# TheFundsChain – Certificate Authorities

Certificate Authorities – a tentative setup. DLT Technology: Hyperledger fabric.

Reference: <http://hyperledger-fabric-ca.readthedocs.io/en/latest/users-guide.html>

Root CA: administered by caretaker

* Defines jurisdiction domains and subordinate CA’s
* Registers and enrolls regulators as CA’s
* Populated attribute: TheFundsChain.Roles
* Identity types managed: regulator

Domestic CA (for a jurisdiction):

* Populated attribute: TheFundsChain.Roles
* Registers and enrolls organizations under its jurisdiction
* Delegate registration and enrollment to affiliates
* Identity types managed: organizations

Organization CA (for a peer’s organization domain):

* Populated attributes
* Identity types managed: peers, users

Distributor CA (for any peer with distribution role)

* Identity types managed: investors

**TODO: SKIP Domestic CA level**

Identity types:

* user: an authenticated user used to deliver certificates
* peer: E-Cert for chaincode peers
* client: E-Cert for SDK-only nodes
* auditor: E-Cert for query only nodes
* caretaker: administrates the root CA domain
* validator: E-Cert for ordering-only peers
* investor: E-Cert for non-member identities (investors)
* jurisdictionOwner: administrates jurisdiction domain
* organizationOwner: administrates organization domain
* investorOwner: administrate distribution domain

Use case:

* caretaker deploys the root CA: TheFundsChain Root CA Authority
* caretaker registers then enroll amf as jurisdictionOwner, then delegates to amf the French domain
  + amf deploys an intermediate CA : « TheFundsChain French CA Authority”
  + amf enrolls CACEIS France, BPSS France, SGSS France, etc… as organizations
  + amf enrolls NAM, AMUNDI, AXAIM, BNPIP, etc… as organizations
  + organizations deploy their own intermediate CA affiliated to the French CA
    - e.g. « TheFundsChain CACEIS France CA Authority”
    - CACEIS enrolls its nodes as peers
    - NAM enrolls its nodes as peers
  + distributors: amf enrolls NAM-DIRECT, NGAM as organizations
  + NAM-DIRECT, NGAM, … (distributors) deploy their own intermediate CA affiliated to the French CA
    - NAM-DIRECT enrolls its investors



# Certificate delivery

Sequence:

1. amf requests a CA certificate for identity type ‘jurisdictionOwner’ from root CA [CSR with CA profile] (path length: 1)
2. amf deploys intermediate CA with this newly acquired certificate
3. amf register its nodes with identity type ‘peer’
   1. TheFundsChain.Roles attribute set to ‘regulator’
4. Caceis France requests from amf a CA certificate for identity type ‘organization’ (path length 0 : this intermediate CA won’t be able to further delegate CA’s)
5. Caceis certificate is set with TheFundsChain attribute: ‘custodian’, ‘accountant’, …
6. Caceis deploys intermediate CA with this newly acquired certificate
7. Caceis registers its own nodes on his CA with identity type ‘peer’
   1. TheFundsChain.Roles attribute set to ‘custodian’, etc..

# Understanding attributes propagation with certificates authorities

Since attributes represent privileges, it is essential that an intermediate CA cannot provide certificate with any kind of attribute.

The idea is that an intermediate CA should be able to set an attribute only if already has this attribute set.

Example:

* only caretaker user on root CA may deliver a regulator attribute
* only amf user on French CA may delivery an organization a ‘custodian’ attribute
* the organization may set for its own peers the attribute set that their own CA has been given by amf

**Unfortunately, this is not the case with Fabric CA: every CA may deliver the attributes of its own choosing…**

**Note: these checks are specific to Fabric CA**

**They are implemented here:** <https://github.com/hyperledger/fabric-ca/blob/master/lib/serverregister.go>

**And could be augmented.**