**Architecture Overview:**

For the purpose of the test, the integration of Priority pass (PP) functionality within the Taxi app is considered.

1. Integration of PP functionality within the Taxi app is facilitated through OAuth2 flow.
2. The taxi app communicates with its backend functionality through an API gateway through REST API's.
3. API access that requires PP access would be intercepted by the API gateway and an Access Token is created and used post successful authentication of the user in the PP UI through redirection.
4. The generated Access Token is used when accessing the backend Priority Pass API's
5. The API gateway interfaces with BFF API's which in turn communicates with business services, which would then use an integration layer to access PP API's.
6. The component diagram shows the Auth components separated from business components.

**Omissions/Trade-offs:**

1. Decided not to have SDK level integration for better UX and also to avoid complexities.
2. OpenID Connect not implemented - The current design uses OAuth 2.0 solely for API authorisation, as no additional identity claims from Priority Pass are required at this stage. However, the architecture can be extended to include OpenID Connect in future if richer profile data or federated login capabilities are needed.
3. The taxi app integration within PP app is not considered for the purpose of the test. It is envisaged to work in a similar architecture. The second reason for omission is because it would be prudent to focus on one integration for the MVP - the MVP integrating PP functionality within the Taxi app is considered to deliver value to the business.
4. Token rotation flow not shown in diagrams - For simplicity, the component diagram omits the detailed refresh token exchange and rotation logic. In the actual implementation, short-lived access tokens would be refreshed using secure, long-lived refresh tokens to maintain a 3-month session without sacrificing security.
5. We are not using any third party flight data services for simplicity and MVP requirements could be met without it.
6. I haven't considered any Taxi backend API's for implementing Taxi app functionality as it is beyond the scope of this test.
7. Push notification service (although used in Step2) is not considered for this test as it is viewed as an optional functionality.

**Features/Polish skipped:**

1. Integration of the Taxi app within the priority app is skipped for the purpose of the test and the MVP.
2. Other PP benefits (other than lounge), are not covered in the UX for simplicity and MVP prioritisation.
3. The UI/UX is designed to help visualise the steps of the core flow. So therefore I skipped the flow for “new to PP” user journey and assumed an existing PP user accessing through the Taxi app.
4. I have not taken into account a user starting an adhoc taxi journey as it is not too relevant for the MVP.

**Any shortcuts taken:**

1. I’ve designed the solution and the UX flow based on my current assumptions and understanding. This could involve some shortcuts - however a requirement workshop and demo would give rise to more clarity on any missing details and can be subsequently implemented.
2. The clickable prototype is vibe coded through Cursor. It can be easily enhanced and is good for a demo.