Lecture-6-
BCT/DLT Terminology
(19 Aug 2019)

Dhiren Patel

VJTI Mumbai, India

Ack/Ref:

ISO/TC 307/WG 1 N 528

ISO/TC 307/WG 1
Foundations

Email of convenor: g.goodell@ucl.ac.uk
Convenorship: BSI (United Kingdom)

CD 23257 Blockchain RA 2019-08-09 CD updated clean

Replaces: N 523

Document type: Committee draft

Date of document: 2019-08-09

Expected action: INFO

Background:

Committee URL: https://isotc.iso.org/livelink/livelink/open/tc307wg1

Why BCT/DLT?

- Records of transactions, based on certain agreed upon conditions form the basis for exchanging assets between two parties.
- (the frictions in the processes, such as ascertaining the real owner of the asset, capital/asset is locked for days until the transaction is cleared, still exists as they could not be solved by technology until now).
- By maintaining an immutable, distributed ledger in a distributed network, blockchain and Distributed Ledger Technology (DLT) systems has the potential to reduce the friction in business networks and thus the operating cost for business transactions…

DLT

- While historically, the focus has been facilitating the exchange of assets, this technology can also be used as a broader solution to public reporting and auditing requirements

Definitions

- Personally Identifiable Information (PII) – any information that (a) can be used to establish the link between the set of PIIs and the natural person to whom such information relates, or (b) is or might be directly or indirectly linked to a natural person
- delete - permanently remove transaction records from a distributed ledger
- purge - permanently erase and remove data from a storage space
- Note: delete or purge cannot be done to records that have been validated and added to the distributed ledger with consensus
- archive - data, transactions or resources saved for later reference or use, possibly off-line

Definitions

- data archiving - digital preservation processing that moves data into a managed form of storage for long-term retention
- backup - process to copy/export data, transactions or resources to the data storage of a backup system to enable retrieval and restoration this data in case of an incident or disruption
- restore - process to recreate record entries, data, transactions or resources from a previously created backup or archive of the records, data or resources.
- Note: Restore needs to be done in a way that preserves the integrity (same number of and content for all records) and content of the DLT ledger

Definitions

- resilience - ability of a system to recover operational condition quickly following an incident or disruption
- incident - anomalous or unexpected event, set of events, condition, or situation at any time during the life cycle of a project, product, service, or system
- disruption - incident, whether anticipated (e.g. hurricane) or unanticipated (e.g. power failure/outage, earthquake, or attack on ICT systems/infrastructure) which disrupts the normal course of operations at an organization location

Abbreviations

- API Application programming interface
- CA Certificate authority
- DLT Distributed Ledger Technology
- ICT Information and communication technology
- P2P Peer-to-peer
- PBFT Practical Byzantine Fault Tolerance
- PII Personally identifiable information
- SSID Self-sovereign identity

Ledger

- ledger - an information store that keeps a final and definitive record of transactions
- Ledgers can be used to record transactions of almost any type: for example, the movements and transfers of physical objects.
- distributed ledger is a ledger that has its entries stored, across a series of nodes in a network, rather than in a single location
- tamper-resistant – transaction records, once entered into the ledger, cannot be altered or cannot be altered without the alteration being clearly evident on inspection, whether the alteration is deliberate or accidental, malicious or benign
- tamper-evident - a system that has the desirable characteristic of enabling any unauthorized changes to be clearly visible

DLT

- Distributed ledger technologies (DLT) are designed to implement distributed ledgers
- a significant challenge due to the need to agree on and maintain the transaction records in the distributed ledger
- ensure that every replicated version of a transaction is the same across all the nodes where it is stored
- The set of records in the distributed ledger needs to be verifiable and auditable

DLT and Blockchain

- DLT in this way does not imply that every node in the network stores exactly the same set of transaction records
- It also does not imply that every party which participates in the distributed ledger has access to all the transaction records
- processing batches of transactions in cryptographically secured data structures known as blocks
- A valid protocol needs to ensure that each block is cryptographically linked to an immediately previous block forming a unique sequence of blocks in time.
- The complete sequence of cryptographically associated blocks forms a globally accessible append -only data structure - the blockchain - which supplies the canonical version of the global transaction history

Blockchain

- blockchain protocols include a consensus mechanism that provides a total ordering of all transactions within the block.
- The ledger update process is understood to be partially synchronous such that the collective action of all nodes in the DLT network functions as a timestamp server that validates pending transactions and updates the current ledger state by sequentially appending blocks to the ledger or blockchain

DLT varieties

- In a typical non-blockchain DLT, a transaction itself is a record of ledger instead of a block.
- Also, the ledger structure is not a chain, but a graph which travels in one direction without cycle
- DAG doesn‘t need miners to confirm each transaction, which results in highly accelerated process for confirmation and low transaction cost without any miner‘s incentive.
- However, ordering of transactions could be different between nodes, which requires additional consideration for synchronization.

Forks (update/reconfigure Blockchain)

- **Hard fork**
  A hard fork is a rule change such that the software validating according to the old rules will see the blocks produced according to the new rules as invalid.
- In case of a hard fork, all nodes meant to work in accordance with the new rules need to upgrade their software.
- **Soft fork**
  In contrast to a hard fork, a soft fork is a change of rules that still creates new blocks recognized as valid by the old software, i.e., it is backwards-compatible.
- As for a hard fork, a soft fork can also split the blockchain when non-upgraded software creates blocks not considered valid by the new rules.

Blockchain (revisited)

- The term "blockchain" is commonly applied both to the structure of some transaction databases and as well as the complete implementation of a distributed ledger which uses the blockchain transaction database.

Characteristics

- Ledger storage architecture
- Ledger control architecture
- Ledger sub-setting
- Ledger permissions
- Ledger implementation

Consensus

- Consensus in the context of DLT systems, addresses the problem of agreeing on the order of blocks
- in a potentially widely distributed system where there could be competing new blocks being added by multiple nodes across this network
- Consensus mechanisms figure out how all these independent nodes in the DLT come to an agreement about the contents and order of these blocks

Consensus

- There are different aspects of consensus, usually resolved over different timescales. First, there is the question of consensus of what transactions or blocks are valid.
- The initial and ongoing acceptance of the validation mechanisms is often established by kinds of informal social consensus (public blockchains or DLT systems) or by contractual means (private blockchains or DLT systems).
- Second, there is the question of which valid transactions need to be included in the most recent block (and by extension, all subsequent blocks).

ledger storage architecture

- an architecture that may be:
- ***Centralized*** (ledger storage) architecture has a central server (broker) which stores a single complete instance of the ledger
- ***Distributed*** (ledger storage) architecture is where each node may store a full or partial replica of the distributed ledger

ledger control architecture

associated with ledger consensus mechanism / model

- an architecture that may be:
- ***Centralized*** (ledger control) architecture is where a central server (or authority) controls the decision making relating to the distributed ledger (i.e. the validation of the new blocks of records).
- ***Distributed*** (ledger control) architecture is where all architecture elements (particularly nodes in the DLT system) control the decision making relating to the distributed ledger, based on a consensus mechanism.

Ledger subsetting and Permissions

- In the context of DLT systems, **ledger subsetting** may be:
- Subset of the ledger (i.e. partial replica)
- Full-set of the ledger (i.e. full replica)

- In the context of DLT systems, **ledger permissions** may be:
- Permissionless (i.e. free access to the ledger)
- Permissioned (i.e. restricted access to the ledger)

Proof of Work

- Involves solving a computational challenging puzzle in order to validate transactions and create new blocks in the Bitcoin blockchain
- to link the new block to the last block in the valid blockchain
- the process is known as ‘mining’, and the nodes in the network that engage in mining are known as ‘miners’ - incentive in economic payoffs
- Changing a block (which can only be done by making a new block containing the same predecessor) requires regenerating all successors and redoing the work they contain (calculating the entire chain of ‘hard mathematical problems’) which is practically impossible.
- This protects the blockchain from tampering.
- Principle - A solution that is difficult to find but is easy to verify
- Bitcoin PoW - Given data A, find a number x such as that the hash of x appended to A results in a number less than B (to a number of leading 0’s)
- Proof of Stake - the creator of the next block is chosen via various combinations of random selection and wealth or age (*i.e.,* the stake)

Byzantine Agreement

- A Byzantine Agreement (BA) is reached when a certain minimum number of nodes (known as a quorum) agrees that the solution presented is correct, thereby validating a block and allowing its inclusion on the blockchain.
- Byzantine Fault Tolerance (BFT) is the ability of a distributed computer network to correctly reach a sufficient **consensus** despite malicious nodes in the system failing or sending out incorrect information

Byzantine General Problem

- *Imagine that several divisions of the Byzantine army are camped outside an enemy city, each division commanded by its own general.*
- *The generals can communicate with one another only by messenger.*
- *After observing the enemy, they must decide upon a common plan of action.*
- *However, some of the generals may be traitors, trying to prevent the loyal generals from reaching an agreement.*
- *The generals must decide on when to attack the city, but they need a strong majority of their army to attack at the same time.*
- *The generals must have an algorithm to guarantee that (a) all loyal generals decide upon the same plan of action, and (b) a small number of traitors cannot cause the loyal generals to adopt a bad plan.*
- *The loyal generals will all do what the algorithm says they should, but the traitors may do anything they wish.*
- *The algorithm must guarantee condition (a) regardless of what the traitors do. The loyal generals should not only reach agreement, but should agree upon a reasonable plan.*

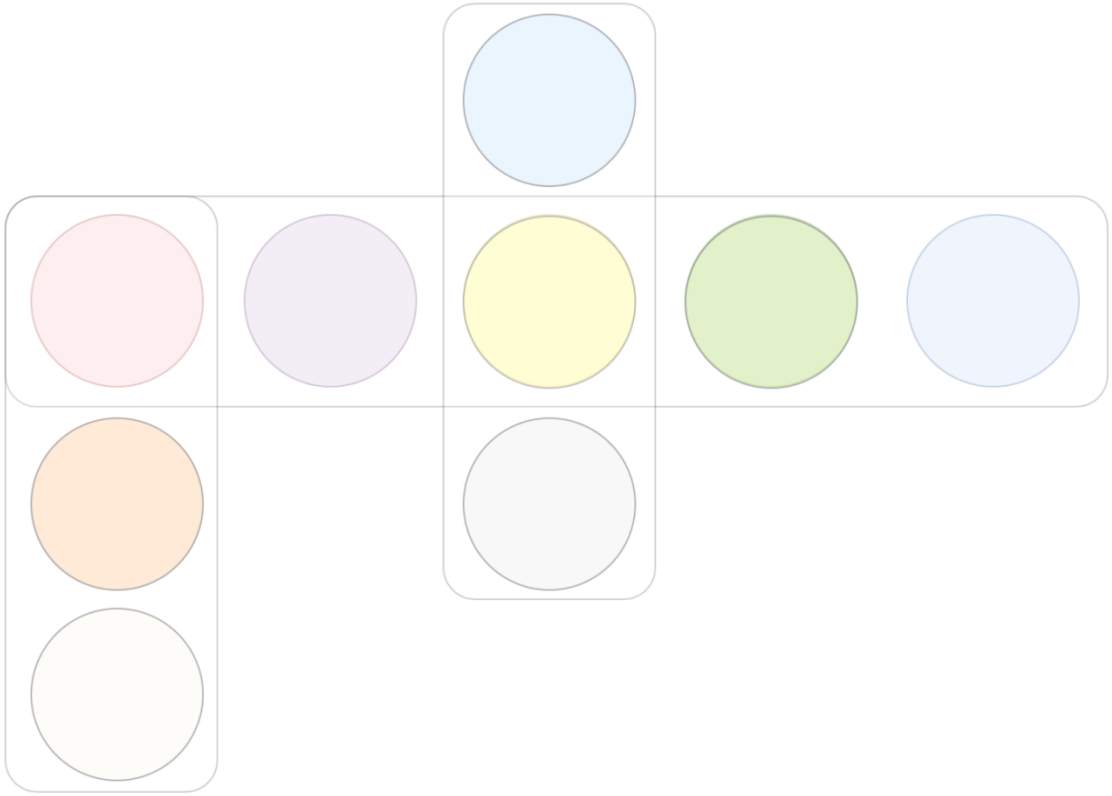
BFT

- Lamport proved that if we have 2m+1 correctly working processors, a consensus(agreement on same state) can be reached if atmost m processors are faulty.
- which means that strictly more than two-thirds of the total number of processors should be honest.
- pBFT (practical BFT) was designed to work efficiently in asynchronous systems

FBA //Federated Byzantine Agreement

- FBA is used for its high throughput, network scalability, and low transaction costs. It requires nodes to be known and verified ahead of time.
- No gatekeeper or central authority — individual nodes can decide whom they trust for information
- A quorum is the number of nodes required to reach agreement within a system.
- FBAs instead use ‘quorum slices’. A quorum slice is a subset of a quorum, which can convince another specific node to agree.
- E.g. ‘Node X’ can say, ‘for us to reach consensus we have to have buy in from the nodes of three of the five banks we have selected’

Quorum slices



Ledger implementations

- In the context of DLT systems, examples of **ledger implementations** include:
- Chain of blocks of records e.g. blockchain
- Blockchain : Proof of Work : Permissionless (BTC, ZEC, ETH)
- Blockchain : Proof of Stake : Permissionless (ETH Constantinople)
- Blockchain : Hybrid (PoW/pBFT) : Permissionless (ZIL)
- Distributed Ledger : pBFT : Permissioned (XRP)
- Distributed Ledger : FBA : Pseudo -permissioned (XLM)
- Distributed Ledger : Directed Graph : Permissioned (IOTA)
- Enterprise DLT Frameworks : <multitude> : Permissioned (HL/Corda)

- Hands on…