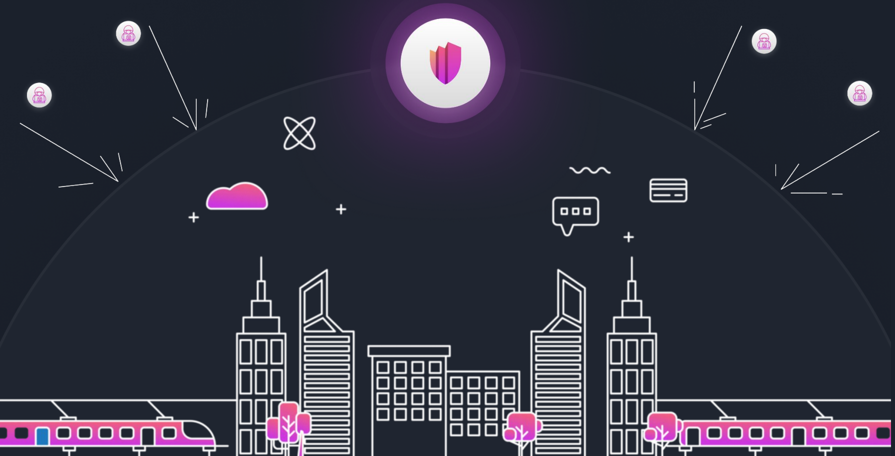
ZTAuth*

Revolutionizing AuthN and AuthZ with Autonomous-Disconnected Challenge

Nicola Gallo, Antonio Radesca (Nitro Agility Srl)



Permguard NitroAgility

TOC

Speaker

Nicola Gallo
Co-founder at Nitro Agility S.r.l.

NitroAgility Permguard

Overview

Zero Trust AuthN/AuthZ **Models** and **Trusted Delegations**

(Zee-Tee-Auth-Star)

ZTAuth*

ZT highlights the adherence to Zero Trust principles

Auth* specifies an approach focused on authentication (AuthN) and authorization (AuthZ). It also includes concepts like trusted elevation and trusted delegation.

NEW

Overview

ZTAuth*: **Zero Trust** AuthN/AuthZ **Models** and **Trusted Delegations**

ZTAuth* was created to address the **Autonomous-Disconnected-Driven** challenge using **Zero Trust** principles.

Spec: https://github.com/ztauthstar/ztauthstar-specs

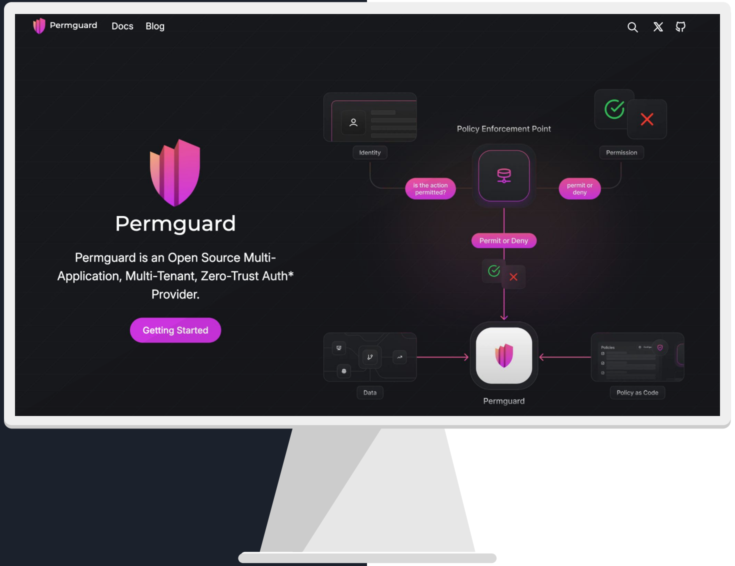
Publications: https://medium.com/ztauth

Paper:https://github.com/autorizzami/autorizzami-research-paper/blob/main/autorizzami.pdf

Contact us opensource@nitroagility.com

Overview

ZTAuth* is **more** than just a **specification effort**.



Permguard

Permguard is an Open Source Multi-Application, Multi-Tenant, Zero-Trust Auth* Provider.

Getting Started

Permguard

Permguard is an Open Source Multi-Application, Multi-Tenant, Zero-Trust Auth* Provider.

license Apache-2.0

www.permguard.com

GitHub **github.com/permguard**

Zero Trust principles.

Never trust, always verify: Never trust implicitly; always verify the identity and context of users, devices, and applications before granting access.

Least privilege access: Grant the minimum level of access necessary for a task, ensuring users or systems only interact with the resources they truly need.

Assume breach: Operate under the assumption that a breach could occur at any time, designing systems to contain potential damage and prevent lateral movement.

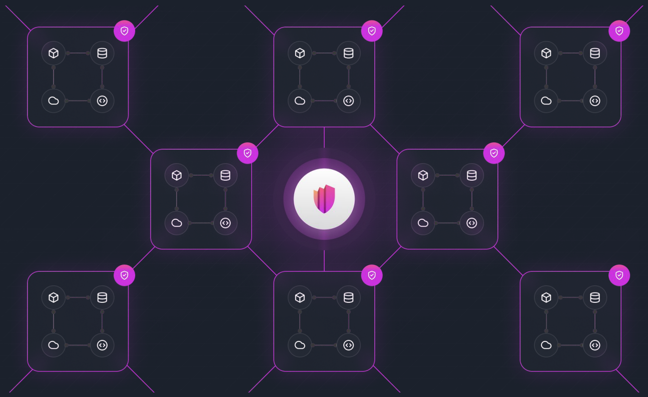
Zero Trust (ZTNA vs ZTAuth*)

ZTNA

Zero Trust Network Access: Ensures secure, identity-based access to networks or applications by applying least privilege at the network boundary.

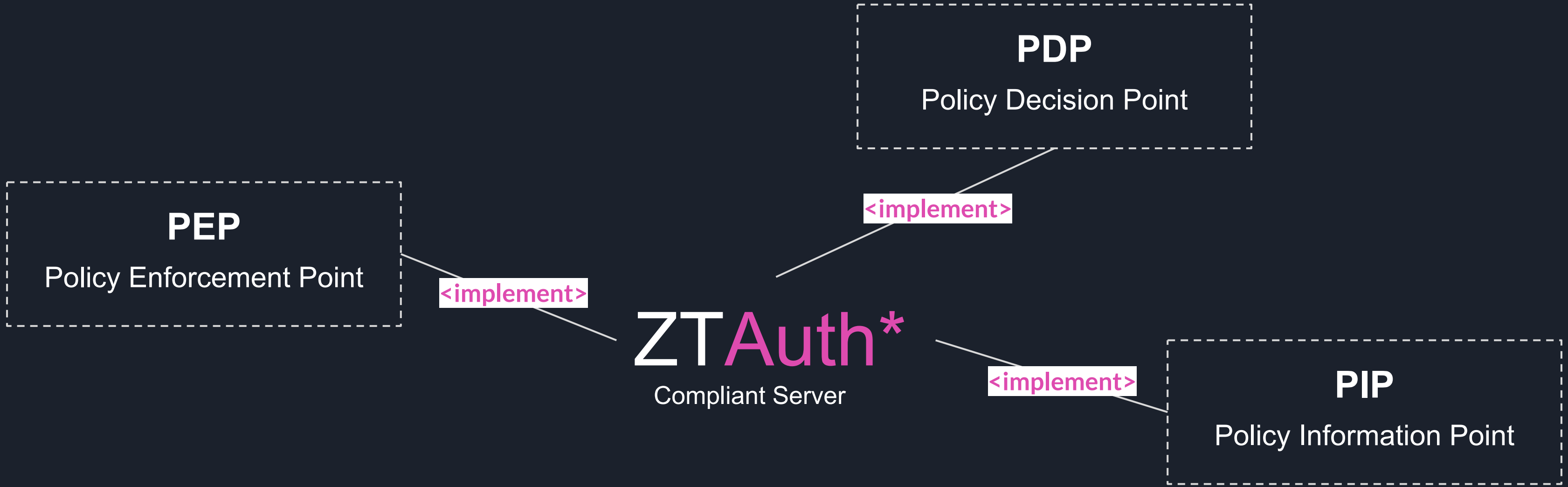
ZTAuth*

Zero Trust Auth*: Ensures secure, identity-based execution of actions on resources by enforcing least privilege at the application boundary. Built for eventual consistency, the security model is incrementally synchronized across applicative nodes in an immutable, versioned manner.



ZTAuth* key concepts.

ZTAuth*

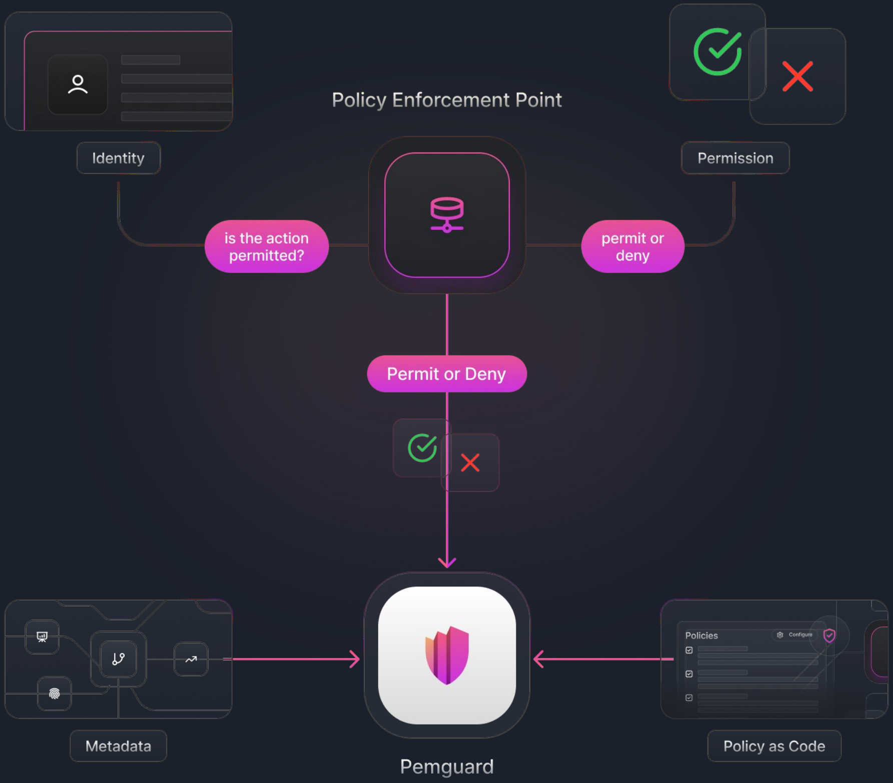

PDP
Policy Decision Point
<implement>
PEP
Policy Enforcement Point
<implement>
ZTAuth*
Compliant Server
<implement>
PIP
Policy Information Point

ZTAuth*

Permguard

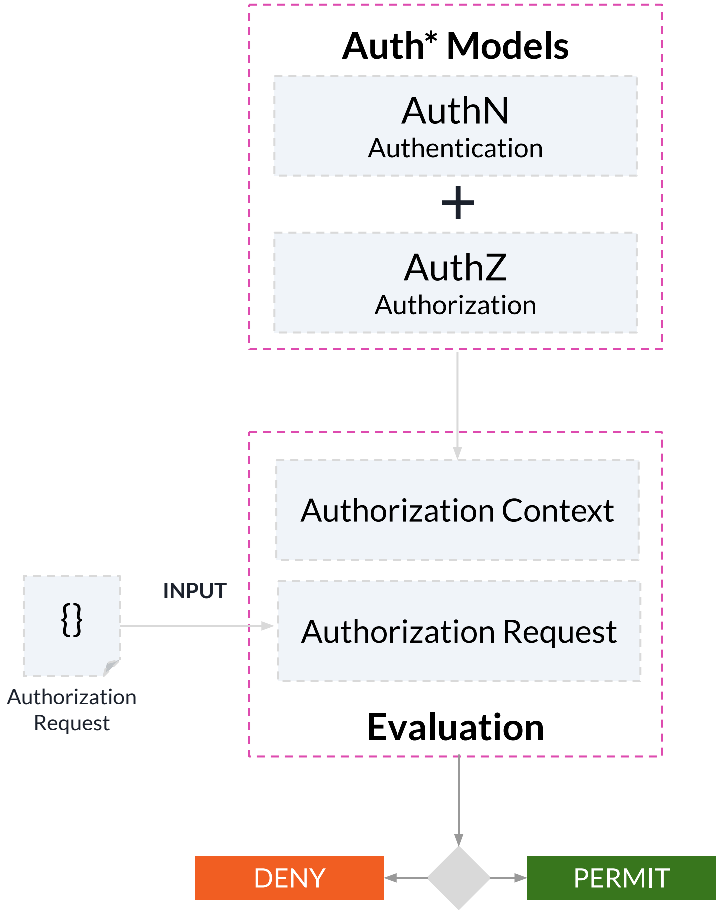
The ZTAuth* compliant server like Permguard:

- **input:** authorization request, which include the subject, resource, action, and context
- **evaluate:** create an authorization context using the Auth* models (AuthN and AuthZ)
- **output:** a decision on whether the request is permitted or denied.



The **ZTAuth* decision** flow.

Permguard



Auth* Models

AuthN

An AuthN model include the informations about Identity Actors and the Identity Types: User, Roles, Groups.

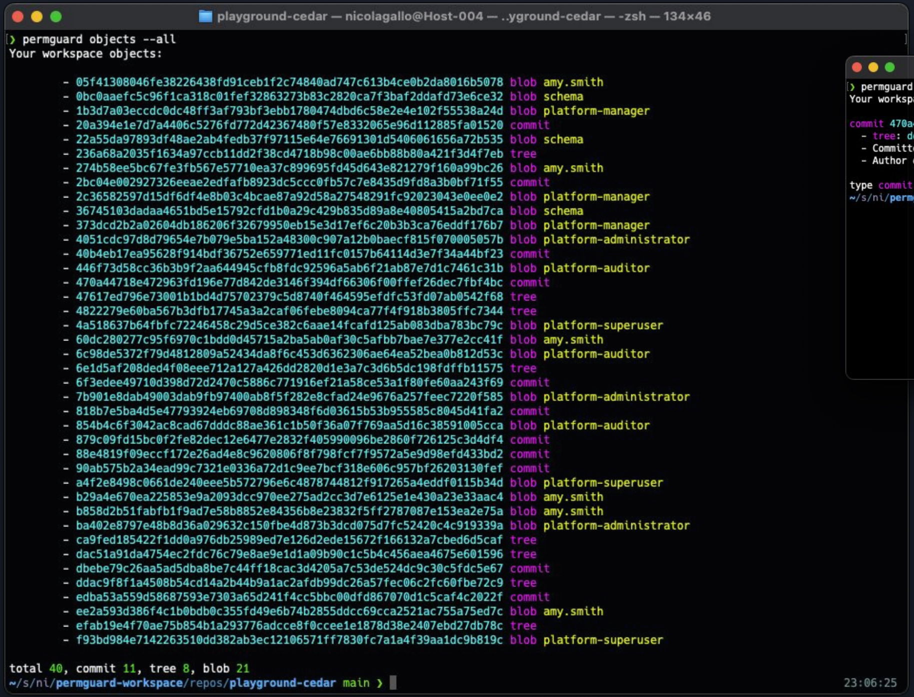
AuthZ

An AuthZ model include:

- **Policy Ledger:** A Git-Like objectstore designed to securely store Policies with guaranteed immutability and versioning.
- **Trusted Elevation**: A statement that represents the ability of an Identity to elevate its authorization context to match that of another Identity.
- **Trusted Delegation**: A statement that defines the ability to manage delegation scenarios, allowing an Identity to act on behalf of another.

Policy Ledger

Permguard

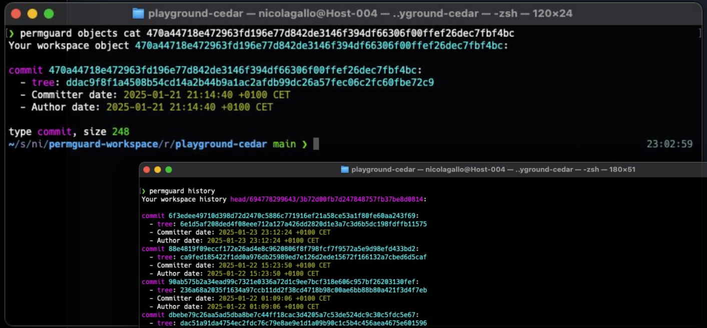


playground-cedar — nicolagallo@Host-004 — ..yground-cedar — -zsh — 134×46

```
❯ permguard objects --all
Your workspace objects:

        - 05f41308046fe38226438fd91ceb1f2c74840ad747c613b4ce0b2da8016b5078 blob amy.smith
        - 0bc0aaefc5c96f1ca318c01fef32863273b83c2820ca7f3baf2ddafd73e6ce32 blob schema
        - 1b3d7a03eccdc0dc48ff3af793bf3ebb1780474dbd6c58e2e4e102f55538a24d blob platform-manager
        - 20a394e1e7d7a4406c5276fd772d42367480f57e8332065e96d112885fa01520 commit
        - 22a55da97893df48ae2ab4fedb37f97115e64e76691301d5406061656a72b535 blob schema
        - 236a68a2035f1634a97ccb11dd2f38cd4718b98c00ae6bb88b80a421f3d4f7eb tree
        - 274b58ee5bc67fe3fb567e57710ea37c899695fd45d643e821279f160a99bc26 blob amy.smith
        - 2bc04e002927326eeae2edfafb8923dc5ccc0fb57c7e8435d9fd8a3b0bf71f55 commit
        - 2c36582597d15df6df4e8b03c4bcae87a92d58a27548291fc92023043e0ee0e2 blob platform-manager
        - 36745103dadaa4651bd5e15792cfd1b0a29c429b835d89a8e40805415a2bd7ca blob schema
        - 373dcd2b2a02604db186206f32679950eb15e3d17ef6c20b3b3ca76eddf176b7 blob platform-manager
        - 4051cdc97d8d79654e7b079e5ba152a48300c907a12b0baecf815f070005057b blob platform-administrator
        - 40b4eb17ea95628f914bdf36752e659771ed11fc0157b64114d3e7f34a44bf23 commit
        - 446f73d58cc36b3b9f2aa644945cfb8fdc92596a5ab6f21ab87e7d1c7461c31b blob platform-auditor
        - 470a44718e472963fd196e77d842de3146f394df66306f00ffef26dec7fbf4bc commit
        - 47617ed796e73001b1bd4d75702379c5d8740f464595efdfc53fd07ab0542f68 tree
        - 4822279e60ba567b3dfb17745a3a2caf06febe8094ca77f4f918b3805ffc7344 tree
        - 4a518637b64fbfc72246458c29d5ce382c6aae14fcafd125ab083dba783bc79c blob platform-superuser
        - 60dc280277c95f6970c1bdd0d45715a2ba5ab0af30c5afbb7bae7e377e2cc41f blob amy.smith
        - 6c98de5372f79d4812809a52434da8f6c453d6362306ae64ea52bea0b812d53c blob platform-auditor
        - 6e1d5af208ded4f08eee712a127a426dd2820d1e3a7c3d6b5dc198fdffb11575 tree
        - 6f3edee49710d398d72d2470c5886c771916ef21a58ce53a1f80fe60aa243f69 commit
        - 7b901e8dab49003dab9fb97400ab8f5f282e8cfad24e9676a257feec7220f585 blob platform-administrator
        - 818b7e5ba4d5e47793924eb69708d898348f6d03615b53b955585c8045d41fa2 commit
        - 854b4c6f3042ac8cad67dddc88ae361c1b50f36a07f769aa5d16c38591005cca blob platform-auditor
        - 879c09fd15bc0f2fe82dec12e6477e2832f405990096be2860f726125c3d4df4 commit
        - 88e4819f09eccf172e26ad4e8c9620806f8f798fcf7f9572a5e9d98efd433bd2 commit
        - 90ab575b2a34ead99c7321e0336a72d1c9ee7bcf318e606c957bf26203130fef commit
        - a4f2e8498c0661de240eee5b572796e6c4878744812f917265a4eddf0115b34d blob platform-superuser
        - b29a4e670ea225853e9a2093dcc970ee275ad2cc3d7e6125e1e430a23e33aac4 blob amy.smith
        - b858d2b51fabfb1f9ad7e58b8852e84356b8e23832f5ff2787087e153ea2e75a blob amy.smith
        - ba402e8797e48b8d36a029632c150fbe4d873b3dcd075d7fc52420c4c919339a blob platform-administrator
        - ca9fed185422f1dd0a976db25989ed7e126d2ede15672f166132a7cbed6d5caf tree
        - dac51a91da4754ec2fdc76c79e8ae9e1d1a09b90c1c5b4c456aea4675e601596 tree
        - dbebe79c26aa5ad5dba8be7c44ff18cac3d4205a7c53de524dc9c30c5fdc5e67 commit
        - ddac9f8f1a4508b54cd14a2b44b9a1ac2afdb99dc26a57fec06c2fc60fbe72c9 tree
        - edba53a559d58687593e7303a65d241f4cc5bbc00dfd867070d1c5caf4c2022f commit
        - ee2a593d386f4c1b0bdb0c355fd49e6b74b2855ddcc69cca2521ac755a75ed7c blob amy.smith
        - efab19e4f70ae75b854b1a293776adcce8f0ccee1e1878d38e2407ebd27db78c tree
        - f93bd984e7142263510dd382ab3ec12106571ff7830fc7a1a4f39aa1dc9b819c blob platform-superuser

total 40, commit 11, tree 8, blob 21
~/s/ni/permguard-workspace/repos/playground-cedar main ❯                                       23:06:25
```

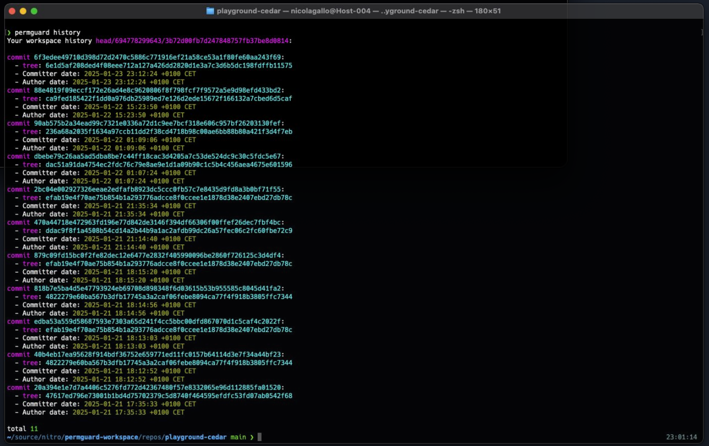


playground-cedar — nicolagallo@Host-004 — ..yground-cedar — -zsh — 120×24

```
❯ permguard objects cat 470a44718e472963fd196e77d842de3146f394df66306f00ffef26dec7fbf4bc
Your workspace object 470a44718e472963fd196e77d842de3146f394df66306f00ffef26dec7fbf4bc:

commit 470a44718e472963fd196e77d842de3146f394df66306f00ffef26dec7fbf4bc:
  - tree: ddac9f8f1a4508b54cd14a2b44b9a1ac2afdb99dc26a57fec06c2fc60fbe72c9
  - Committer date: 2025-01-21 21:14:40 +0100 CET
  - Author date: 2025-01-21 21:14:40 +0100 CET

type commit, size 248
~/s/ni/permguard-workspace/r/playground-cedar main ❯                                   23:02:59
```



playground-cedar — nicolagallo@Host-004 — ..yground-cedar — -zsh — 180×51

```
❯ permguard history
Your workspace history head/694778299643/3b72d00fb7d247848757fb37be8d0814:

commit 6f3edee49710d398d72d2470c5886c771916ef21a58ce53a1f80fe60aa243f69:
  - tree: 6e1d5af208ded4f08eee712a127a426dd2820d1e3a7c3d6b5dc198fdffb11575
  - Committer date: 2025-01-23 23:12:24 +0100 CET
  - Author date: 2025-01-23 23:12:24 +0100 CET
commit 88e4819f09eccf172e26ad4e8c9620806f8f798fcf7f9572a5e9d98efd433bd2:
  - tree: ca9fed185422f1dd0a976db25989ed7e126d2ede15672f166132a7cbed6d5caf
  - Committer date: 2025-01-22 15:23:50 +0100 CET
  - Author date: 2025-01-22 15:23:50 +0100 CET
commit 90ab575b2a34ead99c7321e0336a72d1c9ee7bcf318e606c957bf26203130fef:
  - tree: 236a68a2035f1634a97ccb11dd2f38cd4718b98c00ae6bb88b80a421f3d4f7eb
  - Committer date: 2025-01-22 01:09:06 +0100 CET
  - Author date: 2025-01-22 01:09:06 +0100 CET
commit dbebe79c26aa5ad5dba8be7c44ff18cac3d4205a7c53de524dc9c30c5fdc5e67:
  - tree: dac51a91da4754ec2fdc76c79e8ae9e1d1a09b90c1c5b4c456aea4675e601596
  - Committer date: 2025-01-22 01:07:24 +0100 CET
  - Author date: 2025-01-22 01:07:24 +0100 CET
commit 2bc04e002927326eeae2edfafb8923dc5ccc0fb57c7e8435d9fd8a3b0bf71f55:
  - tree: efab19e4f70ae75b854b1a293776adcce8f0ccee1e1878d38e2407ebd27db78c
  - Committer date: 2025-01-21 21:35:34 +0100 CET
  - Author date: 2025-01-21 21:35:34 +0100 CET
commit 470a44718e472963fd196e77d842de3146f394df66306f00ffef26dec7fbf4bc:
  - tree: ddac9f8f1a4508b54cd14a2b44b9a1ac2afdb99dc26a57fec06c2fc60fbe72c9
  - Committer date: 2025-01-21 21:14:40 +0100 CET
  - Author date: 2025-01-21 21:14:40 +0100 CET
commit 879c09fd15bc0f2fe82dec12e6477e2832f405990096be2860f726125c3d4df4:
  - tree: efab19e4f70ae75b854b1a293776adcce8f0ccee1e1878d38e2407ebd27db78c
  - Committer date: 2025-01-21 18:15:20 +0100 CET
  - Author date: 2025-01-21 18:15:20 +0100 CET
commit 818b7e5ba4d5e47793924eb69708d898348f6d03615b53b955585c8045d41fa2:
  - tree: 4822279e60ba567b3dfb17745a3a2caf06febe8094ca77f4f918b3805ffc7344
  - Committer date: 2025-01-21 18:14:56 +0100 CET
  - Author date: 2025-01-21 18:14:56 +0100 CET
commit edba53a559d58687593e7303a65d241f4cc5bbc00dfd867070d1c5caf4c2022f:
  - tree: efab19e4f70ae75b854b1a293776adcce8f0ccee1e1878d38e2407ebd27db78c
  - Committer date: 2025-01-21 18:13:03 +0100 CET
  - Author date: 2025-01-21 18:13:03 +0100 CET
commit 40b4eb17ea95628f914bdf36752e659771ed11fc0157b64114d3e7f34a44bf23:
  - tree: 4822279e60ba567b3dfb17745a3a2caf06febe8094ca77f4f918b3805ffc7344
  - Committer date: 2025-01-21 18:12:52 +0100 CET
  - Author date: 2025-01-21 18:12:52 +0100 CET
commit 20a394e1e7d7a4406c5276fd772d42367480f57e8332065e96d112885fa01520:
  - tree: 47617ed796e73001b1bd4d75702379c5d8740f464595efdfc53fd07ab0542f68
  - Committer date: 2025-01-21 17:35:33 +0100 CET
  - Author date: 2025-01-21 17:35:33 +0100 CET

total 11
~/source/nitro/permguard-workspace/repos/playground-cedar main ❯                                   23:01:14
```

AuthZ Model

citizen_doc_submit_actor.cedar

```
@id("can_submit")
permit(
   principal,
   action == Municipality::Document::Action::"can_submit",
   resource == Municipality::Document::"doc"
)
when {
   context.isDocumentOwner == true
};
```

Identity Id	Identity Name	Policies	Trusted Statement
1	Mario Rossi	can_submit, can_delete, can_read	
2	Luca Verdi	can_submit, can_delete, can_read	can_elevate_mario_rossi is_delegated_by_mario_rossi
3	workload-id-ac6a8906		can_elevate

can_elevate_mario_rossi

```
Luca Verdi can elevate to Mario Rossi
```

is_delegated_by_mario_rossi

```
Luca Verdi is delegated by Mario Rossi
```

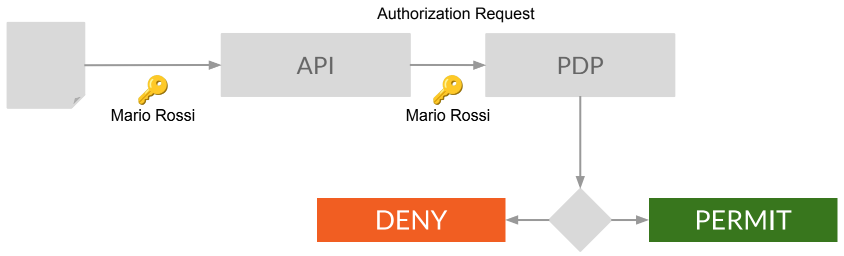
CEDAR

Cedar Policy Language is an Open Source Apache 2.0 Language created by Amazon Web Services.

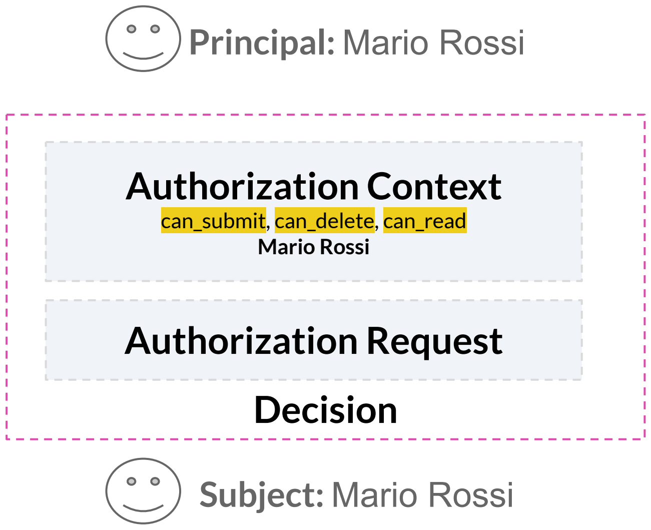
Use Case: Subject

Mario Rossi accesses the municipal website, authenticates, and uploads a document.

```
{
 "principal": {
   "type": "user",
   "id": "mario.rossi@example.com"
 },
 "subject": {
   "type": "user",
   "id": "mario.rossi@example.com"
 },
 "resource": {
   "type": "municipality/document",
   "id": "RSSMRA52A01Z404P"
 },
 "action": {
   "name": "can_submit"
 },
 "context": {}
}
```



Permguard



Auth* Models

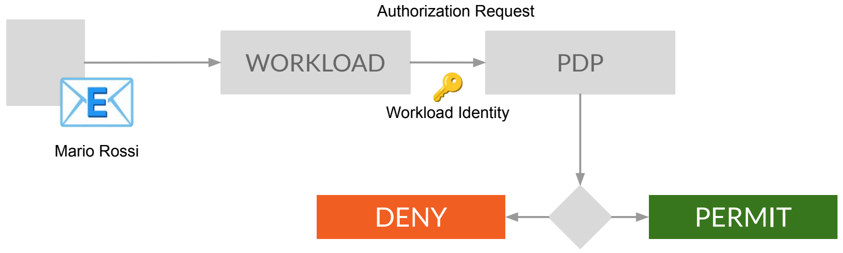
Identity Name	Policies	Trusted Statement
Mario Rossi	can_submit, can_delete, can_read	
Luca Verdi	can_submit, can_delete, can_read	can_elevate_mario_rossi is_delegated_by_mario_rossi
workload-id-ac 6a8906		can_elevate

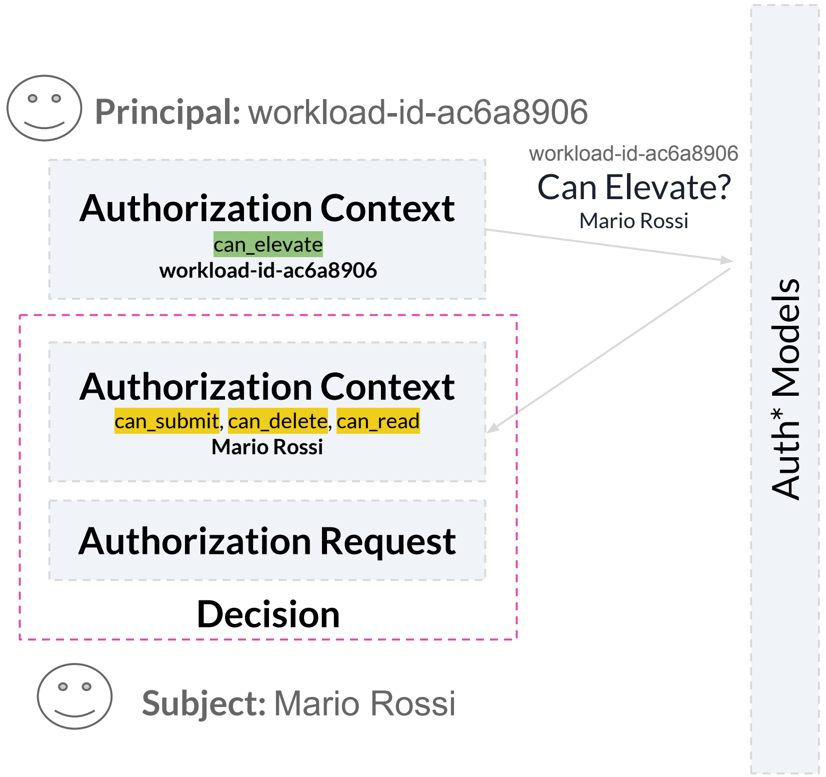
Use Case: Subject + Trusted Elevation

Mario Rossi sends a certified email to a municipality, attaching a document.

Permguard

```
{
 "principal": {
   "type": "user",
   "id": "workload-id-ac6a8906"
 },
 "subject": {
   "type": "user",
   "id": "mario.rossi@example.com"
 },
 "resource": {
   "type": "municipality/document",
   "id": "RSSMRA52A01Z404P"
 },
 "action": {
   "name": "can_submit"
 },
 "context": {}
}
```



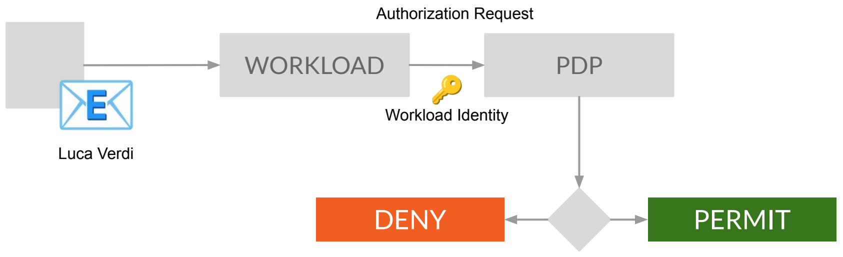


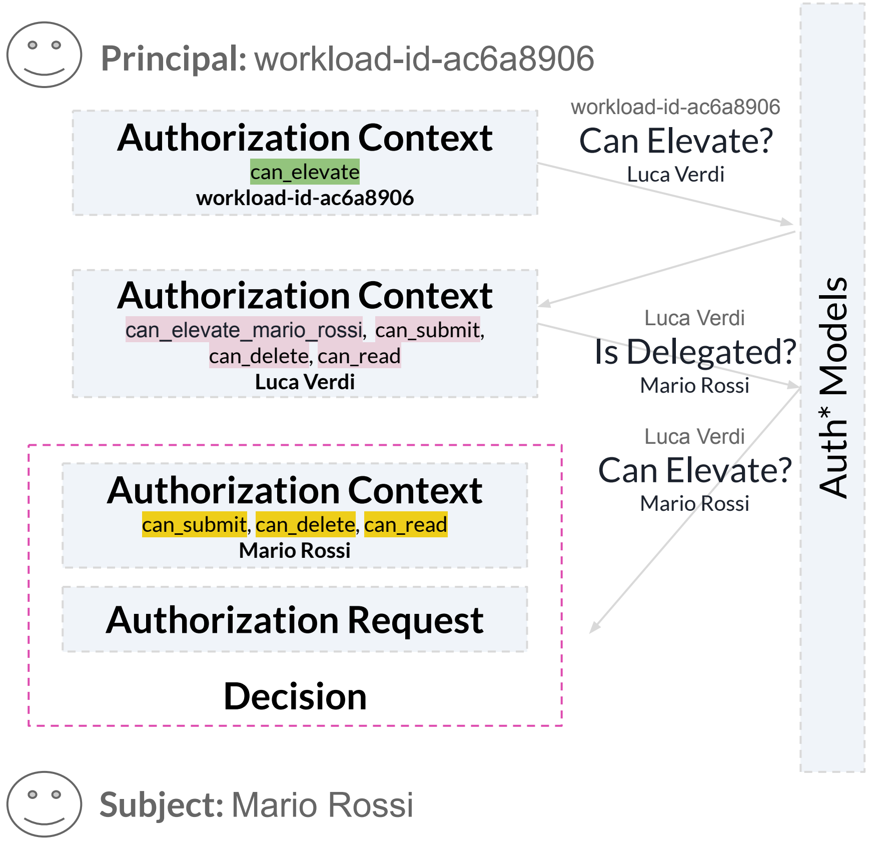
Identity Name	Policies	Trusted Statement
Mario Rossi	can_submit, can_delete, can_read	
Luca Verdi	can_submit, can_delete, can_read	can_elevate_mario_rossi is_delegated_by_mario_rossi
workload-id-ac6a8906		can_elevate

Use Case: Trusted Delegation + Subject

Mario Rossi delegates Luca Verdi to act on his behalf. Luca Verdi then sends a certified email to the municipality, attaching a document on behalf of Mario Rossi.

```
{
 "principal": {
   "type": "user",
   "id": "workload-id-ac6a8906"
   "delegated_type": "user",
   "delegated_id": "luca.verdi@example.com"
 },
 "subject": {
   "type": "user",
   "id": "mario.rossi@example.com"
 },
 "resource": {
   "type": "municipality/document",
   "id": "RSSMRA52A01Z404P"
 },
 "action": {
   "name": "can_submit"
 },
 "context": {}
}
```





Identity Name	Policies	Trusted Statement
Mario Rossi	can_submit, can_delete, can_read	
Luca Verdi	can_submit, can_delete, can_read	can_elevate_mario_rossi is_delegated_by_mario_rossi
workload-id-ac6a8906		can_elevate

Confused Deputy Problem

⚠️

The Confused Deputy Problem happens when a trusted entity is tricked into misusing its privileges to act on behalf of an attacker.

Authorization Context

can_submit, can_delete, can_read

Identity Actor Model

There are two types of Role Based Actor:

NEW

- **Role-Based Actor:** A Role-Based Actor represents a predefined role with a limited, task-specific set of permissions. It adheres to the principle of least privilege by loading only the permissions required for the task at hand.
  - Example: An citizen-doc-submit-actor allows submitting documents but does not grant permissions to delete or read them.

NEW

- **Digital Twin Actor:** A Digital Twin Actor replicates all permissions of the specific Principal. While this can be necessary for scenarios requiring full mirroring of the Principal, it may lead to excessive permissions being granted, potentially violating the principle of least privilege.
  - Example: A mario-rossi-actor mirrors Mario Rossi's identity, granting him permissions to submit, delete, and read documents

Key considerations:

- Security: Elevating to a Role-Based Actor minimizes security risks by restricting permissions to those required for the specific task. Elevating to a Digital Twin Actor, on the other hand, may expose the system to greater risks by unnecessarily loading excessive permissions.
- Best Practices: Use Role-Based Actors whenever possible to enforce minimal privilege. Reserve Digital Twin Actors for scenarios where full mirroring of the Principal is explicitly required.

AuthZ Model

Actor Id	Actor Model	Actor Name	Policies
1	role-based	citizen-doc-submit-actor	can_submit_actor
2	role-based	citizen-doc-delete-actor	can_delete_actor
3	role-based	citizen-doc-read-actor	can_read_actor
4	digital-twin	mario-rossi-actor	can_submit_actor, can_delete_actor, can_read_actor

citizen_doc_submit_actor.cedar

```
@id("can_submit_actor")
permit(
   principal,
   action == Municipality::Document::Action::"can_submit",
   resource == Municipality::Document::"doc"
)
when {
   context.isDocumentOwner == true
};
```

citizen_doc_delete_actor.cedar

```
@id("can_doc_delete_actor")
permit(
 principal,
   action == Municipality::Document::Action::"can_delete",
   resource == Municipality::Document::"doc"
)
when {
   context.isDocumentOwner == true
};
```

CEDAR

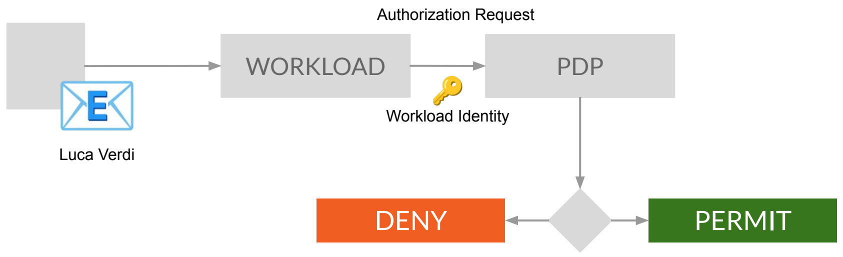
Cedar Policy Language is an Open Source Apache 2.0 Language created by Amazon Web Services.

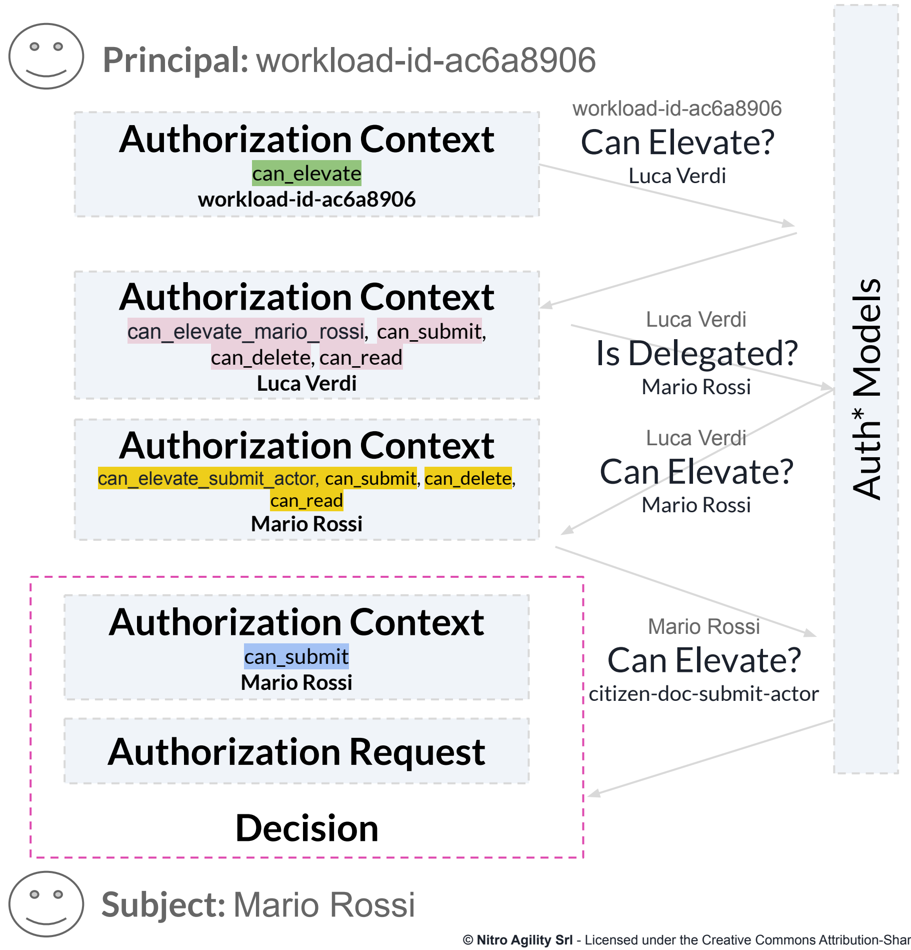
Identity Actor to Address the Confused Deputy Problem

By using a Role-Based Actor, it is possible to narrow down the permission scope and prevent the Confused Deputy Problem.

Permguard

```
{
 "principal": {
   "type": "user",
   "id": "workload-id-ac6a8906"
   "delegated_type": "user",
   "delegated_id": "luca.verdi@example.com"
   "target_type": "user",
   "target_id": "mario.rossi@example.com"
 },
 "subject": {
   "type": "actor",
   "id": "citizen_doc_submitter_actor"
 },
 "resource": {
   "type": "municipality/document",
   "id": "RSSMRA52A01Z404P"
 },
 "action": {
   "name": "can_submit"
 },
 "context": {}
}
```

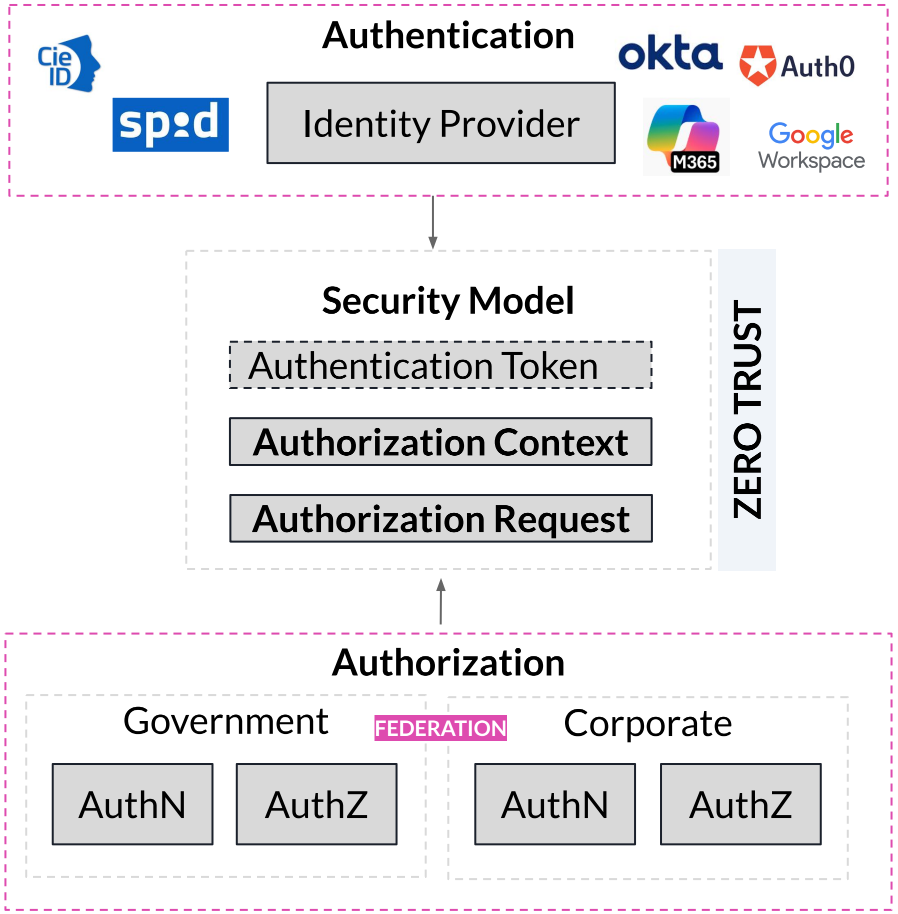




Identity Name / Actor Name	**Policies**	**Trusted Statement**
Mario Rossi	can_submit, can_delete, can_read	can_elevate_submit_actor
Luca Verdi	can_submit, can_delete, can_read	can_elevate_mario_rossi is_delegated_by_mario_rossi
workload-id-ac6a8906		can_elevate
citizen-doc-submit-actor	can_submit	

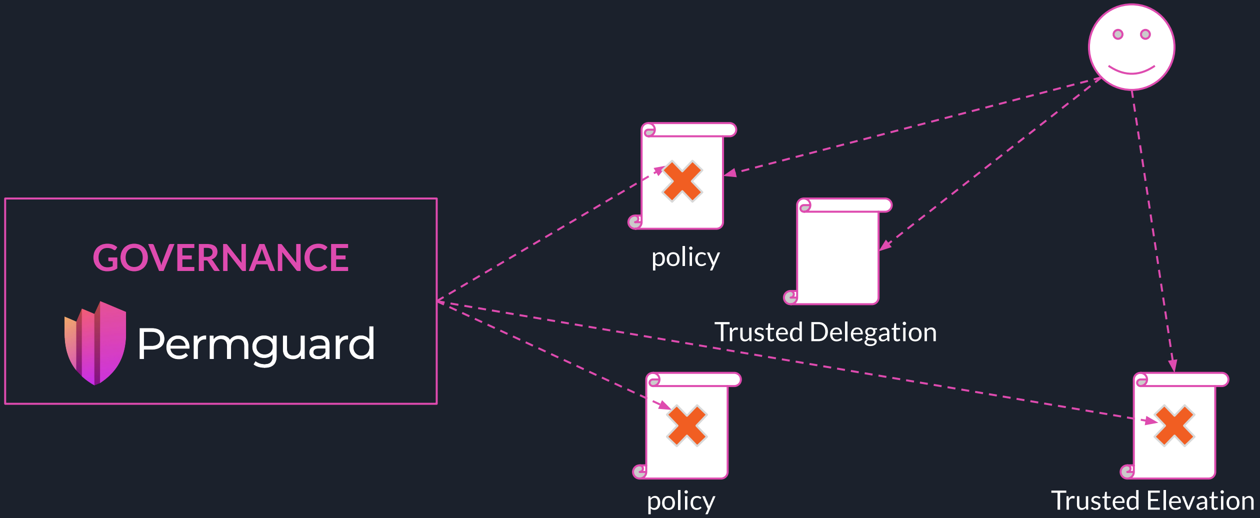
ZTAuth* unlocks **complex federation** capabilities while maintaining **centralized governance**.

Permguard



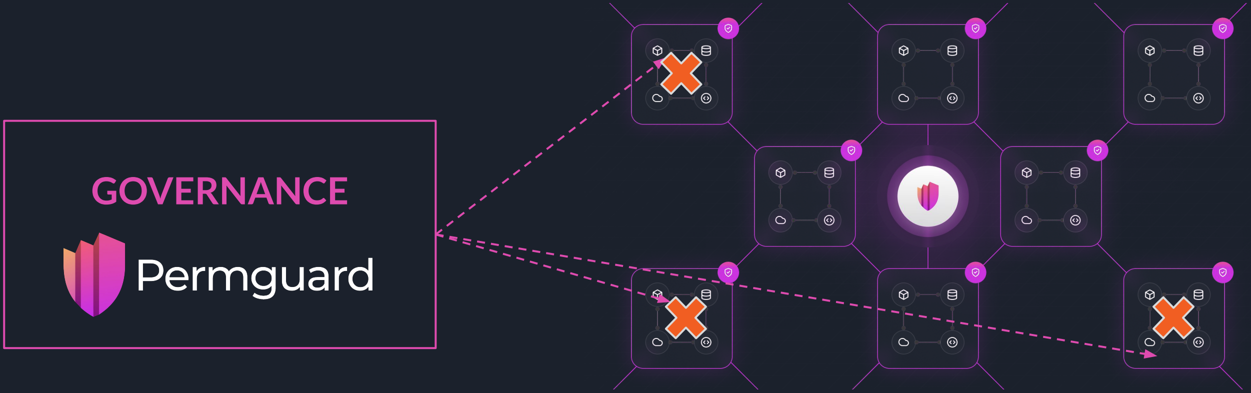
Centralized Governance

ZTAuth* relies on Policies and Trusted Statements (Elevation and Delegation), enabling centralized governance that can be enforced consistently across all applications.



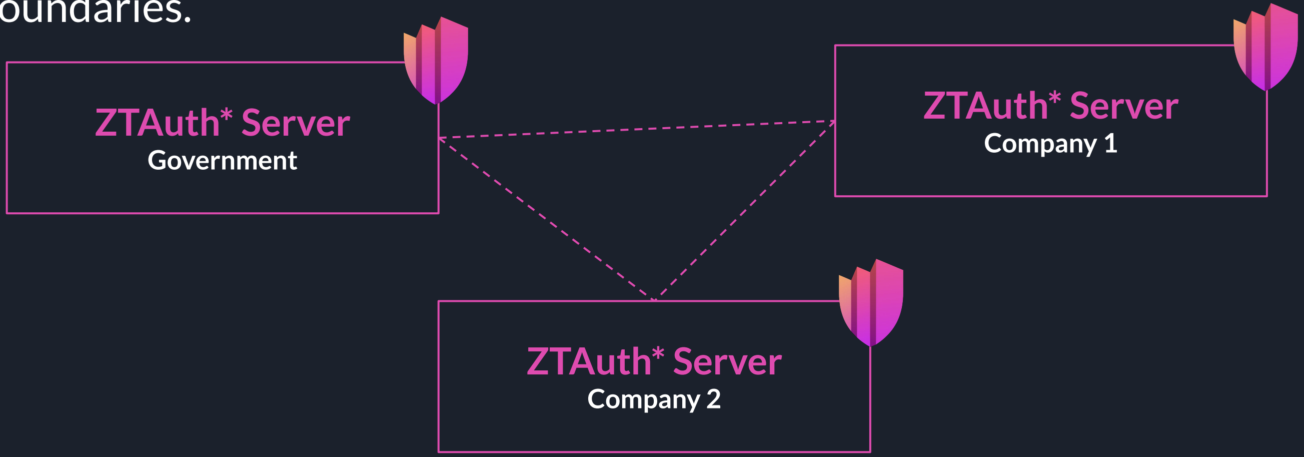
Centralized Governance

ZTAuth* allows enabling and disabling Trusted Statements (Elevation and Delegation), enabling centralized governance over both workloads and nodes within the network.



Federation

Trusted Federation refers to the secure integration of multiple Central Servers across federated environments. This is achieved by the exchange of public keys between Central Servers, enabling them to verify and establish trust relationships beyond their individual boundaries.

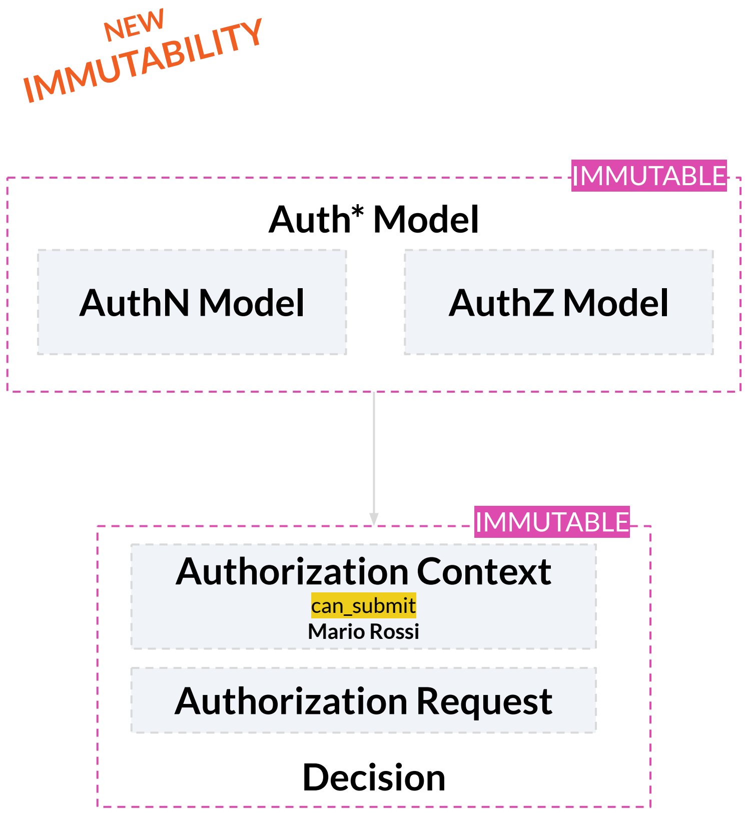


Speaker

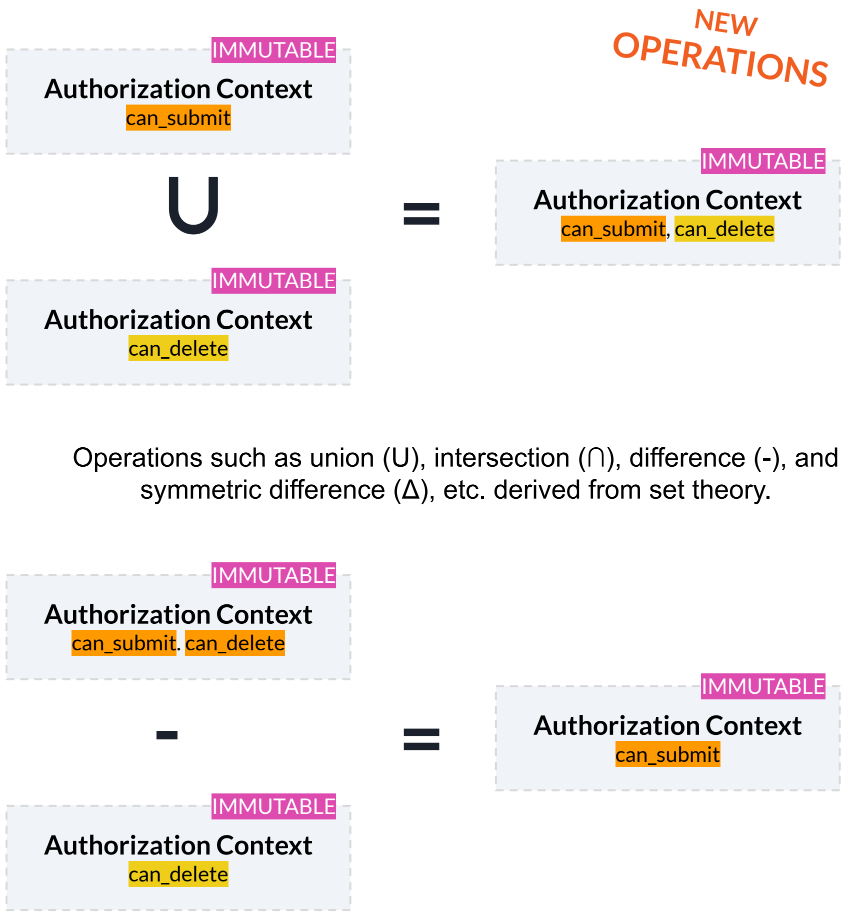
Antonio Radesca
Co-founder at Nitro Agility S.r.l.

NitroAgility Permguard

With **ZTAuth***, decisions are made by **elevating** to the appropriate **Authorization Context**. Each Authorization Context is **isolated**, and most importantly, the key principle is **immutability**.

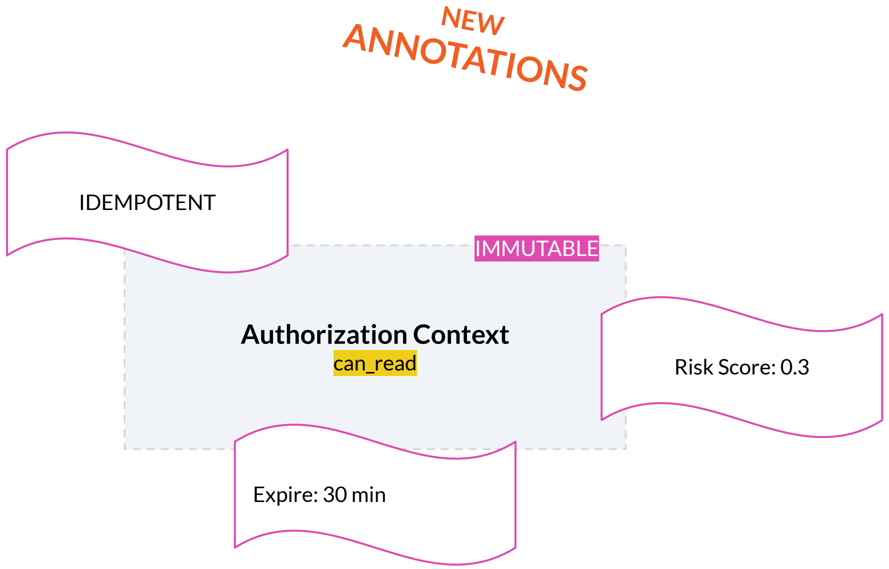


Immutability means that an **Authorization Context** cannot be altered. Instead, **ZTAuth*** enables **Set Operations** to **create** new **Authorization Contexts** as an alternative to modification.

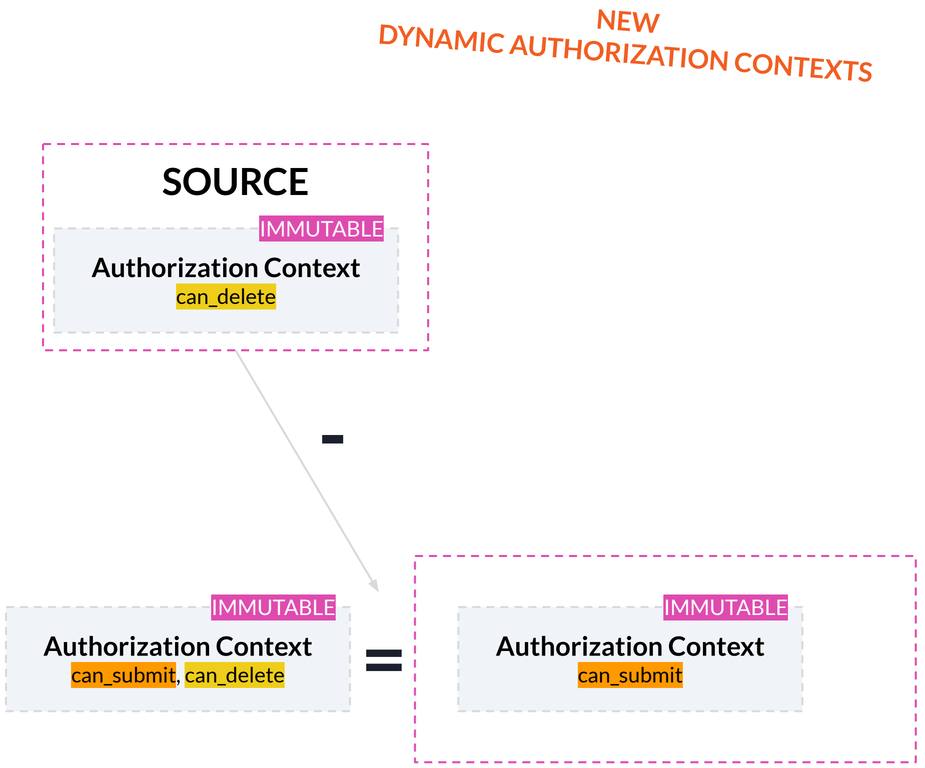


Operations such as union (U), intersection (∩), difference (-), and symmetric difference (Δ), etc. derived from set theory.

Another important **property** is **annotation support**. It is possible to annotate an **Authorization Context** with labels, which can have multiple meanings (e.g., **risk scores**, expiration times, etc.).

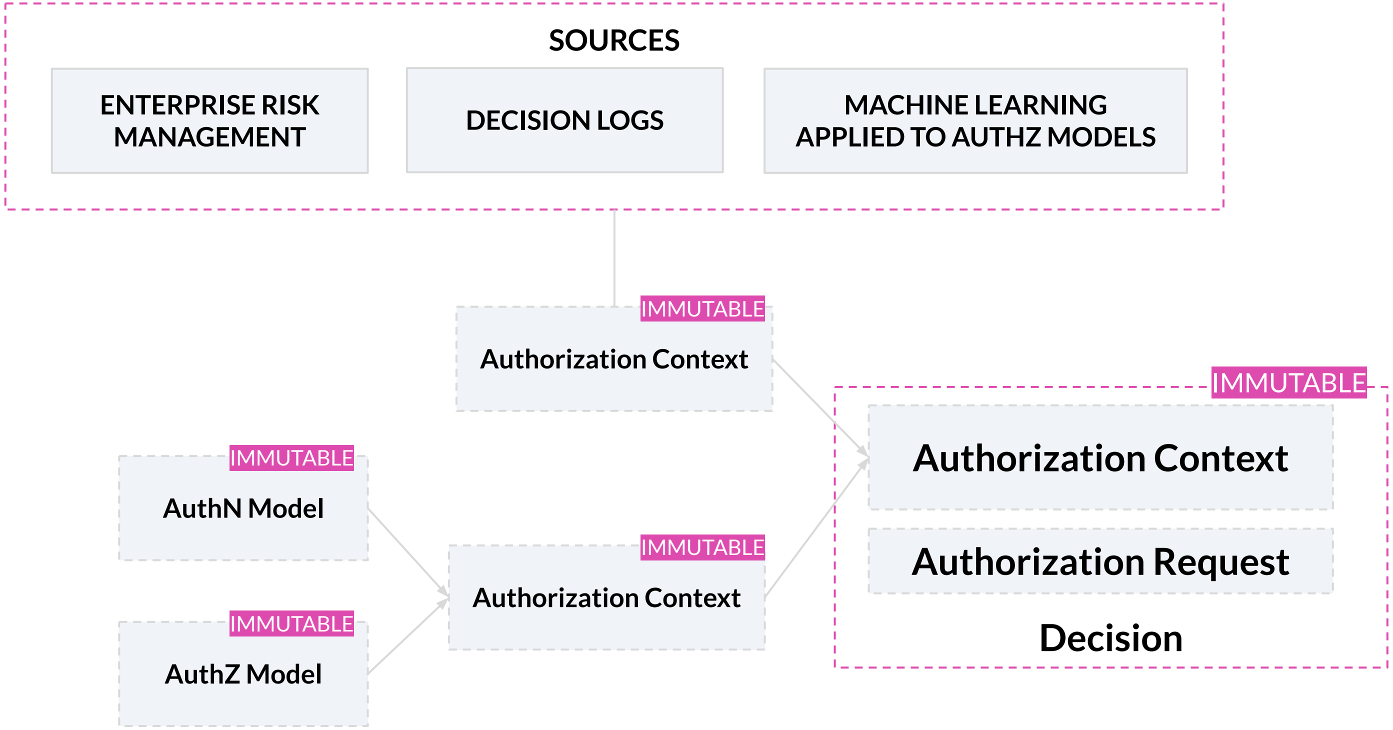


Those **ZTAuth*** principles unlock a **new paradigm** where application models can be **dynamically updated** by external sources.



Auth* Models & Authorization Contexts

ZTAuth* enables the integration of **external sources** that can provide **intents** to **modify** the **Authorization Context**. For example, in **Risk Management**, an external system could **dynamically adjust permissions based** on a **detected high**-**risk** activity, such as an unusual login location or abnormal transaction patterns.


SOURCES
ENTERPRISE RISK MANAGEMENT
DECISION LOGS
MACHINE LEARNING APPLIED TO AUTHZ MODELS
IMMUTABLE
Authorization Context
IMMUTABLE
AuthN Model
IMMUTABLE
AuthZ Model
IMMUTABLE
Authorization Context
IMMUTABLE
Authorization Context
Authorization Request
Decision

Research

Machine Learning applied to AuthZ Models

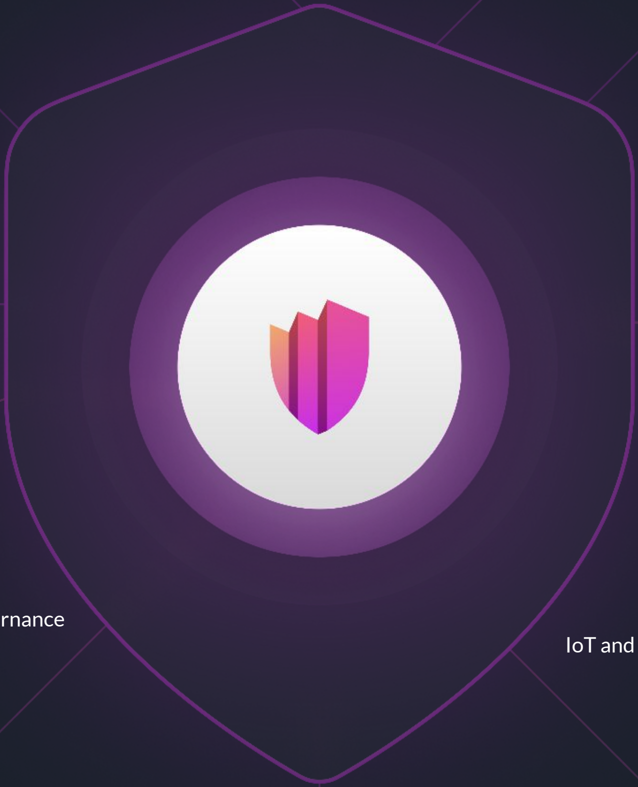
Risks

- Vulnerable Policy Risk
- Policy Impact Risk
    - This is after processed by ML to extract Global Impact Risk that could affect a set of policies
    - Example of Policy Impact Risks are: Reputation, Revenue, Functional, etc

Policy Classification

- **Problem**: Ensure that authorization policies are correct and secure.
- **Solution**: Use classification models to analyze policies and identify potentially risky or non-compliant ones. A model can be trained to classify policies as "safe" or "risky" based on parameters such as complexity, granted permissions, and usage context.
- Already available for CEDAR

What is Next



- AI Agent Security
- Decentralized Access Control
- Zero Trust Extended Framework
- Trusted Delegation for the CIE/SPID
- Governance
- IoT and Edge Computing

What is Next

The **ZTAuth*** effort aims to **explore** ways to **evolve** and move towards **standardization**.

Permguard

If you want to help us with this specification, feel free to get in touch with us at

opensource@nitroagility.com

Thank you!

Permguard

x.com/permguard

github.com/permguard

www.permguard.com

hello@permguard.com

youtube.com/@permguard