Beta Staker Consolidation & Fund Migration Plan

This document serves as the definitive technical guide for consolidating the beta staker operator set. It outlines the strategic objectives, key stakeholders, and the end-to-end process required to successfully migrate network funds and transition to a new, more streamlined operator model.

1.1 Core Objectives

The primary goals of this initiative are to enhance the network's operational efficiency and security posture. Successful execution will result in the following outcomes:

- **Operator Consolidation:** Consolidate the beta staker cohort by removing 16 redundant nodes, resulting in 4 primary operators (one per provider), as part of a transition to a total set of 20 staying operators.
- **Staking Mechanism Upgrade:** Deploy the new `Allowlist` smart contract to replace the legacy T-token staking model for operator authorization.
- **Core Contract Enhancement:** Upgrade the `WalletRegistry` contract to integrate with and enforce the new `Allowlist` mechanism.
- **Infrastructure Refresh:** Generate 10 new, secure wallets operated exclusively by the new, consolidated operator set.
- **Fund Migration:** Securely transfer over 6,000 BTC from legacy wallets to the newly created wallets using the established RFC-12 `MovingFunds` protocol.
- **System Decommissioning:** Decommission 17 deprecated operator nodes once all funds are successfully migrated and verified.

1.2 Key Stakeholders and Responsibilities

The successful execution of this plan requires coordinated action from several key groups, each with distinct responsibilities.

Actor	Responsibilities
Council Multisig	Propose and approve the `WalletRegistry` upgrade via the Timelock, execute governance parameter changes, and approve the final `Allowlist` configuration.
Engineering Team	Manage smart contract deployment, execute Timelock-gated upgrades, and provide continuous system monitoring throughout the migration.
Staying Operators	Ensure their nodes are prepared to automatically join the new sortition pool, participate in the Distributed Key Generation (DKG) for new wallets, and maintain node uptime.
Deprecated Operators	Continue to operate their nodes to provide essential signing services for fund movements from existing wallets, and decommission their nodes only after the migration is complete.
Maintainers/Citizen42	Trigger the creation of new wallets and monitor the progress of `MovingFunds` transactions.

The following sections detail the necessary preparatory work that must be completed before the migration can begin.

2.0 Prerequisites and Preparation

The success of this migration hinges on the meticulous completion of all prerequisite tasks. These foundational actions, which include smart contract development, operator coordination, and precise on-chain configuration, are critical for ensuring a secure and seamless transition.

2.1 Smart Contract Readiness

All required smart contract development and security audits have been finalized, ensuring the core components are ready for deployment.

Task	Owner	Status
Implement `Allowlist` contract	Engineering	DONE
Implement `WalletRegistry` upgrade (integrate Allowlist)	Engineering	DONE
Audit `Allowlist` + `WalletRegistry` changes	Thesis Defense	DONE

2.2 Operator Coordination

Communication and confirmation with all affected network operators are essential to guarantee a smooth transition and the availability of signing services.

Task	Owner	Status
Notify all 37 operators of consolidation plan	Ops Team	Required
Confirm 17 deprecated operators will stay online during migration	Ops Team	DONE

2.3 Allowlist Configuration

The new `Allowlist` contract will be configured with a precise set of 20 operators, composed of 4 Beta Stakers, 8 Professional Operators, and 8 Community Operators. The composition is as follows:

4 Beta Stakers

Provider	Operator Address	Weight
Boar	`0xFfB804c2De78576Ad011f68a7DF63d739b8C8155`	62765213
Staked	`0xf401aaE8C639eB1638Fd99B90EaE8a4C54F9894D`	38416500
P2P	`0xB074A3B960f29a1448a2Dd4De95210ca492c18d4`	20000000
NuCo	`0xD7F138ccF194ca2F49c28870b3b5F556B57Fb8b7`	20000000

8 Professional Operators

- Colossus Digital
- Sub7
- Delight Labs
- Global Stake
- Republic Crypto
- Liquify
- Stake Capital
- Ponkila

8 Community Operators

- Shoegazer
- James Campbell
- Evandro Saturnino
- The Egg
- 0x10003

- Vict0x
- Blitmore
- BitFwd

Explicitly Excluded Operators

The following 17 deprecated operators will **not** be added to the `Allowlist`:

- 5 Staked operators (redundant nodes)
- 5 P2P operators (redundant nodes)
- 5 Boar operators (redundant nodes)
- 1 NuCo operator (redundant node)
- 1 Kiln operator (leaving the operator pool)

With these preparations complete, the first active phase of the migration can be initiated.

3.0 Phase 1: WalletRegistry Upgrade

This phase represents a core architectural evolution of the system. The `WalletRegistry` contract will be upgraded to shift from the legacy T-token-based staking mechanism to the new `Allowlist` for operator selection in all future Distributed Key Generation (DKG) events.

3.1 Upgrade Objectives

The primary outcomes of this upgrade are:

1. Integrate the `WalletRegistry` with the newly deployed `Allowlist` contract.
2. Utilize the weights defined in the `Allowlist` for operator selection instead of T-token stake.
3. Restrict participation in new DKG events exclusively to operators included in the `Allowlist`.

3.2 Timelock Security Protocol

On Mainnet, this critical upgrade is subject to a mandatory 24-hour Timelock for security. The upgrade path involves several key contracts.

Contract	Address	Role
`WalletRegistry (Proxy)`	`0x46d52E41C2F300BC82217Ce22b920c34995204eb`	The target contract to be upgraded.
`WalletRegistry ProxyAdmin`	`0x7affa05f726d293eb1193807a91617318292008e`	The contract that executes the upgrade logic.
`Timelock`	`0x92f2d8b72a7F6a551Be60b9aa4194248E9B4913D`	**Owns the ProxyAdmin** and enforces a **24h delay** between proposing and executing an upgrade.
`Council Multisig`	`0x9F6e831c8F8939DC0C830C6e492e7cEf4f9C2F5f`	The entity authorized to propose upgrades to the Timelock.

Note: For faster iteration during testing on the Sepolia testnet, this Timelock is bypassed, and upgrades are executed directly by an authorized EOA (Externally Owned Account).

Immediately following this upgrade, the system's operator pool will adjust automatically based on the new `Allowlist` rules.

4.0 Phase 2: Governance Parameter Updates

The first active step of the migration involves updating key governance parameters via the `BridgeGovernance` contract. These changes are essential to enable the large-scale, rapid fund movements required to consolidate wallet balances efficiently.

4.1 Required Parameter Changes

The following three parameters must be updated to facilitate the migration.

Parameter	Current Value	New Value	Rationale
`walletMaxBtcTransfer`	10 BTC	1,500 BTC	To allow the full balance of large wallets to be transferred in a single transaction.
`walletMaxAge`	182 days	97 days	To ensure that all target wallets, including 4 that are younger than 182 days, are eligible for the `MovingFunds` operation.
`walletCreationPeriod`	14 days	1 day	To accelerate the creation of the 10 new wallets, reducing the process from months to approximately 10 days.

4.2 Execution Protocol

The Council Multisig will execute these changes according to a specific protocol.

- **Target Contract:** `NEW BridgeGovernance` at `0xcBCFA3eb5E067173b262ACe62f9dD87f1D2Cc0Cf`.
- **Timeline:** The process involves two steps:
  - **Day 0:** The Council Multisig initiates all three parameter changes.
  - **Day 2:** The changes can be finalized and take effect.
- **Security Delay:** This process utilizes a **built-in 48-hour delay** within the `BridgeGovernance` contract itself. No external Timelock contract is required.

- **Parallel Execution:** All three parameter changes can be initiated and finalized in parallel, sharing the same 48-hour waiting period.

With these time-locked governance changes underway, the system prepares for the immediate post-upgrade effects on the operator pool.

5.0 Phase 3: Automated Operator Sortition Pool Onboarding

The upgrade of the `WalletRegistry` triggers an immediate and fully automated realignment of the sortition pool. Operators' keep-core clients will automatically re-evaluate their eligibility based on the new `Allowlist`, resulting in a seamless transition for staying operators and the automatic removal of those being deprecated. This automation is a critical feature of the keep-core client's `MonitorPool` function, removing the need for any manual `joinSortitionPool` transactions from operators and ensuring a deterministic and timely transition.

5.1 Automated Pool Management Logic

The system's behavior is deterministic, based on an operator's presence in the `Allowlist`.

Operator Type	In Allowlist?	`eligibleStake()` Result	Sortition Pool Action
20 STAYING	Yes	Returns `Allowlist` weight	Weight is automatically updated in the pool.
17 DEPRECATED	**No**	Returns `0` (default)	Operator is **automatically removed** from the pool.

The technical mechanism for removing deprecated operators is straightforward: their absence from the `Allowlist` causes the `authorizedStake()` function to return 0. The `SortitionPool` contract interprets this zero-stake value as ineligibility and automatically removes them. This process typically completes within 6 hours as each operator's client runs its periodic `MonitorPool` check.

5.2 Critical Impact Assessment

It is crucial to understand that **removal from the sortition pool does NOT affect the ability of deprecated operators to sign for existing wallets**. Their key shares are stored locally on their nodes, and their participation is still required to authorize the `MovingFunds` transactions from old wallets. Their exclusion only prevents them from being selected to participate in the creation of *new* wallets.

With the contracts upgraded and the operator pool correctly configured, the system is now ready for the creation of new wallets.

6.0 Phase 4: New Wallet Creation

Once the governance parameters are finalized and the `WalletRegistry` is upgraded, the system is prepared to generate the new, secure wallets that will receive the migrated funds. This phase initiates the creation of the target infrastructure for the consolidated operator set.

6.1 Process Initiation and DKG

- **Trigger:** The creation of new wallets will be initiated by the `Wallet maintainer (Citizen42)`.
- **Automation:** The subsequent Distributed Key Generation (DKG) process is fully automatic. The system will select a group of operators to create the new wallet, and this selection will be drawn exclusively from the 20 operators present in the `Allowlist`.

6.2 Creation Cadence and Timeline

The timeline for this phase is governed by a key system parameter.

- **Constraint:** With the `walletCreationPeriod` governance parameter set to 1 day, only one new wallet can be created per 24-hour period.
- **Estimated Timeline:** The creation of all 10 required wallets is estimated to take approximately 10 days. A buffer is included, bringing the total expected timeline for this phase to 12-14 days.

As each new wallet is created, the project can proceed to the primary activity of migrating funds into it.

7.0 Phase 6: Main Fund Migration

This phase represents the core activity of the project: moving over 6,000 BTC from older, geographically diverse wallets into the 10 newly created, consolidated wallets. This is a large-scale, iterative process requiring careful monitoring and execution.

7.1 Migration Scope: 10 Largest Wallets

The migration will target the 10 largest legacy wallets, which collectively hold the vast majority of the network's funds.

Wallet Address (truncated)	BTC Balance
`0xf45bd1a0...`	1,177
`0x294887b1...`	1,012
`0x18f9e26d...`	705
`0x71c3ad85...`	694
`0x58efc64e...`	600
`0x39925919...`	518
`0x7abdee3c...`	469
`0x79d5b6c9...`	397

`0x0854813f...`	363
`0x9ef8bdf3...`	275
Total to Migrate	**~6,210 BTC**

7.2 Migration Process

The fund movement will be conducted iteratively. As each new wallet is created on a daily basis, the full balance from one of the old wallets will be migrated into it. This process utilizes the RFC-12 `MovingFunds` mechanism, which requires signatures from the operators of the old wallet (including the deprecated operators).

Upon the successful transfer of all funds, the project will move into the final verification and validation stage.

8.0 Phase 7: Post-Migration Verification and System Validation

The post-migration verification phase is dedicated to rigorously confirming that all funds have been successfully moved, the new system state is correct, and the bridge is fully functional. This validation is a critical prerequisite before any legacy infrastructure is decommissioned.

8.1 Final State Verification Checklist

The following criteria must be met to confirm the success of the migration:

- Confirm that all 44 old wallets are in a `Closed` state on-chain.
- Verify that all over 6,000 BTC resides securely in the 10 new wallets.
- Validate that the new wallets are exclusively operated by the 20 staying operators as defined in the `Allowlist`.
- Confirm that none of the 17 deprecated operators are members of any new wallet's operator set.
- Perform a system health check to ensure the bridge is functioning normally for standard user deposits and redemptions.
- Conduct a final sweep to ensure there are no stuck transactions or pending `MovingFunds` operations.

Once all items on this checklist are successfully verified, the project can proceed to the final cleanup activities.

9.0 Phase 8: Cleanup and Decommissioning

This final phase of the project involves the orderly decommissioning of the 17 deprecated operator nodes, completing the transition to the new, consolidated operator set.

9.1 Decommissioning Protocol

A critical prerequisite governs this phase to ensure the absolute security of funds during the transition.

- **Condition:** Decommissioning can only occur **after** all old wallets are confirmed to be in the `Closed` state.
- **Rationale:** The 17 deprecated operators hold essential key shares for the old wallets. They must remain online to provide signatures for the `MovingFunds` transactions until the migration of every legacy wallet is fully complete and verified.

The successful decommissioning of these nodes marks the final milestone and completion of the entire consolidation and migration project.

10.0 Integrated Project Timeline and Critical Path

This section synthesizes all project phases into a cohesive, day-by-day timeline. This schedule highlights the key dependencies between phases and defines the critical path for successful project execution.

10.1 Phased Execution Timeline

Phase	Key Actions	Lead Actor(s)	Project Day(s)
2. Governance Changes	Initiate and finalize parameter changes (`walletMaxBtcTransfer`, `walletMaxAge`, `walletCreationPeriod`).	Council Multisig	Day 0 → Day 2
1. `WalletRegistry` Upgrade	Deploy `Allowlist`, schedule upgrade in Timelock, and execute after 24h delay.	Council, Engineering	Day 2 → Day 3
3. Operator Onboarding	Operators automatically join/leave sortition pool based on `Allowlist` status.	Operators (Automated)	Day 3
4. New Wallet Creation	Trigger the DKG process for 10 new wallets, one per day.	Maintainers, Operators	Day 3 → Day 14
6. Main Fund Migration	Execute and monitor `MovingFunds` transactions from old wallets to new wallets as they are created.	Engineering, Operators	Day 4 → Day 14
7. Post-Migration Verification	Verify all funds are moved, old wallets are closed, and system is healthy.	Engineering	Day 14+

8. Cleanup & Decommissioning	Deprecated operators shut down their nodes after confirming all old wallets are closed.	Operators	Day 14+

10.2 Critical Path Analysis

The project's minimum duration is governed by three sequential, time-gated on-chain actions. The creation of the first new wallet is blocked until both the governance parameter updates and the `WalletRegistry` upgrade are finalized.

Action	Mandatory Delay	Timing
Governance Parameter Changes	**48 hours**	Begins on Day 0, completes on Day 2.
`WalletRegistry` Upgrade	**24 hours**	Scheduled on Day 2, completes on Day 3.
First DKG for New Wallet	~1-2 hours	Can begin on Day 3, after the above actions are complete.

11.0 Appendix: Contract Addresses

This appendix provides a centralized reference for all relevant smart contract addresses on both the Ethereum Mainnet and the Sepolia testnet.

11.1 Mainnet Addresses

Contract	Address	Notes
`Bridge`	`0x5e4861a80B55f035D899f66772117F00FA0E8e7B`	
`BridgeGovernance (NEW)`	`0xcBCFA3eb5E067173b262ACe62f9dD87f1D2Cc0Cf`	Has a 48-hour built-in delay for parameter changes.
`WalletRegistry`	`0x46d52E41C2F300BC82217Ce22b920c34995204eb`	
`WalletRegistry ProxyAdmin`	`0x7affa05f726d293eb1193807a91617318292008e`	Owned by the `Timelock`.
`Council Multisig (6-of-9)`	`0x9F6e831c8F8939DC0C830C6e492e7cEf4f9C2F5f`	
`Timelock`	`0x92f2d8b72a7F6a551Be60b9aa4194248E9B4913D`	**Enforces a 24h delay; owns the `WalletRegistry ProxyAdmin`.**

`ProxyAdmin (Bridge)`	`0x16A76d3cd3C1e3CE843C6680d6B37E9116b5C706`	Owned by the `Timelock`.
`Allowlist`	TBD	To be deployed during the migration process.