Summary

TZIP-022 describes a generic smart contract interface for resolving names to Tezos addresses and vice versa.

Motivation

Currently, indexers and wallets use nonstandard methods for associating addresses with human-readable names. They include using pre-configured (in some cases hardcoded) lists of names and addresses or making use of TZIP-16 metadata.

This presents some problems:

- Pre-configured lists are hard to maintain and prone to breaking.
- TZIP-16 metadata are published as part of a contract they relate to, which means the names are not globally unique nor authorative.
- Names and addresses for other types of use, like personal wallets, cannot be resolved.

This document proposes a name resolution interface that can be used by all products in the ecosystem to provide users with a consistent experience when mapping names and addresses.

Specification

TZIP-016 compliance

A contract implementing TZIP-022 MUST also implement TZIP-016, identifying itself with `TZIP-022-<git hash>` in the `interfaces` field of the contract metadata:

```
{
  "description": "An example of a TZIP-022 implementation",
  "interfaces": ["TZIP-022-a71d866", ...],
  ...
}
```

The `resolve-name` off-chain view

An implementing contract MUST implement a TZIP-016 off-chain view called `resolve-name`:

	Type (CameLIGO syntax)	Meaning
Parameter	`bytes`	Name to resolve, encoded in UTF-8.
Return value	`resolution_result option`	Resolved information associated with the name, or `None` if no valid information exists.

The `resolve-address` off-chain view

An implementing contract MUST implement a TZIP-016 off-chain view called `resolve-address`:

	Type (CameLIGO syntax)	Meaning
Parameter	`address`	Address to resolve.
Return value	`resolution_result option`	Resolved information associated with the address, or `None` if no valid information exists.

The `resolution_result` type

A resolution result is a right-combed record containing:

	Type (CameLIGO syntax)	Meaning
name	`bytes`	Resolved name, encoded in UTF-8.
address	`address option`	Resolved address, or `None` if the resolved information does not include an address.
data	`(string, bytes) map`	Any additional data, with key being an arbitrary field identifier and value being a UTF-8 encoded string containing any valid JSON.
expiry	`timestamp option`	Expiry timestamp (the first second when the information ceases to be valid) or `None` if the information does not have an expiration.

Note that the returned `expiry` timestamp will never be in the past. If the information has already expired, the return value of `resolve-name` or `resolve-address` must be `None`, as expired information is not considered valid.

The full definition in CameLIGO is:

```
type resolution_result = [@layout:comb] {
    name: bytes;
    address: address option;
    data: (string, bytes) map;
    expiry: timestamp option;
}
```