**POC Name**

**Product Master – Registration Use Case**. DLT Technology: Hyperledger fabric

**POC objectives**

To demonstrate a viable model to register an ordinary European mutual fund. We strive to demonstrate the following “concepts” with the chosen technology:

* **Shared Master data**: all relevant parties may share a single golden copy of a product master data, which owner is the issuer
* **Product documentation**: legal and marketing documents may be shared on the platform
* **Registration application process:** an issuer may apply for a fund to be registered after validation by a regulator. This process is country dependent. We demonstrate how several regulators may coexist on the platform and how a single fund may be registered by several regulators.
* **Carried on checks:** we demonstrate that a number of checks may be carried on. In this context, checks are simplified.
* **Consensus protocol:** a fund’s agreement results from a consensus protocol between participating parties. We found interesting to demonstrate how optional parties may be included in the process, such as a securities registration authority (e.g ISIN numbering authority), LEI registration authority and Settlement System Authority (e.g. Euroclear). The POC demonstrates a workable protocol when all parties participate to the platform or when some parties are external
* **Proprietary model extensions:** we demonstrate how the model may be adapted to every regulator’s requirements. In this POC, we shall demonstrate a French FCPE, a Luxembourg umbrella SICAV and more common mutual fund.

We do not demonstrate the following in this POC, and shall address such required items in other dedicated POC’s:

* Data privacy
* Distribution agreements
* PRIIPS, UCITS, MiFID 2 or other distribution & fees, costs, transparency policies

**POC simplifying assumptions**

* We assume TWO national fund regulation authorities: French and Luxembourg
* Platform actors may have the following roles:
  1. Issuer (fund promoter)
  2. National Fund Regulation Authority (“regulator”)
  3. Optional: National Security Registration Authority (e.g. Euroclear France and Clearstream Luxembourg)
  4. A chaincode validation authority which is in charge of validating that a fund’s chaincode properly follows the agreed upon rules [in the POC, submitted chain code is assumed already validated]
* Issuers are assumed already registered by the regulator
* The following roles are assumed to be non-participating parties: interactions are managed through messages and events rather than through the consensus protocol managed by the platform:
  1. LEI local operating units (e.g. French INSEE, LuxCSD)
* We do not intend to automate a full registration process and merely focus on the workflow
* As a demonstration, we carry on some simple checks such as:
  1. Mandatory information
  2. Agreement validation sealed stamp
  3. …

**Object model**

Object model: each regulator is the owner of a Directory (r). A Directory is a very general object on our platform, which serves the purpose of providing access to funds by registering their chaincode ID. Remember that a fund (f) may support several investment vehicles (v).

We assume each fund has been already created with a proper chaincode.

The Directory (r) itself has a very simple data structure modeling its state. “Transactions” provide an audit trail of state transitions.

The legal description material used for the agreement is owned by the fund chaincode. The regulator’s directory only keeps track of the agreement workflow (agreement status and signature), some basic check capabilities (hashing the fund description) and some basic directory search capabilities (not implemented in the prototype model).

How does it work?

1. A regulator deploys a RegulatedDirectory chaincode for its jurisdiction area. The regulator and the chaincode validation authority are the endorser, and are both mandatory. Optional endorsers are defined for use with APPLY\_ADDITIONAL\_REGISTRATION() transactions. Other peers may be chosen among Security Registration authorities, custodians and CSD’s.
2. The issuer must pre-validate its chaincode with the validation authority (this workflow is not supported by the POC: the authority automatically endorses fund chaincodes in our POC)
3. An issuer deploys a chaincode for a fund and populate some required descriptive (e.g. POPULATE\_LEGAL, POPULATE\_OPERATIONAL)
4. The issuer then initiates the agreement procedure by invoking a regulator’s directory with the target fund’s chaincode ID
   1. Transaction: APPLY\_REGISTRATION( fund chaincode ID)
   2. The method uploads the data from the fund’s chaincode: GET\_SECTION( “[‘legal’,’operational’,…]”)
   3. Check if already known to the directory (agreement’s update) or if this is a new entry
   4. Perform basic validation then endorse the transaction
   5. An event is sent to the regulator
   6. The issuer then submits the transaction to the consensus service
5. The regulator and the issuer exchange feedback and tentative until a final status is reached (agreement validated or rejected). This workflow generates events to each party.
   1. Regulator replies with:
      * 1. INCOMPLETE\_APPLICATION(fund chaincode ID,”Reply message”)
        2. COMPLETED\_REGISTRATION(fund chaincode ID,”Reply message”)
        3. REJECTED\_APPLICATION(fund chaincode ID,”Reply message”)
        4. An event is sent to the issuer – the message is part of the ledger data, but is not reflected in the state
   2. The issuer replies with:
      1. POPULATE\_XXX on its fund’s chaincode, then:
      2. APPLY\_REGISTRATION( fund chaincode ID) on the RegulatedDirectory chaincode
6. Option: whenever the funds requires an official security number, the Security Registration authority is added in the loop. Remember that with this version of the POC, this authority is assumed to be a platform participant.
   1. Note: this option is not known at initial APPLY\_REGISTRATION() time, so it cannot be part of the endorsement procedure (actually, this is not a very clear point of the HL endorsement procedure: if it is possible to query the fund’s chain code at endorsement time, then best to do it directly: the submitting peer would then decide to forward the transaction to the security registration authority)
   2. POPULATE\_REQUEST(fundchaincode, “{ “security-code:”,”ISIN”, “FR” ) is used for that by the issuer

(future extensions: { “lei-code”, “lei”, “FR” } ). This transaction is endorsed by both regulator and securities registration authority [additional step performed by the issuer when not possible to directly submit the transaction at initial endorsement time)

* 1. At this point, the workflow is in state “PendingSecRegistrationAuth”. An event is posted to this authority which in turn will reply with a POPULATE\_REPLY() transaction [same process for authorizing the circulation of securities on a settlement system such as Euroclear). When the transaction is validated, an event notifies the issuer
  2. The issuer now must updates its chaincode with the replies and then triggers again an APPLY\_REGISTRATION()

1. Option: whenever the funds requires an LEI number
   1. This is a similar protocol as before but in this case, the party is not a platform participant
   2. The protocol is managed by events handled directly by the issuer
   3. The party must sign its responses and the regulator must check that data in “populate\_xxx” on its directory are indeed signed.
2. A similar process (in general optional, but for instance mandatory in France) is the populating of the internal registration code.
3. (6), (7) and (8) only differ by the fact that the initiator of the “populate” actions is the issuer whenever the party is external to the platform.
4. Whenever a final status is reached, this status is “sealed” with a cryptographically signed hash of the fund description (at least the list of attributes “locked” by the agreement). An event is posted to the issuer.
5. The issuer may now update its fund’s chaincode with the seal and proceed by opening the fund.

**Note: although predefined in the model, the POC does not provide a workflow for the agreement of issuers. Issuers are simply assumed to be all registered.**

**Note: the GET\_DIRECTORY method to expose the directory to non-peer participants hide all products which have not completed their registration process (GET\_DIRECTORY gives access to both current directory state and ledger data)**

**Identified straightforward generalizations**

The following generalizations toward a more realistic platform could be easily developed from this POC:

* Manage documents with intelligent trusted forms (e.g. Adobe Forms) to be able to accurately validate meaningful data (such as fees, …) (possibly, with automatic rules) while letting a human reader (in the future, perhaps even a robot) to check the legal prose
* Intelligent validation is performed on the client side (SDK), not in the chaincode (which only acknowledges a signed validation from the regulator)
* Propose variants with LEI authority as a platform participant or alternatively with Security Registration authority NOT as platform participant (e.g. more configurable setup)

Examples of more complex generalizations, that would require an extended rework of data model and workflows:

* Issuer agreement workflow
* General directory of authorized regulators, jurisdictions (domestic and transnational – e.g. UCITS)

**Managing events with Hyperledger**

The registration protocol extensively uses HL events. Available HL doc is here: <https://github.com/hyperledger-archives/fabric/wiki/Custom-Events-High-level-specification>

**Posting messages to external parties**

Some events trigger messages to external parties (e.g. non-participant parties).

This message should be issued by a trusted peer on behalf of another participant (principle of oracle gatekeeper applied to outgoing messages).

**Endorsement policy for the RegisteredDirectory chincode**

**Appendix: Prototype JSON schema (draft 4) for Regulated Directory**

{

"$schema": "http://json-schema.org/draft-04/schema#",

"description": "FundsChain schema supporting a fund's registration process (agreement workflow). Copyright TheFundsChain 2016. This schema entry is a directory for regulated products.",

"type": [

"object",

"array"

],

"items": {

"id": "Product",

"title": "Product",

"description": "Directory Entry for a fund product ",

"type": "object",

"properties": {

"PopulatedData": {

"description": "Contributed data from other parties",

"type": "object",

"properties": {

"SecurityCirculationSystems": {

"type": "array",

"items": [

{

"type": "object",

"properties": {

"SecurityCirculationSystemPK": {

"type": "string"

},

"SecurityCirculationSystem": {

"enum": [

"euroclear",

"clearstream"

]

},

"SecurityCirculationSystemSignature": {

"type": "string"

}

},

"required": [

"SecurityCirculationSystemPK",

"SecurityCirculationSystem",

"SecurityCirculationSystemSignature"

]

}

],

"additionalItems": false

},

"SecurityCode": {

"type": "object",

"properties": {

"SecurityRegistrationAuthorityPK": {

"type": "string"

},

"SecurityRegistrationAuthorityName": {

"type": "string"

},

"SecurityCodeType": {

"enum": [

"isin",

"cusip"

],

"default": "isin"

},

"SecurityCode": {

"type": "string"

},

"SecurityRegistrationAuthoritySignature": {

"type": "string"

}

},

"required": [

"SecurityRegistrationAuthorityPK",

"SecurityRegistrationAuthorityName",

"SecurityCodeType",

"SecurityCode",

"SecurityRegistrationAuthoritySignature"

]

},

"RegulatorCode": {

"type": "object",

"properties": {

"RegulatorPK": {

"type": "string"

},

"RegulatorName": {

"type": "string"

},

"RegulatorCodeType": {

"enum": [

"domestic"

],

"default": "domestic"

},

"RegulatorCode": {

"type": "string"

},

"RegulatorSignature": {

"type": "string"

}

},

"required": [

"RegulatorPK",

"RegulatorName",

"RegulatorCodeType",

"RegulatorCode",

"RegulatorSignature"

]

},

"LEICode": {

"type": "object",

"properties": {

"LEIAuhtorityPK": {

"type": "string"

},

"LEIAuthorityName": {

"type": "string"

},

"LEICode": {

"type": "string"

},

"LEIAuthoritySignature": {

"type": "string"

}

},

"required": [

"LEIAuhtorityPK",

"LEIAuthorityName",

"LEICode",

"LEIAuthoritySignature"

]

}

}

},

"FundID": {

"description": "Chaincode ID of the target chaincode subject to the agreement workflow",

"type": "string"

},

"FundChainCodeSignature": {

"description": "The signature of the chaincode operating this product",

"type": "string"

},

"FundHash": {

"description": "A hash signature of the fund's subset of state variables subject to agreement",

"type": "string"

},

"Vehicles": {

"description": "List of registered vehicles operating under this product",

"type": "array",

"items": [

{

"title": "Vehicle",

"type": "object",

"properties": {

"ProductVehicleID": {

"description": "The ID of the investment vehicle exposed through this product",

"type": "string"

}

},

"required": [

"ProductVehicleID"

]

}

],

"additionalItems": false

},

"Issuer": {

"description": "Characterization of the issuer of the product",

"type": "object",

"properties": {

"IssuerAgreementStatus": {

"description": "The issuer's agreement status",

"enum": [

"initiated",

"pendingRegulator",

"pendingIssuer",

"validated",

"rejected"

],

"default": "initiated"

},

"IssuerName": {

"type": "string"

},

"IssuerID": {

"description": "The product issuer party ID on the platform",

"type": "string"

}

},

"required": [

"IssuerAgreementStatus",

"IssuerID"

]

},

"FundAgreementStatus": {

"description": "Workflow agreement status",

"enum": [

"initiated",

"pendingRegulator",

"pendingIssuer",

"pendingSecRegistrationAuth",

"PendingLEIRegistrationAuth",

"PendingCirculationAuth",

"validated",

"rejected"

],

"default": "initiated"

},

"FundAgreementSignature": {

"description": "The regulator's seal on an agreement which reached a final status",

"type": "string"

}

},

"required": [

"FundID",

"FundChainCodeSignature",

"FundHash",

"Issuer",

"FundAgreementStatus"

]

},

"properties": {

"RegulatedDirectory": {

"description": "The description of the directory. It may only be updated by the owner (regulator).",

"type": "object",

"properties": {

"LastSynchTime": {

"description": "Timestamp of last synch with Fund chaincode",

"type": "string",

"default": "190001010000000"

},

"DirectoryOwnerName": {

"type": "string"

},

"DirectoryOwnerID": {

"type": "string"

},

"Jurisdictions": {

"type": "array",

"items": {

"type": "object",

"properties": {

"DirectoryJuridictionAreaType": {

"enum": [

"country",

"transnational"

],

"default": "country"

},

"DirectoryJurisdictionArea": {

"type": "string"

}

},

"required": [

"DirectoryJuridictionAreaType"

]

}

}

},

"required": [

"LastSynchTime",

"DirectoryOwnerName",

"DirectoryOwnerID"

]

}

},

"required": [

"RegulatedDirectory"

],

"additionalProperties": false

}