Attachment X – Architecture Mapping of Masterchain

Section 1 Summary

Platform summary	
Platform ID	*Masterchain*
Status/Revision	*Masterchain 0.3.0*
Type	*Private, Consortium*
Domain	*Financial*
Description	*Masterchain is a blockchain platform developed for banks and other financial institutions by forming a consortium. It allows for the data and information to be exchanged between the parties. It is developed by FinTech Association of Russian financial institutions.* *It is built using a fork of Ethereum blockchain.*

Section 2 Governance & Compliance Functions

Platform governance	
Governance Type	*Permissioned*
Chain Network Admin	*FinTech Association, a consortium that includes Bank of Russia and other financial institutions in Russia (e.g. Alfa Bank, Bank Otkritie, Tinkoff Bank, and Qiwi)*
Pledge (cost of malicious action)	*Computer Power – measured by hash rates*
Tamper Proof (tamper cost)	>50%
Description	

Platform trust endorsement policy	
Type	*Law, Consensus agreement by the consortium*
Tool	*Operation fully depends on the FinTech Association's central server that controls mining and consensus* *https://www.coindesk.com/russias-largest-bank-is-quitting-central-banks-blockchain-project*

Policy	*N/A*

Economic Model (optional)	
Price Model to Deploy Contracts and do Transactions	*Smart contracts are deployed in the ecosystem. They are charged per transaction.* *Transaction fees are paid in gas.*
Who pays the costs of the network	*Network participants/Application providers*
Monetary Policy of Tokens	*Unlimited supply, all held by FinTech Association, provided to consortium members upon request* *Tokens for paying gas are distributed by the association's node. They are distributed among participants for free, and wallets are refilled from time to time automatically.* *Network participants do not have to rely on AFT for tokens, they can share it with each other.* *https://www.coindesk.com/russias-largest-bank-is-quitting-central-banks-blockchain-project*
Rights of Tokens	*N/A*

Section 3 Application

Platform Smart Contract mechanism	
Language	*Solidity*
Turing Complete?	*Yes*
Compiler	*Solcjs - Solidity*
Runtime VM	*EVM;*
DevTools	*Development: Visual Studio Code; Sublime; Remix;* *Build framework: Truffle, Embark, Remix* *Test framework: Truffle, Embark, Remix*
Extra Tool(s)	*Explorer (Block data view): Masterchain Explorer*
Lifecycle	*The developer has to code if the contract can stop or be killed. It is not possible to update the deployed smart contract, but there are recommendations to that.*

Description	

Section 4 Protocol

Platform AAA Management	
Account type	*Address;*
Distributed ID	*There are two types of accounts which share the same address space: externally owned accounts and contract accounts. Externally owned accounts are controlled by public-private key pairs and have no code. Contract accounts are controlled by the code stored together with the account – the smart contract code.* *User should generate an externally owned account using a local software/hardware in order to keep the private key private;* *Contract accounts are created during deploy.*
AAA support	*N/A*
Description	*The rational is that there are so many possible addresses that the probability of collision is negligible.*

Platform Consensus Mechanism	
Algorithm	*PoW;*
Consensus mode	*Event;*
Management solution	*Internal;*
Description	

Platform Ledger Management	
Model	*balance;*
Extra	*MPT support - modified Merkle Patricia tree (trie)*
Description	*Each account has a storage, a persistent memory area. A contract can neither read nor write to any storage apart from its own.* *From a block header there are 3 roots from 3 MPT: stateRoot, transactionsRoot and receiptsRoot.*

Section 5 Resources

Node Management	
Node Role	*Full Nodes and Full archiving nodes.*
Joining	*The node has to sync with the network and start to participate with the permission: must be included in the node whitelist smart-contract, managed by FinTech Association.*
Leaving	*Nodes can stop working at any time.*
Role changing	*FinTech Association manages nodes' roles based on consortium decisions.*
Description	*Consortium's participants who need more gas are not able to mine more tokens, or it will be switched off from the network.* *https://www.coindesk.com/russias-largest-bank-is-quitting-central-banks-blockchain-project*

Platform Data Storage Mechanism	
Mass storage mitigation	*Concept of Gas* *Some operations may have negative gas cost, for example kill a contract.*
Decentralized Data Storage Support	*MCMS – Masterchain Confidential Messaging System*
Data Privacy Solution	*N/A*
Description	*The fundamental unit of computation is called "gas"; The fee system is to require a person to pay proportionately for every resource that they consume, including computation, bandwidth and storage;*

Platform Network Management	
Node Scalability	*Thousands*
Network Structure	*Distributed*
Network Discovery Protocol	*Kademlia-like;*
Byzantine Node Accepted?	*Yes*
P2P?	*Yes*
Data Exchange Protocol	*RLP over TLS*
Description	https://github.com/ethereum/wiki/wiki/Kademlia-Peer-Selection

	RLP transport protocol, a TCP-based transport protocol used for communication among Ethereum nodes. The protocol carries messages belonging to one or more 'capabilities' which are negotiated during connection establishment. Messaging security is provided with a TLS connection layer.

Section 6 Utils

Platform Messaging Mechanism	
Protocol Type	*RPC*
Description	*JSON-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 process, over sockets, over HTTP, or in many various message passing environments. It uses JSON (RFC 4627) as data format.* https://github.com/ethereum/wiki/wiki/JSON-RPC

Platform Crypto Libraries	
Secure Network Connection Type	*Communication via public Internet (TCP with TLS + UDP).*
Cipher Suites	*Russian GOST 34-10.2012 for it's public-key cryptography and GOST 34-11.2012 for hashing*
Description	*Meth (The official Masterchain client node software) uses UDP connection to exchange information about the P2P network. After establishing peer addresses, Meth nodes exchange blockchain information via encrypted and authenticated TLS connections.*

Section 7 Operation & Maintenance

Platform system management – Node	
Log	*Yes*
Monitoring	*Masterchain Explorer*
Description	

Platform system management – Chain Network	
Permission Control	*Whitelist of nodes which are allowed to connect to the network and which are allowed to produce blocks*
Auditing	*N/A*
Supervisory Support	*N/A*

Description	*Masterchain Explorer shows information about blocks, transactions, tokens, smart contracts, addresses and the history of its transactions. Masterchain Explorer is operated and developed by FinTech Association and can be accessed by consortium members.*

Section 8 External Resource Management

Platform External Resource Management	
Interoperation solution	*Masterchain Confidential Messaging Service*
Description	*The system of smart contracts describing network participants, their roles and data objects which can be accessed by them*

Section 9 Extensions

Platform Extensions – optional	
Name	*Masterchain Confidential Messaging Service*
Extension type[1]	*internal*
Extension mode[2]	*vertical*
Solution	*The web service implementing GraphQL protocol to provide access to read and write data objects based on the rules of MCMS smart contract system.*
Serve domain	*Network/Application*
Description	

[1] 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”

[2] 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).