LF DECENTRALIZED TRUST

03 March 2026

Trusted AI Agents by Design: *From Trust Ecosystems to Authority Continuity*

PIC

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

Permguard (www.permguard.com)

Co-Chair, Trusted AI Agents Working Group
Decentralized Identity Foundation (DIF)

NitroAgility

Permguard

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

TOC

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

What This Talk Covers

Identity solves identification, not authority.
The unsolved problem: how authority propagates through execution.

This presentation looks at the problem from a different point of view:

- no artifacts as core security element
- no authority granted by proof of possession
- a **new security primitive** is needed

It is about intent and execution.

Authority(t) exists iff
execution preserves origin p_0
$\wedge$ exercised authority is origin-bounded
$\wedge$ execution satisfies all continuity guardrails

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Grounded in Formal Methods

$\pi = \langle \alpha_0, \alpha_1, \ldots, \alpha_i \rangle$, where $\alpha_i = (p_0 , ops_i , e_{i-1} , e_i , T_i , C_i)$

$Authority(i) = \{ o \in ops_0 \mid Continuity(o, \alpha_i) \}$

$\forall$ o :
$o \notin ops_0 \Rightarrow \forall i : o \notin Authority(i)$

$\nexists$ i, o such that:
$o \notin ops_0 \wedge o \in Authority(i)$

Continuity(o, α_i) holds iff
origin p_0 is preserved
$\wedge\ ops_i \subseteq ops_0$
$\wedge\ \forall g \in Guardrails : g(o, \alpha_i)$ holds

where Guardrails include executor, temporal, contextual, and any future execution dimensions

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PART I

The Problem

AI Agents are exposing existing problems.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

The Trigger: AI Agents

AI Agents didn't create new security problems.
They removed the assumptions hiding the old ones.

No perimeter

No stable topology

No single owner

No shared config trust

What breaks under AI was already broken.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

The Real Problem

We solved security with:

- Protocols
- Posture
- Configuration
- Policy

But we never solved authority propagation.

> *How can I trust either the posture, configuration, policies of something.*
> *I do not own, do not control, and do not even know?*

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Old Problems are Back

Confused Deputy

A vulnerability in which a service misattributes authority and executes operations using its own authority on behalf of a less-privileged caller.

Privilege Escalation

Authority grows beyond what was originally granted.

Ambient Authority

Services act with all their authority on every request.

Token Substitution

An artifact is replayed or reused outside its intended context.

Not bugs. Structural consequences of the model.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

AI Agents are Workloads

Key mistake:

Treating AI Agents as a new security subject.
They are not. They are workloads.

Workloads are:

- Replicated
- Rescheduled
- Short-lived
- Re-keyed

Identifiers are unstable. Authority must not be.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PART II

Fixing the Mental Model

An Opinionated Mental Model for a New Ontology.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Three Structural Elements

Identity

Represents a subject that can express intent

MUTABLE

Human · Organization
Workload · Agent

Intent

The desired action

MUTABLE

Intent definitions can evolve. Each expressed instance is immutable.

"I want this action"
"I accept responsibility"

Delegation

Enables a third party to use authority on behalf of the delegator

DYNAMIC

On behalf of the delegator
Derived from original intent.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

The Fourth Structural Element

AUTHORITY

Identity

anchors responsibility (origin p_0)

Intent

expresses will and responsibility (ops_0)

Authority

authority instance (p_0, ops_0)

🧬 Authority originates from an identity, must travel across computation and decision points, is continuous, cannot expand, never increase.

MONOTONIC

Traceable to origin
Cannot expand — ops_0 at creation, $ops_{(i+1)} \subseteq ops_{(i)}$

GOVERNANCE

Delegation

Enables a third party to use authority on behalf of the delegator

DYNAMIC

On behalf of the delegator
Derived from original intent.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Understanding the Real Problem

Without the right mental model,
the underlying problem remains invisible.

This is not about possession.
It’s about intent and execution.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Setting the Terms of the Journey

Principal: Entity that can express intent, human or non-human. Defined by intent, not credentials.

Delegation: One principal grants specific intents to another. Acts on behalf of, not as. Outside the authority instance.

Intent: Verifiable expression of will. Defines authorized operations and responsibility. Each instance is immutable. Transforms identity into authority.

Authority: Identity + intent. Carries: who (p_0), what (ops), bound characteristics. Can only shrink, never expand. Not a token, a property of execution from origin.

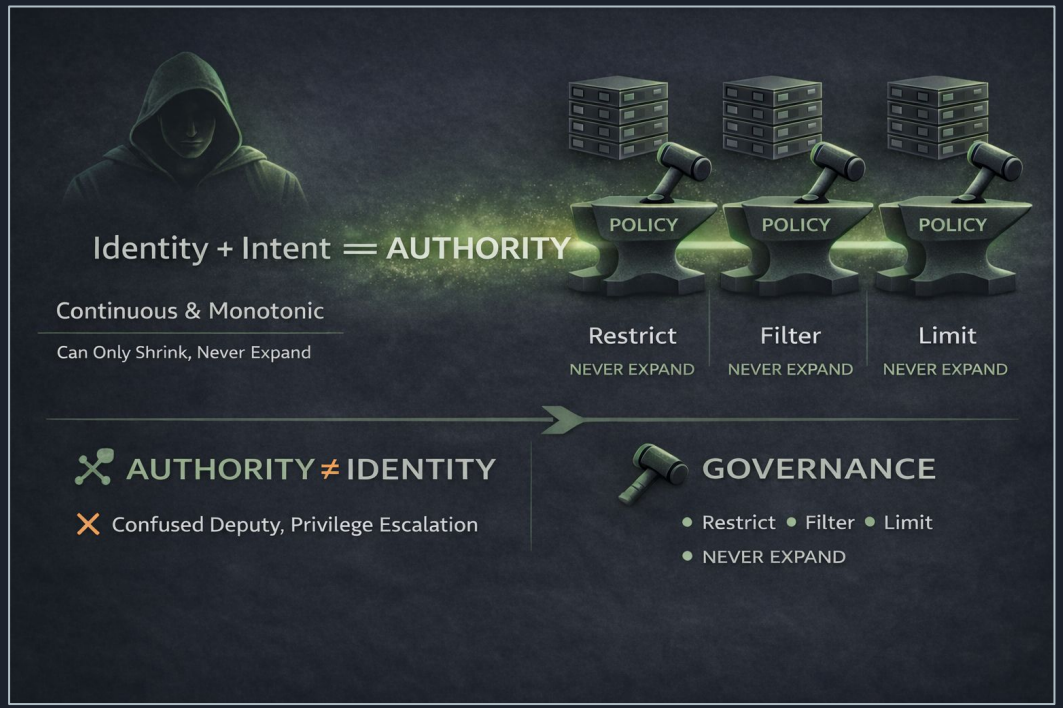
Workload: Executor that continues under received authority (proving relationship) or creates new authority as a new origin. Authority never expands.

Governance: Can stop, restrict, or leave unchanged the authority. Never expand. If a policy expands authority, that is privilege escalation by definition.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Authority & Governance

They are both important, they are different and there is a monotonic relationship between them.



Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PART III

State of the Art and why it fails

The way things are done today.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

The Hidden Assumption

In today's systems:

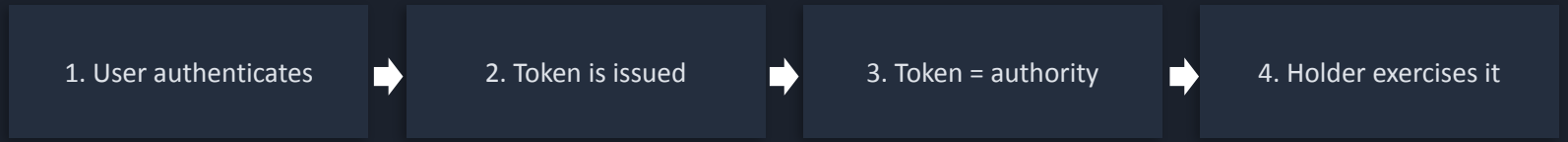
> Authority is treated as an object
> that can be created, stored, transferred, and consumed.

This is the root of the problem.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Authority by Possession

OAuth Access Token — the canonical example:



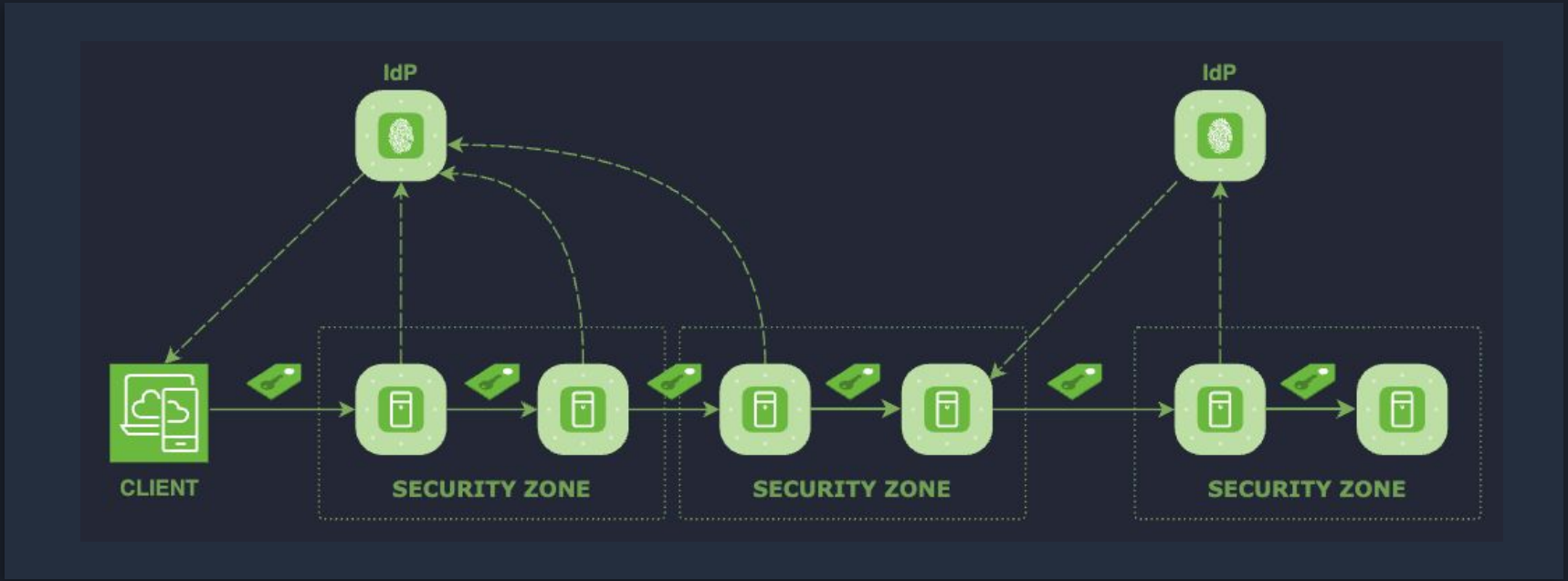
This implies:

- Steal token ⟶ get authority
- Replay token ⟶ get authority
- Use token in unintended context ⟶ get authority

Authority == Possession

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

RFC 8693 OAuth 2.0 Token Exchange



Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Token Exchange Breaks Down

OAuth Token Exchange

- Token A exchanged for Token B
- Same intent (supposedly)
- New authority artifact
- Nothing enforces continuity

Now Insert Async

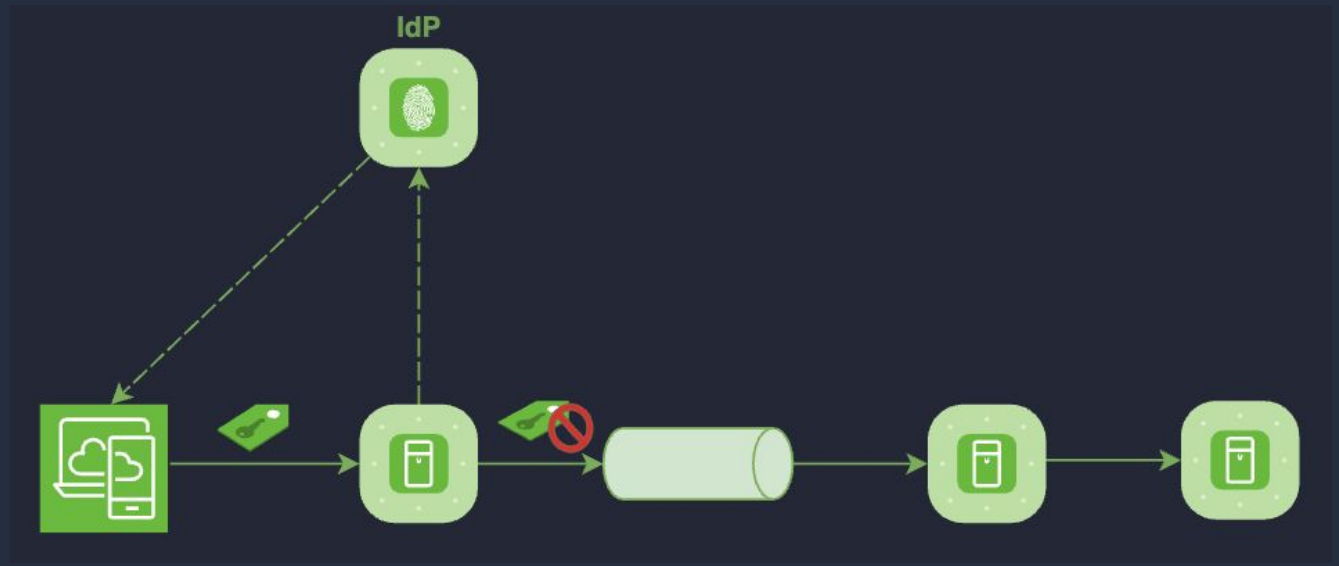
The delegator doesn't know the delegate ahead of time.

- ⚠ How do we pass the token?
- ⚠ Encrypt it? Rotate it?
- ⚠ Replay after expiry?
- ⚠ Security collapses.

Any async workflow, queue, or broker exposes the same structural failure.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

OAuth 2.0 Token Exchange & Async Flows



Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Service Accounts

The industry workaround:

Caller Authority
(Bob)

+

Service Account
Authority

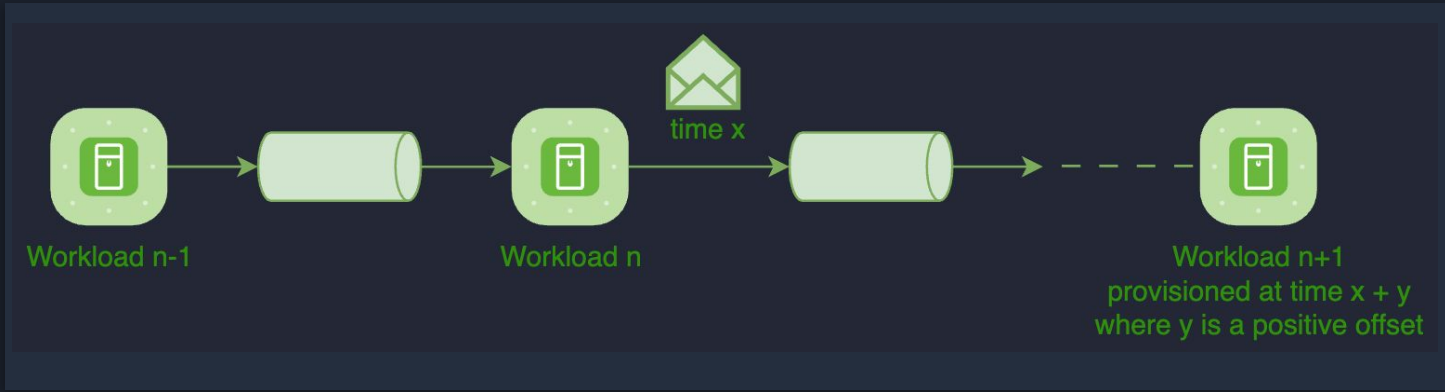
→

Confused Deputy — Guaranteed

Not a bug.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Canonical Execution Model



Execution is temporal and causal, not positional.

- Workload n+1 provisioned at time x + y (y > 0)
- Authority flows from origin — never re-created
- Every hop must prove continuity, not possession

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Configuration Will Not Save You

- Scopes
- Audiences
- Token binding (DPoP)
- Policies

cannot fix a broken ontology.

> *They reduce the attack surface. They never remove the formulability of the attack.*

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Scenario A: Agent Expands Under Own Authority

Alice asks her AI agent: “Summarize the file **/system/logs/audit.txt**”.

Alice cannot read that file. But the AI agent can — it has its own service authority over /system/logs for internal diagnostics.

The agent receives Alice’s request. It opens **/system/logs/audit.txt** using its own service credentials and returns the content to Alice in the summary.

Alice directly designated a resource she has no access to. The agent executed the operation using its own authority — on Alice’s behalf. No malformed input. No exploit. Alice simply asked, and the deputy complied.

This is the confused deputy. The agent had authority over the resource. Alice did not. The agent could not distinguish between “what Alice is allowed to request” and “what I am able to do.“

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Scenario A: Agent Expands Under Own Authority

The Agent receives a request under the caller's authority, but decides to act under its own — creating new authority via its own identity and intent

The same applies to workload (for instance any microservice)

The Agent does not continue under the delegated authority: it creates its own new authority for new resources, through its own identity and its own intent.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PART IV

The Paradigm Shift

From possession to authority continuity.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

The Shift

A subject creates authority by expressing intent.

After that:

> **Authority must only be continued, never recreated.**

Humans (OIDC, VC) **Non-humans** (SPIFFE, DID/VC)

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

A New Primitive is Required

To make this shift possible,
we need a new primitive.

Tokens encode possession.
We need a primitive that encodes intent and execution.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Proof of Continuity (PoC)

To continue authority, a workload must prove:

Causal Continuity

Execution is a valid continuation of its predecessor.

Origin Invariance

The origin (p_0) is immutable throughout the chain.

Monotonic Authority Restriction

Authority is monotonically non-increasing. $ops_{i+1} \subseteq ops_i$ — authority never expands.

Proof of Continuity (PoC) is built from **Proof of Relationship (PoP)**.
Proof of Possession is one mechanism, not the primitive.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Execution is powered by continuity.

No continuity. No execution.

A workload may proceed only if it either:

- proves an authority continuity relationship, or
- explicitly creates new authority

Proof of Relationship is what validates continuity.

Possession is just one way to prove continuity, not the principle.

No authority mixing. No re-materialization.

This is Authority Continuity.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Identity Is Not Required to Continue

To Continue Authority

Identity: optional

Identifiers: optional

What matters: proof of continuity

To Create Authority

Express own intent

Become a new origin

Identity: required

Continuation ≠ Creation

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Structural Elimination

Detect

Monitor for confused deputy at runtime

Traditional approach

Mitigate

Add scopes, bindings, policies to reduce risk

Current best practice

Eliminate

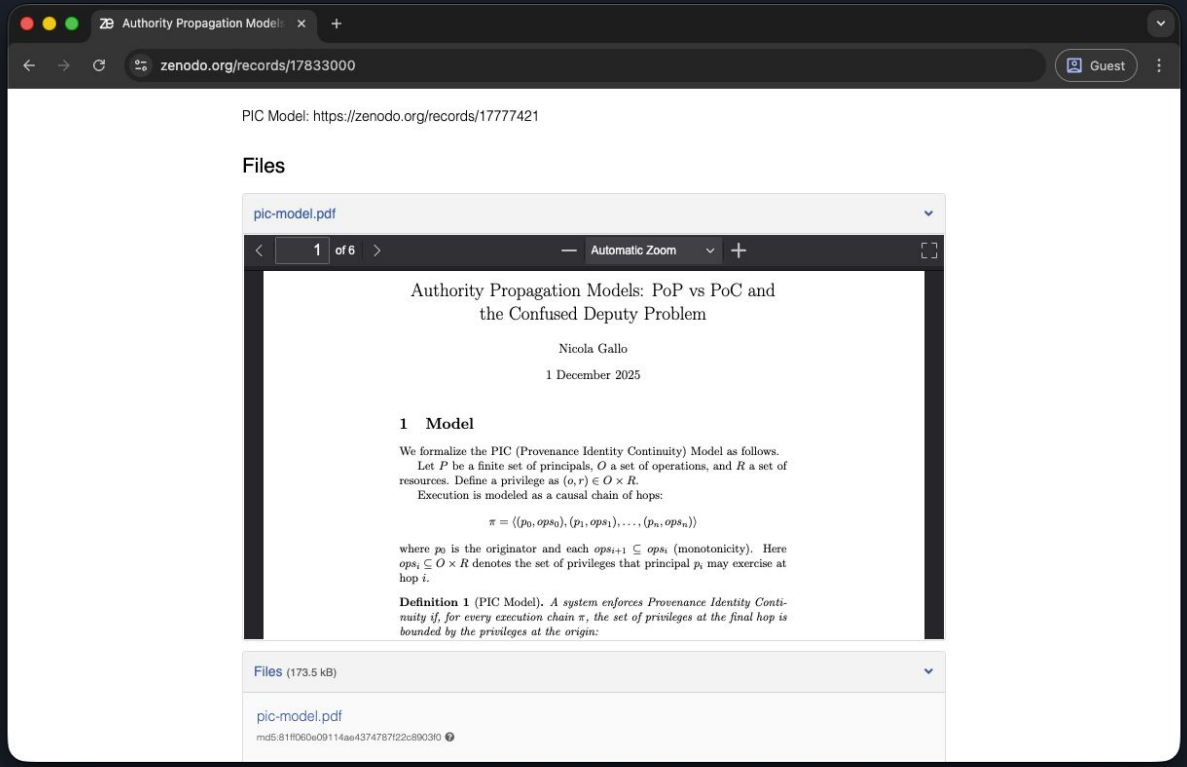
Make the attack structurally inexpressible

PIC

The Confused Deputy becomes non-formulable. Not prevented — inexpressible.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

The formal Proof



Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Scenario A: Under PIC the confused deputy is eliminated

Under PIC: the agent operates under Alice's authority

Authority is bound to Alice's origin: **ops_0 = {read: /users/alice/}**.

Alice designates **/system/logs/audit.txt**.

The agent attempts to read it.

The PIC model rejects: **{read: /system/logs/}** is not a subset of **{read: /users/alice/*}**.

The operation is blocked. It does not matter that the agent has its own credentials for that path. The agent's own authority never enters Alice's chain. The deputy cannot be confused — it can only act within the authority of the origin.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PART V

The PIC Model

Formal elimination of the confused deputy.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Provenance Identity Continuity (PIC) Model

Provenance

The causal chain is always traceable and auditable. From Origin to end, unbroken. If it breaks, it stops.

Identity

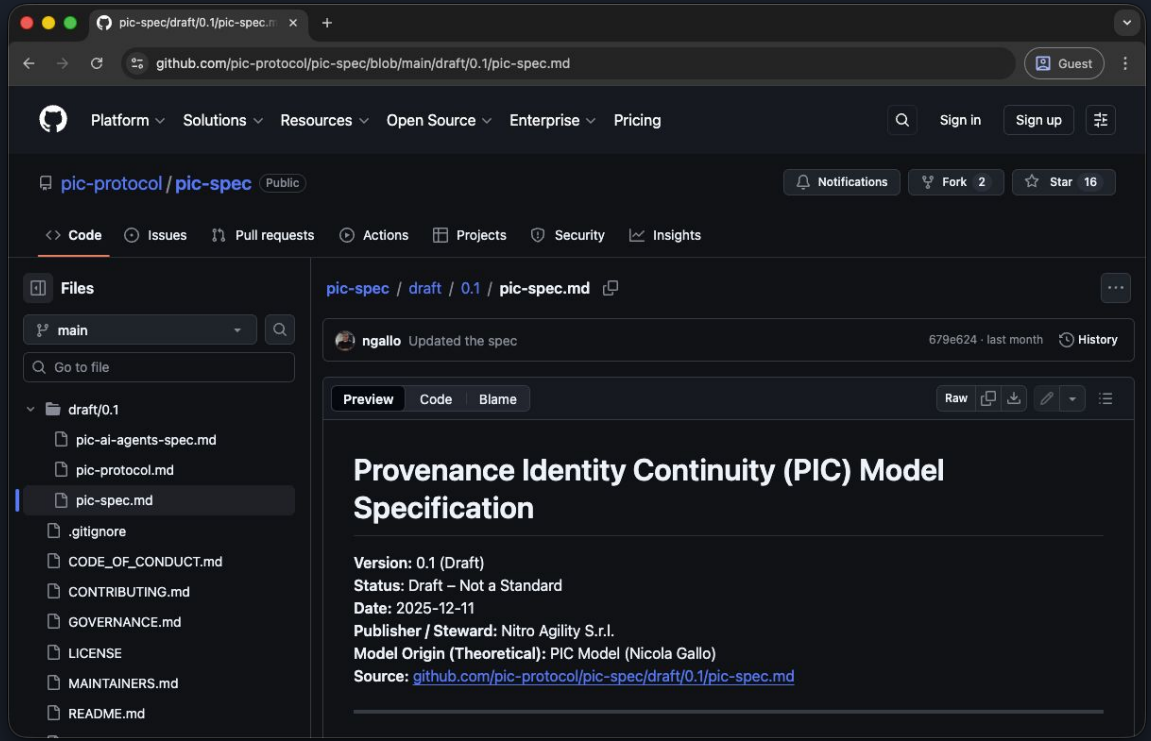
The origin is immutable. It generates authority. It cannot change throughout the chain.

Continuity

Continuity is proven at every step. Proof of Continuity replaces Proof of Possession. Authority can only shrink.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PIC Specification



Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PIC and the Trust Plane

Causal Authority Transition (CAT) is the enforcement mechanism that validates PIC invariants.

PIC Elements

Execution Hop

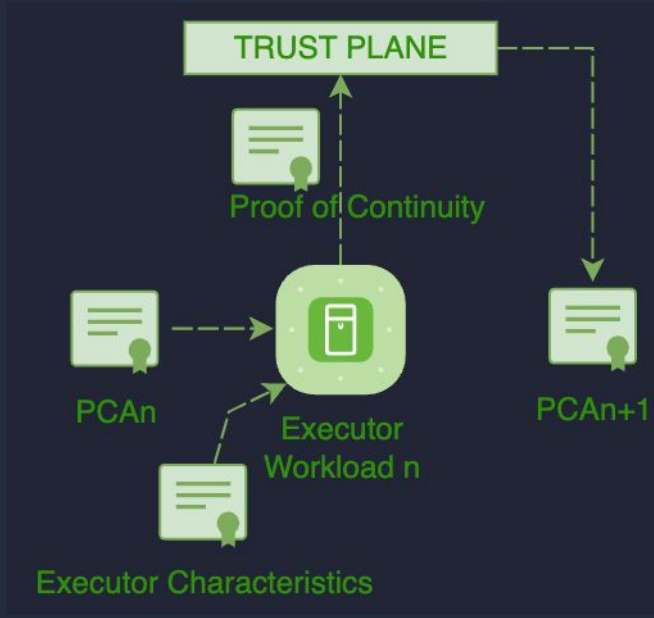
Executor

Executor Characteristic

PIC Causal Authority (PCA)

Causal Authority Transition (CAT) = Trust Plane

Alice's origin: ops_0 = {read: /users/alice/}
context = Sales department



Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PIC vs OAuth: Network Latency Profile

Authority lives in the data, not in the pipe. The transport is irrelevant. Adapters make it real.

Same Network Round-Trips

PIC with Trust Plane requires the same network hops as OAuth token exchange

Both need a round-trip to a trust authority on the network

Network latency is equivalent — the wire cost is identical

What Changes: Compute, Not Network

The difference is in PCA validation and minting time (algorithmic cost)

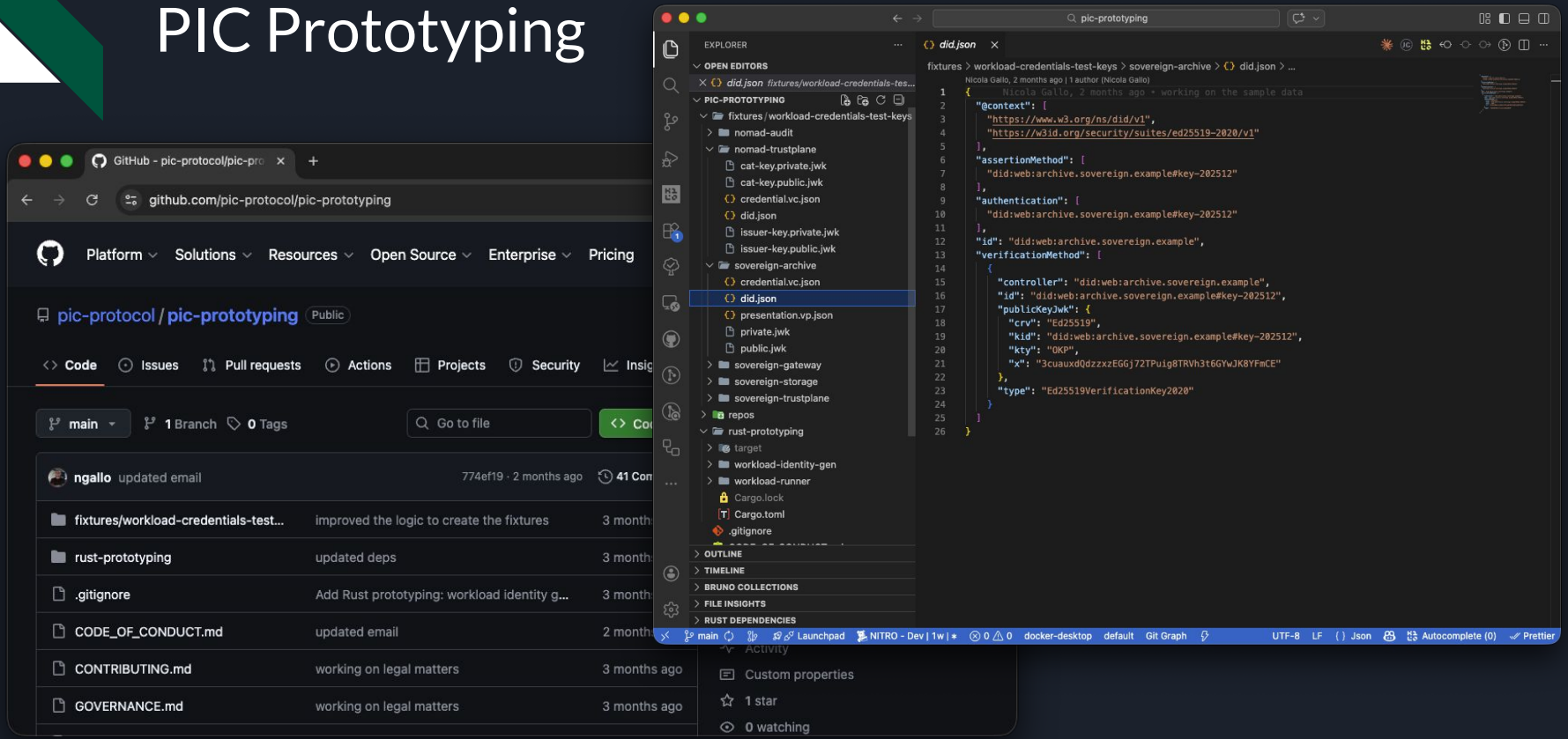
Cryptographic verification replaces token introspection

No token re-materialization, authority continues natively

Refactoring-Invariant Security + Trust Planes unlock decentralized validation patterns — pushing compute to the edge and dramatically reducing end-to-end overhead.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PIC Prototyping



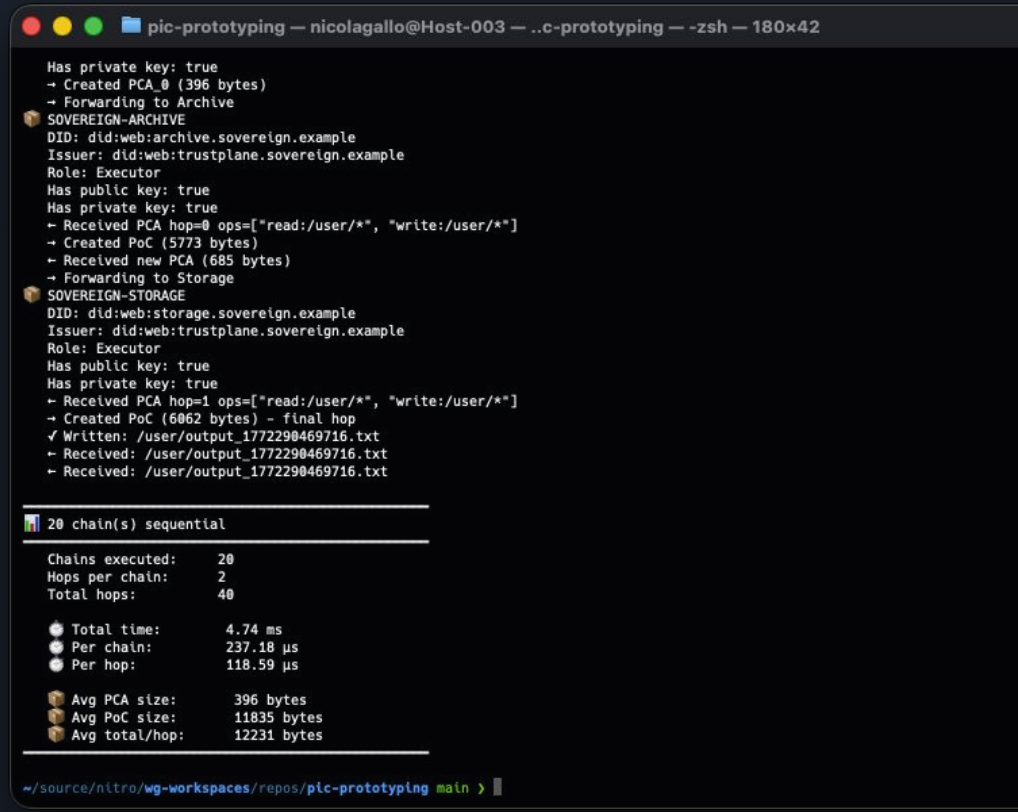
Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PIC Prototyping

PIC is light: the PCA doesn't carry the entire chain, only the last link.

Verification is fast because each hop only needs to validate its immediate predecessor, not the full history.

The chain remains verifiable end-to-end, but the runtime cost stays constant per hop.



Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Provenance Identity Continuity (PIC) Model

Keep in mind that

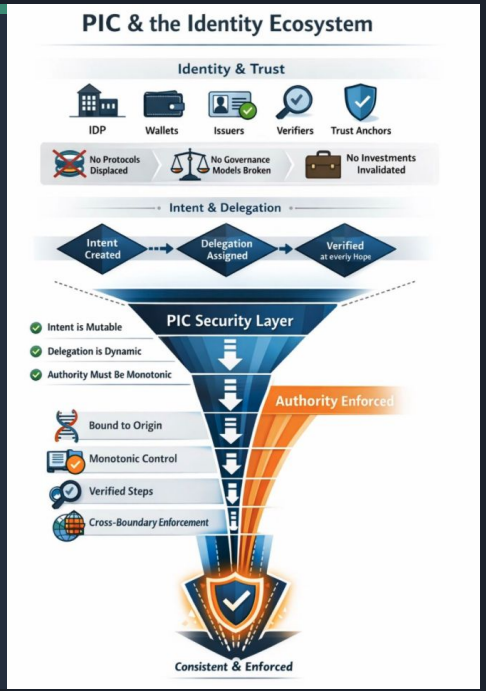
🪪 Intent and delegation continue to be expressed through the same identity infrastructure: IdPs, wallets, policies, consent flows, organizational decisions.

⚙️ PIC does not interfere with that process.

🧩 PIC takes the result of those decisions and turns authority into a structural property of the system.

🧠 PIC is not a new identity protocol ⟶ 🛡️ It is a new security model.

IMPORTANT: Intent is Mutable (as definition). Each instance is immutable.



Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PART VI

What is next?

The adoption path.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PIC is Channel & Transport Agnostic

Authority lives in the data, not in the pipe. The transport is irrelevant. Adapters make it real.

The Guarantee

Authority bound to origin, not to transport

Proof of Continuity works over any pipe

No channel dependency. No confused deputy.

Adapters for Adoption

OAuth / OIDC: custom claims

TSP: Trust Spanning Protocol

DIDComm: decentralized messaging

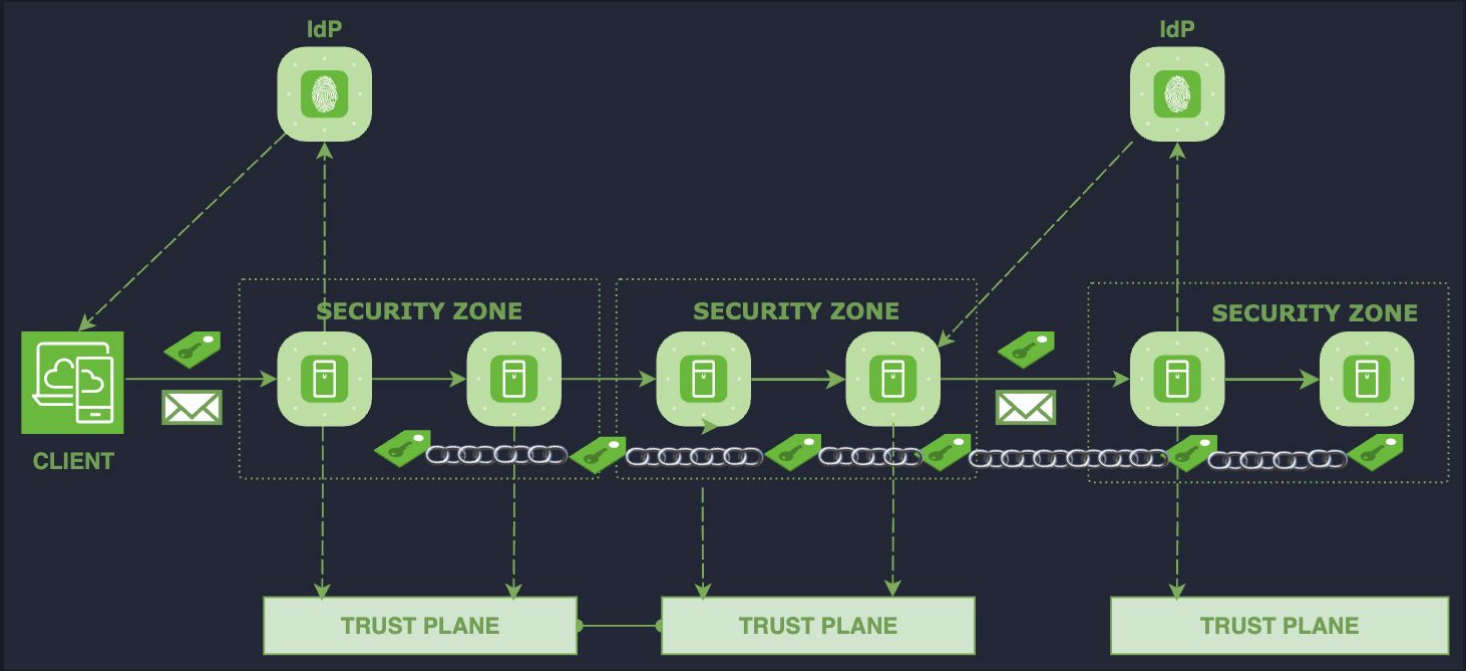
SPIFFE / mTLS: workload identity

PIC Model → **Adapter Layer** → **Any Transport**

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

OAuth as a PIC Federation Backbone

OAuth as the federation backbone: we encode the PCA into the OAuth access token as a custom claim, ensuring full compatibility with existing infrastructure and applications. OAuth gets you through the gate. PIC guarantees what happens after.



Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Crossing the OAuth Perimeter

Once past the gate, full PIC with Trust Planes. Authority is created, continued, and verified natively. No tokens, no re-materialization, no confused deputy.

PIC Enables

- Federated Trust Planes
- Partitioned security domains
- Zero-trust by construction
- Refactoring-invariant security

No Forklift Rewrite

- Integrates with OAuth / OIDC
- Works with SPIFFE / DID
- Federate with existing IdPs
- Incremental adoption path

Transitional Path

1. Federate IdP with Trust Plane → 2. Exchange intent for PCA → 3. Embed PCA as custom claim → 4. Enter perimeter → PIC-native

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

TSP + PIC: Complementary Layers

Trust Spanning Protocol provides the secure channel. PIC provides the authority guarantee.

Trust Spanning Protocol

Layer 2 — Secure Channel

- Establishes cryptographically verifiable trust relationships between endpoints
- Authenticity and integrity guaranteed; confidentiality optional
- Identifier-agnostic
- Transport-independent
- Analogous to what IP does for data, TSP does for trust

PIC Model

Authority Continuity — Security Model

Authority bound to origin, monotonically restricted at every hop

Proof of Continuity replaces Proof of Possession

Confused deputy structurally eliminated

Guarantees what flows through the pipe.

Together: TSP delivers the trusted channel. PIC ensures authority never expands, origin never changes, and the confused deputy is inexpressible. The channel is secure. What flows through it is guaranteed.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Authority is not identity.

Authority is not possession.

Authority is continuity.

PIC is not just a protocol.
It is a correction of the model.
It is a new security primitive.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Get Involved

Website

www.pic-protocol.org

Specification

github.com/pic-protocol/pic-spec

Formal Model

Zenodo (DOI: 10.5281/zenodo.17833000)

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

PIC

Not a Policy.
Not a Token.
An *Ontological Shift*
Proven Mathematically.

Thank you!

www.pic-protocol.org

JOIN THE PIC SLACK CHANNEL

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.

Delegation — Acting on Behalf of the Delegator

The Delegator grants specific intents (not necessarily all) to the Delegatee, who creates authority as if they were the Delegator

Delegator

Holds authority

Selects SPECIFIC intents to delegate.

Not all intents must be delegated.

Delegation

Selective grant

Grants chosen intents to the Delegatee.

Delegatee receives scoped permissions.

Delegated Intent

Granted by Delegator

Delegatee uses the granted intent (not its own).

Acts as a representative of the delegator.

Authority Created

On behalf of Delegator

New resource authority is created on behalf of the Delegator.

Traceability maintained.

The Delegatee uses an intent granted by the Delegator to instantiate authority on behalf of the delegator — acting as its representative, not as the delegator itself.

Nitro Agility S.r.l. — Licensed under CC BY 4.0 International. Attribution to Nitro Agility S.r.l. and PIC (www.pic-protocol.org) is required.