Distributed Programming II

A.Y. 2016/17

Assignment n. 3 part a – NffgService REST API Documentation

NffgService provides the possibility to load, delete and retrieve whole NFFGs. Moreover it is possible to load, delete and retrieve policies and verify if the reachability policies are satisfied or not.

- **Resources Hierarchy**

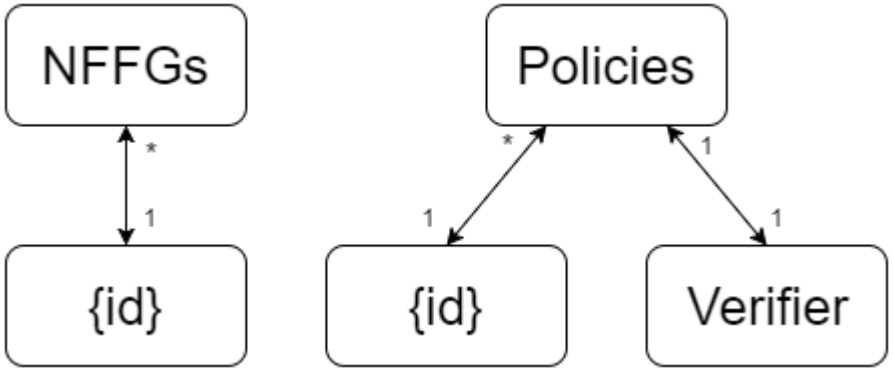

Figure 1: resources hierarchy

- **Resources Mapping**

  - baseURL*/ Documentantion page
  - baseURL*/application.wadl WADL description of NffgService
  - baseURL*/nffgs Collections of NFFGs
  - baseURL*/nffgs/{id} Single NFFG within the collection
  - baseURL*/policies Collections of Policies
  - baseURL*/policies/{id} Single policy within the collection
  - baseURL*/policies/verifier Verifier of the reachabilty policy

*baseURL = base url of the service hosted by the server (e.g. *http://localhost:8080/NffgService/rest*)

- **Operations on NFFGs**

Resource URL*	Method	Query params	Request Body	Response Body	Status code	Meaning
/nffgs	GET	-	-	Content-type: application/xml Entity: entityPointers	200	Return the pointers to the nffgs loaded on the service.
/nffgs	GET	-	-	-	204	There are no nffgs loaded on the service.
/nffgs	POST	-	Content-type: application/xml Entity: nffgs	Content-type: application/xml Entity: entityPointers	201	Return the pointers to the created nffgs.
/nffgs	POST	-	Content-type: application/xml Entity: nffgs	Content-type: text/plain Entity: error message	403	nffgs contained in the request body have not been created. There is at least one nffg with a name already used by a nffg stored on the service.
/nffgs	DELETE	-	-	-	204	All the nffgs and all the policies have been deleted.
/nffgs/{id}	GET	-	-	Content-type: application/xml Entity: nffgs	200	Return the nffg found.
/nffgs/{id}	GET	-	-	Content-type: text/plain Entity: error message	404	nffg not found.
/nffgs/{id}	DELETE	deletePolicies: boolean	-	-	204	If *deletePolicies == true* the nffg and all the policies referring to it have been deleted; if *deletePolicies == false* the nffg has been deleted since there is no policy referring to it.
/nffgs/{id}	DELETE	deletePolicies: boolean	-	Content-type: text/plain Entity: error message	403	nffg not deleted since *deletePolicies == false* and there is at least one policy referring to this nffg.
/nffgs/{id}	DELETE	deletePolicies: boolean	-	Content-type: text/plain Entity: error message	404	nffg to delete not found.

*Resource URL are relative to baseURL

- **Operations on Policies**

Resource URL*	Method	Query params	Request Body	Response Body	Status code	Meaning
/policies	GET	-	-	Content-type: application/xml Entity: entityPointers	200	Return the pointers to the policies loaded on the service.
/policies	GET	-	-	-	204	There are no policies loaded on the service.
/policies	PUT	-	Content-type: application/xml Entity: policies	Content-type: application/xml Entity: entityPointers	201	Return the pointers to the policies created.
/policies	PUT	-	Content-type: application/xml Entity: policies	Content-type: text/plain Entity: error message	403	policies not created. There is at least one policy referring to a missing nffg or node.
/policies	DELETE	-	-	-	204	All the policies have been deleted.
/policies/{id}	GET	-	-	Content-type: application/xml Entity: policies	200	Return the policy found.
/policies/{id}	GET	-	-	Content-type: text/plain Entity: error message	404	policy not found.
/policies/{id}	DELETE	-	-	-	204	policy has been deleted.
/policies/{id}	DELETE	-	-	Content-type: text/plain Entity: error message	404	policy to delete not found.
/policies/verifier	PUT	-	Content-type: application/xml Entity: policies	Content-type: application/xml Entity: policies	200	The policies contained in the request body have been verified and sent back in the response body along with the verification results.
/policies/verifier	PUT	-	Content-type: application/xml Entity: policies	Content-type: text/plain Entity: error message	403	The policies contained in the request body have not been verified. There is at least one policy referring to a missing nffg or node.
/policies/verifier	POST	-	Content-type: application/xml Entity: namedEntities	Content-type: application/xml Entity: policies	200	The policies specified by the request body and stored on the service have been verified. They are sent back in the response body along with the verification results.
/policies/verifier	POST	-	Content-type: application/xml Entity: namedEntities	Content-type: text/plain Entity: error message	403	The policies specified by the request body are missing on the service.

*Resource URL are relative to baseURL

Assignment n. 3 part b – NffgService REST Design and Implementation Documentation

- NffgService Design structure

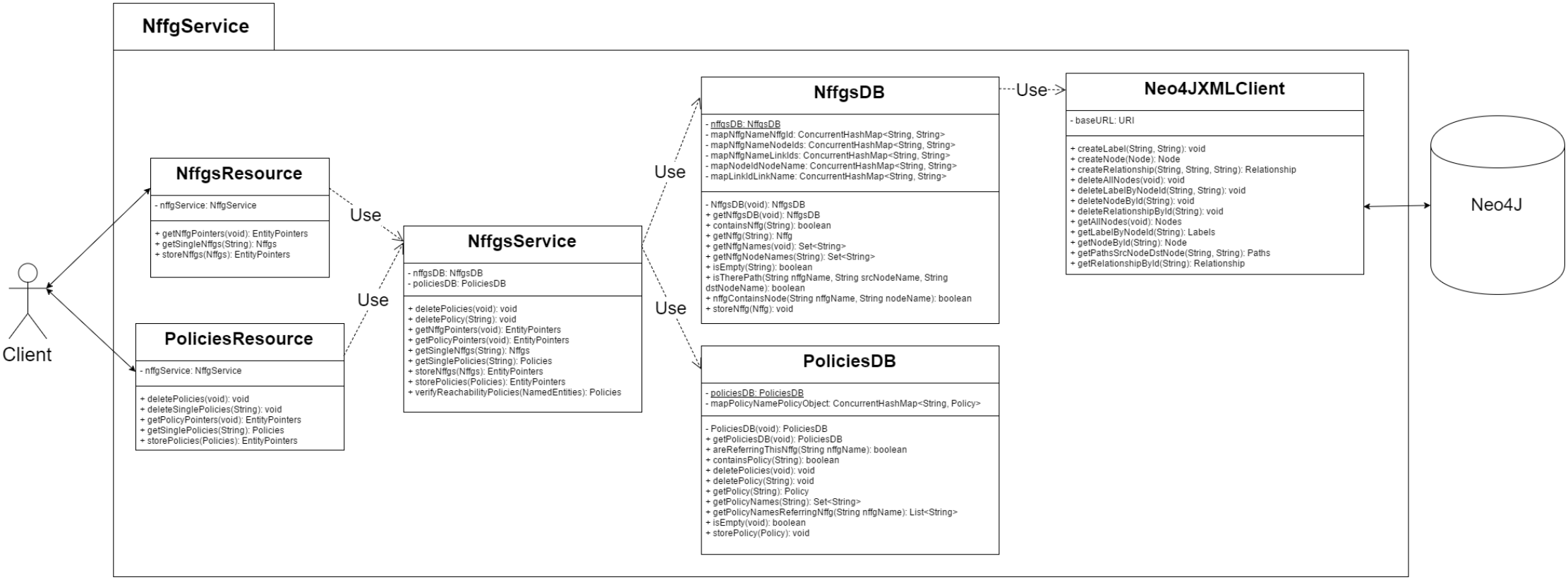

Figure 2: "simplified" class diagram of NffgService. It is intended only to give an idea of the developed solution.

- Design & Implementation choices

The information about NFFGs is stored on Neo4J (including networkFunctionalities and update time). NffgService is just a "cache" where to store simple and small data useful to retrieve nodes and relationships from Neo4J and map them with the data sent to/requested from the client. This implementation allows the clients to load large quantity of data avoiding massive local memory usage; the minus point is that NffgService requires more time to send the response to the client.
Given this implementation, in order to support scalability NffgService will never return the information about all the NFFGs at once in a single response: the client has to perform a GET for each NFFG to download.

When a client asks for the verification of a policy, NffgService return the verified policy containig its verificationResult element. In this way the client can always check the policy actually tested.