

Classy DIDs

Shaun Conway (ixo)

Abstract

Not all DIDs are created equal. This is a feature, not a judgement. Decentralized Identifiers are being differentiated for an expanding range of contexts and use-cases.

DID Methods differentiate decentralised identifiers by how these are created, resolved, updated, and deactivated.

We propose that **DID Classes** can further differentiate decentralised identifiers by their patterns of properties and behaviours.

Class differentiation requires a mechanism for DIDs to inherit class-level properties. Could this be achieved by using DIDs in Linked-data Contexts?

DIDs for Digital Assets

We have developed the Interchain Identifiers (IID) specification as a DID Method for identifying and resolving information about digital assets, such as non-fungible tokens, that are located within blockchain namespaces.

There are many uniquely identifiable types and classes of digital assets.

As an example, let's consider Non-Fungible Tokens of Type **ERC-721** that represent one metric tonne of verified carbon emission reduction (VER). This token instance is part of a uniquely identified set of VER tokens, such as the "Vintage 2022 Carbon Credits Collection".

The Vintage 2022 Collection is identified as belonging to the Class of Type **VER** denomination tokens that share common properties and behaviours.

As an example: All tokens of this class may be *Retired*. This behaviour is invoked to voluntarily offset Carbon Emissions in Carbon Accounting.

Being of Type **ERC-721**, VER non-fungible tokens have properties that comply with a defined application-blockchain interface standard.

From this example, we see that identifiers may fit within a hierarchy of classes that inherit specific Types of properties and behaviours. Using DIDs and Interchain Identifiers for these assets, we should be able to represent this Class hierarchy as a semantic Linked-data graph.

Representing classes of DIDs

Given that DIDs are URIs, we should be able to use DIDs as Linked-data contexts.

For example:

```
"@context":{"class":"did:example:abc123"}
```

A digital asset, such as a Non-Fungible Token, may be identified in this way as an instance of the class. How this inheritance is interpreted by Client applications will be defined in the DID Method for this canonical class of DIDs.

Class-level controls

In our implementation of digital assets using Interchain Identifiers, the DID **controller** is by default the owner of the digital asset, and the controller property is programatically updated whenever there is a transfer of asset ownership.

By implication, the every asset class has a controller with the capability to make class-level changes to the control mechanisms, services, linked resources and rights accorded to all the assets in this class. The scope of these controls may be constrained by Object Capability systems, for instance using Delegated Authorisation Capabilities (zCaps).

We believe this versatile pattern of class-level control supports a wide variety of DID use-cases.

Class-level properties and behaviours

The DID-Core properties may be extended with further properties, as we have done for Interchain Identifiers to add **linkedResource** and **accordedRight** as properties that are extremely useful for digital assets.

Class-level properties, such as a specific **service** that is available to all children of a class, are inherited from the parent DID document.

The cryptographic controls defined by **verificationMethod** and verification relationship properties, in combination with properties such as **accordedRight**, determine the behaviour of the class – essentially “what you can do” with the identified digital asset.

Interim conclusion and next steps

Using Linked-data Contexts to establish the hierarchies of parent and child DID classes may improve the semantic Type definitions of DIDs, according to their property sets and behaviours. Class differentiation of DIDs in this way could expand their utility.

We would like to further explore general patterns of using DIDs as Linked-Data Contexts and as a specific solution for identifying hierarchical classes of identifiers.

Advancing this approach may directly support our protocol for identifying classes of digital assets that represent verified social, economic, environmental

and climate outcomes, and rights to the real-world assets that are used in the production of these outcomes.