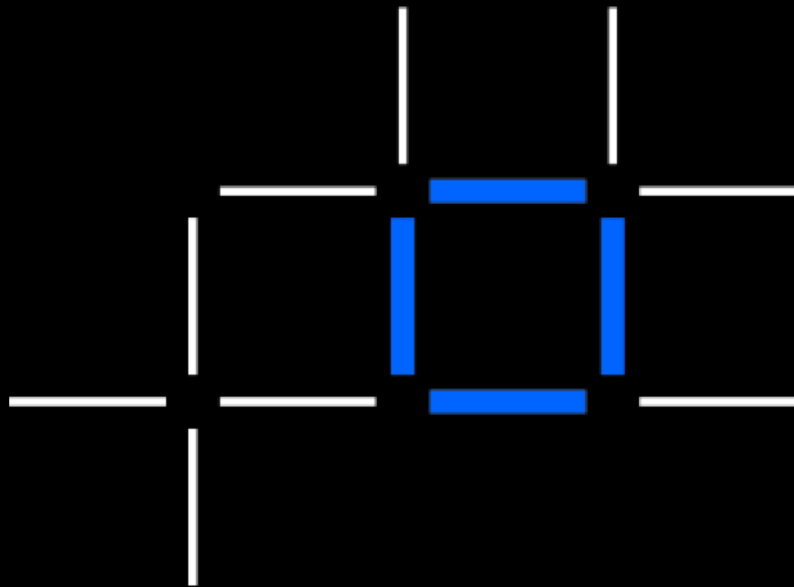


Blockchain Explored, Deep Dive Part 2

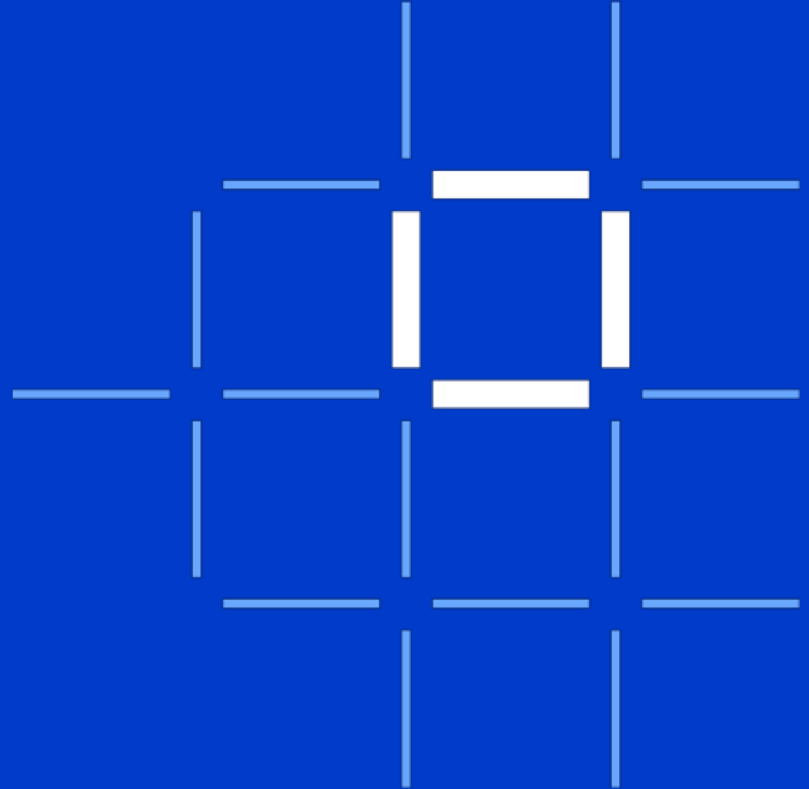
A Technical Deep-Dive on Hyperledger Fabric

Barry Silliman
Blockchain Enablement on IBM Z and LinuxONE
IBM Washington Systems Center
silliman@us.ibm.com



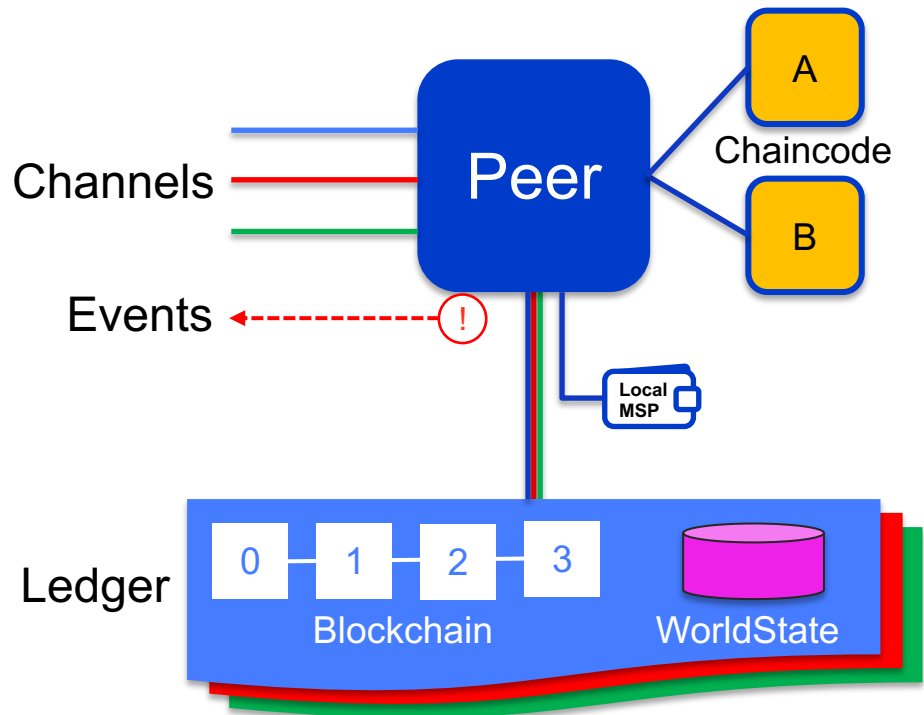
Technical Deep Dive

- Architectural Overview]
- Network Consensus
- Channels and Ordering Service
- **[Components]**
- **Network setup**
- **Endorsement Policies**
- **Membership Services**

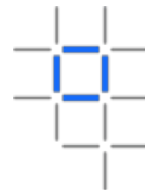


Fabric Peer

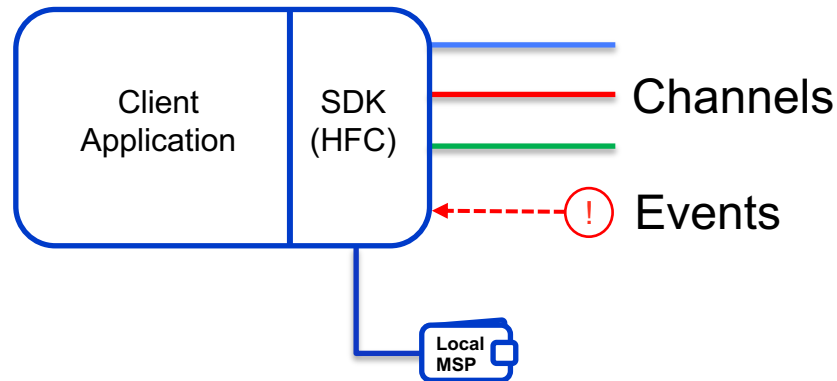
- Each peer:
 - Connects to one or more **channels**
 - Maintains one or more **ledgers** per channel
 - Maintains **installed chaincode**
 - Manages **runtime docker containers** for **instantiated chaincode**
 - Chaincode is instantiated on a channel
 - Runtime docker container shared by channels with same chaincode instantiated (no state stored in container)
 - Has a local MSP (Membership Services Provider) that provides **crypto material**
 - **Emits events** to the client application



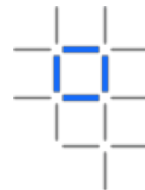
Client Application



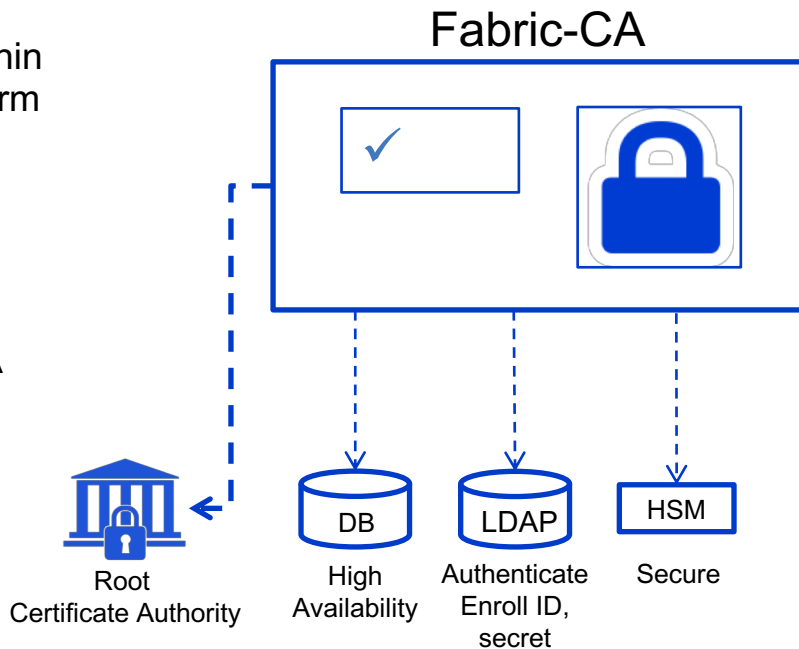
- Each client application uses Fabric SDK to:
 - Connects over channels to one or more peers
 - Connects over channels to one or more orderer nodes
 - Receives events from peers
 - Local MSP provides client **crypto material**
- Client can be written in different languages (Node.js, Go, Java, Python?)



Fabric-CA



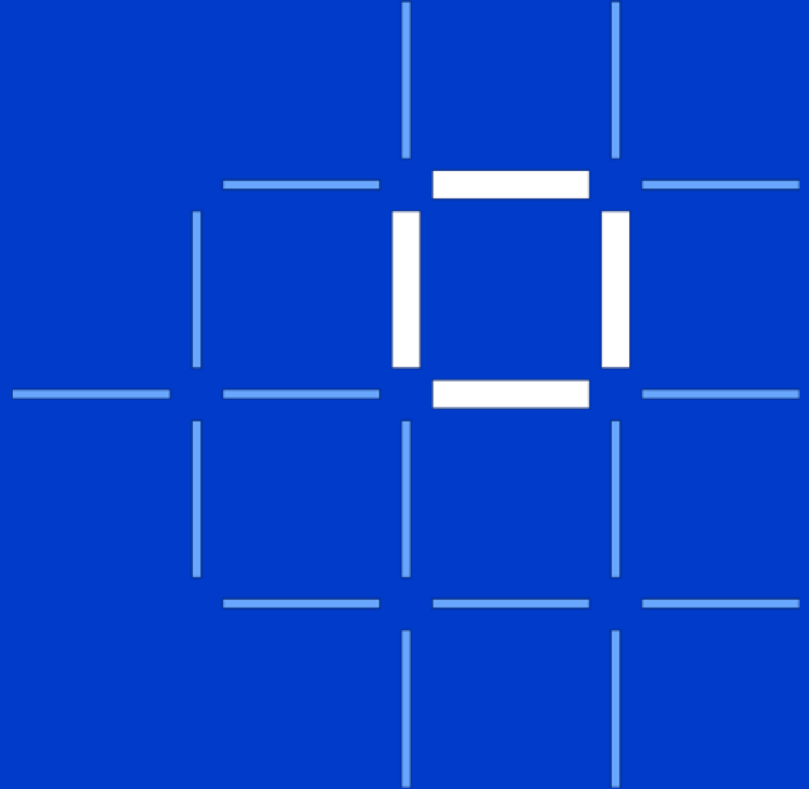
- Default (optional) Certificate Authority within Fabric network for issuing **Ecerts** (long-term identity)
- Supports clustering for **HA characteristics**
- Supports LDAP for **user authentication**
- Supports HSM for **security**
- Can be configured as an intermediate CA



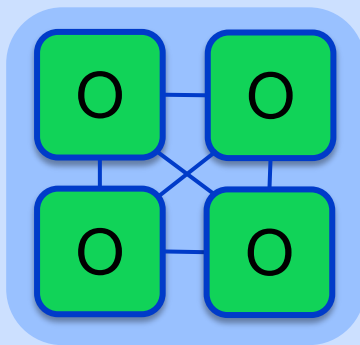
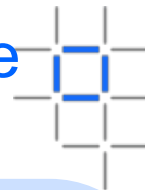


Technical Deep Dive

- Architectural Overview
- Network Consensus
- Channels and Ordering Service
- **Components**
- **[Network setup]**
- **Endorsement Policies**
- **Membership Services**



Bootstrap Network (1/6) - Configure & Start Ordering Service



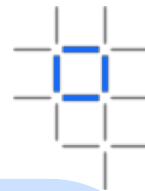
Ordering-Service

Hyperledger Fabric Network

An Ordering Service is configured and started for the network:

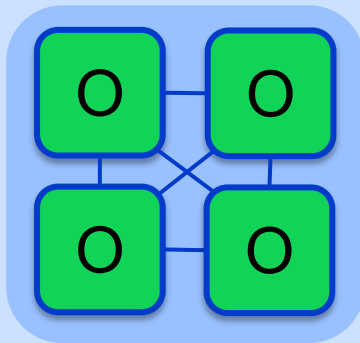
\$ docker-compose [-f orderer.yml] ...

Bootstrap Network (2/6) - Configure and Start Peer Nodes



E_0

E_1



Ordering-Service

E_2

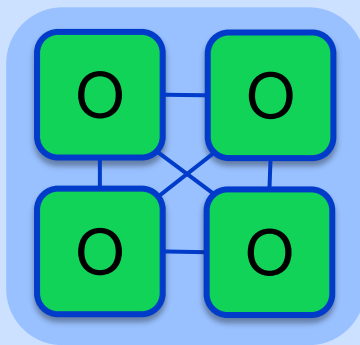
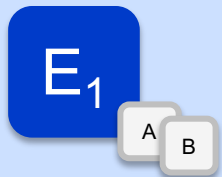
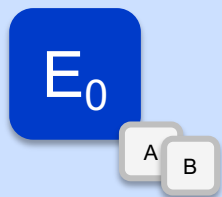
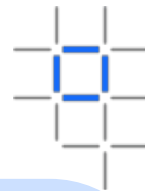
P_3

Hyperledger Fabric Network

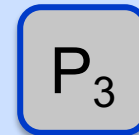
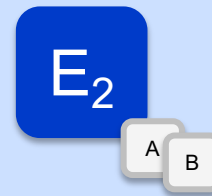
A peer is configured and started for each Endorser or Committer in the network:

\$ peer node start ...

Bootstrap Network (3/6) - Install Chaincode



Ordering-Service

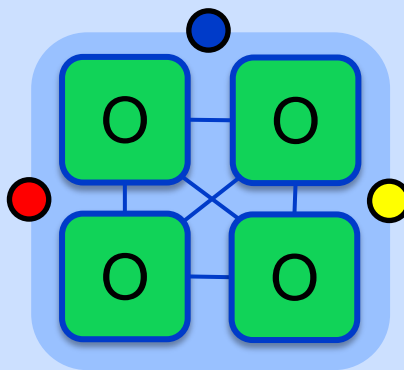
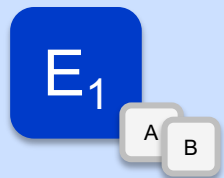
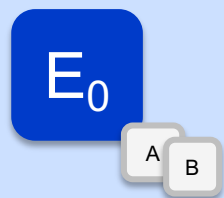
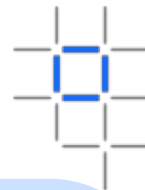


Hyperledger Fabric Network

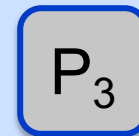
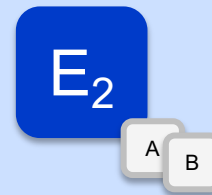
Chaincode is installed onto each Endorsing Peer that needs to execute it:

\$ peer chaincode install ...

Bootstrap Network (4/6) – Create Channels



Ordering-Service

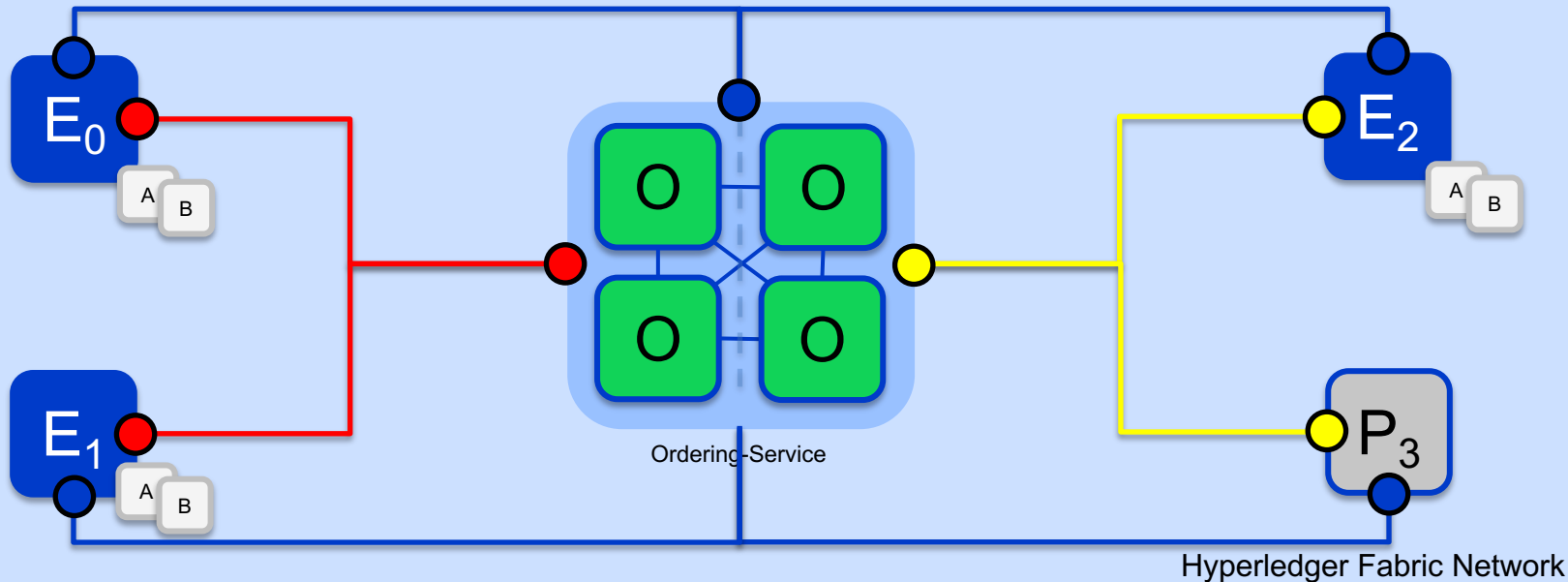
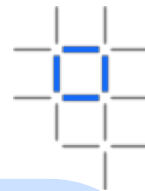


Hyperledger Fabric Network

Channels are created on the ordering service:

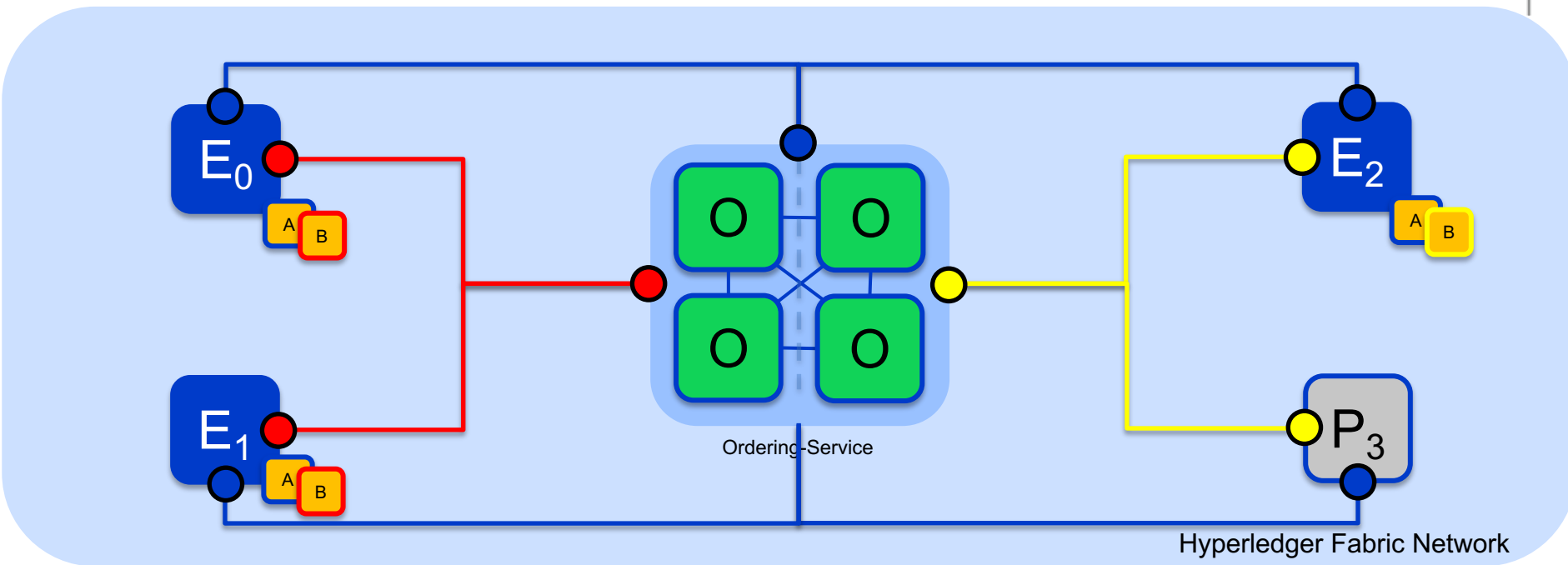
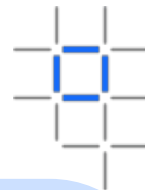
\$ peer channel create -o [orderer] ...

Bootstrap Network (5/6) – Join Channels



Peers that are permitted can then join the channels they want to transact on:
\$ peer channel join ...

Bootstrap Network (6/6) – Instantiate Chaincode

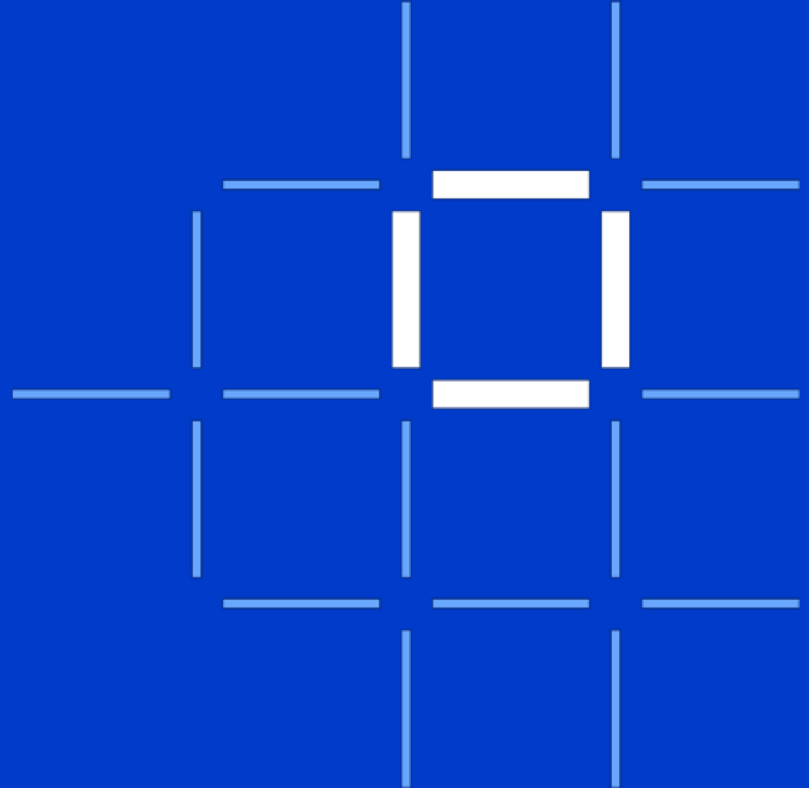


Peers finally instantiate the Chaincode on the channels they want to transact on:
\$ peer chaincode instantiate ... -P 'policy'

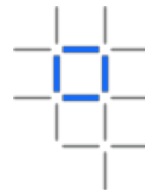


Technical Deep Dive

- Architectural Overview
- Network Consensus
- Channels and Ordering Service
- **Components**
- **Network setup**
- **[Endorsement Policies]**
- **Membership Services**

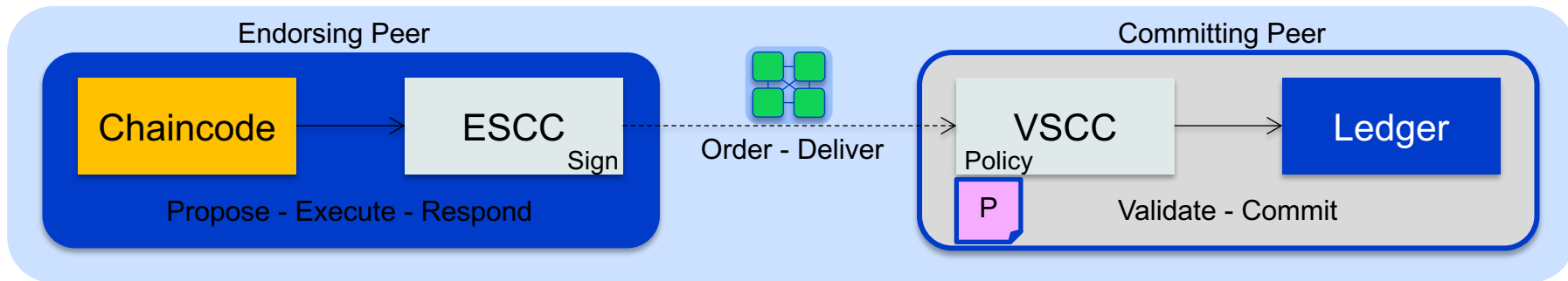


Endorsement Policies

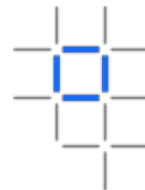


An endorsement policy describes the conditions by which a transaction can be endorsed. A transaction can only be considered valid if it has been endorsed according to its policy.

- Each chaincode is deployed with an Endorsement Policy
- **ESCC** (Endorsement System ChainCode) signs the proposal response on the endorsing peer
- **VSCC** (Validation System ChainCode) validates the endorsements



Endorsement Policy Syntax



```
$ peer chaincode instantiate  
-C mychannel  
-n mycc  
-v 1.0  
-p chaincode_example02  
-c '{"Args":["init","a", "100", "b","200"]}'  
-P "AND('Org1MSP.member')"
```

Instantiate the chaincode **mycc** on channel **mychannel** with the policy **AND('Org1MSP.member')**

Policy Syntax: **EXPR(E[, E...])**

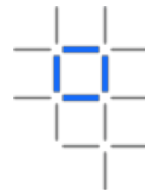
Where **EXPR** is either AND or OR and **E** is either a principal or nested EXPR

Principal Syntax: **MSP.ROLE**

Supported roles are: member and admin

Where **MSP** is the MSP ID, and **ROLE** is either “member” or “admin”

Endorsement Policy Examples



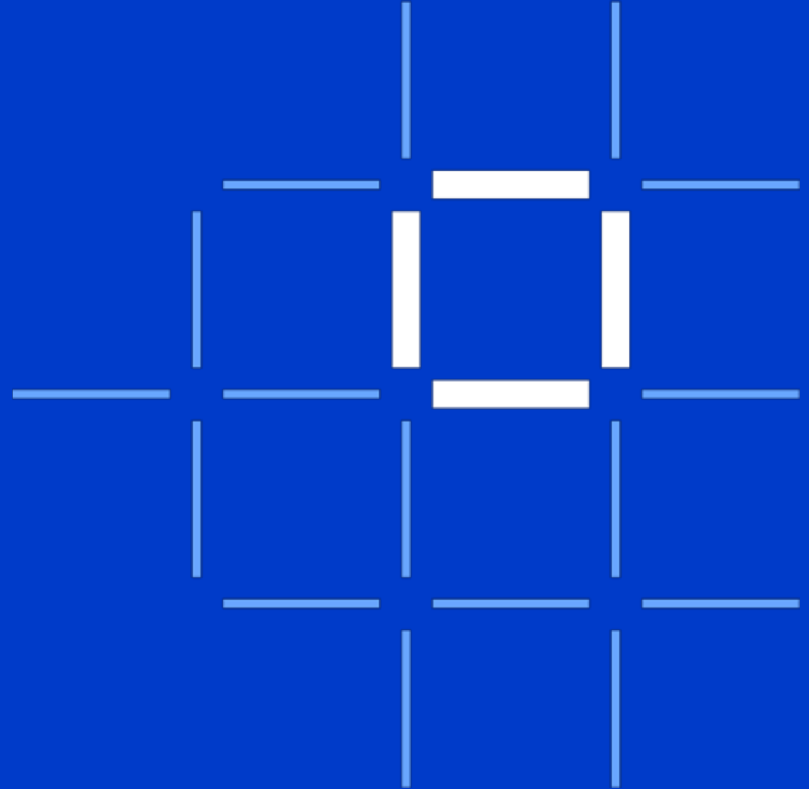
Examples of policies:

- Request 1 signature from all three principals
 - `AND('Org1.member', 'Org2.member', 'Org3.member')`
- Request 1 signature from either one of the two principals
 - `OR('Org1.member', 'Org2.member')`
- Request either one signature from a member of the Org1 MSP or (1 signature from a member of the Org2 MSP and 1 signature from a member of the Org3 MSP)
 - `OR('Org1.member', AND('Org2.member', 'Org3.member'))`

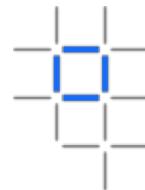


Technical Deep Dive

- Architectural Overview
- Network Consensus
- Channels and Ordering Service
- **Components**
- **Network setup**
- **Endorsement Policies**
- **[Membership Services]**

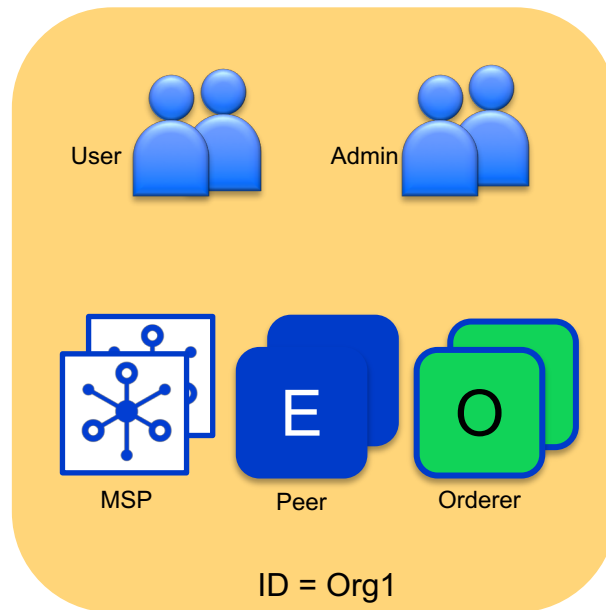


Organizations

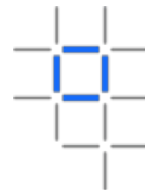


Organizations define boundaries within a Fabric Blockchain Network

- Each organization defines:
 - Membership Services Provider (MSP) for identities
 - Administrator(s)
 - Users
 - Peers
 - Orderers (optional)
- A network can include many organizations representing a consortium
- Each organization has a unique ID

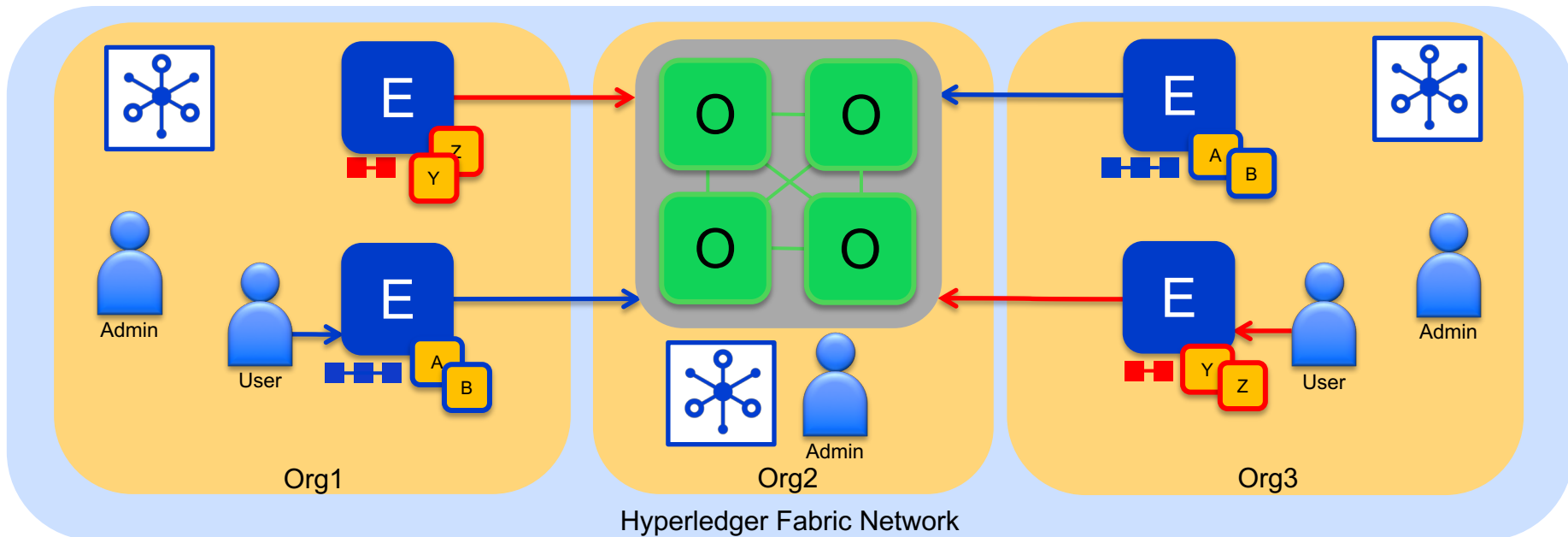


Consortium Network

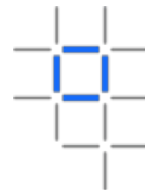


An example consortium network of 3 organisations

- Orgs 1 and 3 run peers
- Org 2 provides the ordering service only

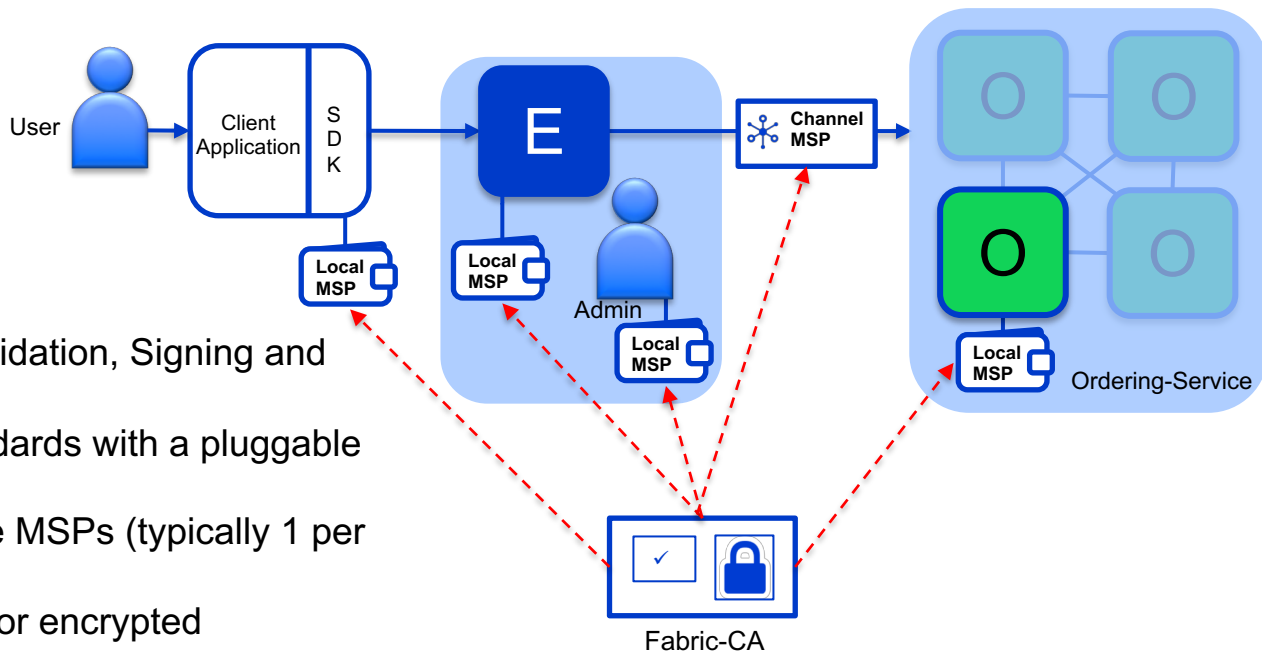


Membership Services Provider - Overview

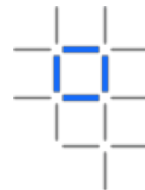


A MSP manages a set of identities within a distributed Fabric network

- Provides identity for:
 - Peers and Orderers
 - Client Applications
 - Administrators
- Identities can be issued by:
 - Fabric-CA
 - An external CA
- Provides: Authentication, Validation, Signing and Issuance
- Supports different crypto standards with a pluggable interface
- A network can include multiple MSPs (typically 1 per org)
- Includes TLS crypto material for encrypted communications

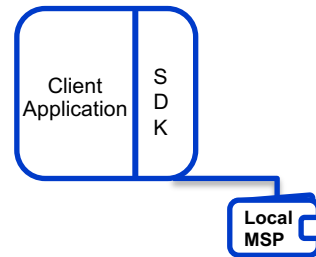


User Identities



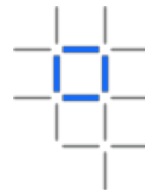
Each client application has a local MSP to store user identities

- Each local MSP includes:
 - **Keystore**
 - **Private key** for signing transactions
 - **Signcert**
 - **Public x.509 certificate**
- May also include TLS credentials
- Can be backed by a Hardware Security Module (HSM)



user@org1.example.com	
keystore	<private key>
signcert	user@org1.example.com-cert.pem

Admin Identities



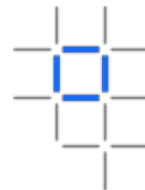
Each Administrator has a local MSP to store their identity

- Each local MSP includes:
 - **Keystore**
 - **Private key** for signing transactions
 - **Signcert**
 - **Public x.509 certificate**
- May also include TLS credentials
- Can be backed by a Hardware Security Module (HSM)



admin@org1.example.com	
keystore	<private key>
signcert	admin@org1.example.com-cert.pem

Peer and Orderer Identities



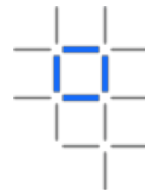
Each peer and orderer has a local MSP

- Each local MSP includes:
 - **keystore**
 - **Private key** for signing transactions
 - **signcert**
 - **Public x.509 certificate**
- In addition Peer/Orderer MSPs identify authorized administrators:
 - **admincerts**
 - List of **administrator certificates**
 - **cacerts**
 - The **CA public cert** for verification
 - **crls**
 - List of **revoked certificates**
- Peers and Orderers also receive channel MSP info
- Can be backed by a Hardware Security Module (HSM)



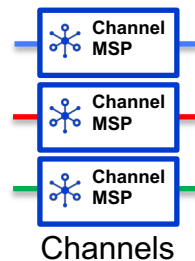
peer@org1.example.com	
admincerts	admin@org1.example.com-cert.pem
cacerts	ca.org1.example.com-cert.pem
keystore	<private key>
signcert	peer@org1.example.com-cert.pem
crls	<list of revoked admin certificates>

Channel MSP information



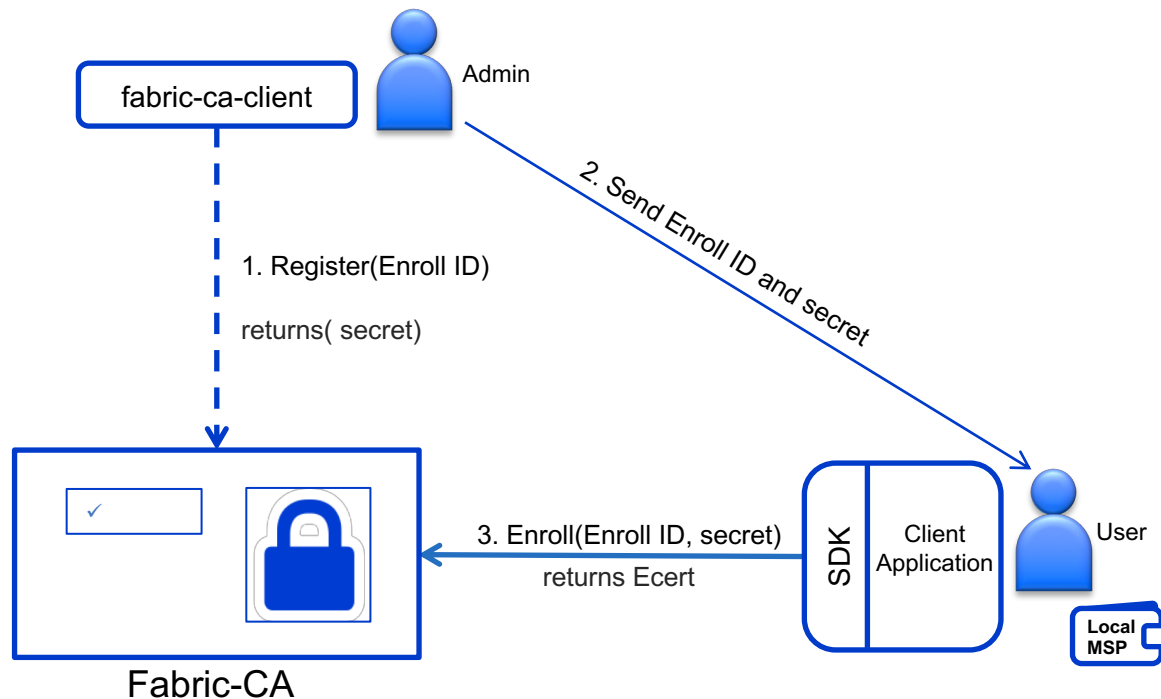
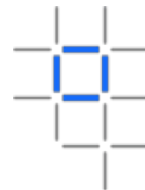
Channels include additional organizational MSP information

- Determines which orderers or peers can join the channel
- Determines client applications read or write access to the channel
- Stored in configuration blocks in the ledger
- Each channel MSP includes:
 - **admincerts**
 - Any public certificates for administrators
 - **cacerts**
 - The CA public certificate for this MSP
 - **crls**
 - List of revoked certificates
- Does not include any private keys for identity



ID = MSP1	
admincerts	admin.org1.example.com-cert.pem
cacerts	ca.org1.example.com-cert.pem
crls	<list of revoked admin certificates>

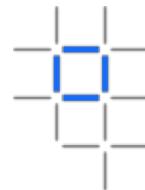
New User Registration and Enrollment



Registration and Enrollment

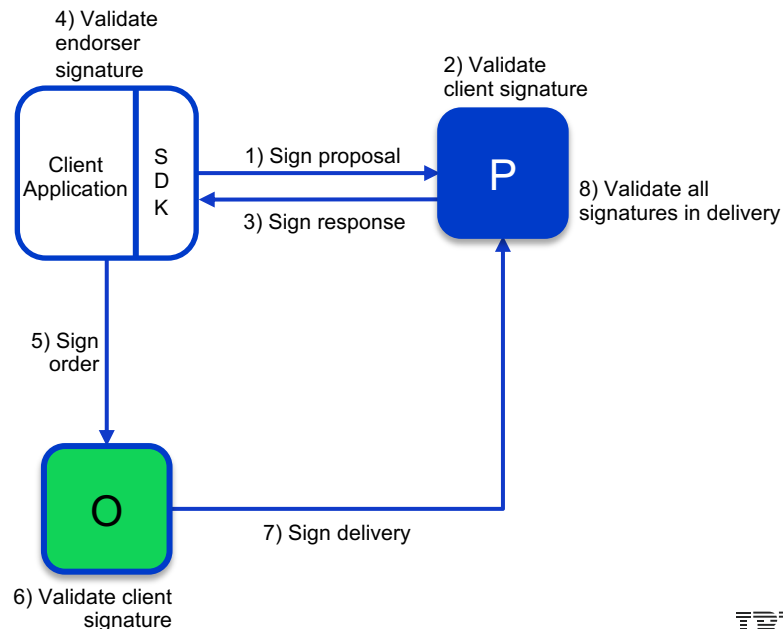
- Admin registers new user with Enroll ID
- User enrolls and receives credentials
- Additional offline registration and enrollment options available

Transaction Signing

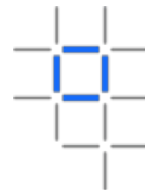


All transactions within a Hyperledger Fabric network are signed by permissioned actors, and those signatures validated

- Actors sign transactions with their enrolment private key
 - Stored in their local MSP
- Components validate transactions and certificates
 - Root CA certificates and CRLs stored in local MSP
 - Root CA certificates and CRLs stored in Org MSP in channel



Further Hyperledger Fabric Information



- Project Home: <https://www.hyperledger.org/projects/fabric>
- GitHub Repo: <https://github.com/hyperledger/fabric>
- Latest Docs: <https://hyperledger-fabric.readthedocs.io/en/latest/>
- Community Chat: <https://chat.hyperledger.org/channel/fabric>
- Project Wiki: <https://wiki.hyperledger.org/projects/fabric>
- Design Docs: <https://wiki.hyperledger.org/community/fabric-design-docs>

Thank you

Barry Silliman
silliman@us.ibm.com

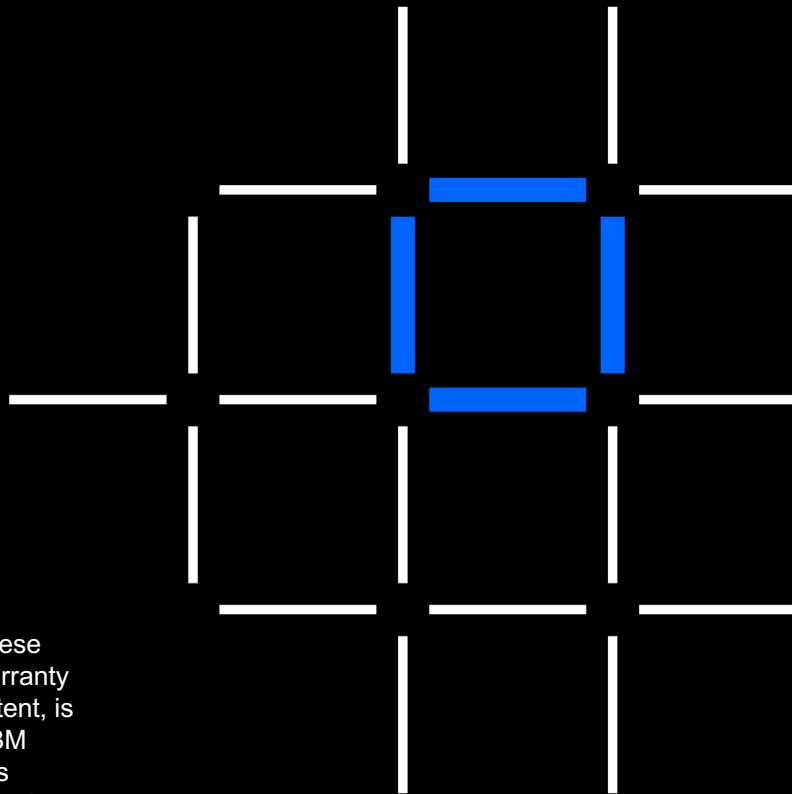
IBM Blockchain

www.ibm.com/blockchain

developer.ibm.com/blockchain

www.hyperledger.org

© Copyright IBM Corporation 2017. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represents only goals and objectives. IBM, the IBM logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.



IBM