Group 3 - Team 2
Requirement Document for Application Manager & LDAP authentication

Members:-

- Ujjwal Prakash (2022202009)
- Priyank Mahour (2022201047)
- Sayantan Trivedy (2022201019)

Functional Overview:-

Our team is responsible for developing two key components: the application manager, and the LDAP authentication.

- Application Manager

1.) The application manager provides the interface to interact with the platform.
2.) It displays the platform's info, its health and other related and important paramters of the platform as provided by the different subsystems to the platform admin.
3.) It allows application developers to upload the compressed file of the application which the end user can deploy based on their requirements.
4.) Finally, it allows the end user to deploy the app of his choice based on his requirements.

- LDAP authentication

1) LDAP provides authentication using Active directory and provides role based authentication depending on the stakeholder (Platform admin, App Dev ,user)
2) LDAP primarily Contains Two Servers : (a)L DAP Server which implements Active Directory using LDAP protocol and Runs it in a Docker Container with Role Based Configs and Directory Information TREE (DIT).
3) (b) Flask Server to which the Application Sends Authentication Request , this Request is then wrapped in (DIT) Structure and Sent To LDAP Server for verification .

Block Diagram:-

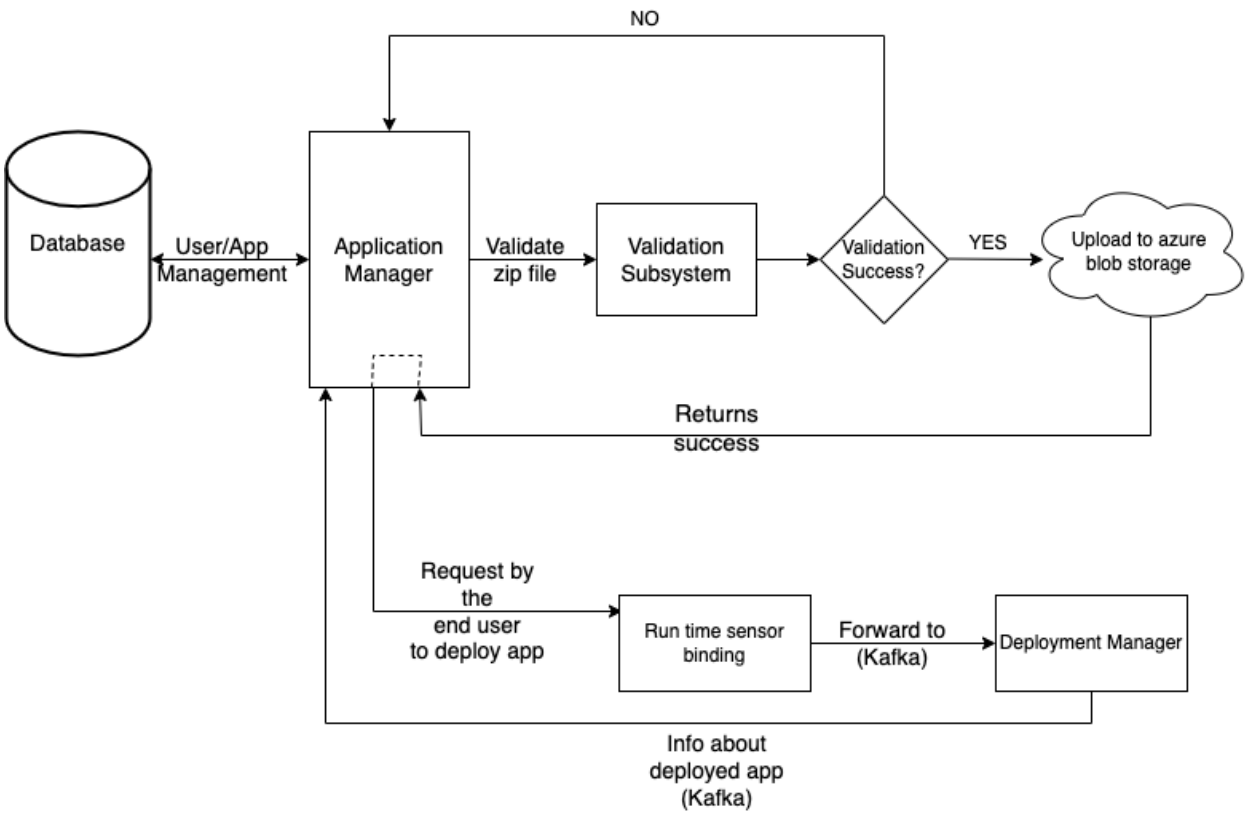

Fig: Application Manager

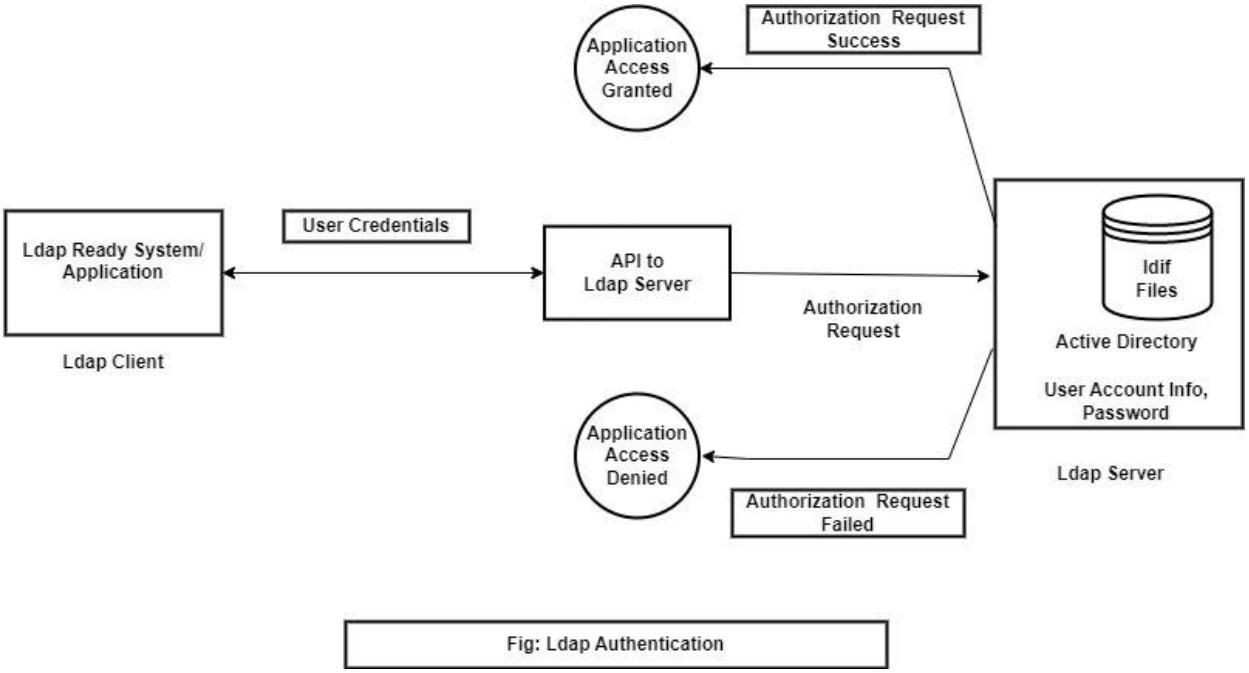

Fig: LDAP Role-Based Authentication

List of SubSystems:-

- App/User DB
- Kafka communication
- Azure storage for uploading the app
- OpenLDAP

List of Services:-

- Application Manager

1.) The application manager gets a zip file to be uploaded the application developer.
2.) It then validates the configs of the uploaded zip and if the it passes all the checks, the zip file is extracted and stored in the azure blob storage.
3.) Now the user has all the list of uploaded applications from which we can choose an app to deploy based on his requirements.
4.) The user selects the app to be deployed and then provides sensor binding to it.
5.) The application manager downloads a replica of the application from the base container and updates the information about the sensor binding in it.
6.) It then sends a Kafka message to the deployment manager to deploy the application on the platform.
7.) Post receiving the Kafka message from the deployment manager with the deployment info, the application manager shares the same with the user.

- LDAP

1.) The Application Sends Authentication Request to the flask Server.
2.) The flask Server Extracts User Information from the request.
3.) It then Wraps the User info in the Directory Information Tree (DIT) Format and Forwards this to the LDAP Server.
4.) LDAP server on receiving request from Flask Server Validates it Against the Rootname , Organizational Unit and Matches For the Relative Domain name of the user.
5.) This helps LDAP server to Authenticate the user Against Active Directory(DB).
6.) Once the user creds are matched, LDAP server Looks for the Authorization Level of User in the ldif-directory to Role based Access.

Interaction between application manager and deployment manager:-

- It takes place through Kafka with the following message structure,