Attachment VII – Architecture Mapping of Hyperledger Fabric

Section 1 Summary

Platform summary	
Platform ID	*Hyperledger Fabric*
Status/Revision	*Proposed; V1.4* (*long term support release),*
Type	*Permissioned, Consortium*
Domain	*A wide range of industry use cases ranging from government, to finance, to supply-chain logistics, to healthcare and so much more.*
Description	*Hyperledger Fabric has been designed for enterprise use from the outset.*

Section 2 Governance & Compliance Functions

Platform governance	
Governance Type	*Permissioned;*
Chain Network Admin	*Entity (Consortium/Private)*
Pledge (cost of malicious action)	*The guilty party can be easily identified and the incident handled in accordance with the terms of the governance model. Stop service.*
Description	*Fabric operates a blockchain amongst a set of known, identified and often vetted participants operating under a governance model. All actions, whether submitting application transactions, modifying the configuration of the network or deploying a smart contract are recorded on the blockchain. So the guilty party can be easily identified and the incident handled in accordance with the terms of the governance model.*

Platform trust endorsement policy	
Type	*Law/Agreement;*
Tool	*Contract ID;*
Policy	*A fabric network can be operated under a governance model that is built off of what trust does exist between participants, such as a legal agreement or framework for handling disputes.*

	An endorsement policy which specifies the set of peers on a channel that must execute chaincode and endorse the execution results.

Economic Model (optional)	
Price Model to Deploy Contracts and do Transactions	*NA*
Who pays the costs of the network	*NA*
Monetary Policy of Tokens	*Fabric v2.0 will release the FabToken feature. FabToken is an Unspent Transaction Output (UTXO) based token management system that allows users to issue, transfer, and redeem tokens on channels.*
Rights of Tokens	*FabToken uses the membership services of Fabric to authenticate the identity of token owners and manage their public and private keys.*

Section 3 Application

Platform Smart Contract mechanism	
Language	*Go, Node.js, Java; ...*
Turing Complete?	*Yes*
Compiler	*Go; Java; …*
Runtime VM	*a secured Docker container; …*
DevTools	*Dev framework(the chaincode/(smart contract) interface, the chaincode "shim" APIs is the ChaincodeStubInterface; fabric-chaintool;…*
Extra Tool(s)	*Hyperledger Explorer (Block data view); Hyperledger Caliper (performance benchmarking); Hyperledger Burrow(Solidty smart contract migration); Hyperledger Composer(contract orchestration)*
Lifecycle	*The Hyperledger Fabric API supports package, install, instantiate and upgrade chaincode on the endorsing peer nodes.*
Description	*the chaincode interface whose methods are called in response to received transactions, the ChaincodeStubInterface is used to access and modify the ledger; fabric-chaintool is a utility to assist in various phases of chaincode development, such as compilation, test, packaging, and deployment;*

Section 4 Protocol

Platform AAA Management	
Account type	*Identity; address;*
Distributed ID	*using an established PKI structure, a digital identity encapsulated in an X.509 digital certificate*
AAA support	*Fabric CA;* *Membership Service Providers,*
Description	*Membership Services authenticates, authorizes, and manages identities on a permissioned blockchain network. A membership identity service that manages user IDs and authenticates all participants on the network. Access control lists can be used to provide additional layers of permission through authorization of specific network operations.* *Fabric provides Identity Mixer, which has a trust model and security guarantees similar to what is ensured by standard X.509 certificates but with underlying cryptographic algorithms that efficiently provide advanced privacy features such as "unlinkability" and minimal attribute disclosure.*

Platform Consensus Mechanism	
Algorithm	*CFT ;*
Consensus mode	*Event;*
Management solution	*external*
Description	*In the currently available releases, Fabric offers a CFT ordering service implemented with Kafka and Zookeeper. It also has a Raft consensus implementation. From version 1.4.2, users can migrate from Kafka to Raft. There're third parties providing other consensus for Fabric like LaSIGE implemented BFT-Smart consensus.*

Platform Ledger Management	
Model	*balance;*
Extra	*Fabric Coin (UTXO cryptocurrencies) described in a peer reviewed paper published by a team from IBM Research;*
Description	*N/A*

Section 5 Resources

Node Management	
Node Role	*There are three types of nodes in Fabric:* *1. Client or submitting-client: a client that submits an actual transaction-invocation to the endorsers, and broadcasts transaction-proposals to the ordering service.* *2. Peer: a node that commits transactions and maintains the state and a copy of the ledger. Besides, peers can have a special endorser role. The special function of an endorsing peer occurs with respect to a particular chaincode and consists in endorsing a transaction before it is committed.* *3. Ordering-service-node or orderer: a node running the communication service that implements a delivery guarantee, such as atomic or total order broadcast.*
Joining	*1.Channel administrators send out channel configuration tx to Orderer, to add an organization to the channel;* *2. Organization administrators sends a join channel proposal to one or more endorsing peers, to let a peer of the organization join the channel;*
Leaving	*Peers have no activity on the channel for a timely basis;*
Role changing	*Organization administrators send out channel configuration tx to Orderer, to update an anchor peer;*
Description	*Network MSP defines who are the members in the network; Orderer MSP list the actors or nodes it trusts;* *A peer need to be registered and enrolled to obtain the enrollment certificate signed by the Organization CA;* *Anchor peer is used by gossip to make sure peers in different organizations know about each other.*

Platform Data Storage Mechanism	
Mass storage mitigation[1]	*NA…*
Decentralized Data Storage Support	*KV style databases*
Data Privacy Solution	*Fabric enables confidentiality through its channel architecture, and has added support for private data encryption. A Hyperledger Fabric channel is a private "subnet" of communication between two or more specific*

[1] On chain storage cost much, solution/mechanism to resolve the problem of large cost of mass storage from node perspective. E.g., data maintenance, data storage and data cleaning.

	network members, for the purpose of conducting private and confidential transactions.
Tamper Proof (tamper cost)	*In the currently available 1.4.0 releases, Fabric only offers a CFT ordering service. And depends on Endorsement policies.*
Description	*zero knowledge proofs (ZKP) available in the future*

Platform Network Management	
Node Scalability	*Hundreds*
Network Structure	*Distributed; Flexible;*
Network Discovery Protocol	*Gossip;*
Byzantine Node Accepted?	*No for now*
P2P?	*Yes(except Orderers)*
Data Exchange Protocol	*Gossip;*
Description	*The gossip-based data dissemination protocol manages peer discovery and channel membership, disseminates ledger data across all peers on a channel, brings newly connected peers up to speed by allowing peer-to-peer state transfer update of ledger data;*

Section 6 Utils

Platform Messaging Mechanism	
Protocol Type	*gRPC;*
Description	*Further description if any*

Platform Crypto Libraries	
Secure Network Connection Type	*TLS;*
Cipher Suites	*Hyperledger Ursa ;*
Description	*The signing key used for signing by the node (currently only ECDSA keys are supported). In Hyperledger Fabric there is no support for certificates including RSA keys for now, a new cipher library Ursa has just been added to Hyperledger projects and may provide various algorithms later.*

Section 7 Operation & Maintenance

Platform system management – Node	
Log	*Yes*
Monitoring	*provides operators with three services to monitor, logging, Health Checks, metrics from peer and orderer nodes*
Description	*Some components of the Fabric peer and orderer expose metrics that can help provide insight into the behavior of the system. Operators and administrators can use this information to better understand how the system is performing over time.* *Hyperledger Explorer project can also be used to monitor on-chain data.*

Platform system management – Chain Network	
Permission Control	*Local MSPs; Channel MSP; Network MSP; Peer MSP; Orderer MSP;*
Auditing	*N/A*
Supervisory Support	*N/A*
Description	*An MSP goes beyond simply listing who is a network participant or member of a channel. An MSP can identify specific roles an actor might play either within the scope of the organization the MSP represents (e.g., admins, or as members of a sub-organization group), and sets the basis for defining access privileges in the context of a network and channel (e.g., channel admins, readers, writers)*

Section 8 External Resource Management

Platform External Resource Management	
Interoperation solution	*NA;*
Description	*N/A*

Section 9 Extensions

Platform Extensions - optional
[the following list can be duplicated for multiple extensions]

Name	*Smart Contract Support:*
Extension type[2]	*Internal;*
Extension mode[3]	*capability (vertical)*
Solution	*Hyperledger Burrow enables one to use the Hyperledger Fabric permissioned blockchain platform to interact with Ethereum smart contracts written in an EVM compatible language such as Solidity or Vyper.*
Serve domain	*Smart Contract Support*
Description	*the Fabric EVM Chaincode(Enables Ethereum smart contracts written in an EVM compatible language such as Solidity or Vyper)*

[2] Standing from DLT system instance perspective, any extension inside the instance is marked as "internal", while any extension outside the instance is marked as "external"

[3] All extension instances are equal (with similar capability and functional features), targeting for the scalability of DLT instance, marked as "horizontal"; extensions with different functional features, targeting to enforce the capability of DLT instance, marked as vertical. Extension type and mode pair(s) is/are used to describe the extension as to the whole DLT system. E.g., sharding (internal – horizontal), lightening – BTC (external – vertical), Corda Contract (internal – vertical).