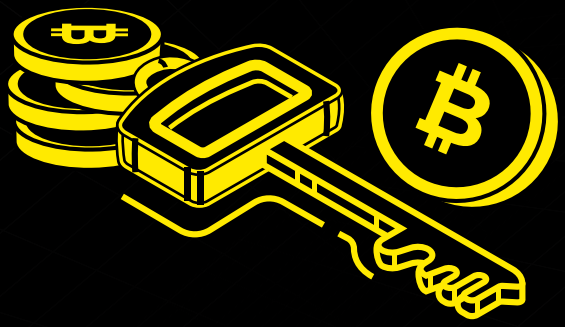
GOAT NETWORK



# SAFEBOX

## Structure & Risk Brief

A technical brief for security teams evaluating  's custody and timelock guarantees.

GOAT NETWORK

**Safebox** is a Bitcoin-native timelock system that locks BTC (for the purpose of yield generation) under a deterministic script requiring (1) an absolute locktime (CLTV) and (2) valid signatures from the user's custody configuration. These two constraints form the only valid spend path. No component of GOAT Network or **Safebox** infrastructure introduces logic capable of altering, bypassing, or influencing custody of the BTC once deposited.

The timelock is embedded directly in Bitcoin Script and enforced at consensus level. Any transaction attempting to spend the UTXO before the locktime expires is rejected by Bitcoin Core nodes and miners, making premature withdrawal structurally impossible. Once the locktime has passed, spending requires signatures from keys held solely by the institution.

GOAT Network has no signing authority, cannot introduce signatures, and cannot create alternative spend paths.

## Timelock Architecture

**Safebox** uses a minimal, auditable script template.
It contains a single, linear unlock condition and no secondary branches.

**Safebox** Locking Script:

**TIMELOCKED SAFEBOX OUTPUT**

**<locktime> OP_CHECKLOCKTIMEVERIFY**
**OP_DROP**
**<Institution Keys> OP_CHECKMULTISIG**

**Before Maturity:**
- **CLTV not satisfied → no spend possible**
- **BTC immovable**

**After Maturity:**
- **CLTV satisfied → spend requires institutional signatures**
- **No alternate or external spend path exists**

The **Offbeat Security** audit confirmed the correctness of this structure, including:

- CLTV enforcement cannot be bypassed
- No valid early-spend path exists
- No mechanism for signature substitution or key injection
- Script contains no hidden conditions or state transitions

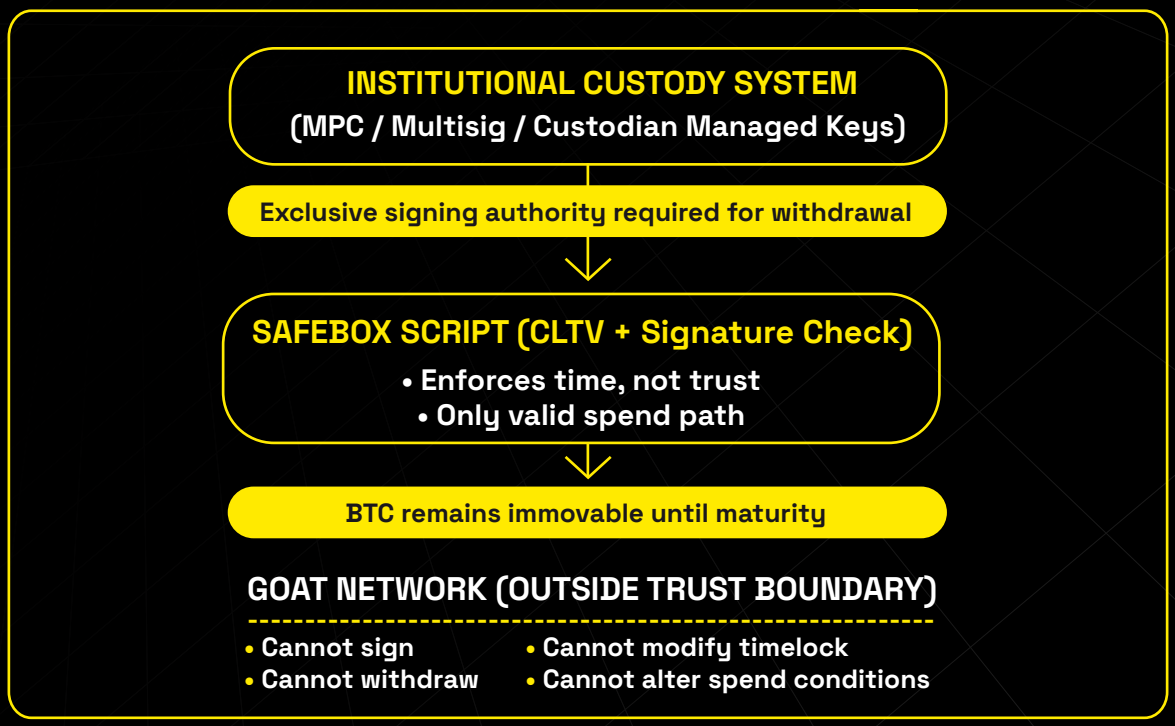
The simplicity of the script reduces the attack surface and prevents ambiguity or unintended execution paths.

## Custody & Control Boundaries

**Safebox**Manager defines permissible term lengths but has no interaction with BTC once deposited. It cannot shorten a lock, weaken script constraints, or change signature requirements. This separation ensures misconfiguration at the coordination layer cannot impact custody.

The locked UTXO is controlled exclusively by the institution.

### Control Boundary Model:



**Safebox** does not wrap BTC, does not escrow it, and does not introduce any intermediary custody layer. BTC security depends only on Bitcoin Script and institutional key management.

## Deterministic Lifecycle

**Safebox** has no multi-state contract logic, no dynamic conditions, and no reactive execution. Its behaviour is fully determined at the moment of deposit.

### **Safebox** Lifecycle:

**1. Deposit**
Institution constructs & signs deposit TX.
BTC enters CLTV script.

---

**2. Locked Period**
CLTV prevents all spending.
BTC immovable.
No actor—including GOAT—can modify this state.

---

**3. Maturity**
CLTV condition becomes true.
Script now permits signatures to release funds.

---

**4. Withdrawal**
Institution signs spend TX.
BTC returns to institution-controlled address.

Yield distribution occurs outside this lifecycle.
Yield payments do not reference the locked UTXO, do not touch it, and cannot influence its constraints.

# Technical Risk Surface

**Safebox** avoids the inherent risks present in borrowing, lending, bridging, or smart-contract yield systems. Because the script is minimal and deterministic, the risk surface is narrow and dominated by standard self-custody considerations.

The primary technical risk is incorrect or unintended script construction, such as an improperly set locktime. This risk is mitigated by script determinism and the institution's ability to test allocations:

- Institutions can verify the script template before use.
- They can execute full-cycle tests (deposit → maturity → withdrawal).
- The locking behaviour is transparent and observable on-chain.

Other risks are operational rather than structural:

- incorrect signing workflow
- lost keys
- misconfigured multisig or MPC setups

These failures are identical to any direct BTC custody scenario and not related to **Safebox**.

Network-level risks such as mempool congestion or temporary chain reorganisations may delay transactions but cannot alter CLTV enforcement or signature requirements.

**Safebox** has:

- no borrower logic
- no collateral ratio
- no liquidation mechanism
- no cross-chain dependency
- no delegated signers
- no custodial intermediaries
- no mutable or upgradeable contract state

No party can seize, reallocate, rewrap, or commingle BTC.

GOAT NETWORK

# Audit Interpretation

The **Offbeat Security** audit examined the **Safebox** locking script, its enforcement properties, and the separation between configuration logic (**Safebox**Manager) and custody logic (Bitcoin Script). The primary goal was to confirm that the mechanism exposes only one valid execution path and that no part of the surrounding infrastructure can weaken timelock or signature requirements.

Offbeat verified that **Safebox**Manager has no custody authority and cannot introduce additional spend paths; its responsibilities end at term definition, not asset control. They also validated that CLTV enforces an unbreakable time boundary and that only the institution's own key material can satisfy the post-maturity spend condition.

In effect, the audit confirms that **Safebox** custody is determined entirely by Bitcoin consensus and institutional signing authority. No third party can accelerate withdrawal, delay it, or intervene in the spend path.

## Key audit findings:

- The script exposes one linear spend condition: locktime expiry followed by institution-controlled signatures.
- No early-spend vector exists; attempts fail at mempool and mining validation.
- No mechanism for signature substitution or key injection is present in the template.
- **Safebox**Manager cannot modify or override script constraints after deposit.
- The script contains no hidden branches or alternative conditions that could redirect control.

The audit's assessment confirms that **Safebox** introduces no new custodial trust assumptions beyond standard BTC key management.

# Structural Guarantees

**Safebox** provides three core guarantees:

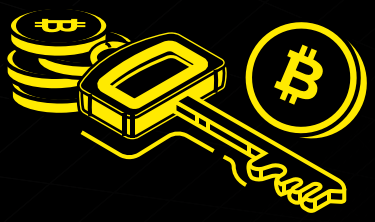
1. Immutability of Lock
   The BTC cannot be moved before the locktime, under any conditions.
2. Exclusive Spending Authority
   Only institutional signatures can release funds after maturity.
3. Isolation from Yield or Protocol Logic
   Yield distribution cannot affect or interact with the timelocked BTC.

Correctness depends solely on:

- Bitcoin's CLTV enforcement
- The institution's ability to sign
- The correctness of the audited script

**Safebox** therefore exposes a narrow, well-defined risk surface that institutions can independently validate through lifecycle testing.

GOAT NETWORK



# SAFEBOX

## Structure & Risk Brief