**Problem Statement:**

Build a digital KYC mechanism which is easy, secure, maintains history and can be shared consistently across entities within and outside the Bank

* KYC is a necessity both from business as well as regulatory imperatives. All organizations onboarding new customers need to complete KYC norms. Challenge is to leverage technology to complete KYC with minimum process steps and sharing of KYC details as-a-service among entities on commercial terms. **This should be developed using Blockchain**

**Solution:**

**Terminology:**

* Valid Wallet Address: An address to or from which a transaction can be committed on the blockchain
* Private Key: A unique identifier known only to the customer of bank
* Digital Signature: A mechanism to verify a public and private key combination.
* Smart Contract:A computer program code that is capable of facilitating, executing, and enforcing the negotiation or performance of an agreement (i.e. contract) using blockchain technology. The entire process is automated can act as a complement, or substitute, for legal contracts, where the terms of the smart contract are recorded in a computer language as a set of instructions. It is a set of If-Then rules that trigger on various events and perform necessary actions.
* Ethereum: A public, open source blockchain network that can be custom made to suit the business needs
* Customer: A node in the blockchain that can perform transactions
* KYC or Know your Customer is the policy to identify and keep track of all the customers who commit transactions on the BoB blockchain.
* Externally owned Account: Account held by an individual customer, which has a balance
* Contract Account: Account that has both balance and contract storage. The contract code execution is triggered by transactions or messages (calls) received from other contracts.
* Header of a candidate block: the first transaction at the beginning of a blockchain
* Nonce: short for Number once, is the number combination that is appended at the end of the message, so that hash of the message will match with the hash of the header of a candidate block
* Transaction: Any exchange of money or property. A transaction contains the following:
  + - the recipient of the message
    - a signature identifying the sender and proving their intention to send the message via the blockchain to the recipient
    - The amount of wei to transfer from the sender to the recipient
    - an optional data field, which can contain the message sent to a contract
    - a value, representing the maximum number of computational steps the transaction execution is allowed to take
    - a  value representing the fee the sender is willing to pay for gas. One unit of gas corresponds to the execution of one atomic instruction, i.e., a computational step
* Consensus on a blockchain: The mechanism by which the third party calculate the nonce of the message that matches the hash of the candidate block header

Using the Smart Contract system in Ethereum framework, we will build an intelligent Digital Signatures based system that will trigger actions for various events- such creating a savings bank account, loan processing or intra bank money transfer, mutual fund products

Verification of KYC documents takes extreme levels of efforts and time.

**For whom the KYC is required:**

* A person or entity that maintains an account and/or has a business relationship with the bank
* One on whose behalf the account is maintained (i.e. the beneficial owner)
* Beneficiaries of transactions conducted by professional intermediaries such as stockbrokers, Chartered Accountants, or solicitors, as permitted under the law
* Any person or entity connected with a financial transaction which can pose significant reputational or other risks to the bank, for example, a wire transfer or issue of a high-value demand draft as a single transaction.

**When is KYC required:**

KYC is required while,

* Opening a bank account
* Availing Locker facility
* Applying for credit card
* Applying for loans
* Existing documents are not enough
* Changing signatory, nominee

Currently, KYC is a legal requirement in many sectors, apart from banks. Whether it is mutual Funds, insurance, broking, or commodity trading KYC has been made compulsory in order to verify the identity of the clients.

**How Smart Contracts will help with KYC:**

The Customer transactions between accounts will be tracked and stored on a blockchain. Let us name it as the “BoB” Blockchain. All transactions on the BoB blockchain will only be known to the peers of this blockchain and no one outside.

Let us consider the case of customer onboarding for retail savings bank account. The first steps, as per the KYC policy of RBI guidelines, would be to collect the documents from the customer for the various requirements presented above.

During the onboarding of the customer onto the blockchain, apart from their wallet address (in this bank account number or the loan disbursal account number) and a private key to access their account and commit transactions using the wallet.

Once the customers have these wallet addresses, they can commit transactions to other wallets on the BoB blockchain (other customers, merchants and many others who have valid wallet address).

While committing transactions, an inherent standard on the blockchain is to use the concept of Digital Signatures. No transaction will be possible without the use of a public key (wallet address) and the private key (one with the customer).

As details of the transaction, we shall store the along with the timestamp and the amount involved, the hash of the kyc details of the customer/institution committing the transaction.

**Wider Architecture of the system:**

* Blockchain on Ethereum, with smart contracts
* Front –end user interface
* DB that has mapping between customer and their public address key
* Admin interface for the bank

**How to avoid KYC duplicity on the blockchain:**

Unique KYC for a customer should be one and the same. Multiple KYC keys cannot exist.

Using the consensus mechanism on the blockchain we will create the method to validate a transaction between two customers. When a majority of consensus is achieved on the blockchain

Reference:

https://github.com/SmartIdentity/smartId-contracts

**Problems to address**:

* Same wallet address can be used by any adversary who steals it from the victim
* Mistyping the addresses of wallets