Attachment XIII – Architecture Mapping of Quorum

Section 1 Summary

Platform summary	
Platform ID	*QUORUM: Go Ethereum (Geth)*
Status/Revision	*Soft-fork of Ethereum, Last Stable version v2.2.4*
Type	*Public-permissioned*
Domain	*Many sectors, e.g., Supply chain; Finance; Retail, etc…*
Description	*QUORUM is an Ethereum-based distributed ledger protocol with transaction /contract Privacy and new consensus mechanisms. That can bring the best from both worlds, every node on the network can validate every transaction on list but only exposing to relevant parties.* https://github.com/jpmorganchase/quorum

Section 2 Governance & Compliance Functions

Platform governance	
Governance Type	*It is modular with a base of BFT by validators nodes and regular nodes.*
Chain Network Admin	*Brainchild of JP Morgan*
Pledge (cost of malicious action)	*Free adoption by permission (OPEN SOURCE)*
Tamper Proof (tamper cost)	No gas by private forks.
Description	***Quorum is an Ethereum-based is an open-source platform for decentralized applications to support enterprise requirements such as privacy.*** https://github.com/jpmorganchase/quorum/blob/master/docs/Quorum%20Whitepaper%20v0.2.pdf

Platform trust endorsement policy	
Type	*Permissioned*
Tool	*Signature Validation*
Policy	*Open Source under LGPL 3.0 License*

Economic Model (optional)	
Price Model to Deploy Contracts and do Transactions	*Private control through automation. For Public state or Private State.*
Who pays the costs of the network	*Stakeholders and memberships.*
Monetary Policy of Tokens	*Non-zero gas model to manage the use of the infrastructure in a responsible way*
Rights of Tokens	*To be defined*

Section 3 Application

Platform Smart Contract mechanism	
Language	*Solidity*
Turing Complete?	*Yes – Solidity*
Compiler	*java, Solidity;*
Runtime VM	*EVM – Ethereum Virtual Machine; ABI, OVM, WAR;*
DevTools	*Quorum Blockchain explorer, Quorum Genesis, Quorum Maker, QuorumNetworkManager, ERC20 REST servie, Nethereum Qourum, web3j-quorum, Apache Camel, Quorum API. Cakeshop, quorum cloud.*
Extra Tool(s)	*Tessera is implemented in Java and it is never than Costellation (implemented in Haskell)*
Lifecycle	*Privacy Manager (Constellation/Tessera) binomial: transaction manager + Enclave.*
Description	*Cakeshop as a set of tools and APIs for working with Ethereum-like ledgers. Supports private transactions and private contracts through public/private state separation. Although private contracts work better than public ones as there is less overhead when it comes to handling private contracts. This means that Quorum private blockchain is effective.*

Section 4 Protocol

Platform AAA Management	
Account type	*Identity validation by signature.*
Distributed ID	*DID, ERC721, ERC725 and others non-fungible identities.*
AAA support	*EIP1812*
Description	*Although it is anonymous state the network has to be identified in order to peer-to-peer maintenance.*

	DIDs are welcome in Quorum and compatible with other solutions for four kinds of digital identiy: People, legal entities (NGOs, Public, Private sector, etc…) things and processes.

Platform Consensus Mechanism	
Algorithm	*PoW; PoS, BFT; HBBFT, PoA*
Consensus mode	*Pluggable RAFT, IBFT and Clique PoA*
Management solution	*Internal; external*
Description	https://github.com/jpmorganchase/quorum/blob/master/docs/Quorum%20Whitepaper%20v0.2.pdf

Platform Ledger Management	
Model	*Private control through automation. For Public state or Private state.*
Extra	*MPT support - modified Merkle Patricia tree (trie)*
Description	Smart Contracts play a crucial role whereby can be customized by business themselves.

Section 5 Resources

Node Management	
Node Role	*Validator node: Validates transactions proposals and create new block on ledger, and they keep a copy of the ledger itself (Validators Nodes only exists on IBFT mode).* *Regular Node: Exists on Raft or in IBFT (this last case considered as Non-validators nodes), responsible to store a copy of ledger and make new proposals to Validators Nodes as responsible to spread updated ledger to non-validator nodes over the network itself.* *Boot node: Permission new nodes.*
Joining	*Create a node key (enode) by using the Bootnode tool, then make a copy of static-nodes.json file into node folder, then copy the enode into the permissioned-node.json file (where all enodes of network are listed), initialize the node trough the "geth" tool , last is through an already active running node, use "addpeer()" command on "geth" tool so the node can make part of the network.*
Leaving	*Through an existing node. You could run a command called removePeer() which then will remove a node trough hes enode number..*
Role changing	*N/A*
Description	https://github.com/jpmorganchase/quorum-examples

Platform Data Storage Mechanism	
Mass storage mitigation[1]	*Off-chain.*
Decentralized Data Storage Support	*Blockchain explorer for Quorum. Swarm is also capable with Quorum and IPFS.IPFS, cloud-services*
Data Privacy Solution	*ZKP; MPC; IPFS; ZSL, ZSC and Anonymous Zether,*
Description	*Privacy by design.*

Platform Network Management	
Node Scalability	*Thousands*
Network Structure	*Distributed*
Network Discovery Protocol	*Kademlia-like;*
Byzantine Node Accepted?	*Yes*
P2P?	*Yes*
Data Exchange Protocol	*RLPx*
Description	*RLPx transport protocol, a TCP-based transport protocol used for communication among Ethereum nodes. The protocol carries encrypted messages belonging to one or more 'capabilities' which are negotiated during connection establishment.* https://github.com/jpmorganchase/quorum/tree/master/rlp https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf

Section 6 Utils

Platform Messaging Mechanism	
Protocol Type	Transaction Manager, Peers, and Enclave use traditional TCP/UDP transport layer to communicate.
Description	*JSON-Remote Procedure Call (RPC) is a stateless, lightweight remote procedure call (RPC) protocol. Primarily this specification defines several data structures and the rules around their processing. It is transport agnostic in that the concepts can be used within the same*

[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.

	process, over sockets, over HTTP, or in many various message passing environments. https://github.com/jpmorganchase/quorum/blob/master/docs/Security/Framework/Quorum%20Network%20Security/Node.md

Platform Crypto Libraries	
Secure Network Connection Type	*Communication via public Internet (TCP + UDP).*
Cipher Suites	*ECDSA (Elliptic Curve Digital Signature Algorithm) for it's public-key cryptography and KECCAK-256 for hashing*
Description	https://github.com/jpmorganchase/quorum/tree/master/ethclient

Section 7 Operation & Maintenance

Platform system management – Node	
Log	*Modular and privacy by design*
Monitoring	Quorum Blockchain Explorer and others.
Description	*Network status allows anyone to see the performance and number of nodes and where they are located.* https://github.com/jpmorganchase/quorum/blob/master/docs/Privacy/Tessera/Usage/Monitoring.md

Platform system management – Chain Network	
Permission Control	*Peer Permissioning, only known parties can join the network.*
Auditing	*Public or Private.*
Supervisory Support	*N/A*
Description	

Section 8 External Resource Management

Platform External Resource Management	
Interoperation solution	*Sharding: Raiden , state channel; IPFS; Swarm. IoT Gateways and Non-DLT system interoperation solution like AWS and Oraclize*
Description	*The schema is designed by the peer-to-peer approaching on the Smart Contracts and can contain different dependencies for their transactions which some are off-chain by obliterability.*

Section 9 Extensions

Platform Extensions – optional	
[the following list can be duplicated for multiple extensions]	
Name	*Contributor License Agreement (CLA) at info@goquorum.com*
Extension type[2]	
Extension mode[3]	
Solution	
Serve domain	
Description	*Quorum is built on open source.*

Platform Extensions – optional	
[the following list can be duplicated for multiple extensions]	
Name	*Anonymous Zether*
Extension type	*Internal*
Extension mode	*capability (vertical)*
Solution	*Zether is an anonymous private payment system extension based on zero-knowledge proof protocol.*
Serve domain	*Smart Contract Support*
Description	*After Zether is deployed at a network it allows users to transfer their EC20 balances to other Zether accounts in a private (amounts) and anonymous (identity) way. At this moment Zether is only enabled to Raft consensus mode.*

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).