The UNIVERSITY of NORTH CAROLINA at CHARLOTTE

College of Computing and Informatics
Department of Software and Information Systems

Hyperledger Fabric Installation

Advisor: Dr. Yongge Wang
Ph.D. Student: Ahmed Al Salih

HYPERLEDGER FABRIC

Hands on binaries

1- **cryptogen** Tool
generate the cert

2- **configtxgen** Tool
Generate Genesis Block
Channel transaction
Inspect channel Transaction

```
configtxgen -outputCreateChannelTx ./first-channel.tx -profile -channelID myChannel
```

The **Tx file** is used by peer for submitting transaction

3- **peer** Tool:
Smart Contract
peer
Channel

```
Peer channel getinfo –c mychannel
```

HYPERLEDGER FABRIC

Orderer binary

Genesis Block needed for intialization

HYPERLEDGER FABRIC

Orderer binary

Orderer Binary depends on the use of crypto service providers

Orderer

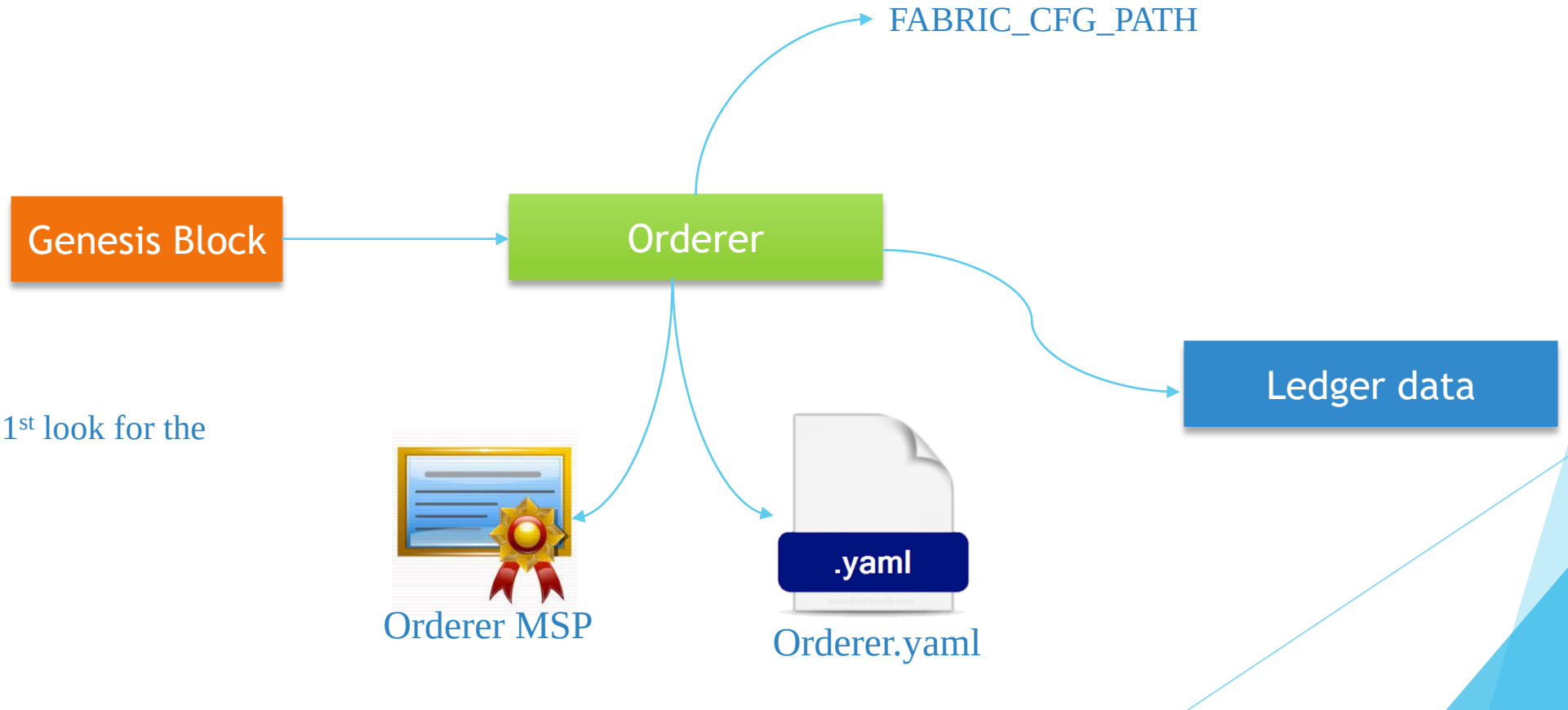
The cryptographic functions are not built in the binary

Crypto
Service Provider

- Encrypt
- Decrypt
- Message signing

Orderer binary

Genesis Block needed for initialization



1st look for the

Orderer binary

Control Model level logging
FATAL | PANIC | ERROR | WARNING | INFO | DEBUG
export FABRIC_LOGGING_SPEC=INFO

Control logging format:
The Default logging format

"%{color}%{time:2006-01-02 15:04:05.000 MST} [%{module}] %{shortfunc} -> %{level:.4s} %{id:03x}%{color:reset} %{message}"

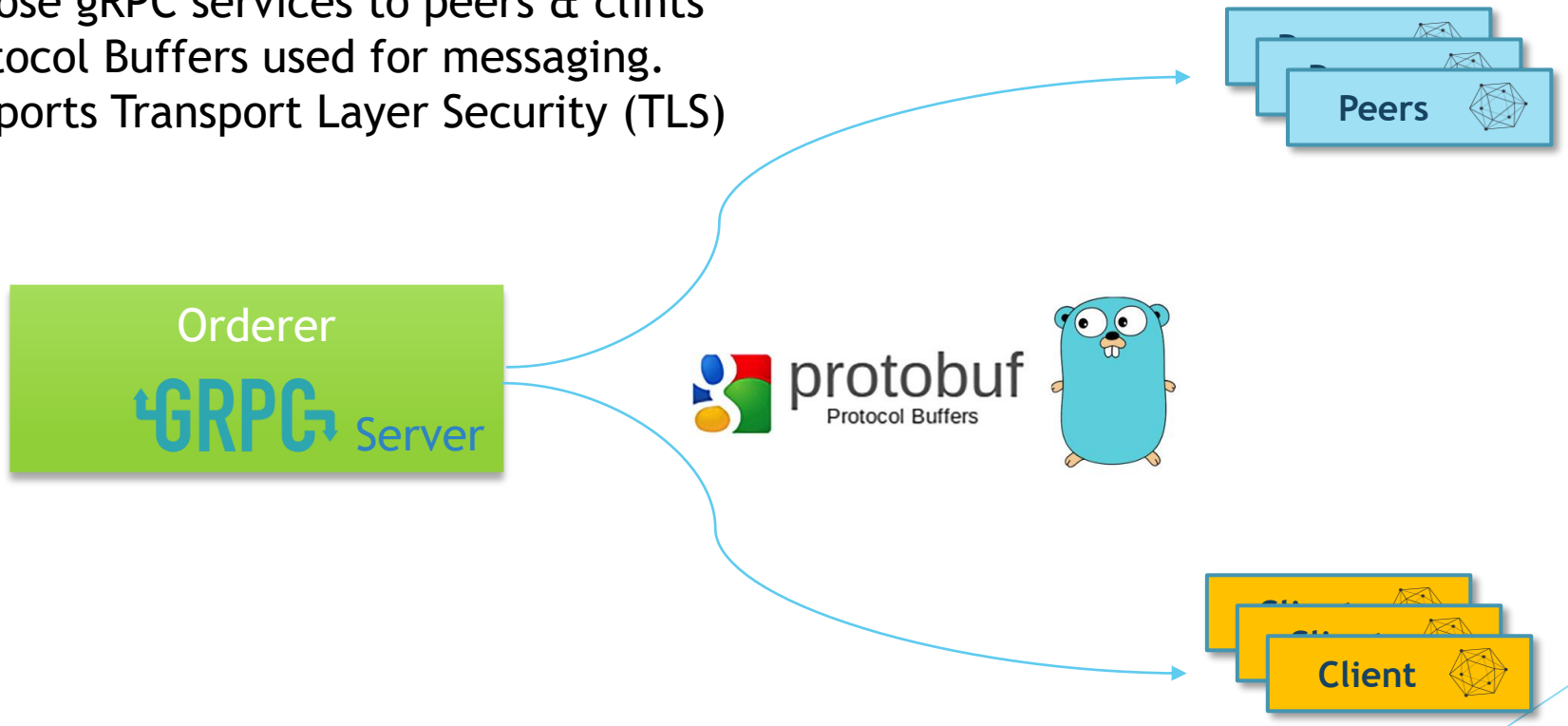
To use custom loggin format:
export FABRIC_LOGGING_FORMAT=

https://hyperledger-fabric.readthedocs.io/en/release-2.2/logging-control.html

HYPERLEDGER FABRIC

Orderer binary

- Expose gRPC services to peers & clints
- Protocol Buffers used for messaging.
- Supports Transport Layer Security (TLS)



HYPERLEDGER FABRIC

Orderer binary summery

- Needs Genesis Block for initialization
- Runtime needs access to  Orderer MSP 

Orderer.yaml
- Expose GRPC it might be secured using TLS
- Writes to Memory or Filesystem

HYPERLEDGER FABRIC

Overview Orderer.yaml

Orderer.yaml

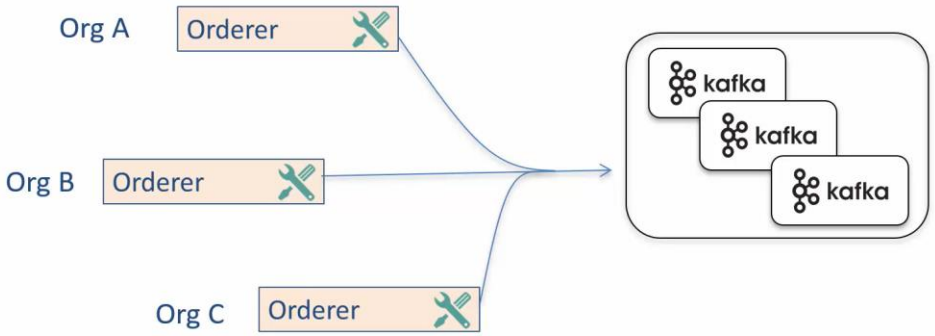
- General: General properties of the Orderer.
- FileLedger: Filesystem location of the block data.
- Consensus: Used for managing storage for Orderer type **etcdraft**
- Kafka: kafka environment setup.
- Debug: Debug information control.
- Operations: Operation server info (network monitoring| alerts) & TLS configuration for the operations server..
- Metrics: Orderer generate metrics info, collected by 3ed party. Metrics provider is one of statesd, Prometheus, or disabled

All configuration can be overridden at runtime by setting the environment variables:
ORDERER_GENERAL_...
ORDERER_FILELEDGER_...

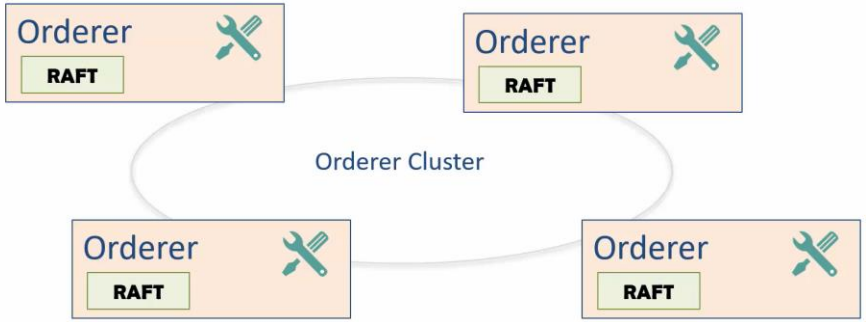
HYPERLEDGER FABRIC

Orderer binary

- Multiple Orderer instances connect to Kafka cluster



- RAFT is built into the binary



HYPERLEDGER FABRIC

Overview Orderer.yaml

Orderer.yaml

```yaml
Debug:

    # BroadcastTraceDir when set will cause each request to the Broadcast service
    # for this orderer to be written to a file in this directory
    BroadcastTraceDir:

    # DeliverTraceDir when set will cause each request to the Deliver service
    # for this orderer to be written to a file in this directory
    DeliverTraceDir:
```

HYPERLEDGER FABRIC

Overview Orderer.yaml

Orderer.yaml

Crypto Service provider (CSP)

CSP expose the cryptographic functions:

Encrypt
Decrypt
Key Pair generation
Private key security
Message Digest
…

Implementations:

- Software CSP
  - Implemented as software libraries
    such as windows DLL or Linux shared object
- Hardware CSP
  - Hardware Security Modules (HSM)
  - Smart Cards

PKCS#11 is a platform independent CSP API

HYPERLEDGER FABRIC

Overview Orderer.yaml

Orderer.yaml

Crypto Service provider (CSP)

Orderer

- Software CSP
- PKCS#11 CSP

HYPERLEDGER FABRIC

Overview Orderer.yaml

Orderer.yaml

Crypto Service provider (CSP)

BCCSP: Blockchain Crypto Service Provider

Default: SW #For Software CSP

Default: PKCS11 #For Hardware CSP

```
General:
    # BCCSP configures the blockchain crypto service
    BCCSP:
        # Default specifies the preferred blockchain
        # to use. If the preferred provider is not a
        # based provider ("SW") will be used.
        # Valid providers are:
        #  - SW: a software based crypto provider
        #  - PKCS11: a CA hardware security module c
        Default: SW

        # SW configures the software based blockchai
        SW:
            # TODO: The default Hash and Security le
            # fully configurable. Changing these def
            # SHA2 is hardcoded in several places, r
            Hash: SHA2
            Security: 256
            # Location of key store. If this is unse
            # chosen using: 'LocalMSPDir'/keystore
            FileKeyStore:
                KeyStore:

        # Settings for the PKCS#11 crypto provider
        PKCS11:
```

Default: SW

SW:

HASH: #Hashing algorithm

Security: #key size

FileKeyStore: #Location of the keystore

KeyStore: #Default to LocalMSPDir/keystore

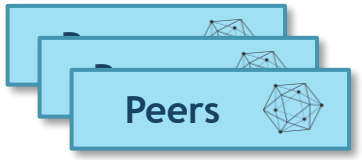
HYPERLEDGER FABRIC

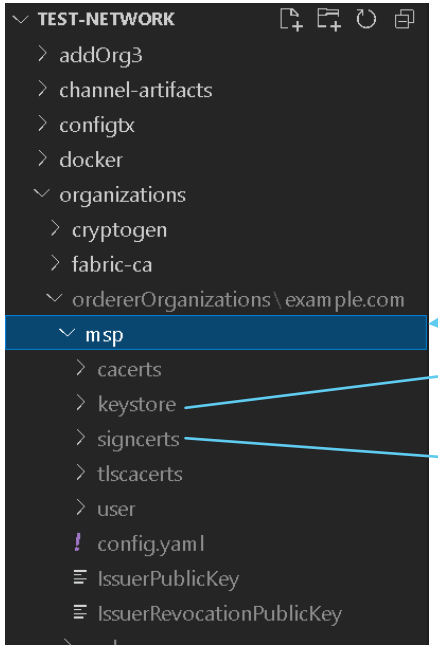
MSP

Orderer.yaml

Orderer

Private key





Private key

x.509

HYPERLEDGER FABRIC

MSP

Orderer.yaml

General:
    LocalMSPDir:
    LocalMSPID:

```yaml
General:
    # LocalMSPDir is where to find the private crypto material needed by the
    # orderer. It is set relative here as a default for dev environments but
    # should be changed to the real location in production.
    LocalMSPDir: msp

    # LocalMSPID is the identity to register the local MSP material with the MSP
    # manager. IMPORTANT: The local MSP ID of an orderer needs to match the MSP
    # ID of one of the organizations defined in the orderer system channel's
    # /Channel/Orderer configuration. The sample organization defined in the
    # sample configuration provided has an MSP ID of "SampleOrg".
    LocalMSPID: SampleOrg
```

HYPERLEDGER FABRIC

Overview Orderer.yaml

Orderer.yaml

- ListenAddress
- ListenPort:

```yaml
General:
    # Listen address: The IP on which to bind to listen.
    ListenAddress: 127.0.0.1

    # Listen port: The port on which to bind to listen.
    ListenPort: 7050
```

HYPERLEDGER FABRIC

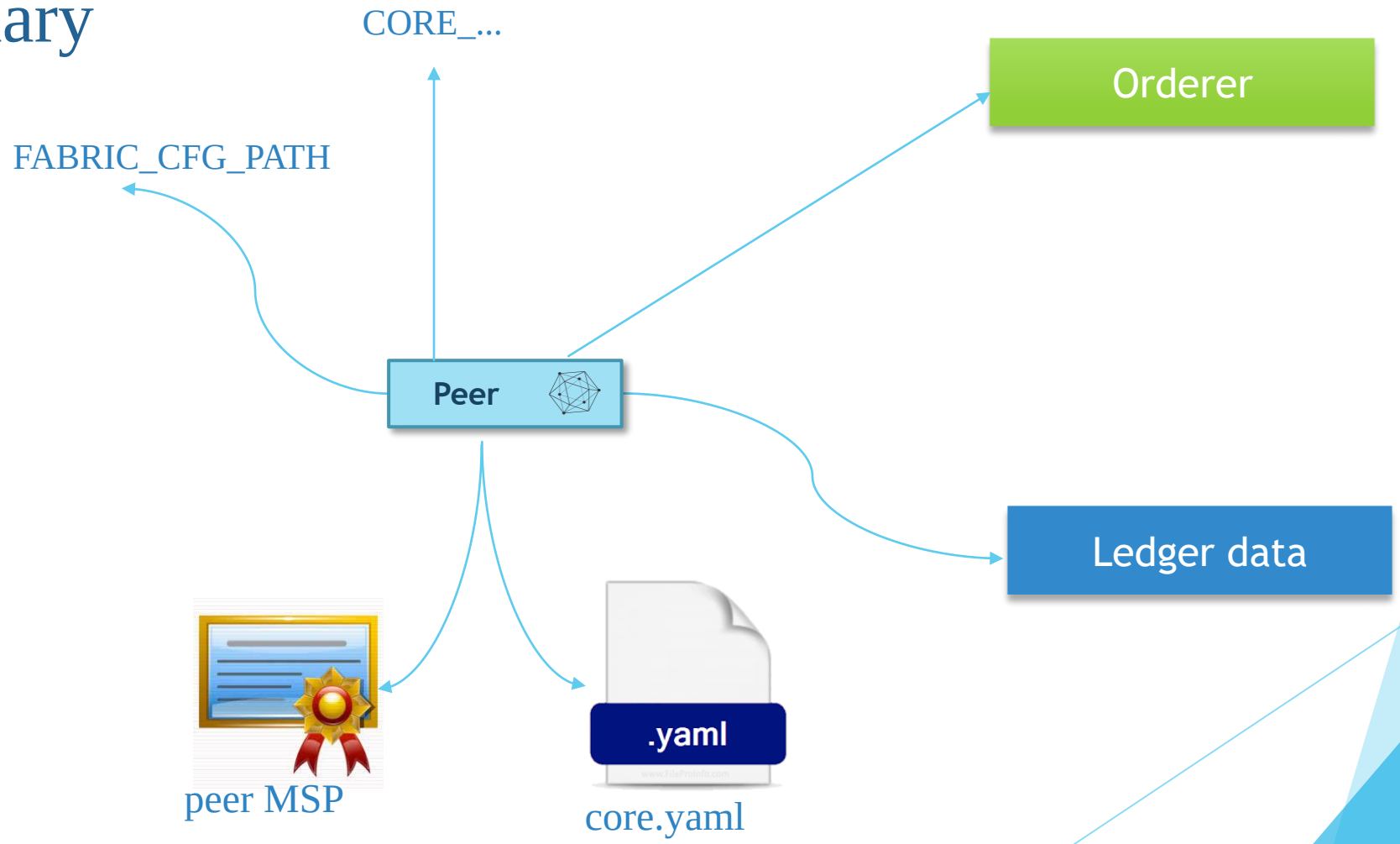
Overview Orderer.yaml

Orderer.yaml

```yaml
# Keepalive settings for the GRPC server.
Keepalive:
    # ServerMinInterval is the minimum permitted time between client pings.
    # If clients send pings more frequently, the server will
    # disconnect them.
    ServerMinInterval: 60s
    # ServerInterval is the time between pings to clients.
    ServerInterval: 7200s
    # ServerTimeout is the duration the server waits for a response from
    # a client before closing the connection.
    ServerTimeout: 20s
```

HYPERLEDGER FABRIC

Peer binary



HYPERLEDGER FABRIC

Peer binary

3- **peer** Tool:

peer [command] [subcommand] --flags
peer help
peer [command] help

```
$ peer help
Usage:
  peer [command]

Available Commands:
  chaincode   Operate a chaincode: install|instantiate|invoke|package|query|signpackage|upgrade|list.
  channel     Operate a channel: create|fetch|join|list|update|signconfigtx|getinfo.
  help        Help about any command
  lifecycle   Perform _lifecycle operations
  node        Operate a peer node: start|reset|rollback|pause|resume|rebuild-dbs|upgrade-dbs.
  version     Print fabric peer version.

Flags:
  -h, --help   help for peer

Use "peer [command] --help" for more information about a command.
```

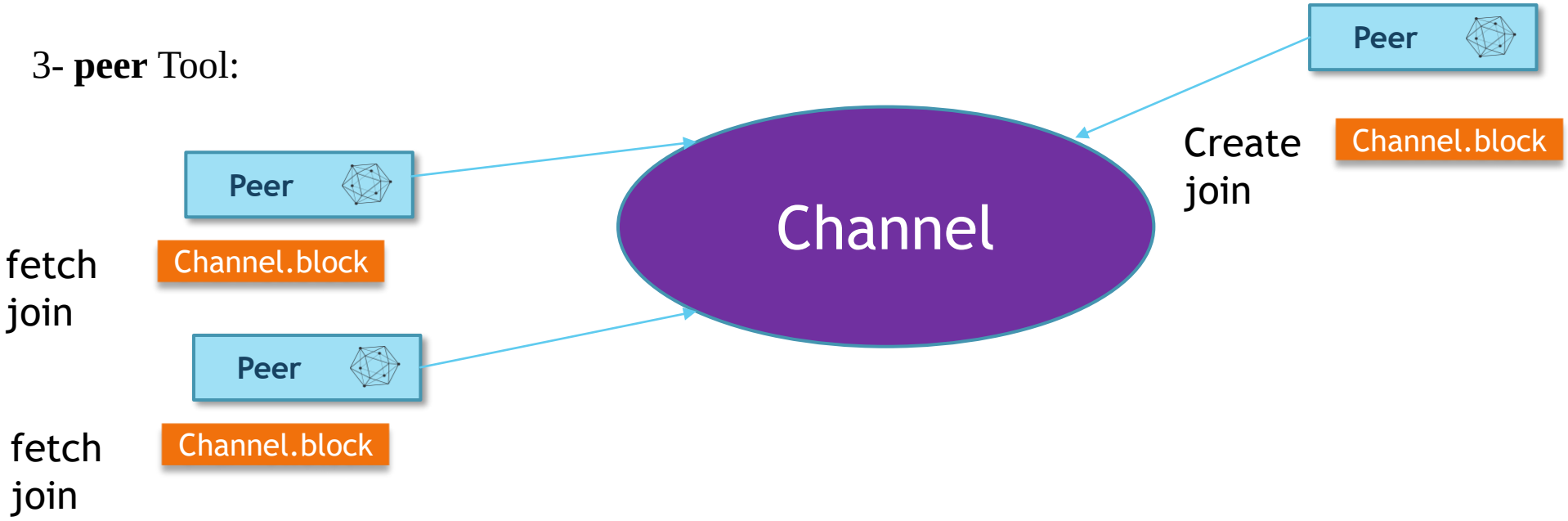
Example:

- peer channel list
- peer channel getinfo –c mychannel

HYPERLEDGER FABRIC

Peer binary

3- **peer** Tool:



peer channel fetch -o localhost:7050
-newest
-oldest
-block number if = 0 Genesis Block
-config gets the latest config block + block Number