Design Document

University
of Houston
Clear Lake

Security and Privacy Modeling and Implementation with ALFA using Hyperledger Fabric Network

Instructors:

- **Dr. Kewei Sha**

Mentos:

- **Dr. Bun Yue,**
- **Dr.Wei Wei,**
- **Joses Selvan (Tietronix)**
- **Kayaanshoosh Collector (Tietronix)**

Team Members:

- **Venkata Satya Siddhartha Illa**
- **Farhana Begum Shaik**
- **Venkata Naga Bhaavagni Maddi**
- **Sripada Vallabh Kaparthi**
- **Ganesh Nyapuane**

Table of Contents

1. INTRODUCTION:

Blockchain started as the technology behind bitcoin but has popularly grown into a promising technology for cybersecurity. It can provide traceability, tamper resistance, transparency, enhanced security and automation. Most importantly, blockchains are being implemented with Model-Based System Engineering (MBSE) model where the system grants access to organizations or users to applications and assets based on the security and privacy permissions. The following paper provides an in-depth discussion of all the requirements needed by the application users and how they can access the resources or assets based on their required permissions.

2. SCOPE:

To implement an Attribute Based Access Control architecture for distributed blockchain technology using smart contracts provided by Hyperledger Fabric Network. Accurate and precise security attributes are translated to security and privacy policies using Abbreviated Language for Authorization(ALFA) to satisfy the approval processes for the Gateway MBSE Project. The two main areas that the project will focus on for security attributes will come from the Certificate Authority and the Member Service Provider and these security and privacy attributes are evaluated against the security policies to get access for the assets depending upon the required permissions. The blockchain will be accessed with the use of smart contracts and the blockchain will store and deploy the security and privacy policies. Finally, the user attributes and all other metadata is stored into a Postgres Database.

3. HYPERLEDGER FABRIC:

- **Hyperledger:** Hyperledger is a global collaboration, hosted by the Linux Foundation, including leaders in finance, banking, Internet of Things, supply chains, manufacturing and technology. It is an open-source collaborative effort created to advance cross-industry blockchain technologies. [7]
- **Hyperledger Fabric:** Hyperledger Fabric is intended as a foundation for developing applications or solutions with a modular architecture. Hyperledger Fabric allows components, such as consensus and membership services, to be plug-and-play. Its modular and versatile design satisfies a broad range of industry use cases. It offers a unique approach to consensus that enables performance at scale while preserving privacy. [7]
- Hyperledger Fabric has a ledger, uses smart contracts, and its system with participants manage the transactions. This Hyperledger Fabric is a permission based blockchain network that is set up by different organizations that have intended to set up a consortium. These organizations members enroll through Membership Service Provider (MSP) will take responsibility for participating in the network and setting up their peers. These peers include in the fabric architecture which takes time to configure with appropriate cryptographic tools like a Certificate Authority (CA) with additional information. The transaction invocation requests from consumers within the company are received by peers in the member organization. A Client can be specific on organization activities. For initiation of transaction invocation request Chain code is installed in peers. [3]

- **Hyperledger fabric Model:**

  There are many features in the Hyperledger fabric Model that fulfills the promise of the enterprise Blockchain. [7]

  - **Assets:** Assets enable to exchange of monetary value over the network. [7]
  - **Chaincode:** Chaincode partitioned from transaction ordering, limiting the confidence and verification levels required across types of nodes, and maximizing scalability and efficiency of the network. [7]
  - **Ledger:** For each channel, it encodes the whole transaction history and includes SQL like query capability privacy. [7]

- **Privacy:** Provides high privacy and confidentiality for channels and private data collections to enable multi-lateral transactions. [7]
- **Security & Membership services:** Participants in Permissioned Membership recognize that all transactions can be identified and tracked by approved regulators and auditors. [7]
- **Consensus:** Enable network starters to select a mechanism of consensus that best represents the relationships between participants. [7]

4. SYSTEM ARCHITECTURE:

4.1 ABAC:

- ABAC consists of three elements:
  - Attributes
  - Policies
  - Architecture
- ABAC controls access to objects by evaluating rules against the attributes of entities operations, and environment relevant to a request.
- ABAC enables precise access control allowing for many discrete inputs into an access control decision, providing a large set of possible combinations of those variables to reflect a diverse set of possible rules, policies, or restrictions on access. Thus, ABAC allows an unrestricted number of attributes to be combined to satisfy a rich set of policies.
- Within the ABAC, there are several functional "points" that are the service nide for retrieval and management of the policy, along with some logical components for handling the context or workflow of policy and attribute retrieval and assessment.
- There are four main functions in ABAC:
  - **Policy Enforcement Point (PEP)**: Enforces policy decisions in response to a request from a subject requesting access to a protected object; the access control decisions are made by the PDP.
  - **Policy Decision Point (PDP)**:
  - Computes access decisions by evaluating the applicable DPs and MPs. One of the main functions of the PDP is to mediate or deconflict DPs according to MPs.

- To compute access decisions, the PDP must have information about the attributes. This information is provided by the PIP.
- **Policy Information Point (PIP**): Serves as the retrieval source of attributes, or the data required for policy evaluation to provide the information needed by the PDP to make the decisions.
- **Policy Administration Point (PAP)**: Provides a user interface for creating, managing, testing, and debugging DPs and MPs, and storing these policies in the appropriate repository.

➢ **Attributes**:

- Subject: attributes that describe the user attempting the access.

  Example:

  - Name
  - Organization
  - Role
  - ID
  - Security clearance

- Resource: attributes that describe the resource being accessed.

  Example:

  - Which type of file user is asking
  - File name
  - Owner of the file
  - File creation date

- Action: attributes that describe the action being attempted.

  Example:

  - View
  - Edit
  - Update
  - Delete

- Environment: attributes that deal with time, location or dynamic aspects of the access control scenario.

  Example:

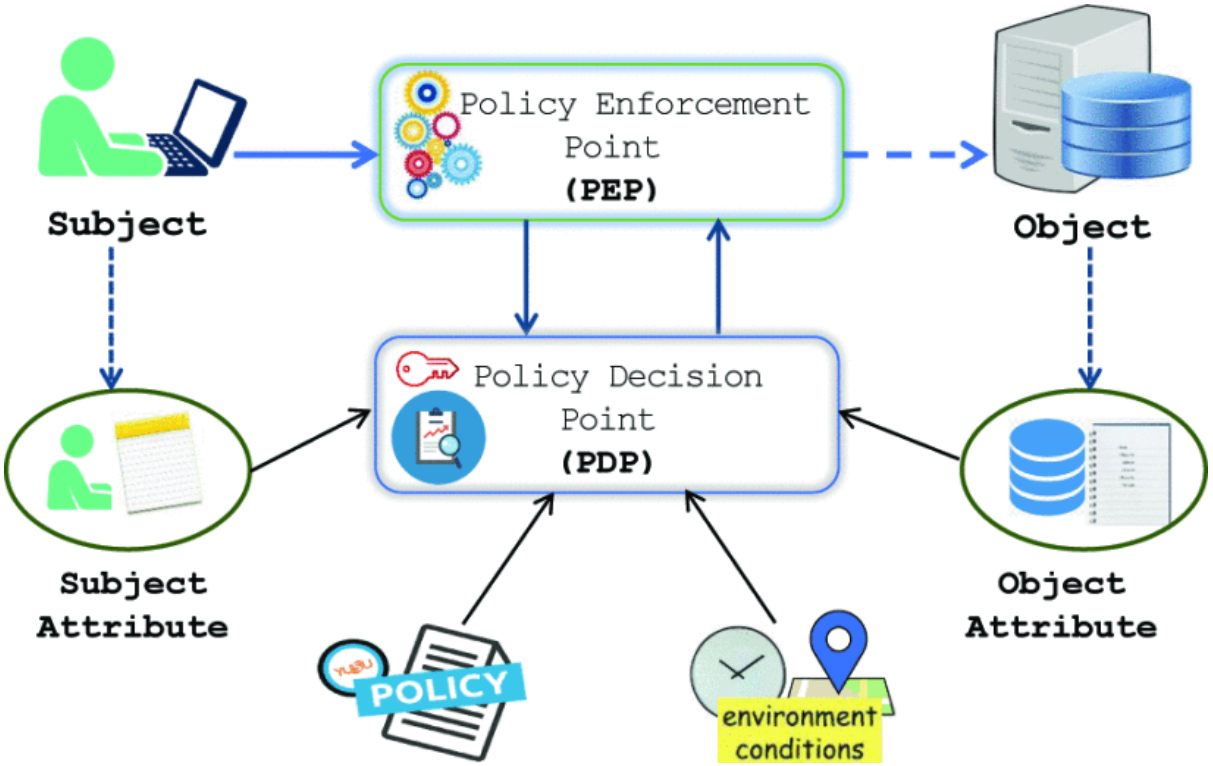
  - Location of the access (from where user is requesting the file)
  - Time of the access

- Date of the access

➢ **Policies**:

Policies are statements that bring together attributes to express what can happen and is not allowed. Policies in ABAC can be granting or denying policies.

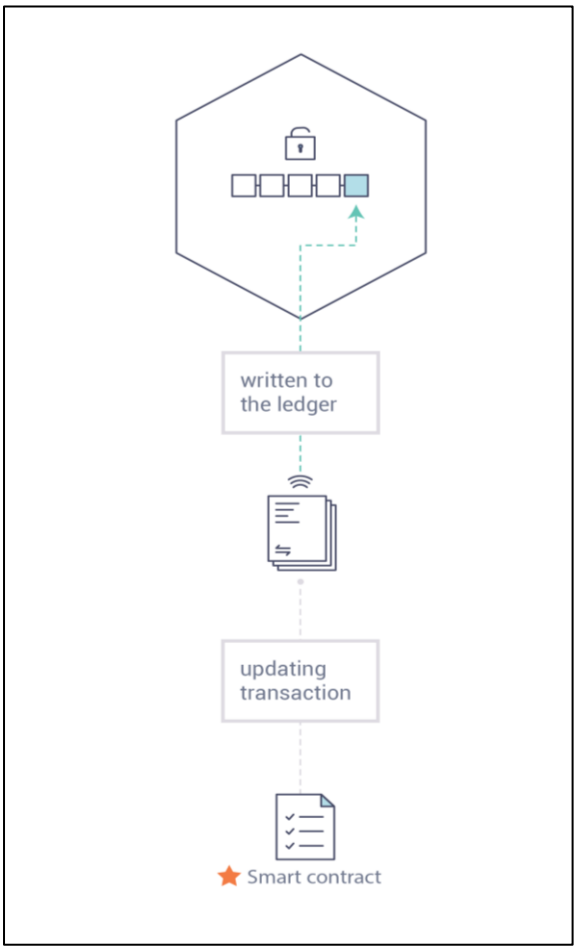
➢ **Architecture**:



4.2 SMART CONTRACTS:

➢ A From an application developer's perspective, a **smart contract**, together with the ledger, form the heart of a Hyperledger Fabric blockchain system. Whereas, a ledger holds facts about the current and historical state of a set of business objects, a smart contract defines the executable logic that generates new facts that are added to the ledger.

- Smart contracts are not only a key mechanism for encapsulating information and keeping it simple across the network, they can also be written to allow participants to execute certain aspects of transactions automatically.
- A smart contract defines the rules between different organizations in executable code. Applications invoke a smart contract to generate transactions that are recorded on the ledger.
- Smart Contract can implement governance rules for any business project. In general, a smart contract defines the **transaction logic** that controls the lifecycle of a business object contained in the world state. Multiple smart contracts can be defined within the same chaincode. It is then packaged into a chaincode which is then deployed to a blockchain network.
- A smart contract programmatically access two distinct pieces of ledger one is a blockchain which is immutably records the history of all transactions and the other one is a world state, stores the current value of these states.

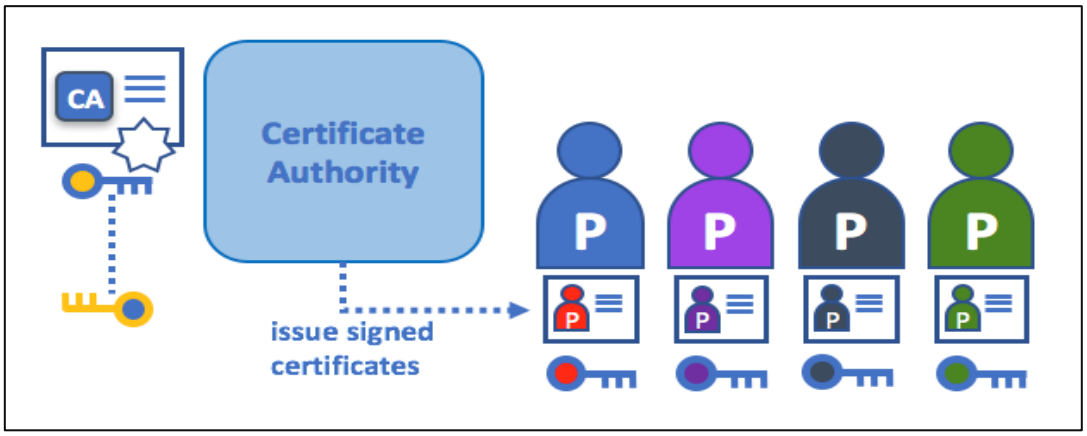


4.3 MSP:

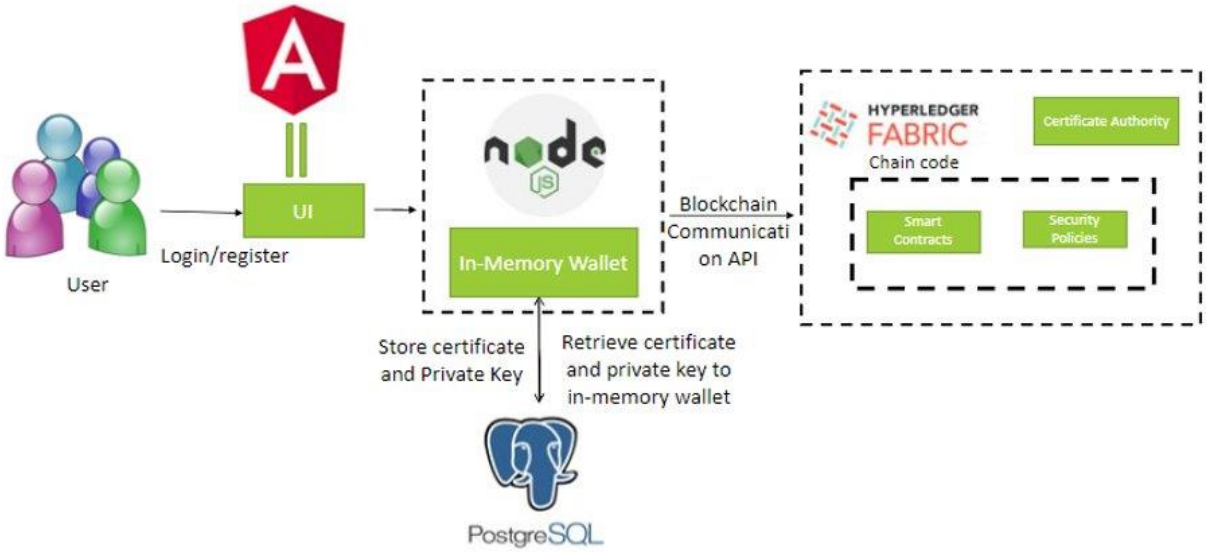
- The MSP on the ordering service contains the peer's public key which is then used to verify that the signature attached to the transaction is valid. The private key is used to produce a signature on a transaction that only the corresponding public key, that is part of an MSP, can match. Thus, the MSP is the mechanism that allows that identity to be trusted and recognized by the rest of the network without ever revealing the member's private key.
- Certificate Authorities generate the certificates that represent identities, the MSP contains a list of permissioned identities.
- The MSP identifies which Root CAs and intermediate CAs are accepted to define the members of a trust domain by listing the identities of their members, or by identifying which CAs are authorized to issue valid identities for their members.
- The power of an MSP goes beyond simply listing who is a network participant or member of a channel. It is the MSP that turns an identity into a role by identifying specific privileges an user has on node or channel.
- MSPs occur in two domains in a blockchain network:
    - Locally on an actor's node (**local MSP**)
    - In channel configuration (**channel MSP**)
- Local MSPs are only defined on the file system of the node or user to which they apply. Therefore, physically and logically there is only one local MSP per node. However, as channel MSPs are available to all nodes in the channel, they are logically defined once in the channel configuration.
- An organization is a logical managed group of members, what is important about organization is that they manage their members under a single MSP. The MSP allows an identity to be linked to an organization. An organization can also be divided into multiple organizational units, each of which has a certain set of responsibilities, also referred to as affiliations.
- There is a special kind of OU, referred to as a Node OU, that can be used to confer a role onto an identity. These Node OU roles are defined in the yaml file and contain a list of organizational units whose members are considered to part of the organization represented by MSP.

4.4 CERTIFICATE AUTHORITY:

- A user is able to participate in the blockchain network via the means of a digital identity issued for it by an authority trusted by the system. In the most common cases, digital identities have the form of cryptographically validated digital certificates that comply with X.509 standard and are issued by a Certificate Authority (CA).
- A certificate Authority dispenses certificates to different users. These certificates are digitally signed by the CA and bind together the user with the user's public key.
- One or more Cas can be used to define the members of an organization's from a digital perspective. It's the CA that provides the basis for an organization's actors to have a verifiable digital identity.
- It consists of fabric client CA and fabric server CA. Fabric CA server generates a root ca, which contains a private key and self-signed certificate, now the fabric ca server can act as a CA.
- A certificate can be a root CA or intermediate CA. Usually intermediate certificate has a parent CA having issued and signed intermediate CA. Intermediate CA can act as root CA or an another intermediate CA.
- Enrollment certificate authority allows the user to register with the blockchain network and enable the registered users to request an enrollment certificate pair. Transaction certificate from the transaction certificate authority are used for deploying the chaincode and for invoking chaincode transactions on the block chain.



4.5 DESIGN:

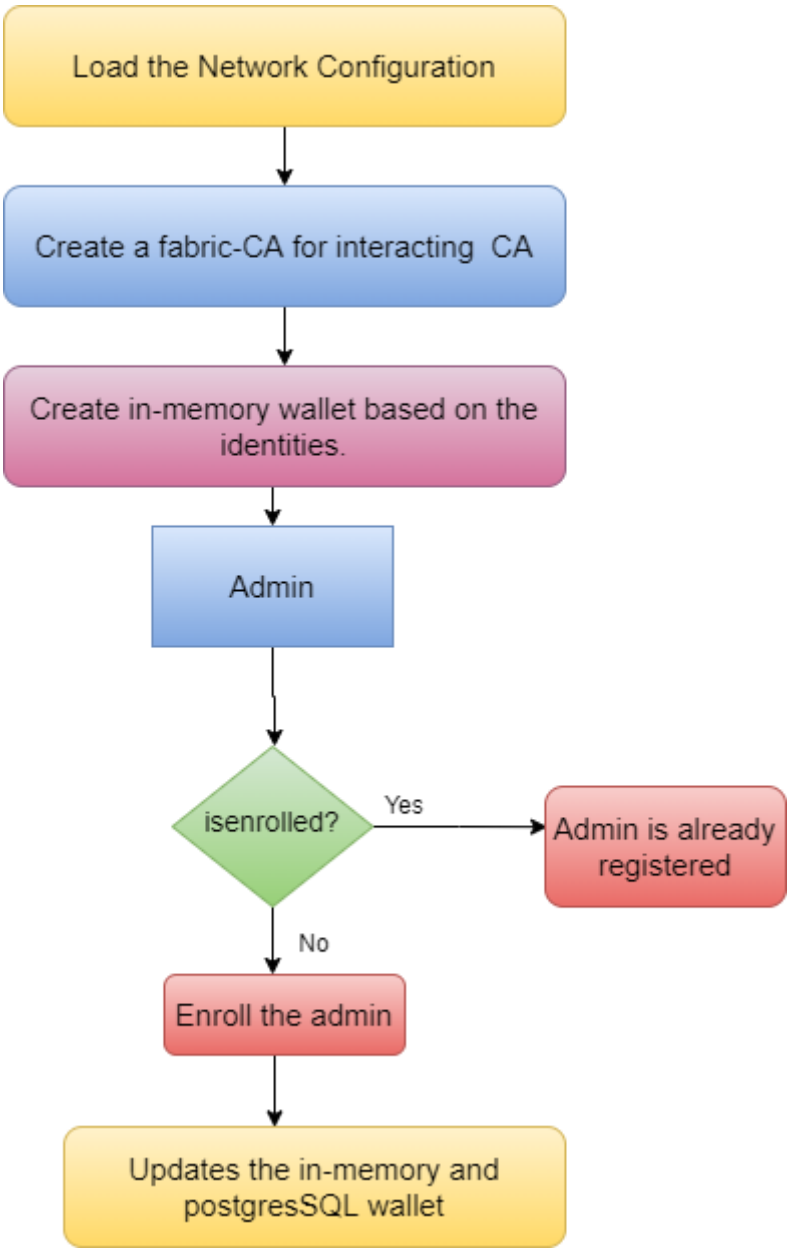


- The user will interact with the blockchain using blockchain API's.
- Blockchain Network is set up, channel is up and created and then Smart Contract(SC) is deployed on the network.
- The network configuration is loaded to create a new Fabric CA Client to interact with CA and new in-memory wallet is created for managing identities.
- If the admin is not there in the postgres wallet, then the admin is enrolled and the certificate and private key is updated into the in-memory wallet and postgres wallet.
- Else, if the admin is already there in the postgres wallet, then the system will throw an error that the admin is already registered.
- After the enrollment process, the user is registered using its credentials with the UI which stores the certificate and private key of user in the postgres database.
- A new in-memory wallet is created which then retrieves the certificate and private key from the postgres database for authentication of user objects with the CA and later updates the in-memory wallet and postgres wallet.
- After the user is enrolled and registered, the smart contract is invoked by the user and then the gateway connection is established for the peer node and channel is obtained where the smart contract is deployed.
- The smart contracts provided by the Hyperledger Fabric evaluate the security and privacy polices against the user security attributes and depending upon the user's

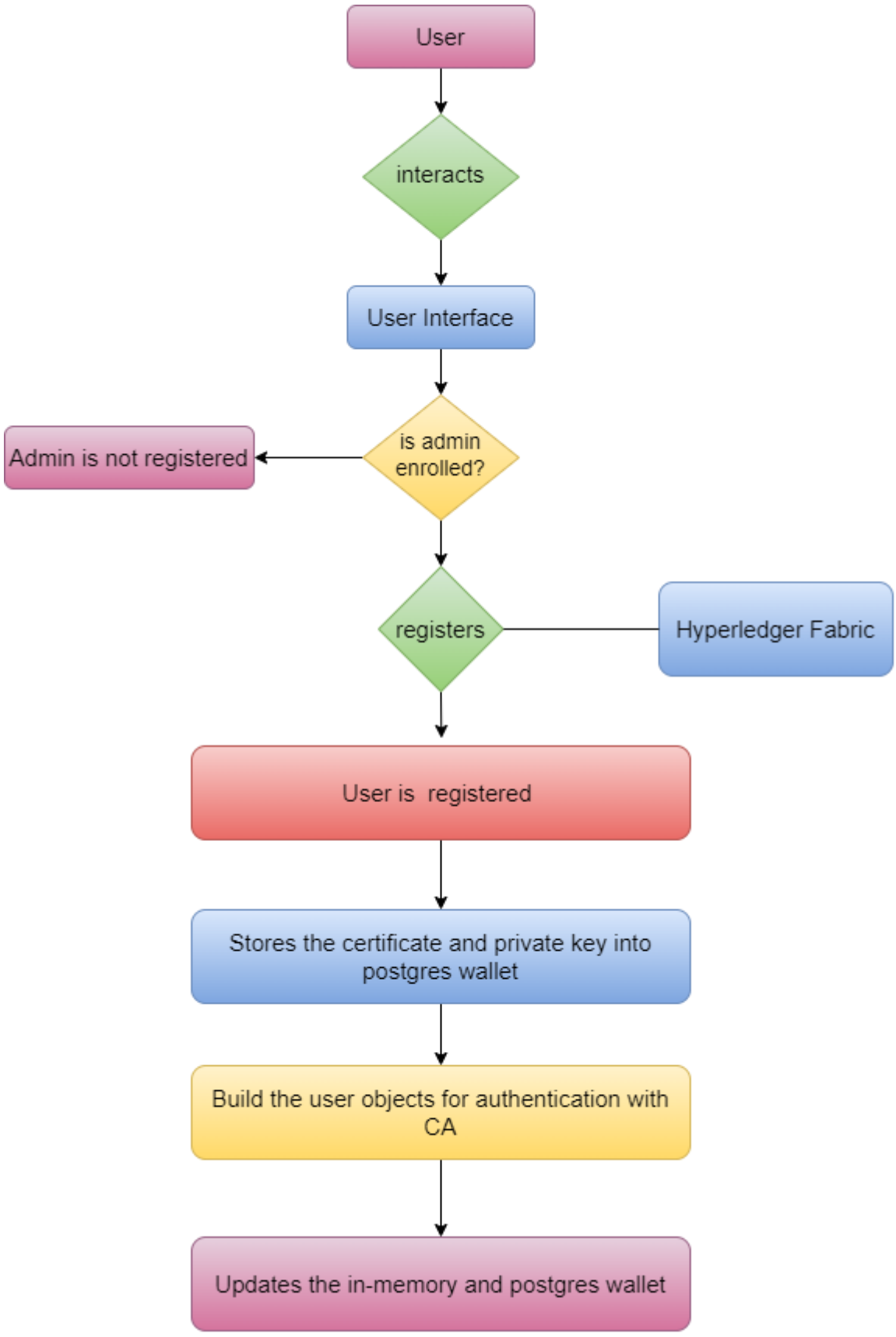
role and level of access the CRUD operations are performed by the user which thus updates or creates a new asset and stores it into the world-state ledger.

5. FLOW CHARTS

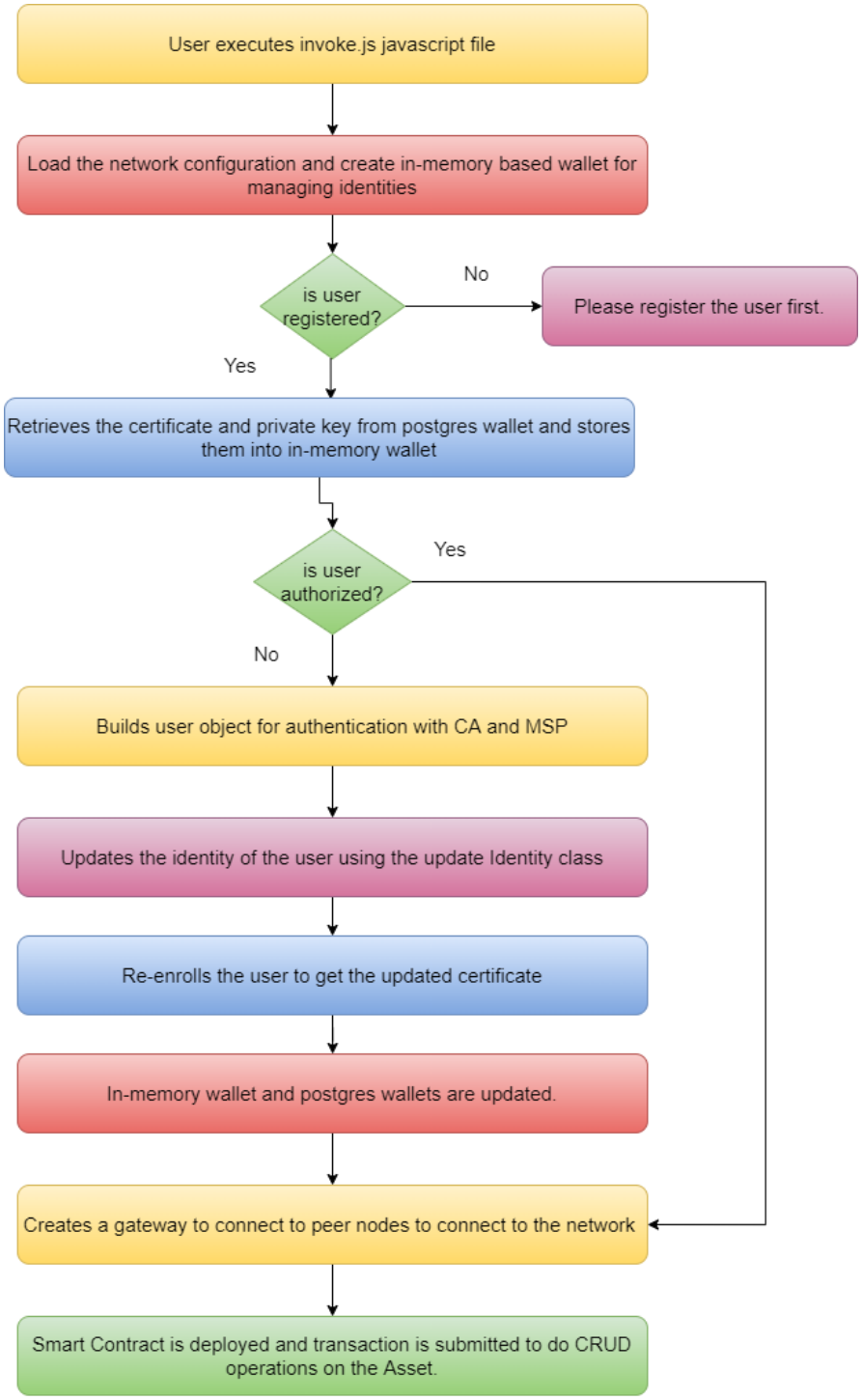
5.1 ENROLL PROCESS:



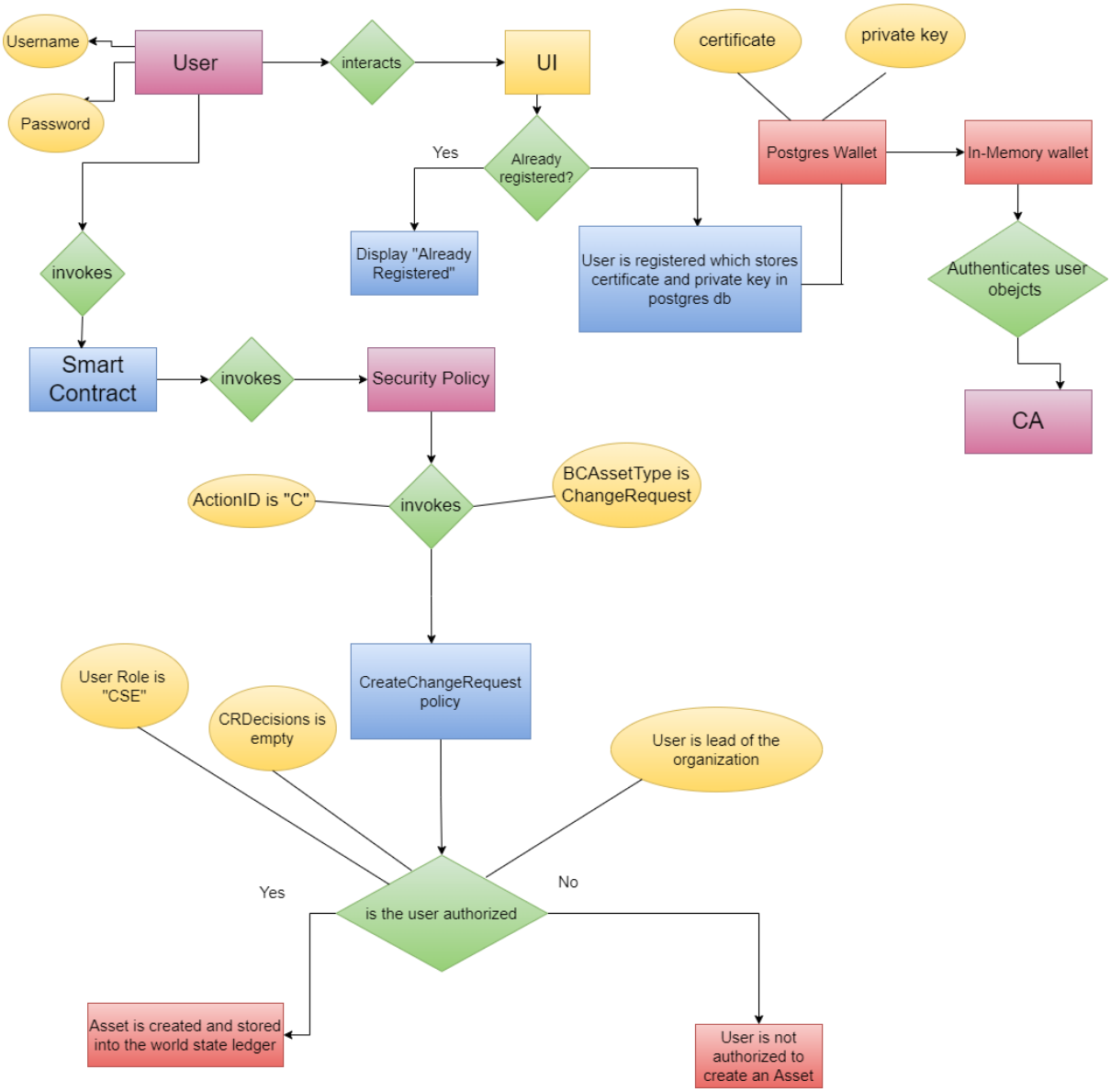
5.2 REGISTER PROCESS:



5.3 UPDATE IDENTITY PROCESS



5.4 FLOWCHART FOR CREATE CHANGE REQUEST SCENARIO



6. REQUIREMENTS:

6.1 FUCTIONAL REQUIREMENTS:

- Write security and privacy policies into an attribute-based rule or rules for the approval and collaboration processes using ALFA.
- Implement the blockchain using the Hyperledger Fabric Network.
- Write smart contracts in Golang.
- Updating and adding new attributes using the Smart Contracts
- Storing and Accessing the credentials and attributes to the external database(PostgreSQL).
- Permitting Application user access depending upon the role to the world state using appropriate smart contracts.
- Storing identities in in-memory wallet and accessing them using postgres database.
- Implement Member Service Provider (MSP)
    - Define organizations that are to be trusted by the Fabric network.
    - Create Local MSP's for peers of the organizations.
    - Creates Global MSP's for the network and organizations.
    - Define and assign roles and permissions to members.
    - Stores attributes of the peers, channels, network using the database and define the policies for the membership to the network.
    - Enable entities to access permissioned blockchain.
- Implement Certificate Authority (CA)
    - Attributes are stored and retrieved.
    - Initialize Fabric-CA Server.
    - Generates Certificates for the members, peers to the network using respective organization's CA's.
    - Generate Root CAs to represent identities.
    - Storing Certificates to the PostgreSQL database.
    - Generate Certificate Revocation List (CRL).

6.2 HARDWARE REQUIREMENTS:

- Operating System - Ubuntu 20.04.1 and Windows 64 bit..
- Processor - 2 GHz dual core
- System Memory - 4 GB Ram
- Hard Drive Space - 50 GB
- Graphics - VGA capable of 1024×728 resolution
- USB or CD/DVD Drive - At least one available for installer media
- Internet Adapter - Wired or wireless network

6.3 SOFTWARE REQUIREMENTS:

- Linux Debian 10 64 bit - 16 Gb RAM
- Ubuntu 20.04.1 64 bit - 16 Gb RAM
- Golang
  - Open source programming language from Google.
  - Smart contracts will be written in this language.
- NodeJS
  - Javascript language used for writing the network applications.
  - It is used for server-side programming, and primarily deployed for non-blocking, event-driven servers.
- ALFA
  - Abbreviated Language For Authorization.
  - Programming language used to write access-control policies.
  - Easily readable .
- curl
  - Open source software used for transferring of data in a CLI.
- Docker
  - Open platform for developing and running applications.

- Used for separating applications from the infrastructure.
- Daemon will help manage docker requests and containers.

➢ Angular 13:

- Angular is an application design framework and development platform for creating single-page apps.
- AngularJS extends HTML attributes with Directives and binds data to HTML with Expressions.
- It changes the static HTML to dynamic HTML.

➢ PostgreSQL 13**:**

- PostgreSQL is a free and open-source relational database management system emphasizing extensibility and SQL compliance
- PostgreSQL is used as the primary data store or data warehouse for many webs, mobile, geospatial, and analytics applications.

7. REFERENCES:

- [1] V. Aleksieva, H. Valchanov and A. Huliyan, "Implementation of Smart-Contract, Based on Hyperledger Fabric Blockchain," 2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA), 2020, pp. 1-4, doi: 10.1109/SIELA49118.2020.9167043.
- [2] ABAC Architecture - https://en.wikipedia.org/wiki/XACML#/media/File:XACML_Architecture_&_Flow.png
- [3] ALFA/XACML - https://en.wikipedia.org/wiki/ALFA_(XACML)
- [4] T. Luong, D. Vo and N. Truong, "An approach to analyze software security requirements in ABAC model," 2019 6th NAFOSTED Conference on Information and Computer Science (NICS), 2019, pp. 184-189, doi: 10.1109/NICS48868.2019.9023902.
- [5] X. Zeng, N. Hao, J. Zheng and X. Xu, "A consortium blockchain paradigm on hyperledger-based peer-to-peer lending system," in China Communications, vol. 16, no. 8, pp. 38-50, Aug. 2019, doi: 10.23919/JCC.2019.08.004.
- [7] Hyperledger Fabric Official Documentation: https://hyperledger-fabric-ca.readthedocs.io/en/latest/