Task 5

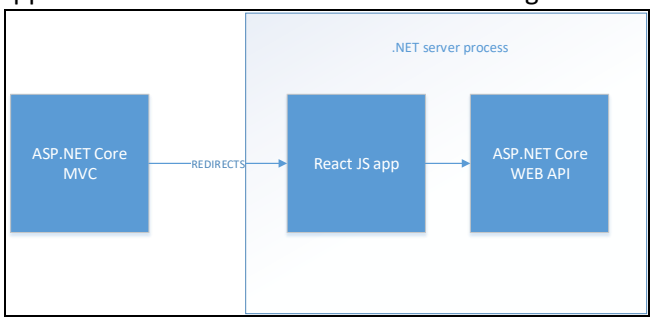
Your customer wants further expanding of application and decided to split existing site further into separate UI components. Your task is to migrate some parts of your application to the new architecture. The customer decided to use React.js as a new approach in building new composition.

Requirements:

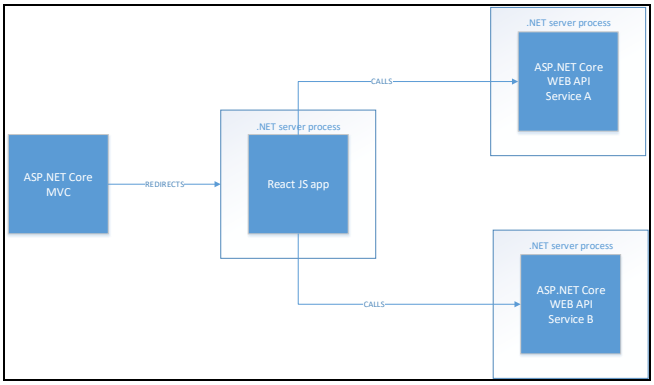
- Migrate **EventManagement** functionality and **User Profile page** to the React.js client-side applications.
- Re-platform existing UI written in *.cshtml* template into *JavaScript* files using a create-react-app template
- You should host your client-side app outside of the main MVC application process. Use ASP.Net core application as a host for your client-side part. For simplicity's sake, you may add client-side applications into existing WEB APIs (for example in event management API or user management API). In case you have a complicated logic that implies calls to different API's you see a possible solution in appendix 1.
- Functionality in the re-platformed application should be the same as in the old one. Use styles for them. Style may differ from the old application.
- Do not remove old UI logic. You redirect all requests to your new application based on the feature flag state. Use a combination of feature flag feature and redirect action filter to achieve that. See appendix 2 for more info.

Appendix 1

Default scheme of application interaction will look like following:



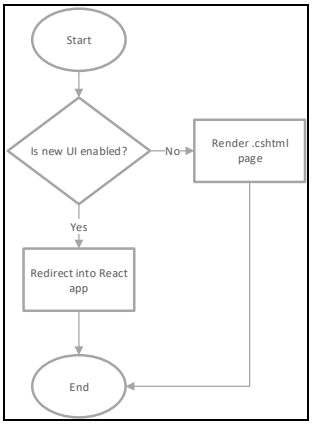
More complex scheme should be resolved in this way:



Use this approach if you have complicated logic in controllers that requires calls to different API in scope of the one action.

Appendix 2

Program flow with the migration in place:



You may use *appsettings.json* as a source of feature flag value.

Useful links:

Link	Description
Feature flags	Default ASP.NET Core approach for handling feature flags
Redirect filter	Redirect filter as a recommended way to handle the switch. This describes only basis of the approach. You have to adapt this to your application
React template in Visual studio	Default approach of creating React client-side app. Use this a base for your project. Pay attention to Startup configs and configs in .csproj file
ASP.NET Core Identity SPA	React + JWT auth tips