Graz University of Technology

Institute for Technical Informatics

Smart Service Development UE, SS2022
Konrad Diwold
kdiwold@tugraz.at

Group Assignment 3: Project Specification (4pt)

Deadline: 12.05, 2022; 23:59 CET

Introduction In this assignment, you will create an initial specification of your project before you start coding. The assignment gives specific instructions about the type(s) of specification we want you to generate.

Motivation As announced in Assignment 2, the **only external requirement** that we impose on your service system is that your service system must contain **at least three stand-alone services** that are **deployed on at least two different hosts**. Now that you start to specify the service system please make sure that service coupling is done **via open APIs** (preferably REST interfaces). Also, we want you to consider interoperability with services by third parties already from the very beginning: therefore, you should **decouple** the individual services in your system and we will ask you to produce **lightweight semantic markup** with your interface descriptions. You can use the service idea you created in Assignment 2 or come up with a different one (please discuss that with me and Josef).

(1) Scoping and Structuring (0.5 pt)

Create the definitive project abstract that includes:

- a brief motivation for your project;
- a brief discussion of its main components in prose;
- a description of how users would typically interact with the service system you plan to create in the form of a short user story.

(2) Services and their Interfaces (3.5 pt)

Create a **UML component diagram** of your service system[1]. Next, **list all individual services** of your service system; for **each** service:

- Give a **one-line description** of what the service's purpose is and which APIs it provides to external clients.
- Formally specify the **service's API**. If the service provides a REST API, use the OpenAPI specification (v3)[2]; if the service provides an asynchronous interface (such as MQTT), use the AsyncAPI specification (v1)[3]; if the service does not provide any APIs, indicate the interfaces it depends on. Make sure that all functionality provided by a component's API is associated with the respective interface in your UML component diagram. The assignment should contain all interfaces required to realize your service!

[1] See http://agilemodeling.com/artifacts/componentDiagram.htm for a good starting point.

[2] You can use the example specification at https://github.com/OAI/OpenAPI-Specification/blob/master/examples/v3.0/petstore-expanded.yaml as a starting point.

[3] You can use the AsyncAPI guide at https://www.asyncapi.com/v1/guide/ as a starting point.

(3) Hand-in Instructions

By the deadline, hand in via TeachCenter/Email: a **report** that states your group name and group members, pitfalls you encountered while working on the assignment, and your responses to the assignment itself. Please use UTF-8 encoding for your documents and avoid special characters.