Jessica Neamtu – Group 258

SERVICE ORIENTED ARCHITECTURE

DOCUMENTATION

# Problem statement

The Event-Tracker is a web application and its scope is to keep people informed about the events.

The application provides two main functionalities:

* Display of all events;
* Add a new event to the list.

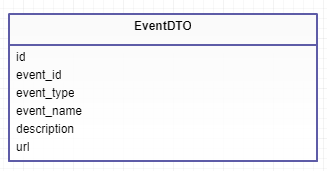
**Architecture overview**

The entire application is based on the Service Oriented Architecture. The application is structured in two components: **eventServer** (the server) and **eventClient** (the client application).

The server part of the system **eventServer** is a web application that connects to the application server https://api.trade.gov/v1/trade\_events/search? . and provides data regarding the available events. In other words this component is the provider and allows two operations such as: get all events and add a new event. **eventServer** is a web service and offers the REST services which are consumed later by the consumer.

The client part of the system, **eventClient** is the system which consumers the services from the server. In other words, this component gets the data through the REST services and process them in order to display them in the application.

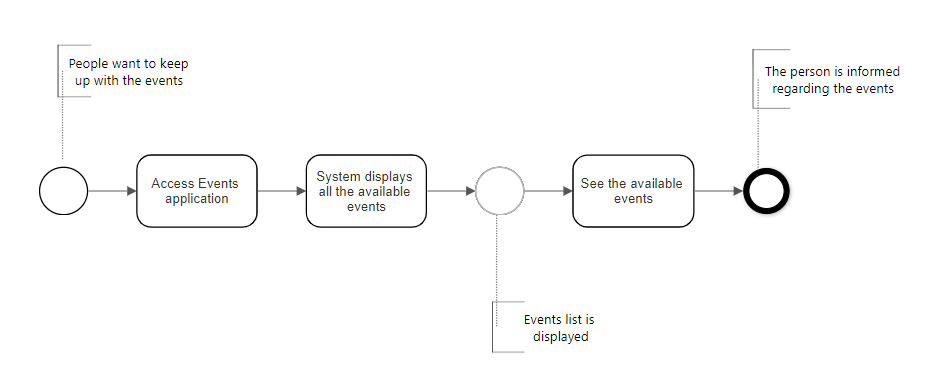
The entity used by both components, server and client, is EventDTO which has the next structure:

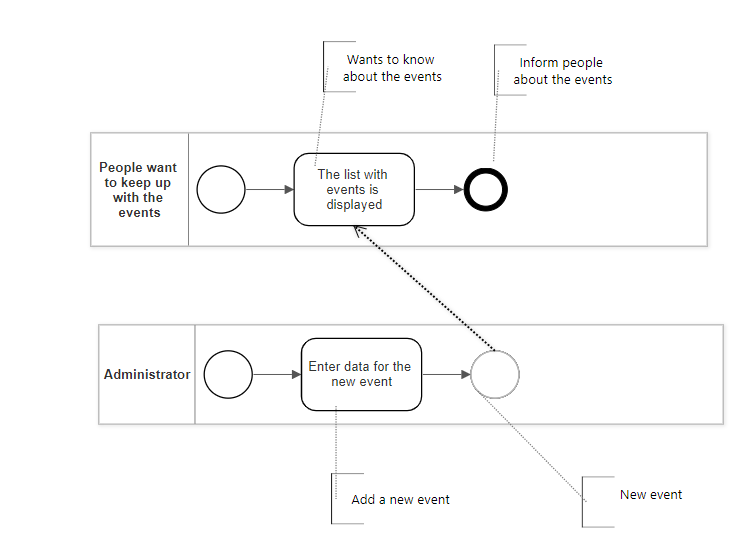


The event entity contains an id, an event id, the type of the event, the name of the event, a description and an url. It’s used to create the event objects.

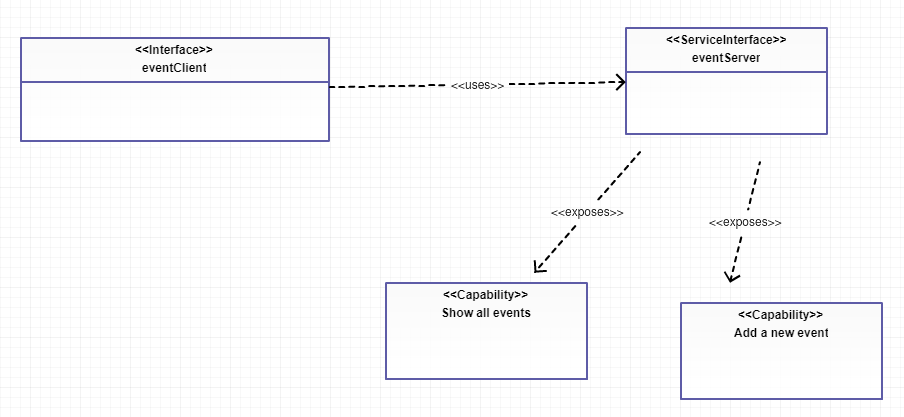
**3. Diagrams**

* **BMPN diagrams:**

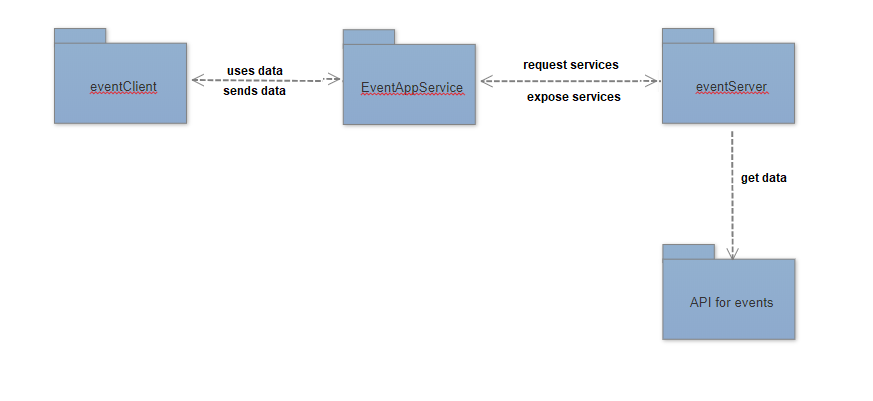




* **SOA ML Diagram**



**4. Message Façade Pattern**



The pattern found in this application is encapsulated behind a façade and this is how it works:

The service **EventAppService** is the one requesting the data from the eventServer (the eventServer gets the data from the API) and sends further the data to the controller in eventClient (EventAppController).