributes set by security officers. The evaluation of these attributes allows for context-aware access control while allowing dynamic adaptation to the operating context. [10]

| Component | Implementation Approach | Integration Characteristics |
|---|---|---|
| Policy-as-Code Frameworks | Declarative domain-specific languages with version control and automated testing | Enables programmatic deployment across heterogeneous environments with formal verification capabilities |

| | | |
|---|---|---|
| Fine-Grained Access Control for Big Data | Filter-based authorization and query rewriting techniques | Performance overhead varies by policy granularity, data protection level, and architectural integration depth |
| Hierarchical Attribute-Based Models | Directed acyclic graph structures supporting attribute inheritance | Reduces administrative assignment requirements through group membership propagation across user and object hierarchies |
| Federated Identity Protocols | SAML for browser SSO, OpenID Connect for token flows and SCIM for automated provisioning | Standardized RESTful operations synchronize identity data across disconnected stores using schema attributes |
| Event-Driven Automation | SIEM alerts, webhooks, and SOAR platforms trigger governance actions | UEBA applies machine learning to baseline access patterns, feeding automated risk scoring and adaptive authentication |

**Table 2:** Policy Automation and Identity Federation Framework [9, 10].

## Conclusion

Identity governance, security automation and orchestration together enable scalable least privileged access and continuous verification for distributed enterprises. The maturity of static role-based access control into hybrid role-based attribute access control solved the problem of role explosion by employing advanced authorization solutions deriving subject attributes, object attributes, environmental conditions and time constraints for runtime request evaluation. Automated lifecycle management controlling Joiner-Mover-Leaver workflows provisions day-one joiner access and deprovisions leaver access through authoritative sources. However, they often fall short of revoking residual privileges due to manual interventions, leading to privilege creep. Just-in-Time elevation and privileged access controls reduce attack surfaces and lessen operational overhead by eliminating standing privileges and replacing them with time-bound elevation upon explicit provisioning request and approval workflow completion and entitlement policy validation. Access is revoked at the end of the session, and justification metadata is recorded in the audit trail. Policy-as-Code engines and federated identity fabrics provide a technical foundation for scalable governance. IAM policy is represented declaratively in a domain-specific language, allowing version control, unit/integration testing, and programmatic deployment to a heterogeneous service environment. In Zero Trust architectures, identity verification, device posture and behavior signals are composed into runtime access control decisions. Policy decision points evaluate access requests against a real-time risk profile that covers the strength of the authentication, geolocation- and travel-based anomalies from the user or entity, and any baselines or patterns of behavior established over time. These access control models now handle proofs and verification in distributed settings with data sources of large cardinality. Fine-grained access control implementations have had varying performance impacts based on both the granularity of policy and depth of integration. Hierarchical attribute-based access control architectures provide formal models for complex policies. Attribute inheritance by directed acyclic graphs in user and object hierarchies provides quantifiable performance benefits over other methods. Other areas to tackle are multi-tenancy for non-hierarchical architectures, non-human identity governance, cross-cloud entitlement normalization and machine learning-based policy synthesis as enterprise identity perimeters extend to edge computing and the Internet of Things.

## References

[1] DANIEL SERVOS and SYLVIA L. OSBORN, "Current Research and Open Problems in Attribute-Based Access Control," ACM Computing Surveys, Vol. 49, No. 4, Article 65, 2017. DOI: http://dx.doi.org/10.1145/3007204 [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3007204

[2] Vincent C. Hu et al., "Guide to attribute-based access control (ABAC) definition and considerations," NIST Special Publication, 2014. DOI: https://doi.org/10.6028/NIST.SP.800-162 [Online]. Available: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-162.pdf

[3] Alessandro Colantonio et al., "A Cost-Driven Approach to Role Engineering," Proceedings of the 2008 ACM symposium on Applied computing (March 2008), hps://doi.org/10.1145/1363686.1364198. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/1363686.1364198

[4] JAIDEEP VAIDYA et al., "The Role Mining Problem: A Formal Perspective," ACM Transactions on Information and System Security (TISSEC), Volume 13, Issue 3 (July 2010), https://doi.org/10.1145/1805974.1805983 [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/1805974.1805983

[5] Aviral Goel and Yogachandran Rahulamathavan, "A Comparative Survey of Centralized and Decentralized Identity Management Systems: Analyzing Scalability, Security, and Feasibility," DOI: https://doi.org/10.3390/fi17010001 MDPI, 2025. [Online]. Available: https://www.mdpi.com/1999-5903/17/1/1

[6] Xin Jin et al., "A unified attribute-based access control model covering DAC, MAC and RBAC," 26th Conference on Data and Applications Security and Privacy (DBSec), Jul 2012, ff10.1007/978-3-642-31540-4_4ff. [Online]. Available: https://inria.hal.science/hal-01534757/file/978-3-642-31540-4_4_Chapter.pdf

[7] LEONARD BRADATSCH et al., "ZTSFC: A Service Function Chaining-Enabled Zero Trust Architecture," IEEE Access, 2023. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10310190

[8] Savith Kandala et al., "An Attribute Based Framework for Risk-Adaptive Access Control Models," [online]. Available: https://www.profsandhu.com/confrnc/misconf/ARES11-RAdAC-final.pdf

[9] Pietro Colombo and Elena Ferrari, "Access Control in the Era of Big Data: State of the Art and Research Directions," Proceedings of the 23rd ACM on Symposium on Access Control Models and Technologies (June 2018) https://doi.org/10.1145/3205977.3205998 [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3205977.3205998

[10] Daniel Servos and Sylvia L. Osborn, "HGABAC: Towards a Formal Model of Hierarchical Attribute-Based Access Control." [Online]. Available: https://www.researchgate.net/profile/Daniel-Servos/publication/295210894