Pluggable Integrity layer for Property Registration

1st Ras Dwivedi
Indian Institute of Technology, Kanpur

2nd Mukul Verma
Trential

2rd Tanmay Yadav
Trential

3th Prof. Sandeep Shukla
Indian Institute of Technology Kanpur

***Abstract*—Property registration plays a crucial role in property transfers; hence, it is necessary to ensure the transparency and integrity of the property registration records. In this paper, we present our successful deployment of a Hyperledger Fabric [1] based integrity layer for property registration in the state of Karnataka, India. Unlike traditional approaches that build the entire system on the blockchain, we propose an integrity layer approach that leverages the benefits of blockchain while addressing its limitations in terms of speed and complexity. Our solution aims to enhance the existing property registration system by integrating a pluggable layer that can be adopted by any state, considering the diverse state laws governing land (property) in India. We comprehensively address various practical scenarios in property records, including the registration of attorneys and encumbrances on properties. To facilitate user interaction with the blockchain, we have developed a Java card as a hardware wallet, which could be of independent interest.**

***Index Terms*—blockchain, property records, integrity layer, Java Cards, smart contract**

I. Introduction

Since its introduction in 2009, blockchain technology [2] has been hailed as a solution to real-world problems rooted in a lack of trust. However, harnessing the potential of blockchain to create practical solutions poses significant challenges. Blockchain is notorious for its scalability issues [3] and comparatively low throughput when compared to other modern technologies. Moreover, its advanced nature presents usability challenges for the average user, limiting its applicability.

The issue of trust and poor record-keeping in property registration has persisted as a significant problem in India. Due to presumptive property ownership [4], the state does not give a guarantee for the title. Ownership is proved using the chain of documents that provide proof of transfer of property from one entity to another all the way to the current owner. Multiple departments handle land(property) related documents, each potentially claiming ownership but none serving as an authoritative source. Despite the ongoing digitization program initiated in 2008, [5], discrepancies and inconsistencies still plague land records in India. Moreover, the diversity of land laws across states adds another layer of complexity, hindering the uniformity of governance in this domain.

The consequences of improper record-keeping and property-related disputes are far-reaching. A study conducted by the World Bank in 2007 [6]indicated that property disputes accounted for two-thirds of all pending court cases in India. Furthermore, according to a report, property disputes take an average of 20 years to be resolved [4]. These disputes hinder economic development by strangling agricultural credit and blocking infrastructural development. Given the magnitude of the challenges in the existing property registration system, we propose the implementation of a blockchain-based pluggable integrity layer as a potential solution. We focus on utilizing blockchain technology to create a transparent, immutable, and scalable property registration portal.

In developing the integrity layer, we carefully considered the diverse landscape of land laws in India and the varying levels of technological proficiency among the Indian population. Our objective was to create a system that is simple, intuitive, and accessible to users from different educational backgrounds. By prioritizing user-friendliness, we aimed to ensure that individuals with varying technological expertise can easily interact with the system and benefit from its advantages.

The paper is organized as follows. We first present an introduction to the public and private blockchain, focusing on Hyperledger Fabric. We discuss the problems with property registration in India in section 3. We discuss our chaincodes and process in section 4, followed by the integrity layer in section 5. We then describe the blockchain architecture.

II. Background and related work

A. Blockchain

Since its emergence in 2009 with the introduction of Bitcoin [2], blockchain technology has gained significant attention for its potential to revolutionize various industries. At its core, a blockchain is a distributed, decentralized ledger that records transactions in a transparent and immutable manner. It relies on cryptographic techniques to ensure security and consensus among participants, eliminating the need for intermediaries and fostering trust in a trustless environment.

Two prominent variants of blockchain technology are public and private blockchains. Public blockchains, exemplified by Bitcoin [2] and Ethereum [7], are open to anyone who wishes to participate in the network. They operate on a peer-to-peer basis, where multiple independent nodes validate transactions and maintain the blockchain's integrity through a consensus mechanism, such as proof-of-work (PoW) or proof-of-stake (PoS). Public blockchains offer transparency, security, and immutability, making them suitable for applications that require decentralized trust and public scrutiny of transactions.

On the other hand, private blockchains restrict access to a limited number of authorized participants. Unlike public blockchains, private blockchains are permissioned, meaning that participants must be granted specific privileges to join the network. Private blockchains offer enhanced privacy and control over the network, making them appealing for enterprise use cases where confidentiality is crucial. They typically employ consensus mechanisms such as practical Byzantine fault tolerance (PBFT) [8] or Raft [9], which prioritize speed and efficiency over the extensive computational requirements of PoW.

B. Hyperledger Fabric organization structure

Hyperledger Fabric [1], an open-source project under the Linux Foundation's Hyperledger initiative, is a notable private permissioned blockchain framework. It offers a flexible and modular architecture for building distributed ledger applications catering to enterprise needs. In the Hyperledger Fabric network, organizations share the blockchain and establish connections through channels, which serve as independent ledgers. This approach allows organizations to maintain privacy and confidentiality within their respective channels while still benefiting from a shared ledger for common transactions. Organizations have multiple peers. The peer structure of the organization is described below.

1) Membership Service Provider (MSP): The MSP node acts as a certificate authority within the Hyperledger Fabric architecture. It utilizes x.509 certificates through a public key infrastructure. Each organization within a business network runs its own certificate authority.
2) Client Node: The client node serves as the end-user interface, housing the middleware code and a REST API server to interact with the middleware. It enables users to interact with the Hyperledger Fabric network.
3) Peer Nodes: The backbone of the Hyperledger Fabric network, peer nodes play crucial roles in maintaining the ledger and validating transactions. There are two types of peer nodes:
    a) Committing Peer: This node updates transactions in the ledger, ensuring its integrity and consistency.
    b) Endorsing Peer: The endorsing peer node executes the chaincode to validate transactions, endorsing their validity.
4) Ordering Node: Ordering nodes manage the transaction order and block inclusion. They reorder transactions if necessary and determine which block the transactions will be included in. Unlike MSP nodes, ordering nodes do not need to be present in each organization. However, at least one organization must operate an ordering node.

III. Problems with the Property Registration Process in India

In the Indian context, property records encompass a range of documents such as property maps, sale deeds, and records of rights that carry pertinent information. Different departments at the district or village level maintain these records. Unfortunately, these departments often operate independently, resulting in discrepancies and inconsistencies. In some instances, property records may not accurately reflect the current state of affairs on the ground due to incomplete or missed surveys.

Adding to the complexity, property ownership in India is presumptive, implying that the person in possession is considered the rightful owner unless proven otherwise. In India, the property transfer is a two-step process, registration of the sale deed, followed by the mutation. However, it is important to note that these documents serve as supporting evidence for establishing ownership and are not a government guarantee of title. Apart from the complicated legal framework, we found that the following problems plague the Indian property registration system.

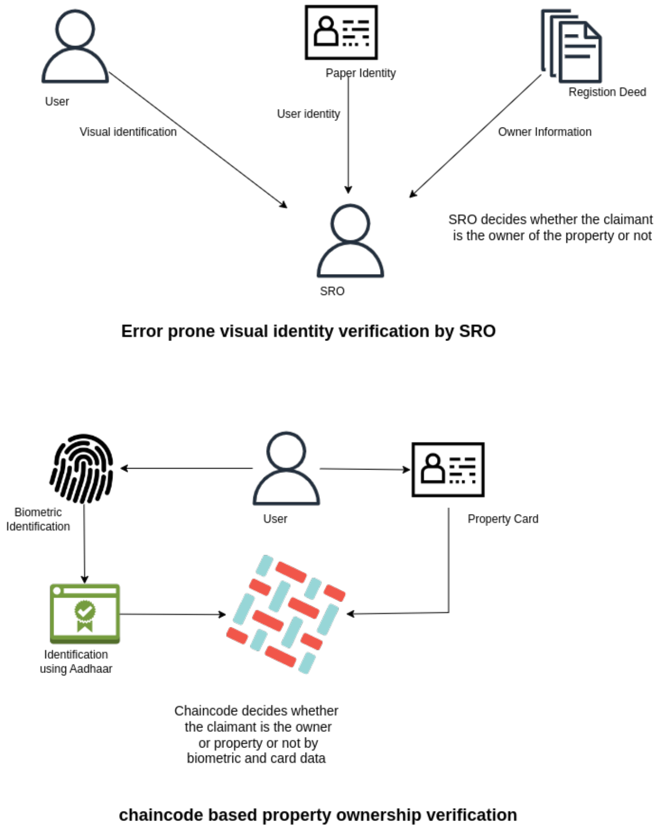
User
Paper Identity
Registion Deed
Visual identification
User identity
Owner Information
SRO decides whether the claimant is the owner of the property or not
SRO
Error prone visual identity verification by SRO
Biometric Identification
User
Property Card
Identification using Aadhaar
Chaincode decides whether the claimant is the owner or property or not by biometric and card data
chaincode based property ownership verification

Fig. 1: Identity verification flow

A. Identity verification

Despite focusing on digitized documents, we encountered many paper-based identity documents used by individuals, including Aadhaar cards [10], voter ID cards, and driving licenses, among others. This diversity of paper-based identity proofs presented two distinct challenges. Firstly, these documents were susceptible to forgery, making verification a complex and unreliable task, especially through visual inspection alone. Secondly, the existence of multiple forms of paper identity proof complicated the process of establishing a clear mapping between property owners and their respective properties.

Currently, the Sub Registrar is responsible for verifying the paper identity, asserting that the holder is the same as depicted in the identity proof and asserting that the property actually belongs to the holder. We tried to address these challenges through the widely available digital identity layer provided by Aadhaar. By associating each user with a unique 12-digit number, we were able to establish a more streamlined connection between property owners and their properties. Additionally, Aadhaar's integration with biometric data added an extra layer of security, making it difficult to forge or manipulate. While utilizing Aadhaar resolved a part of the problem, we relied on the judgment of the Sub-Registrar to link users with their respective properties in cases where property owners had not been previously associated with an Aadhaar number. Once the properties have been linked with the Aadhaar number, future property ownership verification can be done via smart contract. Fig 1 depicts the change in the process flow with aadhaar based verification.

B. Consent management and User Wallet

The legal language used in registration deeds is quite complex. Users often sign these documents but later dispute the text of the deed. We created a user-friendly portal to take consent on each transaction via digital signature. However, storing user keys was a major challenge. Users cannot be trusted to store their private keys securely without a wallet. Relying on centralized servers for wallet storage introduces a single point of failure, leaving multiple keys vulnerable to compromise in the event of an attack.

To mitigate these concerns, we chose to implement hardware wallets. Nonetheless, hardware wallets come with their own set of challenges, including increased operational costs, particularly in densely populated countries like India. While hardware wallets ensure secure key storage, the task of linking these keys to the Aadhaar identity remained a significant challenge.

In traditional systems such as Bitcoin or Ethereum, where accounts are solely identified by the public key, losing the hardware wallet would lead to irretrievable key loss. To overcome this, we adopted Java Card [11] as a more cost-effective alternative for hardware wallets. Java Card provided a secure means of storing keys and preventing leaks. Additionally, we developed an elaborate protocol to establish a linkage between the user's identity and the card number, allowing for key rotation in case of wallet loss or damage.

1) Security considerations: The choice between online and offline wallets depends on the security considerations. Our system required the following security feature for the private keys

1) Secure Generation and Non-Exportability: It is imperative that the private keys are generated in a secure manner within the wallet itself. This ensures that the keys are not exposed externally and remain confined to the wallet. By generating the private keys securely, the risk of unauthorized access or compromise is significantly reduced.
2) PIN-based Locking and Device Lockout: The private keys stored in the wallet should be protected by a PIN code. This PIN serves as an additional layer of security, preventing unauthorized individuals from gaining access to the keys. Additionally, the wallet should be programmed to initiate a lockout after a certain number of incorrect PIN attempts (e.g., three incorrect attempts). This feature mitigates the risk of brute-force attacks and protects against unauthorized access to private keys.
3) Key Usage for Authorization, Not Identification: In our system, it is essential to differentiate between user identification and authorization. The private key stored in the card is used solely for authorization purposes or to perform specific actions. User identification, on the other hand, is conducted online through Aadhaar using biometric authentication. Even if a user's authorization is granted (i.e., possession of the private key is verified), without successful identification, the user is not permitted to carry out any transactions. This approach strengthens security by ensuring that only authenticated individuals can execute operations, adding an extra layer of protection against unauthorized usage.
4) Key Reset Mechanism and Property Loss Prevention: To address the scenario where a user forgets their PIN, it is vital to have a mechanism in place that allows for the reset of user keys. This ensures that users are not permanently locked out of their private keys and can regain access after a PIN reset process. It is essential to design the system in such a way that the loss or unavailability of the card does not lead to the loss of property.

2) Java Cards vs HSM: Considering our aim for a decentralized architecture in the property registration system, we evaluated different options for key storage, ultimately narrowing down the choices to HSM (Hardware Security Module) wallets and Java Cards. HSMs are specialized devices designed for secure key storage, typically provided by licensed Certificate Authorities (CAs) after thorough verification. However, due to cost considerations, we opted for Java Cards as a more affordable alternative to HSM devices.

Java Cards offer secure key generation and random number generation mechanisms similar to HSMs. Private keys are generated within the secure section of the card and remain stored within it without leaving the card's boundaries. The card itself is protected by a triple DES key, ensuring that unauthorized third parties cannot access the keys without proper authorization.

Although Java Cards lack the hardware protection and self-destruct features found in HSMs, they provide greater versatility. Java Cards allow for the loading and execution of multiple applets, which can be isolated from each other using the applet firewall. This capability is particularly advantageous in the property registration system, where different departments may need to run their own business logic within the card. Java Cards not only store keys securely but also accommodate identity and property data, as well as custom applets.

To address the inherently weaker hardware-based security mechanisms of Java Cards, we introduced an additional layer of identity verification through biometrics. This ensures that even if there is a potential compromise of the private keys, unauthorized access would be prevented due to the biometric identity verification requirement.

C. Diversity of the land laws

Land in India is subject to state-level jurisdiction, with each state having its own laws and procedures for property mutation and transfer. Although there are similarities in the systems across states, minor differences exist, necessitating individual property registration and mutation portals for each state. This diversity of laws becomes a major challenge in creating a unified solution that can be adopted nationwide. Recently, the government has introduced the National Generic Document Registration System (NGDRS) [12], a project initiated by the Department of Land Resources, Ministry of Rural Development, Government of India. NGDRS is a common, generic, and configurable application developed for registration departments across the country

We wanted to design a generic system that could be used nationwide, so we created a pluggable layer over the existing registration system. This pluggable layer would accommodate the common laws across the states and provide a layer of integrity to the current system.

Creating a pluggable layer also comes with added benefits. Since users are already familiar with the existing system, the learning curve for the new system is reduced.

The State of Karnataka employs software called Kaveri [13] to record all registration documents. In the pilot implementation, we focused on specific articles within the 106 governing articles of the Karnataka property registration process. These articles included the registration of property transfer deeds through sale, donation, or gift, as well as the registration and removal of encumbrances, registration of power of attorney, registration of partition deeds, and relinquishment of rights.

IV. Chaincode and Processes

A. Card creation

In our property registration system, each owner is identified by a unique card number, which is associated with their Aadhaar number. The mapping between cards and Aadhaar numbers follows a many-to-one relationship, allowing multiple cards to be associated with a single Aadhaar number. This flexibility accommodates various scenarios where a user can represent themselves or an institution. Each card is only linked to one Aadhaar number at a time. Additionally, properties are mapped to the card numbers to establish ownership records. When a new user registers, a unique card number is generated, and a card is created. A public-private key pair is generated inside the card and secured with the user's PIN. The Card Creation chaincode is then invoked to store the mapping between the card number and public key on the ledger. The card number depends on the location of the SRO and is generated in a serial-wise manner. The last bit of the card is the check bit to check that the user does not accidentally provide incorrect card numbers. We use the Verhoeff algorithm [14] to generate the check bit. The chaincode ensures the uniqueness of the card number during this process.

However, it is important to note that malicious users could exploit this method to link a card number with a key that was not generated inside the card. To address this concern, we have implemented another method linking the user with the card. This ensures that the mapping between the card and the user is accurate and prevents incorrect mapping of the card with a public key.

B. Mapping of the user to card

In order to mitigate the incorrect mapping of the card number with the card, we created a protocol to link the card number, public key and the Aadhaar number together. We utilized the e-signature facility offered by UIDAI (Unique Identification Authority of India) [10]. This facility temporarily assigns a key to a user with Aadhaar authentication and authorization, allowing the key to be used once for signing on behalf of the user. After one signature, the e-sign key is deleted. We leveraged this e-sign facility to link the blockchain key with Aadhaar.

In our protocol, the user initiates the process by retrieving the eKYC details from Aadhaar through biometric verification. These eKYC details are then combined with the card number and the public key. The data is signed using the e-sign facility provided by Aadhaar. Subsequently, the user signs the already signed data using their own card's private key and personal PIN. This doubly signed data is then transmitted to the blockchain, invoking the corresponding chaincode. Fig 2 depicts the data signed using the e-sign facility and by the private key stored in the card.

Upon receiving the data, the chaincode performs several verification steps. Firstly, it verifies that the eKYC data is appropriately signed by UIDAI, ensuring its authenticity. Next, it confirms that the e-sign key corresponds to the Aadhaar eKYC data and validates the signature on the card number and the associated public key. This validation ensures that the user has provided explicit consent for the specified card number and its linked public key. Furthermore, the chaincode verifies that the card number is correctly mapped to the corresponding key within its own ledger. Chaincode then verifies the signature via the public key associated with the card. This removes the possibility of incorrect mapping between the user and the card.

C. mapping of property details with the card

The process of mapping a property with its owner involves several steps to ensure data authentication and validation. Initially, the citizen visits the sub-registrar's Office (SRO) for property registration, where e-KYC is performed to verify their identity and generate a unique Aadhaar hash.

The Sub-Registrar verifies the original deed and index data from the Kaveri system by comparing them with the physical copies provided by the citizen. This verification step ensures the accuracy of the deed information. Subsequently, the SRO signs the e-KYC data, deed details, index data, and card numbers to validate the deed information and invokes the chaincode.

The smart contract verifies the sub-registrar's signature, Aadhaar hash, card number, e-KYC details, and property deed details. The smart contract includes checks such as validating

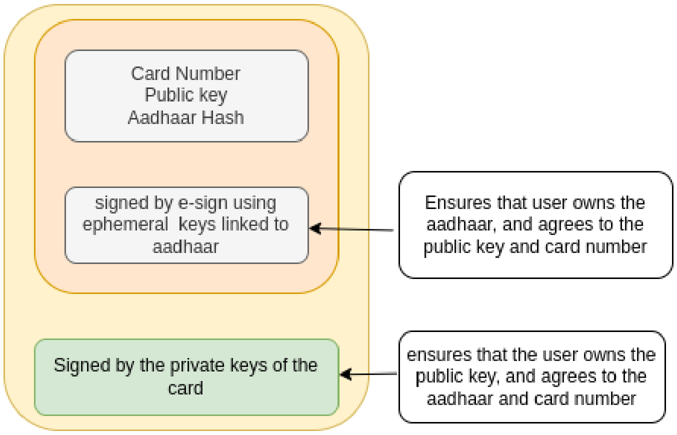
Card Number
Public key
Aadhaar Hash
signed by e-sign using ephemeral keys linked to aadhaar
Ensures that user owns the aadhaar, and agrees to the public key and card number
Signed by the private keys of the card
ensures that the user owns the public key, and agrees to the aadhaar and card number

Fig. 2: User data is signed both by Aadhaar linked e-sign key and private key in the card

the card number against the Aadhaar hash, confirming the mapping between the card's Aadhaar hash and the e-KYC data (establishing ownership), and verifying the UIDAI signature on the e-KYC details. The sub- registrar's signature is also validated.

Upon completing these checks, the smart contract maps the card number with the property, securely recording the ownership association within the blockchain. This mapping ensures transparent and secure tracking of property ownership information.

D. Property ownership verification

The verification process for the ownership of the property involves the following steps:

1) The verifier places the card on the card reader in the sub-registrar's Office (SRO).
2) The Sub-Registrar fetches the card number and Aadhaar hash from the card, along with the list of properties associated with the card number in the blockchain.
3) The verifier selects the specific property they want to verify.
4) The verifier retrieves the e-KYC details from UIDAI (Unique Identification Authority of India) and invokes the ”verify property” chaincode.
5) The chaincode verifies the e-KYC data and the UIDAI signature on the e-KYC data. It also confirms the linkage between the card and the Aadhaar in the e-KYC data and between the property and the card number. The chaincode then returns the result of the verification. It is important to note that in the case of multiple property owners, each owner must undergo the e-KYC verification process. All verifiers' card numbers and e-KYC details are then sent to the chaincode for verification.

E. Registration of sale deed

The Property Registration Chaincode is responsible for facilitating the registration of sales deeds and ensuring the verification of the involved parties. The chaincode performs various checks to ensure a secure and valid transaction. The detailed steps of verification carried out by the chaincode are as follows:

1) Verification of Owners:
   a) The chaincode verifies that all property owners have signed the transaction.
   b) Each owner must provide their e-KYC details obtained from Aadhaar.
   c) The e-KYC details should be within a reasonable time limit and not outdated.
   d) Chaincode checks that the Aadhaar hash stored in the owner's card matches the Aadhaar hash sent by UIDAI (Unique Identification Authority of India).
2) Verification of Transaction Type:
   a) The chaincode selects the type of transaction (e.g., sale deed) and ensures that all owners have signed it.
3) Verification of Buyers:
   a) Similar to owners, all buyers must provide their identity verification through e-KYC details and card information.
4) Verification of Index Data:
   a) The chaincode verifies the index data retrieved from Kaveri, which contains information about the property and its associated records.
5) Verification by Subregistrar (SRO):
   a) The chaincode ensures that the Subregistrar (SRO) has signed the transaction, validating their authorization and involvement in the registration process.
6) State Mutation and Transfer of Property
   a) If all the verification steps are successfully completed, the chaincode executes the transaction, leading to the mutation of the state in the blockchain.
   b) The property ownership is transferred from the seller to the buyer, officially completing the registration process.

By performing these thorough verifications, the Property Registration Chaincode guarantees the authenticity and integrity of the sales deed registration, providing a secure and reliable process for property transactions.

F. Other Chaincodes

We also have the following other chaincodes which we omit due to the paucity of space

1) Register Attorney: to register a power of attorney
2) Transfer Through attorney: Executing sale deed through an attorney
3) Property Partition Chaincode: Partitioning a parcel of land into two or more fragments
4) Registration and Removal of encumbrance: To register and remove encumbrances on the property

5) Relinquishment of Rights: To forfeit right on the property
6) Loss of Card: Issuing new card in case card is lost
7) Death of Card Holder: transferring property to the successor in case of death of card holder

V. Integration with Kaveri

Our integration approach involved the development of a pluggable integrity layer compatible with the existing property registration systems. This generic layer can be easily integrated with any property registration system, offering versatility and adaptability. We successfully integrated our blockchain layer with Kaveri, the state registration system, at two distinct stages.

The first integration point occurred while mapping the property with the card. This process entails identifying the property within the Kaveri system and subsequently mapping it onto the blockchain. At this stage, the sub-registrar validates the property and its owner and maps them in the ledger.

The second integration point took place during the registration of property deeds. This multi-step process involves initiating the registration at Kaveri, verifying the registration data with the blockchain, and finalizing the registration within Kaveri. Once the registration at Kaveri is completed, the corresponding deed information is securely recorded in the blockchain.

A. Mapping of the property with the card

The integration process with the Kaveri layer begins when a user intends to transfer their property to the blockchain. The user visits the Sub Registrar's Office (SRO), where they undergo the e-KYC process. During this process, the Sub Registrar verifies the Aadhaar hash obtained from the e-KYC with the Aadhaar hash stored in the user's card. If there are multiple owners, the e-KYC and card details of all owners are selected.

Subsequently, the Sub Registrar selects the desired property and retrieves the property details and index data from the Kaveri system. At this stage, the Sub Registrar invokes the "mapUserWithProperty" chaincode on the blockchain, triggering a transaction from Kaveri to the blockchain signed by the SRO, ensuring its authenticity. The data is then transmitted to the blockchain for processing.

Upon receiving the transaction, the blockchain returns a unique blockchain property ID, which is then stored within the Kaveri system. This allows Kaveri to maintain a comprehensive record of the properties successfully linked to the blockchain. Similarly, Kaveri keeps track of properties that still need to be integrated.

B. Registration of the sales deed

The Integration of the Kaveri and the blockchain portal involved the following steps The integration of the Kaveri registration system with the blockchain portal for property registration follows a series of steps to ensure a seamless and reliable process. The integration process can be outlined as follows:

1) Identification of the Property by Kaveri:
   a) Kaveri first identifies the property based on the property number, including the current and previous numbers.
   b) It locates the transaction associated with the property and retrieves the latest transaction if multiple transactions are found.
   c) If no associated property ID is found, Kaveri sets the ID as empty.
2) Sending Property Details to the Blockchain Portal
   a) Kaveri sends the blockchain property ID, transaction type details, and index data to the blockchain portal for further processing.
3) Handling New Property Registration:
   a) If the blockchain property identifier is not present (empty), the blockchain portal considers it a request to register a new property.
   b) The blockchain portal performs a localized search using the current and previous property numbers.
   c) If a matching property is found, the Sub Registrar is notified and allowed to accept or reject the assertion.
   d) If the assertion is accepted, the same property ID is considered for the registration.
   e) If the assertion is rejected or no matching property is found, a new property is created.
4) Verification of Buyers and Seller:
   a) The blockchain portal collects the e-KYC and card data of all the buyers and sellers involved in the registration process.
   b) The transaction is signed using the private key stored in the card of the respective individuals.
   c) If it is not a new property registration, the verification chaincode is called to verify the property details provided by the buyers.
5) Confirmation and Data Appending:
   a) The blockchain system sends confirmation back to the Kaveri system, indicating successful property verification and capturing the identity details.
   b) The property registration process continues within Kaveri, including capturing thumb impressions, signatures, and photographs of the stakeholders.
   c) The registration deed is scanned and uploaded in Kaveri.
6) Appending Scanned Registration Details:
   a) Kaveri sends the scanned registration details back to the blockchain portal.
   b) The blockchain portal appends this data to the already collected e-KYC, card information, and property data.
7) Sub-Registrar Signing and Invocation of Registration Chaincode:

a) All the details, including the scanned registration data, are signed by the Sub-Registrar.
b) The registration chaincode is invoked to process the final registration of the property.

It is important to note that Kaveri serves as the software responsible for property registration, while the blockchain portal acts as an integration layer. The blockchain portal can only register a document if it has been registered in Kaveri. The final decision to accept or reject the application lies with Kaveri, even though the blockchain portal verifies the registration process. This integration ensures that the registration process remains consistent and reliable throughout the interaction between Kaveri and the blockchain portal. This makes the blockchain layers pluggable. The entire process of Kaveri is not affected, even if the blockchain portal is detached.

VI. Blockchain and Architecture

A. Organiztions

Hyperledger Fabric is a permissioned blockchain where the organization owns the network. Each organization has its own set of peers that maintain a copy of the distributed ledger and execute smart contracts. The organizations involved in our network include:

1) Centre for e-Governance (CEG): CEG is responsible for e-governance and maintains the IT infrastructure. It plays a crucial role in managing the chaincode, which contains the business logic of the network. CEG organization has 1 peer, 1 orderer, and 1 Certificate Authority (CA) for network operations.
2) Inspector General of Registration and Stamps (IGRS): IGRS maintains registration records and includes Sub Registrars who handle property registration. IGRS organization has 1 peer, 1 orderer, and 1 CA in the network.
3) Finance Department: The Finance Department is responsible for managing revenue records. It plays an important role in maintaining financial data within the network. The Finance Department has 1 peer, 1 orderer, and 1 CA in the network.
4) Audit Nodes: Audit nodes are responsible for conducting audits and detecting any collusion within the network. Although they do not have write permissions, they actively participate in the consensus process. We have selected two audit nodes, one serving as a far data recovery centre located outside the state. Each audit node has 1 peer and 1 CA.

We used raft as a consensus mechanism among the three orderers.

B. Users

Users The blockchain portal caters to different categories of users with specific roles and permissions. Here is a section outlining the user types and their corresponding functionalities within the portal:

- Admin: Every organization within the blockchain network will have one or more admin users. The admin user holds the responsibility of managing user registrations and enrollments. They are authorized to register new users into the portal and revoke access for existing users. However, admin users do not have write access to the chaincodes, specifically the card and property chaincodes. Their primary role revolves around user management and ensuring the smooth functioning of the portal.

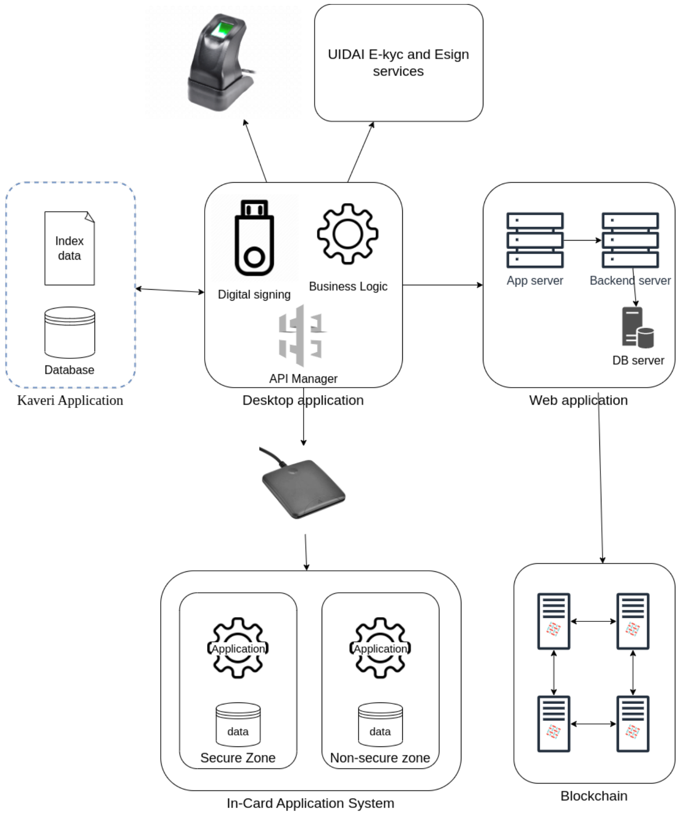

Fig. 3: Technical Architecture

- App User: App users are affiliated with each organization in the blockchain network and have access to the backend systems. They are granted read access to the portal, enabling them to retrieve information from the blockchain. The credentials of app users are securely stored within the backend system, allowing them to authenticate themselves and access the necessary data.
- SRO User: SRO (Sub-Registrar Office) users are associated with the IGR (Inspector General of Registration and Stamps) organization. They are granted read and write access to the card and property chaincodes. SRO users can perform operations related to both reading and updating information in the card and property chaincodes. Their role includes verifying property details, executing property registrations, and maintaining the integrity of the registration records.
- CEG User: CEG (Centre for e-Governance) users belong to the CEG (Centre for e-Governance) organization. They possess read and write access to the card chaincode. CSC users are responsible for performing operations related

to card management within the blockchain portal. This includes tasks such as creating new cards, updating card information, and managing the associated user data. Their access permissions are limited to the card chaincode, ensuring data integrity and security. The user structure within the blockchain portal is designed to accommodate each stakeholder's specific roles and responsibilities. With differentiated access levels and permissions, the portal ensures that users can efficiently carry out their designated tasks while maintaining data privacy and security

VII. Technical Architecture

Our technical architecture comprises a Hyperledger Fabric blockchain network consisting of four organizations. The architecture integrates the following components to facilitate the pluggable blockchain system.

1) **Kaveri (Existing Registration Application):** Kaveri serves as the existing registration application, forming an integral part of our architecture. Kaveri is the starting point of the transaction. A record is first fetched from the Kaveri and only added to the blockchain after the necessary verifications. Blockchain is only the pluggable layer, and Kaveri still offers registration services.
2) **Desktop Application:** A desktop application is incorporated to interact with essential hardware components such as the Java card and fingerprint reader.
3) **Frontend:** The frontend interface is designed to interact with users, providing an intuitive and user-friendly platform.
4) **Backend:** The backend serves as the connecting link between the frontend and the blockchain network. It processes user requests received from the frontend and communicates with the blockchain to initiate corresponding smart contract executions.
5) **Java Card Applet:** We designed the Java card applet to store the private keys securely. Java Card has two memory zones, secure and insecure zone. Private keys are stored in the secure zone and locked with the pin. The owner ID and the card number are stored in the insecure zone and can be read without the user's PIN.

Fig. 3 gives an overview of the technical architecture.

VIII. Conclusion and Future work

In this paper, we successfully deployed a Hyperledger Fabric-based integrity layer for property registration in Karnataka, addressing loopholes and ensuring data integrity. We focused on creating a unified and reliable property registration system by incorporating multiple departments, such as mutation, survey, and settlement, into the blockchain framework.

We developed a pluggable layer, integrating it with the Karnataka property registration system in 6 SRO offices and registering over 3000 property documents. However, future work involves extending the integration to encompass all states, establishing a comprehensive nationwide system. Additionally, we envision exploring the use of non-fungible tokens (NFTs) for secure property record trading within the blockchain framework, leveraging users' existing hardware wallets.

While Aadhaar served as the backbone for digital identity verification, we acknowledge the potential for a non-Aadhaar-based process. However, it may lack the same level of confidence and robustness.

The implementation of the blockchain-based integrity layer opens the possibility for online registration facilities with digital signatures, revolutionizing the registration process and enhancing convenience, efficiency, and accessibility for users. Through further development and integration, we aim to create a comprehensive and secure system that addresses all aspects of property registration, promotes transparency, and facilitates online registration processes.

References

[1] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolic, Sharon Weed Cocco, and Jason Yellick. Hyperledger fabric: A distributed operating system for permissioned blockchains. *CoRR*, abs/1801.10228, 2018.

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[3] Abdurrashid Ibrahim Sanka and Ray C.C. Cheung. A systematic review of blockchain scalability: Issues, solutions, analysis and future research. *Journal of Network and Computer Applications*, 195:103232, 2021.

[4] PRS Legislative Research. Land records and titles in india. https://prsindia.org/policy/analytical-reports/land-records-and-titles-india. Accessed: June 14, 2023.

[5] Digital India Land Records Modernization Programme. Computerization of land records. https://dilrmp.gov.in/faces/percent/rptComputerizationOfLandRecord.xhtml. Accessed: June 14, 2023.

[6] World Bank. Land policies for growth and poverty reduction. Technical report, 2003.

[7] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger.

[8] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, pages 173–186, 1999.

[9] Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm: Raft. In *USENIX Annual Technical Conference*, pages 305–320, 2014.

[10] Government of India. Uidai - unique identification authority of india. https://uidai.gov.in/en/. Accessed: June 14, 2023.

[11] Oracle Corporation. Java card. https://www.oracle.com/java/java-card/. Accessed: June 14, 2023.

[12] Government of India. National generic document registration system (ngdrs). https://ngdrs.gov.in/NGDRS_Website/. Accessed: June 14, 2023.

[13] Government of Karnataka. Kaveri online services. https://kaveri.karnataka.gov.in/. Accessed: June 14, 2023.

[14] J. Verhoeff. *Error Detecting Decimal Codes*. Number 29 in Mathematical Centre Tracts. Mathematisch Centrum, Amsterdam, 1969.