ATTEST-R: A Standardized Framework for Sovereign Release Authorization

ENTERPRISE AUTHORIZATION CONTROL PLANE

(Draft positioned for IEEE-style standardization and technical whitepaper submission)

EXECUTIVE SUMMARY

Modern software delivery systems have become highly effective at automating execution, but they remain fundamentally weak at governing authority. In most CI/CD pipelines today, the same platforms that build and deploy software are implicitly trusted to approve, authorize, and audit those deployments. This coupling of execution and authority is sufficient for speed-oriented environments, but it fails the assurance requirements of government, defense, banking, and other regulated domains where **who approved a release is as critical as what was released**.

Existing DevSecOps controls—policy engines, artifact signing, and audit logs—verify integrity and process adherence, but they do not provide sovereign, non-repudiable authorization. Approval remains implicit, infrastructure-bound, and ultimately overrideable by privileged operators.

This paper introduces **ATTEST-R**, a standardized framework for cryptographically enforced release authorization that decouples execution from approval. ATTEST-R establishes an independent authorization control plane, ensuring that production deployments are impossible without explicit, policy-compliant, cryptographically verifiable human approval recorded in an immutable system of record.

ATTEST-R does not replace CI/CD systems.

It governs them.

---

1. Introduction and Problem Statement

1.1 The Hidden Risk in Modern CI/CD

In high-security and regulated environments: - A production deployment is a high-impact event - Insider threats are explicitly part of the threat model - Auditors require provable change control and accountability

Yet in most organizations: - Approval logic lives inside CI/CD tools - CI administrators can bypass or alter approval workflows - Audit logs are mutable by privileged operators

This creates a fundamental weakness: **the executor is also the authority.**

1.2 Why Existing Controls Fall Short

Control	Limitation
CI/CD approvals	Enforced by the same system that deploys
Change tickets	Procedural, not cryptographically bound
Artifact signing	Proves integrity, not authorization
SIEM / WORM logs	Detective, not preventive

What is missing is an **independent, enforceable authorization authority.**

1.3 Why Traditional DevSecOps Is Necessary — but Insufficient (v1.1)

Modern DevSecOps practices have significantly improved software supply-chain integrity. Tools such as policy engines (e.g., Kubernetes admission controllers), artifact signing frameworks (e.g., Notary and cloud-native signers), and centralized audit logging provide important safeguards against accidental misconfiguration and unauthorized changes.

However, these controls share a common assumption: **the infrastructure itself is trusted to act as the authority.**

---

2. Design Goals and Principles

ATTEST-R is built on five non-negotiable principles:

1. Separation of Authority and Execution
2. Non-Repudiation by Cryptographic Signature
3. Policy-Driven, Multi-Party Approval
4. Minimal On-Ledger Data (Hashes Only)
5. Enterprise-Grade Identity & Governance Alignment

---

3. Reference Architecture and Conceptual Model

Figure 2: ATTEST-R Reference Architecture (Separated Authority and Execution)

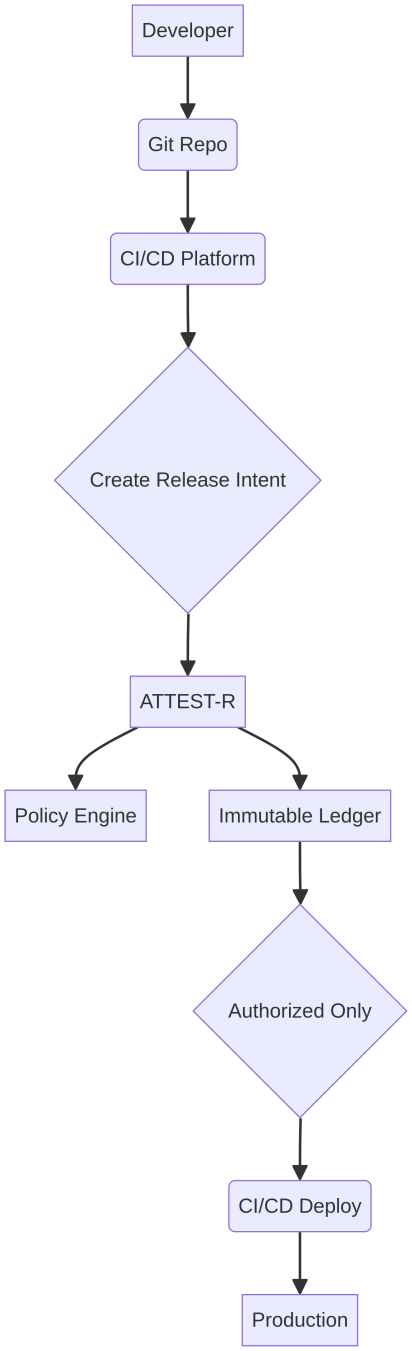


Figure 2 presents the ATTEST-R reference architecture, explicitly separating release authorization from execution. CI/CD systems are hard-gated on authorization outcomes recorded in an immutable system of record.

3.1 Release Intent

A **Release Intent** is a cryptographic declaration created by CI/CD prior to deployment. It includes: - Artifact digest (immutable) - Source commit reference - Target environment - Approval policy - Expiry window

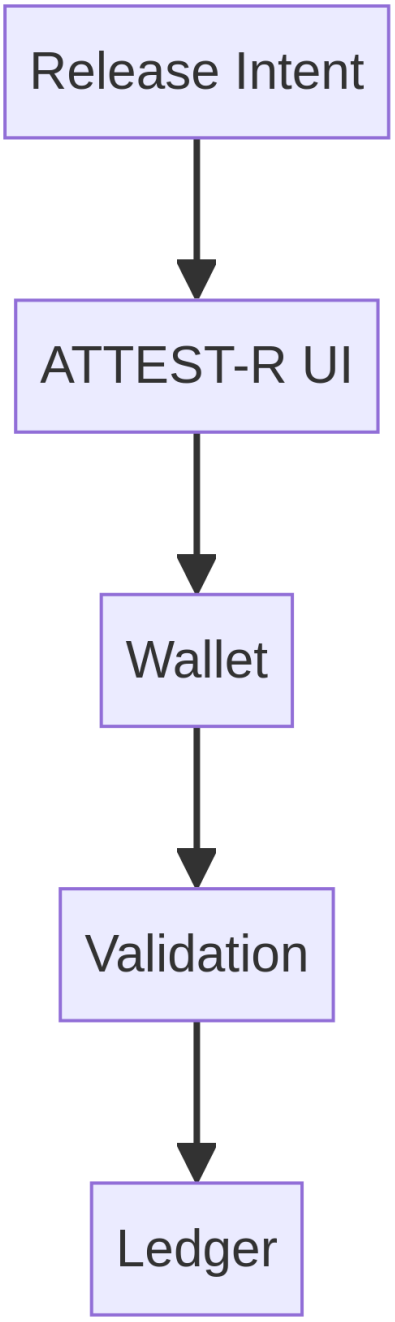
A release cannot proceed unless its intent is authorized.

3.2 Approval as Cryptographic Attestation

Approval in ATTEST-R is not a UI click. It is a cryptographic signature from an authorized identity over a Release Intent.

Each approval is: - Identity-bound - Time-bound - Policy-scoped - Non-repudiable

Figure 3: Wallet-Based Cryptographic Approval Flow



---

4. Identity, Cryptographic Identity Containers, and Wallet Model

4.1 Wallets as Enterprise Identity Primitives (v1.1 Clarification)

In ATTEST-R, a wallet is not a financial construct. It is a cryptographic identity container used to hold signing keys that the CI/CD platform cannot impersonate.

Wallets are: - Bound to enterprise identity via SSO / OIDC - Backed by HSMs or secure signers in production - Browser-based only for proof-OCs

Their sole purpose is to ensure that **no system, administrator, or automation pipeline can approve a release on behalf of a human authority.**

Example mapping: `alice@bank.com → wallet 0xA1B2...` `role: SECURITY_APPROVER`

---

5. Authority Tokenization Model (Non-Financial)

5.1 Authority Tokens (Soulbound)

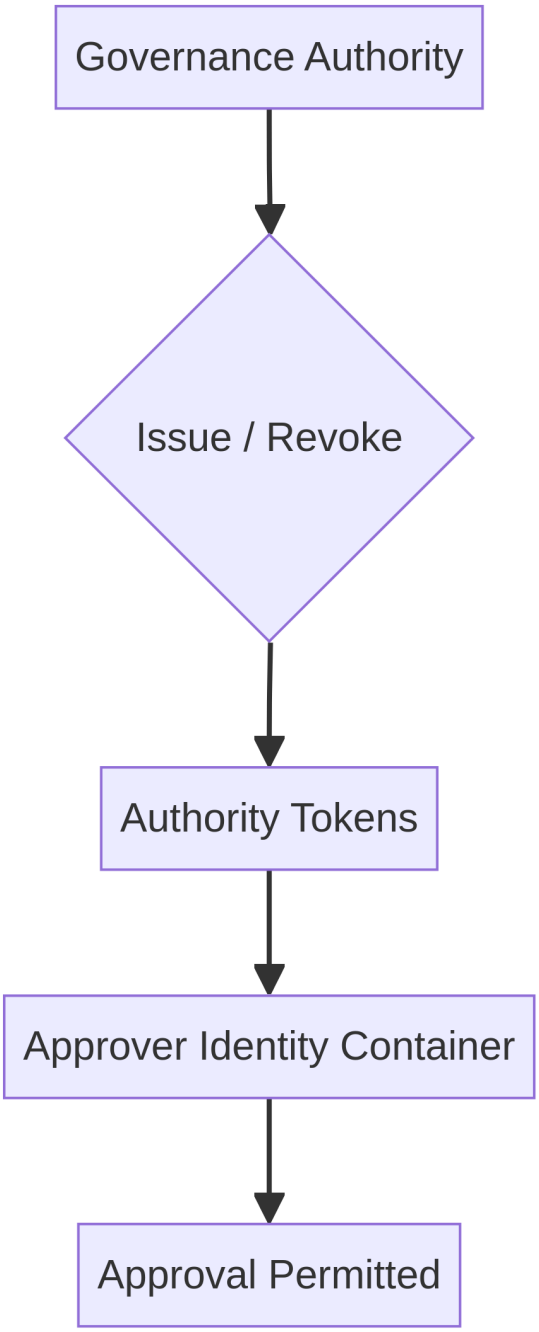
ATTEST-R uses non-transferable authority tokens to represent mandate and role.

Examples: - PROD_APPROVER - SECURITY_APPROVER - INFRA_PROVISIONER - BREAK_GLASS

These tokens: - Are issued by governance authorities - Cannot be transferred - Can be revoked or time-limited

They map directly to enterprise RBAC and segregation-of-duties models.

Figure 4: Authority Token Governance Lifecycle



5.2 Explicitly Out of Scope

ATTEST-R does not: - Incentivize approvals - Reward approvers - Require token payments - Introduce speculative economics

---

6. End-to-End Authorization and Execution Flow

1. CI builds artifact and computes digest
2. CI creates Release Intent
3. Approvers sign intent
4. ATTEST-R validates identity, authority, and policy
5. Release becomes authorized
6. CI/CD deploy job is triggered
7. Execution proof is recorded

Any deviation results in hard failure.

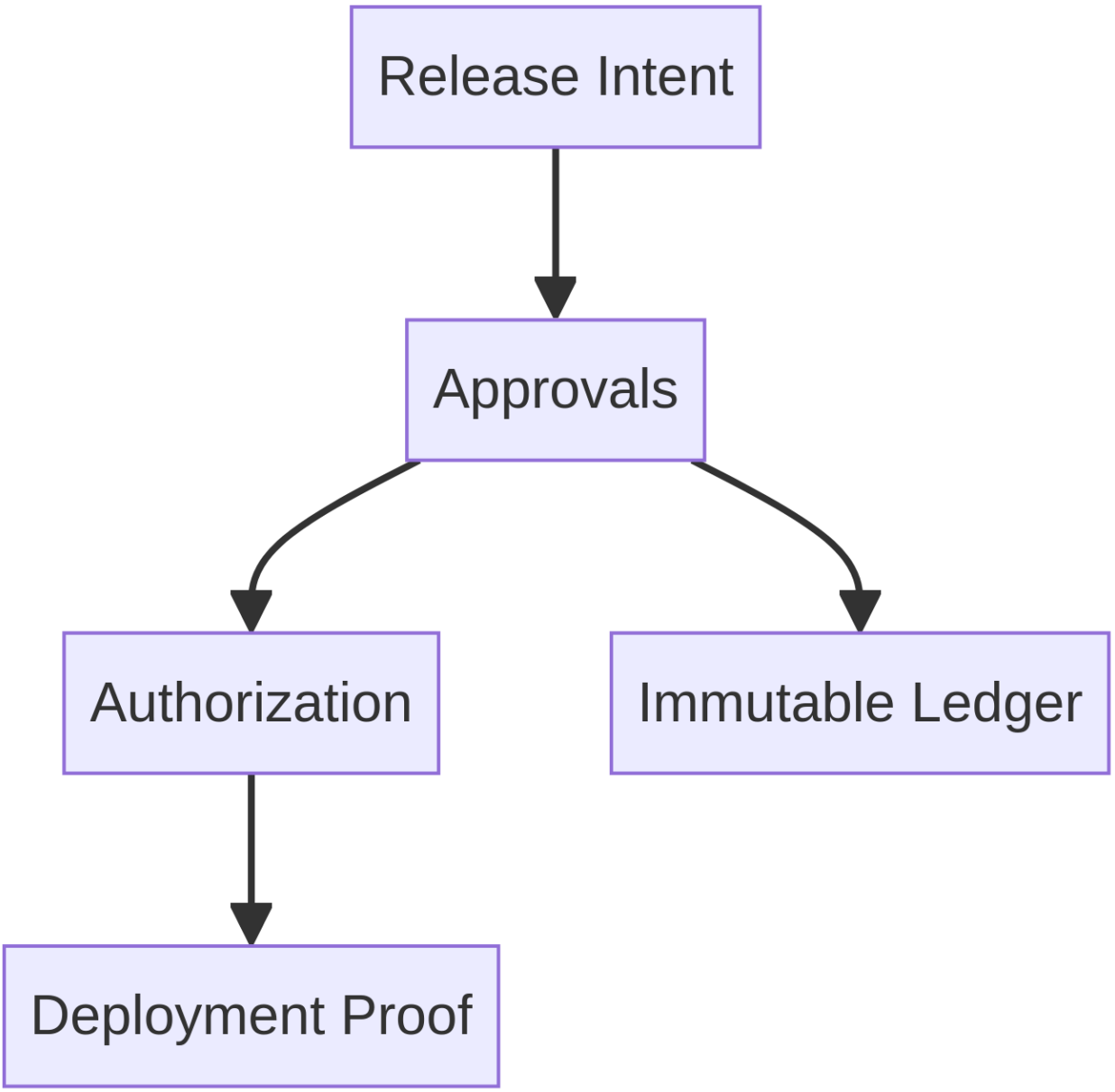
---

7. Auditability, Compliance, and Assurance Properties

ATTEST-R produces: - Immutable approval evidence - Cryptographic proof of authorization - Deterministic linkage between artifact and deployment

Auditors can independently verify authorization without trusting CI/CD logs alone.

Figure 5: Immutable Release Audit Chain



---

8. Deployment and Trust Models

- **Government / Defense**: Permissioned ledger, enclave-controlled
- **Banking / Fintech**: Consortium or private ledger
- **Proof-of-Concept**: Local EVM chain

Only hashes and signatures are recorded on-ledger.

---

9. Differentiation and Unique Contributions

ATTEST-R introduces a release-authorization control plane that existing systems do not address: - CI/CD focuses on execution automation - GRC focuses on documentation - Supply-chain tools focus on integrity

None enforce sovereign authorization.

9.1 Unique Selling Propositions

1. **Sovereign release authority**
2. **Non-repudiable approvals**
3. **Hard enforcement**
4. **Authority tokenization**
5. **Independent auditability**

No widely adopted solution today provides all five simultaneously.

---

10. Conclusion and Future Standardization Considerations

ATTEST-R demonstrates that release authorization can be treated as a first-class governance and security problem, independent of execution. By formalizing release intent, cryptographic approval, and authority tokenization, ATTEST-R establishes a foundation for verifiable, tamper-resistant software release governance.

Future work includes: - Formal standardization - Alignment with SLSA, NIST SP 800-series, ISO/IEC 27001 - Interoperability profiles - Standardized audit exports

ATTEST-R is proposed as a candidate framework for standardization within IEEE, ISO, or NIST-aligned working groups.

Proof, not trust.

ATTEST-R creates authorization records that survive tool changes, vendor changes, and organizational churn.