**Synergy Network Address Formatting Specification**

**1. Address & Key Examples**

This section provides examples of different address and key formats used in the Synergy Network.

**1.1 Wallet Address Example**

sYnQ1zxy8qhj4j59xp5lwkwpd5qws9aygz8pl9m3kmjx3

* Generated using **Bech32m encoding** with a random sYnQ, sYnU, sYnX, or sYnZ prefix.

**1.2 Public Key Example**

6fd47f3a8dca7e47c5f9a9128b3a45dc1f91de789da3e69f54a8a13fd0a937a2

* Derived from **Dilithium-3 (CRYSTALS PQC Standard)** for post-quantum security.

**1.3 Private Key Example**

d14c8d2e5b3f7a9a0f2b3c8d1e2f3a7c6d5e4f2a1b9c3d7e8a0f1b2c3d4e5f6a

* Securely stored and used for signing transactions.

**1.4 Smart Contract Address Example**

sYnS-CONTRACT-8a7b5c9f3d6e1a2b4c7d8f9e0a5b6c3d

* Unique contract address format with “sYnS-CONTRACT-” or “sYnC-CONTRACT-” prefix.

**1.5 Transaction Identifier Example**

sYnT-tXn-abcdef1234567890abcdef1234567890

* Prefixed with “sYnT-tXn-” or “sYnA-tXn-” to denote transaction records.

**1.6 Synergy Naming System (SNS) Example**

alice.syn

* Human-readable name resolving to a Synergy Network address.

**2. Overview**

This document defines the unique address format for the Synergy Network. The goal is to create an address system that:

* Differentiates from existing blockchains (e.g., Ethereum, Bitcoin, Solana).
* Supports **human-readable identifiers**.
* Ensures compatibility with **quantum-safe cryptography**.
* Allows seamless cross-chain transactions through **Universal Meta-Addresses (UMA)**.
* Uses a **randomly chosen prefix** between sYnQ, sYnU, sYnX, or sYnZ.

**3. Address Structure**

Each Synergy Network address will have the following structure:

sYnX-XXXXXXXXXXXXXXX

Where:

* sYnQ, sYnU, sYnX, or sYnZ → Randomly chosen **network prefix** (ensuring address uniqueness).
* X...X → Encoded hash of the public key (varies based on encoding scheme).

**Encoding Method: Bech32m (Selected)**

* Example: sYnQ1zxy8qhj4j59xp5lwkwpd5qws9aygz8pl9m3kmjx3
* **Character Length:** 41 characters (fixed, if possible; otherwise, dynamically selected between 30-42).
* **Pros:** Error detection, case-insensitive, QR code friendly, optimized for SegWit and newer blockchain standards.

**Address Length Considerations:**

* **Target: 41 characters** (if adjustable within Bech32m standard).
* **High entropy keyspace** to prevent address collisions.

**4. Address Generation Process**

1. **Generate a Quantum-Safe Key Pair**
   * Default: **Dilithium-3 (CRYSTALS PQC Standard)**
   * Alternative: **Kyber (for hybrid encryption key generation)**
2. **Compute Address Hash**
   * Hash public key using **SHA3-256 or BLAKE3** (quantum-resistant choice).
   * Encode with **Bech32m (fixed 41-character length, if configurable)**.
3. **Assign a Prefix (sYnQ, sYnU, sYnX, or sYnZ) Randomly**
   * Use a **secure random function** to choose prefix.
   * Append encoded hash to finalize address.

**5. Address Use Cases**

**5.1 Wallet Addresses**

* Standard user accounts.
* Supports **quantum-resistant signing**.

**5.2 Token Contract Addresses**

* Special format: sYnS-CONTRACT-XXXXX… OR sYnC-CONTRACT—XXXXX…
* Prefixed to distinguish from regular addresses.

**5.3 Transaction Identifiers**

* Uses a similar encoding but prefixed with “sYnC-Txn-” or “sYnT-tXn-”.
* Example: sYnt-TxN-abcdef123456.

**6. Cross-Chain Compatibility & Universal Meta-Addresses (UMA)**

To enable **atomic swaps and interoperability**, addresses must:

* Map **Synergy addresses** to external chains (Ethereum, Bitcoin, Solana, etc.).
* Use **threshold cryptography (FROST/TSS)** for cross-chain transactions.
* Support **Synergy Naming System (SNS)** for human-readable names (alice.syn).

**7. Next Steps**

* Implement Bech32m encoding in the Synergy Wallet with 41-character target length.
* Develop UMA resolution logic for cross-chain compatibility.
* Finalize security benchmarking for quantum-safe cryptography.

This document will be updated as the address format evolves based on implementation and testing feedback.