Token ARCs

Overview of token-related ARCs

September 20, 2022 Joe Polny

Algorand™ FOUNDATION

Why ARCs?

- Wallets and Dapps need to know how to view data
  - Standards make this easy to implement
- There are different use cases of ASAs and tokens
  - Different uses require different standards
- Native ASA fefatures is limited
  - Can be expanded with ARCs

ASA Fields

Field	Type	Description
Total	uint64	The total number of base units of the asset to create. This number cannot be changed.
Decimals	uint32	The number of digits to use after the decimal point when displaying the asset.
DefaultFrozen	bool	True to freeze holdings for this asset by default.
UnitName	string	The name of a unit of this asset. Supplied on creation. Example: USDT
AssetName	string	The name of the asset. Supplied on creation.Example: Tether
URL	string	Specifies a URL where more information about the asset can be retrieved.
MetaDataHash	[]byte	This field is intended to be a 32-byte hash of some metadata that is relevant to your asset and/or asset holders.

ASA Address Fields

Field	Type	Description
ManagerAddr	Address	The address of the account that can manage the configuration of the asset and destroy it.
ReserveAddr	Address	The address of the account that holds the reserve (non-minted) units of the asset.
FreezeAddr	Address	The address of the account used to freeze holdings of this asset.
ClawbackAddr	Address	The address of the account that can clawback holdings of this asset.

ARC-0003

- General standard for displaying ASA information
- Uses off-chain JSON metadafile
  - Often stored on IPFS
- Defines standards for on-chain configuration fields
- Largely similar to ERC 1155 and Open Sea standards

ARC-0069

- Uses note field for metadata
- Limitations
  - Metadata limited to 1KB due to note size limit
  - Indexer must be used to query asset config transaction
  - Image is immutable

ARC-0019

- Uses reserve address field for point to IPFS metadata
- Metadata has no size restrictions
- Only algod is needed to get metadata
- Image is mutable
- Limitation
    - Can't use reserve address field as intended

ARC-0020

- "Smart ASAs" via smart contract
- Can follow some existing ARCs, such as ARC3
- Allows more functionality than ASAs
  - Transfer logic
  - Mutability
- Should only be used when ASAs aren't an option
  - Smart contracts have much more security considerations

ARC-0020 Interface

- Has ABI methods for performing same operations you can on ASAs
  - Create
  - Config
  - Freeze
  - Destroy
  - Transfer (and clawback)
- Adds the following methods ontop of ASAs
  - Asset Freeze vs Account Freeze
  - Get circulating supply

ARC-0018

- Leverages ARC-0020 (smart contracts)
- On-chain royalty enforcement
  - Asset cannot be moved unless royalties are paid
  - OR the admin calls the royalty-free move function
- Enables royalties for payments in ALGO or ASAs
- Compatible with secondary marketplaces

How to decide?

- Start with ARC-0003
- Need mutability?
  - Need reserve address?
    - **Note:** reserve address has no affect at protocol level
    - ARC-0069
  - Otherwise
    - ARC-0019
- Need functionality ASAs don't provide?
  - Need enforced royalties on sales?
    - ARC-0018
  - Otherwise
    - ARC-0020