Protocol 22 Announcement

In use since 5 Dec 2024.

Here's what's new in Protocol 22: The two new CAPs introduced in Protocol 22 are CAP-0058 and CAP-0059. Let’s delve into these a bit more so we know what to expect with Protocol 22.

CAP-0058: Constructors for Soroban contracts

CAP-0058 introduces constructors for Stellar smart contracts, simplifying the initialization process. A constructor is a special function that runs automatically when a contract is deployed, setting up the contract's initial state, such as assigning values to variables or configuring permissions. This ensures the contract starts in a valid and usable state with all necessary data. Currently, Stellar developers must manually include additional logic or checks to handle initialization, but with constructors, this process becomes automatic upon deployment.

Implementing constructors makes contracts more efficient by reducing their size, lowering CPU usage during execution, and minimizing storage requirements. It also enhances security by making it harder for developers to unintentionally expose their contracts to front-running (an attack where someone uses their knowledge of a pending transaction to gain an unfair advantage) during initialization.

Constructors are supported in other smart contract frameworks and languages (like Solidity in Ethereum), so this CAP helps to align Stellar with other networks and eases friction in developer onboarding.

Read more technical details about CAP-0058: Constructors for Soroban contracts on GitHub.

CAP-0059: Host functions for BLS12-381

BLS12-381 is a widely adopted type of elliptic curve used in cryptography, known for its efficiency and strong 128-bit security. This curve enables pairing-based cryptography, which enables advanced cryptographic operations like zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge). zk-SNARKs allow a prover to demonstrate to a verifier that they possess certain information or that a statement is true without revealing the underlying information.

A key use case for zk-SNARKs is anonymous login, such as zkLogin or zkEmail, where users can authenticate using real-world Web2 identities (like their Gmail address) to sign a transaction, where the transaction is verified on-chain without exposing the user's actual email address. This capability supports a more seamless onboarding experience into Web3 for Web2 users.

Operations involving the BLS12-381 elliptic curve are computationally intensive, making it difficult to implement them directly into a smart contract. To address this, these operations are implemented in the Soroban host environment, allowing smart contracts to leverage them without having to handle the complex computations themselves.

Learn more about BLS12-381 on Stellar, including a deep dive into the 11 new host functions on GitHub

.