Blockchain Oracles – Notes

* Blockchains are public databases (ledgers) that track validated transactions in a series of blocks.
* Blockchains utilize a validation mechanism known as a consensus algorithm.
* Blockchain generations:
  + First generation: Bitcoin
  + Second generation: Ethereum (Added programmability to blockchain technology through smart contract execution)
* Blockchain systems: Ethereum, Hyperledger Fabric, Cosmos, Polkadot, Chainlink, AION.
* The first block is known as genesis.
* Blockchain classes:
  + public (permissionless)
    - Participants in permission-less blockchains have access to the ledger without authentication.
    - Examples: Bitcoin and Ethereum
  + private (permissioned)
    - Permissioned blockchains require user authentication.
    - Examples: Hyperledger Fabric, Corda, Quorum, Tendermint, and Multi-chain
* A consensus is a strategy used by a group of computers to agree on what is true.
* Failure modes:
  + Fail-stop failure: (sender does not get a reply from the recipient)
  + Byzantine failure (gets a fabricated response ‘‘tampered with’’)
* Consensus categories:
  + Symmetric: allows any server or node participating in the system to respond to a write request
  + Asymmetric: allows only selected leaders to respond to requests and issue commands, leaders are elected by candidates.
* The proof of work and proof of stake were the earliest consensus algorithms proposed for blockchains.
  + They are typically slow because most work is proof based and require huge amounts of computations.
  + Other powerful consensus algorithms: practical Byzantine fault tolerance and delegated proof of stake.
  + More distributed consensus algorithms have been proposed, such as Paxos, Raft, and Calvin.
* In recent years, considerable effort has been directed toward enabling interoperability among blockchains, whether homogeneous (similar) or heterogeneous (different).
* Practical solutions are still limited and lack standardization among various types of blockchains.

1) CROSS-BLOCKCHAIN COMMUNICATION

* Cross-Chain Communication Protocol (CCCP)
  + Defines the correct synchronization of cross chain transactions between interacting blockchains.
  + allowed for homogeneous blockchain communication.
* Cross-Blockchain Communication Protocol (CBCP)
  + Involves a blockchain where a transaction is initiated (source) and a (target) blockchain where the transaction should be executed.
  + Defines the synchronization of cross-blockchain transactions.
  + allowing heterogeneous blockchains to communicate
* Interoperability in general is the process of exposing the blockchain’s internal state to others.
* Cross-chain asset transfers:
  + rely on Cross-Chain Communication Protocol (CCCP)
  + follows a different methodology involving three phases:
    - (1) asset lock on the source blockchain
    - (2) commitment of the transfer by the blockchain
    - (3) asset creation on the target blockchain.
* A trusted third party is crucial for a CCC protocol to withstand misbehaving nodes.
  + Trusted third parties can either be:
    - centralized (such as trusted validators)
    - decentralized (another blockchain)
* Distributed consensus is used by cross-chain protocols as an abstraction for trusted third parties.
* The global ledger state is agreed upon by participants via consensus algorithms.

2) INTEROPERABILITY DATA TYPES

* Two data types for Blockchain-to-Blockchain interoperability:
  + Digital asset exchange
    - This allows for movement or exchange of assets.
    - Example: cryptocurrencies between multiple blockchains.
    - This capability is supported by blockchains with simple programmable options.
  + Arbitrary data exchange
    - This allows one blockchain to impact another blockchain.
    - This could be something like blockchain-to-blockchain API calls.
    - An event to take place on one blockchain as a result of a smart contract code invocation on another blockchain.

3) INTEROPERABILITY CLASSIFICATION

* Two main approaches:
  + Buterin’s classification.
  + Table

    Description automatically generatedWorld Economic Forum’s classification.
* Oracles
  + Oracles are digital agents that aim to fetch external world information into a blockchain.
  + One of the strengths of Oracles is that they are easy to implement, providing data feed about external events.
  + Chainlink is an example of a decentralized data-feed Oracles system.
  + Oracles are considered a means of supporting interoperability between different blockchains.
  + Interoperability can be classified into two major types:
    - On-chain (a third blockchain is used to overpass two different blockchains).
      * This method is used in projects, such as AION, wan-chain, and ICON).
    - Off-chain (interoperability is achieved by middleware)
  + Oracles are one of the off-chain techniques that facilitate communication across enterprise systems and blockchains.
  + Oracles were classified based on different aspects:
    - network administration of nodes (trust)
      * Centralized Oracles: relies on a single source of data or an Oracle running on a single server.
      * Decentralized Oracles (distributed): resolves the single point of failure problem. Distributed Oracles are multiple Oracles servers forming a peer-to-peer network.
    - type of data source
      * Software Oracles: Online sources, such as APIs, websites, servers, or even other smart contracts are used to fetch data.
      * Hardware Oracles: Hardware Oracles feed in data from the real world.
      * Human Oracles.
    - direction of data flow
      * Inbound Oracles: Pull data from data sources (off-chain) to smart contracts (on-chain).
      * Outbound Oracles: push data from smart contract to the external world.
* Consensus-based Oracles as a type in which data fed to blockchains is based on the consensus carried between all Oracles participating in the query.
* Oracles have not been studied in their fundamental aspects, they assumed that this gap is addressed by learning and presenting Oracles patterns from two views:
  + (i) The data flow direction (whether inbound or outbound) from the blockchain’s side.
  + (ii) The initiator of the data flow (whether the communication is pull or push-based).
* Major issue is that data required for blockchain operations has first to be published on the blockchain; otherwise blockchains cannot function on them.
* It is preferable to look for off-chain options, such as Oracles, rather than attempting to tackle on-chain scaling options targeting network speed enhancement, which are still in the early stages of development.
* In blockchains, the execution of standard transactions is replicated at each node, which is not the case for smart contracts.
  + For instance, instead of on-chain recording for an entire temperature data stream from a warehouse sensor, off-chain temperature pre-processing is carried out, where only the minimum temperature reading, maximum, and average values are recorded on-chain once per day.
  + Complex processing can also be performed off-chain, with the results recorded on-chain.
  + Flexible privacy controls can be set off chain to control on-chain information exchange rules.
* Blockchain execution environments are insulated from the outside world, necessitating the use of blockchain Oracles to fetch off-chain data for on-chain use.

Diagram

Description automatically generated

COMPARING ORACLES TO VARIOUS INTEROPERABILITY TECHNIQUES

* Sidechains: scalability is significantly enhanced using side chains, where batches of transactions are processed before they are submitted to the main blockchain.
  + Drawback: sidechain logic can be invalidated if the main chain is compromised
* Oracles do not require compatibility between different blockchains because they serve as intermediaries between different blockchains in which they do not need to directly interact.
* Table

  Description automatically generatedCentralized off-chain components (Oracles) were claimed to be points of failure in the entire blockchain system. This was resolved using decentralized Oracles solutions.

ORACLES’ LATEST MARKET SOLUTIONS

* Oracles fall into two main categories:
  + Data Feed Oracles: Act as an intermediary between business-level smart contracts and off-chain events.
  + Computation Oracles: Perform user-defined off-chain computation tasks for blockchains.