

LinkColorrgb0,0,0 MarginColorrgb.29,.03,.06

Design and Implementation of a TOSCA compliant Interface for the OpenSDNCore Framework

Next Generation Networks
TU Berlin

submitted by
Marek Sobolewski,

Berlin, 20. March 2015

First reviewer : Prof Dr. Thomas Magedanz
Second reviewer : Alexander Willner, M.Sc.
Day of submission :

Abstract











Introduction and Problem Statement Continuous and broad innovation in the Information and Communications sector has extensively changed the way that communication takes place between peers during the last years, as well as it has changed who and how services are provided to users. The rapid growth in data traffic demand and an increasing competitive pressure has decreased profits for telecommunication-network providers and has resulted in overall value chain challenges. To go along with altered market requirements and conditions a new technological approach had to be found.

Related Work and Research Questions: Current telecom infrastructure consists of a large amount of heterogeneous network elements such as routers, gateways, firewalls, policy control and so on. Orchestrating these elements to work together to rapidly create a new service is demanding if not nearly impossible. Semi-automatic creation and management of application layer services must be orchestrated in accordance with constraints or policies of the underlying network infrastructure. SDN in conjunction with VNF is about to cope with this challenge. SDN and NFV can be seen as a new network architecture paradigm applying new technological capabilities to network functions, network design and service platforms in order to achieve agile, flexible, scalable and efficient networks, which will help to lower management costs and OPEX in the long run. The OASIS TOSCA standard defines a metamodel for defining IT services. It introduces the concept of a service template which defines both the structure of a service (topology) as well as how to manage it (orchestration). The generic service template describes what needs to be preserved when an application is migrated over alternative network or cloud environments. This ensures interoperability of deployment and management throughout the complete service lifecycle. On the other side, members from IT and telecoms industries are currently working to develop standards for Network Functions Virtualisation (ETSI NFV). Early implementations demonstrate that network design itself has to be more agile instead of having an ever increasing variety of proprietary hardware with single, dedicated functions. Instead the network needs to be able to respond instantaneous to the dynamic needs of the traffic and services running over it.

Own Approach: To establish a direct and reliable linkup between both the ETSI and OASIS data models can be a tedious task if one of the standards definitions change and a running system implementation needs to be adapted each time. Most likely some kind of changes will occur in future evolution. Hence there is a distinct need for an intermediary solution which will work independent of current data model specification. This can be done through a REST-paradigm approach and the usage of Plain Old Java Objects (POJOs) which will act as a robust bridge in-between both data model domains to ensure a hitch-free information flow.

Research Contributions: The main contributions of this thesis will be a concept (approximately 40 pages) and the development of a RESTful application Interface prototype which will implement the functionality to convert TOSCA information into POJOs and vice versa. All functions will be implemented with the aid of Java API for RESTful Web Services (JAX-RS).

Bachelor's Thesis Schedule

Bachelor's Thesis	Start	Frist	April	Mai	Juni	Juli
Thesis	1 April	29 Juli				
Abstract , Table of Contents	1 April	10 April				
Introduction	10 April	20 April				
Design, Implementation	20 April	20 Juni				
Evaluation, Conclusions	20 Juni	10 Juli				
Revise Thesis	10 Juli	29 Juli				
Implementation	1 April	29 Juli				
Access to POJO repository specification	1 April	10 April				
Design conversion method via J2EE-JAX-RS	10 April	1 Mai				
Implement REST-Interface	10 Mai	20 Juni				
Refactor, cleanup code	10 Juli	25 Juli				
Presentation						
First	1 Mai	1 Mai				
Second	30 Juli	30 Juli				