Principia Trust Engine: A Framework for Compliant On-Chain Debt Capital Markets

GLEIF Hackathon 2025: Final Technical Report
Date: October 2025

1. Executive Summary

This report details the architecture and implementation of the Principia Trust Engine prototype, a foundational compliance framework for institutional-grade digital assets on the Cardano blockchain. The project's objective was to design and build a functional solution to the "Corporate Authority Gap" in tokenised assets, where the identity and authority of on-chain actors cannot be cryptographically verified.

The core achievement of this project is a successful, live demonstration of a Verifiable Smart Contract on the Cardano testnet. We have implemented an end-to-end "Flow of Verifiable Trust" by integrating a GLEIF v-LEI Credential Chain with a novel on-chain enforcement mechanism. Our architecture is a direct, practical implementation of the concepts outlined in the "v-LEI on-chain: Verifiable Smart Contracts" report, specifically demonstrating Phase 1: Smart Contract Provenance Attribution and on-chain authority verification.

The central technical innovation is the "VC-in-Redeemer" model, a highly scalable and private method for on-chain compliance. This model passes Verifiable Credentials directly into a transaction's redeemer, allowing the smart contract to perform real-time, trustless verification of identity and authority. This report provides a comprehensive overview of the system's architecture, the deep mechanics of the on-chain verification process, the key achievements of the hackathon, and the strategic next steps.

2. The Principia Trust Engine: Architectural Design

The Trust Engine's architecture is composed of three distinct layers, ensuring a clear separation of concerns between off-chain credentialing, user interaction, and on-chain enforcement. This layered model enhances modularity, scalability, and security.

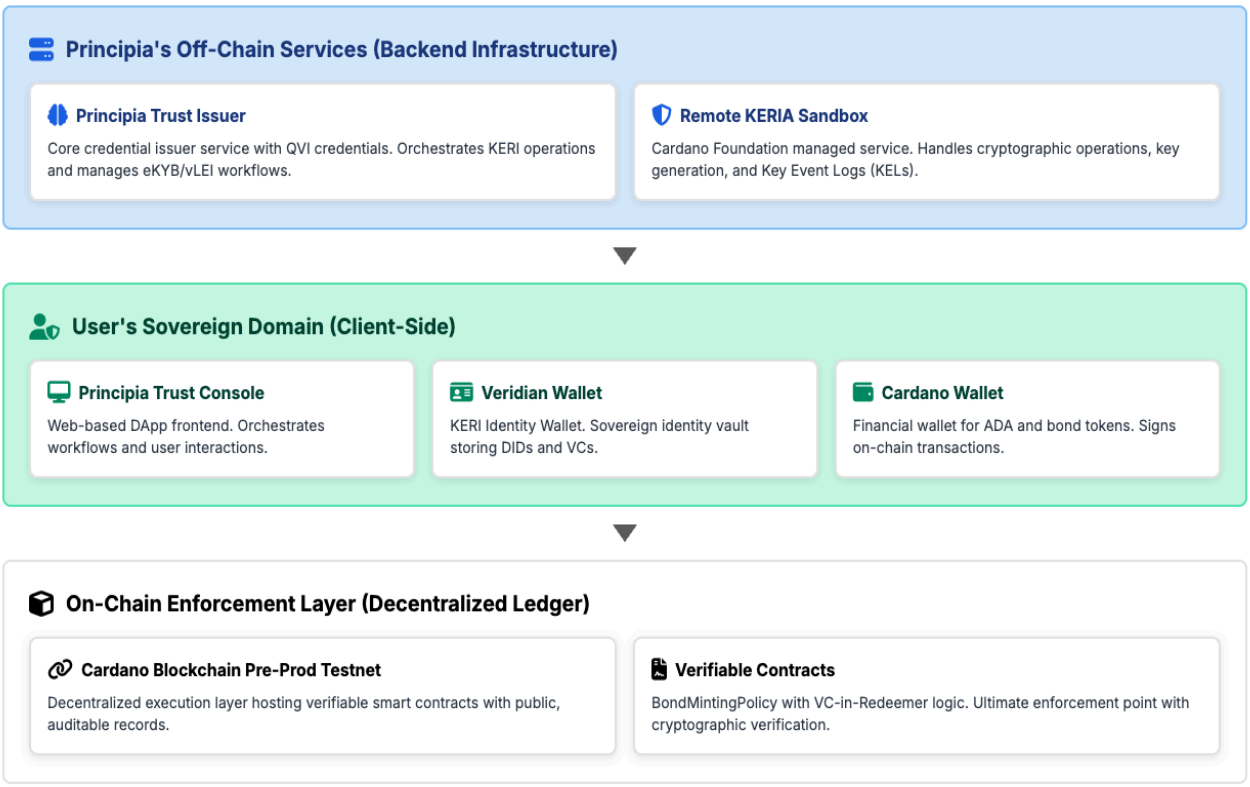

Figure 1: A high-level overview of the Principia Trust Engine's architecture.

Layer 1: Identity & Credentialing Infrastructure

This layer is responsible for all off-chain identity operations and the issuance of standards-compliant Verifiable Credentials.

- **Principia Trust Issuer:** A deployed service that functions as a Qualified vLEI Issuer (QVI) within a sandbox environment. It holds the logic for the eKYB/vLEI onboarding workflow and is the authoritative issuer of all credentials in the demonstration.
- **Remote KERIA Sandbox:** A managed KERI Agent service (provided by the Cardano Foundation) that acts as the cryptographic workhorse. The Principia Trust Issuer connects to this sandbox via API to execute all KERI-based operations.

Layer 2: User & Application

This is the user-facing layer where participants control their identities and construct transactions.

- **Principia DApp:** A web-based application that serves as the primary user interface and orchestration engine.

- **Veridian-Style Identity Wallet:** A user-controlled application that securely stores the user's private KERI keys, their DID, and all Verifiable Credentials.

Layer 3: On-Chain Enforcement (Cardano Testnet)

This is the immutable, decentralized environment where compliance rules are ultimately and deterministically enforced.

- **Verifiable Smart Contracts:** A suite of Plutus smart contracts written in Aiken, primarily the BondMintingPolicy. These contracts are the ultimate enforcement points, containing the "VC-in-Redeemer" logic.

3. Deep Dive: The On-Chain Verification Mechanism

The Trust Engine's on-chain component is defined by two core technical implementations that work in concert to achieve trustless verification.

3.1. The Dual-Signature Wallet Binding

To create an undeniable link between a user's self-sovereign identity (their KERI AID) and their financial identity (their Cardano wallet), we implemented a dual-signature cryptographic attestation.

- **Process:** The DApp orchestrates a ceremony where the user provides two distinct proofs:
  1. **Proof of Wallet Control:** A signature from their Cardano Wallet over a Sig_structure as per CIP-30.
  2. **Proof of Identity Control:** A signature from their KERI AID (via the Veridian Wallet) over a canonical binding message (BIND|v1|...).


```
"binding": {
   "v": "KERI10JSON00057d_",
   "t": "cardano_address_binding",
   "issuer": "EHtHel57-67z7KMGXKanDuoI-IsD_kcCv6iYmiLS6VPG",
   "holder": "EIk0F5px0GHvyw-bFvW7FAbGHz82fDx5VSGg77K-lEgB",
   "cardanoAddress":
"addr_test1qrcsss93xau2p8dq8gkuu5mtpk7cl74glpaxat74gr6l8r0lxt23ydw4myntryvlrzunyunqy5k2rxglzwsywrtdj6wscz2xf5",
   "cardanoPublicKey": "eda779b2c99f5915fe4841ca35017a41be5fd79046ebb87e02a0988f6f5550b2",
   "canonicalMessage":
"BIND|v1|EIk0F5px0GHvyw-bFvW7FAbGHz82fDx5VSGg77K-lEgB|addr_test1qrcsss93xau2p8dq8gkuu5mtpk7cl74glpaxat74g
r6l8r0lxt23ydw4myntryvlrzunyunqy5k2rxglzwsywrtdj6wscz2xf5",
   "signature": {
     "cardano": "845846a20127676164647265737358390 0f10840b..",
     "veridian":
"0BDjMaC3zPZV8OfS1KNiQKUYbA_Ms3ICW42f-mvNMTcJAbUapEI_4mWNQ3YvMLZxPB39c1hZqJLlQkBIRVTi7LMO"
   }
 }
```


The on-chain validator then verifies the Veridian signature like so:let keri_valid =
 verify_ed25519_signature(
  redeemer.holder_public_key,
  // 32-byte Ed25519 public key from holder AID

```
    redeemer.canonical_message,
    // Canonical binding message
    redeemer.veridian_signature,
  )
```

- **Artifact:** These proofs are assembled into a comprehensive BindingRedeemer data structure, a self-contained attestation that is passed on-chain. This binding is then anchored in the issuer's KERI Key Event Log (KEL), creating a permanent and auditable record of the ceremony.

3.2. The "VC-in-Redeemer" Enforcement Logic

This is the core of our Verifiable Smart Contract implementation. When a user initiates a transaction, the DApp places the required "proof package" (e.g., the vLEI Credential Chain and the BindingRedeemer) into the transaction's Redeemer. The on-chain validator script then performs a rigorous, multi-step verification:

1. **Verify the Wallet Binding:** It performs the full dual-signature verification on the BindingRedeemer to prove that the transaction signatory legitimately controls the claimed KERI AID.
2. **Verify the vLEI Chain of Trust (MVP):** For the hackathon MVP, we implemented a pragmatic, gas-efficient approach. The on-chain validator verifies the structure and cryptographic linkage of the simplified credential chain provided in the redeemer. It performs two key checks:
    - *LEI Consistency:* It confirms that the LEI is identical across all credentials in the chain.
    - *Edge Verification:* It confirms the cryptographic links (SAIDs) between the credentials are unbroken (Role → LE → QVI).
3. **Enforce Policy:** It checks the specific claims within the credentials against the action being performed (e.g., confirming the user has the 'Bond_Minter' role).

```
validator bond_minting_policy {
  mint(redeemer: BondMintingRedeemer, _own_policy: PolicyId, tx: Transaction) {
    let bond_datum: BondDatum = redeemer.bond_datum

    // 1. Verify 3-credential chain (LEI consistency + edge verification)
    let credential_chain_valid =
      verify_credential_chain(
        redeemer.qvi_lei,
        redeemer.le_lei,
        redeemer.role_lei,
        redeemer.qvi_credential_said,
        redeemer.le_credential_said,
        redeemer.role_credential_said,
        redeemer.le_qvi_edge,
        redeemer.role_le_edge,
      )

    let key_hash: VerificationKeyHash =
```

```
      builtin.blake2b_224(redeemer.binding_proof.cardano_public_key)
    let signed_by_key = list.has(tx.extra_signatories, key_hash)

    // 2. Verify wallet binding (Verifiable Smart Contract authorization)
    let binding_valid = verify_binding(redeemer.binding_proof)

    // 3. Verify entity-AID match (holder_aid from binding == entity_aid from vLEI)
    let aid_match =
      redeemer.binding_proof.holder_aid == bond_datum.issuer_entity_aid

    // 4. Verify entity attribution integrity (LEI from bond datum matches credential chain)
    let attribution_valid = bond_datum.issuer_lei == redeemer.le_lei

    // 5. Validate LEI format (must be 20 characters)
    let lei_valid = builtin.length_of_bytearray(redeemer.le_lei) == 20

    // 6. Validate bond parameters
    let params_valid = validate_bond_parameters(bond_datum)

    // 7. Verify initial status is Funding
    let status_valid = bond_datum.status == Funding

    // 8. Verify transaction is signed
    let signed = list.length(tx.extra_signatories) > 0

    // All validations must pass
    credential_chain_valid? && binding_valid? && aid_match? && attribution_valid? && lei_valid? && params_valid? && status_valid?
&& signed?
    binding_valid? && status_valid && signed_by_key?
  }

  else(_) {
    fail
  }
}
```

The transaction is authorized if and only if every cryptographic check in this sequence passes.

4. Key Achievements of the Hackathon

The project successfully delivered a functional, end-to-end prototype, proving the viability of the architecture. The key achievements are:

1. **Credential Issuance:** We successfully deployed the Principia Trust Issuer service and integrated it with the remote KERIA sandbox. This service can programmatically create KERI identifiers and issue a complete, ACDC-formatted vLEI Credential Chain, including the dual-signature wallet binding.
2. **Live On-Chain Verification:** We deployed a BondMintingPolicy smart contract to the Cardano testnet. This contract successfully functions as a Verifiable Smart Contract, capable of parsing the full proof package from the redeemer and performing the

on-chain verification of delegated corporate authority.
3. **End-to-End Compliant Workflow:** We successfully demonstrated the entire "Flow of Verifiable Trust" on the Cardano testnet. This included the off-chain issuance of credentials, the construction of a compliance-aware transaction, and the successful on-chain enforcement, culminating in the creation of an identity-verified digital asset.

5. Next Steps & Future Work

This successful prototype provides a clear and validated foundation for the production-ready Principia Protocol. The next steps are focused on hardening the system and expanding its capabilities.

1. **Full On-Chain Credential Chain Validation:** The current MVP verifies the structure of the vLEI chain on-chain for gas efficiency. The next step is to research and implement optimized on-chain parsers for ACDC and CESR signature formats. This would allow the validator to perform the full cryptographic signature verification of the entire credential chain on-chain, removing any trust assumption and completing the on-chain trust model.
2. **Wallet Binding Security Enhancement:** The next enhancement is to integrate a check against an on-chain revocation registry. This would allow the validator to confirm in real-time that the Wallet Binding Credential has not been revoked by the issuer since it was created.
3. **Full Protocol Integration:** The Trust Engine's verification logic will be integrated into the other core smart contracts of the bond protocol that require compliance checks, namely the CommitmentValidator (for investor KYC) and the SecondaryMarketValidator (for compliant trading).
4. **Regulatory Engagement:** This involves moving from the sandbox to a production KERI infrastructure, undergoing a formal process to obtain official GLEIF QVI accreditation, and using this powerful prototype as a key asset in our engagement with regulators in the UAE (VARA, ADGM) to champion the adoption of the vLEI standard.
5. **Pilot Program Launch:** The ultimate next step is the launch of the Principia Bond Protocol Pilot Program on the Cardano mainnet. This will involve the issuance of the first set of real, legally-binding, and fully-compliant tokenized bonds through our platform. This pilot will provide invaluable data on market demand, operational efficiency, and user experience, paving the way for a full commercial launch.

6. Conclusion

The Principia Trust Engine successfully demonstrates that the vLEI standard, when combined with a robust self-sovereign identity framework like KERI and a novel on-chain enforcement model, can effectively close the Corporate Authority Gap. Our "VC-in-Redeemer" approach provides a scalable, secure, and provably compliant solution for integrating real-world legal identity into decentralized applications. The achievements of this hackathon lay the groundwork for a new generation of institutional-grade financial products on the blockchain, built on a foundation of verifiable trust.