
HYPERLEDGER
FABRIC

HYPERLEDGER
FABRIC

- 2015: December - Founded by The Linux Foundation

- 2016: IBM becomes member
  - IBM remains a lead contributor today
  - IBM CTO of Open Technology Christopher Ferris heads the HF Technical Steering Committee

- 2017:
  - July: Hyperledger Fabric 1.0 released
  - July – September:
    - London Stock Exchange Group in a partnership with IBM announced that it will create a blockchain platform designed for digitally issuing shares of Italian companies with Hyperledger Fabric as the basis of the platform
    - Oracle joined the Hyperledger consortium and announced its Blockchain Cloud Service offering
    - Royal Bank of Canada (RBC) started using Hyperledger for its US - Canada interbank settlements

HYPERLEDGER
FABRIC

Open Management:

- 35 organizations and nearly 200 developers

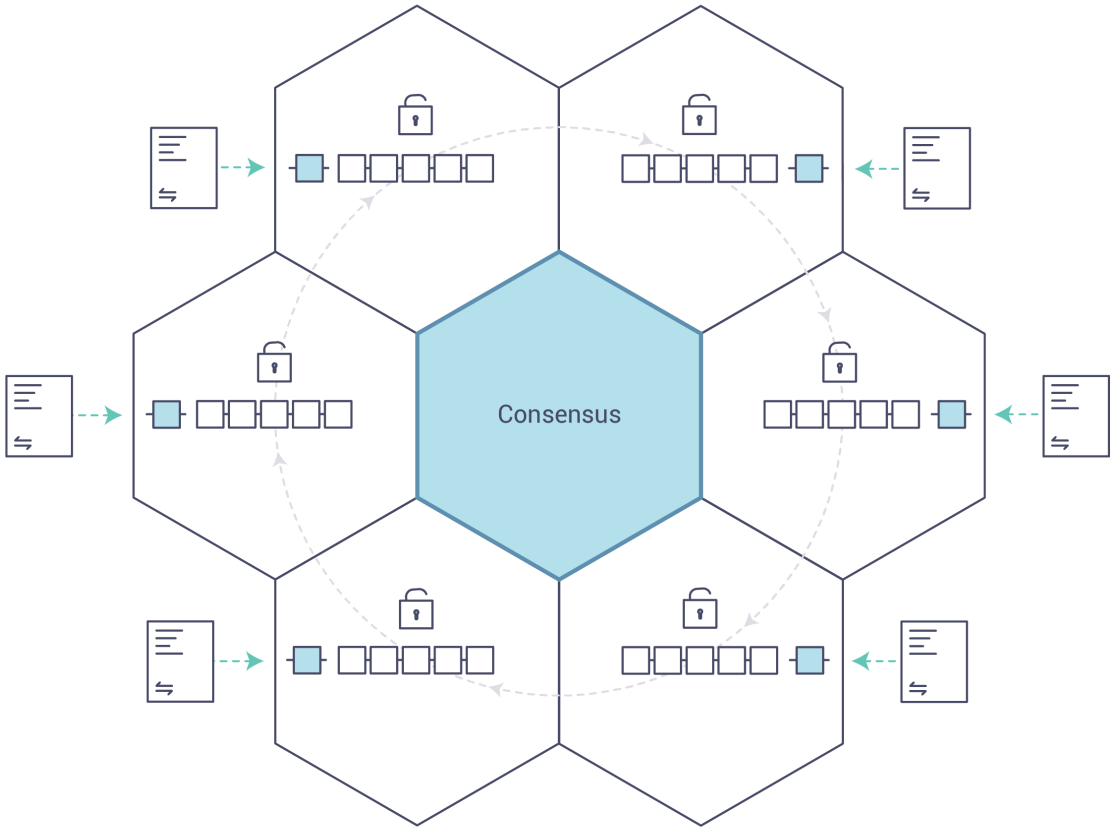
Enterprise-grade platform:

- Modular / Configurable to meet various industry needs
- Permissioned access
- Channel isolation (on the same network)
- Private Data storage options

https://hyperledger-fabric.readthedocs.io/en/release-1.2/whatis.html

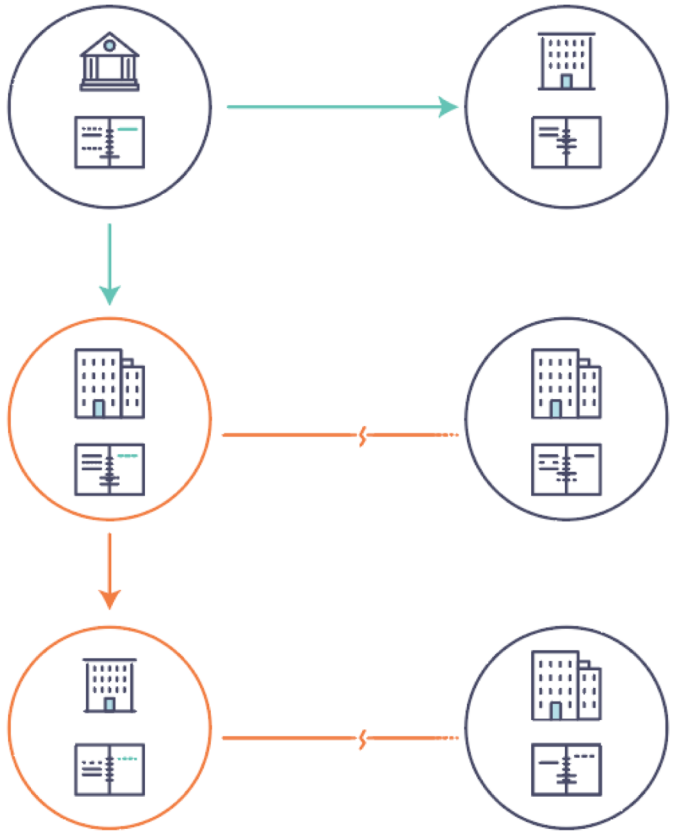
HYPERLEDGER
FABRIC

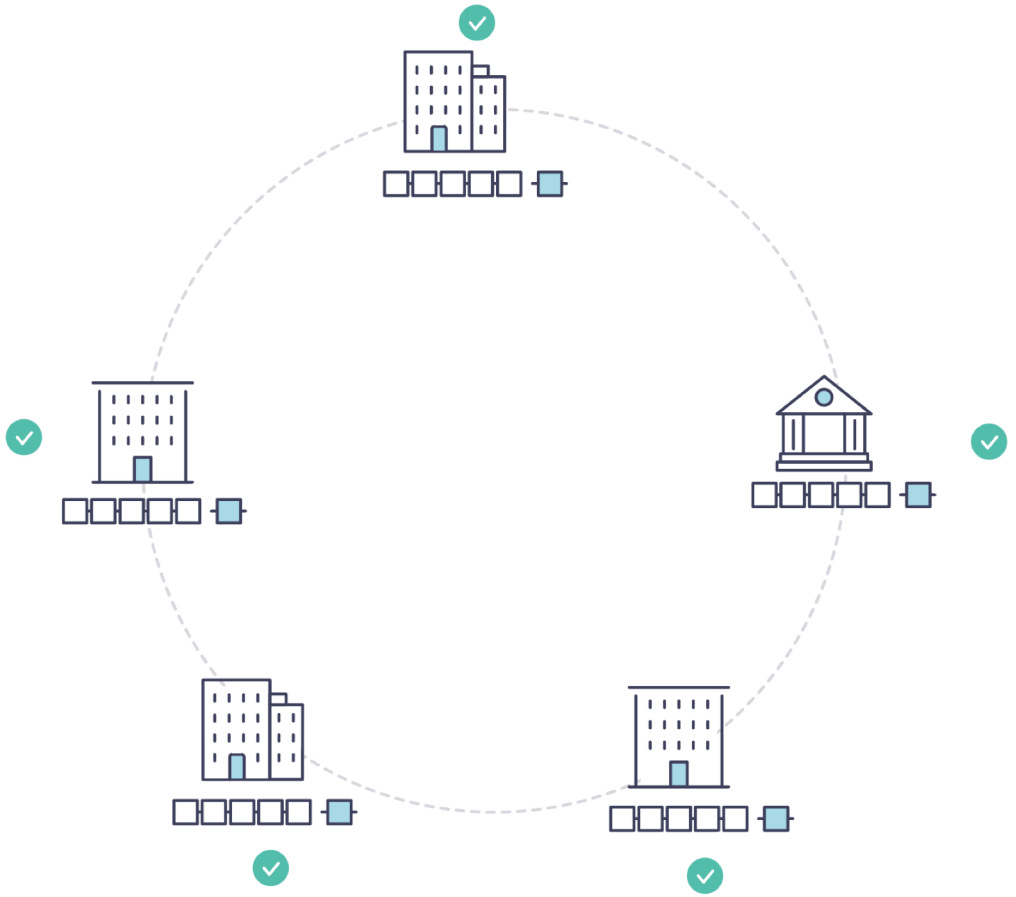
Blockchain / DLT (Distributed Ledger Technology)



HYPERLEDGER
FABRIC

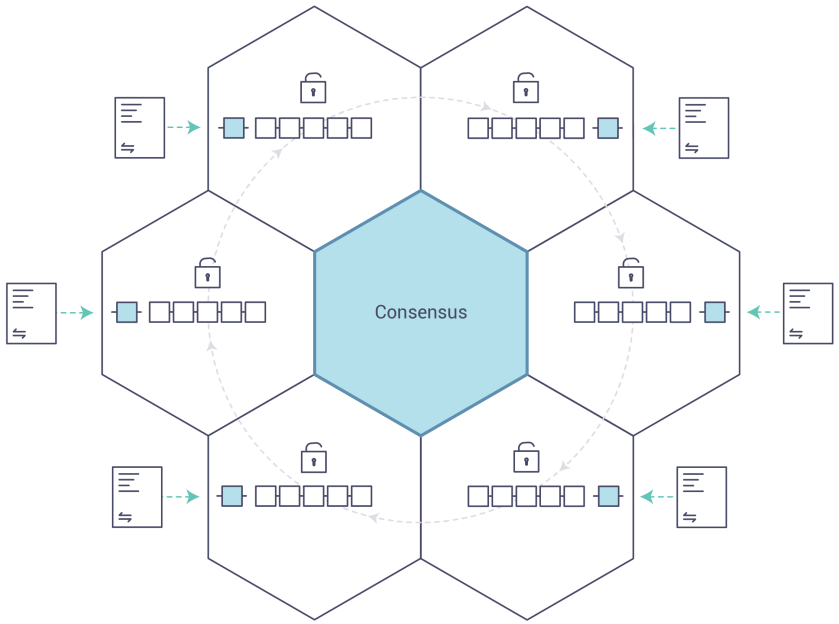
Blockchain / DLT (Distributed Ledger Technology)



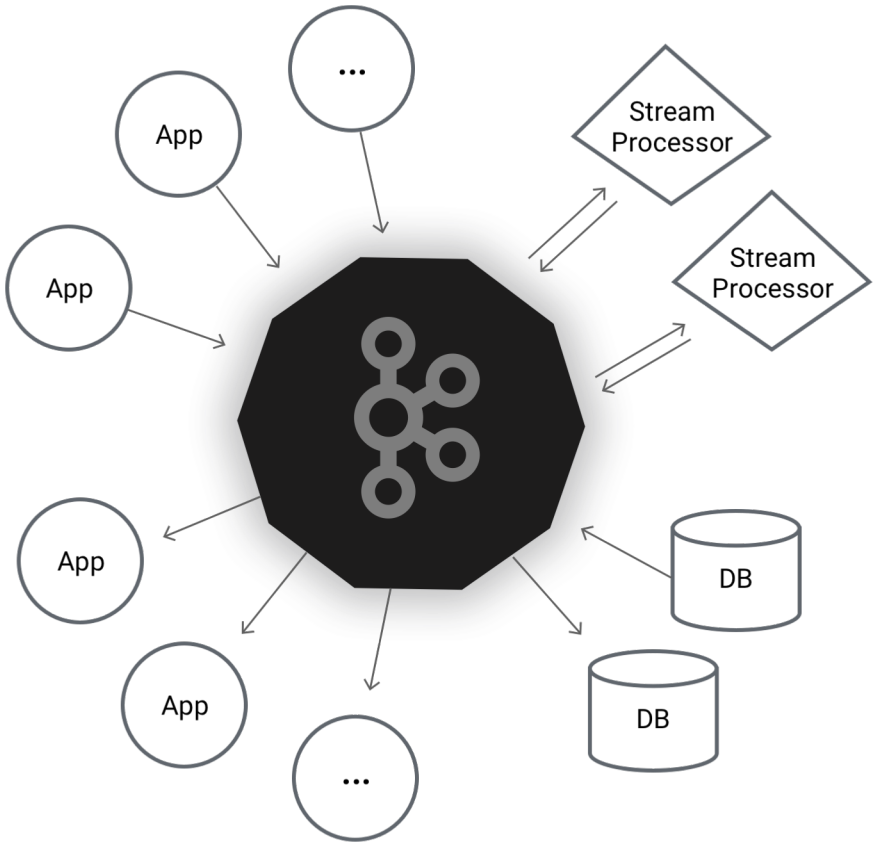


HYPERLEDGER
FABRIC

Consensus Model

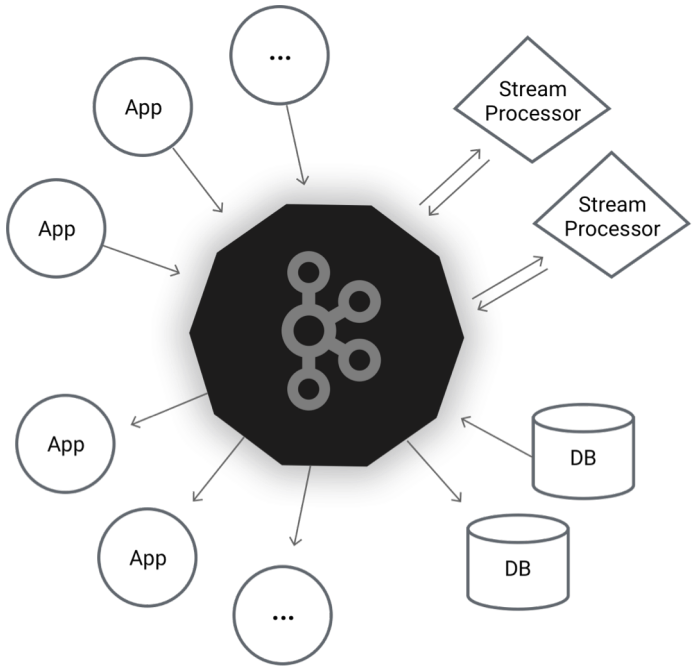


APACHE kafka®
A distributed streaming platform



HYPERLEDGER FABRIC

APACHE kafka®
A distributed streaming platform

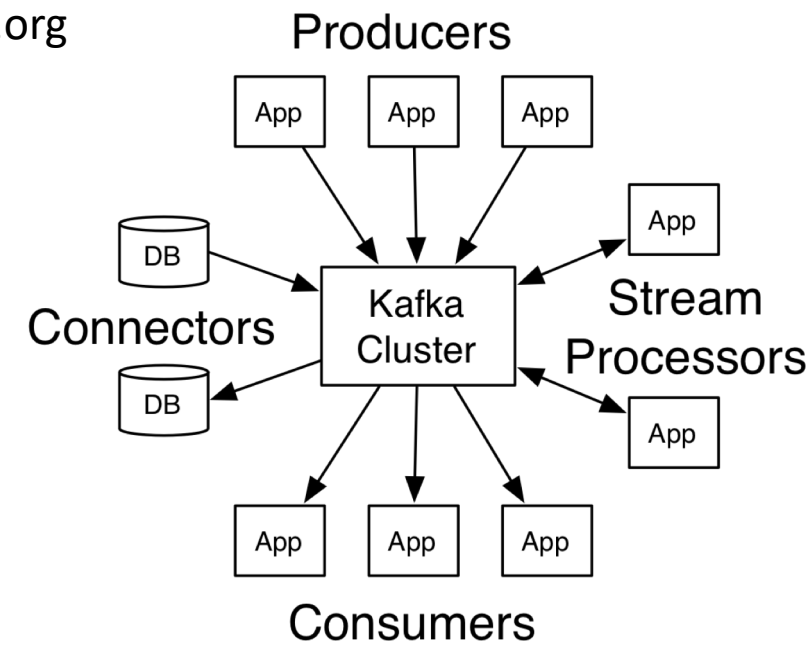


Apache Kafka® is *a distributed streaming platform*.

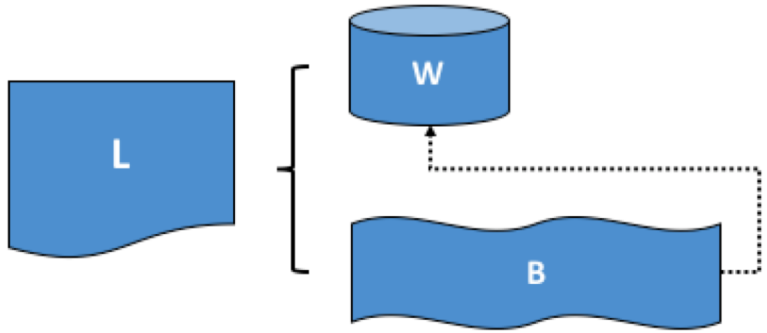
- **Publish** and **subscribe (pub/sub)** to streams of records, similar to a message queue or enterprise messaging system
- Process streams of records as they occur

- Kafka is run as a cluster on one or more servers
- The Kafka cluster stores streams of ***records*** in categories called ***topics***
- Each record consists of: **key**, **value**, **timestamp**

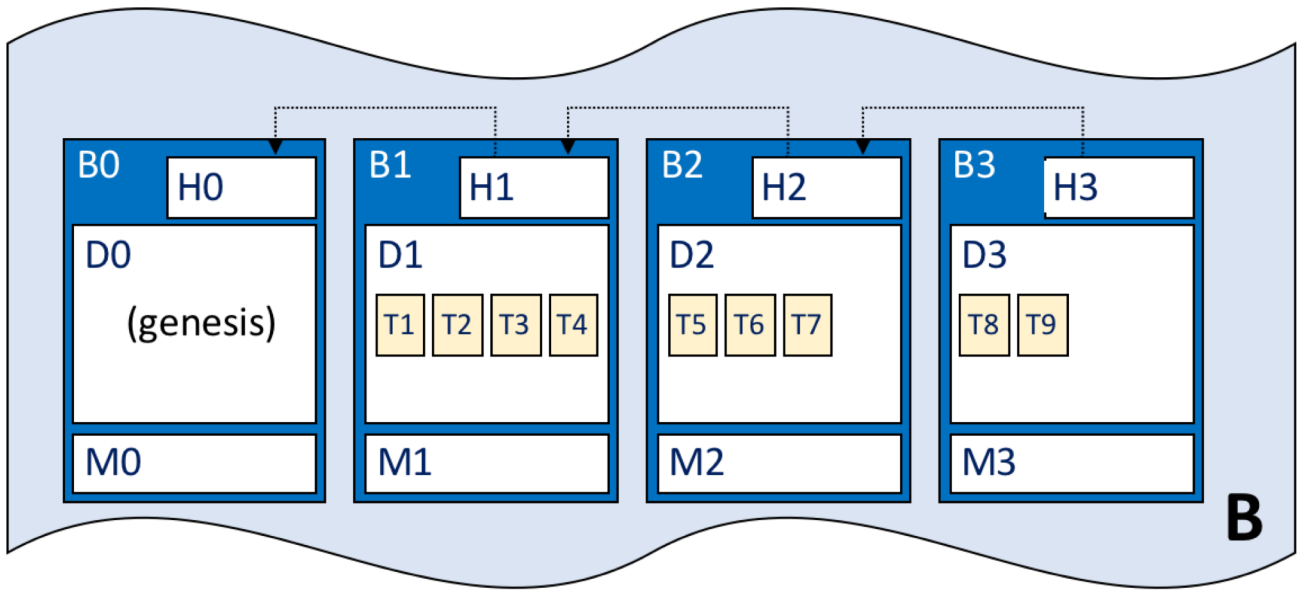
kafka.apache.org



HYPERLEDGER
FABRIC



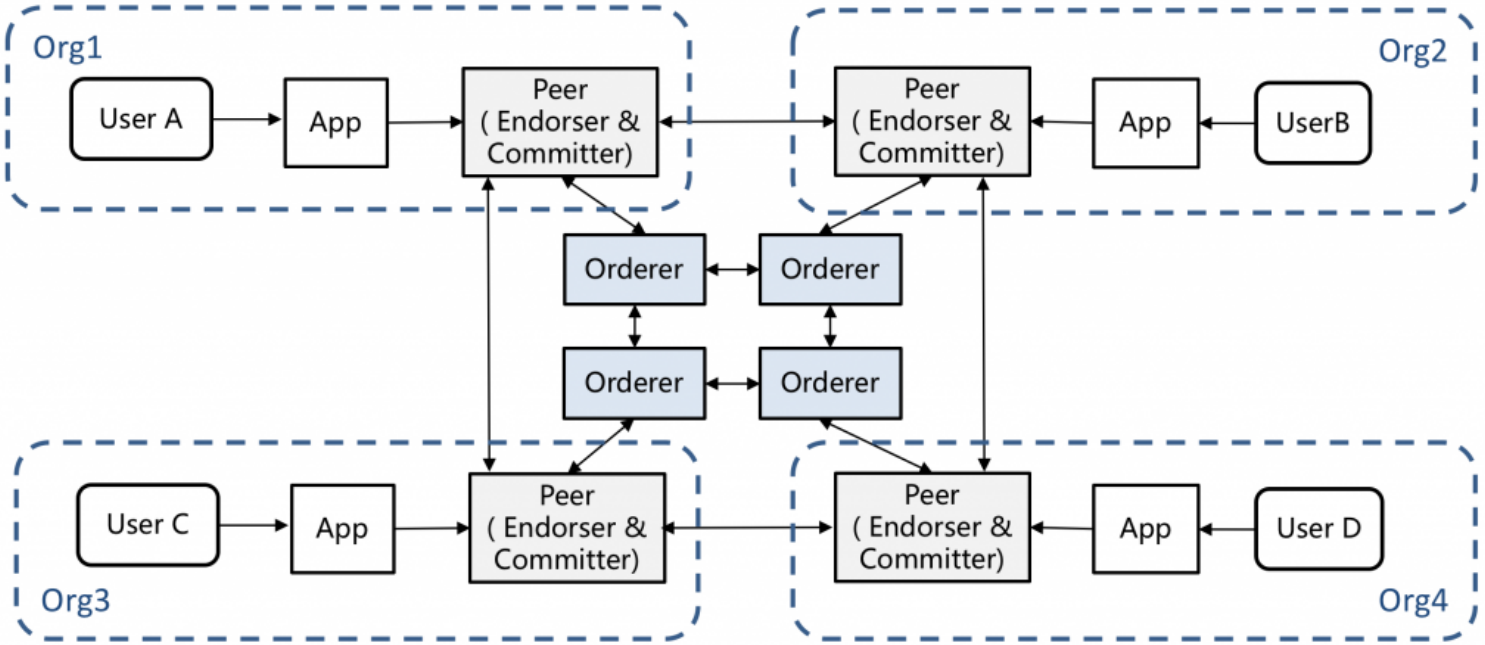
L	Ledger
W	World State
B	Blockchain
L: B, W	L comprises B and W
W ← B	B determines W



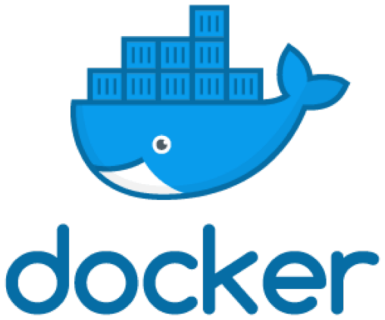
B	Blockchain
B1	Block
H3	Block header
D1	Block data
T5	Transaction
M3	Block metadata
H1 ← H2	H2 is chained to H1

HYPERLEDGER
FABRIC

hackernoon.com



Fabric 1.0 deployment model. Some arrows are omitted for simplicity



docker

```
seanhart$ docker ps -a
CONTAINER ID  IMAGE                          PORTS                                                NAMES
750ed9b60395  hyperledger/fabric-ca-tools     0.0.0.0:3000->3000/tcp                               org1-cli
e438700257c5  dev-org1-peer0-ccWallet-1.0-1d9a…                                                   dev-org1-peer0-ccWallet-1.0
292e12448e7a  hyperledger/fabric-ca-peer      0.0.0.0:7051->7051/tcp, 0.0.0.0:7053->7053/tcp       org1-peer0
dd155311bf03  hyperledger/fabric-ca-orderer  0.0.0.0:7050->7050/tcp                               org1-orderer
5b5f4a6d2c57  hyperledger/fabric-ca           0.0.0.0:7054->7054/tcp                               org1-ca
```

HYPERLEDGER
FABRIC

```go
// ChaincodeTemplate is a empty structure to represent the chaincode object
type ChaincodeTemplate struct{}

func main() {
    err := shim.Start(new(ChaincodeTemplate))
    if err != nil {
        fmt.Printf("Error starting chaincode: %s", err)
    }
}

// Init initializes the chaincode
func (t *ChaincodeTemplate) Init(stub shim.ChaincodeStubInterface) pb.Response {

}
```

HYPERLEDGER
FABRIC

```go
func (t *ChaincodeTemplate) Invoke(stub shim.ChaincodeStubInterface) pb.Response {

    // Get the passed function name (for the shim), and the arguments
    // The first returned value is the function name
    function, args := stub.GetFunctionAndParameters()
    if function == "invoke" {
        return t.invoke(stub, args)
    } else if function == "delete" {
        return t.delete(stub, args)
    } else if function == "query" {
        return t.query(stub, args)
    }
    return shim.Error("Invalid invoke function name. Expecting \"invoke\" \"delete\" \"query\"")
}



func (t *ChaincodeTemplate) invoke(stub shim.ChaincodeStubInterface, args []string) pb.Response {

}

func (t *ChaincodeTemplate) delete(stub shim.ChaincodeStubInterface, args []string) pb.Response {

}

func (t *ChaincodeTemplate) query(stub shim.ChaincodeStubInterface, args []string) pb.Response {

}
```

HYPERLEDGER
FABRIC