Tezos User Verification System

Project Overview

This project was developed as an entry for the Game On! Tezos – Self-Sovereign Identity Bounty. The aim of the project was to implement a system for creating and managing user identities which can be used across a variety of apps and games. This was accomplished through the creation of a back-end system which mints a non-fungible token for the user to represent their identity. The NFT smart contract also contains entry points which allow the modification of token metadata. In this way, apps and games can request a user's permission to write data to their identity, allowing the NFT to serve as a collection of profile data for the applications the user interacts with. This system front-end provides the user with complete control over their online identity by providing only them with the permission to write and delete data. In addition to the back-end and front-end prototypes, a separate game was developed as an example of how an online application can make use of a player's on-chain identity.

Implementation Description

Following are descriptions of the state of the back-end, front-end, and external game at the time of project submission.

Back-End

The back-end prototype consists of an NFT smart contract currently deployed to the Hangzhounet test net. The contract address is "KT1C5rF5eWfVm6TrL5wze3E8vWhnqYE5trN6". The entry points which are currently implemented allow the minting of new NFTs, the modification of NFT metadata, the transfer of NFTs between users, and the checking of user balances. These entry points provide the functionality needed for a working system prototype. Users are first able to mint a new NFT identity via the font-end site. Subsequently, external applications can request permission to make calls to contract entry points to add to the metadata of their NFT.

Each NFT contains unique metadata in the form of a list of data entries. Data entries consist of a title string, a map of strings to natural numbers, and a map of strings to strings. New entries can be added through an existing entry point. These data types were selected because they provide useful functionality to applications such as storing username, high score, or appearance preferences while still being simple enough to implement as a part of this prototype.

Front-End

The front-end prototype consists of a web application implemented using JavaScript and React. The site is hosted using Amazon Web Services and can be found here. The interface allows the user to connect a software wallet to associate with their identity. Upon wallet connection, the application parses the contract storage and checks if the given account already has an existing identity. If an identity is found, the contained metadata is automatically displayed to the user. If no identity is found, the user is presented with the option to mint a new NFT to serve as their identity.

External Game

The external game, titled Tez-Snake, is an implementation of snake using JavaScript and React and several purposes for the project. First, it is an example and proof-of-concept of the use of this system by

third-party application. The game requires the user to connect a software wallet associated with an account which owns an NFT identity. If their account does not own an identity, they are directed to the system front-end to create one. After each round of the game, the user is presented with the option to save their high score, username, and appearance preferences to their account. The appearance preferences consist of a hex color value which is used as the color of the snake while playing the game.

In addition to serving as an example of how this system can be used by developers, Tez-Snake aims to be an example for players of how they can benefit from using an on-chain identity. The primary benefit which the game presents to the player is the ability to have a user profile while maintaining complete transparency of how their information is stored. Rather than creating an account which requires providing personal information such as an email address, the user simply provides their public key. Moreover, their profile data is stored entirely in the metadata of their NFT identity. This way they can see what information the game has access to and can choose to delete this data from their identity at any time. In this way, the user is granted all the benefits of a traditional user profile while preventing the need to share personal information.

Backlog for Project Completion

The most important feature missing from this prototype is the ability to easily burn an NFT or remove an entry from an NFT's metadata. Currently, entries can only be removed by overwriting with an empty entry of the same title. This is not ideal, and so the front-end interface does not offer the user the ability to do this. Because the back-end of the project was designed with simple creation and deletion in mind, it should not take more than a day of work to have a functional entry point for removing entries. The same is true for burning NFTs, for which an entry point has been implemented but has not been thoroughly tested or included in the front-end interface. These features are only absent from this prototype due to time constraints.

Although the deletion of NFTs and data entries would complete the project as described in the initial proposal, there are many features which can be implemented in the future to continue development. First, the project would benefit greatly from design changes aimed at making integration with existing web applications easier. When designing the front-end interface and the Tez-Snake game, the goal was to create the game entirely on its own, then incorporate the Tezos User Verification System in as simple a way as possible. Although this was partially achieved, strong knowledge of the design of the front-end interface is still required to effectively incorporate the system into an existing project. With some design changes, it could be made very simple for developers to begin accessing and storing user information using their NFT profiles. This would greatly increase the likelihood that this system becomes widely adopted by developers.

Another feature which would greatly improve the project is the option to use asymmetric encryption. Currently, when a user agrees to let an application write data to their NFT identity, that data is made public in the storage of the smart contract. This is not a problem when storing usernames and high scores. However, it prevents the user from being able to share personal information, such as an email address, with one application while otherwise keeping the information hidden. Asymmetric encryption can provide a solution for this problem by keeping all data encrypted and only allowing applications to access information for which they have explicitly been granted access to by the user. This functionality would greatly expand the potential applications of the Tezos User Verification System.