Attachment XIV – Architecture Mapping of Hyperledger Sawtooth

Section 1 Summary

Platform summary	
Platform ID	*Hyperledger Sawtooth*
Status/Revision	*V1.1.4, …*
Type	*Permissioned, Consortium*
Domain	*Financial, to supply chain, to assets marketplace and so much more.…*
Description	*Hyperledger Sawtooth is an enterprise blockchain platform for building distributed ledger applications and networks.*

Section 2 Governance & Compliance Functions

Platform governance	
Governance Type	*Permissioned (private); …*
Chain Network Admin	*Entity (Consortium/Private)…*
Pledge (cost of malicious action)	*Blacklisting; stop service*
Description	*The blockchain stores the settings that specify the permissions, such as roles and identities, so that all participants in the network can access these informations.*

Platform trust endorsement policy	
Type	*Trusted execution environment;*
Tool	*Contract ID*
Policy	*Offers a solution to the Byzantine Generals Problem that utilizes a TEE (trusted execution environment).Cheating is prevented through the use of a TEE, identity verification and blacklisting based on asymmetric key cryptography. Using TEE provided by intel chips implies endorsement from Intel corporation.*

Economic Model (optional)

Price Model to Deploy Contracts and do Transactions	*NA*
Who pays the costs of the network	*NA*
Monetary Policy of Tokens	*NA*
Rights of Tokens	*NA*

Section 3 Application

Platform Smart Contract mechanism	
Language	*Go; JavaScript; Python;Rust; ...*
Turing Complete?	*Yes*
Compiler	*Go; Rust; …*
Runtime VM	*Process or Docker; …*
DevTools	*Smart Contract Templates (transaction family), …*
Extra Tool(s)	*Hyperledger Caliper (performance benchmarking); Hyperledger Burrow(smart contract migration)*
Lifecycle	*Manually managed*
Description	*A transaction family includes a transaction processor to define the business logic for your application, a data model to record and store data and a client to handle the client logic for your application.*

Section 4 Protocol

Platform AAA Management	
Account type	*Identity; address; …*
Distributed ID	*The private key's associated public key be formatted as a hexadecimal string to prove your identity on the blockchain.*
AAA support	*Transactor key permissioning,* *Validator key permissioning ,*

Description	*Transactor key permissioning controls who can submit transactions and batches, based on signing keys.* *Validator key permissioning controls which nodes are allowed to establish connections to the validator network.* *The data of state is accessed using an addressing scheme that an address begins with a namespace prefix and the hex-encoded hash values of the string or strings that make up the address elements.*

Platform Consensus Mechanism	
Algorithm	*PoET; …*
Consensus mode	*Event;*
Management solution	*Internal; external*
Description	*PoET(Proof of Elapsed Time), the peer with the smallest sample wins the election, relies on secure instruction execution, a Nakamoto-style consensus algorithm that is designed to be a production-grade protocol capable of supporting large network populations.*

Platform Ledger Management	
Model	*balance;*
Extra	*Sawtooth Private UTXO, allows for assets to be tracked and traded on the Ledger;*
Description	*Sawtooth represents state for all transaction families in a single instance of a Merkle-Radix tree on each validator*

Section 5 Resources

Node Management	
Node Role	*Validator (node in Sawtooth) is the component ultimately responsible for validating batches of transactions, combining them into blocks, maintaining consensus with the network, and coordinating communication between clients, other validators, and transaction processors.*
Joining	*Validator Registry transactions are sent to add new validators to the network.*

Leaving	*Validator Registry transactions are sent to let validators leave the network.*
Role changing	*NA*
Description	*Validator key permissioning controls which nodes are allowed to establish connections to the validator network.*

Platform Data Storage Mechanism	
Mass storage mitigation[1]	*NA…*
Decentralized Data Storage Support	*Custom formatted file;*
Data Privacy Solution	*NA*
Tamper Proof (tamper cost)	*51% of the nodes decide to tamper (PoeT is a Nakamoto-style consensus algorithm).*
Description	*Further description if any*

Platform Network Management	
Node Scalability	*Node scale*
Network Structure	*Distributed; Flexible; …*
Network Discovery Protocol	*Kademlia-like; Private …*
Byzantine Node Accepted?	*Yes*
P2P?	*Yes*
Data Exchange Protocol	*Gossip; ...*
Description	*Further description if any*

Section 6 Utils

Platform Messaging Mechanism

[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.

Protocol Type	*ZeroMQ Message Transfer Protocol;*
Description	*ZeroMQ includes a TLS-like certificate exchange mechanism and protocol encryption capability*

Platform Crypto Libraries	
Secure Network Connection Type	*ZeroMQ TLS-like;*
Cipher Suites	*ECDSA;*
Description	*Sawtooth uses OpenSSL Toolkit. ECDSA key using the secp256k1 curve. Also it will share more algorithms provided by Hyperledger Ursa in the future.*

Section 7 Operation & Maintenance

Platform system management – Node	
Log	*Yes*
Monitoring	*Display Sawtooth metrics with Grafana, using InfluxDB to store the metrics data.*
Description	*Sawtooth network allows nodes with different versions co-exist and can set the Allowed Transaction Types;*

Platform system management – Chain Network	
Permission Control	*local validator configuration and network-wide on-chain permissioning*
Auditing	*NA*
Supervisory Support	*NA*
Description	*Sawtooth network allows nodes with different versions co-exist and can set the Allowed Transaction Types;*

Section 8 External Resource Management

Platform External Resource Management

Interoperation solution	*NA*
Description	*NA*

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	*Ethereum Contract Compatibility with Seth*
Serve domain	*vertical: Smart Contract*
Description	*Seth, extends the interoperability of the Sawtooth platform to Ethereum. EVM (Ethereum Virtual Machine) smart contracts can be deployed to Sawtooth using the Seth transaction family.*

[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).