**FIDO Server changes required for Enterprise Policy Management Website**

I believe the Enterprise Policy Management Website will be the first tool the user will log into, to set policies for himself (basically to choose what authenticators are supported for his grade and which all he wants to use out of them).

User credentials, employee code and grade will be pre-populated in the DB and will be fetched based on user’s name. (Username should be a unique email address of the user, which will be acting as a PRIMARY KEY of that particular table in DB.)

1. When user logs into Enterprise Policy Management Website (credentials will be checked against “userDetails” table), he will be taken to a form with his username, employee ID and grade already displayed on the form (will be fetched from the same table using the “username” primary key). In this form user just have to select the enabled Authenticators (single/multiple) and then click on “submit”
2. On submit, user will be taken to a new page which will summarize the details he has given. This will again display the username, grade, employee id and the authenticator/list of authenticators that he/she has chosen, so that the user can verify and confirm the data. Once the user confirms the data, he will click on “Make Policy” button. The “Go Back” button will take you to the previous page, which will allow user to edit the authenticators.
3. Once user clicks on “Make Policy” button, an internal FIDO server API will be called to create Policy for the user. I’m thinking of taking the list of authenticators as input to the API, fetch the details of each authenticators from the “authenticators” tables in FIDO server DB and give a Policy object as response to this API. On UI, the end user will get to see few details of the authenticators (for ex, few important fields like AAID, KeyID, etc from Policy object). ---- ***One new API needed here.***
4. Next, the user will install the app and accept push notifications for the app. This will fire an API on the server. Earlier this API used to take only deviceid and devicetoken as parameters and store them in “devicedetails” table in FIDO server DB. But now, we will have to pass the username also, along with the above two fields to know which user it is exactly. ---- ***One existing API modification needed here.***
5. When registering a new user to RP website, see that the email address entered is same as username used so far in enterprise policy management website as well as username passed when user accepts push notifications on the FIDO client app. This will help in maintaining a link of user’s email address in all important tables, which might be required for further usage in server.
6. After the user completes the QR scanning for RP registration and goes on to make the regRequest call, it should now pick up the new policy that was created and stored in FIDO server DB. ---- ***One existing API modification needed here.***
7. The same policy will also be picked up at the time of Authentication/Transaction when FIDO client app fires the authRequest API. ---- ***One existing API modification needed here.***

**Note:** Above API changes will include**:**

**1) Designing and implementing the Enterprise Policy Management Tool website. (Design + Backend API Calls).**

**1) Main logic code, DB code changes to all new APIs to be developed on FIDO server.**

**2) Main logic code, DB code changes to all affected existing APIs on FIDO server.**

**3) New Database and Table creation in MySQL to store the Enterprise Policy Management website details for user.**

**4) Database schema changes to existing tables (in MySQL) in the FIDO server Database as per new requirement.**