AMI Names Terminology¶

AMI Name¶

AMI Name is Cosmos’ first Self-Sovereign Identity which is permanently allocated to a user and does not expire ever, unlike other Naming Systems (like DNS, ENS, etc.) which are only temporarily allocated. AMI names are globally resolvable as Decentralized Identifiers (DIDs), a pioneer Web3 technology, which means your identity will be accessible, and verifiable, in a purely decentralized way, from anywhere in the globe. AMI names also implement account abstraction, which means you will never be ‘locked out’ of your identity control, due to forgetting password or losing one of the keys, making them the only singular identity you will ever require to manage all your assets. Hence the core tenets of AMI Names are:

Self-Sovereign Identity¶

you create your own identity without anyone’s permission or authorization.

Permanently Allocated¶

you have ‘true perpetual control’ of your identity which never expires or de-allocates, unlike Naming Systems (e.g., DNS, ENS) which are temporarily allocated, and de-allocated.

DID Based¶

globally accessible identity from any web3 / web2 system, resolved as a DID document, and verifiable in a decentralized way, instilling “zero downtime”.

Account Abstraction¶

your identity is controlled by you using multiple ‘provision addresses’ as keys, making sure you will never be ‘locked out’ and can always recover in case you lose a key.

SSI¶

Self Sovereign Identity is a model for managing digital identities in which individuals or businesses have sole control over their accounts and personal data. Individuals with self-sovereign identity can store their data on their devices and provide it for verification and transactions without the need to rely upon a central repository of data. With self-sovereign identity, users have complete control over how they create their identity, and how their personal information is kept and used.

DID¶

Decentralized identifiers (DIDs) are a new type of identifier that enables verifiable, decentralized digital identity. A DID refers to any subject (e.g., a person, organization, thing, data model, abstract entity, etc.) as determined by the controller of the DID. In contrast to typical, federated identifiers, DIDs have been designed so that they may be decoupled from centralized registries, identity providers, and certificate authorities. DIDs are globally unique, and resolvable as a DID Document, which means one can access and verify their identity from anywhere, with no ‘down time’.

Account Abstraction¶

Account Abstraction refers to de-coupling the association of your account from a single ‘blockchain address’, and making it an independently existing entity, where one can allocate multiple “blockchain address’ based keys to operate the account. This dissociates from the idea of ‘account’ that exists in blockchains like Ethereum, where “blockchain address” IS the account itself, and makes it into a separate entity, where “blockchain addresses” can be added / removed as ‘provision keys’. Account abstraction is now getting popular in Ethereum as well, by introduction of ERC-4337, a standard for Account Abstraction.

Provision Address / Provision Address Key¶

Provision address is your blockchain address that you use as the ‘key’ to operate your Account. This address will be used to create transactions that can transfer assets to and from your account or perform other operations. You can allocate multiple provision addresses to your account, adding an extra layer of security to your single source of truth (SSI). If you lose the private key of one provision address, you can use another to replace the compromised one, preventing any loss of control. This de-couples the account from being allocated to a single blockchain address, controlled by a single private key.

Multiple Rotatable Keys¶

Multiple Rotatable Keys (MRK) is a concept of Account abstraction where multiple provision addresses can be allocated to an account as ‘keys’ which control the account. These keys can then be ‘rotated’ in and out of operation. One can rotate out their keys in regular intervals so as to provide extra security in case a key gets compromised. This is similar to changing your passwords regularly as a best practice for security, in web2.

Previous Overview Next AssetMantle Whitepaper