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

Meeting Date: April 1st, 2022.

Start Time: 10:30 AM

End Time: 12:00 PM

Attendees: Dr. Yue, Siddhartha Illa, Venkata Naga Bhaavagni Maddi, Farhana Shaik Begum, Sripada Vallabh Kaparthi, Ganesh Nyaupane, Preethi Vuchuru, Madhuri Koduru

Prepared By: Venkata Naga Bhaavagni Maddi, Sripada Vallabh Kaparthi, Siddhartha Illa

---

Items Discussed:

1) Bhaavagni and Farhana (Working on implementing Change request policy)

Implementing the policy based on the JSON Schema.

Dr. Yue

A) Did you change the JSON Schema? We have created a structure in Smart Contracts, storing JSON attributes in the form of a structure. Where is Marshalling and Unmarshalling done?

Explained the code that has been implemented so far. So, if the JSON Schema has BCTransaction as a key, you need to implement it as a struct. Gave review on the struct code that has been created. You need not include payload, the top level structure is BCAsset. Have a struct for CRDecision. Implementing the entire JSON Schema to a struct would be difficult. Try including the details which are relevant to the policy. Few attributes which are relevant to the Change request policy are FromCRId, CRDecisions, CRComments. The CRDecisions would be initially an empty array in the blockchain. You need to define the asset type for CR and the project. The CR may contain the document package. You try to include atleast project. Include isTopLevel attribute. For CRDecisions, try including the submission time, AssetId, AssetType, ProjectId.

Dr. Sha

A) Why the names in JSON Schema are differed in your struct implementation? We have implemented a basic struct that is representing the JSON Schema. Try focusing on the ABAC related and CR related data while defining the struct.

2) Sripada : Presented storing the certificate content in the postgreSQL database.

Dr.Yue:

A) The idea of storing a certificate in local database is for the use of the individual organization. So the local organization once it authenticate the particular user then it will allow the user to submit a call to smart contract. When you invoke a SC you need to submit a certificate and therefore a certificate is always stored in a local organization.

Read the article provided. You need to store the type of the certificate as well as user information so that whenever you invoke the smart contract you can go and look up for a right certificate. You need to have map to local organization user. For example, if there are two different users named creator1 and creator 2 then you need to define the Fab User. When ever they invoke a SC, you look up at the right fab user id and right certificate and submit it to the Smart Contract.

Summary of Dr.Yue and Dr. Sha discussion :

Original focus is storing individual user certificate. Individual user certificate will be managed by CA who need to provide the authentication. Idea is that people like to integrate certificate which is

managed by CA and we would like to somehow look up to postgres so they can link with local organization data and authentication.

Dr.Sha: According to some research done in the last semester, certificated managed by MSP is actually referred to the file.

Dr.Yue: By demonstrating the architecture diagram from the article – Basically we need to have database stored in the wallet so that we can manage it when we submit to a Smart Contract. We do store some certificates in the postgres, and client try to get the certificate and submit. But the certificates will be the user certificates.

3) Siddhartha (Working on managing the CA Identities) :

Presented on accessing the attributes inside the chaincode using the cid package to the team which will be used to write security policies for the smart contracts. Also added the mcba package which will be used to maintain all the security policies in order to access them from the Smart contracts to evaluate the invoke requests to accept/reject based on the attribute values of the invoking identity.

Dr. Yue : gave a clarification that the contractapi is referring to the cid package internally and the getAttributeValue can be used from the ctx object part of the contractapi.

Items for next week:

1) Developing a Struct type representing the JSON Schema to implement the Change request policy.
2) Implementation of Alfa Policies in the Smart Contract.
3) Adding attributes using Smart Contract.
4) Retrieving the certificates from database.