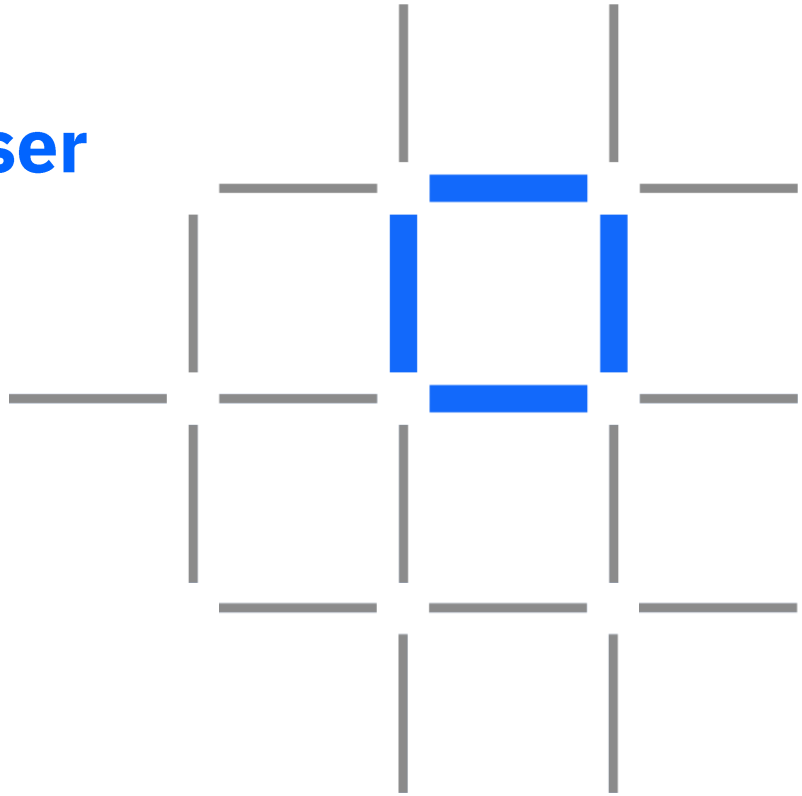
Module 3

Identity and Membership in Hyperledger Fabric + Composer



IBM **Blockchain**

IBM

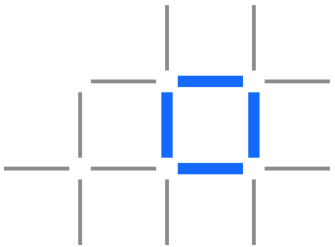
What is a certificate?

A certificate is a document that verifies the identity.

Digital certificates are used to verify a user is who her or she claims to be. It is a form of identification, like a passport.

Certificates are issued by Certificate Authorities (CAs), and represent that issuing organization is vouching for the identity of the organization for whom the certificate is issued.

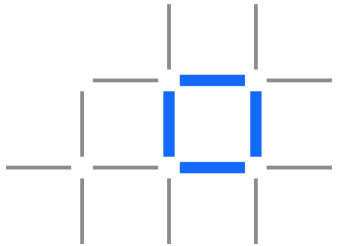
Certificates have a common structure defined by the X.509 cryptography standard



Structure of an X.509 Certificate


```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            40:05:54:3a:12:c0:b9:9c:64:5e:ca:42:e2:0f:75:06
    Signature Algorithm: ecdsa-with-SHA256
        Issuer: C = US, ST = California, L = San Francisco, O = wills.com, CN = tlsca.wills.com
        Validity
            Not Before: Nov  1 20:15:16 2017 GMT
            Not After : Oct 30 20:15:16 2027 GMT
        Subject: C = US, ST = California, L = San Francisco, CN = orderer0.wills.com
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:60:54:36:4b:f2:90:c1:ae:72:4d:d2:ff:70:4f:
                    8f:e9:9b:f8:34:62:32:ef:8e:ef:40:08:94:70:f9:
                    81:ab:f7:42:2d:7a:fc:43:f1:e2:40:a3:90:21:29:
                    6a:b6:92:db:f0:88:dd:d6:a1:c4:7a:f1:b1:0f:bd:
                    44:f3:72:15:1b
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Authority Key Identifier:
                keyid:46:1E:25:C5:EE:AC:77:48:A0:57:0B:F7:AC:21:E0:3A:6D:49:F1:EE:47:5B:F9:A4:7A:87:5E:D6:04:F5:9C:53

            X509v3 Subject Alternative Name:
                DNS:orderer0.wills.com, DNS:orderer0
    Signature Algorithm: ecdsa-with-SHA256
         30:44:02:20:67:68:33:fa:37:83:41:90:ab:72:bf:2e:f6:c6:
         d9:19:a1:ce:fe:19:5d:2a:86:0e:f3:a6:c6:e2:2f:94:fc:d7:
         02:20:49:bb:11:c1:fc:84:06:f8:f3:8f:85:08:9f:05:73:70:
         b9:54:2f:d6:dc:ff:3a:fa:f6:5f:1c:23:a6:b9:9c:d4
```

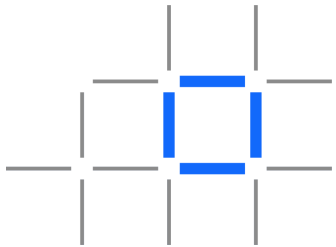


Certificates in Hyperledger Fabric

Certificates are used extensively within Hyperledger Fabric to prove identity within the network, and prove assocation to an **organization**

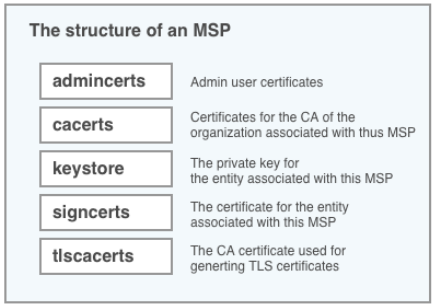
Within an organization, certificates are used for two main purposes

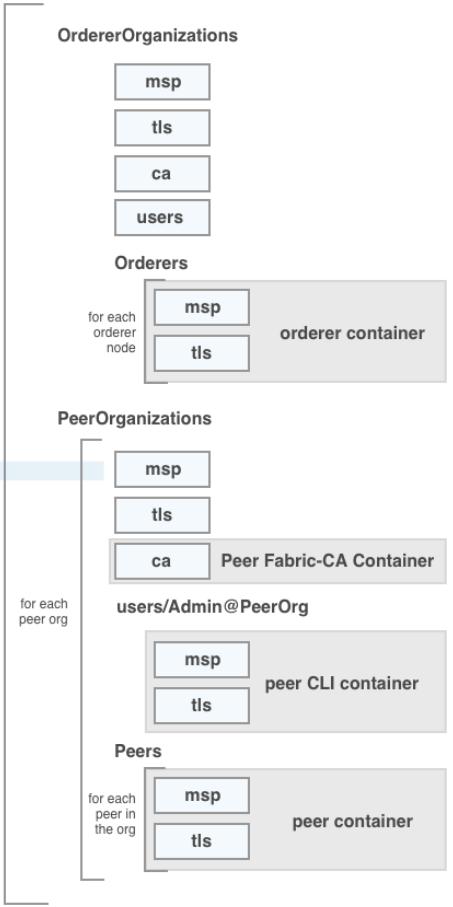
1. To identify computing resources within a network (e.g. peer nodes)
2. To identify individuals who belong to an organization



Network Identity in Hyperledger Fabric

Config files are used to define the participant organizations within a network, and to generate the certificates for the network components to use.





User Identity in Hyperledger Fabric

During the network identity generation process, certificates are created for use by the Certificate Authorities for each organization in the network.

Individual user certificates are then generated by the CA for the organization and used to sign individual transactions on the blockchain

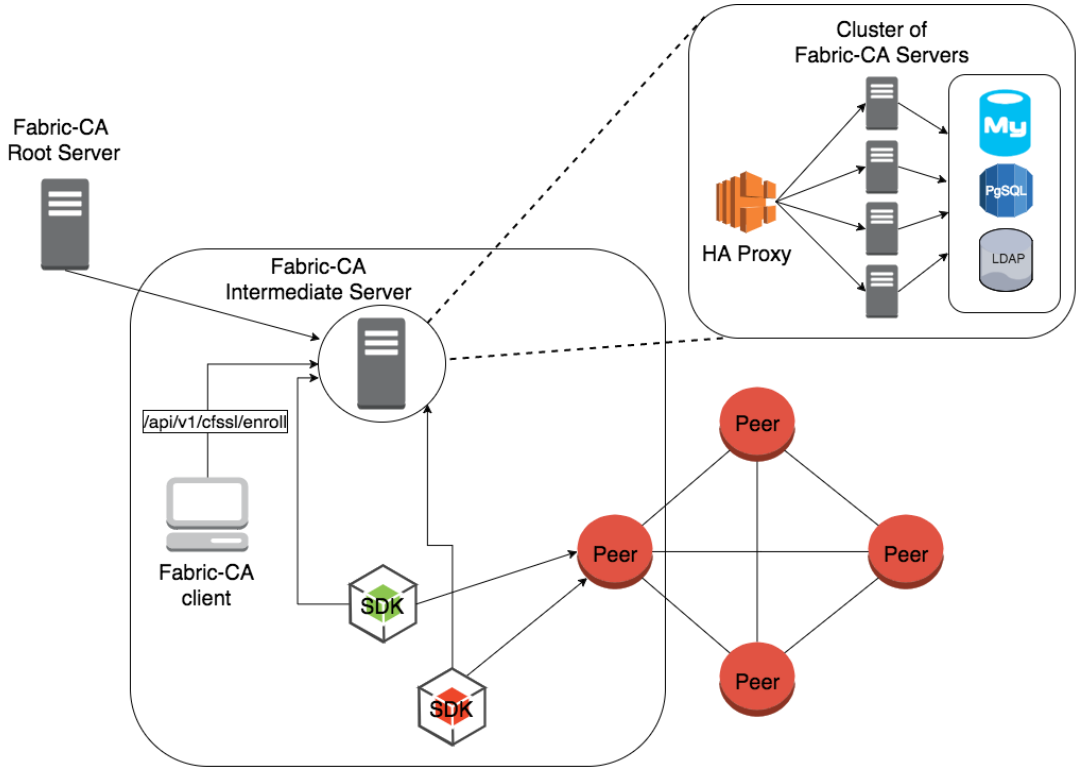
When using Fabric-CA, user certificates can be issued in three ways:

1. Through the Fabric-CA command line client
2. Through the Fabric-CA SDK
3. Through the Fabric-CA REST API

With Fabric-CA there is a **two-step process** to get a certificate

1. The user is first **registered** and provided with a one-time-use **secret**
2. The secret is then provided during **enrollment**, which results in a certificate and associated key material being created.

Fabric CA - Architecture



Blockchain

Identity in Hyperledger Composer

When using Hyperledger Composer **identities** (defined by the existence of certificates) are associated with **Participants** who are defined within the Business Network definition

You can use the Composer command line utilities to manage identities in the business network.

`composer participant add`
Adds a participant to a participant registry

`composer identity issue`
Issue a new identity to a participant

`composer identity bind`
Bind an existing identity to a participant

`composer identity list`
List all identities in a business network

`composer identity revoke`
Revoke an identity from a participant

`composer identity import`
Import an identity to your local identity wallet

Identity in Hyperledger Composer REST Server

When using the Hyperledger Composer REST Server component, an additional concept of a **wallet** is introduced.

Wallets are the representation of how the REST server stores the certificates / credentials for the users that are calling the REST API, so that those certificates can be used to sign the blockchain transactions performed by the REST API.

Identities are placed in wallets using the system (REST) APIs exposed by the Composer REST server