Attachment IX – Architecture Mapping of LACChain

Section 1 Summary

Platform summary	
Platform ID	*LACChain*
Status/Revision	*Pro Test-net*
Type	*Public-permissioned*
Domain	*Multi-sector, multi-community*
Description	*LACChain is a public-permissioned infrastructure developed by a global alliance led by the IDB Lab [1] to offer public, authenticated and regulated networks in Latin America and the Caribbean with digital identity and tokenized regulated fiat money as native layers. LACChain has three main techno-legal components: LACChain DLT, LACChain ID and LACChain TFM. [2]* *LACChain DLT: A set of public-permissioned, decentralized, transparent, interoperable, zero transaction fee, regulated and quantum-safe blockchain networks. Using Pantheon [3] and Quorum [4] as core software, LACChain DLT is developing specific taxonomy composed by validator, gear, writer and observer nodes; smart-contract-based permissioning managed by the Satellite Permissioning Committee (SPC) and the Core Permissioning Committee (CPM); a DAPP for managing permissioning; cloud integration for node deploy and maintenance; a smart-contract-based gas schema to prevent network collapse; and tools to monitor nodes activity and data analytics tools, among others features.* *LACChain ID: LACChain ID is a native layer of the LACChain infrastructure that enables and requires the identification and authentication of every node, using LACChain standards aligned with international standards from W3C, EEA and IEEE. LACChain ID is also developing an application for end-user self-sovereign identity (SSI) to be used to authenticate in any platform.* *LACChain TFM: LACChain TFM is a native layer of the LACChain infrastructure that enables the issuance and transference of tokenized fiat money. This tokenized money will be issued and controlled by regulated and authorized financial institutions, backed-up by fiat money and represented by digital tokens.* [1] https://www.iadb.org/en/news/global-alliance-promote-use-blockchain-latin-america-and-caribbean [2] https://medium.com/@lacchain.official/lacchain-la-internet-del-valor-74bdb649095

	[3] https://github.com/PegaSysEng/pantheon [4] https://github.com/jpmorganchase/quorum

Section 2 Governance & Compliance Functions

Platform governance	
Governance Type	*Public-permissioned*
Chain Network Admin	*NA/Community of authenticated and regulatory compliant entities (public)*
Pledge (cost of malicious action)	*Stake*
Tamper Proof (tamper cost)	>50%
Description	*LACChain is developing a techno-legal framework to operate the infrastructure at two layers. The first layer consists of core nodes that maintain the layer, permission new nodes and apply the consensus protocols. These nodes will guarantee services under SLAs and other legal agreements. Only the LACChain Partners will be permissioned as core nodes. LACChain Partners are those that contribute to the LACChain program. [5]* *The second layer consists of satellite nodes that run applications and generate transactions. Parties operating these nodes will be legally responsible for their activity and the services they provide. Any entity can deploy and maintain satellite nodes.* *The admission of new nodes is responsibility of the Satellite Permissioning Committee (SPC) and the Core Permissioning Committee (CPC). [6]* *At an infrastructure level, LACChain is enabling smart-contract-based permissioning. This allows the SPC and CPC to manage the permissioning of new nodes via DAPP. Once new nodes are permissioned, a DID and a set of verifiable credentials is issued to the node owner. [7]* [5] https://medium.com/@lacchain.official/lacchains-networks-taxonomy-651138a70346 [6] https://medium.com/@lacchain.official/lacchains-permissioning-protocols-f18e6290949a [7] https://medium.com/@lacchain.official/lacchains-networks-roadmap-600b58872e43

Platform trust endorsement policy

Type	*Legal agreements*
Tool	*SLAs, digital identity, smart contracts, off-chain legal agreements*
Policy	*LACChain techno-legal framework* *National regulations from Latin America and the Caribbean countries*

Economic Model (optional)	
Price Model to Deploy Contracts and do Transactions	*Zero transaction fee* *Membership model for participating in LACChain, which will include not only access to the infrastructure but also to additional services as training or data visualization*
Who pays the costs of the network	*Users, developers, nodes, 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	*Solcjs - Solidity*
Runtime VM	*EVM – Ethereum Virtual Machine*
DevTools	*Development: Visual Studio Code; Sublime; Remix;* *Build framework: Truffle, Embark, Remix, Web3j, Web3js, ethersjs* *Test framework: Truffle, Embark, Remix, Web3j, Web3js, ethersjs*
Extra Tool(s)	*Any Ethereum compatible Blockchain Explorer.* https://github.com/Councilbox/cbx-quorum-explorer https://github.com/gobitfly/etherchain-light
Lifecycle	*LACChain techno-legal framework will address the death of the network* *The developers of the applications and the satellite nodes providing services will be responsible for the services provided*
Description	*LACChain techno-legal framework*

	Github repository

Section 4 Protocol

Platform AAA Management	
Account type	*Authentication mandatory at a node level and recommended at a user level*
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* *LACChain ID will provide a suite of standards and recommendations in the different areas composing digital identity, as DIDs, Verifiable Claims, Verifiable Presentations, Levels of Assurance, Key Management System or Data Privacy* *LACChain ID will provide an application following the LACChain ID standards and recommendations, for users to generate and manage DIDs and certificates. This application provided will be intended to be used by other applications or platforms*
AAA support	*LACChain ID*
Description	*LACChain ID*

Platform Consensus Mechanism	
Algorithm	*IBFT2.0*
Consensus mode	*Event, state*
Management solution	*Internal*
Description	*LACChain techno-legal framework*

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.* https://github.com/PegaSysEng/pantheon/tree/master/ethereum/trie/src/main/java/tech/pegasys/pantheon/ethereum/trie https://github.com/jpmorganchase/quorum/tree/master/trie

Section 5 Resources

Node Management	
Node Role	*Validator node: Apply consensus protocol. Maintain the ledger.* *Gear node: Connect validator nodes with satellite and observer nodes. Set up new nodes.* *Writer node: Generate transactions. Provide services to end-users.* *Observer nodes: Read the blockchain.*
Joining	*Validator and gear nodes will be required to sign SLAs and prove technical capabilities. Only LACChain Partners will be allowed to run these nodes.* *Writer and observer nodes will be legally accountable for the activity of their node and the transactions it generates.* *Every node must be identified and operate authenticated.*
Leaving	*Validator and gear nodes have to notify as agreed in the SLAs.* *Writer and observer nodes can leave at any time without notification.*
Role changing	*A node can change the role as long as it satisfied the conditions for performing the new role.*
Description	*LACChain techno-legal framework*

Platform Data Storage Mechanism	
Mass storage mitigation[1]	*Concept of Gas* *Some operations may have negative gas cost, for example kill a contract.*
Decentralized Data Storage Support	*IPFS, cloud-services*
Data Privacy Solution	*LACChain techno-legal framework will specify the policies related to data privacy*

[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.

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; However, gas fee will not be associated with a monetary fee; Gas will be distributed at no cost for the users, ensuring the correct functioning of the network*

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://github.com/PegaSysEng/pantheon/tree/master/ethereum/rlp https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf

Section 6 Utils

Platform Messaging Mechanism	
Protocol Type	*RPC* *GraphQL*
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/PegaSysEng/pantheon/blob/master/docs/Pantheon-API/JSON-RPC-API.md 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* *New quantum-safe cipher suites will be introduced*
Description	*Geth (modified) and Pantheon Enterprise Ethereum Client uses UDP connection to exchange information about the P2P network. After establishing peer connections, nodes exchange blockchain information via encrypted and authenticated TCP connections.* https://github.com/PegaSysEng/pantheon/blob/master/docs/index.md https://github.com/jpmorganchase/quorum/tree/master/ethclient

Section 7 Operation & Maintenance

Platform system management – Node	
Log	*Yes*
Monitoring	http://netstats.testnet.lacchain.io/
Description	*Network status allows anyone to see the performance and number of nodes and where they are located.* *There is no special nodes (masternodes, special block producers etc) in the network.* https://github.com/PegaSysEng/pantheon/blob/master/docs/Monitoring/Monitoring-Performance.md https://github.com/jpmorganchase/quorum/blob/master/docs/Privacy/Tessera/Usage/Monitoring.md

Platform system management – Chain Network	
Permission Control	*Smart contract-based*
Auditing	*Information public available in blockchain explorers* *Public information from DIDs, verifiable credentials and similar will be captured and used to measure the social impact achieved through the use of LACChain*
Supervisory Support	*N/A*
Description	*N/A*

Section 8 External Resource Management

Platform External Resource Management

Interoperation solution	
Description	

Section 9 Extensions

Platform Extensions – optional	
[the following list can be duplicated for multiple extensions]	
Name	
Extension type[2]	
Extension mode[3]	
Solution	
Serve domain	
Description	

[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).