eKYC INTEROPERABILITY STANDARD

Prepared By :
TUM Serey Vathana

version: 1.0

Techo Startup Center

eKYC Interoperability Standard Technical Documentation

REVISION: 1.1.0 REVISION DATE: 28-Feb-2024

PREPARED BY: TUM SEREY VATHANA

Table of Contents

1. Introduction

The Open KYC verification API offers valuable features for eKYC providers, allowing them to verify client information, conduct face similarity checks, video liveness verification, and OCR on identification documents. However, a limitation is that eKYC information remains confined within providers' domains, hindering data sharing and collaboration. To address this, the eKYC Interoperability Standard (eKYCIS) enables secure data exchange among stakeholders while preserving privacy. It establishes protocols and security measures for seamless and secure sharing of eKYC information, enhancing customer verification processes.

This document serves as a comprehensive documentation outlining the standard implementation and rules to be followed by eKYC providers and requesters. Its purpose is to ensure complete interoperability among all members involved in the eKYC ecosystem. By adhering to these standards, organizations can minimize the effort required for integration while maintaining cross-compatibility among systems used by various organizations.

To gain a clearer understanding of the challenges faced by industries, government entities, and end-users in the conventional eKYC ecosystem, it is recommended to use this document in conjunction with the eKYCIS proposal. The proposal document provides an elaboration of the problems encountered in the current eKYC landscape and explains how the proposed standard aims to address these issues. It offers insights into the difficulties faced by key stakeholders and highlights how the implementation of the new standard can benefit them. By referring to both the document and the standard proposal, stakeholders can obtain a comprehensive understanding of the existing challenges and the detailing solutions provided by the interoperability standard.

2. Conventional eKYC vs eKYC Interoperability Standard

2.1. Conventional eKYC

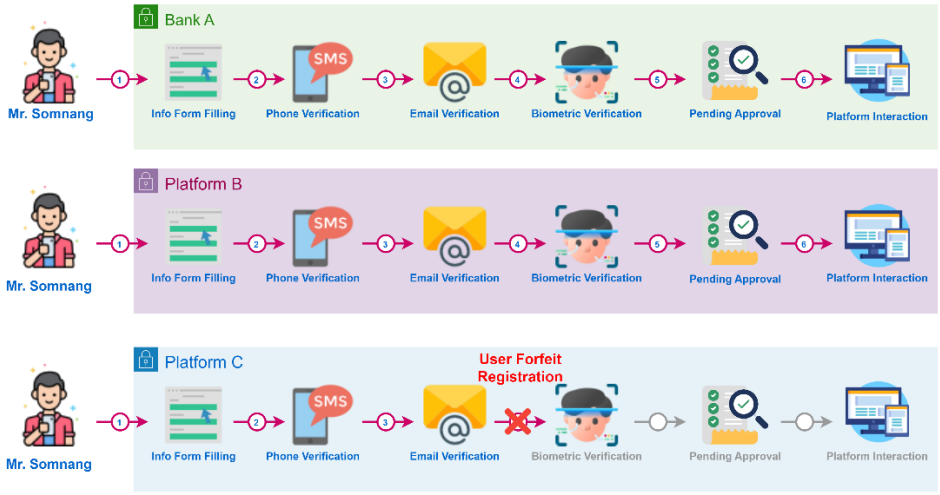

Figure 1 User's experience using conventional eKYC

Conventional eKYC processes pose multiple hurdles for individuals like Mr. Somnang during digital platform registration. He must input personal information, undergo various verifications, provide facial data, and await approval from each platform. This repetitive process is burdensome and discourages participation in digital services. Additionally, eKYC providers incur substantial costs for verifying user input, including information, phone, email, document, and biometric verification. The complex registration procedure frustrates users and often leads to registration abandonment, resulting in the loss of potential users.

2.2. eKYC Interoperability Standard

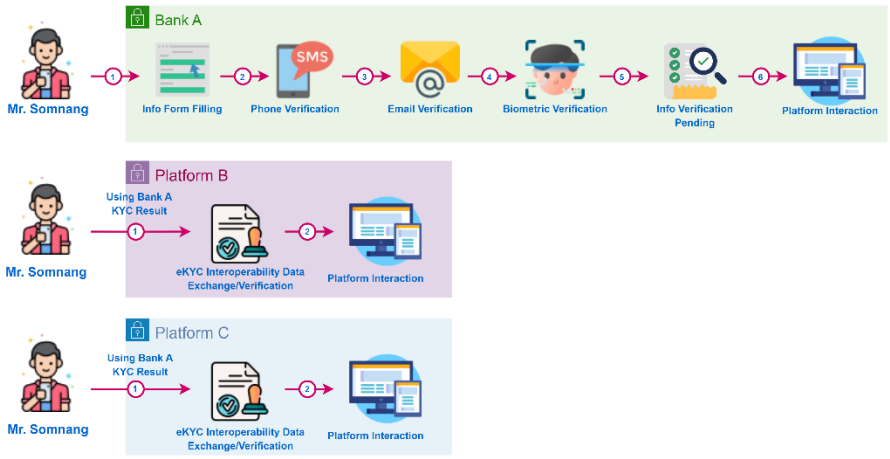

Figure 2 User's experience using eKYCIS

The eKYCIS significantly improves user experience and reduces costs for customer eKYC processes. With this standard, Mr. Somnang only needs to complete a full eKYC process with one platform. Then, when he wants to register or authenticate himself with another platform, he can effortlessly reuse the eKYC result from the previous platform, eliminating manual input of information, document submission, or biometric data. This standard enhances user convenience by enabling the reuse of eKYC results across multiple platforms, promoting collaboration among eKYC providers and platforms. It maximizes the benefits of this standardized approach, extending the one-only principle beyond e-government.

3. Prototyping

The standard includes an early prototype to showcase the technology's functionality for end users. It also provides details on the system's interactions to achieve efficient and secure eKYC data exchange.

3.1. User Journey (UI/UX Prototyping)

The prototype covers a scenario involving Mr. Mean Somnang, who has registered an account and completed KYC verification with Mango Bank. Mr. Somnang intends to register with Apple Bank and seeks to streamline the process by utilizing the verified KYC information

obtained from Mango Bank. To achieve this, he employs the eKYCIS, allowing him to seamlessly complete his registration with Apple Bank.

3.1.1. Mobile Prototyping

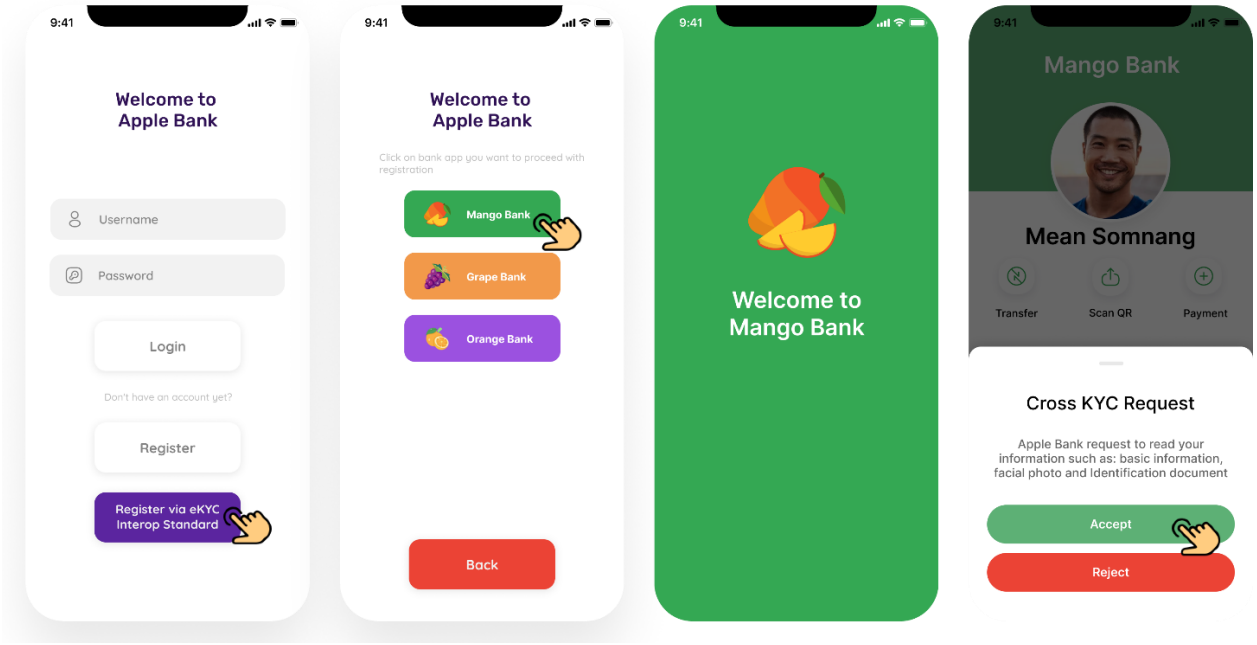

Figure 3 Mobile Prototyping (Part 1)

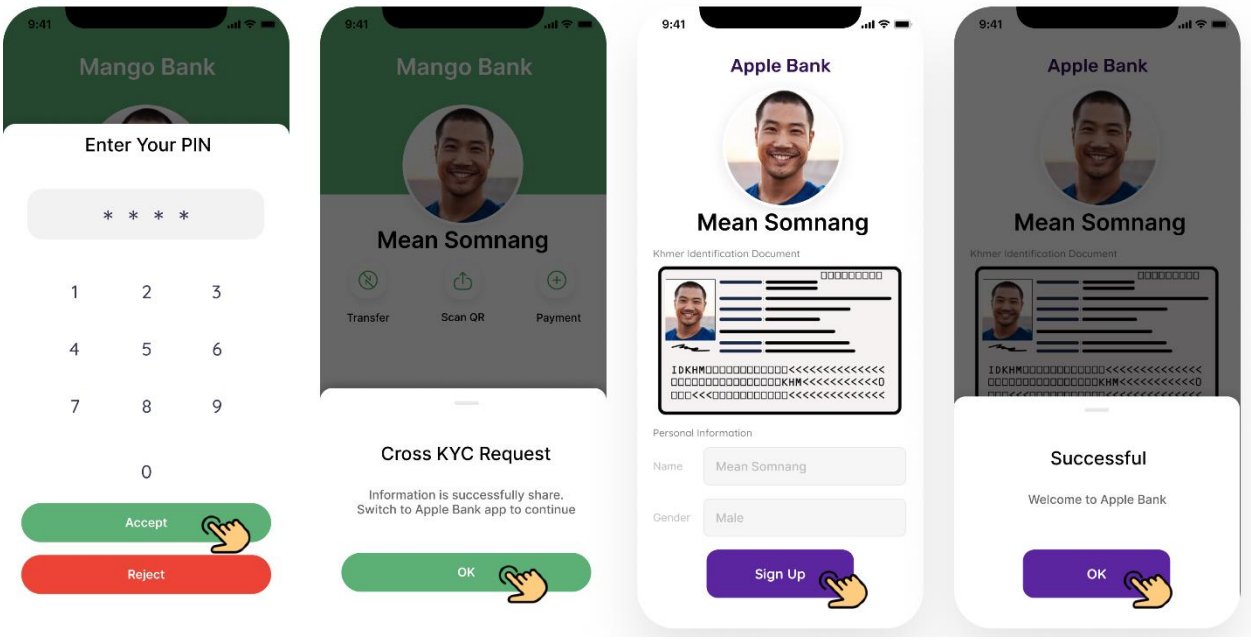

Figure 4 Mobile Prototyping (Part 2)

3.1.2. Web Prototyping

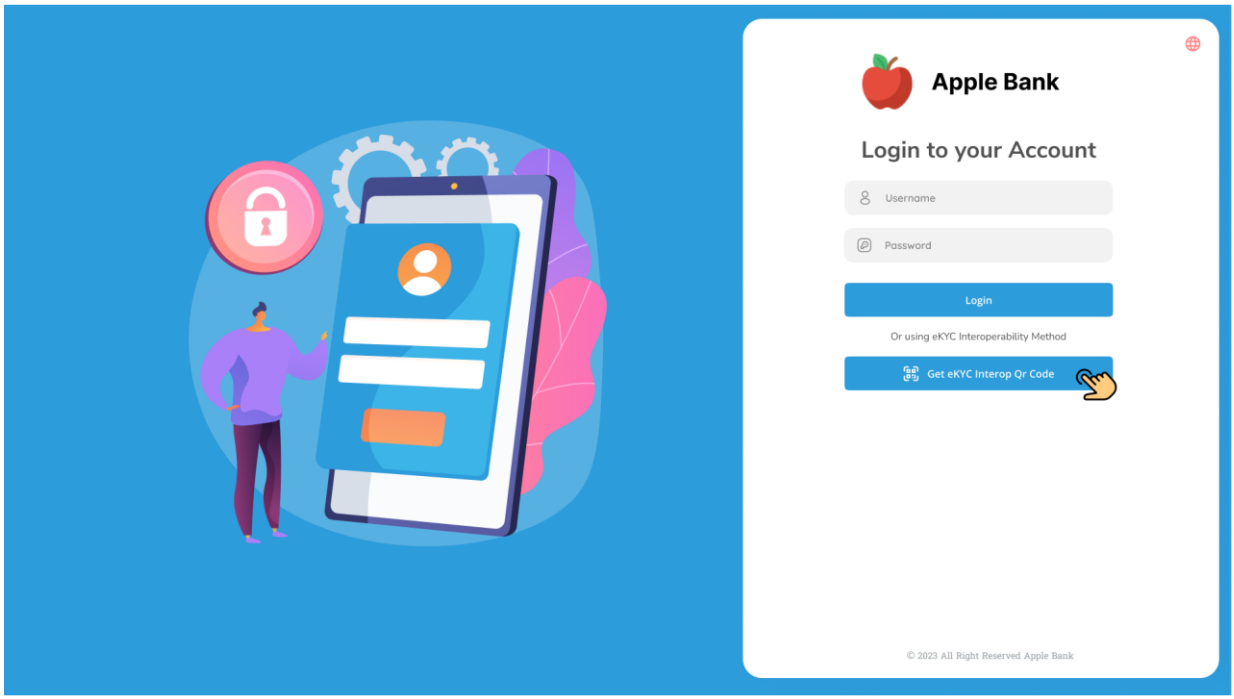

Figure 5 Web Protoype eKYC Interop (part 1)

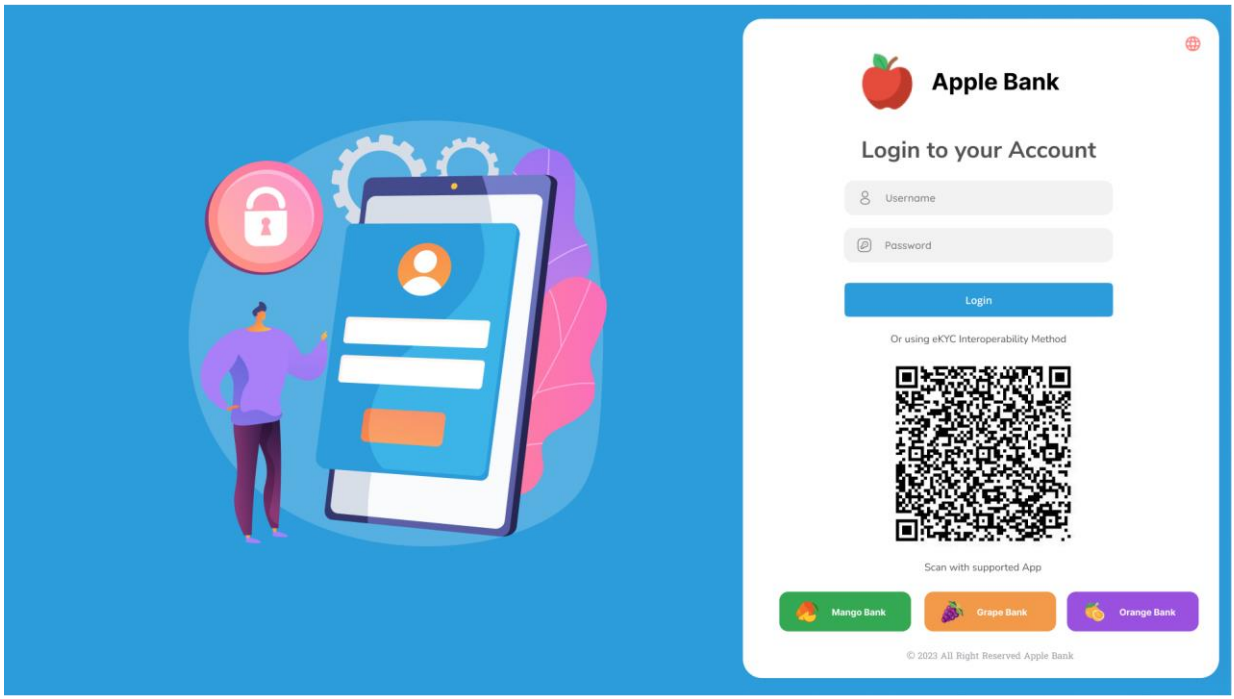

Figure 6 Web Protoype eKYC Interop (part 2)

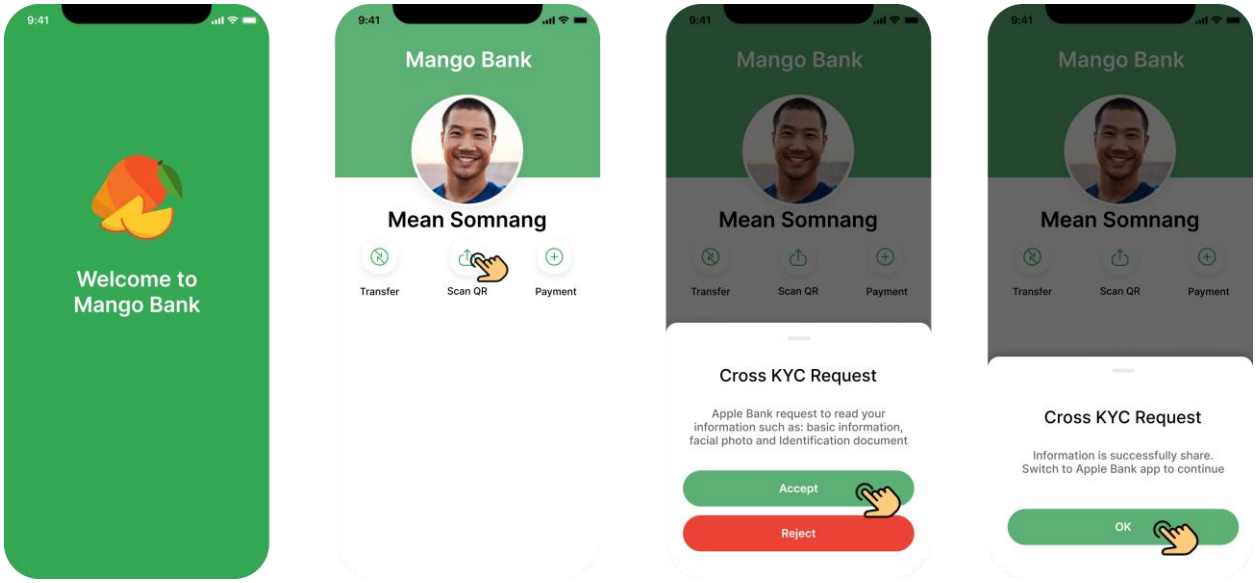

Figure 7 Web Prototype eKYC Qr Scan by Mobile App (part 3)

4. eKYC Interoperability Standard

The eKYCIS aims to establish a standardized process for eKYC exchanges, covering information verification, secure data exchanges, data formatting, and more. Its goal is to enable effective collaboration among eKYC providers, ensuring secure and reliable exchanges while creating additional revenue opportunities. Implementing this standard allows providers to offer Cambodian citizens a seamless digital registration and authentication experience, encouraging greater digital adoption. By streamlining the eKYC process and promoting standardization, it enhances convenience, efficiency, and accessibility, fostering a user-friendly digital ecosystem in Cambodia.

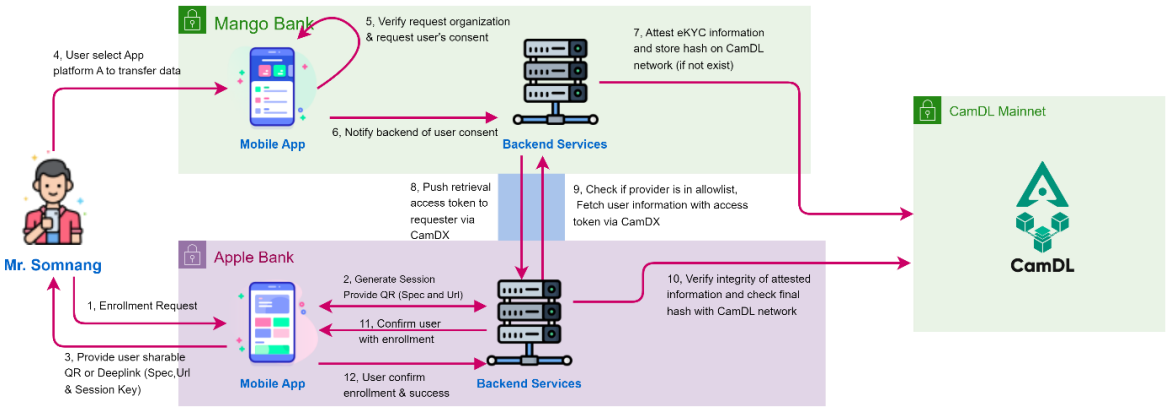

Figure 8 Full system flow of the eKYC exchange by the standard

Within the flow of the process, there are numerous request/respond steps, and this standard document aims to encompass most of them. However, it is important to note that certain request/respond interactions that do not involve cross mobile app communication or cross server data exchange have been deliberately excluded.

In this eKYCIS will cover in detail within various aspect including:

- eKYC session request payload format
- eKYC request payload exchange (Deeplinking or Qr code)
- Request payload verification and organization filtering
- Attestation templates (eKYC tiered list)
- Attestation framework, data obfuscation and proof of issuance (Blockchain)
- User consent for eKYC information exchange
- Authorization token
- Authorization token validation and provider filtering
- User information fetching
- User information response
- API fee
- Attestation verification

4.1. eKYC Session Request Payload Format

The eKYC session request payload serves as the data embedded with deeplinking or QR code, facilitating the user's sharing of the eKYC request from the requester platform to the provider platform. This payload is structured as a JSON Web Token (JWT) token, encompassing a header, body, and signature, as illustrated in the figure below. The header specifies metadata about the token, while the body contains the actual data relevant to the eKYC session request. Finally, the signature ensures the integrity and authenticity of the payload.

```
{
    "alg": "ES256",
    "typ": "JWT"
}.{
  "qr_type": "ekyc_interop_request_v1",
  "template": "LOGIN,T1_KYC,T2_KYC,T3_KYC",
  "jti": "RANDOM_STRING_UNIQUE_SESSION",
  "exchange_mode": "CAMBODIA_CAMDX_POST/GET OR DIRECT_API_POST/GET",
  "callback": "CALLBACK_URL_TO_REQUESTER_PLATFORM or CAMDX",
  "iss": "ISSUER",
  "iat": "ISSUE_AT",
  "exp": "EXPIRY"
}.ECDSA256(Header + payload)
```

Figure 9 eKYC session request payload format

Table 1 Request token keys and description

Key	Description
alg	ES256 algorithm is an asymmetric algorithm that uses a private key to sign a JWT and a public key to verify that signature
typ	JWT
qr_type	ekyc_interop_request (to specify this jwt is format of eKYC interop)
template	specify the intention of the request eg: Login, T1_KYC, T2_KYC, T3 KYC. Please see the templating section for details
jti	unique string for requester organization to link current login session to the respond return from provider platform
exchange_mode	CAMBODIA_CAMDX_POST/GET for data to callback via CamDX or DIRECT_API_POST/GET for direct api callback
callback	url or CAMDX api identifier for provider platform to respond after user consent on the eKYC exchange
iss	standard JWT Issuer claim
iat	standard JWT issue at claim
exp	standard JWT expiry claim

4.2. eKYC Request Payload Exchange (Deeplinking or Qr Code)

The eKYC request payload can be exchanged using two methods: deeplinking (app-to-app exchange) or QR code (web-to-app exchange). Deeplinking involves modifying the app to redirect users to a different authentication provider, whereas QR code exchange allows users to scan a QR code using their device's camera or media to initiate the eKYC process. QR code exchange requires fewer modifications to the app and offers a better user interface design, avoiding the need to overload the app with multiple deeplinking options. Ultimately, the choice between the two methods is left open for the requester and provider platform to decide upon.

Key	Description
qr	Request token in the form of JWT
callback	Deeplink scheme to be called when user authorized or reject

Figure 10 Deeplink Scheme

Upon user authorized or reject provider app is required to perform deeplink call back to requester app via the provided callback embedded with query param “status=fail/success” to notify the final result of user authorization

4.3. Request Payload Verification and Organization Filtering

4.3.1. Payload Verification

The request payload, in the form of a JWT token, is generated using the ES256 algorithm. This token signing mechanism enables verification of the token without requiring an API request to the token issuer. The integrity and issuance of the token can be verified by utilizing the issuer's public key. The public key of the issuer is obtained through the Domain Name System (DNS), a topic that will be covered in more detail in a appendix section.

It is crucial for the member to diligently verify the request payload to ensure that it has been issued by the correct organization. This verification is crucial for displaying accurate information on the user interface (UI) and obtaining user consent. Neglecting to verify the JWT token opens up the possibility of exploitation, where malicious actors could impersonate legitimate organizations. This could result in the unauthorized access and potential theft of user information during the eKYC exchange process.

4.3.2. Organization Filtering

Members have the freedom to select the organizations they wish to collaborate with, while also having the option to deny sharing customer information with organizations for various reasons. By inspecting the request payload, members can make informed decisions and deny eKYC requests if necessary. However, it is encouraged for members to foster cooperation in order to facilitate the digital adoption of users across different digital platforms. This collaborative approach helps promote seamless integration and enhanced user experiences in the digital realm.

4.4. Attestation Template (eKYC tiers list)

In order to facilitate seamless interoperability among systems involved in eKYC information exchange, it is essential to establish an attestation template. This template should be designed to cater to different tiers based on the specific use cases within various digital platforms. It is advisable for organizations to request only the necessary information that aligns with their specific business requirements. Requesting user information that is not essential for a particular use case is discouraged, as it has potential risk of exposing sensitive user data. By adopting this approach, organizations can strike a balance between obtaining the required information and safeguarding user privacy.

The eKYCIS recommends four types of user information templates.

- Login Information: could be used for login authorization. User could share minimal version of their information to prove authentication to various digital platforms.

```
{
  "user_info":{
    "phone_number": "USER_PHONE_NUMBER",
    "first_name": "USER_FIRST_NAME",
    "last_name": "USER_LAST_NAME"
  }
}
```

Figure 11 Login data template

- Tier 1 Template: which extend from the login information. This template provides additional information such as: identity document type, document id, email, date of birth, gender, issue date and expiry date. This is recommended for low level KYC requirement business scenario

```
{
  "user_info":{
    "phone_number": "USER_PHONE_NUMBER",
    "first_name": "USER_FIRST_NAME",
    "last_name": "USER_LAST_NAME",
    "document_type": "NID or PASSPORT or DRIVING_LICENSE",
    "document_id": "DOCUMENT_ID_NUMBER",
    "email": "EMAIL_ADDRESS(OPTIONAL)",
    "dob": "DATE_OF_BIRTH (yyyy-MM-dd)",
    "gender": "M_or_F",
    "issue_date": "DATE_OF_ISSUANCE (yyyy-MM-dd)",
    "expiry_date": "DATE_OF_EXPIRY (yyyy-MM-dd)"
  }
}
```

Figure 12 Tier 1 eKYC data template

- Tier 2 Template: this template extends on the previous tier. This template contains user facial image which is suitable for application which requires user to perform face registration etc.

```
{
  "user_info":{
    "phone_number": "USER_PHONE_NUMBER",
    "first_name": "USER_FIRST_NAME",
    "last_name": "USER_LAST_NAME",
    "document_type": "NID or PASSPORT or DRIVING_LICENSE",
    "document_id": "DOCUMENT_ID_NUMBER",
    "email": "EMAIL_ADDRESS(OPTIONAL)",
    "dob": "DATE_OF_BIRTH (yyyy-MM-dd)",
    "gender": "M_or_F",
    "issue_date": "DATE_OF_ISSUANCE (yyyy-MM-dd)",
    "expiry_date": "DATE_OF_EXPIRY (yyyy-MM-dd)"
  },
  "face_image": "BASE64_ENCODE_USER_FACE"
}
```

Figure 13 Tier 2 eKYC data template

- Tier 3 Template: this template extends previous tier further. This template adds user's document image which is suitable for application which requires official government document image for their registration.

```
{
  "user_info":{
    "phone_number": "USER_PHONE_NUMBER",
    "first_name": "USER_FIRST_NAME",
    "last_name": "USER_LAST_NAME",
    "document_type": "NID or PASSPORT or DRIVING_LICENSE",
    "document_id": "DOCUMENT_ID_NUMBER",
    "email": "EMAIL_ADDRESS(OPTIONAL)",
    "dob": "DATE_OF_BIRTH (yyyy-MM-dd)",
    "gender": "M_or_F",
    "issue_date": "DATE_OF_ISSUANCE (yyyy-MM-dd)",
    "expiry_date": "DATE_OF_EXPIRY (yyyy-MM-dd)"
  },
  "face_image": "BASE64_ENCODE_USER_FACE",
  "document_image": "BASE64_ENCODE_USER_DOCUMENT"
}
```

Figure 14 Tier 3 eKYC data template

4.5. Attestation Framework, Data Obfuscation and Proof of Issuance (Blockchain)

The design incorporates the Open Attestation Framework and the Cambodia Distributed Ledger (CamDL) blockchain network to enhance the eKYC process. The Open Attestation Framework establishes a standardized method for verifying and validating eKYC data, safeguarding it against forgery and tampering. The CamDL blockchain network further strengthens data security and integrity by leveraging the decentralized and immutable nature

of blockchain technology. This ensures that the eKYC data remains secure and resistant to unauthorized modifications. Additionally, the use of blockchain enables transparent and auditable records of the attestation process, instilling confidence among all stakeholders involved.

4.5.1. Open Attestation

Open Attestation is an open-sourced framework to endorse and verify documents using the blockchain developed by GovTech Singapore (https://www.openattestation.com). Documents issued with framework are cryptographically trustworthy and can be verified independently.

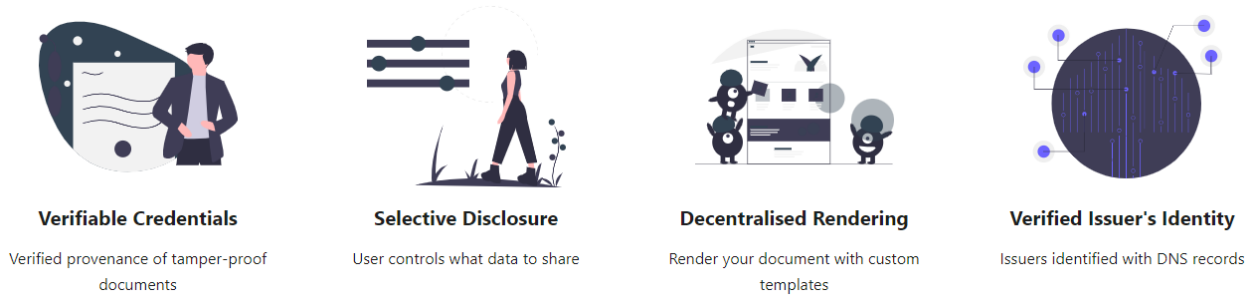

Figure 15 Principles of Open Attestation

```
{
  "version": "https://schema.openattestation.com/2.0/schema.json",
  "data": {
    "issuers": [
      {
        "identityProof": {
          "type": "salt:string:DNS-TXT",
          "location": "salt:string:DNS_ISSUER"
        },
        "name": "salt:string:ISSUERNAME",
        "tokenRegistry": " salt:string:SMART_CONTRACT_ADDRESS"
      }
    ],
    EKYC_TIER_DOCUMENT_WRAPPED_&_SALTED
  },
  "signature": {
    "type": "SHA3MerkleProof",
    "targetHash": "TARGET_HASH_OF_ATTESTED_DATA",
    "proof": [],
    "merkleRoot": "MERKLE_ROOT_OF_ATTESTED_DATA_BATCHES"
  },
  "privacy": {
    "obfuscatedData": ["LIST_OF_OBFUSCATED_FIELD"]
  }
}
```

Figure 16 Open Attestation - eKYC formatted document

4.5.2. Data Obfuscation

The eKYC tier list can leverage the selective disclosure functionality offered by the Open Attestation framework. This allows the eKYC provider to attest the highest tier eKYC document specific to a user. When receiving a request from an eKYC requester, the provider can employ data obfuscation techniques to conceal certain fields, while selectively displaying only the information requested by the requester platform. This approach eliminates the need for the eKYC provider to attest user information repeatedly or in multiple formats for attestation. By utilizing selective disclosure, the eKYC process becomes more efficient and streamlined, ensuring that only the necessary information is shared while maintaining data privacy and security.

4.5.3. Proof of Issuance (CamDL – Blockchain)

Once the attestation is created, the eKYC provider publishes the merkleRoot value into their organization's smart contract. In this context, each organization involved in the eKYC interoperability system has its own separate smart contract, which provides them with full sovereignty and control over the attestation process.

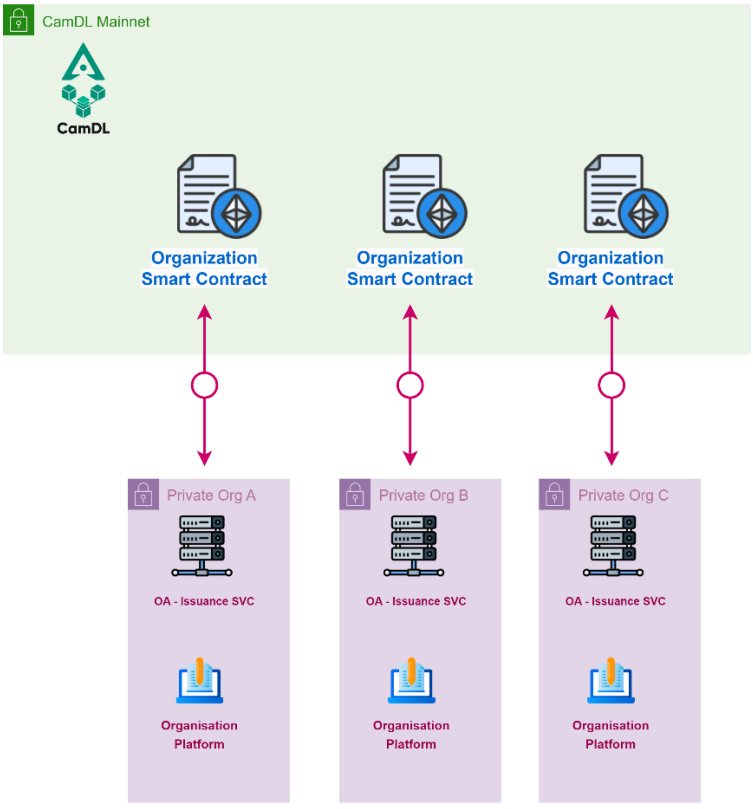

Figure 17 Organization ownership of smart contract within CamDL Blockchain Network

Within the eKYC interoperability system, CamDL plays a crucial role by providing a robust blockchain infrastructure that benefits all stakeholders involved. One of the key participants in this system is the eKYC provider, who can leverage the blockchain network offered by CamDL to attest KYC information. This process ensures the verification and validation of customer identity details.

4.6. User consent for eKYC information exchange

The eKYC provider is obligated to obtain user consent for information sharing. Users must be fully informed about the specific details of the information being exchanged and the platforms involved. Platforms have the flexibility to choose appropriate methods to confirm user consent, such as user digital signature (private key signing), PIN code, user access token, or username and password. However, it is crucial for platforms to ensure that consent is genuinely obtained from the rightful owner of the data.

4.7. Authorization Token

Once user consent is obtained, the provider platform is responsible for generating an authorization token. This token plays a crucial role in granting the platform the necessary rights to access and retrieve user information.

```
{
  "alg": "ES256",
  "typ": "JWT"
}.{
  "qr_type": "ekyc_interop_auth_v1",
  "template": "LOGIN,T1_KYC,T2_KYC,T3_KYC",
  "jti": "EKYC_REQUEST_ID_EXTRACT_FROM_REQUEST",
  "auth_id": "RANDOM_STRING_UNIQUE_SESSION",
  "callback": "CALLBACK_URL_TO_PROVIDER_PLATFORM",
  "exchange_mode": "CAMBODIA_CAMDX_POST/GET, DIRECT_API_POST/GET",
  "iss": "ISSUER",
  "iat": "ISSUE_AT",
  "exp": "EXPIRY"
}.ECDSA256(Header + payload)
```

Figure 18 eKYC authorization token format

Table 2 eKYCIS authorization token description

Key	Description
alg	ES256 algorithm is an asymmetric algorithm that uses a private key to sign a JWT and a public key to verify that signature
typ	JWT
qr_type	ekyc_interop_auth_v1 (to specify this jwt is format of eKYC interop authentication)
template	LOGIN,T1_KYC,T2_KYC,T3_KYC
jti	unique string for requester organization to link current login session to the respond return from provider platform
auth_id	unique string for provider organization to link authentication session within their system
callback	url or CAMDX api identifier for provider platform to request for user information
exchange_mode	CAMBODIA_CAMDX_POST/GET for data to make request via CamDX or DIRECT_API_POST/GET for direct api callback
iss	standard JWT Issuer claim
iat	standard JWT issue at claim
exp	standard JWT expiry claim

4.8. Authorization Token Validation and Provider Filtering

4.8.1. Token Validation

Upon receiving the authorization token, the requester platform must carry out a verification process to ensure the authenticity and integrity of its source. This verification step is essential in preventing potential exploits, where malicious actors could impersonate the issuer platform and provide forged information back to the requester platform.

4.8.2. Provider Filtering

Just as the provider has the ability to filter organizations, the requester is also given the opportunity to select the organizations they wish to collaborate with or fetch user information from. If the provider organization is included in the requester's whitelist, the requester can proceed with fetching the user information. However, if the provider organization is not included in the whitelist, the requester has the option to terminate the process without proceeding to fetch the user information.

4.9. User Information Fetching

After verifying the authorization token and provider organization, the requester organization can proceed with fetching user information. The fetching process involves retrieving specific information stored within the payload of the authorization token, such as the URL or exchange mechanism. However, to enhance security and prevent unauthorized access by malicious actors who might intercept the data exchange, a fetching signature is required. The requester organization generates this fetching signature using the authorization token's header and payload. This token follows the standard ES256 and utilizes the requester's key. Subsequently, the provider verifies this signature to ensure the requester's identity before responding with the requested user information

```
{
  "auth_token": "RECEIVED_AUTH_TOKEN",
  "signature": "GENERATED_FETCH_SIGNATURE"
}
```

Figure 19 User information fetching

4.10. User Information Response

In this step, provider organization is required to carry out the final verification of the issued authentication token and also confirm the identity of the requester organization by confirming the fetch token within the request payload. This meticulous validation process is essential to prevent any malicious actors from impersonating the requester organization and gaining access to sensitive user information. Once the validation is successfully completed, the step provider organization promptly responds by attesting and providing the requested user information back to the requester.

4.11. API Fee

In order to foster fairness and incentivize responsible data exchange, API providers have the option to apply a fee charge to requester organizations. It is important to note that this fee application is not mandatory, but rather a choice available to the API providers. If a fee is implemented, it should only be applied once the requester organization proceeds with the actual retrieval of user information. Transactions that are abandoned or fail before the exchange of user information should not be subject to any charges. This approach ensures fairness among the members of the open KYC system, as fees are only incurred when there is a successful and completed transaction for accessing user information.

4.12. Attestation Verification

Upon receiving the attestation document, the requester information is required to undergo attestation verification using either the Open Attestation framework or the Open Attestation Service provided by the operator. This verification process ensures the integrity of the document. Subsequently, proof of issuance can be performed by checking the DNS record of the issuer within the document, which contains the smart contract address or the EVM-compatible network utilized. Once the smart contract and blockchain network are accessed, a smart contract method can be invoked to verify if the merkleRoot has been issued within

the smart contract. This serves as proof that the organization has indeed issued the document, establishing its authenticity and legitimacy.

After undergoing attestation verification, the requester organization gains the confidence to seamlessly enroll customers into their system, reassured by the fact that the customer's information has been meticulously verified and confirmed by the trusted eKYC provider. This ensures a streamlined process, eliminating any doubts or concerns regarding the authenticity and accuracy of the customer's data.

5. Integration

5.1. Open Attestation – CamDL Helper Service

This mono repository is designed to assist developers in testing and integrating the eKYC (Electronic Know Your Customer) interoperability standard. It provides a centralized codebase containing various services, a mock web portal, a mock mobile application, and components for deeplink requesting and providing.

The repository is organized as a mono repository, which means that all the necessary code and components are stored in a single repository, facilitating easier development and testing. The mono repository structure allows developers to work on different parts of the system simultaneously while maintaining a consistent codebase.

The key components included in this mono repository are as follows:

- **Services**: The repository includes the necessary backend services required for eKYC interoperability. These services handle data processing, verification, and communication with external systems.
- **Mock Web Portal**: A mock web portal is provided to simulate the user interface for accessing the eKYC functionality. The portal provides a user-friendly interface for initiating and managing eKYC requests.
- **Deeplink Requester Demo Mobile App**: This component allows developers to test the deep linking functionality of the eKYC system. It enables the initiation of eKYC requests from external applications by generating deeplinks and interacting with the eKYC services.
- **Deeplink Provider Demo Mobile App**: The deeplink provider component is responsible for handling incoming eKYC deeplinks and processing the requests. It interfaces with the eKYC services to perform the necessary verifications and provide the requested information.

To get started with using this mono repository, please refer to the README file located within the submodule in the "apps" folder. The README will provide detailed instructions on setting up the environment variables required for the system to function properly.

Once the environment variables are configured, you can use the provided docker-compose.yaml file to start the necessary services and components. This will enable you to begin testing and integrating the eKYC interoperability standard into your application.

Please note that this description provides a general overview of the mono repository and its components. For more specific instructions and details, please refer to the documentation and README files provided within the repository.

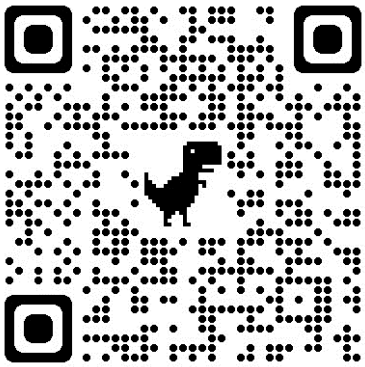

Figure 20 ekyc Interoperability Standard Mono Repository

6. Appendix

6.1. JWT Issuer Identity

Within the framework of this eKYCIS, a multitude of JWT token exchanges occur between parties throughout various processes. The ability to seamlessly and swiftly verify the identity of these tokens without relying on direct API verification is of utmost importance. The standard specifies ES256 as the designated token standard for issuance. This token algorithm allows for the verification of the token's issuance identity using the issuer's public key, eliminating the need for an API call to the issuer for verification. However, it is essential to obtain the public key in order to verify the identity of the issuer.

6.1.1. DNS Identity Proof

Organizations that utilize this eKYCIS have the flexibility to configure their public key by leveraging DNS TXT records on their Domain Name System (DNS). This approach empowers all organizations to have complete control over their key settings or necessary key rotations, without the need for approval from another central authority. By utilizing DNS TXT records, organizations can independently manage and update their public keys, ensuring a secure and efficient process. This decentralized approach eliminates the dependency on external entities, granting organizations the autonomy to maintain and modify their key configurations as per their specific requirements.

Our standard offers users the flexibility to register two public keys on their DNS TXT record. This feature serves a crucial purpose in preventing downtime during key rotation. By allowing organizations to have multiple keys registered, they can seamlessly phase out outdated keys without triggering a cascading failure in the validation of JWT tokens. This intelligent approach ensures a smooth transition during the key rotation period, maintaining uninterrupted

service. Below is an illustrative example of a TXT record that demonstrates compliance with our standard.

Table 3 DNS TXT Record of JWT Public Key

Type	Name	Value
TXT	example.com	" ekyc_jwt pub=<URL_ENCODE_BASE64>"
TXT	example.com	" ekyc_jwt pub=<URL_ENCODE_BASE64>"

During the JWT verification process, a comprehensive check is performed to ensure the validity of the token. This involves verifying the JWT with every key listed within the DNS TXT record of the corresponding domain. Only after confirming that the token does not match any of the keys in the list can it be deemed invalid.

6.1.2. CamDX Member Code Identity Proof

In certain scenarios involving the exchange of data via Cambodia Data eXchange (CamDX), the process of retrieving public key records operates differently. Due to the absence of DNS TXT capability within the CamDX architecture, a centralized public API is made available to all members. This API serves the purpose of resolving CamDX member codes to their respective system domain names. Once the domain name is obtained through the API, organizations can proceed with the previously mentioned process of retrieving the public key and commence the verification of JWT tokens. This adaptive approach ensures a seamless integration of CamDX within the existing infrastructure, enabling secure data exchange and reliable authentication.

The maintenance of this central API is the responsibility of the CamDX – eKYCIS operator, who acts as the central repository for member codes. This ensures accurate resolution of member codes to their respective domains. It is worth noting that each organizations have the autonomy to perform key rotation without the need to inform the operator, as discussed in the previous section.

6.2. Document Attestation Store

As discuss in previous section, Open Attestation guarantees the integrity of attested documents by employing robust hashing algorithms, thereby preventing any unauthorized alterations. Moreover, it offers comprehensive support for tracking the issuance and revocation status of documents. Open Attestation facilitates document issuance and revocation through two distinct methods: utilizing Decentralized Identifiers (DIDs) and leveraging smart contracts on EVMs compatible blockchain networks.

Within our standard, a smart contract on the EVM is selected. Each organization owns a dedicated smart contract where they can store the issued merkleRoot of the document they want to attest on their smart contract. At the same time, revocation of the merkleRoot can also be carried out within this smart contract. Details about the smart contract code can be viewed using the link below.

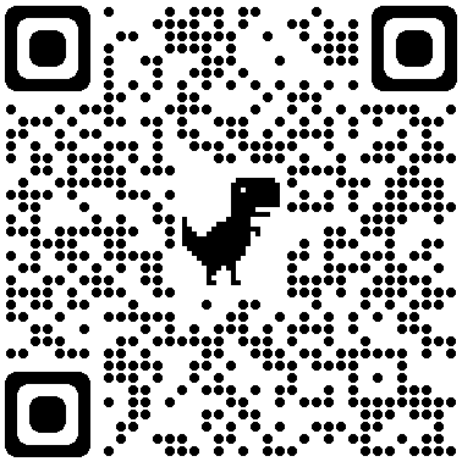

Figure 21 Open Attestation Document Store link

6.2.1. DNS Identity Proof

However, smart contract addresses are designed to be anonymous, which means that the ownership of the smart contract is initially unknown. Nevertheless, organizations have the ability to attest their smart contracts by utilizing DNS records. By configuring DNS record settings for a domain, organizations can establish a means to demonstrate their utilization of the contract for issuing and revoking attestation documents.

Table 4 Sample DNS TXT Setting for Smart Contract Identity Proof

Type	Name	Value
TXT	example.com	"openatts net=ethereum netId=<Chain_ID> addr=<SMART_CONTRACT_ADDRESS>"

- netID: Chain ID number for Ethereum network (95 for CamDL Mainnet)
- addr: Smart contract address

6.3. Attestation Verification and Issuance

Verification and issuance of attestations are the most important aspects of the eKYCIS. Organizations must ensure that the attested document has not been tampered with and verify the identity of the issuer. This can be achieved by using: eKYC interoperability - The OA service is provided by the operator of this standard or OpenAttestation with the Web3 library.

6.3.1. Open Attestation & Web3 Library

The Open Attestation framework provides a range of repositories and libraries that serve as a starting point for utilizing the technology. However, it's important to note that these framework libraries primarily focus on attestation verification and creation. The generation of tokens or issuance of merkleRoot to smart contracts is typically handled by other code libraries such as Web3 or JWT library etc.

Figure 22 QR code link to the Open Attestation library

To facilitate interaction with the blockchain, the web3 library is utilized for committing transactions and retrieving data from smart contracts. The web3 library is available in various languages, including Web3js (JavaScript), Web3J (Java), Nethereum (C#), and more. Organizations have the freedom to select the library that aligns with their specific tech stack requirements.

6.3.2. eKYC Interoperability – Open Attestation Service

To streamline the integration with the Open Attestation framework for document verification, issuance, and revocation, the eKYC interoperability operator provides a dedicated service that can be seamlessly integrated into existing systems. This eKYC interoperability service, written in TypeScript, is an open-source application and specifically designed to assist organizations in carrying out various interactions related to Open Attestation functionalities and beyond.

Organizations can conveniently interact with this service through a range of API endpoints that are provided. This service extends the capabilities of the existing Open Attestation framework by incorporating additional mechanisms such as issuer DNS verification, issuance status verification using smart contracts, revocation of existing attestations, and much more.

Please note that at the time of writing this document, the mentioned service is still under development and will be released once the development and testing phase is completed. For further information, we kindly request you to reach out to the standard operator for more information.

6.4. Attestation Obfuscation & KYC Tiered System

The Open Attestation framework provides obfuscation features that enable organizations to selectively conceal specific fields within attested documents. This functionality allows organizations to share information selectively, granting them the option to hide certain fields from requesters while still ensuring the verifiability of the attested document.

The obfuscation features provided by the Open Attestation framework can greatly benefit the eKYC tiered system. Organizations have the opportunity to attest user information once with the highest possible tier. When a request is received from a requester organization for a lower tier, the organization can selectively hide fields that are not part of the specific eKYC tiered request without requiring another attestation.

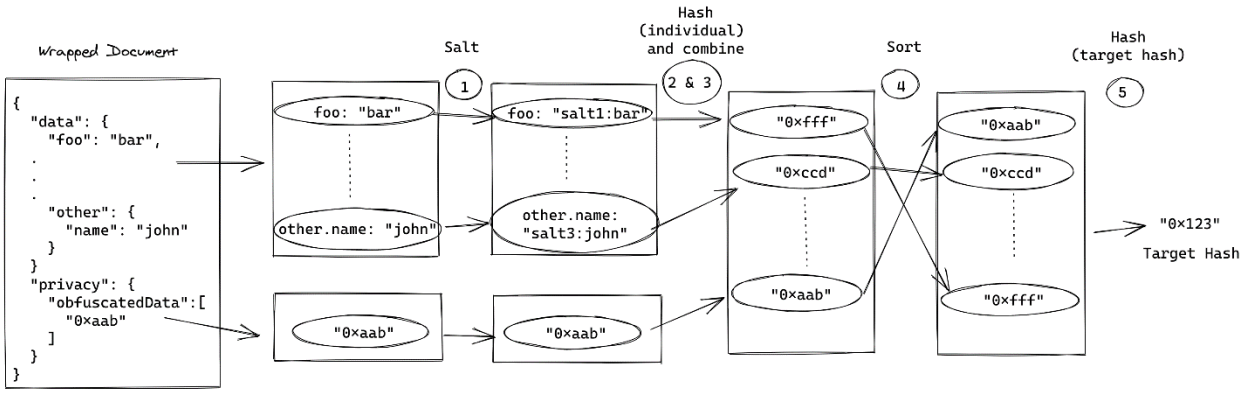

Figure 23 Mechanism of generating targetHash with selective disclosure

6.5. CamDX Billing System

The CamDX Billing System is a service provided by CamDX operators. This service enables organizations within the CamDX ecosystem to access various features. It allows organizations to view transaction counts associated with their security server and generate invoices for API consumption by other organizations. The system plays a vital role in helping organizations monetize their APIs and provides visibility into the API consumption made by their organization.


TECHO STARTUP CENTER


CamDL