

'eKYC Infinite'

*A streamlined blockchain
solution for financial
institutions' eKYC process*

ARCHITECTURE DESIGN & GOVERNANCE

Decentralized Applications I – BCDV 1012

George Brown College

Professor Dave Mckay & Professor Tarun Sharma

July 2020

CHERUKKATIL NASEER – 101280592

HENRY ERIKO MWENGE – 101286818

LORRAN SUTTER – 101276593

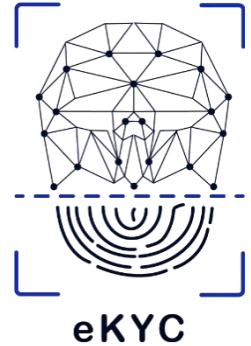
RAYMOND LAWAL – 101229011

WANJA MASCARENHAS – 101280022

ZAKARIYA JASAT – 101092428



eKYC Infinite



AN ELECTRONIC 'KNOW YOUR CLIENT' START-UP.

We are leveraging *Distributed Ledger Technology* by integrating *Blockchain Infrastructure* and *Smart Contracts* from *Hyperledger Fabric* in our business model.

We strive to streamline how the KYC procedure is carried out in the financial industry, facilitating a more efficient verification process for clients and financial institutions.

Our business will accelerate the financial institutions' process of verifying client information, enabling a secure and effective standard for selling securities.

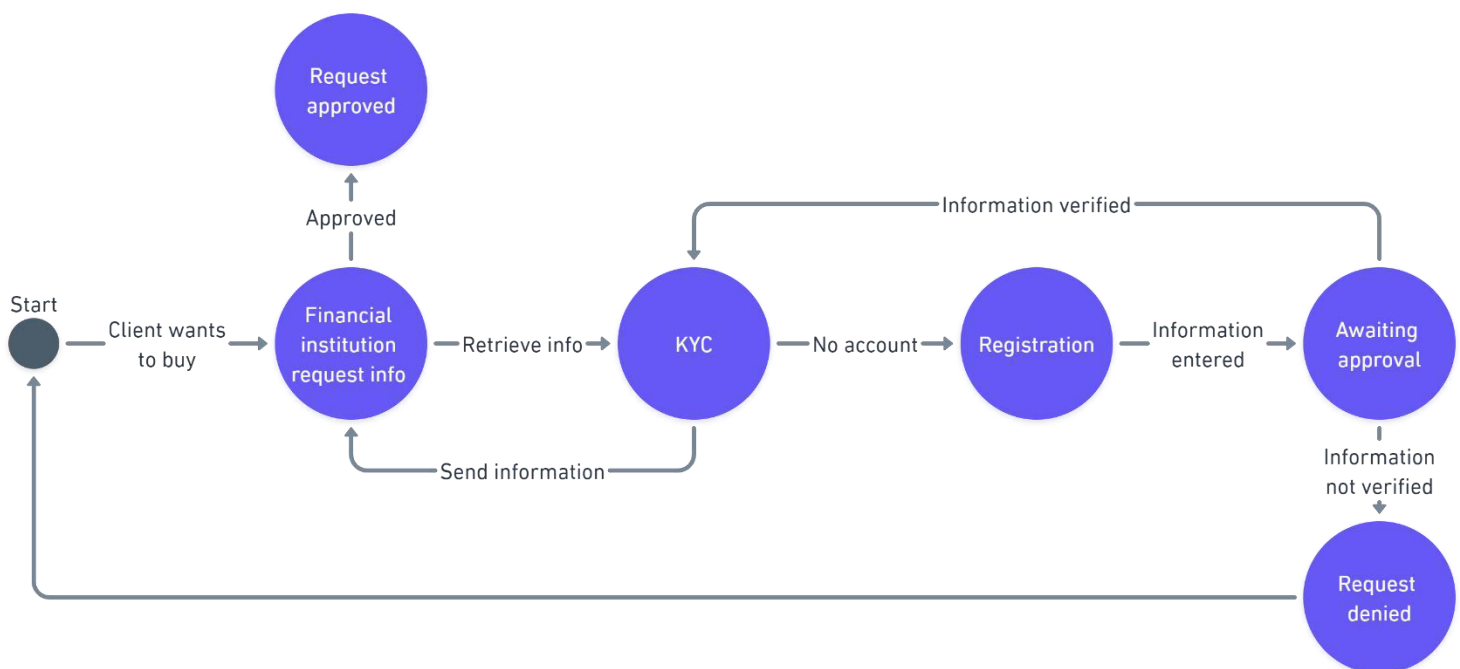
Table of Contents

<u>EXECUTIVE SUMMARY</u>	<u>4</u>
<u>THE PROBLEM.....</u>	<u>5</u>
<u>OUR SOLUTION</u>	<u>7</u>
<u>GOVERNANCE</u>	<u>13</u>
<u>APPENDIX</u>	<u>15</u>
<u>REFERENCES.....</u>	<u>16</u>
<u>TEAM BIO</u>	<u>17</u>

Executive Summary

We are using 'Blockchain Technology' to help combat the issues of very long wait times, unverified client identities, and an unreasonably repetitive process for each institution when selling securities. We have leveraged *Hyperledger Fabric* to create a consortium solution for various financial institutions, resulting in a modernized interaction structure. We have created an interface for clients who want to buy securities from a financial institution to go through a *one-time verification* process with the institution, thereafter, not having to go the verification process again if they decide to buy securities from a different institution. When a client has been verified by a financial institution, other financial institutions will be able to retrieve verified client information instantaneously. Our solution also considers authorization implications by creating a process for the client to give only certain institutions access to certain information, protecting them from having institutions accessing their information without approval.

Figure 1. eKYC State Diagram



The Problem

The global financial services sector continues to be rocked by a series of regulatory fines and money-laundering scandals, totaling more than US\$36B in fines over the past decade.

The C-suite in most financial institutions are under considerable pressure to deliver safer but faster customer onboarding journeys. The key to achieving this comes down to their ability to manage client lifecycle events and milestones via re-usable, verified, and up-to-date customer information.

Real KYC Challenge

The core of what takes onboarding so long is the process of collecting, validating, and processing customer data to achieve the right compliance decision. In today's 'digital-first' world, business' subject to know your customer (KYC) requirements must find a way to create better onboarding and authentication procedures that achieve compliance and combat fraud, without alienating good customers.

For about 1 in 5 (21%) financial institutions, onboarding timeframes have increased by 4 – 8 weeks. What is worse is the fact that they expect onboarding times to rise even further. Instead of getting to the root of problem, many firms are still solving the compliance challenge with brute force or added headcount.

It is crucial that financial institutions can access the right data at the right time. Re-usability and transparency are essential for this. We believe, if a client onboarding team can easily access the customer information needed to complete KYC instantaneously, the result will be a much faster onboarding process and better client experience.

Digital KYC and eKYC utilities to the rescue

To solve these challenges and become more digital, streamlined, and cost-effective, the industry is looking at two key areas to transform and revolutionize KYC compliance. The First one is internal Digital KYC processes that create an ecosystem of data and system providers to enable straight through processing for compliance and onboarding. The other area is the second coming of electronic Know Your Customer (eKYC) utilities. The first iteration of KYC utilities (2012-2015) proved that there is a need for shared, validated customer information, however, the technology and distribution models underpinning these KYC utilities failed to deliver and create any real impact.

There are three key drivers behind financial institutions' interest in KYC technology as a utility: rising customer expectations, the emergence of new technology, and the drive toward compliance and operational efficiencies.

The industry is actively looking at improving the client experience, especially reducing onboarding times and delivering value-added services to their client. eKYC utilities can help this by providing digital and fully packaged access to up to date KYC data, thus speeding up and simplifying the client onboarding process. Businesses across the world are moving towards exploring new technologies such as Blockchain and ID&V-type technologies.

Our Solution

Introduction / Motivation

The Electronic Know Your Client (eKYC) is a layer for digital identity that allows a web application on which the seller can look up if the buyer meets clear requirements before the transaction, as well as register the buyer through (eKYC). The sellers need reliable and independent source data or information. Also, buyers can have a relationship without an intervening authority – self-sovereign. Nowadays, the lack of security with a person's identity exists in each application that users store their information and give control to central authorities to manage them.

Blockchain is applicable for digital identity because it is immutable, legitimate, and validated by multiple participants in the networking. This architecture prevents Anti-Money Laundering (AML) and protects sellers of the securities aware of the financial status of the buyer and prevents fraudulent behaviour, as well as client information.

How it Works

- Each organization in the ledger represents a financial institution (FI) and has an enrolled user in order to access the system.
- Only FIs are able to perform submit or evaluate transactions and also register new clients that want to keep their information in the system.
- Once a new client is registered, the FI who registered that client is automatically able to access the client data and that client cannot remove access to this institution.

- Clients are able to approve and remove approval from other registered institutions to access their data.
- Considering that only FIs can perform transactions, in order to approve or remove approval clients must have organization number and ledger user information. For such, these information are encrypted and stored in the database. Once the request is made, the backend decrypts that information to perform the transaction.
- A composite key is employed to map the approvals. We have clientId~fild and fild~clientId composite keys to map this relationship.

Web App Interfaces and Roles

The application uses a permissioned system based on roles. Furthermore, uses Hyperledger Fabric to control transactions, approve transactions and establish new rules.

Functionality

The blockchain would enable:

- Storing KYC information on a user
- Putting out a request for the KYC of a user
- Responding to the request
- Response using only necessary information
- If there are no responses, then get the data from the user

The application has 3 different roles:

- System – Our ‘eKYC Infinite’ solution.

- Seller – Financial Institutions selling securities
- Buyer – Clients who want to buy securities

State Data

- States: [“Not-registered”, “Registered”, “Approved”, “Rejected”]
- Name
- Date of birth
- Address
- Identification number (Drivers License, Passport, Photo Card, ...)

Transitions

- RequestInfo by uniqueID(Name, Date of birth, Identification number)
- Registration(Name, Date of birth, Address, Identification number)
- Incomplete(Name)
- RequestDenied(Name)
- Approved(Name)

Functions

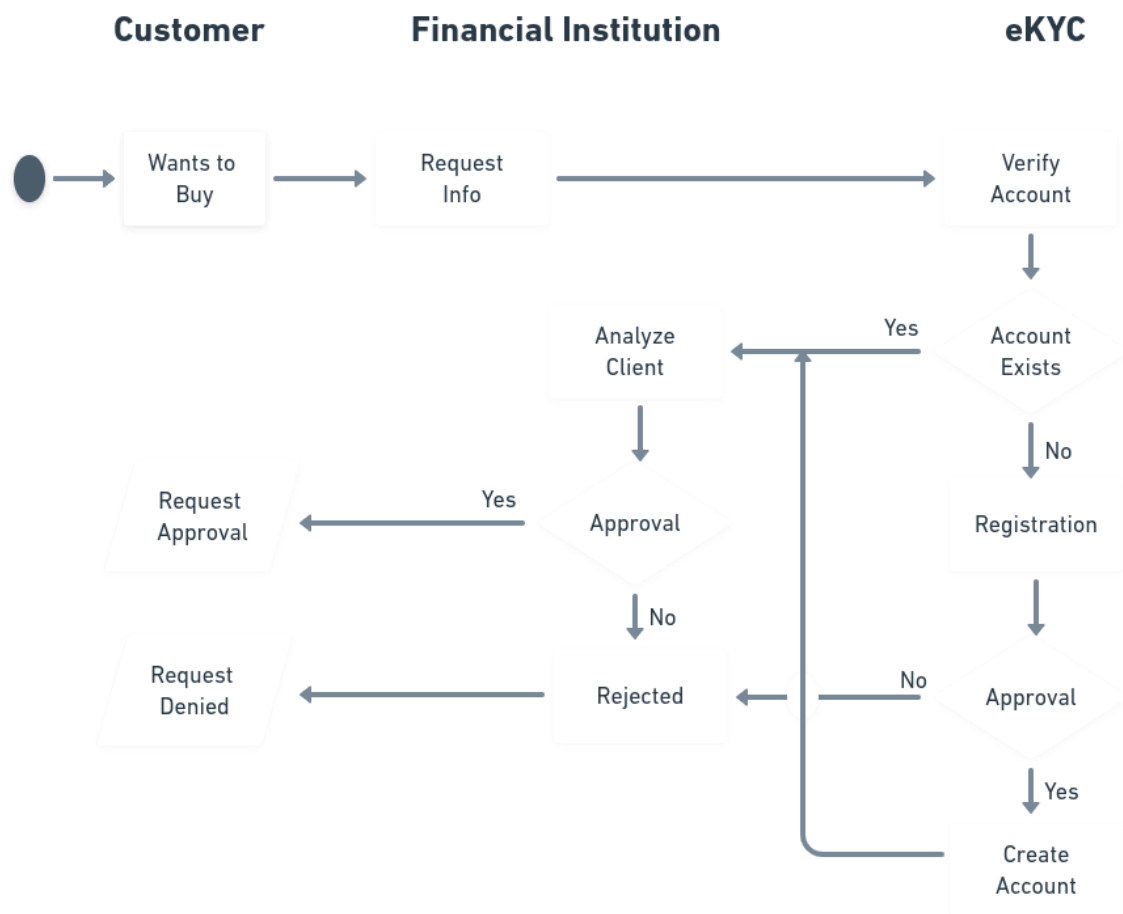
- CheckStatus(Name)

Roles

- RequestInfo(Name, Date of birth) – Seller
- Registration(Identification number, Name, Date of birth, Address) – System

- Incomplete(Name) – System
- RequestDenied(Name) – System
- Approved(Name, Date of birth, , Address) – System
- CheckStatus(Name) – Buyer/ Seller

Figure 2. eKYC Flow Chart



Database data

- User schema
 - login
 - password
 - userType
 - 'client' or 'fi'
 - ledgerId
 - ID generated by the ledger
 - whoRegistered
 - Information of who registered the client
 - Only clients have this field
 - orgNum and ledgerUser fields are encrypted and stored
 - orgCredentials
 - Credentials of the organization
 - Only financial institutions have this field
 - orgNum and ledgerUser fields are encrypted and stored

Endpoints

- /client/create
 - method: POST
 - params: login, password, name, date of birth, address, id number
 - description: register new client
 - chaincode: calls 'createClient' function
- /client/login
 - method: POST
 - params: login, password, user type
 - description: client login
- /client/getClientData
 - method: GET
 - params: ord number, ledger user, ledge ID

- description: return the data from the client
 - chaincode: calls 'getClientData' function
- /client/approve
 - method: POST
 - params: org number, ledger user, ledger ID, financial institution ID
 - description: approve a financial institution to read client data
 - chaincode: calls 'approve' function
- /client/remove
 - method: POST
 - params: org number, ledger user, ledger ID, financial institution ID
 - description: remove a financial institution to read client data
 - chaincode: calls 'remove' function
- /client/getApprovedFis
 - method: GET
 - params: org number, ledger user, ledger ID
 - description: return a list of approved financial institutions by the client
 - chaincode: calls 'getRelationByClient'
- /fi/create
 - method: POST
 - prams: login, password, name, id number
 - description: register new financial institution
 - chaincode: calls 'createFinancialInstitution'
- /fi/createClient
 - method: POST
 - params: login, password, name, date of birth, address id number, org number, ledger user
 - description: register new client
 - chaincode: call 'createClient'
- /fi/login

- method: POST
 - params: login, password, user type
 - description: financial institution login
- /fi/getFiData
 - method: GET
 - params: org number, ledger user
 - description: return the data from the financial institution
 - chaincode: calls 'getFinancialInstitutionData' function
- /fi/getClientData
 - method: GET
 - params: org number, ledger user, client ID, fields wanted
 - description: return the data from the client
 - chaincode: calls 'getClientData' function
- /fi/getApprovedClients
 - method: GET
 - params: org number, ledger user
 - description: return a list of clients that approved this financial institution
 - chaincode: calls 'getRelationByFi'

Ledger data

- Client
 - name: Name of the client
 - dateOfBirth: Date of birth of the client
 - address: Address of the client
 - idNumber: Some identification number of the client
 - whoRegistered: { orgNum: number of the organization that registered the client, ledgerUser: name of the ledger user that registered the client }

- Financial Institution
 - name: Name of the financial institution
 - idNumber: Some identification number of the financial institution
 - orgCredentials: { orgNum: number of the organization, ledgerUser: name of the ledger user of that organization }

- Chaincode functions
 - initLedger(ctx)
 - Initiate ledger storing financial institution and client data

 - getCallerId(ctx)
 - Internal function - return the ledger user that called the method

 - isWhoRegistered(ctx, clientId)
 - Internal function - return the ledger user the registered that client

 - createClient(ctx, clientData)
 - Create a new client. Who registers the client is who called the method

 - getClientData(ctx, clientId, fields)
 - Return client data. A list of fields wanted is passed as a parameter

 - getFinancialInstitutionData(ctx)
 - Return financial institution data

 - approve(ctx, clientId, fild)
 - Approve a financial institution to access client data

 - remove(ctx, clientId, fild)

- Remove financial institution access data approval
- getRelationsArray(ctx, relationResultsIterator)
 - Internal function - iterate a composite key iterator
- getRelationByClient(ctx, clientId)
 - Return a list of approved financial institution by that client
- getRelationByFi(ctx)
 - Return a list of clients who approved the caller FI
- queryAllData(ctx)
 - Return a list of all data stored in the ledger
- Approvals
 - Composite key clientId~fild
 - Maps all FIs approved by the client
 - Composite key fild~clientId
 - Maps all clients that approved that FI

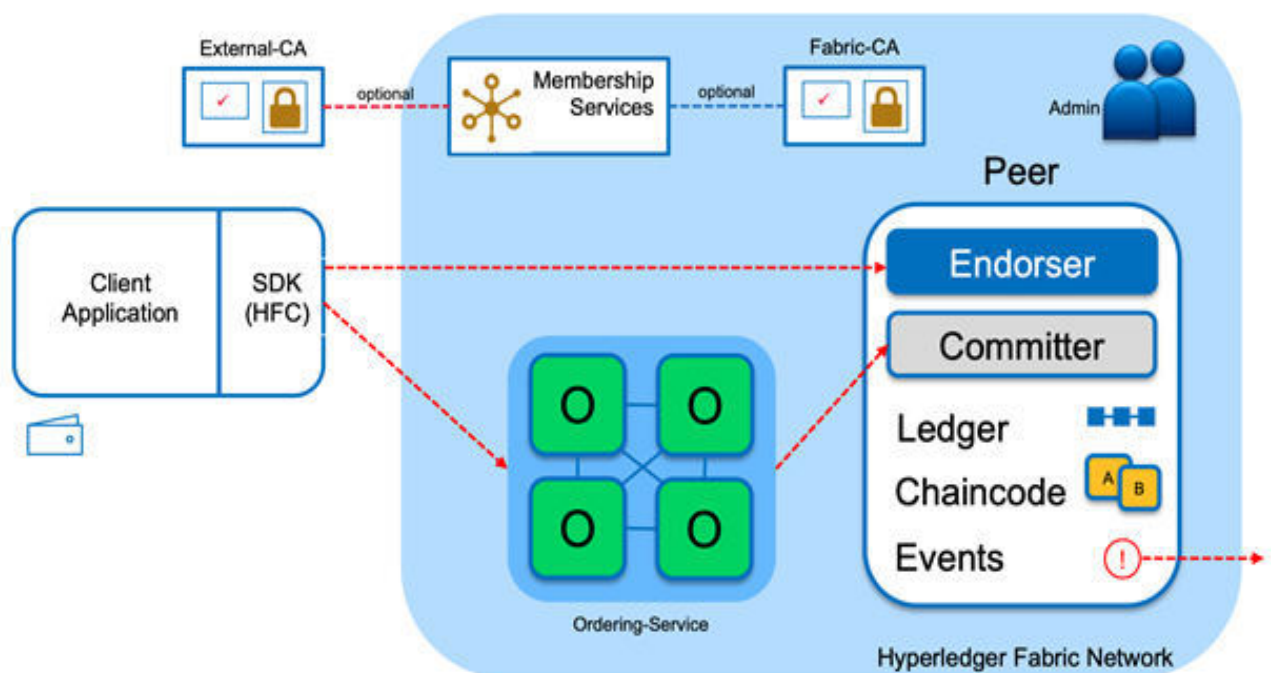
Governance

Our 'eKYC Infinite' platform helps to store KYC data in a Distributed Ledger, where data can be accessed by financial institutions with authorization. It helps to improve data governance and as result the organization/consortium manages KYC compliance swiftly and efficiently.

Membership Service Providers(MSP)

'eKYC Infinite' will act as the MSP and the Blockchain network is governed by 'eKYC Infinite' for all the organizations using eKYC service. MSP will perform onboarding process and will generate X.509 certificates to encrypt the digital identities of all the blockchain members in the network.

Figure 3. Hyperledger Consensus Framework



Endorsing Peers

Financial Institutions will act as Endorser peers. Endorsing peers verify signature and execute transactions. Financial institutions will scan through the Distributed Ledger to check if the client data already exists, if not, this financial institution will update client KYC information into the ledger, and it will be the responsibility of the endorsing peers to ensure submitted information satisfies the policy of the blockchain network.

Ordering service nodes

FINTRAC Canada will act as the Ordering service nodes. Ordering services assure deterministic features the consensus algorithms, which means any block validated by the peer is guaranteed to be final and correct. Orderers also maintain the list of organizations that are allowed to create channels.

Channels

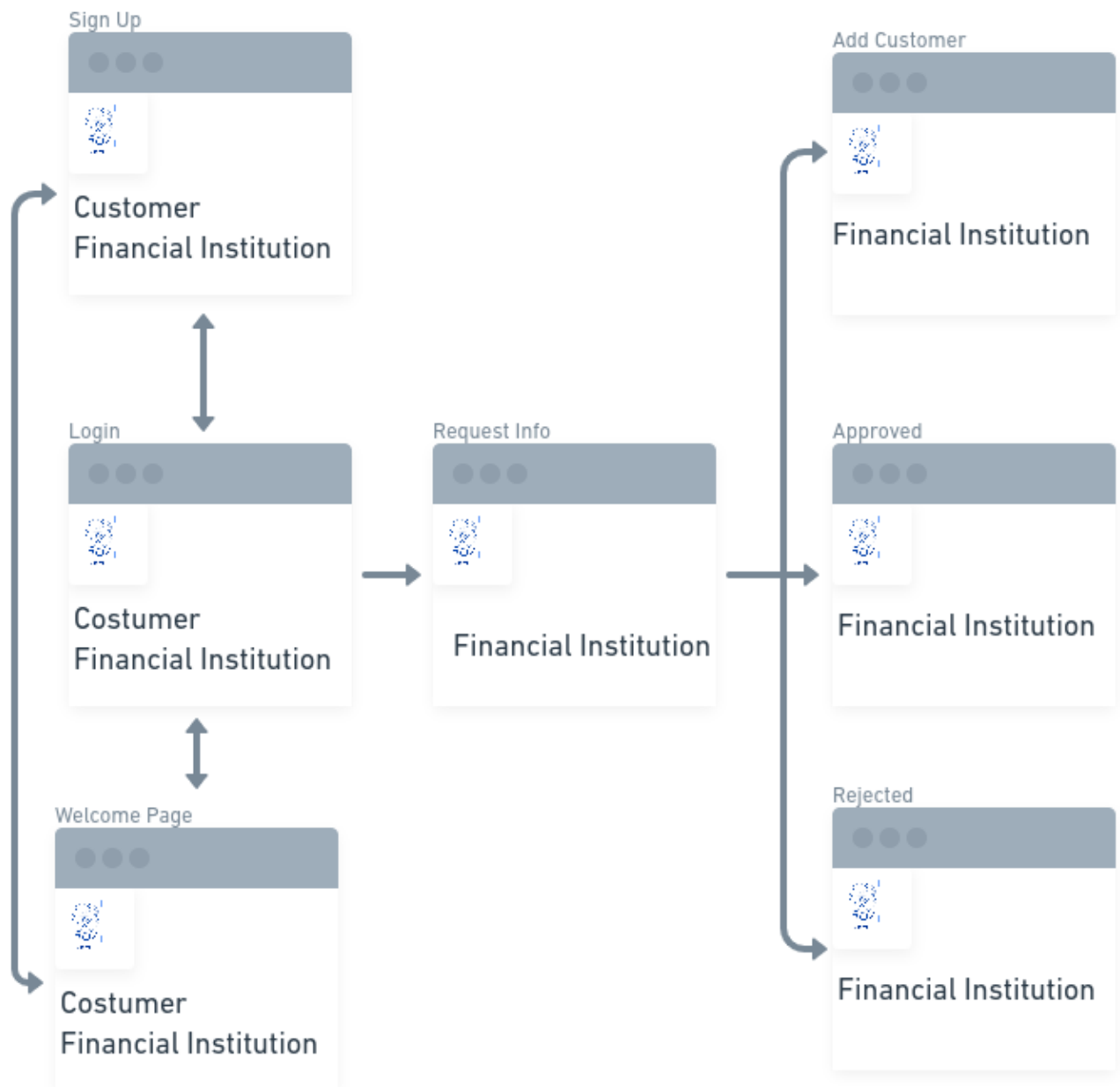
'eKYC infinite' only has one channel which stores client data in key/value pairs. Only approved members can access the ledger.

Client Application

'eKYC infinite' stores clients' private key in a custodial wallet. It is the responsibility of eKYC infinite to ensure security of the client's private keys during the duration of their membership.

Appendix

Figure 4. eKYC Web Interface



References

- Heckel, M., & Collette, B. How should we tackle new KYC challenges? Retrieved 9 July 2020, from https://www2.deloitte.com/content/dam/Deloitte/fr/Documents/risk/deloitte_how-should-we-tackle-the-new-kyc-challenges.pdf
- Hyperledger Fabric Model — hyperledger-fabricdocs master documentation. Hyperledger-fabric.readthedocs.io. (2017). Retrieved 12 July 2020, from https://hyperledger-fabric.readthedocs.io/en/v1.0.5/fabric_model.html
- Membership Service Providers (MSP) — hyperledger-fabricdocs master documentation. Hyperledger-fabric.readthedocs.io. (2017). Retrieved 12 July 2020, from <https://hyperledger-fabric.readthedocs.io/en/v1.0.5/msp.html>
- Szymański, B. (2019). The untapped potential of eKYC digital account opening journeys | LiveBank | Virtual Branch Banking. Retrieved 9 July 2020, from <https://livebank24.com/the-untapped-potential-of-ekyc-digital-account-opening-journeys/>
- The ABCs of eKYC - FinTech Futures. (2020). Retrieved 9 July 2020, from <https://www.fintechfutures.com/2020/04/the-abcs-of-ekyc/>
- Twomey, N. (2020). Can e-KYC Utilities Solve the Compliance Challenge? Retrieved 9 July 2020, from <https://www.finextra.com/blogposting/18402/can-e-kyc-utilities-solve-the-compliance-challenge>

Team Bio



Cherukkatil Naseer

Nas is a Blockchain Development student at George Brown College and Director of NLM UA-Systems. He believes that rapid development in AI, IoT and Blockchain would create better efficiency and security in most of the industries and would open more jobs/business opportunities in new sectors. Nas has been well-trained on avionics products of Boeing and Airbus commercial Aircrafts. He holds UA

(Unmanned Aircraft) Pilots certificate approved by CAAS. He served Red-cross, Singapore as a volunteer for two years where he trained on First Aid and CPR. He spends his free time flying, building, and modifying drones.

Henry Eriko Mwenge

Eriko is the co-founder of The Briefcase Trader and is responsible for producing the company's equity research reports. He is the chief editor of The Briefcase Trader Soft Commodity and Livestock Report. Eriko is passionate about commodities and developing a fully integrated Structured Commodity Trade Finance network specific to emerging markets. He enjoys playing rugby, bird watching, and beekeeping.



Lorrان Sutter

Lorrان is a Computational Engineer that studied a couple of different fields, such as computational modeling, numerical methods, computer graphics and machine learning. He has work experience in web development mainly employing C# in developing web crawlers and automated tests for large legal management systems. Engineer by background and

programmer by heart, in love with Python and JavaScript. Always looking for innovative and disruptive technologies, which led him to start studying Blockchain Development in George Brown College. He believes that humanity must rely on technology to build a better future.

Raymond Lawal

My name is Lawal Raymond, I am currently studying blockchain development at George Brown College in Canada. I write smart contracts with the code language Solidity and I also Trade Crypto Currencies. Whenever I am not working on an application or project, I make Music. My favourite quote is "Keep Believing".



Wanja Mascarenhas

Wanja Mascarenhas is an IT professional and educator. As a developer, Wanja worked in the banking industry. As an educator, she taught computational algorithms, programming, data structures, and operational research. She spends her free time volunteering. She loves talking with friends and sharing experiences. Wanja is currently studying Blockchain Development at George Brown College.

Zakariya Jasat

Zak has lived and worked in nations across the globe, making him an empowered multicultural business administrator. He is exceptional at adapting to change, managing human resources, and viewing situations through several lenses. He has helped set up businesses from the ground up and lead multiple keystone projects. He truly appreciates the importance of owning strategic objectives by improving organizational performance through boosting employee engagement and creating a healthy organizational culture through fostering diversity and inclusion. Zak is a calculated risk taker, motivated self starter and is always committed to leaving a legacy that inspires positive change everywhere he goes. He is the strong open-hearted wind that ignites powerful contribution.

