DynSC: Dynamic Smart Contract

Goal

Proof of Concept using **Hyperledger Fabric** and **Hyperledger Composer** to illustrate a research paper about *dynamic smart-contracts for permissioned blockchains*.

We extended our initial use-case on **Convector & Hurley** because **Composer** has been deprecated in the meantime. Moreover, we also did a benchmark study comparing *Sattelite pattern* over **Ethereum** against our solution based on *dynamic parameterization* over **Hyperledger Fabric**.

Usage

About the installation procedures of the tools you should refer to the official documentations:

- NPM
- Hyperledger Fabric
- Hyperledger Composer
- Hyperledger Composer-Playground
- Worldsibu Convector
- Worldsibu Hurley
- Truffle
- Truffle Ganache-CLI
- Ethereum Web3 library

1st version

See *'v1/'*.

Hyperledger Composer Playground

The first possibility, in a development purpose, is to use the *composer-playground* tool to directly import the BNA (or copy-paste the .cto and .js files). Then, you will be able to play and test the smart contract without the Hyperledger Fabric dependencies.

```
$ composer-playground -p 8080
$ firefox http://localhost:8080
```

Hyperledger Fabric and Composer

The second possibility, in a full test purpose, is to use the *composer* tool to deploy the chaincode from the *BNA* over a existing network instance of *Fabric* blockchain.

```
[Fabric] Starts network nodes...
$ docker-compose -f network.yaml up

[Composer] Deploys smart contract over nodes...
$ composer network install --card <card> --archiveFile dyn-sc.bna
```

```
$ composer network start --card <card> --networkName dyn-sc --
networkVersion 0.1.0 -A <Admin> -S <AdminPassword>
$ composer transaction submit --card <card> --data <json_data>
```

2nd version

See *'v2/'*.

Ethereum with Truffle & Ganache

See *'v2/dynsc-eth'*.

Hyperledger Fabric with Convector & Hurley

See *'v2/dynsc-hlf'*.

Paper

Submitted on the **25-10-2019**.

Code snippet

The code available in this repository is developed and tested with *composer-playground*. It is composed of two main files: one model file *'dynsc.cto'* and one script file *'dynsc.js'*. These files are defining a small chaincode based on a simple use-case as a proof of concept. The *'dynsc.bna'* file is including the previous files, this is the smart contract exported by *composer* in a Business Archive Network.