**BigchainDB: Hitchhiker’s Guide - Summary**

* Structure data as assets. An asset is any physical or digital object. Ex. Car, a Data set, an Intellectual property right, Lanetc.
* These assets can be registered on BigchainDB in 2 ways:
  1. By users in CREATE
  2. Transferred (or updated) to other users in TRANSFER transaction.
* BigchainDB focusses more on asset-centric rather than process-centric.
* CREATE TRANSACTION:
  1. Any user can create an asset on BigchainDB using the CREATE transaction.
  2. Components of the CREATE TRANSACTION:
     + Input: The input of the CREATE transaction contains information that about the asset being created. It also contains the digital signature of the creator to ensure non-reputability.
     + Asset: An asset will have immutable characteristics that cannot be changed anymore after registration. It is also associated with Metadata which can be changed at a later time in future transactions.
     + Transaction ID: Is a hash of all the information contained in this transaction (Input, Asset, etc..). This is used to crate references of this transaction in future transactions.
     + Output: The output of the CREATE transaction specifies that the caller of the create transaction is now the owner of the asset. It also lays down the conditions to transfer the ownership. The ownership can be transferred to someone else only if the owner provides a signature with their private key.
  3. Components of the TRANSFER TRANSACTION:
     + Input: The input of the TRANSFER transaction contains a reference to the output of the CREATE transaction and states.
     + Asset: An asset will have immutable characteristics that cannot be changed anymore after registration. It is also associated with Metadata which can be changed at a later time in future transactions.
     + Transaction ID: Is a hash of all the information contained in this transaction (Input, Asset, etc..). This is used to crate references of this transaction in future transactions.
     + Output: Specifies the new owner of the Asset. It also lays down the conditions to transfer the ownership. The ownership can be transferred to someone else only if the owner provides a signature with their private key.
* An asset can be represented in many ways like an Asset as a 1. Claim 2. Token 3. Versioned Document 3. State Machine 4. As a permission.
* Process of building any application in BigchainDB:
  1. Import driver -> include driver as module -> connect to any BigchainDB node (Test network)
  2. Creation of the key pair. Every user in BigchainDB is identified as a pair of private and public key.
  3. Digital registration of an asset on BigchainDB – Using CREATE transaction
  4. Transfer of an asset of BigchainDB - Using TRANSFER transaction
* Role based Access Control (RBAC)
  1. Role based access control is a way to restrict the system access to certain users.
  2. In BigchainDB, this function enables the creation of hierarchies of roles and permissions as assets.
  3. Users can be assigned roles to “act on behalf of” or “represent” other users or groups.
  4. The different users of the system can be assigned roles and permissions to perform different tasks in a system.
  5. In order to provide different permissions to different users, BigchainDB defines a simple asset hierarchy Refer: <https://blog.bigchaindb.com/role-based-access-control-for-bigchaindb-assets-b7cada491997>