Requirement Analysis Document

University
of Houston
Clear Lake

Security and Privacy Modeling and Implementation with XACML/ALFA and Fabric

Instructors:

- **Dr. Kewei Sha**

Mentos:

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

Team Members:

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

Table of Content

1. Introduction

Blockchain:

A blockchain is a distributed database that is shared among the nodes of a computer network. As a database, a blockchain stores information electronically in digital format. Blockchains are best known for their crucial role in cryptocurrency systems, such as Bitcoin, for maintaining a secure and decentralized record of transactions. Blockchain is Programmable, Secure, Anonymous, Unanimous, Distributed, Immutable and Time-Stamped. Blockchains are being implemented with Model-Based System Engineering (MBSE). It grants access to the organizations or users to applications and resources based on the permissions. MBSE applications call for the use of permissioned blockchains in which access is based on permissions granted to organizations and users.

Benefits of Blockchain:

With the Blockchain, business processes will be better protected with the help of high-level security.

Any Organization will be able to do faster transactions with the help of Blockchain.

Every transaction is transparent and easy to track.

Whatever that occurs on the Blockchain is a function of the entire network.

2. Scope of Project

Scope of the project is to understand the Approval Process Collaboration Diagram, identify the attributes and write the security and privacy policies. ABAC policies are developed using Abbreviated Language for Authorization (ALFA), then converted to XACML and accessed from Smart Contracts/ChainCode using GOLANG. Initially all the security and privacy policies are written in plain English which will satisfy the approval and collaboration process for the Gateway MBSE project. Once security policies are translated into ALFA, Smart Contracts will use the ALFA policies to execute any transaction on ledger for the purpose of approval. Certificate Authority and Membership Service Provider are the other two things which this project concentrates on. CA and MSP add the required attributes that are identified, stored and retrieved respectively from the database. Blockchain Network will be accessed with the use of smart contracts and the Blockchain will store and deploy the security and privacy policies.

3. Prerequisite Definitions and Abbreviations

Channel: A channel is a private blockchain overlay which allows for data isolation and confidentiality. A channel-specific ledger is shared across the peers in the channel, and transacting parties must be authenticated to a channel in order to interact with it.

Consensus: Consensus is which serves to generate an agreement on the order and to confirm the correctness of the set of transactions constituting a block.

Consortium: A consortium is a collection of non-orderer organizations on the blockchain network. These are the organizations that form and join channels and that own peers. While a blockchain network can have multiple consortia, most blockchain networks have a single consortium.

Chaincode: A chaincode definition is used by organizations to agree on the parameters of a chaincode before it can be used on a channel. Each channel member that wants to use the chaincode to endorse transactions or query the ledger needs to approve a chaincode definition for their organization. Once enough channel members have approved a chaincode definition to meet the Lifecycle Endorsement policy (which is set to a majority of organizations in the channel by default), the chaincode definition can be committed to the channel.

Endorsement: Refers to the process where specific peer nodes execute a chaincode transaction and return a proposal response to the client application. The proposal response includes the chaincode execution response message, results (read set and write set), and events, as well as a signature to serve as proof of the peer's chaincode execution.

Invoke: Invoke is used to call chaincode functions. Generally, a client application invokes chaincode by sending a transactional proposal to a peer. The peer will execute the chaincode and return an endorsed proposal response to the client application. The client application will gather enough proposal responses to satisfy an endorsement policy, and will then submit the transaction results for ordering, validation, and commit.

Ledger: A ledger consists of two distinct, though related, parts – a "blockchain" and the "state database", also known as "world state". Unlike other ledgers, blockchains are immutable – that is, once a block has been added to the chain, it cannot be changed. In contrast, the "world state" is a database containing the current value of the set of key-value pairs that have been added, modified or deleted by the set of validated andcommitted transactions in the blockchain.

Peer: A network entity that maintains a ledger and runs chaincode containers in order toperform read/write operations to the ledger. Peers are owned and maintained by members.

Smart Contract: A smart contract is code – invoked by a client application external to theblockchain network – that manages access and modifications to a set of key-value pairsin the World State via Transaction. In Hyperledger Fabric, smart contracts are packagedas chaincode. Chaincode is installed on peers and then defined and used on one or morechannels.

Transaction: Transactions are created when a chaincode is invoked from a client application to read or write data from the ledger. Fabric application clients submit transaction proposals to endorsing peers for execution and endorsement, gather the signed (endorsed) responses from those endorsing peers, and then package the resultsand endorsements into a transaction that is submitted to the ordering service.

Abbreviations:

ABAC – Attribute Based Access Control
ALFA – Abbreviated Language for Authorization
CA – Certificate Authority
CC – Chain Code
CR – Change Request
DCR – Draft Change Request
CISE – Chief Integration System Engineer
MBSE – Model Based System Engineer
SE – Smart Contract
MSP – Member Service Provider
VCN – Verification Closure Notice
XACML – eXtensible Access Control Markup Language

4. Hyper Ledger Fabric

Hyperledger Fabric is an open-source enterprise-grade permissioned distributed ledger technology (DLT) platform, designed for use in enterprise contexts, that delivers some key differentiating capabilities over other popular distributed ledger or blockchain platforms. [2]

Fabric has a highly **modular** and **configurable** architecture, enabling innovation, versatility and optimization for a broad range of industry use cases including banking, finance, insurance, healthcare, human resources, supply chain and even digital music delivery. Fabric is the first distributed ledger platform to support **smart contracts authored in general-purpose programming languages** such as Java, Go and Node.js,rather than constrained domain-specific languages (DSL). This means that most enterprises already have the skill set needed to develop smart contracts, and no additional training to learn a new language or DSL is needed. [2]

Advantages of Hyperledger Fabric:

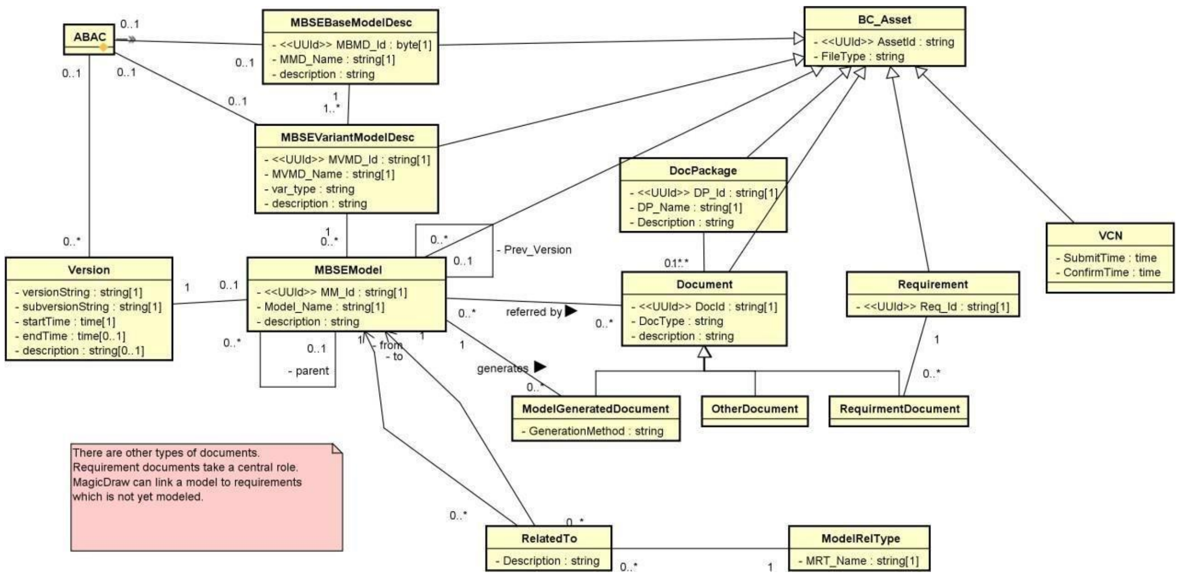
1. Permissioned Membership

2. Performance, Scalability and Level of trust

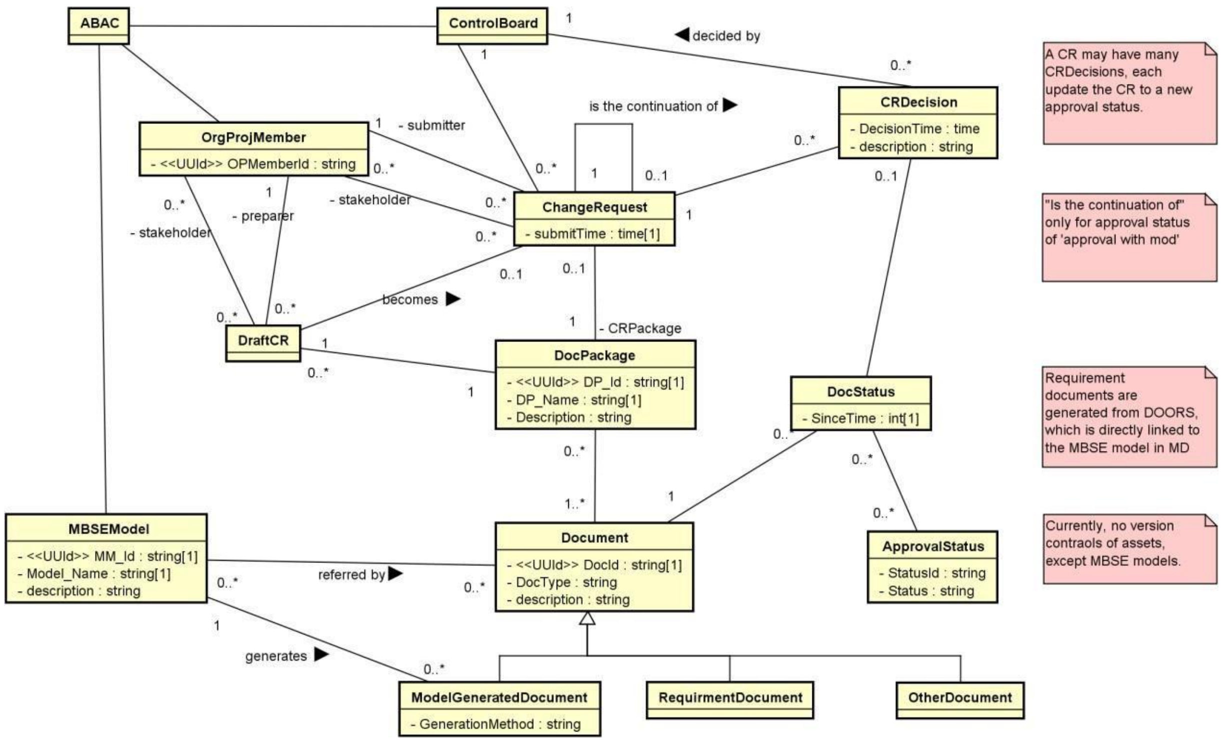
3. Modular Architecture supporting plug-in components.

5. USE CASE DIAGRAMS

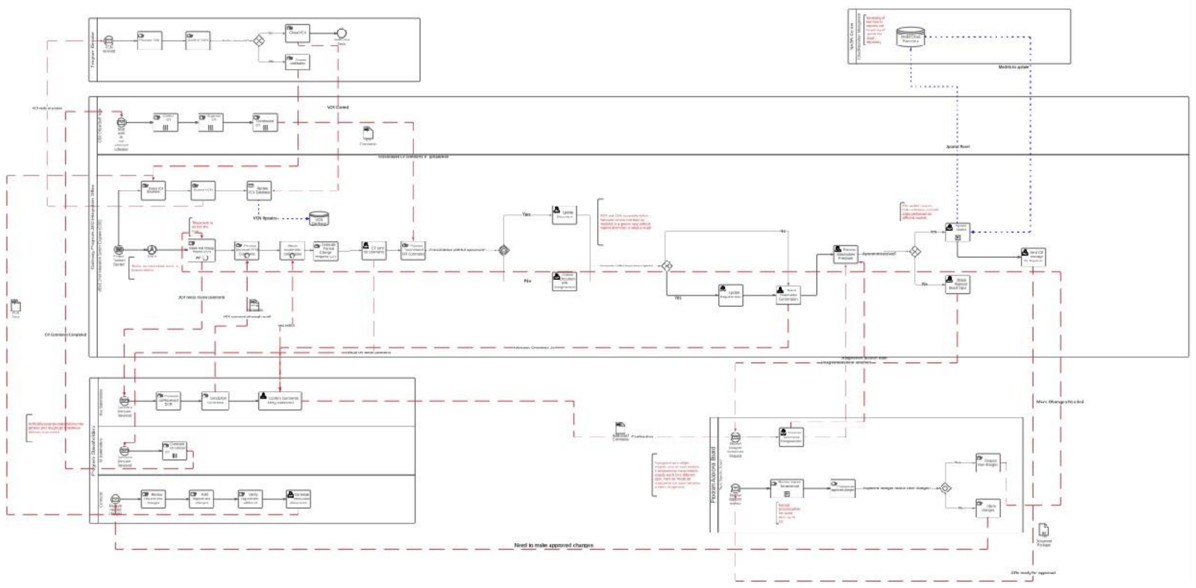
5.1. Top Level



5.2. Approval Process:



6. Brief Explanation of Collaboration and Approval Process



Explanation:

Participants:

- Gateway Program Director
- Gateway Program JSC Integration Office
- Gateway Program Stakeholders
- Gateway Program Approval Board
- NASA Cloud

Initially MBSE Chief Integration System Engineer (CISE) initiates a Draft copy of the change request and sends an email to Key Stakeholders requesting to verify and write comments on it. After receiving the comments through the email, change request process the received DCR comments and send to obtain stakeholder confirmation. Obtaining stakeholder confirmation is an iterative process until the process completes successfully. Next it is sent to create a Formal Change Request which is stored in Blockchain. Then Change Request sent for comments for All Stakeholders (Program Stakeholder). It receives the comments and make it official and sent to CISE office staff team where they collect, organize, and consolidate the Change Request and send to process consolidated CR comments.

If every stakeholder and CISE staff team fully agree with the CR, then it will update the document to prepare for submission of approval, else it updates the document with disagreement notice. Next if there is no further update, then it is sent to solicit stakeholder confirmation, else it updates requirements and sends. Then the change request is sent to the Process Stakeholder. If agreement is yes here, then it needs to update the model and send a package for approval. Here the Change Request is ready for approval and sent to the ProgramApproval Board.

Now in the Program approval board, once it receives the requests, it reviews and communicatesapproved changes. If it needs more changes the process is repeated for new change requests otherwise it informs changes and sent to Contractors to review, fulfill, verify and generate verification documents and send them to VCN (Verification Confirmation Notes). Here for every requirement program which relies on the contractor, they must come back and show proof that they met the requirement only then VCN is approved.

Next, Submitted VCN is sent to the Program Director. Here it processes and confirms VCN. If it is successful, then it performs the VCN close and the process ends. If it is unsuccessful, it requests for modification and repeats the steps again till the VCN is closed. After submitting theVCN, it updates in the VCN Local Database.

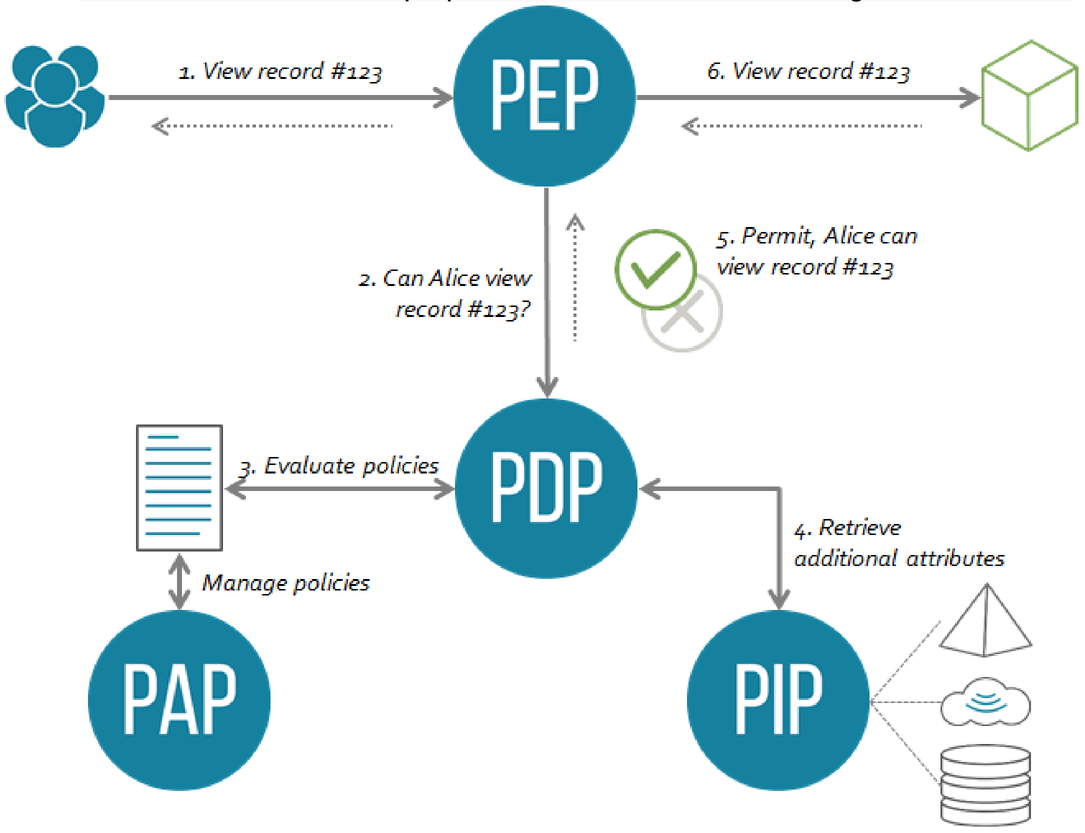
7. Certificate Authority

The Certificate Authority (CA) provides a number of certificate services to users of a blockchain. More specifically, these services relate to user enrollment, transactions invoked on the blockchain, and TLS-secure connections between users or componentsof the blockchain. [2]

- ➢ **Enrollment Certificate Authority:** The enrollment certificate authority (ECA) allows new users to register with the blockchain network and enables registered users to request an enrollment certificate pair. One certificate is for data signing, one is for data encryption.
- ➢ **Transaction Certificate Authority:** Once a user is enrolled, he or she can also request transaction certificates from the transaction certificate authority (TCA). These certificates are to be used for deploying Chaincode and for invokingChaincode transactions on the blockchain.
- ➢ **TLS Certificate Authority:** In addition to enrollment certificates and transaction certificates, users will need TLS certificates to secure their communication channels. TLS certificates can be requested from the TLS certificate authority (TLSCA).

8. Attribute Based Access Control

Attribute Based Access Control has been developed on attributes, policies and architecture. Below is the architecture proposed for ABAC and exact design ofUML Model:


1. View record #123
PEP
6. View record #123
2. Can Alice view record #123?
5. Permit, Alice can view record #123
3. Evaluate policies
PDP
4. Retrieve additional attributes
Manage policies
PAP
PIP

The architecture is build on four different components:

1. **Policy Enforcement Point**: PEP is capable of intercepting requests for entry,initiating the decision process and executing the relevant outcome and by enabling or refusing the access execution. [1]
2. **Policy Administration Point**: PAP is the element that controls the policy of access authorization. It provides a user interface for creating, managing, testingpolicies in appropriate repositories. [1]
3. **Policy Information Point:** PIP is for addition, different Attribute Managerscontrol the set of characteristics needed for the policy assessment. [1]
4. **Policy Decision Point:** PDP is the validation mechanism that retrieves a policywritten by resource owner, an entry request and tests the proposal and sends back the decision whether it is allowed or denied. [1]

➢ **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

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

- **a.** A user can view the document if the document is in the same department as theuser.
- **b.** A user can edit a document if they are the owner and if the document is in draftmode.

9. Sample Smart Contract:

```
// SmartContract provides functions for managing a book
type SmartContract struct {
    contractapi.Contract
}

// Baic
type Book struct {
    Organisation   string `json:"organisation"`
    Author  string `json:"author"`
    Title string `json:"title"`
    Owner  string `json:"owner"`
}

// QueryResult structure used for handling result of query
type QueryResult struct {
    Key    string `json:"Key"`
    Record *Book
}
/ CreateBook adds a new book to the world state with given details
func (s *SmartContract) CreateBook(ctx contractapi.TransactionContextInterface,
owner string, organisation string, author string, title string ) error {
    book := Book{
        Organisation:   organisation,
        Author:  author,
        Title: title,
        Owner:  owner,
    }

    bookAsBytes, _ := json.Marshal(book)

    return ctx.GetStub().PutState(owner, bookAsBytes)
}
```

10.REQUIREMENTS

A. FUNCTIONAL REQUIREMENTS:

- ❖ Hyperledger Fabric is the framework used to implement the Blockchain.
- ❖ Write security and privacy policies for approval and collaboration processin plain English.
- ❖ Smart Contracts are written in GOLANG because it is easy to maintain inthe long run.
- ❖ Analyzing and converting ABAC attributes using ALFA to accurate security policies.
- ❖ Implementing Security policies to the Block Chain applications directly using Smart Contracts.
- ❖ Update and adding new attributes using the Smart Contracts.
- ❖ Storing and Accessing the credentials and attributes to the external database(PostgresSQL).
- ❖ Permitting Application user access depending upon the role to the world state using appropriate smart contracts.
- ❖ Implement Certificate Authority (CA)
  - ➢ Attributes are stored and retrieved.
  - ➢ Initialize Fabric-CA Server.
  - ➢ Generates Certificates for the members, peers to the network usingrespective organization's CA's.
  - ➢ Storing Certificates to the PostgresSQL database.
  - ➢ Generate Certificate Revocation List (CRL).
- ❖ Implement Member Service Provider (MSP)
  - ➢ Define organizations that are trusted by Fabric network.
  - ➢ Create Local MSP's for peers of the organizations.
  - ➢ Creates Global MSP's for the network and organizations.
  - ➢ Stores attributes of the peers, channels, network using thedatabase and define the policies for the membership to thenetwork.
  - ➢ Enable entities to access permissioned Blockchain.
- ❖ Implement Attribute Based Access Control (ABAC) using ALFA.
  - ➢ Design a Policy Enforcement Point (PEP)
  - ➢ Design a Policy Decision Point (PDP)
  - ➢ Design a PolicyInformation Point (PIP)
  - ➢ Design a Policy Administrative Point (PAP).
  - ➢ Identifying Precise ABAC attributes for Block Chain Application.
  - ➢ Attributes are stored and retrieved.

B. Software Requirements:

- Ubuntu 20.04.1.64 - 16 Gb RAM
- Linux 10 64 bit - 16 Gb RAM
- GOLANG - Open Source programming Language from Google.
- ALFA (Abbreviated Language for Authorization) - Programming Languageused to write access control policies.
- XACML (eXtensible Access Control Markup Language)
  - Programming Language used for attribute based access- controlpolicies.
- curl - Open source software used for transferring data in a CLI.
- NodeJS - Javascript Language used for writing the network applications.
- Docker:
  - Open platform for developing and running applications.
  - Used for separating applications from the infrastructure.

C. Hardware Requirements:

Operating System	- Ubuntu 20.04.1 and 64.
Processor	- 2GHz dual core
System Memory	- 12 GB RAM
Hard Drive Space	- 50 GB
Graphics	- VGA capable of 1024x728 resolution
USB or CD/ DVD Drive	- At Least one available for installer media
Internet Adapter	- Wired or Wireless Network.

11. REFERENCES

[1]. R. Guo, H. Shi, Q. Zhao, D. Zheng, “Secure AttributeBased Signature Scheme withMultiple Authorities for Blockchain in Electronic Health Records Systems”, IEEE Access, vol. 6, pp. 11676-11686, 2018.

[2]. https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html#hyperledger-

fabric [3]. https://doubleoctopus.com/security-wiki/digital-certificates/certificate-

authority/