H2020 5G-TRANSFORMER Project

Grant No. 761536

5GT-SO DEVELOPER GUIDE

Abstract

This document contains guidelines for developers to expand the functionality of the 5GT-SO. These expansions consist of a) creation of new wrappers to support new MANO platforms and b) creation of new placement algorithms for the placement of network service using the underlying infrastructure provided by the 5G-MTP module.

Document properties

Document title	5GT-SO Developer Guide
Document responsible	Jorge Baranda (CTTC)
Document editor	Jorge Baranda (CTTC)
Editorial team	Josep Mangues-Bafalluy (CTTC), Luca Vettori (CTTC), Ricardo Martínez (CTTC)
Target dissemination level	Public
Status of the document	In progress
Version	0.1

Document history

Revision	**Date**	**Issued by**	**Description**
0.1	10 May 2019	Jorge Baranda (CTTC)	Initial version

Disclaimer

This document has been produced in the context of the 5G-Transformer Project. The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement № H2020-761536.

All information in this document is provided “as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors view.

Table of Contents

List of Acronyms

Acronym	Description
GA	Genetic Algorithm
MANO	Management and orchestration

1 Introduction

5GT-SO has been designed to facilitate its extension with new lifecycle management logic, new network service placement algorithms and its interfaces towards other external entities, in particular to other management and orchestration (MANO) platforms. In the following, we provide some details about how to extend the 5GT-SO to:

- Provide support for the creation of a new wrapper to interact with a different MANO platform.
- Provide support for new network service placement algorithms.

2 Creation of a new MANO wrapper

Currently, the 5GT-SO supports seamlessly the use of two open source MANO platforms, namely, OSM and Cloudify. Thanks to a wrapper system, if a user wants to use the 5GT-SO platform with another MANO platform, it only needs to do the following steps:

1) Modify the coreMano.properties file to include the relevant information of the new MANO platform. The coreMano.properties file is in the /5g-transformer.5gt-so/5GT-SO/coreMano/
2) Create a new file called <Mano>Wrapper.py in the folder /5g-transformer.5gt-so/5GT-SO/coreMano/ , which includes the definition of a the new <Mano>wrapper class. An example of the osmWrapper class can be found in /5g-transformer.5gt-so/5GT-SO/coreMano/osmWrapper.py file.
   This new class has to implement the following methods:
   - *__init__:* This method creates the new <MANO>wrapper object with the parameters included in the coreMano.properties to communicate with the new MANO platform.
     - In case of requiring an external client, like the case of OSM, this client needs to be installed in the computer where running the 5GT-SO and perform the appropriate imports. The files of this external client can be included in a folder called <mano>client in the /5g-transformer.5gt-so/5GT-SO/coreMano/ folder.
   - *instantiate_ns*: This method communicates with the MANO platform to instantiate a new network service. The parameters of this method are:

Parameter	Description
Nsi_id	Identifier of the network service instance
Ns_descriptor	Json objects containing the NS descriptor in IFA014 format
Vnfds_descriptors	A list of Jsons objects containing the VNF descriptor in IFA007 format
Body	Request body coming from the NBI
Placement info	Json object which is the output of the placement algorithm indicating the position of the different VNFs/AppDs in a network service
Resources	Json object containing the available resources computing and networking resources at the 5GT-MTP
nestedInfo	Json object containing mapping between nested and composite virtual links in case of service composition/federation

   - *scale_ns*: This method communicates with the MANO platform to perform a scaling operation in an instantiated network service. The parameters of this method are:

Parameter	Description
Nsi_id	Identifier of the network service instance to be scaled

Ns_descriptor	Json objects containing the NS descriptor in IFA014 format
Vnfds_descriptors	A list of Jsons objects containing the VNF descriptor in IFA007 format
Body	Request body coming from the NBI including scaling information
Current_df	String containing the current deployment flavour of the service to be scaled.
Current_il	String containing the current instantiation level of the service to be scaled
Placement info	Json object which is the output of the placement algorithm indicating the position of the different VNFs/AppDs in a network service

- *terminate_ns*: This method terminates the network service previously instantiated

Parameter	Description
Nsi_id	Identifier of the network service instance to be scaled

3) Extend the createWrapper() method of the coreManoWrapper.py file to collect the information included in the coreMano.properties file of step 1) and be able to instantiate new objects according to the class defined in the file create at step 2). This file is in the /5g-transformer.5gt-so/5GT-SO/coreMano/ folder.

3 Creation of a new placement algorithm

Currently, the 5GT-SO supports the use of three different placement algorithms to determine the position of the different constituents VNFs/Appds of a network service. These algorithms are:

- Placement Algorithm 1 follows the Genetic Algorithm (GA) paradigm, which uses a generic heuristic methodology for solving optimization problems by efficiently exploring the space of potential solutions.
- Placement Algorithm 2 provides the solution of an optimization problem where the objective function is the minimization of either the distance (MD) or the latency (ML) between VNFs located at remote NFVI-PoPs that compose a targeted service.
- Placement Algorithm 3 is based on a cluster-based approach, trying to find the best tradeoff between the deployment cost for the operator and the service latency.

Support for a new placement algorithm at the 5GT-SO can be achieved following the next steps:

1) Design a placement algorithm which also runs a REST server accepting requests and providing answers in the form of the API defined at /5g-transformer.5gt-so/pa/api/PA_API_defs.yaml. The information provided to this API is extracted from the network service descriptors in IFA014 format at the ROOE module of the 5GT-SO.
2) Include the code under a new folder in /5g-transformer.5gt-so/pa, like the other placement algorithms previously mentioned.
3) The use of different placement algorithms at the 5GT-SO is configured with the *rooe.properties* file in /5g-transformer.5gt-so/5GT-SO/sm/rooe. Basically, the developer needs to change the pa.ip address to point to the server where the new added placement algorithm is running.