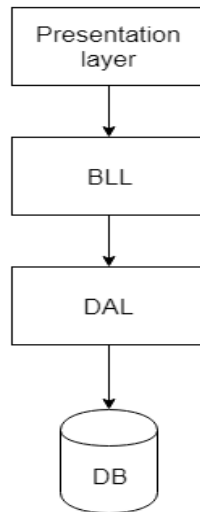
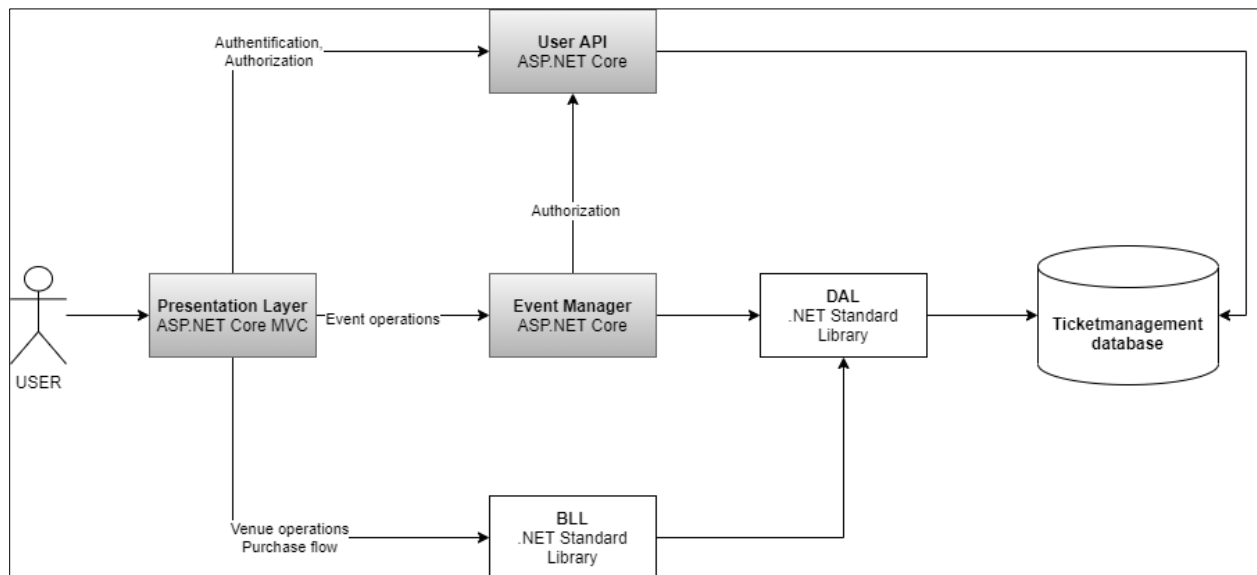


#### Task 4

Refactor Ticket Management application that was created in Task #2. Update current solution architecture (Figure 1) to service-oriented architecture (Figure 2).



(Figure 1)



(Figure 2)

It means that in result you should have **at least 3 executable non-test projects** (colored in gray on the Figure 2).

## Requirements:

1. Create a new **ASP.NET Core WebAPI** project that will expose service contacts for the following operations:

- Event Management
- Add documentation for API (swagger)

Migrate existing event management functionality to this service.

2. Create a new **ASP.Net Core WebAPI** project. This service should be close to RESTful API:

- Authentication endpoint
- User CRUD operations
- Use JWT as an authentication and authorization approach.
- Add documentation for API (swagger)

Migrate existing user functionality to this service.

3. Remove BLL references related to the User functionality and event management functionality from the Web application project. All operations should be performed via HTTP (with the help of **HttpClient**) calls. **OPTIONAL**. See point №7 for more info.

4. As a requirement you should create or amend existing README.md with details on:

- How to build and run the whole solution, with all services necessary to correct application work. Optionally specify how to start your solution in debug mode properly.
- **OPTIONAL**. Make a .bat file to simplify this process. It can contain a script to build and run all necessary projects.

5. All unit, integration should be able to run and should be all passed.

6. All components should have logging in place. Recommended using **Serilog** for that. Incoming requests should be logged in to the console and file.

7. **OPTIONAL**. Move purchase flow and venue manager to its own service. Remove all references to BLL and DAL from the presentation layer project. Add swagger documentation for that as well.