# Attachment I – Architecture Mapping of Alastria (Quorum version)

## Section 1 Summary

| Platform summary | |
|---|---|
| **Platform ID** | *Alastria (Quorum version)/ALAQ* |
| **Status/Revision** | *Mainnet for limited use cases – 1.0* |
| **Type** | *Public-Permissioned* |
| **Domain** | *Many sectors* |
| **Description** | *Alastria (Quorum version) is a country-level Public-Permissioned open-source platform for decentralized applications, compatible with Ethereum but with a proper governance model for better performance, better decentralization and better compliance with regulations.* |

## Section 2 Governance & Compliance Functions

| Platform governance | |
|---|---|
| **Governance Type** | *Public-Permissioned, governed as Common-Pool Resource* |
| **Chain Network Admin** | *Non-profit association, open to anybody.*<br>*Commissions, Working Groups and General Assembly.*<br>*AIP – Alastria Improvement Proposal* |
| **Pledge (cost of malicious action)** | *Node compromise (standard BFT enhanced with Proactive Recovery)* |
| **Tamper Proof (tamper cost)** | *Rewriting history is impossible (assuming Ethereum digital signatures are safe), even with all consensus nodes compromised.*<br>*Malicious actors can not invent or modify transactions, just censor them (only if more than 1/3 of nodes compromised). But this event can be detected and actions taken against the responsible (Alastria operates in an efficient legal environment).*<br>*Any malicious action not avoided by IBFT can be audited post-facto and the responsible entity penalized accordingly.* |
| **Description** | *Alastria (Quorum version) is a Public-Permissioned network using Quorum (a light fork of Ethereum), operating in a  good legal environment (inside the European Union).* |

| | |
|---|---|
| | *The network is promoted by non-profit association, with a governance style "Common-Pool Resource". Any entity can join the network, but requires node permissioning. Network is collaboratively operated by the members. Anti-trust rules and no artificial or arbitrary rules of entry. The governance bodies allow the participation of any member of the association.*<br>*Being permissioned and using IBFT, Alastria (Quorum version) has "transaction finality" (required for most transactions in the productive economy), and the protocol does not allow reorgs or chain forks, so it is impossible that malicious actors can rewrite the history of the chain in a honest node.*<br>*In addition, digital signatures are used for everything (transactions and consensus execution), so any malicious action is auditable at least post-facto. Alastria monitoring tools make any such actions transparent to the whole set of members. Incentive to cheat is extremely low.* |

| **Platform trust endorsement policy** | |
|---|---|
| **Type** | *Common-Pool Resources collaborative governance, coupled with Legaly binding SSI identities and transparent consensus execution monitoring tools.* |
| **Tool** | *Permissioning, IBFT, extensions to BFT (proactive recovery), digital signatures, consensus execution transparent monitoring tools.* |
| **Policy** | *Common-Pool Resources, auditability, transparency, fairness.* |

| **Economic Model (optional)** | |
|---|---|
| **Price Model to Deploy Contracts and do Transactions** | *Deploy a new contract is a kind of transaction,*<br>*Charged by transactions only* |
| **Who pays the costs of the network** | *Users* |
| **Monetary Policy of Tokens** | *There is no cryptocurrency embedded in the infrastructure, and there is no mining.*<br>*However, Gas (the same concept from Ethereum) is used both to protect the network and to meter the usage by the users. However, in Alastria (Quorum version), the Gas is not associated to any token or cryptocurrency subject to speculation.*<br>*The incentives of the consensus nodes are long-term and associated to the Common-Pool Resource governance style (basically, all members of the association need a viable the network to run their transactions in the real economy, which is where they obtain the economic benefits).* |

| | |
|---|---|
| **Rights of Tokens** | *Not applicable* |

## Section 3 Application

| **Platform Smart Contract mechanism** | |
|---|---|
| **Language** | *Same as Ethereum* |
| **Turing Complete?** | *Yes – Solidity* |
| **Compiler** | *Solidity or any other which compiles to the Ethereum VM* |
| **Runtime VM** | *EVM;* |
| **DevTools** | *Same as Ethereum* |
| **Extra Tool(s)** | *Alastria Block Explorer (Block data view)*<br>*Alastria NetStats ()* |
| **Lifecycle** | *Same as Ethereum (the network is permissioned only when joining a node, afterwards the members are autonomous)* |
| **Description** | *It is mostly compatible with Ethereum* |

## 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.*<br>*User should generate an externally owned account using a local software/hardware in order to keep the private key private;*<br>*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** | *IBFT enhanced with Proactive Recovery and Stand-by Consensus Set* |
| **Consensus mode** | *Event;* |
| **Management solution** | *Internal;* |
| **Description** | *The basic consensus algorithm is IBFT. At a given moment there is an Active Set of consensus nodes creating the blocks, and a Stand-by Set of nodes ready to be included in the Active Set. In a proactive way (even in the absence of failures), consensus nodes are rotated from the Active Set to the Stand-by Set and vice versa. When a node changes from Active Set to Stand-By set, the node is "rejuvenated" using safe techniques (eg. from safe read-only storage) to eliminate any infection made by a malicious actor while the node was in the Active Set.*<br>*The recovery mechanism allows the system to tolerate any number of faults over the lifetime of the system provided fewer than 1/3 of the replicas become faulty within a small window of vulnerability.*<br>*The Stand-By set increments further the robustness, inclusivity and trust on the system, by allowing any member of the association that so wishes to participate in the consensus algorithm even if it is BFT.* |

| 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.*<br>*From a block header there are 3 roots from 3 MPT: stateRoot, transactionsRoot and receiptsRoot.* |

## Section 5 Resources

| Node Management | |
|---|---|
| **Node Role** | *Regular nodes, Consensus nodes and Bootstrap (permissioning) nodes.* |
| **Joining** | *For Regular nodes, the member of the association has to request permissioning for its node, which is always granted as per the rules of Alastria.The Regular nodes can use Bootstrap or other Regular nodes to sync with the network and start participating.*<br>*For Consensus and Bootstrap nodes, the process is basically the same but the member has to agree on compliance to a much more stringent* |

| | |
|---|---|
| | *Technical and Operational Policy. Any member complying with those policies can become a Consensus or Bootstrap node.* |
| **Leaving** | *Regular nodes can stop working at any time.*<br>*Consensus and Bootstrap nodes have agreed to a 24x7 operation and some other compromises to allow for stability of the network, even though there is not a standard formal SLA contract.* |
| **Role changing** | *A node can change role if they follow the stablished procedures.* |
| **Description** | - |

| **Platform Data Storage Mechanism** | |
|---|---|
| **Mass storage mitigation**[1] | *Concept of Gas*<br>*Some operations may have negative gas cost, for example kill a contract.* |
| **Decentralized Data Storage Support** | *Same as Ethereum (IPFS, etc). In addition there is in the roadmap a decentralized storage system more compatible with privacy and regulation than IPFS, especially for AlastriaID (SSI).* |
| **Data Privacy Solution** | *Members can use Private Transactions provided by Quorum.* |
| **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; in Alastria the "gas" is not associated to any cryptocurrency or token susceptible to speculation.* |

| **Platform Network Management** | |
|---|---|
| **Node Scalability** | *Regular nodes: Thousands as in Ethereum.*<br>*Consensus and Bootsrap nodes: tens to hundreds.* |
| **Network Structure** | *Distributed* |
| **Network Discovery Protocol** | *Kademlia-like;* |
| **Byzantine Node Accepted?** | *Yes* |
| **P2P?** | *Yes* |
| **Data Exchange Protocol** | *RLPx* |

[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** | https://github.com/ethereum/wiki/wiki/Kademlia-Peer-Selection<br>*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.*<br>https://github.com/ethereum/devp2p/blob/master/rlpx.md |

## 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.*<br>https://github.com/ethereum/wiki/wiki/JSON-RPC |

| **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*<br>*There is a discussion about how these algorithms were implemented at:* https://ethereum.stackexchange.com/questions/71657/cipher-suites-open-source |
| **Description** | *Alastria uses Quorum, a light fork of Geth (The official Ethereum client node software), which uses UDP connection to exchange information about the P2P network. After establishing peer connections, Geth nodes exchange blockchain information via encrypted and authenticated TCP connections.* |

## Section 7 Operation & Maintenance

| **Platform system management – Node** |
|---|

| | |
|---|---|
| **Log** | *Yes* |
| **Monitoring** | *Alastria NestStats and Alastria Block Explorer* |
| **Description** | *Any member of Alastria can see the basic network operation.*<br>*Alastria Netstats also displays basic information abot execution of consensus algorithm by individual consensus nodes, so they can be made accountable in front of the whole community.*<br>*Additional monitoring tools can provide more detailed information to any member wishing to control or audit consensus nodes operation (eg, by Regulators, Civil society organizations or other entities interested in transparency).* |

| **Platform system management – Chain Network** | |
|---|---|
| **Permission Control** | *Currently manual, in roadmap to make it automatic, thanks to participation of proper official bodies and Trust Framework (eg. any business registered in the official Business Registry of Spain can participate).* |
| **Auditing** | *Information public available in Alastria Netstats and blockchain explorer (for members and non-members)* |
| **Supervisory Support** | *N/A* |
| **Description** | *Alastria Block explorer shows information about the network and blocks, transactions, tokens, smart contracts, addresses and the history of its transactions. Any member can operate its own block explorer in parallel.* |

## 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).