课程安排

03/14 区块链赋能产业价值和商业模式
03/21 Hyperledger 项目概览 社区介绍
03/28 Fabric 1.4 LTS 功能介绍 架构概览
04/04 Peer 解析
04/11 Orderer 解析
04/18 MSP 与 CA
04/25 应用开发指南
05/09 部署实践

欢迎关注微信公众号
“IBM开源技术”
获取更多资讯

公众号中发送**“replay”**
获取往期视频地址

公众号中发送“**报名**”,
即有机会参加Fabric线下训练营

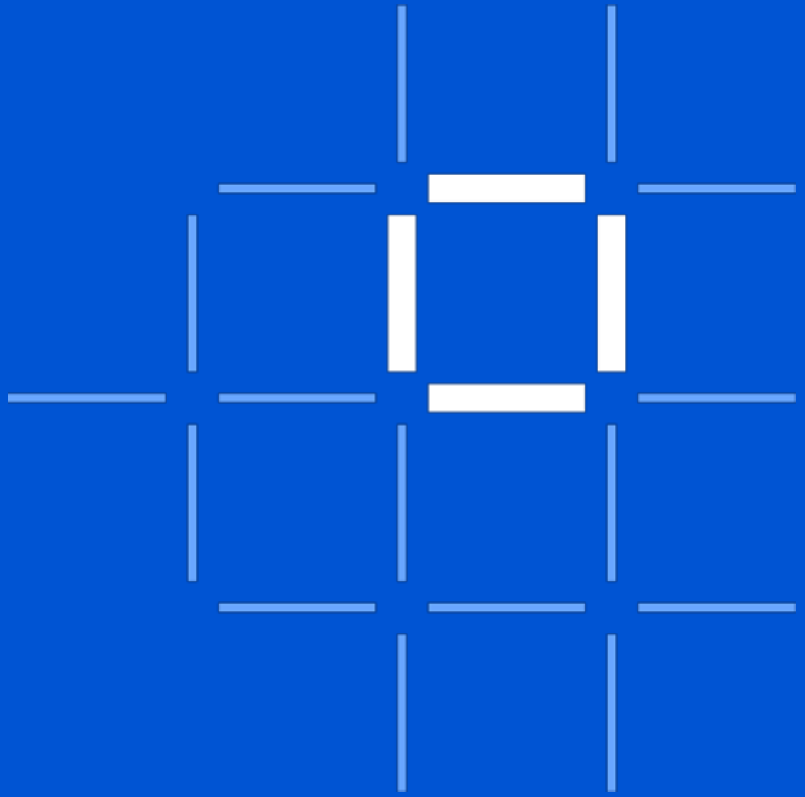
Fabric CA Overview

PKI – X.509

MSP structure and usage

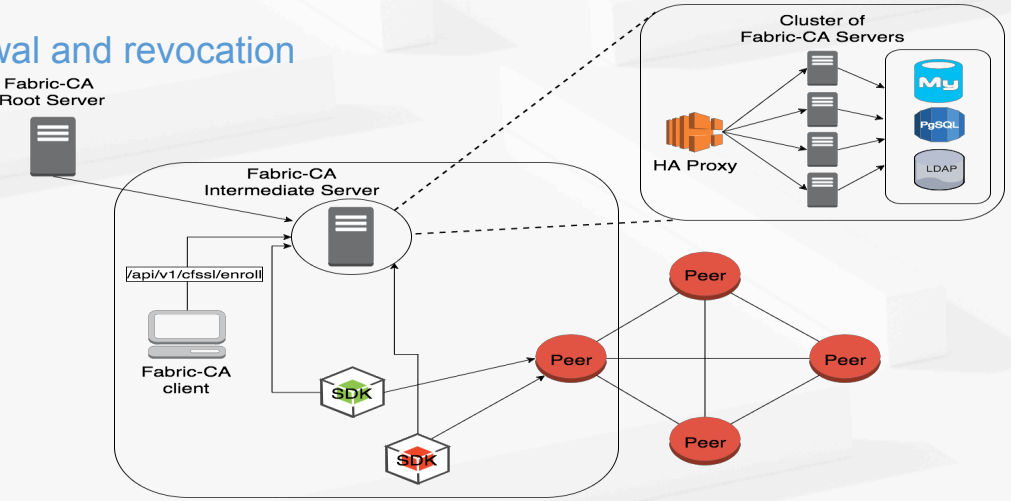
Identity Mixer

Hardware Security Module (HSM) and advantages



Hyperledger Fabric CA

- Features:
  - Registration of identities
  - Enrollment Certs
  - Certificate renewal and revocation
- C/S
- Architecture

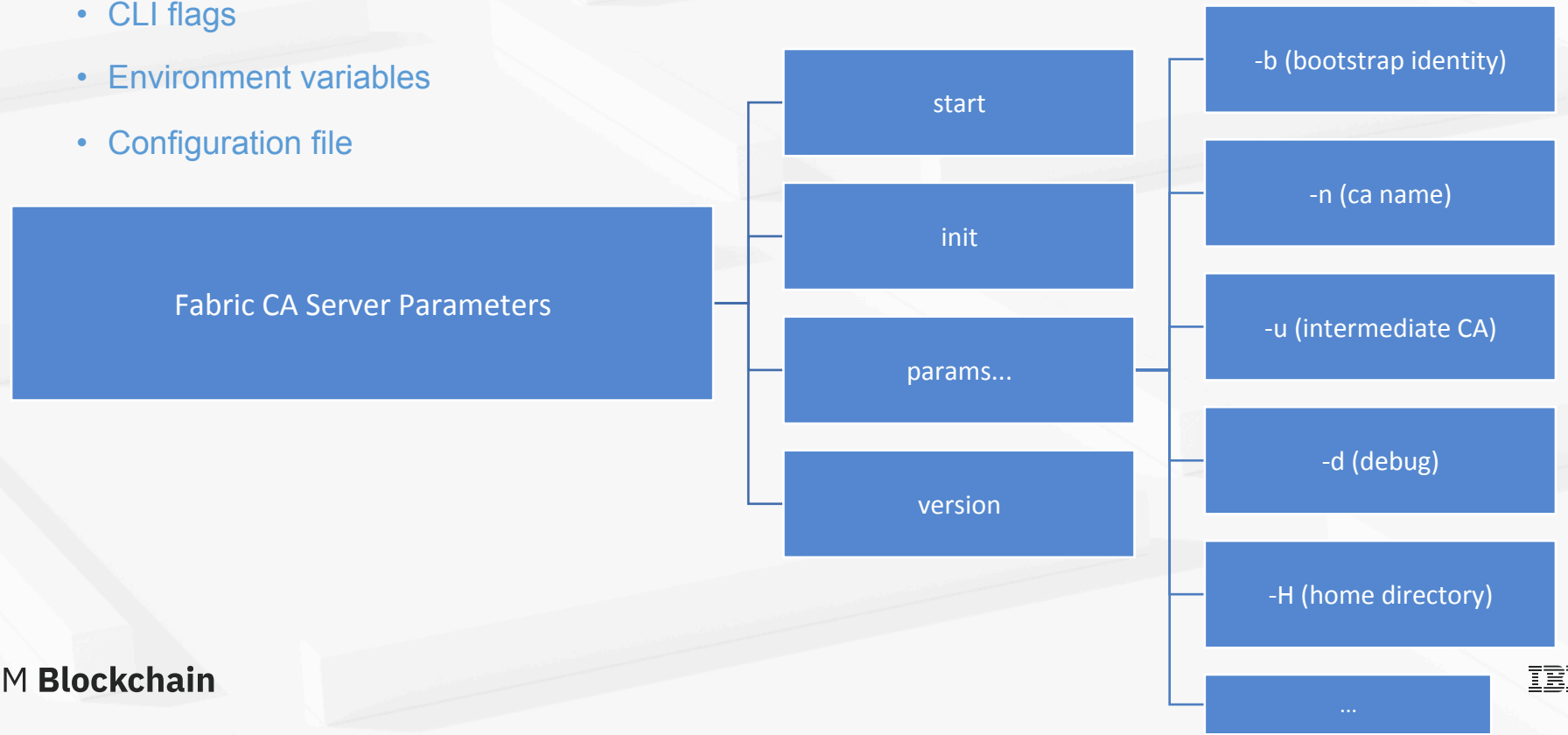


Blockchain

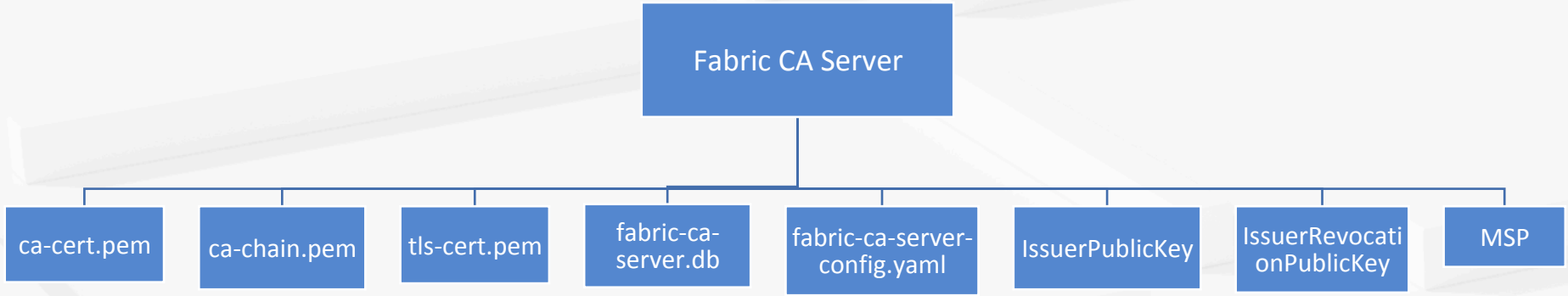
Text

Fabric CA Server

- Configure settings:
  - CLI flags
  - Environment variables
  - Configuration file

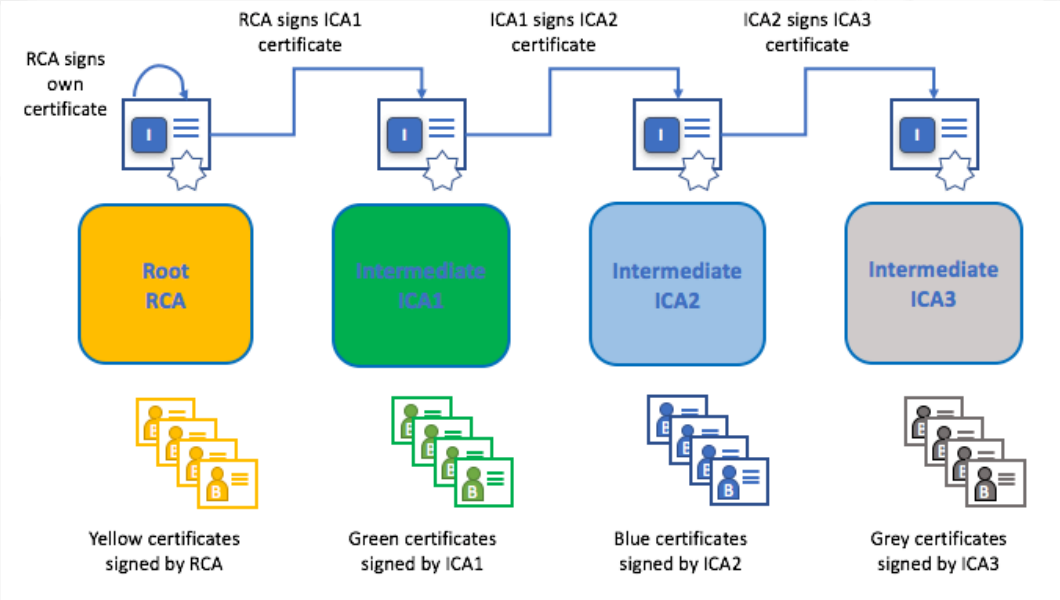


Fabric CA Server Init

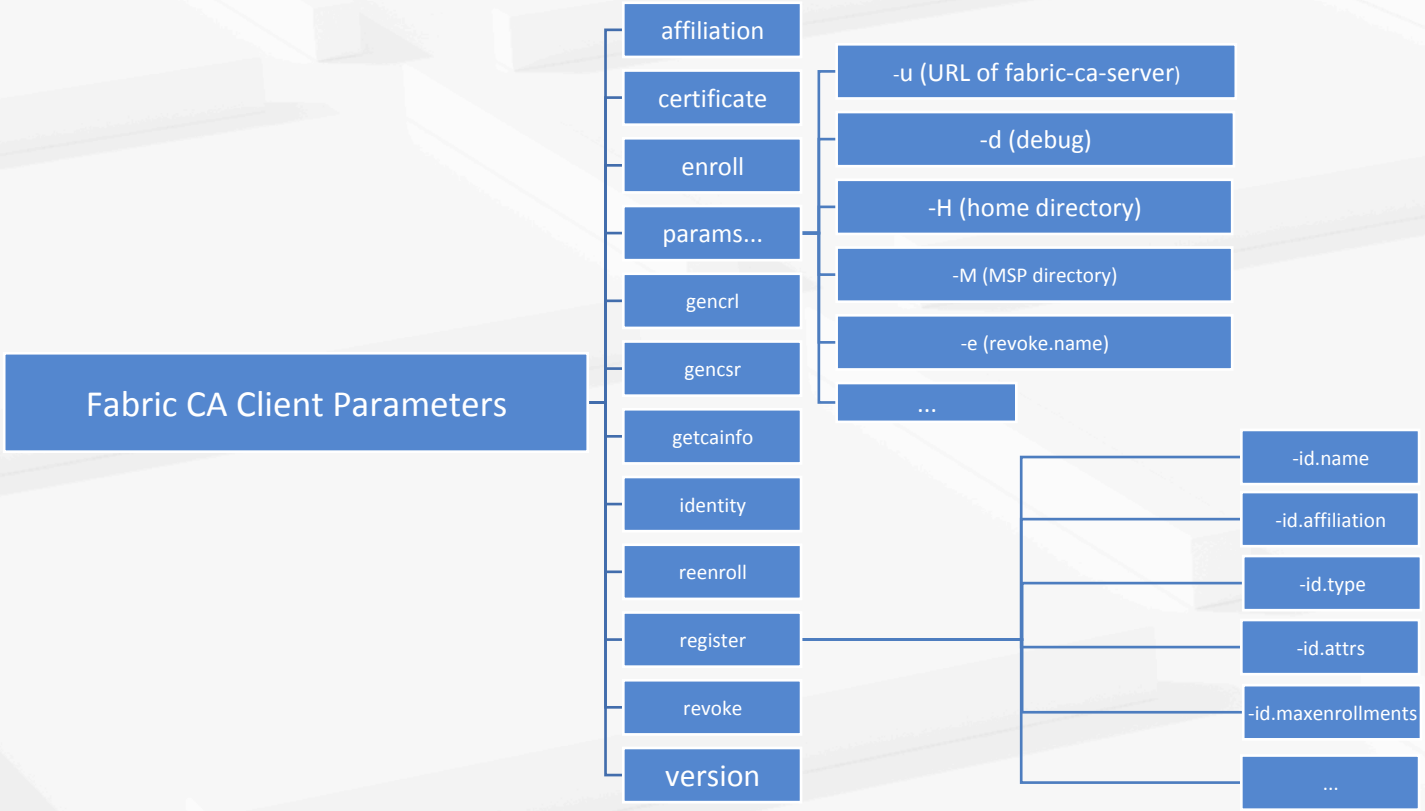


Intermedia CA

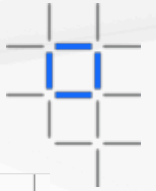
- Limit exposure of Root CA
- Across multiple organizations
- Enroll intermedia CA with Root CA
- Certificate Chain trust between Root CA and a set of Intermediate CA



Fabric CA Client



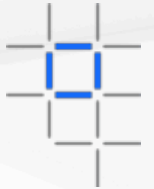
ABAC(Attribute-Based Access Control)



- Access control decision can be made by chaincode
  - Register with attributes - '**id.attrs**'
  - Enroll with attributes - '**enrollment.attrs**'
  - 3 default attributes in Ecert:
    - hf.EnrollmentID
    - hf.Type
    - hf.Affiliation
  - ':ecert' to add attribute into Ecert

Name	Type
hf.Registrar.Roles	List
hf.Registrar.DelegateRoles	List
hf.Registrar.Attributes	List
hf.GenCRL	Boolean
hf.Revoker	Boolean
hf.AffiliationMgr	Boolean
hf.IntermediateCA	Boolean

CSR (certificate signing request)

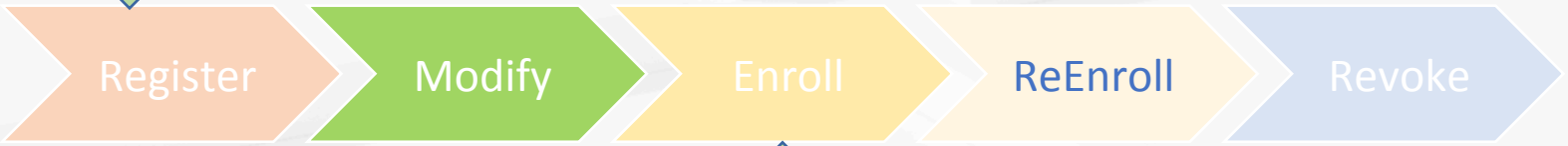


- Provide to a Certification Authority (CA)
- X.509 certificates and keys(ECDSA)

```
csr:
  cn: <<enrollment ID>>
  key:
    algo: ecdsa
    size: 256
  names:
    - C: US
      ST: North Carolina
      L:
      O: Hyperledger Fabric
      OU: Fabric CA
  hosts:
    - <<hostname of the fabric-ca-client>>
  ca:
    pathlen:
    pathlenzero:
    expiry:
```

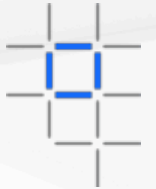
Identity lifecycle

fabric-ca-client register -d --**id.name** demouser --**id.affiliation** org1.department1 --**id.type** peer --maxenrollments -1 --**id.attrs** '"hf.Registrar.Roles=peer,user",hf.Revoker=true:**ecert**' -u <fabric-ca-server>:<port>



fabric-ca-client enroll -u https://demouser:HSrcxfuFcoDg@<fabric-ca-server>:<port> -H <msp directory>--caname <cn.name>

Hyperledger Fabric SDKs



- Currently, Node.js and Java SDKs are supported.
  - Hyperledger Fabric Node SDK documentation.
  - Hyperledger Fabric Java SDK documentation.

- Sample of fabric-ca-client SDK:

  - **fabric-ca-client**:
    - register a new user
    - enroll a user to obtain the enrollment certificate signed by the Fabric CA
    - revoke an existing user by enrollment ID or revoke a specific certificate
    - customizable persistence store

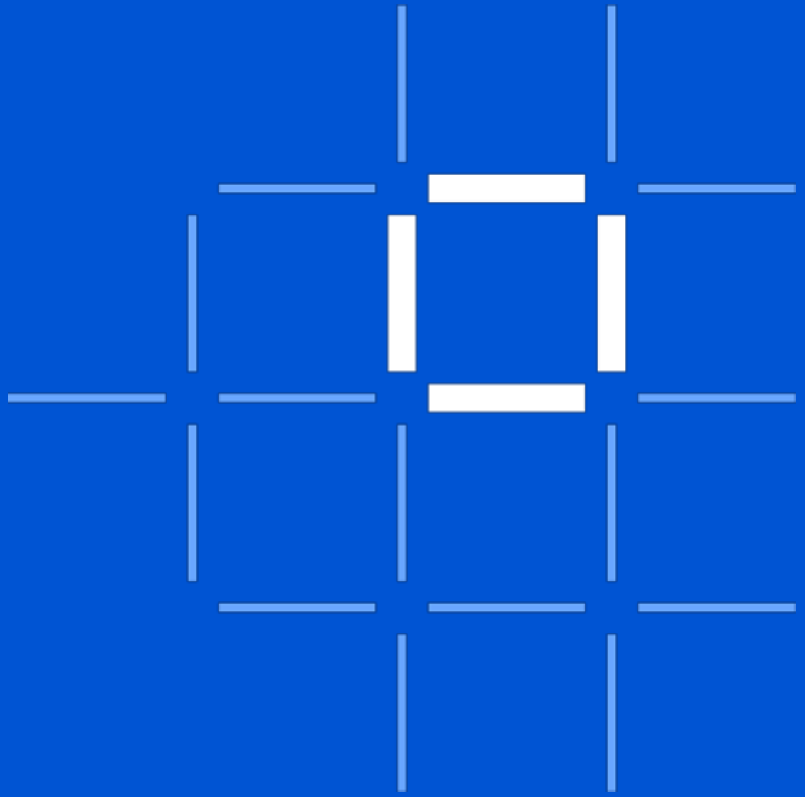
Fabric CA Overview

PKI – X.509

MSP structure and usage

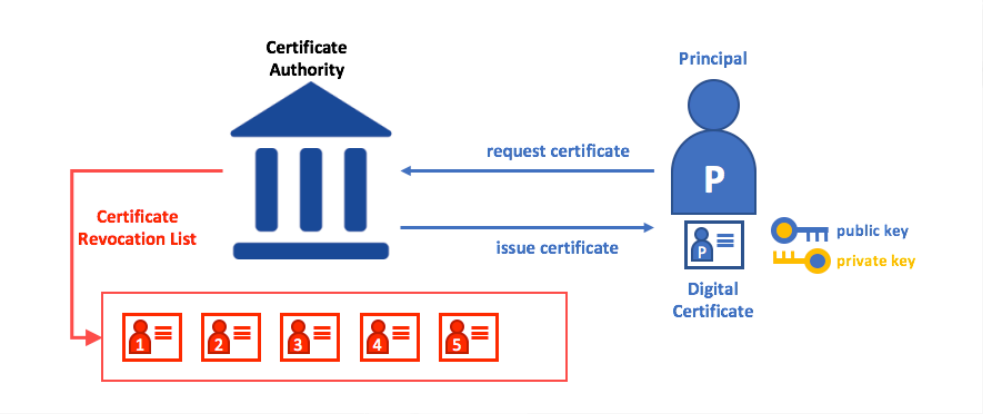
Identity Mixer

Hardware Security Module (HSM) and advantages



PKI

- PKI(Public Key Infrastructure)
-
  - Digital Certificates
  - Public and Private keys
  - Certificate Authorities
  - Certificate Revocation List



X.509 certificates by Fabric CA

- Digital certificate is compliant with X.509 standard
- fabric-ca-client register -d --**id.name** demouser --**id.affiliation** org1.department1 --**id.type** peer --maxenrollments -1 --**id.attrs** '"hf.Registrar.Roles=peer,user",hf.Revoker=true:**ecert**'
- X.509 cert content：
  - X.509 version
  - Certificate Serial Number
  - Signature Algorithm(ecdsa-with-SHA256)
  - Issuer
  - Validity date
  - Subject
  - Subject Public key info
  - Public key

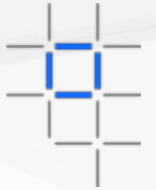
```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            59:a1:d5:17:9d:dd:c9:b0:5e:34:06:3a:f4:e2:3d:70:88:81:f2:c0
    Signature Algorithm: ecdsa-with-SHA256
        Issuer: C=US, ST=North Carolina, O=Hyperledger, OU=Fabric, CN=tlsca-common
        Validity
            Not Before: Apr 17 08:53:00 2019 GMT
            Not After : Apr 16 08:58:00 2020 GMT
        Subject: C=US, ST=North Carolina, O=Hyperledger, OU=peer, OU=org1, OU=department1, CN=demouser
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:44:9f:c2:d4:6e:85:8b:a7:69:02:67:6c:19:f8:
                    c3:04:c8:e0:ca:7e:ad:2b:8d:7a:07:b4:f5:90:d4:
                    85:df:37:45:75:4e:fc:d5:9e:92:b8:31:11:6e:50:
                    05:35:1f:ef:9e:5b:e2:bc:2a:2c:0f:f1:ab:6c:75:
                    bf:ee:50:9c:93
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment, Key Agreement
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Subject Key Identifier:
                D8:F0:C5:7E:09:C0:AD:1E:DD:C9:63:BC:C9:ED:F2:89:1A:B9:37:27
            X509v3 Authority Key Identifier:
                keyid:20:C0:24:29:FD:87:32:6D:10:CC:0C:BC:88:19:75:C2:D0:68:38:B1

            X509v3 Subject Alternative Name:
                IP Address:9.47.152.179
            1.2.3.4.5.6.7.8.1:
                {"attrs":{"hf.Affiliation":"org1.department1","hf.EnrollmentID":"demouser","hf.Revoker":"true","hf.Type":"peer"}}
    Signature Algorithm: ecdsa-with-SHA256
         30:44:02:20:75:9c:59:b9:09:c9:7a:fb:0f:79:39:29:a9:a8:
         94:85:8e:02:66:45:d2:93:62:47:4d:97:1e:43:50:af:66:8e:
         02:20:18:0c:93:62:18:26:8f:f9:72:1c:1e:6e:e7:5c:c8:cf:
```

Issuer

Identity

Attributes of Identity

PKI and X.509



- Blockchain network relies on PKI :
  - Ensure secure communication between various network participants
  - Ensure posted message are properly authenticated
- MSP uses X.509 certificates
- PKI provides a list of identities vs. MSP are members of a given organization that participates in the network.
- PKI certificate authorities vs. MSPs provide a similar combination of functionalities

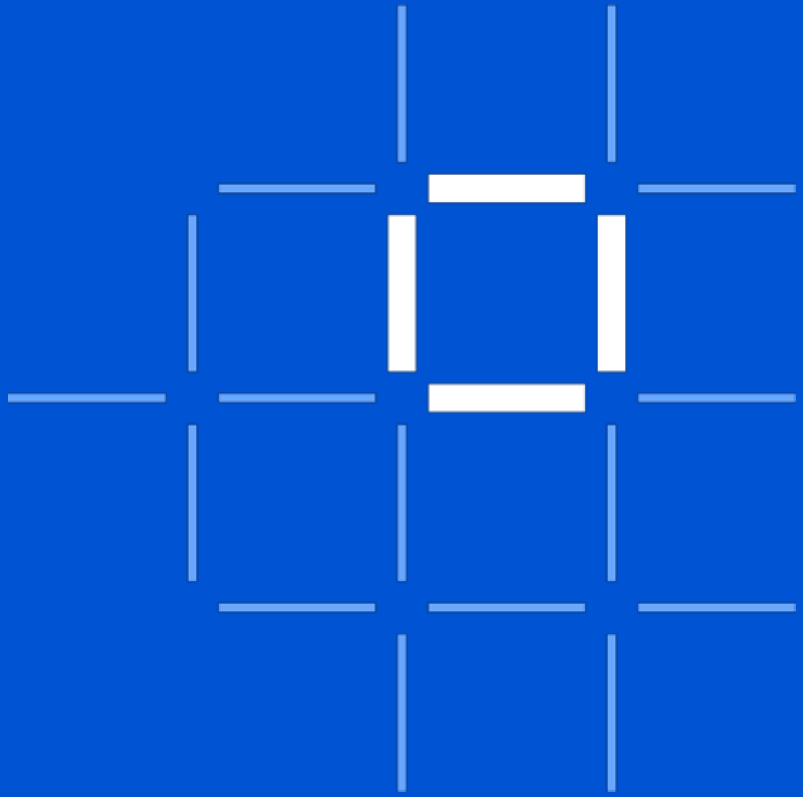
Fabric CA Overview

PKI – X.509

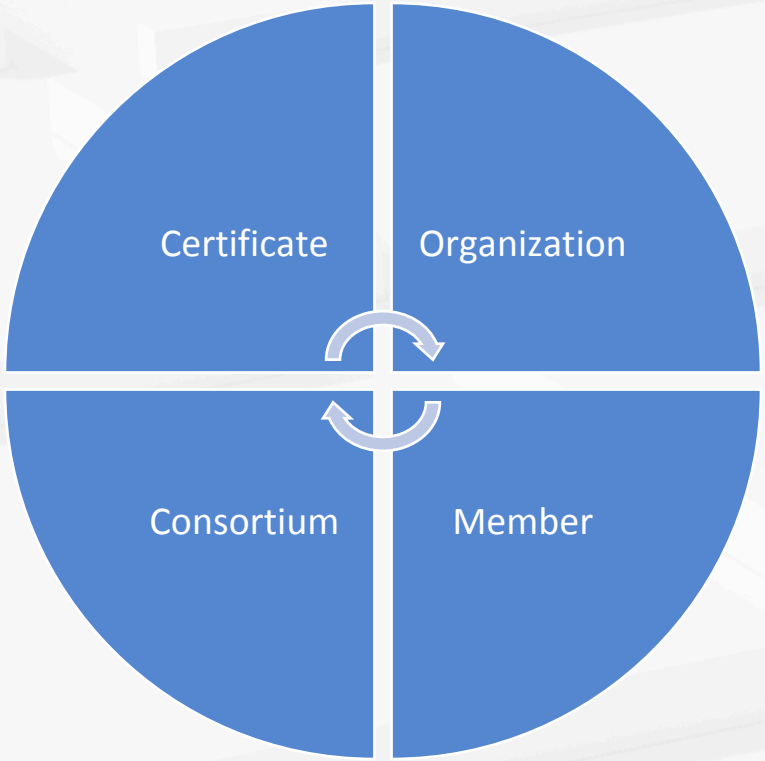
MSP structure and usage

Identity Mixer

Hardware Security Module (HSM) and advantages

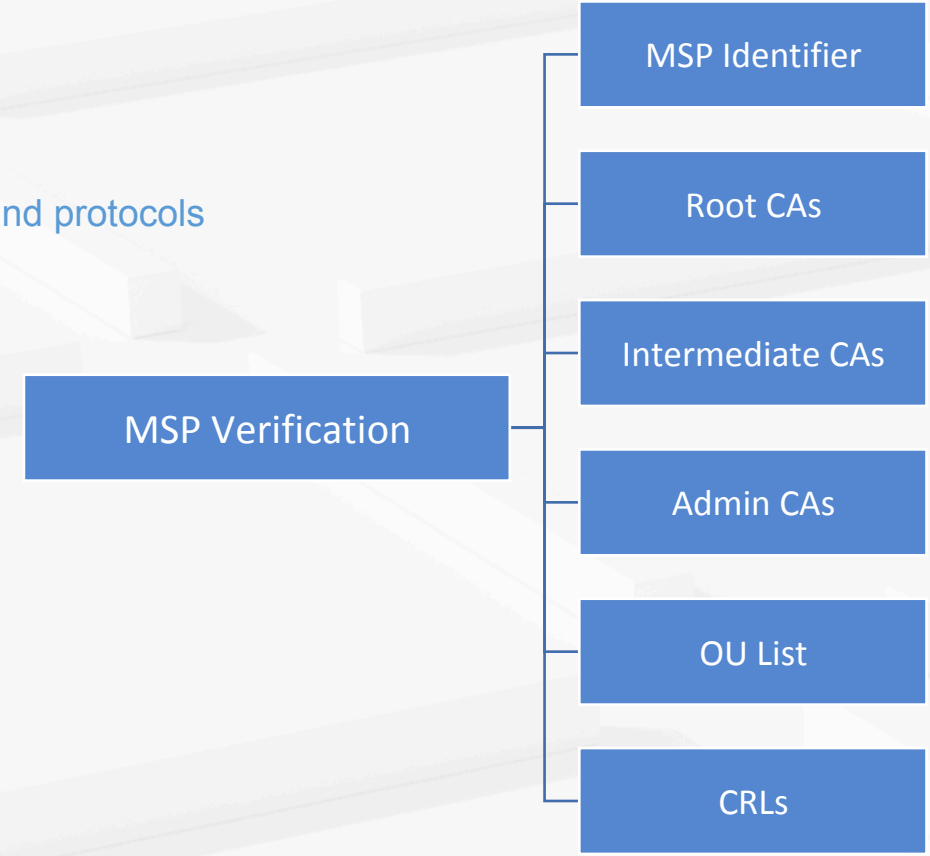


Membership Service Provider



Membership Service Provider

- Abstracts all cryptographic mechanisms and protocols
- Provide credentials to clients and peers
  - clients - authenticate transaction
  - peers - endorsements
- 1-N MSP & MSP ID **unique**
- Fabric CA, OpenSSL, Cryptogen …

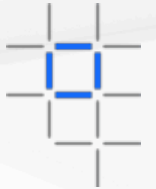


BCCSP

- Blockchain cryptographic service provider
- Implementation:
  - PKCS #11
  - SW

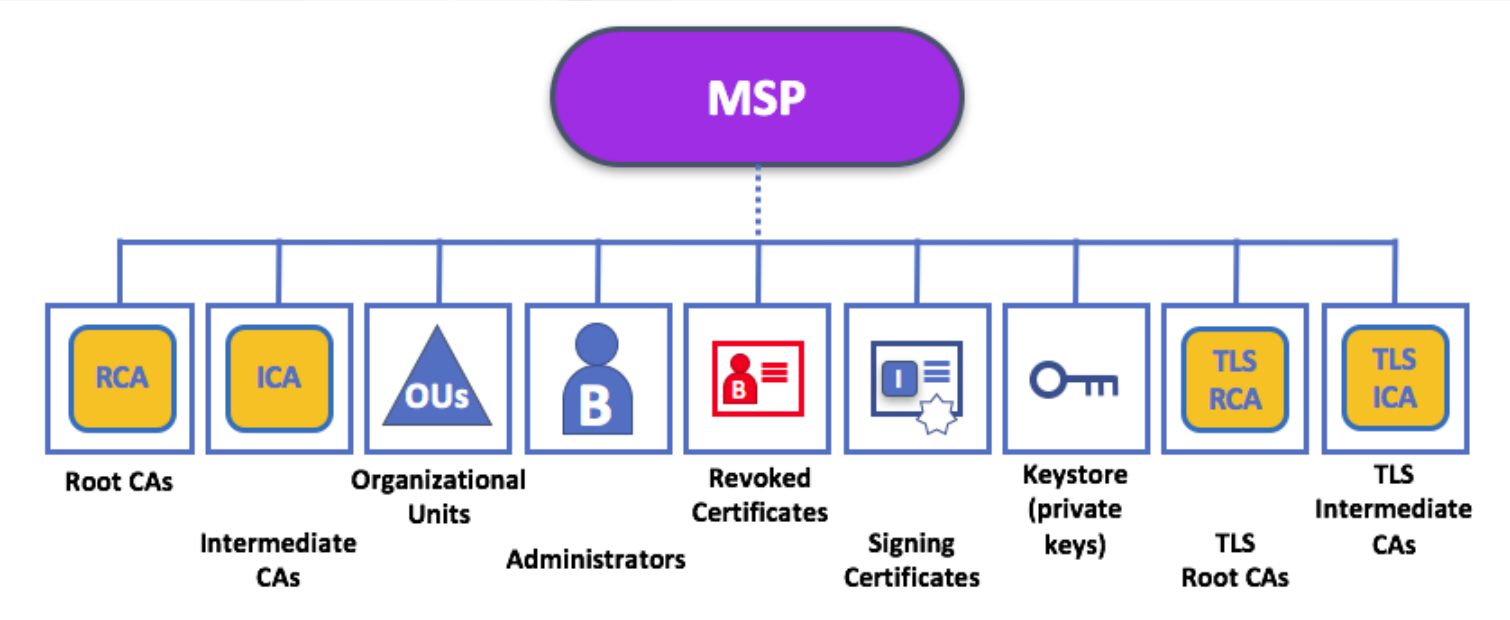
```
###############################################################################
# BCCSP (BlockChain Crypto Service Provider) section is used to select which
# crypto library implementation to use
###############################################################################
bccsp:
    default: SW
    sw:
        hash: SHA2
        security: 256
        filekeystore:
            # The directory used for the software file-based keystore
            keystore: msp/keystore
```

MSP Configuration

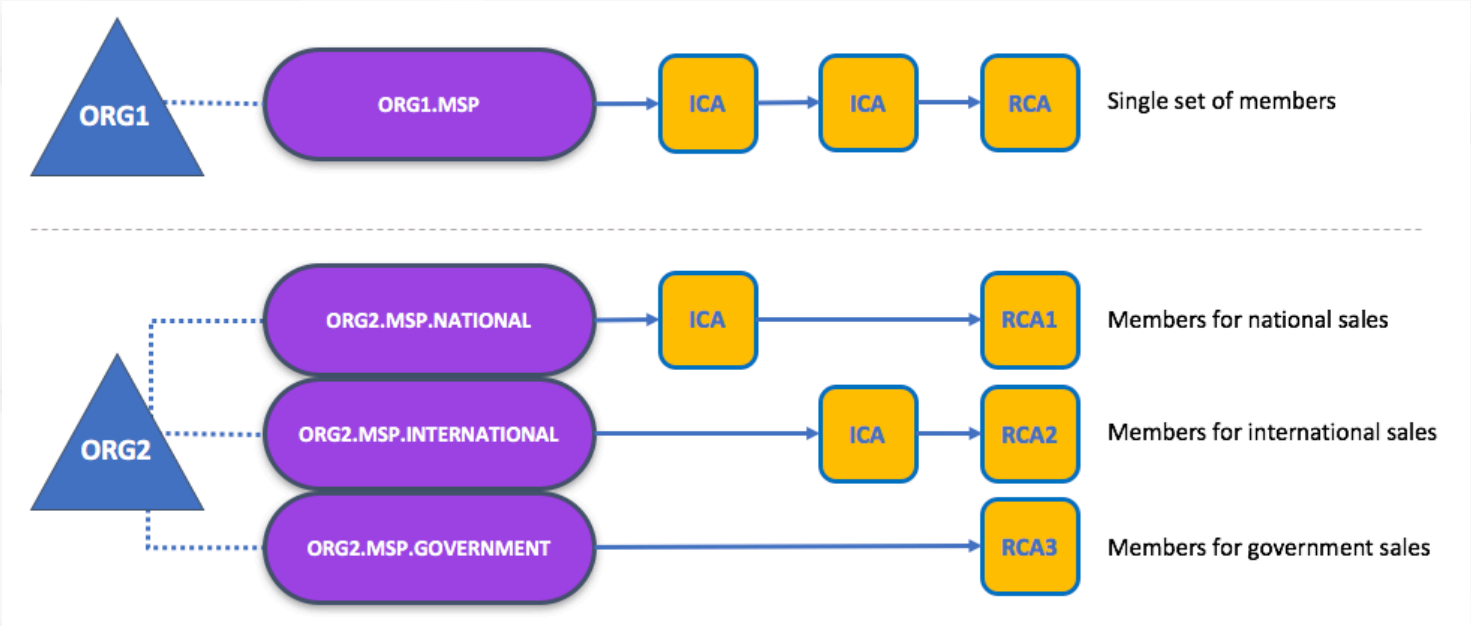


- Specified at each peer and orderer and channels
  - Enable peer, orderer, client identity validation, and signature verification(authentication)
- Reduce from **RFC5280**
  - A list of self-signed X.509 certificates to constitute the root of trust and the TLS root of trust for TLS certificate
  - A list of X.509 certificates with verifiable certificate path to represent MSP of administrators, and authorized to request changes to this MSP configuration(eg. Root Cas, intermediate CAs)
  - Intermediate certificates, certified exactly by one of the certificates in the root of trust;
  - A list of OU
  - CRL
  - …
- **Valid identities of MSP**:
  - X.509 certificates, verifiable certificate
  - Not included in any CRL;
  - List one or more of OU of MSP configuration in their X.509 certificate structure;

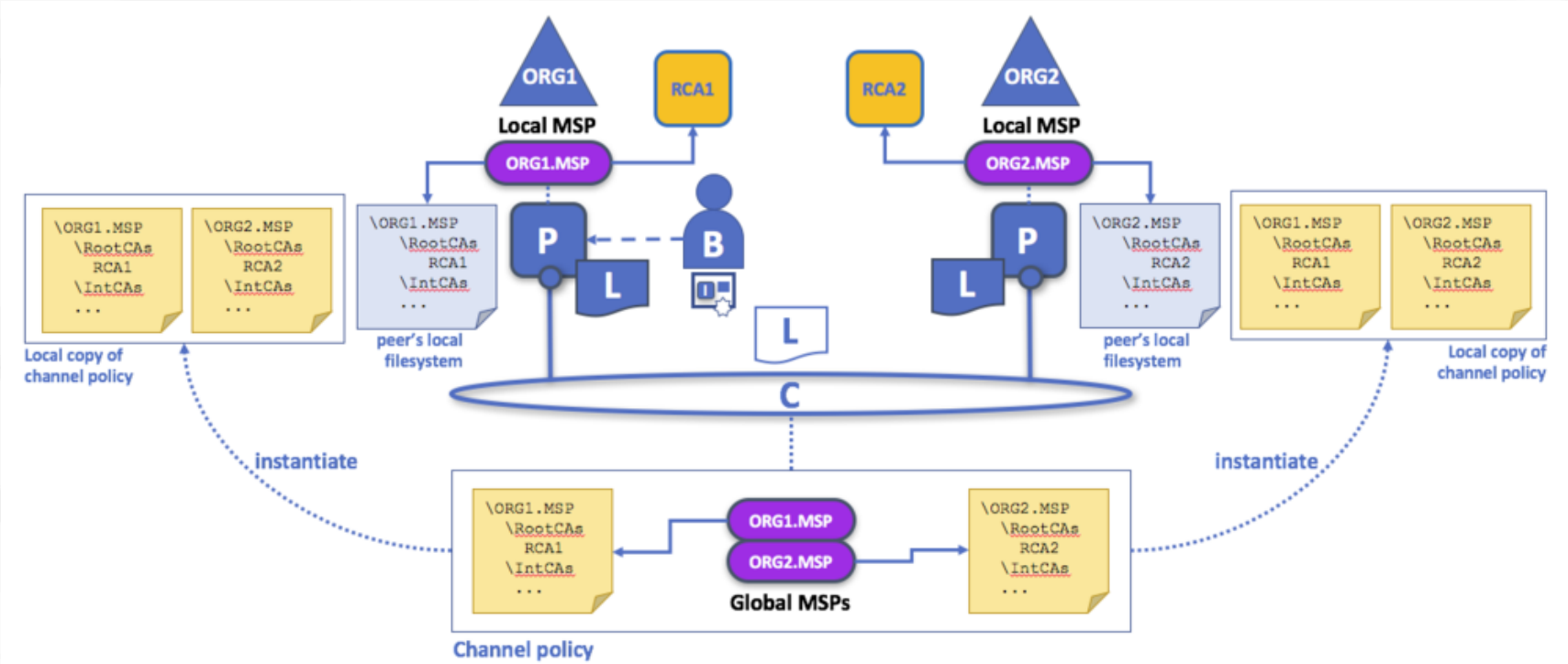
MSP Tree


MSP
RCA
ICA
OUs
B
B
TLS RCA
TLS ICA
Root CAs
Intermediate CAs
Organizational Units
Administrators
Revoked Certificates
Signing Certificates
Keystore (private keys)
TLS Root CAs
TLS Intermediate CAs

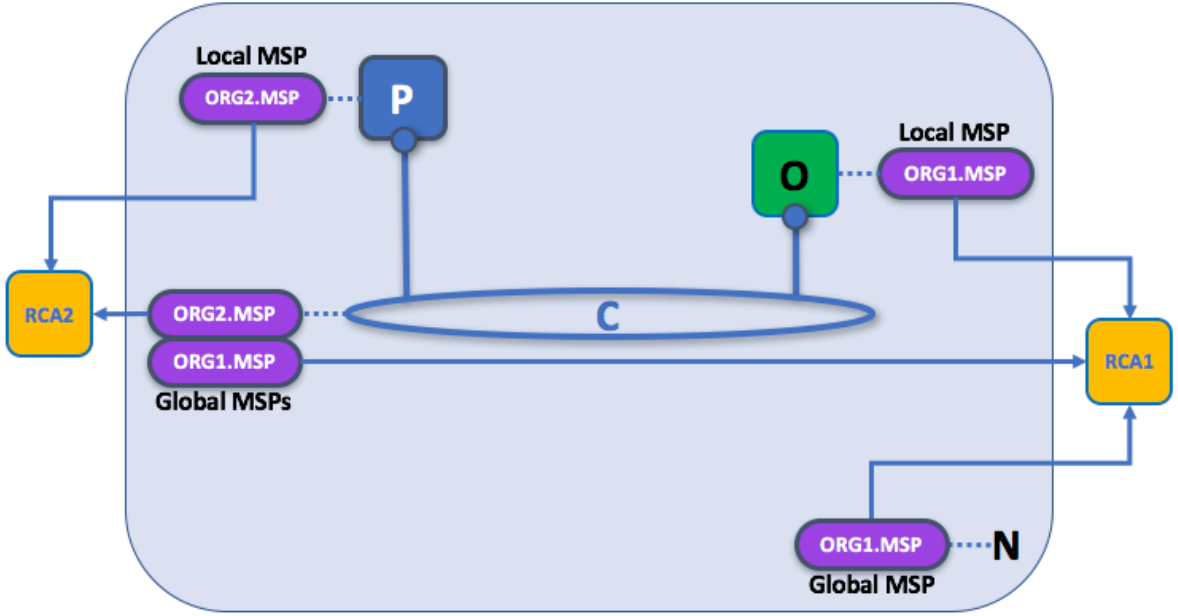
Organization - MSP



MSP in Channel

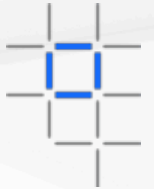


MSP Levels



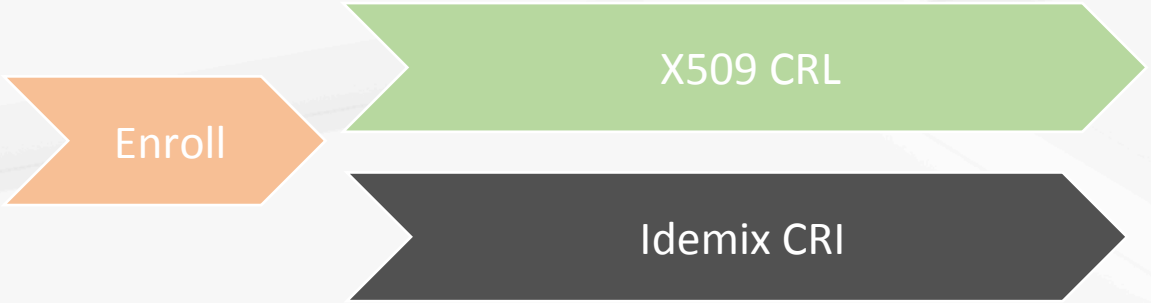
N	Blockchain Network
C	Channel
MSP	Membership Services Provider
P	Peer
O	Orderer
CA	Certificate Authority

MSP configuration in Peer/Orderer

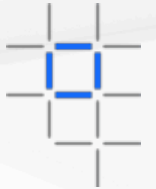


```
# Path on the file system where peer will find MSP local configurations
    mspConfigPath: msp


    # Identifier of the local MSP
    # ----!!!!IMPORTANT!!!-!!!IMPORTANT!!!-!!!IMPORTANT!!!!----
    # Deployers need to change the value of the localMspId string.
    # In particular, the name of the local MSP ID of a peer needs
    # to match the name of one of the MSPs in each of the channel
    # that this peer is a member of. Otherwise this peer's messages
    # will not be identified as valid by other nodes.
    localMspId: SampleOrg
# Type for the local MSP - by default it's of type bccsp
    localMspType: bccsp
```

MSP Lifecycle

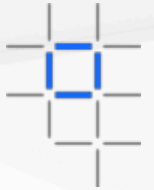


MSP Practice

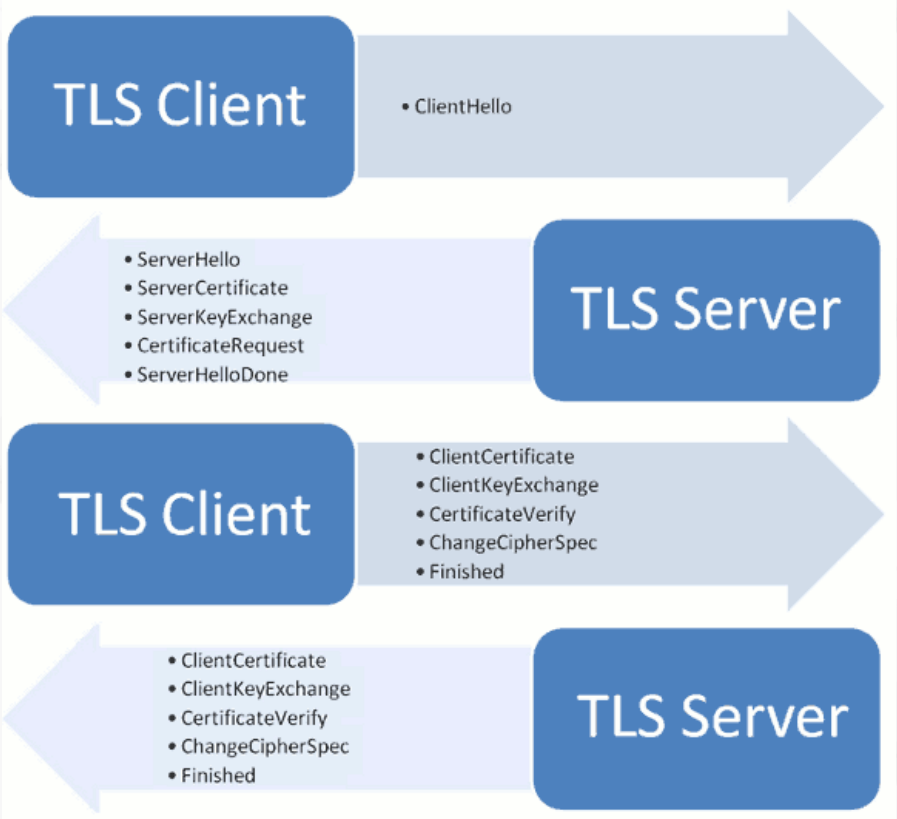


- 1 Org -1 MSP (recommended)
  - 1 Org – N MSPs
  - N Org – 1 MSP
- Grant different divisions different access to different channels
- Admin and CA certs
- Blacklisting an intermediate CA
- CAs and TLS CAs

Cert in TLS communication



- Secure communication between nodes using TLS
  - one-way(server only)
  - two-way(server-client)
- Peer node is both TLS server and TLS client
  - TLS server When another peer node, application, or CLI connect to it
  - TLS client when it connect to another peer or orderer
- TLS client authentication is turn off be default
- In channel, root CA certificate chain of channel members are read from config block are added to TLS client and server root Cas data structure. So, communication between peer -peer, peer-orderer are seamlessly.


TLS Client
• ClientHello
• ServerHello
• ServerCertificate
• ServerKeyExchange
• CertificateRequest
• ServerHelloDone
TLS Server
TLS Client
• ClientCertificate
• ClientKeyExchange
• CertificateVerify
• ChangeCipherSpec
• Finished
• ClientCertificate
• ClientKeyExchange
• CertificateVerify
• ChangeCipherSpec
• Finished
TLS Server

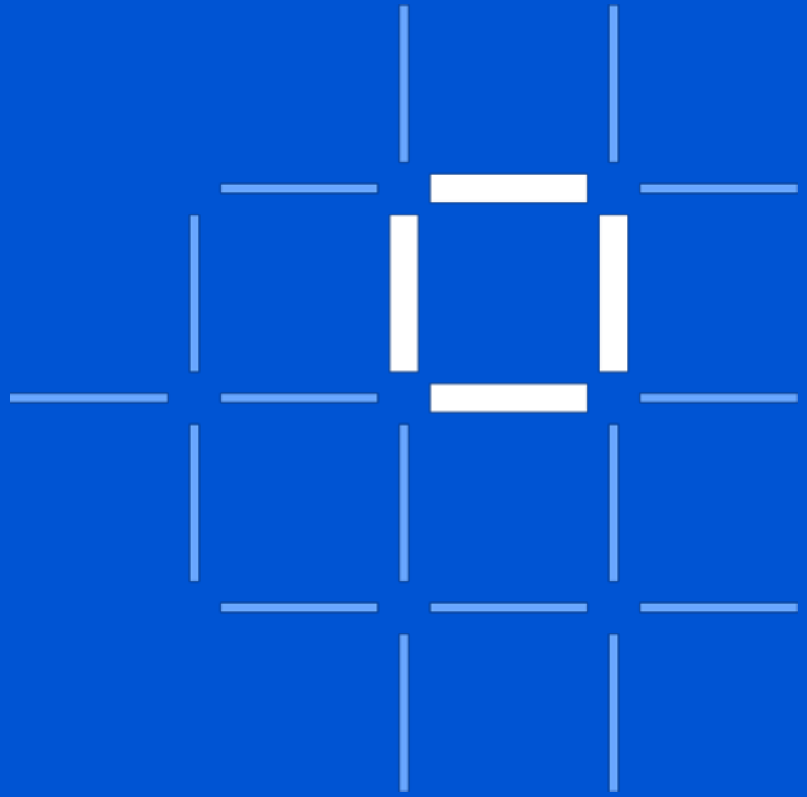
Fabric CA Overview

PKI – X.509

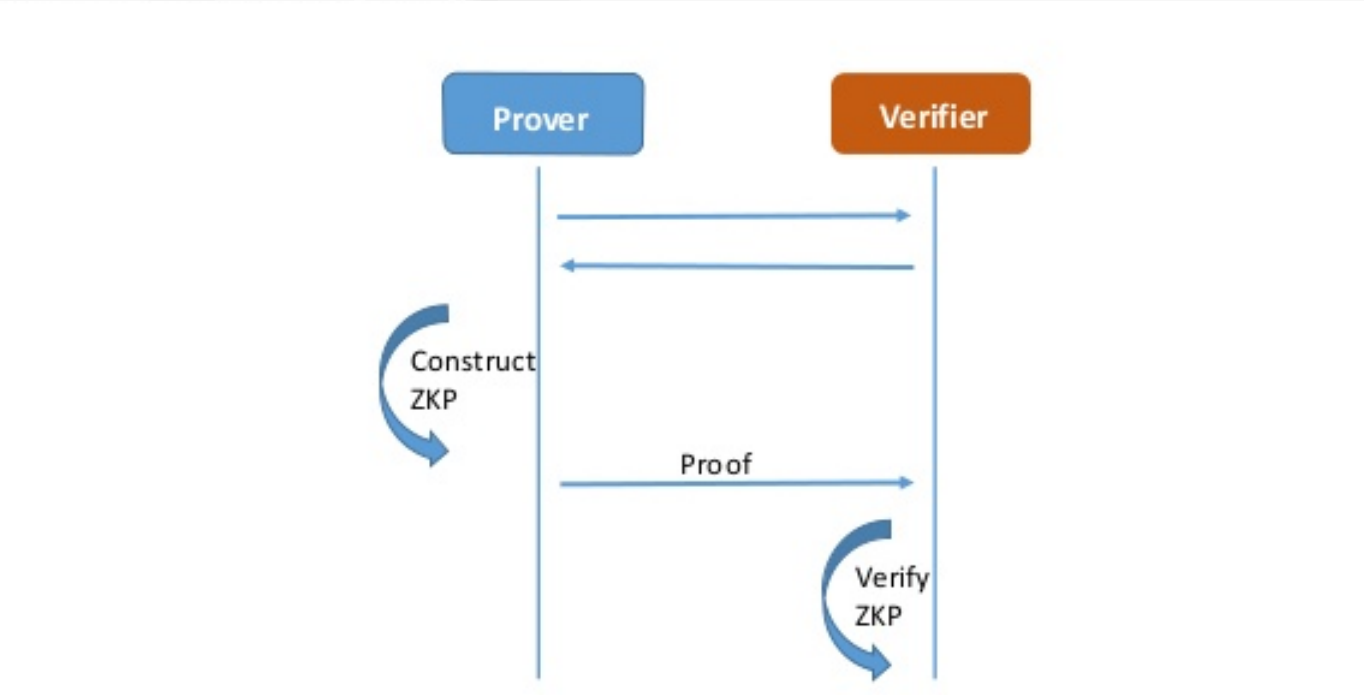
MSP structure and usage

Identity Mixer

Hardware Security Module (HSM) and advantages

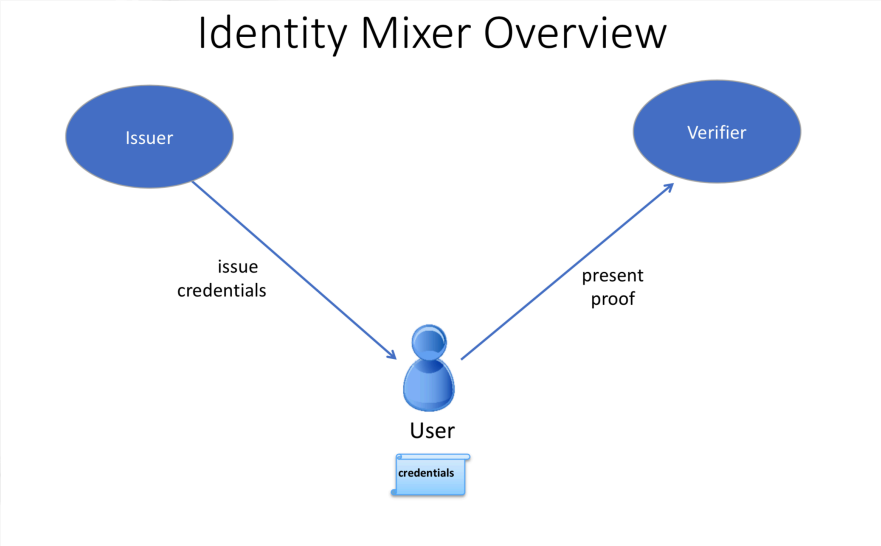


Zero Knowledge Proof

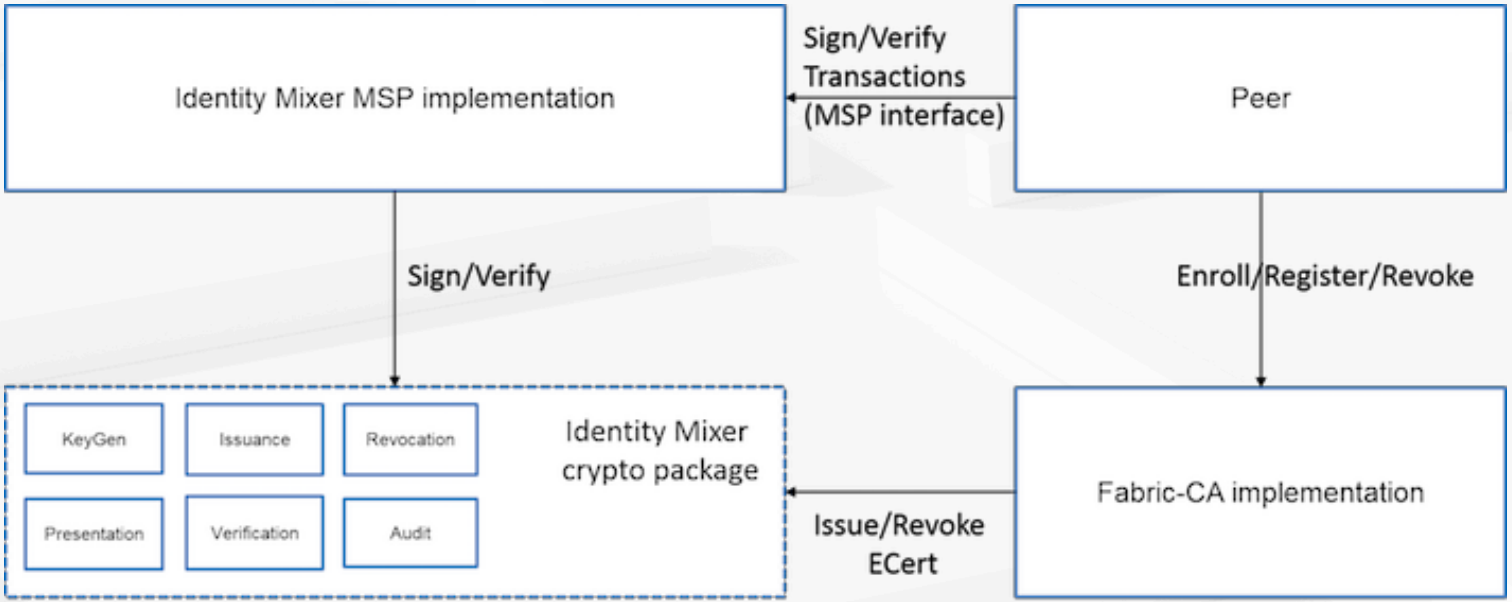


Identity Mixer(Idemix)

- Cryptographic protocol suite
  - Privacy-preserving as anonymity
  - Transact without revealing the identity of transactor
  - Unlinkability
- Three actors:
  - User
  - issue
  - verifier
- Idemix = X.509+cryptographic algorithm
- Generate Zero-knowledge Proof

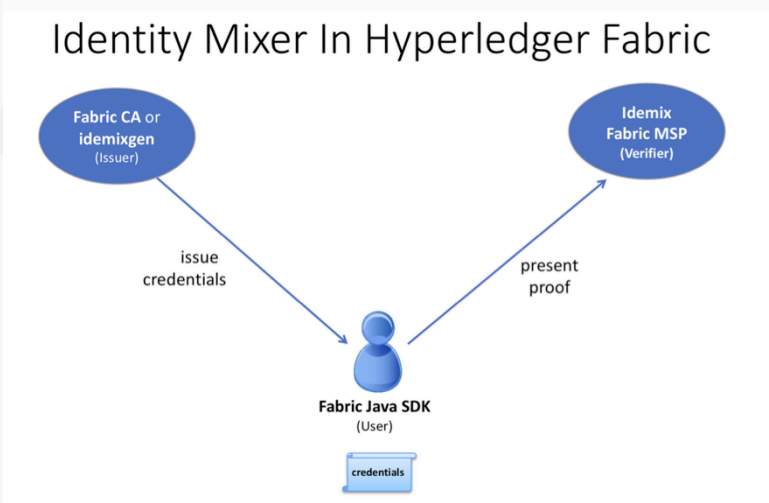


Identity Mixer for Hyperledger

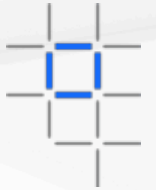


MSP with Identity Mixer in Fabric

- Issue idemix credentials in addition to X.509 certificate
  - IssuerPublicKey
  - IssuerRevocationPublicKey
- Generate Idemix credential
  - **OU**
  - **IsAdmin**
  - **Enrollment ID**

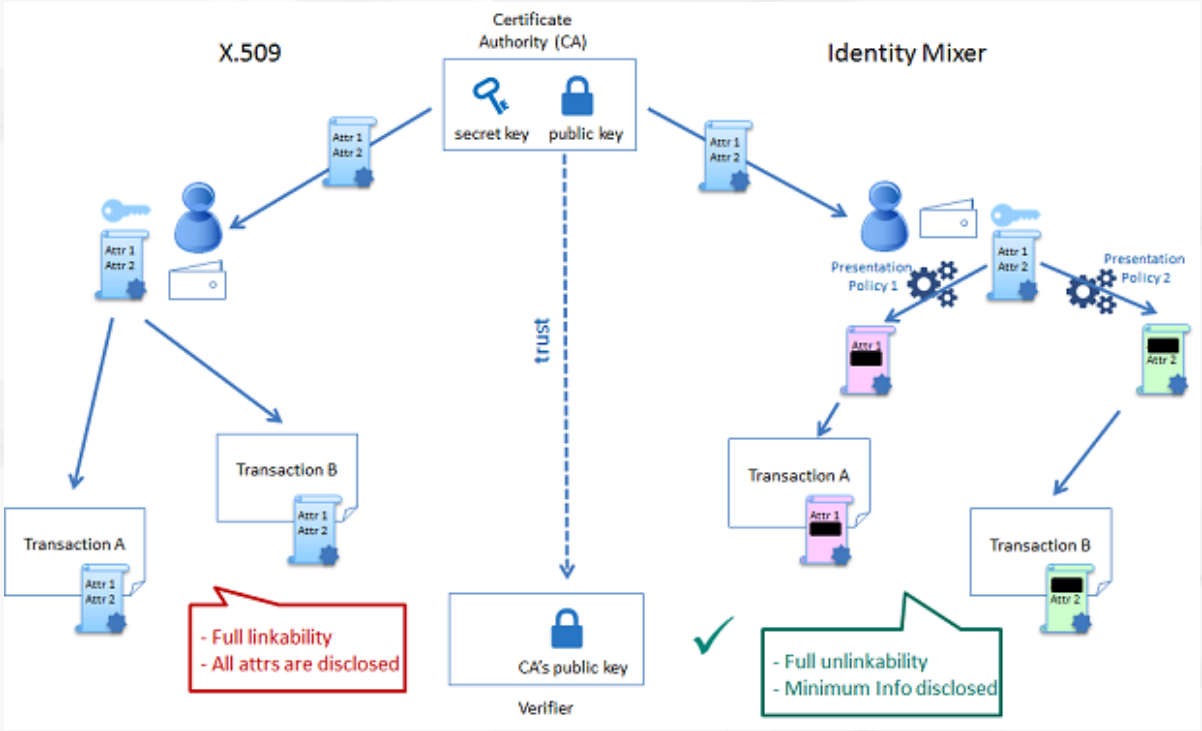


Idemix vs. X.509



	Same	Diff
Idemix	• A set of attributes signed with signature that can not be forged • Credential is cryptographically bounded to a secret key	• **zero-knowledge proofs to ensure 'knowledge' or 'information' is not revealed and user in possession of credential secret key** • **Non linkability**
X.509		• **Verified by public key that originally signed, only user know secret key can generate proofs.** • **All attributes are revealed, so all certificate usage for signing transactions are linkable**

X.509 vs. Idemix



X.509 vs. Idemix MSP difference in Fabric CA

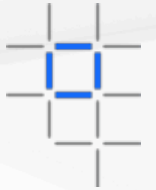
- organizational identities in configtx.yaml

```
- &Org1
    # DefaultOrg defines the organization which is used in the sampleconfig
    # of the fabric.git development environment
    Name: Org1MSP

    # ID to load the MSP definition as
    ID: Org1MSP

    MSPDir: crypto-config/peerOrganizations/org1.example.com/msp

    AnchorPeers:
        # AnchorPeers defines the location of peers which can be used
        # for cross org gossip communication.  Note, this value is only
        # encoded in the genesis block in the Application section context
        - Host: peer0.org1.example.com
          Port: 7051
```

```
- &Org1Idemix
    # defaultorg defines the organization which is used in the sampleconfig
    # of the fabric.git development environment
    name: idemixMSP1

    # id to load the
    id: idemixMSPID1

    msptype: idemix
    mspdir: crypto-config/peerOrganizations/org3.example.com
```

MSP Type

Limitation of Idemix



- Fixed set of attributes
  - OU
  - Role attribute
  - Enrollment ID
  - Revocation Handle attribute
- Revocation for Idemix is not supported
- Peers do not use Idemix for endorsement
  - Idemix MSP by peers only for signature verification
  - Signing with Idemix only via Client SDK
- Recommended one Idemix-based MSP per channel or per network as Idemix currently provides only anonymity of clients among the same organization (MSP).

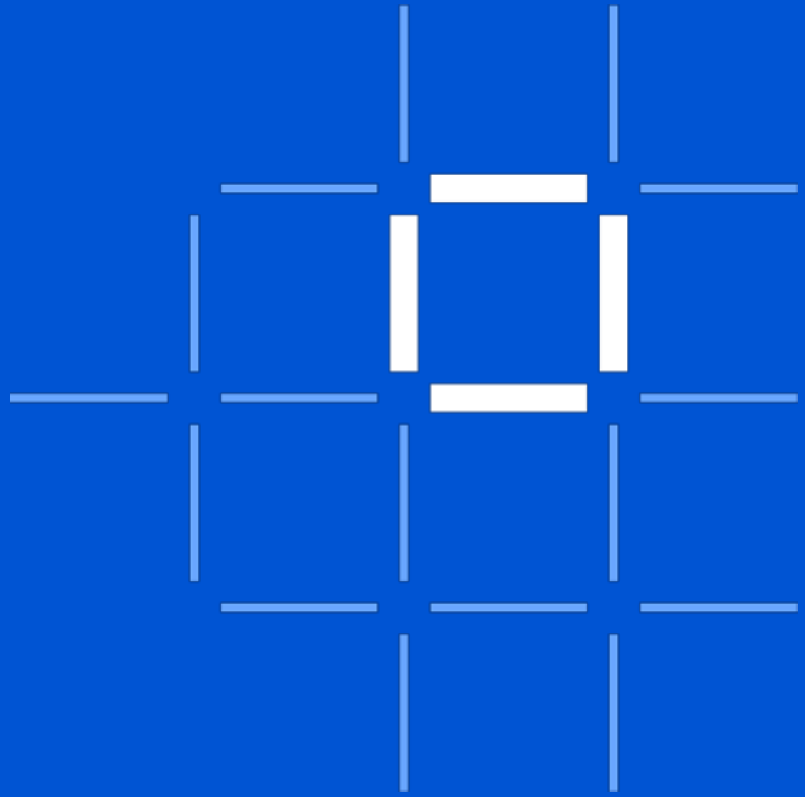
Fabric CA Overview

PKI – X.509

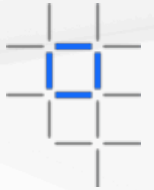
MSP structure and usage

Identity Mixer

Hardware Security Module (HSM) and advantages

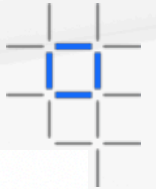


Hardware Security Module(HSM)



- Store private keys in an HSM (Hardware Security Module) via PKCS11 APIs
  - PEM-encoded file
  - PKCS#11 – Public-key Cryptography Standards
- IBM Hardware Security Module
  - FIPS 140-2 Level 4
  - tamper-resistant, tamper-evident
  - 20,000 ECC and 10,000 RSA per second
  - Lower latency
- Configured in the BCCSP in Fabric CA

HSM Plugin In BCCSP



```
###############################################################################
# BCCSP (BlockChain Crypto Service Provider) section is used to select which
# crypto library implementation to use
###############################################################################
bccsp:
    default: PKCS11
    sw:
        hash: SHA2
        security: 256
        filekeystore:
            # The directory used for the software file-based keystore
            keystore: msp/keystore
    pkcs11:
        library: /usr/lib/s390x-linux-gnu/opencryptoki/libopencryptoki.so.0.0.0
        pin: 98765432
        label: CEX6P
        hash: SHA2
        security: 256
        filekeystore:
            keystore: msp/keystore
```

课程安排

03/14 区块链赋能产业价值和商业模式
03/21 Hyperledger 项目概览 社区介绍
03/28 Fabric 1.4 LTS 功能介绍 架构概览
04/04 Peer 解析
04/11 Orderer 解析
04/18 MSP 与 CA
04/25 应用开发指南
05/09 部署实践

欢迎关注微信公众号
“IBM开源技术”
获取更多资讯

公众号中发送**“replay”**
获取往期视频地址

公众号中发送“**报名**”,
即有机会参加Fabric线下训练营