**ACL with TheFundsChain**

FRED - 20/09/17

**Requirements**

* A MSP provides several certificates as credentials when calling a chaincode:
  + An E-cert (delivered by a subordinate CA)
  + An A-cert
* Introspection: a chaincode may inquire about its endorsers, committers and the ordering service
* Functional roles define a hierarchy: member of a parent role automatically gets all child roles
* Functional role may be delegated
* Each chaincode comes with a self-contained authorization vector, defining "contractual roles"
  + Stored as a JSON document
* Chaincode authorization vector defines:
  + Role membership (this is our "contractual roles"), as organizations, sub-organizations or list of MRP's (e.g. "CACEIS is custodian", "CACEIS SS/Custody Lux is custodian" or "nodes 1, 2 and 3 from CACEIS are custodian"
  + Functional vs H.F technical role mapping
    - E.g: custodian, transfer agent and issuer are endorsers, …
  + Method (Query/Invoke/Deploy) vs Role mapping
    - Using profiles
  + All mappings may be REG-locked (e.g. they are hard-wired and cannot be modified by a chaincode administrator, or Contract-locked (e.g. a signed active contract prevents from modifying it)

**Identification**

*Decentralized certificate authority*

Identity recognition is based on certificates.

Any party found on the platform must have a pair (certificate, private key – PVK -) that has been delivered by a Certificate Authority (CA) approved by the platform.

Such certificates, we call e-certs, are used for identification and authentication.

Identity recognition is decentralized. The root CA identifies direct members, as organizations. It is administered by the platform's caretaker (or "super-admin").

Each organization defines a certificate domain. Ex: 'caceis'.

A limited number of individuals, staff members of such organizations are delivered e-certs bound to these organizations. Owners of such roots certificates may in turn operated their own CA.

Participants organizations operate their own delegated CA, and may in turn deliver certificates for their domains. Subdomains are organized following the rules of the delegated CA.

Organization units may be defined as sub-organizations, following X509 recommendations.

Delegation: e-certs may be delivered to parties acting on behalf of another one.

Ex: "DLTShift, on behalf of French AMF"

*Identified entity types*

e-certs are provided to different entity types, humans and machines as well

* Individuals representing an organization (e.g. CA admin for this organization)
* Individuals from an organization, interacting with the platform (e.g. compliance office, fund manager…)
* Individual nodes operating the platform (or MRP's), either as peers, ordering or SDK-only nodes

*Identification process: enrollment*

* Individuals and organizations identity credentials must be thoroughly checked
  + "Know Your Member" application (KYM)
  + "Know Your Staff" application (KYS)
  + "Know Your Client" application (KYC)

These applications are based on chaincode databases where the credentials handed out are stored.

Identity privacy must be preserved: such chaincodes are shared by a limited number of participants.

KYM chaincode is distributed on caretaker and regulators nodes only.

KYS chaincode is distributed on participant nodes only (possibly only some of them).

KYC chaincode is distributed on distributor nodes only (again, possibly only some of them). Some jurisdiction may require a larger set of peers, e.g. regulator, registrar (Lux.) …

KYS and KYM may be deployed as several, separated instances by participants, e.g. to implement "Chinese wall" rules or abide by different jurisdiction schemes.

When several such instances are deployed, the base principle is that a party is stored only once. However, different instances may collaborate to exchange some data (according to authorization rules). This way, a staff member may be an investor as well and be known from various instances. An investor may have to fill specific documents for investing under a given jurisdiction etc…

KYM, KYS and KYC are for audit purpose. A future replacement for such apps is Hyperledger Indy (though the GUI part would probably remain). Advanced workflows to monitor the validity of credentials (ID's expiration, …) may be setup there. Credentials may be traditional paper ID (passport, ID card, …) or electronic credentials (e.g. EU digital passport). For KYS, enterprise directory entries could be used for staff members.

The main purpose of KY\* is thus to translate "universal" identity credentials into privately managed identities (e-certs).

KYM, KYS & KYC (KY\*) post certificate requests to the adequate CA and return certificates/PVKs. This allows to ring-fence the CA node, which may collect requests asynchronously and push back certificates without any end-user having to contact the CA directly.

Exception: a bootstrap population is required to set KY\* up and running in the first place.

*Identification domains*

* Domains bound to an organization: caceis ; caceis.lux; bpss.custody ; …
* Domains bound to a jurisdiction: eu.fr ; uk ; …

Typical setup:

Caretaker(s) administer the root CA domain

Regulators administer a jurisdiction domain (eu.fr, …), possibly with subdomains (e.g. eur.fr.fcpe)

Other participants administer their organization domain, possibly with subdomains.

*Managing changes*

Certificates may change over time:

* Expiration & renewal (planned renewal?)
* Revocation
* Domain change

Expiration: e-certs have a long life, but ae not eternals. 5 years sounds a proper duration.

Staff move/changes must be declared to revoke previously handed certificates.

All modules dealing with certificates must thus understand the OCCP revocation protocol (revoked certificates are "blacklisted").

Domain change: must be a sequence of revoke / renewal of individual certificates.

*Handling private keys*

Private keys delivered to MRPs (nodes) must be stored locally in a private key storage.

This may be implemented with a Hardware Security Module (HSM)or with software-based key stores.

Private keys delivered to individuals should be:

1. Sent separately from the certificate (SMS, e-mail, snail mail with a USB or smart card…)
2. Ideally stored in a HSM (smart carts, USB dongles…)

Note: unit price for a 16K basic smart card is about $6.

**Authorization**

Authorizations are managed by a different platform module, more deeply integrated with the chaincode ecosystem. We call this bit of the infrastructure an SPKI.

SPKI is decentralized: every owner of a given resource may manage authorization.

SPKI is administrable. Resource owners are provided with an authorization management module. Ideally, this piece of code is factorized.

SPKI is tightly coupled with chaincode business logics.

SPKI relies on two levels to define authorizations:

* Attribute certificates: roles, clearance and profiles
* Chaincode local authorizations: expressions and rules at chaincode level:
  + Chaincode services (invoke, deploy, query)
  + Chaincode data (r/w, reduce, filter)

*Roles, clearance levels and profiles*

* Functional roles define a hierarchy of roles and sub-roles to be granted to parties
* Typically, a regulator grants a role to a participant covering its jurisdiction:
  + Ex: AXA IM is a chartered French asset manager
  + The French regulator signs an attribute certificate on domain 'eu.fr' empowering this organization
* Not all roles require such a certificate: distributor, investor, for instance do not require a regulator's clearance (however, other conditions may be demanded)
* Clearance levels are typically granted by organizations within their domain. The Security Officer may grant clearance levels to e-certs within a given domain
* Profiles are typically defined within an organization

*Attribute Certificates*