Distributed Programming II

A.Y. 2017/18

Assignment n. 3 – part a)

All the material needed for this assignment is included in the *.zip* archive where you have found this file. Please extract the archive to an empty directory (that will be called `[root]`) where you will work.

This assignment is about the design of an XML-based RESTful API for remote access to the DP2-NFV system considered in the previous assignments. The web service has to be designed according to the following specifications.

- The web service must allow its clients to:
    1. Read all the information available in DP2-NFV about NF-FGs and hosts, including the same information that was defined for Assignment 1 and the information about reachable hosts that was used in Assignment 2.
    2. Deploy a new NF-FG into the system (when a new NF-FG is deployed, its nodes are allocated onto the IN hosts and the new NF-FG is added to the list of NF-FGs; if it is not possible to allocate all the NF-FG nodes, deployment fails, and nothing is added). If deployment is successful, deploy time is set to the current time.
    3. Undeploy an NF-FG (when an NF-FG is undeployed, its allocation relationships are removed and the NF-FG itself is removed from the list of NF-FGs).
    4. Add or remove a single node in an already deployed NF-FG (when a node is added, it is added without links and it is also allocated on a host; if allocation is not possible, the addition of the node fails and nothing is added; removal must be possible only if the node has no incoming links and no outgoing links; removing the node also removes its allocation).
    5. Add or remove a single link in an already deployed NF-FG (in case of addition, the client must be able to specify what should happen if the link is already present, i.e. whether the link information should be overwritten or an error should be returned).
- For the operations that require allocation of NF-FG nodes onto IN hosts the service is free to use any allocation algorithm but the client must be able to choose among the following two possible options for each node that has to be allocated:
    1. The decision about which host is used for the node is taken internally by the service with no input from the client
    2. The client suggests the desired host onto which the NF-FG node should be allocated. If possible, the service must use the suggested host, otherwise the service can change it.

The output of this design phase has to be written into an XML schema, which describes the data types used by the service, and an HTML report, which documents the design of the RESTful API for the users. The schema has to be stored in the file `[root]/xsd/NfvDeployer.xsd`, and the HTML report has to be stored entirely into the folder `[root]/doc/`, and made accessible starting from the file `[root]/doc/index.html`. If deemed useful, the schema can be split into multiple files. In this case, `NfvDeployer.xsd` must be the main file which includes the other local files, and all files must be stored in `[root]/xsd`.

The HTML report must include the following items:

1. Description of the conceptual structure of resources (identify the resources, with their hierarchical structure; use plural names for collections and singular names for resources that are not collections; specify the HTTP methods allowed for each resource).
2. Mapping of the resources to URLs (for simplicity, use URLs that are relative to the base URL of the service; for resources that belong to collections, use the notation {id}, where id is the name of the field that identifies the resource in the collection)
3. Description of each possible operation. For each resource, and for each method allowed on the resource, specify the allowed query string parameters and/or the allowed request body, the possible status codes, and, for each possible status code, the response body. For each allowed query string parameter, specify name and XML schema type of the value (either a standard XML schema simple type or one defined by you in the `NfvDeployer.xsd` schema document). For request body and response body, specify the XML element, which must be defined at the global level in the `NfvDeployer.xsd` schema document. Schema elements and types in this description must be referenced by name. In practice, the `NfvDeployer.xsd` schema document must include a set of global named types and a set of global elements. As far as possible, the definitions already developed in the schema for Assignment 1 should be reused. For each possible operation, a short description should also be added when not obvious.

The web service must be designed so as to be compliant with the REST principles and conventions, robust, easy to be used by clients, and enabling efficient execution of all the operations for which it has been designed. The documentation should be self-contained (it should include all that is necessary for a user to know to be able to consume the service, apart from what can be deduced from the service itself and from the HTTP standard).

When designing the service, consider that it should be able to scale to hundreds of NF-FGs (with a total of many thousands of nodes).

Correctness verification

In order to be acceptable for examination, the produced XML schema document must be valid. This can be checked by means of the Eclipse schema validator.

Submission format

The produced documents will be submitted along with Part b) of Assignment 3, for which instructions will be given later on.