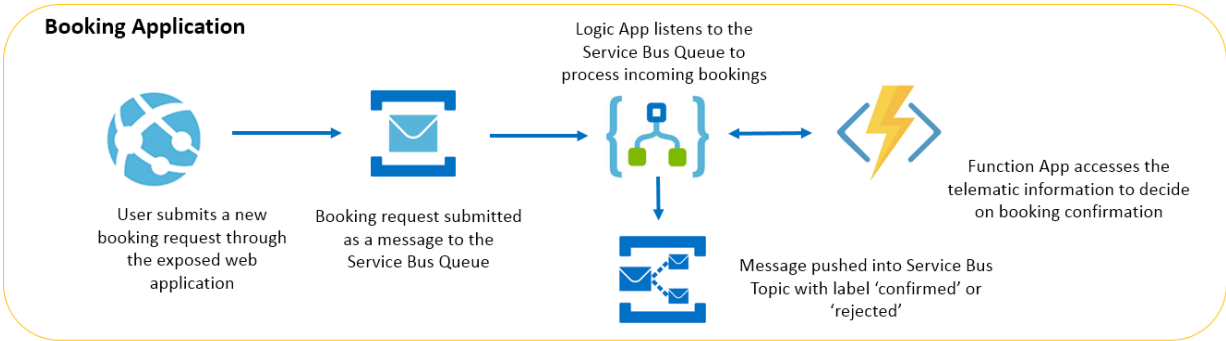
CST8917 – Serverless Applications

Lab 4: Implement a Booking System Workflow with Azure Services

Overview

The goal of this lab is to design and implement a booking system using Azure Services. Students will use a combination of Azure Service Bus, Azure Logic Apps, and Azure Functions to build an end-to-end booking processing system. The system should be capable of handling user booking requests, processing them asynchronously, and confirming or rejecting the bookings based on specific logic.



Team Formation

Please form teams of **1 to 3 students**. We encourage collaboration and peer-learning through this team dynamic. Once formed, please designate a team leader to coordinate with the instructor and submit the team list by **Thursday, March 28, 2024**.

Tasks and Deliverables:

Here is suggested Steps to create this Booking system. But feel free to use your own approach.

1. **Prepare the Azure Environment:**
    - Create a new resource group for this lab in the Azure Portal.
    - Inside this resource group, create the following resources:
        - Azure Service Bus Namespace
        - Logic App
        - Function App

2. **Setup Azure Service Bus:**
    - Inside your Service Bus Namespace, create a new queue named **booking-requests**.
    - Create a topic named **booking-status** with two subscriptions: **confirmed** and **rejected**.
3. **Develop the Booking Application:**
    - Simulate a booking application using a language of your choice or pseudo-code.
    - Implement the functionality to submit a message to the **booking-requests** queue when a user makes a booking.
4. **Implement Azure Logic App:**
    - Design a Logic App that triggers when a new message is placed onto the **booking-requests** queue.
    - Add an action to send a response back to the Service Bus topic **booking-status** with a label **confirmed** or **rejected** based on a condition you define (e.g., availability, user verification).
5. **Create Azure Function:**
    - Develop a function in the Function App that is triggered by the **booking-status** topic subscriptions.
    - The function should read the label of the incoming message and perform an action:
        - If the label is **confirmed**, send an email to the user confirming the booking.
        - If the label is **rejected**, send an email to the user indicating the rejection of the booking.
6. **Testing:**
    - Manually inject messages into the **booking-requests** queue to simulate booking requests from users.

- Ensure that the Logic App triggers as expected and places messages onto the **booking-status** topic.
- Verify that the Azure Function triggers on messages from the topic subscriptions and executes the confirmation or rejection logic.

7. **Report:**
    - Document the entire process, including:
        - Architectural diagram of your implementation
        - Explanation of the logic used in the Logic App and Azure Function
        - Challenges encountered and how they were resolved
        - Screenshots of the configuration and results from testing
8. **Submission:**
    - Submit the report along with any code developed for the booking application and the Azure Function.
    - Include a reflection on how this lab could be adapted for real-world scenarios and the potential scalability considerations.

Evaluation Criteria

- Correct implementation of the Azure Service Bus, Logic App, and Function App
- Successful demonstration of message flow through the system
- Quality and clarity of the documentation and code
- Creative problem-solving and error handling within the workflow