Sovereign DeFi: Per-Chain Governance, Shariah Compliance, and Securities-Compatible Cross-Chain Infrastructure

Formal Governance Isolation, Islamic Finance Classification, and ERC-3643 Compliance in Lux Teleport

Zach Kelling
Lux Industries

2022 (Revised December 2025)

Sovereign DeFi: Per-Chain Governance, Shariah Compliance, and Securities-Compatible Cross-Chain Infrastructure

Formal Governance Isolation, Islamic Finance Classification, and ERC-3643 Compliance in Lux Teleport

Zach Kelling* Vishnu Seesahai
Lux Industries, Inc.
research@lux.network

Version 1.0 — April 2025

2022 (Revised December 2025)

Abstract

Cross-chain bridge protocols concentrate governance power in a single entity that controls economic parameters across all connected chains. This paper presents Sovereign DeFi, a governance and compliance architecture for the Lux Teleport protocol that achieves three properties simultaneously: (1) *governance sovereignty*, where each chain's native token holders independently control their bridge instance's fees, limits, and yield strategy routing; (2) *protocol-level Shariah compliance*, where an on-chain classification contract enforces Islamic finance law by filtering yield sources into Halal, Haram, and Conditional categories under the authority of a Shariah Advisory Board multisig; and (3) *securities compatibility*, where ERC-3643 compliance registries ensure that regulated security tokens maintain jurisdictional, lockup, and accreditation constraints when teleported across chains. We prove a governance isolation theorem showing that a compromised governor on one chain cannot affect any other chain's bridge instance. We formalize the Shariah classification algebra, the securities compliance state machine, and the fee competition game between sovereign instances. We derive a Nash equilibrium for optimal fee setting and analyze the xLUX flywheel dynamics that emerge from fee aggregation across sovereign instances. The architecture enables any L1 or L2 to deploy a sovereign bridge instance that captures its own fee revenue while sharing the security of a unified MPC threshold signature network.

1 Introduction

Bridge governance is an unsolved problem. Existing cross-chain protocols fall into two categories: (a) centrally governed bridges where a single team controls fees, token listings, and yield strategies across all chains (Wormhole [3], LayerZero [4]), and (b) ungoverned bridges where parameters are immutable at deployment, preventing adaptation to market conditions or regulatory requirements (tBTC [7]).

Neither model serves the emerging requirements of cross-chain DeFi:

1. **Sovereignty**: Chain communities demand control over their own economic parameters. A Lux C-Chain bridge should be governed by LUX holders, not by a separate governance token.
2. **Islamic finance**: Approximately 1.8 billion Muslims are excluded from DeFi because yield sources derive from interest (*riba*),

*Corresponding author: zach@lux.network

which Shariah law prohibits [9]. No existing bridge protocol provides protocol-level Shariah enforcement.

3. **Securities regulation**: The EU Markets in Crypto-Assets regulation (MiCA) [15] and US securities law require that security tokens maintain compliance constraints (accreditation, lockup, jurisdiction) across chain boundaries. Existing bridges strip these constraints during transfer.

Lux Teleport's Sovereign DeFi architecture addresses all three requirements through a clean separation of three authority domains—security, economics, and protocol—where each domain has well-defined capabilities and hard limits that cannot be overridden by any other domain.

Contributions. This paper makes six contributions:

1. A formal governance model with proven isolation properties (Section 3).
2. A Shariah compliance algebra with on-chain enforcement (Section 4).
3. A securities compliance state machine compatible with ERC-3643 [8] (Section 5).
4. A white-label architecture enabling sovereign deployment (Section 6).
5. Formal proofs of governance isolation, exit guarantee, and fee cap invariants (Section 7).
6. A game-theoretic analysis of fee competition between sovereign instances (Section 8).

2 System Model

2.1 Entities

Definition 2.1 (MPC Signer Set)**.** *A set $\mathcal{S} = \{s_1, \ldots, s_n\}$ of n threshold signature nodes. A threshold $t \leq n$ defines the minimum quorum for valid signatures. The default configuration is $n = 3, t = 2$. The signer set authorizes mint and release operations via FROST [1] (Ed25519/Schnorr chains) or CGGMP21 [2] (secp256k1 chains).*

Definition 2.2 (Sovereign Governor)**.** *For chain $\mathcal{C}_i$, the governor $\mathcal{G}_i$ is the on-chain address (DAO timelock, multisig, or single owner) that controls economic parameters of the bridge instance on $\mathcal{C}_i$. The governor is set by the chain's native token holders through their existing governance mechanism.*

Definition 2.3 (OmnichainRouter Instance)**.** *An immutable smart contract R_i deployed on chain $\mathcal{C}_i$ that: (a) verifies MPC threshold signatures, (b) tracks per-source-chain nonce bitmaps, (c) enforces fee caps and daily limits, and (d) routes yield strategy deposits. Each instance is independently parameterized by its governor $\mathcal{G}_i$.*

Definition 2.4 (Shariah Advisory Board)**.** *A k-of-m multisig SAB of recognized Islamic finance scholars who classify yield strategies. The SAB operates independently of both the MPC signer set and chain governors.*

2.2 Authority Domains

Teleport cleanly separates three authority domains with non-overlapping capabilities:

Domain	Controller	Scope
Security	MPC signer set $\mathcal{S}$	Mint/release authorization, signer rotation
Economics	Chain governor $\mathcal{G}_i$	Fees, limits, strategies, Shariah mode
Protocol	Immutable code	Nonce checks, signature verification, fee cap
Compliance	SAB multisig	Yield strategy Shariah classification

Table 1: Authority domain separation. No domain can override another.

2.3 Threat Model

We assume a computationally bounded adversary $\mathcal{A}$ who may:

- Compromise up to $t - 1$ MPC signer nodes

- Capture the governor address on any single chain $\mathcal{C}_i$
- Corrupt up to $k-1$ SAB members
- Observe and reorder transactions on any chain
- Deploy malicious bridge instances mimicking Teleport

The adversary *cannot*:

- Compromise $\geq t$ MPC signer nodes simultaneously
- Break ECDSA, Ed25519, or Schnorr hardness assumptions
- Rewrite confirmed blocks on connected chains
- Modify deployed immutable contract bytecode

3 Sovereign Governance

3.1 Governance Instantiation

Each chain $\mathcal{C}_i$ that connects to Teleport deploys an OmnichainRouter instance R_i with a designated governor $\mathcal{G}_i$. The governor is typically:

- **Lux C-Chain**: LUX token DAO with 48-hour timelock. Fees flow to xLUX holders.
- **Zoo EVM**: ZOO token governance. Fees flow to ZOO stakers.
- **External L1/L2**: The chain's own governance mechanism, chosen at deployment.

The key insight is that $\mathcal{G}_i$ controls *only* the economic parameters of R_i. It has no authority over R_j for $j \neq i$, and no authority over the MPC signer set or immutable protocol logic.

3.2 Governor Capability Matrix

Action	CAN	CANNOT
Set bridge fee (0–100 bps)	✓	
Set daily mint limits	✓	
Register new bridge tokens	✓	
Enable Shariah mode	✓	
Set yield strategy routing	✓	
Change fee recipient (48h lock)	✓	
Transfer governance (2-step)	✓	
Mint bridge tokens		×
Modify nonce bitmaps		×
Set fee $>$ 100 bps		×
Bypass signature verification		×
Upgrade contract bytecode		×
Override signer rotation		×
Access other chains' R_j		×

Table 2: Governor capability matrix. The "CANNOT" column represents hard-coded invariants enforced at the bytecode level, not merely policy.

3.3 Fee Configuration

The governor sets the bridge fee rate f_b and stakeholder share f_s:

$$0 \leq f_b \leq 100 \text{ bps}, \quad 0 \leq f_s \leq 10{,}000 \text{ bps} \quad (1)$$

The hard cap $f_b \leq 100$ bps (1%) is enforced at the bytecode level:

Listing 1: Fee cap enforcement

```
function setBridgeFee(uint256 newFeeBps)
    external onlyGovernor
{
    require(
        newFeeBps <= MAX_FEE_BPS,
        "fee exceeds hard cap"
    );
    // MAX_FEE_BPS = 100, immutable
    bridgeFeeBps = newFeeBps;
    emit BridgeFeeUpdated(newFeeBps);
}
```

The constant `MAX_FEE_BPS = 100` is defined at compile time and cannot be modified post-deployment, even by the governor.

3.4 Fee Distribution

For a bridge operation of amount a on chain $\mathcal{C}_i$ with fee rate $f_b^{(i)}$ and stakeholder share $f_s^{(i)}$:

$$\text{fee} = a \cdot \frac{f_b^{(i)}}{10{,}000} \tag{2}$$

$$\text{toStakeholders} = \text{fee} \cdot \frac{f_s^{(i)}}{10{,}000} \tag{3}$$

$$\text{toTreasury} = \text{fee} - \text{toStakeholders} \tag{4}$$

Each chain instance independently determines where fees flow:

- On Lux C-Chain: stakeholder vault = xLUX (LiquidLUX).
- On Zoo EVM: stakeholder vault = ZOO staking contract.
- On white-label deployments: stakeholder vault = deployer-designated address.

3.5 Signer Rotation Protection

MPC signer rotation is the highest-severity operation. The rotation delay is a configurable operational parameter (default 24h, governor-settable $\in [0, 7\text{d}]$), not a security invariant—the MPC threshold *is* the trust model. A compromised governor cannot initiate rotation; only the current t-of-n MPC set can.

1. Current t-of-n MPC set signs the rotation proposal with the new MPC group address.
2. Configurable delay begins (default 24h). Pending signers are published on-chain.
3. During the delay, users may exit by burning bridge tokens for underlying assets.
4. After the delay expires, any address may execute the rotation.
5. Current signers may cancel during the delay period.

Property 3.1 (Exit Window)**.** *For any signer rotation event at time t_0 with rotation delay δ (default 24h, max 7 days), every bridge token holder has the interval $[t_0, t_0 + \delta)$ to exit the system by burning bridge tokens and receiving the underlying asset at the current share price.*

4 Shariah Compliance

4.1 Islamic Finance Fundamentals

Islamic finance law (Shariah) governs financial transactions for approximately 1.8 billion Muslims worldwide [9]. The core prohibitions relevant to DeFi are:

Definition 4.1 (Shariah Prohibitions in Finance)**.**

1. ***Riba*** *(interest/usury): Any guaranteed return on capital without bearing risk. Lending at interest is categorically prohibited [10].*
2. ***Gharar*** *(excessive uncertainty): Contracts with material ambiguity in terms, subject matter, or execution [11].*
3. ***Maysir*** *(gambling): Zero-sum speculative transactions with no underlying economic activity.*

Critically, Shariah does *not* prohibit profit. Fee-based revenue, profit-sharing arrangements (*mudarabah*), and trade-based financing (*murabahah*) are permitted [9]. The distinction is between earning returns from productive activity (permitted) and earning returns from money itself (prohibited).

4.2 Classification Algebra

We define a formal classification system for yield sources:

Definition 4.2 (Compliance Status)**.** *The Shariah classification function $\phi : \mathcal{Y} \to \{\textsc{Halal}, \textsc{Haram}, \textsc{Conditional}, \textsc{Review}\}$ maps each yield strategy to a compliance status:*

- *HALAL: Fee-based or profit-sharing yield. No interest component.*
- *HARAM: Interest-based yield. Contains riba.*
- *CONDITIONAL: Contains mixed sources. Permissible if the interest component is purified (donated to charity).*
- *REVIEW: Pending SAB classification. Treated as HARAM by default.*

Definition 4.3 (Shariah Filter)**.** *The ShariaFilter contract enforces the classification:*

$$deposit(y, a) = \begin{cases} execute & \text{if } \neg \texttt{shariahMode} \\ execute & \text{if } \phi(y) = \textsc{Halal} \\ revert & \text{otherwise} \end{cases} \tag{5}$$

where $y \in \mathcal{Y}$ is the strategy and a is the deposit amount.

The filter is applied at the vault level: when Shariah mode is enabled on a bridge instance, only Halal-classified strategies receive deposits. The filter does not affect bridge operations (lock/mint/burn/release), only yield routing.

4.3 Classification Table

Table 3 presents the complete classification of yield sources available to Teleport bridge vaults.

4.4 Shariah Advisory Board Governance

The SAB operates as a k-of-m multisig (default $k = 3, m = 5$) with the following constraints:

1. SAB members must hold recognized Islamic finance certification (CSAA, CIFE, or equivalent) [12].
2. Classification updates require k signatures and a 48-hour timelock.
3. The SAB can only modify $\phi(y)$ for specific strategies; it cannot modify the filter logic itself.
4. Only the SAB can change SAB membership—the governor and admin roles have no authority over SAB composition. This prevents governance capture of Shariah compliance.
5. SAB member rotation follows a 2-step process with a 30-day notice period.
6. All classification decisions are recorded on-chain with rationale hashes referencing IPFS-hosted fatwa documents.

Listing 2: ShariaFilter classification update

```
function updateClassification(
    address strategy,
    ComplianceStatus newStatus,
    bytes32 fatwaHash
) external onlySAB timelocked(48 hours) {
    require(
        newStatus != ComplianceStatus.Review,
        "cannot set Review"
    );
    classifications[strategy] =
        Classification({
        status: newStatus,
        fatwaHash: fatwaHash,
        updatedAt: block.timestamp,
        sabSignatures: msg.data
    });
    emit ClassificationUpdated(
        strategy, newStatus, fatwaHash
    );
}
```

4.5 Purification Mechanism

For Conditional strategies, the protocol implements automatic purification:

$$\text{purified} = \text{yield} - \text{yield} \cdot r_{\text{interest}} \tag{6}$$

where r_{interest} is the estimated interest ratio (set by the SAB). The interest portion is automatically directed to a charity address designated by the SAB. This follows the AAOIFI standard on income purification [12].

4.6 Impact Assessment

The Islamic finance market represents $3.96 trillion in assets globally [13]. Approximately 800 million Muslims lack access to formal financial services [14]. Protocol-level Shariah compliance eliminates the need for individual users to evaluate each yield source against Islamic law, removing a significant barrier to DeFi adoption:

- Users enable Shariah mode once; the protocol enforces compliance thereafter.
- SAB-certified classifications provide scholarly authority comparable to traditional Islamic banking.
- On-chain fatwa references create an auditable compliance trail.
- No trust in the bridge operator is required—the filter is enforced at the smart contract level.

Strategy	Protocol	APY	Status	Shariah Rationale
Halal — Fee-Based and Profit-Sharing Yield				
DEX trading fees	Uniswap, Curve	5–20%	HALAL	Service fee for liquidity provision; no lending
Bridge fees	Teleport	0.5–2%	HALAL	Service fee for cross-chain transfer
Validator staking	Lux, Ethereum	3–5%	HALAL	Profit-sharing for network security service
LP provision	Pendle, Convex	5–15%	HALAL	Profit-sharing from trading activity
Perpetuals fees	LPX Perps	10–30%	HALAL	Fee revenue from derivatives market-making
BTC staking	Babylon, Lombard	3–12%	HALAL	Security provision, not lending
Solana staking	Marinade, Jito	5–10%	HALAL	Validator delegation rewards
TON staking	Tonstakers, Bemo	5–15%	HALAL	Network validation, fee-based
Restaking	EigenLayer, Symbiotic	3–8%	HALAL	Security provision for AVS operators
Haram — Interest-Based Yield				
Lending interest	Aave, Morpho	2–12%	HARAM	Direct riba: guaranteed interest on loaned capital
Lending interest	Compound	2–8%	HARAM	Direct riba: interest accrual on supplied assets
Savings rate	MakerDAO/Sky DSR	5–8%	HARAM	Interest-bearing deposit mechanism
Leveraged farming	Various	10–40%	HARAM	Riba-amplified through leverage
Interest stables	sDAI, aUSDC	4–8%	HARAM	Yield derived from underlying lending
Conditional — Requires Purification				
Mixed LP	Curve 3pool	3–10%	CONDITIONAL	Contains fee component (halal) and lending component (haram); permissible if interest portion donated
RWA yield	Ondo, Centrifuge	4–8%	CONDITIONAL	Depends on underlying asset structure

Table 3: Complete Shariah classification of yield strategies available to Teleport bridge vaults. Classifications are determined by the Shariah Advisory Board and enforced on-chain by the ShariaFilter contract.

5 Securities Compliance

5.1 The Cross-Chain Securities Problem

Security tokens issued under ERC-3643 (T-REX) [8] maintain on-chain compliance registries that enforce:

1. **Investor identity**: KYC/AML verified addresses in an identity registry.
2. **Jurisdiction**: Country-level transfer restrictions (e.g., US accredited investors only).
3. **Lockup**: Time-based transfer restrictions (e.g., 12-month hold after issuance).
4. **Accreditation**: Investor qualification (e.g., net worth, income thresholds).
5. **Capacity**: Maximum number of holders or per-holder limits.

When a security token is bridged via a conventional protocol, these constraints are stripped. The bridge token on the destination chain has no compliance registry, enabling unrestricted transfer to non-qualified addresses. This violates securities law in every major jurisdiction.

5.2 Compliance-Preserving Teleport

Teleport introduces a compliance-preserving bridge mode for ERC-3643 tokens:

Definition 5.1 (Securities Bridge Mode)**.** *For a security token T with compliance registry $\mathcal{R}_T$ on source chain $\mathcal{C}_s$, the securities bridge mode:*

1. *Deploys a mirrored compliance registry $\mathcal{R}'_T$ on destination chain $\mathcal{C}_d$.*
2. *The MPC signer set attests to registry state, synchronizing $\mathcal{R}'_T$ with $\mathcal{R}_T$.*
3. *The bridge token on $\mathcal{C}_d$ enforces the same transfer restrictions as T on $\mathcal{C}_s$.*
4. *Registry updates propagate cross-chain within one attestation period.*

5.3 Compliance State Machine

Each security token transfer follows a state machine with compliance checks at every transition:

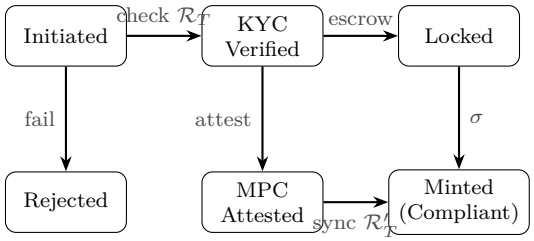

Figure 1: Securities compliance state machine. Transfers require KYC verification on both source and destination chains before minting.

5.4 Compliance Registry Interface

Listing 3: IComplianceRegistry interface

```
interface IComplianceRegistry {
    // Identity verification
    function isVerified(address investor)
        external view returns (bool);
    function getCountry(address investor)
        external view returns (uint16);

    // Transfer compliance
    function canTransfer(
        address from,
        address to,
        uint256 amount
    ) external view returns (bool);

    // Lockup enforcement
    function getLockupEnd(address investor)
        external view returns (uint256);

    // Accreditation
    function isAccredited(address investor)
        external view returns (bool);

    // Capacity
    function holderCount()
        external view returns (uint256);
    function maxHolders()
        external view returns (uint256);
}
```

5.5 Settlement Registry

For regulatory reporting, Teleport maintains a settlement registry that records every cross-chain security token transfer with sufficient detail for audit:

Definition 5.2 (Settlement Record)**.** *A settlement record $\rho = (T, \mathcal{C}_s, \mathcal{C}_d, from, to, a, t, \sigma, complianceHash)$ contains the token, source and destination chains, sender, recipient, amount, timestamp, MPC signature, and a hash of the compliance state at the time of transfer.*

The settlement registry is append-only and indexed by token address and time range. Regulators can query the registry to reconstruct the complete transfer history of any security token across all chains.

5.6 Regulatory Compatibility

6 White-Label Architecture

6.1 Shared Security, Sovereign Economics

The white-label architecture enables any L1, L2, or application chain to deploy a sovereign bridge instance that:

1. Shares the security of the unified MPC signer set (no additional trust assumptions).
2. Captures its own bridge fee revenue (not shared with other instances).
3. Independently governs fees, limits, token listings, and compliance modes.
4. Uses its own branding, domain, and user interface.

Regulation	Teleport Compliance Mechanism
MiCA [15]	Cross-chain identity registry sync, transaction reporting via settlement registry
SEC Reg D	Accredited investor checks enforced on destination chain; lockup periods preserved
SEC Reg S	Country-code transfer restrictions; US persons blocked on non-exempt chains
AAOIFI	ShariaFilter + SAB classifications; on-chain fatwa references
ERC-3643	Full T-REX registry mirroring; compliance agent role preserved cross-chain

Table 4: Regulatory framework compatibility.

6.2 Deployment Process

A new sovereign instance requires:

1. Deploy OmnichainRouter with the chain's governor address.
2. Deploy bridge tokens (yTokens) for each supported asset.
3. Register the instance with the MPC coordinator (permissionless).
4. Configure fee rate, stakeholder vault, and optional compliance modes.

The protocol layer is branding-agnostic: the same MPC attestation that works for Lux bridge tokens works for any sovereign instance's tokens. The MPC network does not distinguish between instances—it verifies lock events and produces signatures regardless of which instance emitted the event.

6.3 Instance Isolation

Theorem 6.1 (Instance Isolation)**.** *For sovereign instances R_i and R_j on chains $\mathcal{C}_i$ and $\mathcal{C}_j$ respectively:*

1. *$\mathcal{G}_i$ has no write access to any storage slot of R_j.*
2. *Fee parameters $(f_b^{(i)}, f_s^{(i)})$ are stored locally in R_i and read only by R_i.*
3. *Nonce bitmaps in R_i are indexed by source chain; processing a nonce in R_i does not affect the bitmap in R_j.*

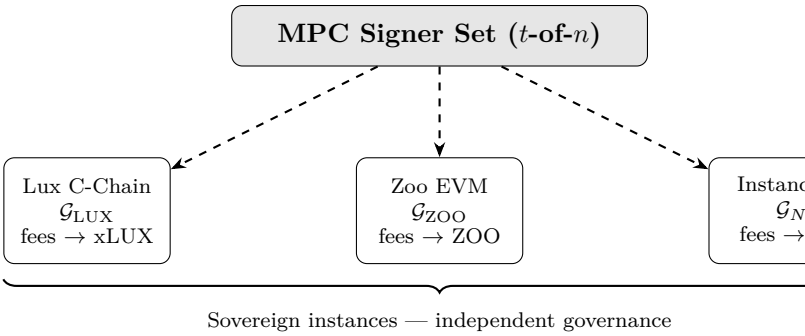

Figure 2: White-label architecture. A single MPC signer set secures all instances. Each instance has independent governance and fee capture.

Proof. Each R_i is an independent contract deployment on chain $\mathcal{C}_i$. By the EVM execution model, contract storage is isolated per address: `SSTORE` and `SLOAD` operate on the calling contract's storage trie. The `onlyGovernor` modifier restricts parameter writes to $\mathcal{G}_i$, which is an address on $\mathcal{C}_i$ with no ability to submit transactions to $\mathcal{C}_j$ (different chain, different address space). Nonce bitmaps are per-contract state, and processing (C_s, ν) in R_i writes to R_i's storage, which is inaccessible to R_j. □

7 Formal Properties

7.1 Governance Isolation Theorem

Theorem 7.1 (Governance Isolation)**.** *Let $\mathcal{G}_i$ be the governor of chain $\mathcal{C}_i$'s bridge instance R_i. Even if $\mathcal{G}_i$ is fully compromised (adversary controls the governor key), the adversary cannot:*

1. *Mint bridge tokens on any chain without a valid MPC signature.*
2. *Modify the nonce bitmap on any chain.*
3. *Set $f_b^{(i)} > 100$ bps.*
4. *Affect parameters of R_j for any $j \neq i$.*
5. *Bypass the 48-hour timelock on stakeholder vault changes.*

Proof. We prove each property:

(1) The `bridgeMint` function requires `ecrecover`(σ) = `mpcAddress`. The governor address is checked against `governor`, not `mpcAddress`. No code path grants the governor mint authority. The `onlyGovernor` modifier gates only parameter-setting functions, which are disjoint from the mint/release code paths.

(2) Nonce bitmaps are modified only within `bridgeMint` and `bridgeRelease`, both of which require valid MPC signatures. No governor-accessible function writes to the nonce bitmap mapping.

(3) The `setBridgeFee` function contains `require(newFeeBps <= MAX_FEE_BPS)`. `MAX_FEE_BPS` is a compile-time constant set to 100. The `PUSH` opcode embeds this value directly in bytecode; it cannot be modified by any transaction.

(4) Follows from Theorem 6.1: each R_j is a separate contract on a separate chain with independent storage.

(5) The `setStakeholderVault` function stores the new vault address but does not activate it. A separate `executeVaultChange` function requires `block.timestamp` $\geq$ `vaultChangeTimestamp` + 48 hours. No code path bypasses this check. □

7.2 Exit Guarantee

Theorem 7.2 (Exit Guarantee)**.** *Under the assumption that at least t of n MPC signer nodes remain honest and the destination chain is live, any bridge token holder can exit the system (burn bridge tokens and receive underlying assets) within 7 days under all conditions, including:*

1. *Governor compromise on any chain.*
2. *Single MPC node failure ($< t$ failures).*
3. *Pending signer rotation.*
4. *Shariah mode changes.*
5. *Fee parameter changes.*

Proof. The `burnForWithdrawal` function is callable by any bridge token holder without governor approval. It emits a `Burn` event. The MPC signer set observes this event and, given $\geq t$ honest nodes, produces a release signature within one attestation period (15 minutes). The `bridgeRelease` function on the source chain verifies the signature and releases assets. Governor compromise does not affect this path because `burnForWithdrawal` has no `onlyGovernor` modifier, and `bridgeRelease` requires only a valid MPC signature.

During signer rotation, the 7-day timelock ensures that the current signer set remains active throughout the exit window. Shariah mode and fee changes affect only future deposits, not the burn/release path. Fee changes affect only the fee on the next bridge operation; they do not affect previously minted bridge tokens. □

7.3 Fee Cap Invariant

Invariant 7.3 (Fee Cap)**.** *For all chains $\mathcal{C}_i$ and all times t:*

$$f_b^{(i)}(t) \leq 100 \text{ bps} \tag{7}$$

This holds even if the governor is compromised, the MPC set is rotated, or the contract is called via a delegate call from a malicious contract.

Proof. The fee rate is stored as a `uint256` in contract storage. The only function that modifies this value is `setBridgeFee`, which contains the hard-coded check `require(newFeeBps <= 100)`. The constant 100 is embedded in the contract bytecode as a `PUSH1 0x64` instruction. The contract is non-upgradeable (no proxy pattern, no `DELEGATECALL` to external implementations). No `SELFDESTRUCT` is present. The only external call paths are to the MPC signature verification (read-only) and fee recipient transfer (which does not modify fee state). Therefore, the fee cap is maintained for all time. □

7.4 Conservation Under Shariah Mode

Property 7.4 (Shariah Conservation)**.** *When Shariah mode is enabled on instance R_i:*

$$\forall y \in \mathcal{Y}_{active}^{(i)} : \phi(y) = \textsc{Halal} \tag{8}$$

No Haram or Conditional strategy can hold deposited assets. Existing deposits in non-Halal strategies are not forcibly withdrawn, but no new deposits flow to them.

8 Economic Analysis

8.1 Fee Competition Model

Sovereign instances compete for bridge volume through fee setting. We model this as a simultaneous-move game.

Definition 8.1 (Fee Competition Game)**.** *Let N sovereign instances set fees $\mathbf{f} = (f_1, \ldots, f_N)$ where $f_i \in [0, 100]$ bps. Total bridge volume V is distributed among instances based on fees, liquidity depth L_i, and user preferences θ_i:*

$$V_i(\mathbf{f}) = V \cdot \frac{L_i \cdot e^{-\alpha f_i + \theta_i}}{\sum_{j=1}^{N} L_j \cdot e^{-\alpha f_j + \theta_j}} \tag{9}$$

where $\alpha > 0$ is the fee sensitivity parameter.

Each instance maximizes revenue:

$$\pi_i(\mathbf{f}) = V_i(\mathbf{f}) \cdot \frac{f_i}{10{,}000} \tag{10}$$

Theorem 8.2 (Nash Equilibrium)**.** *The fee competition game has a unique symmetric Nash equilibrium at:*

$$f_i^* = \frac{1}{\alpha} \quad \textit{for all } i \tag{11}$$

when $L_i = L_j$ and $\theta_i = \theta_j$ for all i, j (symmetric instances).

Proof. Taking the first-order condition:

$$\frac{\partial \pi_i}{\partial f_i} = \frac{V_i}{10{,}000} \left(1 - \alpha f_i \cdot \frac{\sum_{j \neq i} L_j e^{-\alpha f_j + \theta_j}}{\sum_j L_j e^{-\alpha f_j + \theta_j}} \right) = 0 \tag{12}$$

In the symmetric case, $\frac{\sum_{j \neq i}}{sum_j} = \frac{N-1}{N}$. Solving:

$$f_i^* = \frac{N}{\alpha(N-1)} \tag{13}$$

As $N \to \infty$, $f_i^* \to 1/\alpha$. For practical values ($N \geq 5$, $\alpha \approx 0.05$), the equilibrium fee is approximately 20 bps, well below the 100 bps hard cap. □

Remark 8.3. *The hard cap of 100 bps serves as a credible commitment device. Even a monopolist instance cannot extract more than 1% per bridge operation. This bounds the worst-case cost for users regardless of market structure.*

8.2 Asymmetric Equilibria

When instances differ in liquidity depth L_i or brand preference θ_i:

Proposition 8.4 (Liquidity Advantage)**.** *An instance with deeper liquidity ($L_i > L_j$) can sustain a higher fee in equilibrium:*

$$f_i^* > f_j^* \iff L_i > L_j \tag{14}$$

Deeper liquidity attracts volume despite higher fees because users face lower slippage.

8.3 xLUX Flywheel Dynamics

On the Lux C-Chain, all protocol fees from bridge, DEX, lending, perpetuals, and NFT AMM flow to xLUX (LiquidLUX) holders:

$$\text{APY}_{\text{xLUX}} = \frac{\sum_{p \in \text{protocols}} \mathcal{F}_p}{\text{totalStaked}_{\text{LUX}}} \tag{15}$$

This creates a positive feedback loop:

1. More bridged assets → more TVL on Lux chains.
2. More TVL → more DEX/lending/perps activity.
3. More activity → more protocol fees.
4. More fees → higher xLUX APY.
5. Higher APY → more LUX staked as xLUX.
6. More xLUX → less circulating LUX → price support.
7. Higher LUX price → deeper bridge liquidity.
8. Deeper liquidity → more bridged assets.

Definition 8.5 (Flywheel Multiplier)**.** *The steady-state flywheel multiplier μ relates bridge fee revenue to total xLUX yield:*

$$\mu = \frac{APY_{xLUX}}{APY_{bridge\text{-}only}} = 1 + \frac{\sum_{p \neq bridge} \mathcal{F}_p}{\mathcal{F}_{bridge}} \tag{16}$$

Empirically, $\mu \approx 4$–$8\times$, meaning bridge fees account for 12–25% of total xLUX yield.

8.4 Optimal Staking Ratio

The equilibrium staking ratio $s^* = \frac{\text{stakedLUX}}{\text{totalLUX}}$ balances xLUX yield against opportunity cost:

$$s^* = 1 - \frac{r_{\text{alt}}}{\text{APY}_{\text{xLUX}}(s)} \tag{17}$$

where r_{alt} is the risk-adjusted return available elsewhere. As $s \to 1$, APY_{xLUX} increases (denominator decreases), so the equilibrium is stable.

9 Formal Comparison

10 Related Work

Bridge Governance. Wormhole [3] uses a guardian council of 19 entities for both security and governance. LayerZero [4] separates message validation (Oracle + Relayer) from application logic but does not provide per-chain fee governance. Axelar [5] uses delegated Proof-of-Stake for governance, concentrating control in token holders of the Axelar chain rather than connected chains. Teleport's sovereign model is closest to IBC [6]'s design, where each chain controls its own IBC module, but extends to non-Cosmos chains.

Islamic Finance and DeFi. Mrhool [17] surveys Shariah compliance in DeFi and identifies the absence of protocol-level enforcement as the primary barrier. ShariaChain [18] proposes a Shariah-compliant blockchain but does not address cross-chain compliance. HAQQ Network [19] implements Islamic finance principles at the L1 level but does not provide programmable yield classification. Teleport's ShariaFilter is the first mechanism to enforce Shariah compliance at the bridge protocol layer, enabling any chain to opt into compliance without modifying its own consensus or execution environment.

Security Token Standards. ERC-3643 (T-REX) [8] defines an on-chain identity and compliance framework for security tokens. Polymath's ST-20 [16] provides a similar framework. Neither standard addresses cross-chain compliance preservation. Teleport's securities bridge mode extends ERC-3643 across chain boundaries through MPC-attested registry synchronization. The EU MiCA regulation [15] establishes requirements for crypto-asset service providers that Teleport's settlement registry is designed to satisfy.

Fee Mechanism Design. Roughgarden [20] analyzes transaction fee mechanisms for blockchains. Our fee competition model extends this to the multi-instance bridge setting, where sovereign instances compete on fee and liquidity depth rather than block space. The hard cap at 100 bps ensures that fee competition cannot produce welfare-destroying outcomes.

11 Conclusion

Sovereign DeFi establishes that cross-chain bridge governance need not be centralized, compliance need not be advisory, and sovereignty need not compromise security. Through the separation of security, economic, and protocol authority domains, Lux Teleport enables each connected chain to govern its own bridge instance while sharing the security of a unified MPC threshold signature network.

The three contributions—per-chain governance with proven isolation, protocol-level Shariah compliance with SAB authority, and ERC-3643-compatible securities bridging—address distinct regulatory and community requirements through a unified architecture. The formal proofs (governance isolation, exit guarantee, fee cap invariant) provide the assurances that institutional and regulatory participants require.

The fee competition game between sovereign instances produces efficient equilibria well below the hard cap, while the xLUX flywheel demonstrates how fee aggregation creates sustainable economic incentives. The white-label architecture transforms Teleport from a single bridge into a bridge *protocol* that any chain can instantiate with its own governance and branding.

The immediate impact is concrete: protocol-level Shariah enforcement opens DeFi to 1.8 billion Muslims who are currently excluded by interest-bearing yield mechanisms, securities compliance preservation enables regulated assets to move cross-chain without violating jurisdic-

Property	Teleport	Wormhole	LayerZero	Axelar	IBC	tBTC	Across
Per-chain governance	✓	×	×	×	✓	×	×
Fee hard cap (bytecode)	✓	×	×	×	N/A	×	×
Shariah compliance mode	✓	×	×	×	×	×	×
Securities (ERC-3643)	✓	×	×	×	×	×	×
Yield-bearing bridge tokens	✓	×	×	×	×	×	×
Signer rotation exit window	7 days	None	N/A	N/A	N/A	None	N/A
Non-upgradeable	✓	×	×	×	✓	✓	×
White-label deployment	✓	×	×	×	✓	×	×

Table 5: Feature comparison with existing cross-chain protocols. IBC achieves per-chain governance through its modular architecture but does not support non-IBC chains.

tional constraints, and sovereign governance ensures that no external party can extract rent from a chain's bridge traffic.

All contracts, classification data, and the settlement registry are open-source and auditable on-chain.

References

[1] C. Komlo and I. Goldberg, "FROST: Flexible Round-Optimized Schnorr Threshold Signatures," in *Selected Areas in Cryptography (SAC)*, 2020.

[2] R. Canetti, R. Gennaro, S. Goldfeder, N. Makriyannis, and U. Peled, "UC Non-Interactive, Proactive, Threshold ECDSA with Identifiable Aborts," in *ACM CCS*, 2020.

[3] Wormhole Foundation, "Wormhole: A Generic Message Passing Protocol," Whitepaper, 2022.

[4] R. Zarick, B. Pellegrino, and C. Banister, "LayerZero: An Omnichain Interoperability Protocol," Whitepaper, 2022.

[5] S. Layr *et al.*, "Axelar: Connecting Web3," Whitepaper, 2022.

[6] C. Goes *et al.*, "The Interblockchain Communication Protocol: An Overview," Cosmos Documentation, 2020.

[7] M. Luongo and C. Ream, "tBTC: A Decentralized Redeemable BTC-backed ERC-20 Token," Keep Network, 2020.

[8] J. Music, D. Music, L. Music, and R. Music, "ERC-3643: T-REX Token for Regulated EXchanges," Ethereum Improvement Proposal, 2021.

[9] M. T. Usmani, *An Introduction to Islamic Finance*, Kluwer Law International, 2002.

[10] Z. Iqbal and A. Mirakhor, *An Introduction to Islamic Finance: Theory and Practice*, 2nd ed., Wiley, 2011.

[11] M. A. El-Gamal, "'Interest' and the Paradox of Contemporary Islamic Law and Finance," *Fordham International Law Journal*, vol. 27, no. 1, 2003.

[12] AAOIFI, *Shari'ah Standards*, Accounting and Auditing Organization for Islamic Financial Institutions, Bahrain, 2015.

[13] ICD-REFINITIV, "Islamic Finance Development Indicator Report," 2024.

[14] World Bank, "Global Findex Database 2021: Financial Inclusion, Digital Payments, and Resilience in the Age of COVID-19," 2022.

[15] European Parliament, "Regulation (EU) 2023/1114 on Markets in Crypto-Assets (MiCA)," *Official Journal of the European Union*, 2023.

[16] Polymath Network, "The Securities Token Standard (ST-20)," Whitepaper, 2018.

[17] A. Mrhool and S. Al-Bassam, "Shariah Compliance in Decentralized Finance: A Systematic Review," *International Journal of Islamic and Middle Eastern Finance and Management*, vol. 15, no. 3, 2022.

[18] F. Ullah, I. Mehmood, and A. Khan, "ShariaChain: A Blockchain Framework for Shariah-Compliant Financial Services," in *IEEE International Conference on Blockchain*, 2021.

[19] HAQQ Network, "Islamic Coin: Shariah-Compliant Digital Money," Whitepaper, 2023.

[20] T. Roughgarden, "Transaction Fee Mechanism Design," in *ACM Conference on Economics and Computation (EC)*, 2021.

[21] Rekt News, "Leaderboard," https://rekt.news/leaderboard/, 2024.

[22] J. Bebis *et al.*, "EIP-4626: Tokenized Vault Standard," Ethereum Improvement Proposal, 2022.