Attachment IV – Architecture Mapping of Corda

Section 1 Summary

Platform summary	
Platform ID	*Corda …*
Status/Revision	V4.0
Type	*Private, Consortium…*
Domain	*Mainly Financial, do to R3 consortium focus but can be adapted to many other segments and needs as Cordapps and network structurers can be easily adapted.*
Description	*Corda is an Open Source DLT that allow business to transact in a strict privacy P2P way by using Cordapps (smart contracts), reducing costs for the network owners.*

Section 2 Governance & Compliance Functions

Platform governance	
Governance Type	*Permissioned;*
Chain Network Admin	*Entity (Consortium/Private)*
Pledge (cost of malicious action)	***Business agreement, third parties liabilities (open source version)***
Description	*Governing body who led the Consortium. Need 3-rd party to arbitrate the dispute based upon the agreement.*

Platform trust endorsement policy	
Type	*Law/Agreement;*
Tool	*N/A*
Policy	

Economic Model (optional)	
Price Model to Deploy Contracts and do Transactions	*N/A*
Who pays the costs of the network	*N/A*
Monetary Policy of Tokens	*N/A*
Rights of Tokens	*N/A*

Section 3 Application

Platform Smart Contract mechanism	
Language	*Java, Kotlin*
Turing Complete?	*Yes*
Compiler	*Java, Kotlin*
Runtime VM	*JVM*
DevTools	*IntelliJ IDEA, Eclipse IDE*
Extra Tool(s)	*Node Explorer, Load Testing, Corda Network Builder*
Lifecycle	*Until now Cordapps must be installed manually inside Nodes specific folder and then restarted.*
Description	*Corda, uses a "Contract" code to validate State transactions.* *The contract code is a "pure" function executed in a deterministic environment, on a need-to-know basis which verifies transactions.*

Section 4 Protocol

Platform AAA Management	
Account type	*Address*

Distributed ID	*Commonly used Public/Private RSA 3072 bit Keypair with X.509 v3 Standard Certificates on a TLS v1.2 Standard Protocol*
AAA support	*Fabric CA;* *Membership Service Providers,*
Description	*Corda's network permissioning is composed by an certificate hierarchy as follows: Root Network CA, The doorman CA, Node CA, legal identity CA.* *Root Network CA: Used to issue the Doorman and control the Network.* *The Doorman CA (intermediary): Used to sign Node Keys on a day-to-day to not compromise Root's CA Private Key.* *Node CA: Each node issues its own certificate that is used to sing its identity keys and TLS certificates*

Platform Consensus Mechanism	
Algorithm	*Contract Code*
Consensus mode	*Event*
Management solution	*External*
Description	*Corda offers 2 types of Consensus:* *a) Where each required signer node, must validate the proposal before they sign the transaction.* *b) The transaction is only checked and validated by a 3rd party node "Notary Service".*

Platform Ledger Management	
Model	*UTXO*
Extra	*State*
Description	*Corda uses UTXO (Unspent transaction output) model where every state on the ledger is immutable.*

Section 5 Resources

Node Management	
Node Role	*The roles of the Corda nodes are exposed to the entire network through the Network Map and also Corda's certificates have a custom X.509 v3 extension that specifies the role the certificate relates to. This is how roles are defined inside the Network, as Doorman, Network Map, Node CA, etc…The extension contains a single ASN.1 integer identifying the identity type the certificate is for:* 1. *Doorman* 2. *Network map* 3. *Service identity (currently only used as the shared identity in distributed notaries)* 4. *Node certificate authority (from which the TLS and well-known identity certificates are issued)* 5. *Transport layer security* 6. *Well-known legal identity* 7. *Confidential legal identity*
Joining	*To Join to a Corda network a Regular Node must make the request to a "Doorman" server (Intermediary) so it can validate and authenticate the request. In addition to the Network Map, all the nodes must also use the same set of network parameters. These are a set of constants which guarantee interoperability between the nodes. The HTTP network map distributes the network parameters which are downloaded automatically by the nodes. Every new node must be listed inside the network map with their roles and profiles.*
Leaving	*If a Corda Node gets offline for any reason, he will still be listed inside the network map as a member of that network, so every transaction that is sent to him, it will be "on hold" until his return. It is up to Network Admin, to clear Network Map cache (updating the list), and kicking the specific "dark node".*
Role changing	*To change a Node role in corda few steps must be made.* *1) Change Node configuration file attending his new Role inside network structure.* *2) Issue new Certificate for the node accordingly to his new role inside network.* *3) Update Network Map accordingly to his current new functions and values.*
Description:	

Platform Data Storage Mechanism	
Mass storage mitigation[1]	*N/A*
Decentralized Data Storage Support	*N/A*
Data Privacy Solution	*Enables confidentiality trough Node P2P transaction (need-to-know basis).*
Tamper Proof (tamper cost)	
Description	

Platform Network Management	
Node Scalability	*Hundreds*
Network Structure	*Flexible*
Network Discovery Protocol	*HTTP Network Map*
Byzantine Node Accepted?	*Not Natively*
P2P?	*Yes*
Data Exchange Protocol	*AMQP/1.0 TLS*
Description	*A Notary demo, based on BFT-Smart Protocol was released.*

Section 6 Utils

Platform Messaging Mechanism	
Protocol Type	*RPC external and AMQP/1.0 TLS for internal Network Messaging*
Description:	*Nodes owners uses RPC Client to communicate with the Node.*

[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.

Platform Crypto Libraries	
Secure Network Connection Type	*TLS*
Cipher Suites	*ECDSA Nist P-256 curve (Secp256r1) or RSA with 3072bit keys*
Description:	

Section 7 Operation & Maintenance

Platform system management – Node	
Log	*yes*
Monitoring	*Node Explorer*
Description	*Corda Network Builder, Load Testing tool*

Platform system management – Chain Network	
Permission Control	*The Root CA*
Auditing	*In Schedule*
Supervisory Support	*N/A*
Description	

Section 8 External Resource Management

Platform External Resource Management	
L2 solution:	*N/A*
Non-DLT system interoperation solution:	*Support for Oracle and SQL Server Database*
Description:	

Section 9 Extensions

Platform Extensions - optional	
[the following list can be duplicated for multiple extensions]	
Name	*Business Network*
Extension type[2]	*Internal*
Extension mode[3]	*Capability (vertical) and* Scalability (*horizontal)*
Solution	*Corda Multiple Cordapps/Contract*
Serve domain	*Scalability: Cordapps/Contracts*
Description	*Corda can have Multiple Cordapps/Contracts inside same node, providing as many individual P2P Business Networks Extensions needed. This way each network can enforce its own access control policies and process but at same time, they can have their own determination about which business networks they choose to participate.*

Platform Extensions - optional	
[the following list can be duplicated for multiple extensions]	
Name	*Corda Settler*
Extension type	*Internal*
Extension mode	*Capability (vertical)*
Solution	*Corda Settler is a DLT Cordapp that allows settlements payments transactions between crypto and traditional assets.*
Serve domain	*Scalability: Cordapps/Contracts*
Description	*Corda Settler is already working with Ripple XRP and also implemented SWIFT gpi link integration, that allows DLT users to settle payments obligation to DLT's, blockchains and traditional non-DLT rails.*

[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).