Understanding Authentication and Authorization: A Comparative Analysis of Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), and Relationship-Based Access Control (ReBAC) Authorization Models

Madhuri Margam
Director, Software Engineering, Capital One, Virginia, USA

***Abstract*—This study elucidates the distinctions between authentication and authorization within information security, two fundamental yet frequently conflated concepts. While authentication serves to confirm an entity's identity, authorization determines the permissible actions that entity may execute. A thorough understanding of these mechanisms is critical for architecting secure, scalable systems and reducing vulnerabilities. The study further explores three widely adopted authorization paradigms using a gymnasium analogy: Role-Based Access Control (RBAC), which assigns privileges based on predefined roles; Attribute-Based Access Control (ABAC), which leverages a dynamic evaluation of user and contextual attributes; and Relationship-Based Access Control (ReBAC), which determines access based on defined relationships among entities. The concluding discussion emphasizes that optimal security is realized when authentication and authorization function cohesively.**

***Keywords*—Authentication; authorization; access control; Role-Based Access Control (RBAC); Attribute-Based Access Control (ABAC); Relationship-Based Access Control (ReBAC); information security; least privilege; Zero Trust**

I. Introduction

In modern API security and system design, authentication and authorization are foundational, yet distinct processes often misconstrued as interchangeable. Authentication verifies who an entity is, while authorization governs what that entity can do once authenticated. Misinterpretation and improper implementation of these principles can lead to critical security flaws, enabling privilege escalation or unauthorized access. Organizations implement strong authentication but assume it provides authorization, or have minimal investments in authorization. Common implementation mistakes within authorizations include poor role design, over-privileged access, inadequate access reviews, and policy management.

To illustrate, consider the analogy of a gymnasium. Authentication occurs when a member scans their access card to validate identity and membership status. Authorization follows, defining which facilities, such as the pool or private training areas, the member may access based on their membership type. Both mechanisms are indispensable for a secure and user-oriented access management framework.

II. Authentication and Authorization: Conceptual Overview

A. Authentication

Authentication is the process of verifying the identity of a user or a service attempting to access a system. It answers the question, "Who are you?" Common authentication methods include password-based login, multi-factor authentication (MFA), and cryptographic verification.

In the gymnasium analogy, scanning a valid membership card authenticates the member, verifying their legitimacy. Only authenticated members proceed to the authorization stage.

B. Authorization

Authorization defines the access level and actions an authenticated user may perform. It answers, "What are you allowed to do?" For instance, in the gym, premium members may access group classes, while basic members are restricted to general equipment. The Principle of Least Privilege is supported by this distinction. The Principle of Least Privilege (PoLP) is a fundamental information security concept that requires users, processes, and systems to be granted only the minimum level of access necessary to perform their authorized functions. The goal is to prevent unauthorized access to data and services, coupled with making the access control enforcement as granular as possible [1].

Authentication and authorization together create a secure, layered defense mechanism that validates identity before granting specific privileges, as shown in Fig. 1.

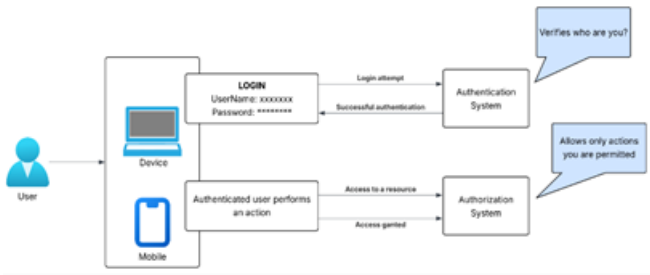

Fig. 1. Authentication and authorization mechanisms.

III. Common Authorization Models

Different domains implement authorization based on operational complexity and compliance needs. The three most prevalent models are RBAC, ABAC, and ReBAC.

A. Role-Based Access Control (RBAC)

RBAC associates permissions with predefined roles, simplifying access management for organizations with structured hierarchies. Access is granted based on the user's assigned role rather than individual identity.

In a gymnasium example, membership levels define roles:

- Basic members: access to cardio and weight areas
- Premium members: additional access to group classes
- VIP members: full access, including pool and lounge facilities

Advantages of RBAC:

- Enhanced security through role-based restriction
- Simplified policy management
- Regulatory compliance support
- Scalable and transparent administration

RBAC's simplicity and predictability make it ideal for environments with stable, well-defined roles.

B. Attribute-Based Access Control (ABAC)

ABAC extends RBAC by incorporating dynamic attributes, such as user characteristics, environmental context, and resource metadata, into policy decisions. Instead of static role assignments, ABAC evaluates combinations of attributes to determine access in real-time.

In a gymnasium example:

- Discount eligibility may depend on attributes like age < 25 or membership duration > 1 year.
- Facility access may be restricted to staffed hours, introducing contextual rules.

Advantages of ABAC:

- Fine-grained and adaptable access control.
- Context-aware and scalable.
- Dynamic response to environmental or user changes.

ABAC is a cornerstone for modern, policy-driven security architectures that demand adaptability and compliance. ABAC serves as the foundation for contemporary, policy-driven security architectures requiring both compliance and adaptability [2].

C. Relationship-Based Access Control (ReBAC)

ReBAC builds upon ABAC by evaluating relationships between entities rather than static roles or attributes. It is particularly effective in systems where access rights depend on social, organizational, or contextual links.

In a gymnasium example:

- Trainer-client relationship: Trainers access workout data only for assigned clients.
- Parental access: Parents manage reservations for children on family plans.
- Guest access: Guests enter only when sponsored and accompanied by a primary member.

Advantages of ReBAC:

- Granular and intuitive policy expression.
- Dynamic adaptation as relationships evolve.
- Reduced risk of privilege overextension.
- Scalable management in graph-like systems [3].

Systems such as Google Zanzibar exemplify ReBAC in practice, offering a globally consistent, low-latency authorization framework based on relational mappings between users and resources [4].

IV. Evaluation of Authorization Models

In the evolving landscape of digital security, access control mechanisms must balance security rigor, operational scalability, and user experience. While RBAC, ABAC, and ReBAC are often presented as distinct models, their practical implementation in large-scale systems tends to be complementary rather than mutually exclusive. Each model exhibits unique strengths, limitations, and contextual suitability.

A. Limitations in the Current Models

The Role-Based Access Control (RBAC) faces critical scalability and flexibility challenges in modern enterprise environments. Organizations frequently experience role explosion, creating hundreds of granular roles that become administratively complex and error-prone to manage. RBAC's static nature cannot adapt to dynamic business requirements such as time-sensitive access, location-based restrictions, or contextual factors without extensive manual reconfiguration. Complex role hierarchies often result in unintended privilege accumulation through inheritance, violating the principle of least privilege, while the model struggles to accommodate temporary access requests or exceptional permissions outside predefined role boundaries.

Attribute-Based Access Control (ABAC) and Relationship-Based Access Control (ReBAC) address some RBAC limitations but introduce distinct operational challenges. ABAC's policy authoring complexity requires specialized expertise, and real-time policy evaluation can introduce performance overhead in high-transaction environments [5]. Managing attribute accuracy across distributed systems remains operationally challenging, as stale attributes lead to inappropriate access decisions [6]. ReBAC faces scalability concerns with large, deeply nested relationship graphs, where traversing complex chains becomes computationally expensive [4]. Maintaining accurate relationships as organizational structures requires continuous synchronization, and transitive relationships can create unintended access paths. Additionally,

ReBAC's relative immaturity compared to RBAC means fewer standardized tools and best practices exist for implementation [4].

B. Complementary Rather than Replacement

RBAC remains one of the most widely adopted authorization mechanisms due to its simplicity, transparency, and alignment with organizational hierarchies. It excels in environments with stable structures, such as enterprises where roles (e.g., admin, developer) and privileges are well-defined and rarely change. However, RBAC's rigidity becomes a limitation in dynamic systems where access requirements evolve rapidly or depend on context-specific data.

ABAC addresses this limitation by introducing policy-based flexibility. It allows access decisions to be based on attributes that change over time, such as time of day, age [3]. This dynamic nature supports continuous access evaluation. Yet, ABAC's granularity also introduces complexity in policy definition and enforcement, which can be challenging to manage at scale.

ReBAC, as demonstrated by Google's Zanzibar [4], introduces a graph-based perspective focusing on the relationships among entities such as users, groups, and resources. This model captures nuanced access patterns prevalent in social, collaborative, and multi-tenant systems. For instance, "user A can edit document B if they are a collaborator or share a group with the document owner". ReBAC's strength lies in its ability to model these relational dynamics efficiently, often with real-time consistency guarantees.

However, it is important to note that ReBAC does not inherently subsume ABAC. While relationships define who can access what, attributes often define under what conditions access should be granted. For example, a trainer-client relationship (ReBAC) may grant access, but ABAC attributes like training session time or active membership status further refine when access is valid. Thus, a hybrid ABAC–ReBAC integration is necessary for robust and context-aware authorization [7].

C. Towards Hybrid Access Control Systems

Emerging industry and research trends increasingly favor hybrid access control architectures that integrate role-based, attribute-based, and relationship-based principles to address the complexity of authorization in dynamic and distributed environments [8],[9]. Hybrid models that combine RBAC and ABAC overcome limitations of individual paradigms by retaining RBAC's structural simplicity while introducing ABAC's contextual flexibility for fine-grained, dynamic policy evaluation [8], [10]. Additionally, hybrid frameworks incorporating relationship-based semantics (ReBAC) capture evolving trust relationships among entities, supporting access decisions in collaborative and federated systems [11]. Together, these approaches align with Zero Trust principles, where policies must adapt to both static organizational roles and dynamic operational context, as shown in Fig. 2.

In practical deployments, policy-based access control frameworks centralize policy definition while enabling distributed enforcement near the workload, such as in API gateways, service mesh proxies, and edge runtimes [10]. RBAC provides the foundational governance structure for role assignments and baseline privileges, simplifying administration and compliance oversight [8]. ABAC enhances this foundation with dynamic, context-aware decisions based on user, resource, and environmental attributes, supporting adaptive and fine-grained access in cloud-native and IoT systems [9]. ReBAC extends policy expressiveness by modeling relationships among entities, such as collaborators, delegated roles, or federated trust links, which traditional models cannot easily represent [11]. Composing these models in a unified framework enables defense-in-depth authorization, ensuring no single model becomes a point of rigidity or failure while maintaining scalable, context-aware access control across modern distributed systems [8], [10].

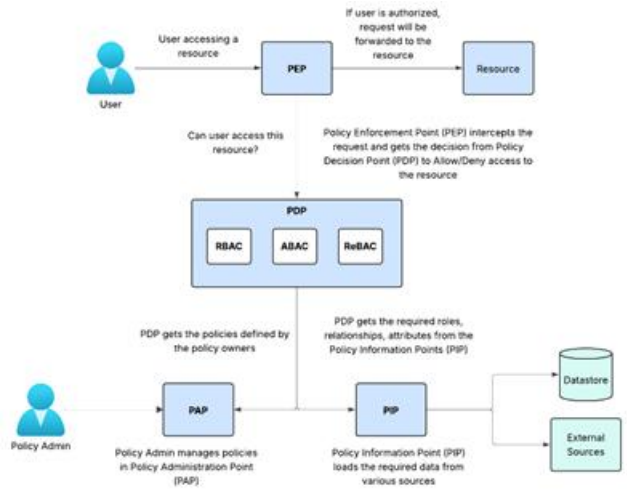

Fig. 2. Hybrid policy-based authorization framework integrating RBAC for role evaluation, ABAC for contextual policy evaluation, and ReBAC for relationship-aware access control.

D. Scalability, Performance, and Policy Governance

Scalability is another critical consideration. While RBAC scales efficiently through hierarchical role inheritance, ABAC's real-time policy evaluation and ReBAC's graph traversal can introduce computational overhead. Zanzibar suggests mitigating this through tuple-based data modeling and consistency guarantees, achieving millisecond-level authorization checks across millions of users [4]. Similarly, policy decision points (PDPs) and policy enforcement points (PEPs) in ABAC systems are often optimized through caching and pre-computation to maintain performance.

Policy governance also plays a central role in large-scale deployments. As the number of attributes and relationships increases, ensuring policy correctness, avoiding conflicts, and maintaining auditability become significant challenges. Recent research suggests using policy engineering frameworks and formal verification techniques to validate complex access rules before deployment.

E. Security and Compliance Implications

Regulatory compliance further influences model selection. RBAC aligns well with auditability and separation-of-duty requirements. ABAC enhances compliance by enabling contextual and attribute-based access controls, while ReBAC provides lineage and traceability by modeling user to resource relationships explicitly.

In highly secure systems, combining these models ensures not only that the access is correctly controlled but also that compliance requirements are demonstrably met. This multifaceted alignment of technical and regulatory priorities exemplifies the maturing discipline of robust authorization design. Table I shows the comparison of all three models.

TABLE I. COMPARISON OF AUTHORIZATION MODELS ACROSS KEY DIMENSIONS

Aspects	RBAC	ABAC	ReBAC
Complexity	Low	High	Medium
Flexibility	Low	High	High
Performance	High	Medium	High
Scalability	Medium	High	High
Context Awareness	Low	High	Medium
Relationship Modeling	Low	Medium	High
Policy Management Complexity	Low	High	Medium

V. AUTHORIZATION MODELS IN MODERN DISTRIBUTED ARCHITECTURES

Modern cloud-native architectures characterized by microservices, service meshes, serverless execution, event-driven communication, and hybrid multi-cloud deployment models dissolve traditional network perimeters and undermine implicit trust assumptions embedded in perimeter-based security models [12]. As services and workloads are distributed across heterogeneous environments and dynamically instantiated, static access control and coarse-grained network controls become insufficient to prevent lateral movement and unauthorized access [12]. Consequently, authorization must be enforced continuously and at fine granularity across users, services, and workload identities, incorporating contextual signals such as identity attributes, runtime state, and least-privilege policies [12], [13]. These requirements motivate the adoption of Zero Trust-aligned authorization frameworks that treat every access request as untrusted by default and require explicit, policy-driven authorization decisions across all access boundaries [12].

To operationalize these principles at scale, modern distributed systems increasingly adopt hybrid authorization models that decouple centralized policy governance from distributed, low-latency enforcement closer to the workload [12]. Deploying policy enforcement points alongside API gateways, service mesh proxies, and edge runtimes enables per-request authorization without incurring the latency and availability risks of centralized decision points [13], [14]. Token-centric authorization mechanisms, in which scoped and short-lived credentials are locally validated by services, further reduce runtime overhead while preserving fine-grained access control semantics [14]. Collectively, these approaches enable Zero Trust-aligned authorization to be applied consistently across cloud-native platforms without compromising the performance, scalability, and resilience requirements of highly distributed systems.

VI. CONCLUSION

This study examined the foundational distinction between authentication and authorization, emphasizing their complementary roles in establishing secure and scalable information systems. While authentication verifies the identity of users and services, authorization governs the permissible actions to authenticated entities, and failures in either dimension can undermine system security. By articulating this distinction and grounding it in practical access control models, the study highlights the importance of treating authentication and authorization as distinct yet interdependent pillars of security architecture. The discussion demonstrates that no single model universally satisfies the requirements of modern distributed systems; instead, practical deployments increasingly adopt hybrid approaches that combine these paradigms to balance manageability, flexibility, and security guarantees.

Finally, the study situates authorization frameworks within the context of modern cloud-native and Zero Trust architectures, where implicit trust assumptions are no longer tenable, and access decisions must be continuously evaluated across users, services, and workloads. In such environments, scalable authorization frameworks that integrate centralized policy governance with distributed, low-latency enforcement are essential for achieving both strong security and high performance. Ultimately, optimal security posture emerges not from the isolated deployment of authentication or authorization mechanisms, but from their cohesive, policy-driven integration across system boundaries, enabling organizations to reduce attack surfaces while maintaining the agility and resilience demanded by contemporary distributed architectures.

REFERENCES

[1] NIST, "Zero Trust Architecture," NIST Special Publication 800-207, National Institute of Standards and Technology, August 2020.

[2] M. Hu and D. Ferraiolo, "Attribute-based Access Control," IEEE Computer, vol. 48, no. 2, pp. 85–88, 2015.

[3] P. W. Fong, "Relationship-based access control: Protection model and policy language," Proc. of CODASPY, 2011, pp. 191–202.

[4] R. Pang, N. Jain, H. Majumdar, G. Maldonado, C. Kochhar, I. Jayachandran, C. Taylor, et al., "Zanzibar: Google's Consistent, Global Authorization System," USENIX ATC, 2019.[1] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, "Attribute-Based Access Control," IEEE Computer, vol. 48, no. 2, pp. 85-88, 2015

[5] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, "Attribute-Based Access Control," IEEE Computer, vol. 48, no. 2, pp. 85-88, 2015

[6] S. Bhatt, F. Patwa, and R. Sandhu, "An Attribute-Based Access Control Extension for OpenStack and its Enforcement Utilizing the Policy Machine," Proceedings of the 2nd IEEE International Conference on Collaboration and Internet Computing, pp. 37-45, 2016.

[7] S. Ghosh et al., "Hybrid Access Control Models for Distributed Systems," IEEE Access, vol. 9, pp. 105611–105624, 2021

[8] Y. Meng, "Research on the Implementation of RBAC–ABAC Hybrid Models in Large-Scale Distributed Architectures," Adv. Comput. Commun., vol. 2025, Sep. 30, 2025.

[9] R. Wang, C. Li, K. Zhang, et al., "Zero-Trust Based Dynamic Access Control for Cloud Computing," Cybersecurity, vol. 8, art. no. 12, Feb. 16, 2025.

[10] S. Ameer, J. Benson, and R. Sandhu, "Hybrid Approaches (ABAC and RBAC) Toward Secure Access Control in Smart Home IoT," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 5, pp. 4032-4051, Sept.-Oct. 2023, doi: 10.1109/TDSC.2022.3216297.

[11] Y. Xu, W. Zhou, H. Han, et al., "A Secure and Scalable IoT Access Control Framework with Dynamic Attribute Updates and Policy Hiding," Sci. Reports, vol. 15, art. no. 11913, Apr. 7, 2025.

[12] R. Chandramouli, S. Rose, M. McCarthy, and N. Shetty, "Zero Trust Architecture," NIST Special Publication 800-207, National Institute of Standards and Technology, Gaithersburg, MD, USA, Aug. 2020.

[13] A. Venčkauskas, R. Damaševičius, and J. Toldinas, "Authentication and Authorization in Microservices Architecture: A Systematic Literature Review," Axioms, vol. 12, no. 2, 2023.

[14] Cloud Native Computing Foundation (CNCF), "The Five Laws of Cloud Native Authorization," Jan. 2023.