

Raport - Tema 5

PrivateSky structure

UserManager Asset:

- public attributes:
 - alias(string:alias): the alias of UserManager
 - users(map): a dictionary that will contain the users with key "userId" and value "user object"
- init:
 - params: alias(string)
 - set the alias and initialize "users" with an empty map
- addUser:
 - params: id(string), user(object)
 - adds the user with key "id" and value "user" in the map, if the key already exists in the map the value is updated
- removeUser:
 - params: id(string)
 - removes from the map the user with key "id"
- listUsers:
 - returns all the users from the dictionary
- getUserById:
 - params: id(string)
 - returns the user with key "id" if exists, otherwise returns Undefined

UserManagement Transaction:

- init:
 - params: alias(string)
 - searches in the blockchain if there exist an Usermanager with alias "alias"
 - initialize the UserManager
 - adds the user manager in the blockchain
- addUser:
 - params: alias(string), user(object)
 - searches in the blockchain for the Usermanager with alias "alias"
 - adds the user to the UserManager
 - commits the changes in the blockchain

- removeUser:
 - params: alias(string), id(string)
 - searches in the blockchain for the Usermanager with alias “alias”
 - removes the user with userId “id” to the UserManager
 - commits the changes in the blockchain
- listUsers:
 - params: alias(string)
 - searches in the blockchain for the Usermanager with alias “alias”
 - returns all the users stored in the Usermanager
- getUserById:
 - params: alias(string), id(string)
 - searches in the blockchain for the Usermanager with alias “alias”
 - returns the user with userId “id” from the UserManager

Client App

Initialisation:

```
const interact = pskclientRequire("interact");
interact.enableRemoteInteractions();
this.ris = interact.createRemoteInteractionSpace('testRemote',
  'http://127.0.0.1:8080', 'local/agent/example');
this.ris.startSwarm('UserManagement', 'init', 'usermanagement').onReturn(() =>
{
  console.log('SWARM started');
});
```

PrivateSky methods calls:

- this.ris.startSwarm('UserManagement', 'listUsers', 'usermanagement')
- this.ris.startSwarm('UserManagement', 'getUserById', 'usermanagement', id)
- this.ris.startSwarm('UserManagement', 'addUser', 'usermanagement', user)
- this.ris.startSwarm('UserManagement', 'removeUser', 'usermanagement', id)