**Attachment III  – Architecture Mapping of BTC**

**Section 1 Summary**

|  |  |
| --- | --- |
| **Platform summary** | |
| **Platform ID** | *Bitcoin / BTC* |
| **Status/Revision** | *Core Version 0.18.0* |
| **Type** | *Public* |
| **Domain** | *Peer to peer payments, Financial* |
| **Description** | *Bitcoin is the first global, open source peer-to-peer decentralized monetary system. It is based on the original Bitcoin white paper published by the anonymous Satoshi Nakamoto.*  <https://bitcoin.org/bitcoin.pdf>  <https://www.lopp.net/bitcoin-information.html>  [https://bitcoin.org/en/release/v0.18.0#wallet-gui](#4d34og8) |

**Section 2 Governance & Compliance Functions**

|  |  |
| --- | --- |
| **Platform governance** | |
| **Governance Type** | *Permissionless* |
| **Chain Network Admin** | *Community (public)*  *Bitcoin Improvement Proposal (BIP)* |
| **Pledge (cost of malicious action)** | *Resources (hardware + electricity) – measured by hash rate (H/s)* |
| **Tamper Proof (tamper cost)** | >50% of network H/s |
| **Description** | *“Bitcoin Core” is the main implementation of the node software and acts as the de-facto protocol specification. Bitcoin Core is an open governance model where everyone is free to propose and discuss changes to the system through BIP.*  *A BIP is a design document for introducing features or information to Bitcoin. This is the standard way of communicating ideas since Bitcoin has no formal structure.*    [*https://bitcoin.org/en/bitcoin-core/contribute/*](https://bitcoin.org/en/bitcoin-core/contribute/)  [*https://github.com/bitcoin/bips*](https://github.com/bitcoin/bips)  [*https://en.bitcoin.it/wiki/Bitcoin\_Improvement\_Proposals*](https://en.bitcoin.it/wiki/Bitcoin_Improvement_Proposals) |

|  |  |
| --- | --- |
| **Platform trust endorsement policy** | |
| **Type** | *Tokenomics* |
| **Tool** | *BTC* |
| **Policy** | *Schelling point, mechanism design with Proof of Work consensus, bounded rationality, specialised ASICs as a grim trigger policy* |

|  |  |
| --- | --- |
| **Economic Model (optional)** | |
| **Price Model to Deploy Contracts and do Transactions** | *Bitcoin supports a limited set of smart contract functionalities. These are charged per transaction.* |
| **Who pays the costs of the network** | *Users* |
| **Monetary Policy of Tokens** | *Finite supply of BTC: 21,000,000 BTC*  *No pre-mine.*  *Currently 12.5 new BTC are minted per block as rewards for miners. The number of new BTC minted per block halves every 210,000 blocks, approximately every 4 years. Next halving will occur around June 2020.* |
| **Rights of Tokens** | *N/A* |

**Section 3 Application**

|  |  |
| --- | --- |
| **Platform Smart Contract mechanism** | |
| **Language** | *C++* |
| **Turing Complete?** | *No* |
| **Compiler** | *N/A* |
| **Runtime VM** | *N/A* |
| **DevTools** | *Bitcoin Script IDE and list of other DevTools and Resources*  <https://www.lopp.net/bitcoin-information/developer-tools.html> |
| **Extra Tool(s)** | *List of websites providing Bitcoin network statistics*  <https://www.lopp.net/bitcoin-information/statistics-metrics.html> |
| **Lifecycle** | *N/A* |
| **Description** | *N/A* |

**Section 4 Protocol**

|  |  |
| --- | --- |
| **Platform AAA Management** | |
| **Account type** | *UTXO* |
| **Distributed ID** | *There is no identification system attached to wallet addresses.* |
| **AAA support** | *N/A* |
| **Description** | *N/A* |

|  |  |
| --- | --- |
| **Platform Consensus Mechanism** | |
| **Algorithm** | *SHA-256* |
| **Consensus mode** | *Hashcash Proof of Work (PoW)* |
| **Management solution** | *Internal* |
| **Description** | *Bitcoin uses the hashcash Proof\_of\_work function as the mining core. All bitcoin miners (whether CPU, GPU, FPGA or ASICs) are expending their effort creating hashcash proofs-of-work which act as a vote in the blockchain evolution and validate the blockchain transaction log.*  *More information may be found here:*  [*https://en.bitcoin.it/wiki/Proof\_of\_work*](https://en.bitcoin.it/wiki/Proof_of_work)  [*https://en.bitcoin.it/wiki/Hashcash*](https://en.bitcoin.it/wiki/Hashcash) |

|  |  |
| --- | --- |
| **Platform Ledger Management** | |
| **Model** | *Balance* |
| **Extra** | *Merkle tree* |
| **Description** | *Each block contains a list of transactions that it validates. The header contains, among other things, (i) the root of the merkle tree of these transactions, (ii) the hash of the previous block, a “nonce” number that the miners can arbitrarily set, and (iii) the hash of the block itself.*  *The hash of the block itself must be below a certain difficulty target. The process of finding a nonce producing a block hash below a certain difficulty target is what makes proposing a new block difficult.*  *Due to hashing function (SHA256) characteristics, there is no other way than to use brute force to find the nonce until a satisfying block hash is found, giving a statistical “proof of work”. On average, N-different hashes will have to be tried by all miners to find a single satisfactory result.* |

**Section 5 Resources**

|  |  |
| --- | --- |
| **Node Management** | |
| **Node Role** | *Full mining validating nodes and full non-mining validating nodes* |
| **Joining** | *No permission is required for joining the network. One can simply set up a node and begin the Initial Block Download (IBD)* |
| **Leaving** | *Nodes can discontinue operation at any time.* |
| **Role changing** | *Nodes can independently change roles at any time.* |
| **Description** | *Full node info and basic hardware requirements:* [*https://bitcoin.org/en/full-node#what-is-a-full-node*](#2s8eyo1)  *IBD info:* [*https://bitcoin.org/en/full-node#initial-block-downloadibd*](#17dp8vu) |

|  |  |
| --- | --- |
| **Platform Data Storage Mechanism** | |
| **Mass storage mitigation[[1]](#footnote-1)** | *N/A* |
| **Decentralized Data Storage Support** | *N/A* |
| **Data Privacy Solution** | *N/A* |
| **Tamper Proof (tamper cost)** | *N/A* |
| **Description** | *N/A* |

|  |  |
| --- | --- |
| **Platform Network Management** | |
| **Node Scalability** | *No upper bound* |
| **Network Structure** | *Distributed* |
| **Network Discovery Protocol** | *TCP* |
| **Byzantine Node Accepted?** | *Yes* |
| **P2P?** | *Yes* |
| **Data Exchange Protocol** | *Gossip;* |
| **Description** | *More information may be found here:*  [*https://en.bitcoin.it/wiki/Network*](https://en.bitcoin.it/wiki/Network) |

**Section 6 Utils**

|  |  |
| --- | --- |
| **Platform Messaging Mechanism** | |
| **Protocol Type** | *N/A* |
| **Description** | *N/A* |

|  |  |
| --- | --- |
| **Platform Crypto Libraries** | |
| **Secure Network Connection Type** | *SSL; TLS.* |
| **Cipher Suites** | *ECDSA; Secp256k1* |
| **Description** | *Elliptic Curve Digital Signature Algorithm or ECDSA is a cryptographic algorithm used by Bitcoin to ensure that funds can only be spent by their rightful owners.*  *More information may be found here:*  [*https://en.bitcoin.it/wiki/Elliptic\_Curve\_Digital\_Signature\_Algorithm*](https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm) |

**Section 7 Operation & Maintenance**

|  |  |
| --- | --- |
| **Platform system management – Node** | |
| **Log** | *Yes* |
| **Monitoring** | *bitcoind* |
| **Description** | *bitcoind is the daemon client that manages all interactions with the Bitcoin network. It also acts as the interface between wallet software and the Bitcoin network. A number of log levels may be activated with the software. It is a crucial element of node management.* |

|  |  |
| --- | --- |
| **Platform system management – Chain Network** | |
| **Permission Control** | *N/A* |
| **Auditing** | *Auditing mechanisms are self-contained within each wallet and pertains to each wallet address managed by the wallet software.*  *Anyone can audit the history and current balance associated to any address by having a copy of the blockchain or using a public “block explorer” that facilitates visualizing this information.* |
| **Supervisory Support** | *N/A* |
| **Description** | *N/A* |

**Section 8 External Resource Management**

|  |  |
| --- | --- |
| **Platform External Resource Management** | |
|  |  |
| **Interoperation solution** | *N/A* |
| **Description** | *N/A* |

**Section 9 Extensions**

|  |  |
| --- | --- |
| **Platform Extensions - optional** | |
| *[the following list can be duplicated for multiple extensions]* | |
| **Name** | *Lightning Network* |
| **Extension type[[2]](#footnote-2)** | *Second Layer Interaction Solution* |
| **Extension mode[[3]](#footnote-3)** | *Hash Time Locked Contracts (HTLCs)* |
| **Solution** |  |
| **Serve domain** | *Financial Transactions* |
| **Description** | *Lightning Network is a proposed implementation of Hashed Timelock Contracts (HTLCs) with bi-directional payment channels which allows payments to be securely routed across multiple peer-to-peer payment channels. This allows the formation of a network where any peer on the network can pay any other peer even if they don't directly have a channel open between each other. As of March 2019, there were more than 37,000 channels carrying more than 764 bitcoins.* |

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. [↑](#footnote-ref-1)
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” [↑](#footnote-ref-2)
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). [↑](#footnote-ref-3)