Beyond Tokens: A Zero-Trust AuthZ Protocol for the Agentic Era

Nicola Gallo[1]

Abstract— **This paper addresses some of the security challenges posed by AI agents, framing them as part of the broader problem of distributed systems. It draws parallels with enterprise patterns for distributed systems and aims to lay the foundation for a new authorization (AuthZ) protocol. The key insight is that tokens, while useful, are not sufficient: authorization must move beyond a token-centric model. Instead, what is needed is a new zero-trust-authz protocol that, when combined with tokens, can support the development of next-generation AuthZ systems.**

In practice, companies will continue to rely on standards such as OIDC and SAML, and OAuth will certainly remain an initiator for many flows. However, the real gap lies in what happens after token delivery — a space where existing protocols have paid little attention, but which is critical for the future of secure and dynamic authorization. The proposed model also leaves room for the integration of decentralized approaches, such as UCAN, ZCAP-LD, and other capability-based frameworks, enabling hybrid ecosystems that combine traditional standards with emerging decentralized authorization models.

I. Introduction

THE **Agentic Era** is emerging in industry discussions, as many recognize that certain assumptions no longer hold and past trade-offs are becoming increasingly fragile. Change can often be unsettling, especially when existing systems and business models are built on established approaches. Yet it may be time to start with a blank page and re-examine the compromises we have inherited.

The community should take this as a challenge to rethink the **foundations of authorization**, especially in the context of distributed systems and **AI agents**.

In the near future, companies will continue to rely on **OAuth** to authenticate humans, and on solutions such as **WIMSE** or **SPIFFE** to authenticate non-human workloads. Yet at some point, we must stop patching legacy models and acknowledge that **new architectures demand new foundations**.

This new foundation can open entirely new markets. It can still leverage existing technologies, but with a fresh approach that enables **innovation**, **new value creation**, and the development of novel products. Industry has been built around **identities**, and most importantly this new foundation could enable the real adoption of **AI agents** within the enterprise.

The **authentication layer** is relatively solid, whether using centralized or decentralized approaches, but **authorization** still works only under certain **trade-offs**. These trade-offs cannot support AI agents and, more generally, distributed systems.

History of science teaches us that progress often comes from looking across disciplines, borrowing solutions, and applying them in new contexts, that is what this paper aims to do looking at **Enterprise Patterns** for distributed systems. It is also important to consider this in the context of **Zero Trust Network Access (ZTNA)** and, more broadly, the **Zero Trust model**, which will be crucial for the future of secure access.

This paper adopts such an approach, challenging the core trade-off that the industry has relied on: the **identity- and HTTP-centric model**. An **AI agent protocol** would need to include many other features; however, this paper focuses specifically on **authorization (authz)**.

II. Tokens in the Authorization Perspective

Tokens have worked very well for **authentication (AuthN)**. An **authentication token** simply states: *"I am Nicola"*. If this statement is true now, it will remain true ever from now.

However, there are essentially two problems:

1) **Trust**: anyone can claim such a statement. To address this, tokens must be **signed**, giving rise to mechanisms such as **JWTs**.
2) **Replay**: anyone can replay the same token. The only option to mitigate this risk is to add an **expiration time**.

These are **trade-offs** and represent inherent **limits of security**. In practice, tokens exploit a **PACELC-style trade-off**: in the absence of partitions, they take advantage of the **latency–consistency** trade-off.

> The **PACELC theorem** [9] states that in case of **network partitioning (P)** in a **distributed system**, one has to choose between **availability (A)** and **consistency (C)** (as per the CAP theorem), but else (E), even when the system is running normally in the absence of partitions, one has to choose between **latency (L)** and **loss of consistency (C)**.

This trade-off becomes dangerous when tokens represent **authorization (AuthZ)**. For example, *"Nicola is entitled to a certain role now"* may be true at one moment but false only a few seconds later. Here **tokens become fragile**: while they have worked well for **authentication**, it is a mistake to assume they must also be the **golden standard for authorization**.

Below we present a more formal representation of this problem.

A. Temporal Validity and Risk

Let T_a be the validity period of an **authentication token** and T_z the validity of an **authorization**. To ensure security:

$$T_z \ll T_a, \quad \text{ideally } T_z \to 0$$

In practice, T_z should be kept as short as feasible, though operational trade-offs may apply.

The **risk of compromise** grows with validity:

$$R(T) = f(T), \quad f'(T) > 0$$

Thus, **long-lived static authorization tokens** increase exposure.

[1]Nicola Gallo, Software Architect at Nitro Agility. This work was carried out as part of professional activities within Nitro Agility and should be understood in that professional context. nicola.gallo@nitroagility.com

B. Transitive Delegation

A **delegation** between A and B is denoted:

$$D_i : A \to B$$

A chain of n delegations is:

$$D_1 \circ D_2 \circ \cdots \circ D_n$$

If a token explicitly encodes a single linear chain of n delegations, its size grows linearly:

$$O(n)$$

However, when delegation supports **branching**, **conditions**, or **attenuation**, the number of possible valid paths increases combinatorially. Encoding all possible transitive delegations into a **static token** leads to exponential growth in the worst case:

$$O(2^n)$$

This is not scalable. Therefore, **authorization must be reconstructed dynamically at request time**.

C. Context and Runtime Validity

System state evolves with time:

$$S(t) = \{E, W, P\}$$

where E are **entities**, W **workloads**, and P **policies**.

At request time t_r, validity requires:

$$\exists p \in P \ \text{ valid for } (e, w) \mid t = t_r$$

Static tokens cannot guarantee consistency with the **live system state** $S(t_r)$.

D. Trust Model

Authorization is a function of the **live context**:

$$T(E, W, P, t) = \begin{cases} 1 & \text{if authorized at time } t \\ 0 & \text{otherwise} \end{cases}$$

Static tokens implicitly encode **authorization state** at issuance time t_0, and reuse assumes that the same decision holds for all later times:

$$T(E, W, P, t_0) = T(E, W, P, t), \ \forall t > t_0$$

which is false in **dynamic systems**.

E. Conclusion

Token-based authorization is:

1) **Insecure**: $R(T)$ increases with T.
2) **Non-scalable**: **delegation graphs** may induce combinatorial explosion (up to $O(2^n)$).
3) **Inconsistent**: $S(t) \neq S(t_0)$ in **dynamic systems**.

Therefore, **authorization must be computed on-demand**, consistent with **Zero Trust principles**.

F. Practical Considerations

The model above is a **theoretical ideal**; in practice it is almost impossible to achieve perfectly. As in many areas of **security**, trade-offs must be made to balance **feasibility** and **protection**. What matters is not the invention of a **"better token"** but the design of a more **sophisticated authorization engine** capable of assessing **multiple sources of trust** — including **tokens**, **capabilities** (e.g. zcaps), **contextual signals**, and other **dynamic inputs** — in a **best-effort approach**.

The key point is that the **next generation of authorization protocols** cannot remain **token-centric**. Instead, **authorization** must emerge from a richer machinery that integrates **diverse identities** and **trust sources**, and can even operate *without* **tokens** altogether.

III. The Industry Evolution to Zero Trust

Cloud computing has reshaped how companies build software, shifting the focus to **scalable solutions** that leverage **elastic infrastructure**. Applications are increasingly **containerized** and managed with platforms such as **Kubernetes**, while **serverless computing** has also gained traction.

Being inherently **distributed**, **cloud-native systems** face the challenges outlined by the **CAP theorem**: architects must balance **Partition Tolerance**, **Consistency**, and **Availability**. This has led to designs that relax **strong consistency** in favor of **availability** through **Eventual Consistency**.

> The **CAP theorem** [1] states that, in the presence of a **network partition**, a system must trade off between **Consistency** and **Availability**. **Eventual Consistency** aligns with **PA (Partition Tolerance and Availability)**, allowing a system to remain responsive during partitions while deferring full consistency until recovery.

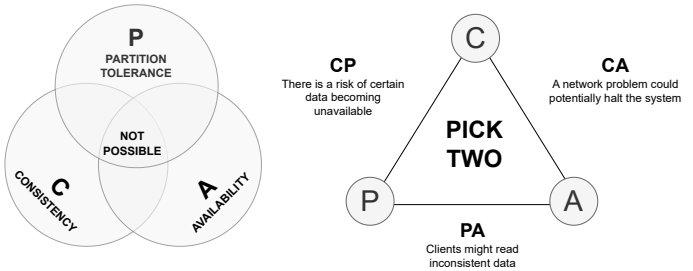

Fig. 1. CAP Theorem

Another major trend is the adoption of **asynchronous patterns** and **event-streaming frameworks** such as **Apache Kafka**, which handle continuous flows of events without a clear beginning or end. In this setting, **orchestration** has proven complex and difficult to scale, whereas **choreography-based systems** often provide better scalability.

Figure 2 shows a typical **cloud architecture**: clients authenticate with a centralized **Identity Provider**, which issues a **token** subsequently used to access server resources. Two widely adopted protocols here are **OpenID** [4] for authentication and **OAuth** [5] for authorization.

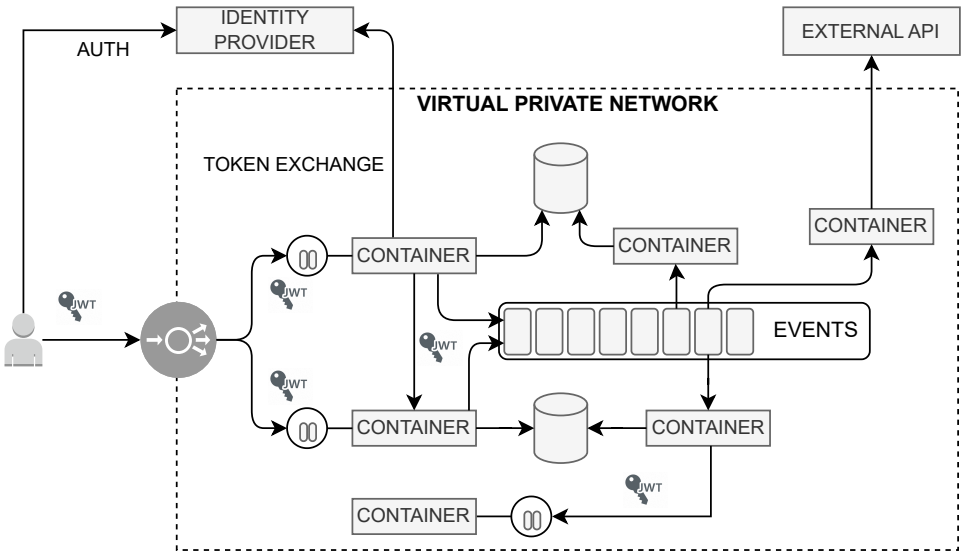

Fig. 2. Cloud Architecture

Within these protocols, **JSON Web Tokens (JWT)** [6] have become central. Compact and digitally signed, they ensure **message integrity** and enable **secure communication** between client and server.

Access control remains a critical concern. Models such as **Role-Based Access Control (RBAC)** and **Attribute-Based Access Control (ABAC)** have paved the way for **Policy-Based Access Control (PBAC)** [7].

A further challenge arises in **transport-layer communication**, for example with **Kafka** and **stream-processing systems**, where it is not always feasible to propagate a token.

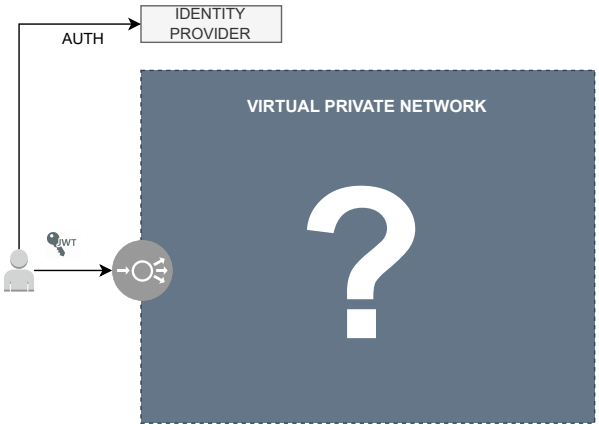

Fig. 3. Building inside a Perimeter

This exposes a key limitation of the **current model**: it works at the **perimeter**, where **tokens** are validated at delivery, but what happens **inside the enterprise** afterward is not standardized. **Authorization** is essentially assumed to remain valid once accepted at the boundary, and the system continues to treat it as true for the rest of the time. While this trade-off was once acceptable, it cannot be sustained in environments that must support **AI agents**.

Moreover, modern **distributed systems** increasingly operate *without clear perimeters*. In **cloud-native**, **multi-tenant**, and **hybrid environments**, workloads, services, and agents continuously interact across organizational and network boundaries. In such scenarios, the very assumption of a **fixed perimeter** for validation no longer holds, making **token-based authorization** even more fragile.

It follows that the **current model** is not suitable for **AI agents**, which require **dynamic**, **context-aware**, and **continuously validated authorization**. The goal is not to replace existing protocols but to complement them, addressing the **gaps** they have historically overlooked.

IV. Zero Trust AuthZ Protocol

In order to build a **Zero Trust AuthZ protocol**, it is necessary to build around **trust**, and design an engine capable of elaborating this trust dynamically, filling the gap beyond simple token delivery. This work has started under the name **ZTAuth*** (**Zero Trust Auth***). Such a system must rely on **autonomous components**, able to operate even in case of disconnections.

A Zero Trust architecture must include a **Policy Decision Point (PDP)** built around Zero Trust principles. The first point to understand is that when a PDP is asked to take a decision, there are always **two identities** involved:

- the identity of the **workload** requesting the decision (it cannot be manual, otherwise it would not scale);
- the identity of the **subject** (human or non-human) that is the target of the evaluation.

These identities may be expressed via **tokens**, **SPIFFE**, **ZCAP-LD,**, or any other valid identity proof.

In this way, a **Trust Chain** can be built as the composition of multiple **Trust Inputs**, possibly with parallel branches. These inputs must connect with autonomous components and be usable across multiple transport layers.

The PDP receives as input an **authorization context**, which includes:

- the subject identity proof and related attributes (possibly from a **Policy Information Point (PIP)**),
- the workload identity proof,
- information about the action requested.

Each of these is a **trust input**, which may be a token, a contextual signal, or other forms of trust such as **UCAN** or **ZCAP-LD**, depending on the use case and flow.

To evaluate a request, the PDP must already have access to:

- a synchronized set of **policies**,
- additional **trust statements** (e.g., UCAN, ZCAP-LD, or others),
- a mechanism for decision **logging and auditing**.

During evaluation, the PDP creates an **authorization context** for the workload, verifying whether it can act on behalf of the subject. If conditions are satisfied, the system may perform a **Trust Elevation**, allowing the workload to inherit subject permissions. System owners must also be able to define under which conditions this is allowed, possibly grouping workloads into **Trust Levels**, which require synchronization and governance.

> Example: John has a valid OAuth token to print a document, but policy restricts classified documents to printers in a secure segment with verified workload identity. Even with a valid token, printing must be blocked if the printer is not trusted at that moment.

Finally, the model must define how **Trust Delegation** works, ensuring safe delegation of rights across entities.

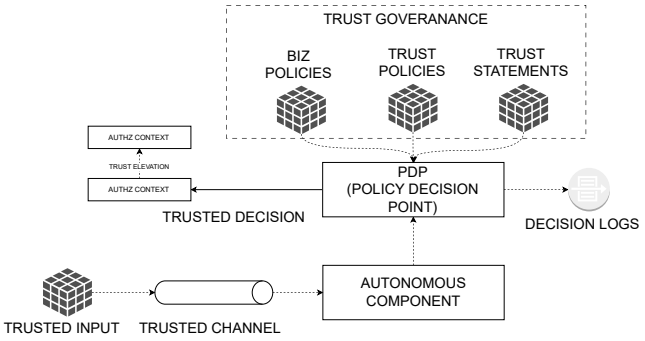

Fig. 4. AuthZ Protocol

To enable such a protocol reffig:authzprotoco , it is necessary to define and standardize:

- **Trust Inputs**,
- **Trust Decisions**,
- **Trust Elevation**,
- **Trust Levels**,
- **Trust Delegation**.

V. Key Aspects to Analyze for Protocol Evaluation

To properly evaluate a Zero Trust authorization protocol, the following aspects should be analyzed:

- **Communication:** Consider the use of state machines and workflows (this paper uses state machines for simplicity). Each node should be capable of making a decision based on policies and then invoking the next node in the chain.
- **Transport Layer:** Analyze how information flows across edges using different transport layers. Examples include traditional APIs or event-driven systems such as Kafka.
- **Flow Types:** Different types of flows may exist (e.g., returning a direct result or not). Why not have an agent that queries systems and provides a consolidated status? Even feedback itself could be delegated to another agent, making flows more resilient when intermediate agents are down.
- **Scopes:** Analyze the request scope and how down-scope propagation is handled across services and agents.
- **Long-Running Processing:** Workflows may last a long time, especially when implemented over Kafka or similar event-streaming platforms. For example, handling an employee's last day in a company may require multi-step, long-running authorization checks.
- **Rollback and Recovery:** Consider how rollback and recovery are managed in case of failures during authorization workflows.
- **Two-Identity-One-Action:** Authorization typically revolves around resources, but every action involves at least two identities: the workload requesting the action and the subject (human or non-human). To prevent impersonation, machine identity must always be included.
- **Delegation:** Handling impersonation or "acting on behalf of" models introduces delegation challenges. A robust protocol must define secure delegation mechanisms.
- **Federation:** Authorization must work across federated organizations, which introduces challenges in calling into external domains.
- **Governance:** Authorization governance is a mix of user-level policies and platform-level priorities. Both dimensions must be managed together.
- **Auditing:** Define how auditing can be applied to authorization workflows to ensure accountability and traceability.

VI. Market Landscape

Looking beyond simple token delivery, companies can start building solutions that operate without implicit trust, fully aligning with Zero Trust principles. This approach is particularly well-suited for Zero Trust Network Access (ZTNA) and the emerging requirements of AI agents.

In this scope, there are significant business opportunities. Enterprises will need new forms of **Zero Trust governance**, capable of auditing and controlling authorization decisions in real time. A new generation of **Policy Decision Points (PDPs)** can enable innovative products around trust evaluation and enterprise decision governance.

In parallel, logging providers will play a central role by developing tools to capture, analyze, and interpret decision logs. As highlighted by Wardley Maps, while **policy languages** are rapidly becoming commoditized, the true differentiators will be stronger **trust models**, **dynamic permissions**, and advanced **anomaly detection**.

This ecosystem will be essential to manage AI agents. Agents will connect to existing applications and extend them with intelligent capabilities, but companies will still need a unified solution to govern their software landscape.

For this reason, a next-generation Zero Trust authorization

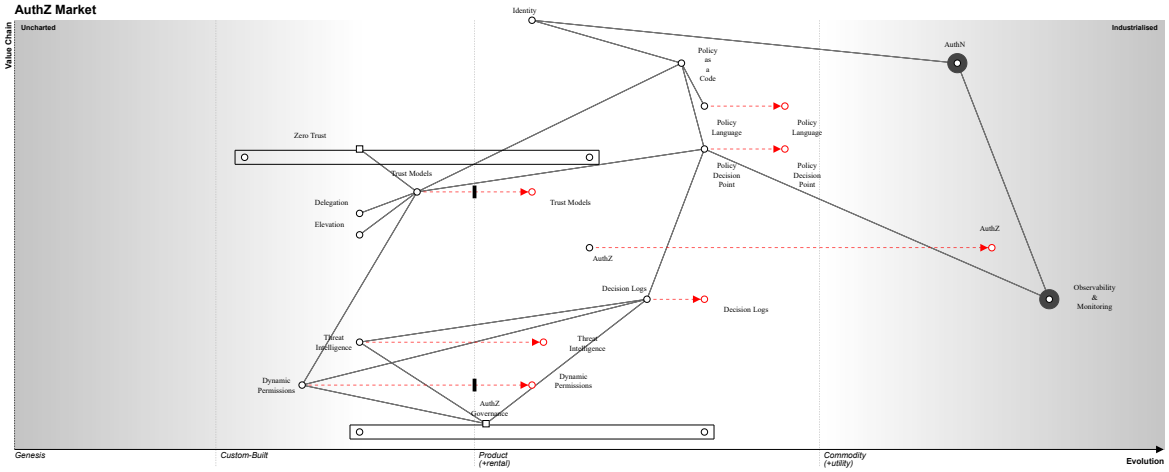

Fig. 5. Wardley Maps for Zero Trust AuthZ

protocol is required to achieve the next level of security and governance.

The **Wardley Map** in Figure 5 illustrates a possible **market landscape** for the evolution of Zero Trust and authorization technologies. It shows how **policy languages** are moving towards **commoditization**, while higher-order capabilities such as **trust models**, **dynamic decision points**, and **anomaly detection** remain areas of differentiation and innovation. This visualization provides a framework for anticipating how the market may evolve and where new opportunities could emerge.

ACKNOWLEDGMENTS

This text was originally written by the author in English. As a non-native English speaker, AI-based writing tools were employed solely to refine grammar, clarity, and fluency in order to speed up the writing process and reduce language errors. All ideas, arguments, and original contributions are entirely the author's own.

The views and proposals expressed in this paper are intended for discussion and research purposes only. They should not be interpreted as finalized standards or production-ready specifications.

REFERENCES

[1] Brewer, Eric. (2000). Towards robust distributed systems. PODC. 7. 10.1145/343477.343502.
[2] V. Velepucha and P. Flores, "A Survey on Microservices Architecture: Principles, Patterns and Migration Challenges," in IEEE Access, vol. 11, pp. 88339-88358, 2023, doi: 10.1109/ACCESS.2023.3305687.
[3] V. Vernon, Domain-Driven Design Distilled. Reading, MA, USA: Addison-Wesley Professional, 2016.
[4] [OpenID.2.0] OpenID Foundation, "OpenID Authentication 2.0," December 2007.
[5] [RFC6749] M. Jones, D. Hardt, "The OAuth 2.0 Authorization Framework," October 2012.
[6] [RFC6750] M. Jones, D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage", October 2012
[7] L. Zhi, W. Jing, C. Xiao-su and J. Lian-xing, "Research on Policy-based Access Control Model," 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, China, 2009, pp. 164-167, doi: 10.1109/NSWCTC.2009.313.
[8] Nicola Gallo, Antonio Radesca, "A Multi-Account and Multi-Tenant Policy-Based Access Control (PBAC) Approach for Distributed Systems Augmented with Risk Scores Generation".
[9] D. J. Abadi, "Consistency Tradeoffs in Modern Distributed Database System Design: CAP is Only Part of the Story," *IEEE Computer*, vol. 45, no. 2, pp. 37–42, Feb. 2012. DOI: 10.1109/MC.2012.33