String API

*** DRAFT ***

Revision History

Revision	Date	Author	Major Changes
1.0	02-Mar-2014	chris@daxcloud.com	Initial draft for Phase 1

Data Model (For Phase 1)

Users

userID: string	dax
password: string	Password123
userGUID: UUID	(128-bit integer)
email: string	chris@daxcloud.com

128-bit UUID is normally represented by unsigned char data[16]

Assets and Services

assetHash: UUID	0d6dce44a124443d845af191c3d1b64
service: int	5
URI: string	https://my.service.com/lookup/blah?asset=%assetHash%
description: string	The parent owner of this asset and all of its derivatives is the USC ETC.
metadata: key/value pair	[“Original”, “True”], [“Color-corrected”, “False”], [“Copyright”, “2014 USC ETC”]

Service Registry

service: int (autogen unique)	5
serviceName: string	Ownership Information
serviceToken: string	OWNER
serviceDescription: string	This service type provides ownership information about the asset. A non-parameter call to the URI will provide more information on how to access this service.

API

The following is a basic set of services to call the String metadata API. While the HTTP call will most likely use a JSON POST, this remains to be architected. We have included an example of the HTTP calli in the login API call. In addition, the return format will need to be determined as well.

Login/Identify User

This is used to login to the String API for the first time. After Phase 1, we will use the Keystone authentication API. It returns a session token to be used for the remaining calls.

```
(userGuid, token) = authenticate(user, password)

https://maid.host.com/login
        POST    {       "user": "dax",
                        "password": "Password123"
                }
```

Change Password

Necessary to change the password. Reset will allow for a new password to be set and emailed to the user's email address.

```
success = changePW(user, oldpassword, new)
success = resetPW(user)
```

Register Asset ID with Service

The primary registration method to associate an asset ID with a service type and a URI for more information. The call takes an optional description for this association to be retrieved as part of the lookup. In addition, publicly available metadata key-value pairs can be registered through the optional metadataPairs structure, which is a JSON-encoded, square-bracketed, quoted list of data.

```
success = associateAsset(assetHash, serviceToken, URI, description, metadataPairs)
```

Register User/Service Provider

To register a user or service provider into the system, provide the userID, password, and contact email. If successful, the system will return the assigned userGUID.

```
userGUID = registerUser(userID, password, email)
```

Register Service Type

To register a service type, provide the human-readable name for the service, a token value, and a longer description. The service ID will be autogenerated.

success = registerService(serviceName, serviceToken, serviceDescription)

Query Registry

https://maid.host.com/queryAsset?hash=0d6dce44a124443d845af191c3d1b64

Response:

```
{
    {
        “service”:”OWNER”,
        “ URI”:” https://my.service.com/lookup/blah?asset=%assetHash%”,
        “description”:”The parent owner of this asset and all of its derivatives is
    the USC ETC.”,
        “metadata”:
            {
                “Original”:”True”,
                “Color-corrected”:“False”,
                “Copyright”:“2014 USC ETC”
            }
    }
}
```

Add Parent/Peer Registry (Phase 2)

TBD

Configure Synchronization (Phase 2)

TBD