Architecture of Cloud Telecommunication Network Monitoring Platform Based on Knowledge Graphs

Kirill Krinkin, Igor Kulikov, Alexander Vodyaho
Saint-Petersburg Electrotechnical University "LETI"
Saint-Petersburg, Russia
kirill@krinkin.com, i.a.kulikov@gmail.com,
aivodyaho@mail.ru

Nataly Zhukova
St. Petersburg Federal Research Centre of the Russian Academy of Sciences (SPCRAS)
St. Petersburg, Russia
nazhukova@mail.ru

***Abstract*—The article is aimed to describe the architecture of a new cloud platform for telecommunications networks (TN) monitoring based on knowledge graphs (KG). The platform allows telecom operators receive modern analytics based on monitoring data for all segments of their networks, regardless of the devices generation and monitoring systems used, while requiring low expenses. The proposed architecture makes it possible to connect telecom operators TNs to the monitoring platform with low cost due to the flexible mechanism for platform ontology managing. In the case study section an example of combining operators network segments which use various monitoring systems with different domain ontologies is discussed. Also building a single report based on monitoring data in different segments is demonstrated. The directions for further functional development of the proposed platform are identified.**

I. Introduction

The traditional tasks of TN monitoring are successfully solved by operators through the use of various existing monitoring systems. These tasks include [1], [2]: monitoring network devices and telecommunication channels, monitoring networks performance, monitoring networks key indicators, monitoring networks applications, generating reports and notifications about TN events.

In the article [2] new modern tasks of TN monitoring were formulated. These tasks require more complex integrated network models that combine data on various aspects of network functioning. Many telecom operators, when building monitoring systems for their networks, are faced with a situation when different generations of devices and different software for network monitoring are used in different segments of their networks. At the same time operators need to obtain generalized analytical reporting on monitoring in various network segments. Traditionally this type of tasks can be solved by combining reports generated in various monitoring systems with subsequent analysis. This requires a lot of time both for building the required report and for each of its modifications, since it requires the involvement of experts and this type of tasks often cannot be automated. On the other hand, the migration of all network segments to a single monitoring system is a very expensive procedure for operators and in most cases the migration is not rational. In this situation data on each of the network segments is processed in isolation and the operators either do not receive integrated reporting across the entire network or receive it in a very limited form.

Also the article [2] proposes the monitoring system architecture which allows solving both traditional monitoring tasks and new formulated tasks. For solving all the raised tasks it is proposed to use the knowledge graphs technology for building integrated TN models and store monitoring data.

This article develops the idea of using KG as a core for a new monitoring platform, expanding the architecture proposed in the article [2] to a cloud monitoring platform that can be used by many TN operators. Connection of all operators monitoring systems and other network management systems allows build unified network models. Due to monitoring data combination with integrated TN model, it becomes possible to do unified analytics of monitoring data due the initial TN models and monitoring data are semantically connected.

The use of the ontologies and the mechanism for ontology management provides the possibility of direct integration with monitoring systems which are also built on the base of ontologies by mapping their entities. For legacy monitoring systems that work with their own internal data models, it is possible to map the entities of the monitoring platform ontology with elements of their own data models at the level of data adapters. This approach is based on the telecommunication domain ontologies development [3-8] and makes it possible to operate the monitoring data of legacy systems as data corresponding to the monitoring platform ontology.

Summarizing the discussed above items, operators can get ability to integrate monitoring data from their systems into the integrated network model provided by the new monitoring platform without changing the functioning monitoring systems in various segments, including legacy systems that cannot be upgraded, and receive unified analytics across the entire network. The absence of the need to make changes in the operating monitoring systems ensures low costs for connecting to the new platform. In addition, the cloud architecture of the platform, allows provide the monitoring platform as a service for many operators at the same time, allows further reduce the cost of its use for telecom operators.

II. Monitoring platform architecture

The structure of the TN monitoring platform from the point of view of interaction with TN operators monitoring systems is shown in Fig. 1.

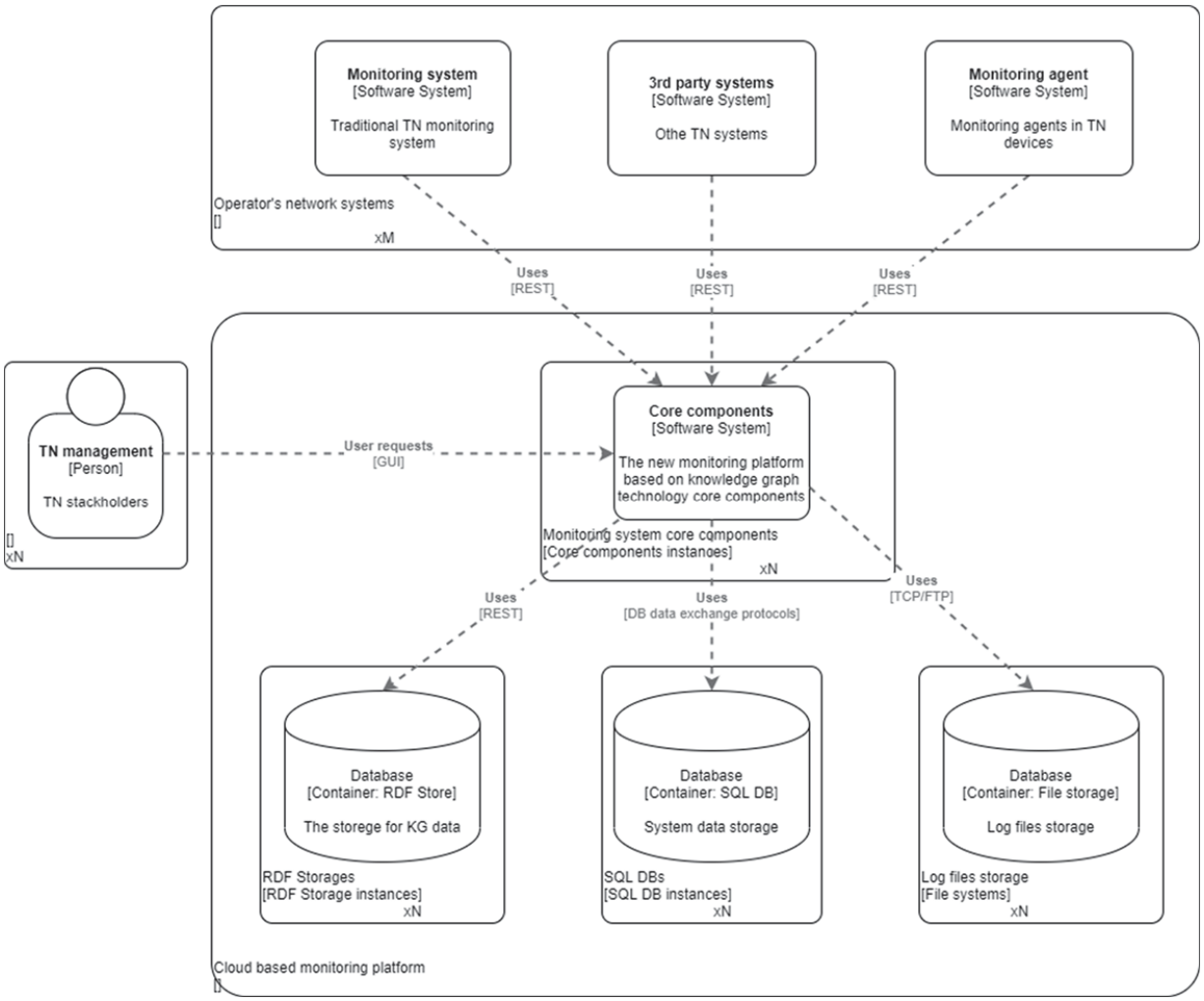

Fig. 1. The structure of the TN monitoring platform from the interaction with TN operators monitoring systems point of view

The monitoring platform includes the following components:

- **Core components.** The platform core contains data adapters, graphical user interface module, main business logic module, ontology management module and log files processing module.
- **RDF store.** RDF data store is the core of knowledge graph.
- **SQL DB.** Relational database designed to store metadata and service data of the monitoring platform in a tabular format.
- **File storage.** File storage intended for storing log files of network devices and other service data of the monitoring platform in file format with different structures.
- **Network systems of operators.** Including:
  - Operator's monitoring systems,
  - Platform monitoring agents on network devices,
  - Network management systems, which are the sources of initial graph network models (for example, billing model, access rights model, network topology model, data model, etc.) for building an integrated model. A detailed description of approaches to the construction of TN integrated graph models is presented in the articles [2], [9], [10].

The links between the components of the structural diagram and with users of the platform are described in Table I.

TABLE I. COMPONENTS AND LINKS OF THE MONITORING PLATFORM ON THE LEVEL OF TN OPERATOR INTEGRATION

Component	Related components	Relations type
Core components	RDF store	"Requests"
	File storage	"Uses"
	SQL DB	"Requests"
	TN management	"Is used by"
	Monitoring systems	"Is used by"
	Monitoring agents	"Is used by"

Component	Related components	Relations type
	3rd party systems	"Uses" or/and "Is used by"
RDF store	Monitoring platform	"Is used by"
File storage	Monitoring platform	"Is used by"
SQL DB	Monitoring platform	"Is used by"
Monitoring platform users (TN management)	Monitoring platform	"Uses"
Monitoring systems	Monitoring platform	"Uses"
Monitoring agents	Monitoring platform	"Uses"
3rd party systems	Monitoring platform	"Uses" or/and "Is used by"

The structure of the monitoring platform core is shown in Fig. 2.

The monitoring platform core consists of the following components:

- Business Logic Component.
- Data Adapters.
- Graphical User Interface (GUI) module.
- Monitoring data processing module.
- Log files processing module.
- Ontology management module.

Business Logic Component is designed to implement the main scenarios of the monitoring platform:

- Building of the operator's network integrated model based on monitoring systems data and the initial graph models provided by the operator's information systems.
- Updating the TN integrated model if necessary.
- Receiving data from operator's monitoring systems and monitoring agents and writing them to RDF storage and, if necessary, to SQL DB / File storage.
- Request log files from network devices and write them to File storage.
- Parsing log files and writing results to RDF storage and, if necessary, to SQL DB / File storage.
- Editing of the monitoring platform ontology.
- Generation of monitoring reports on user requests.

Data adapters are designed for converting data to / from formats of external systems and data storage to the internal format of the monitoring platform.

Graphical User Interface module is designed for providing users ability to receive the necessary reports on monitoring data and configure platform parameters and report formats.

Monitoring data processing module is designed for processing data received from monitoring systems, monitoring agents, as well as data obtained from log files of network devices and their writing to RDF storage and, if necessary, to SQL DB / File storage.

Log files processing module is intended for parsing logs according to the patterns transmitted to the module and transferring the received data to the monitoring data processing module.

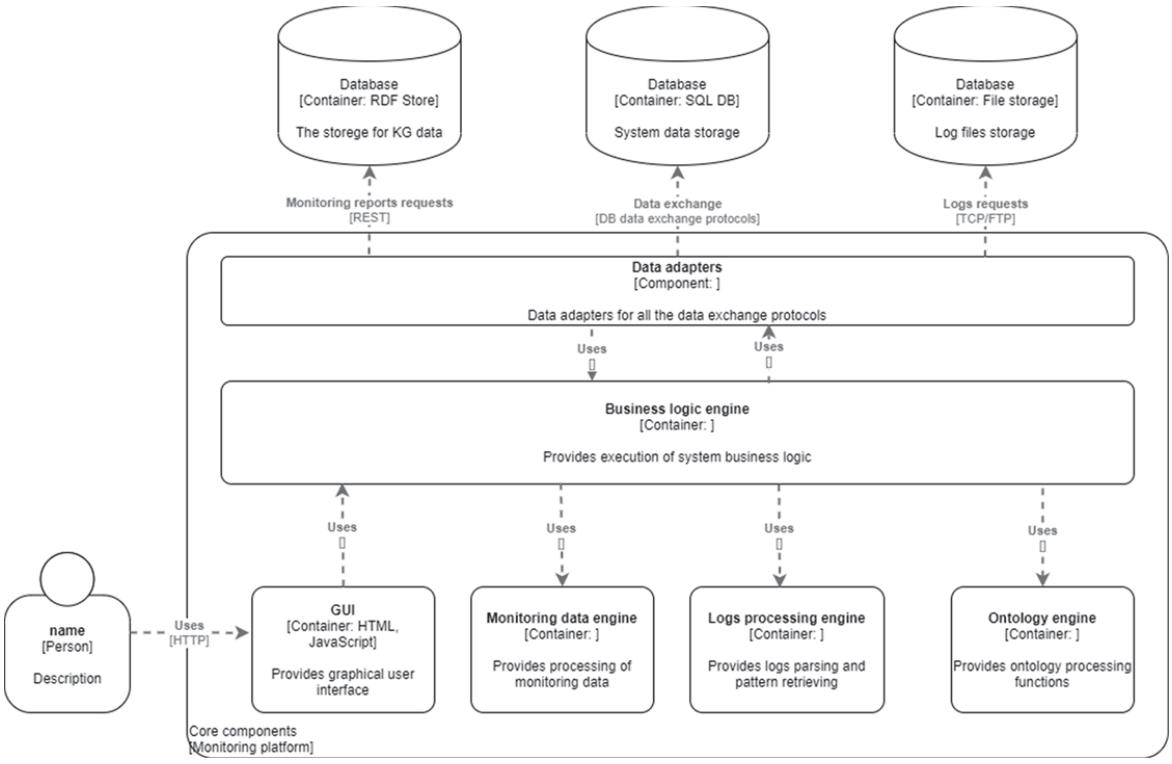

Fig. 2 The internal structure of the monitoring platform core is shown

Ontology management module allows users to edit the ontology of the monitoring platform. This may be necessary when integrating new systems or segments of the operator's network into the monitoring platform, as well as when changing the settings of the monitoring systems operating at the operator's side.

III. Ontology based integration

A. The monitoring platform ontology

The structure of the proposed monitoring platform ontology is multi-level and consists of the following levels: the level of basic ontologies, the level of domain ontologies, and the level of application ontologies. The hierarchy of the used ontologies is shown in Fig. 3.

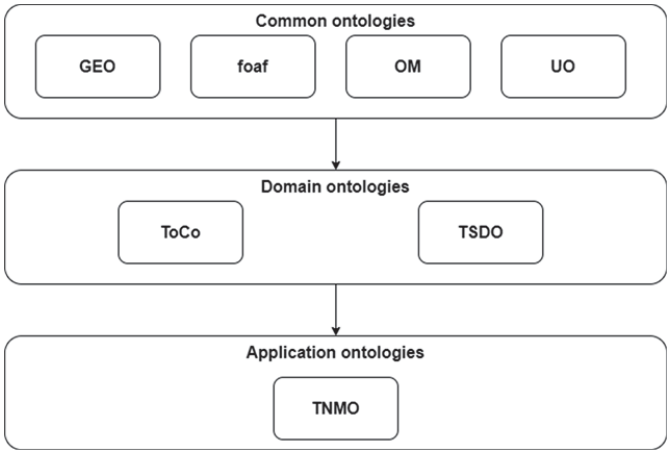

Fig. 3 Ontologies hierarchy

The following basic and domain ontologies are used in the monitoring platform ontology:

- Geo – Basic Geo (WGS84 lat/long) Vocabulary [11].
- FOAF - Friend of a Friend ontology [12].
- OM - Ontology of units of Measure (OM) [13].
- UO - Units of measurement ontology (UO) [14].
- ToCo Domain Ontology [15].
- TSDO Domain Ontology [16].

At the application level, it is proposed to use Telecommunication Network Monitoring Ontology (TNMO) [17].

Integration of domain-level ontology and application-level ontology is performed by semantic linking of entities in ontologies of different levels. For these purposes, the following entities are used in the domain-level ontology:

- Device,
- Link,
- Interface,
- Service,
- Data.

Integration is carried out with the following entities:

- User action,
- TN event,
- TN devices parameter.

The main classes of the TNMO ontology in OWL format are given below:

```
<!-- http://127.0.0.1/tnmo/Event -->
  <owl:Class rdf:about="http://127.0.0.1/tnmo/Event">
  <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:label>tnmo:Event</rdfs:label>
</owl:Class>

<!-- http://127.0.0.1/tnmo/Parameter_M -->
<owl:Class rdf:about="http://127.0.0.1/tnmo/Parameter_M">
  <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:label>tnmo:Parameter_M</rdfs:label>
</owl:Class>

<!-- http://purl.org/toco/Device -->
<owl:Class rdf:about="http://purl.org/toco/Device">
  <rdfs:label>net:Device</rdfs:label>
</owl:Class>

<!-- http://purl.org/toco/Service -->
<owl:Class rdf:about="http://purl.org/toco/Service">
  <rdfs:label>net:Service</rdfs:label>
</owl:Class>

<!-- http://purl.org/toco/User -->
<owl:Class rdf:about="http://purl.org/toco/User">
  <rdfs:label>net:User</rdfs:label>
</owl:Class>

<!-- http://127.0.0.1/tnmo/Request -->
<owl:Class rdf:about="http://127.0.0.1/tnmo/Reques">
  <rdfs:label>tnmo:Request</rdfs:label>
</owl:Class>

<!-- http://www.w3.org/2003/01/geo/wgs84_pos#Point -->
<owl:Class
rdf:about="http://www.w3.org/2003/01/geo/wgs84_pos#Point">
  <rdfs:label>geo:Point</rdfs:label>
</owl:Class>
```

B. Integration of monitoring systems segments for different TN segments based on ontologies unification

It is proposed to integrate various segments of the operator's network within the integrated TN model using the mechanism for combining used ontologies. The unification of ontologies occurs at the level of the application ontology - TNMO.

The integration mechanism by the example of the "Interface" entity is shown in Fig. 4.

The proposed mechanism implies that when different segments of the operator's network use different domain or application ontologies, rules for matching the entities of the ontologies of the network segments and the ontology of the monitoring platform are added to the TNMO ontology. If own data model is used instead of ontology using, the entities of the monitoring platform ontology are mapped with elements of their own data models at the level of data adapters. For both cases, the definition ***tnmo: sameAs*** is used, which specifies the correspondence of the classes of the various ontologies used by the network segments and the TNMO ontology.

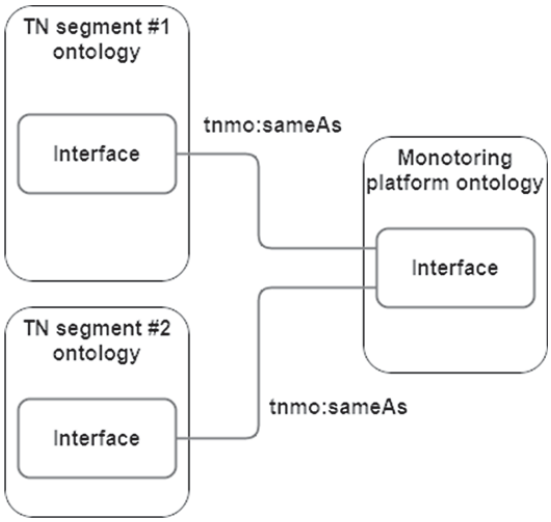

Fig. 4. TN segments ontologies integration

IV. CASE STUDY

As a practical example, the possibility of combining two different network segments within the TN monitoring platform, which use different monitoring systems built on different domain-level ontologies is demonstrated. Also, a report on monitoring data for two segments was obtained by executing a SPARQL query to the KG.

A. The task

Two segments of the telecom operator's network are connected to the monitoring platform. **The first segment** of the operator's TN is a cable Ethernet network that distributes Internet traffic to stationary addresses of subscribers. In **the second segment** of the network mobile Internet to users' devices is distributed. The users in both network segments are the same. Each user can operate several devices. It is required to generate reports on the total usage of Internet traffic by users for both network segments. To achieve the goal of the experiment, the following tasks were solved:

- Define the TN model structure for both segments in the KG format.
- Expand the definitions in the monitoring platform ontology.
- Develop the program for generating models of the TN segments structures and simulate data from monitoring systems by segments in RDF / XML format.
- Load the model data into the KG.
- Develop and execute the SPARQL queries that retrieve a report on Internet traffic usage by users for both segments of the TN from the KG.

The parameters of the experimental model are presented in Table II.

TABLE II. CASE STUDY MODELS PARAMETERS

Parameter	Value
Number of devices in each TN segment	1500
Number of TN users	1000
Frequency of internet traffic monitoring per hour	10
Duration of monitoring, hours	24

B. Knowledge graph structure

For the first TN segment the prefix TNSeg-1 is used when describing the structure of an RDF graph and ontology. For the second TN segment - TNSeg-2. The structure of the knowledge graph, including the static part and monitoring data for transmitted / received traffic for both network segments, is shown in Fig. 5.

C. Ontology extension

The monitoring platform ontology has been extended to integrate the two network segments. The modified definitions of TNMO ontology entities are presented below:

```
<!-- http://purl.org/toco/Device -->
  <owl:Class rdf:about="http://purl.org/toco/Device">
    <rdfs:label>net:Device</rdfs:label>
    <tnmo:sameAs>TNSeg-1:Device</tnmo:sameAs>
    <tnmo:sameAs>TNSeg-2:Device</tnmo:sameAs>
  </owl:Class>

  <!-- http://purl.org/toco/User -->
  <owl:Class rdf:about="http://purl.org/toco/User">
    <rdfs:label>net:User</rdfs:label>
    <tnmo:sameAs>TNSeg-1:User</tnmo:sameAs>
    <tnmo:sameAs>TNSeg-2:User</tnmo:sameAs>
  </owl:Class>

  <!-- http://purl.org/tnmo/Parameter_M -->
  <owl:Class rdf:about="http://purl.org/ tnmo/Parameter_M ">
    <rdfs:label>tnmo:Parameter_M</rdfs:label>
    <tnmo:sameAs>TNSeg-1: Parameter_M </tnmo:sameAs>
    <tnmo:sameAs>TNSeg-2: Parameter_M </tnmo:sameAs>
  </owl:Class>

  <!-- http://127.0.0.1/tnmo/sameAs -->
  <owl:ObjectProperty rdf:about="http:///127.0.0.1/tnmo/sameAs">
   <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
    <rdfs:domain rdf:resource="http://purl.org/toco/Device"/>
    <rdfs:domain rdf:resource="http://purl.org/toco/User"/>
    <rdfs:domain rdf:resource="http://purl.org/tnmo/Parameter_M"/>
    <rdfs:range rdf:resource="http:///127.0.0.1/TNSeg-1/Device"/>
    <rdfs:range rdf:resource="http:///127.0.0.1/TNSeg-2/Device"/>
    <rdfs:range rdf:resource="http:///127.0.0.1/TNSeg-1/User"/>
    <rdfs:range rdf:resource="http:///127.0.0.1/TNSeg-2/User"/>
    <rdfs:range rdf:resource="http:///127.0.0.1/TNSeg-1/Parameter_M"/>
    <rdfs:range rdf:resource="http:///127.0.0.1/TNSeg-2/ Parameter_M"/>
     <rdfs:label>tnmo:sameAs</rdfs:label>
  </owl:ObjectProperty>
```

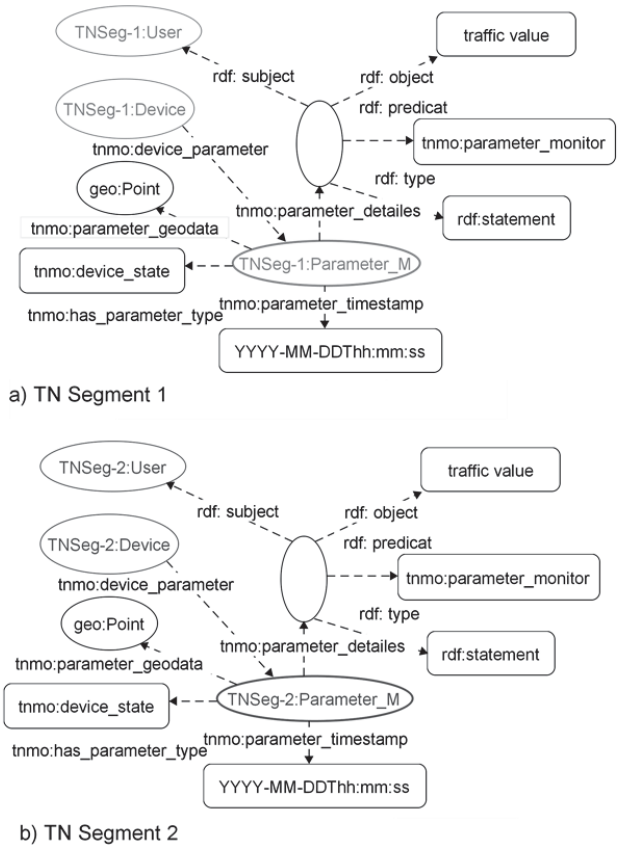

Fig. 5. Knowledge graph structure. a) TN Segment 1, b) TN Segment 2

D. *Monitoring reports*

In purpose of reporting ability demonstration, the following SPARQL queries were developed:

- Request #1 returns the top 5 lines of summarized traffic across network devices over the both TN segments.
- Request #2 returns the top 5 lines of summarized traffic across users over both TN segments.
- Request #3 returns the summarized traffic for selected user in Segment #1 of TN.
- Request #4 returns the summarized traffic for the selected device.

SPARQL REQUEST #1:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
PREFIX net: <http://purl.org/toco/>
PREFIX geo: <http://www.w3.org/2003/01/geo/>
PREFIX tnmo: <http://localhost/tnmo.owl>
PREFIX TNSeg-1: <http://127.0.0.1/TNSeg-1/>
PREFIX TNSeg-2: <http://127.0.0.1/TNSeg-2/>
SELECT ?Device (SUM(?Traffic_value) as ?Traffic)
WHERE
{
 ?a rdf:type "rdf:statement" .
 ?a rdf:object ?Traffic_value .
 ?a rdf:subject ?Users .
 ?Event <http://127.0.0.1/tnmo/parameter_detailes> ?a .
 ?Event <http://127.0.0.1/tnmo/device_parameter> ?Device .
 ?Users <http://127.0.0.1/tnmo/sameAs> ?Same_Users .
 }
GROUP BY ?Device
LIMIT 5
```

RESPONSE:

Device	***Traffic***
<http://127.0.0.1/TNSeg-2/Device_812/>	*519966526*
<http://127.0.0.1/TNSeg-2/Device_1295/>	*527801713*
<http://127.0.0.1/TNSeg-2/Device_1006/>	*519207569*
<http://127.0.0.1/TNSeg-2/Device_8/>	*517085750*
<http://127.0.0.1/TNSeg-2/Device_1255/>	*512740808*

SPARQL REQUEST #2:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
PREFIX net: <http://purl.org/toco/>
PREFIX geo: <http://www.w3.org/2003/01/geo/>
PREFIX tnmo: <http://localhost/tnmo.owl>
PREFIX TNSeg-1: <http://127.0.0.1/TNSeg-1/>
PREFIX TNSeg-2: <http://127.0.0.1/TNSeg-2/>
SELECT ?Users (SUM(?Traffic_value) as ?Traffic)
WHERE
{
 ?a rdf:type "rdf:statement" .
 ?a rdf:object ?Traffic_value .
 ?a rdf:subject ?Users .
 ?Event <http://127.0.0.1/tnmo/parameter_detailes> ?a .
 ?Event <http://127.0.0.1/tnmo/device_parameter> ?Device .
 ?Users <http://127.0.0.1/tnmo/sameAs> ?Same_Users .
 }
GROUP BY ?Users
LIMIT 5
```

RESPONSE:

Users	***Traffic***
<http://127.0.0.1/TNSeg-2/User_10/>	*541996421*
<http://127.0.0.1/TNSeg-2/User_100/>	*521316977*
<http://127.0.0.1/TNSeg-2/User_109/>	*1029826558*
<http://127.0.0.1/TNSeg-2/User_1000/>	*516152143*
<http://127.0.0.1/TNSeg-2/User_110/>	*1513253207*

SPARQL REQUEST #3:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
PREFIX net: <http://purl.org/toco/>
PREFIX geo: <http://www.w3.org/2003/01/geo/>
PREFIX tnmo: <http://localhost/tnmo.owl>
PREFIX TNSeg-1: <http://127.0.0.1/TNSeg-1/>
PREFIX TNSeg-2: <http://127.0.0.1/TNSeg-2/>
SELECT ?Users (SUM(?Traffic_value) as ?Traffic)
WHERE
{
  ?a rdf:type "rdf:statement" .
  ?a rdf:object ?Traffic_value .
  ?a rdf:subject ?Users .
  ?Users TNSeg-1:ID "56" .
  ?Event <http://127.0.0.1/tnmo/parameter_detailes> ?a .
  ?Event <http://127.0.0.1/tnmo/device_parameter> ?Device .

}
GROUP BY ?Users
```

RESPONSE:

Users	*Traffic*
<http://127.0.0.1/TNSeg-1/User_56/>	*515136903*

SPARQL REQUEST #4:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
PREFIX net: <http://purl.org/toco/>
PREFIX geo: <http://www.w3.org/2003/01/geo/>
PREFIX tnmo: <http://localhost/tnmo.owl>
PREFIX TNSeg-1: <http://127.0.0.1/TNSeg-1/>
PREFIX TNSeg-2: <http://127.0.0.1/TNSeg-2/>
SELECT ?Device (SUM(?Traffic_value) as ?Traffic)
WHERE
{
  ?a rdf:type "rdf:statement" .
  ?a rdf:object ?Traffic_value .
  ?a rdf:subject ?Users .
  ?Event <http://127.0.0.1/tnmo/parameter_detailes> ?a .
  ?Event <http://127.0.0.1/tnmo/device_parameter> ?Device .
  ?Device net:hasMAC "MAC_Seg-1_6" .

}
GROUP BY ?Device
```

RESPONSE:

Device	*Traffic*
<http://127.0.0.1/TNSeg-1/Device_6/>	*548343329*

The source code of Python script for the RDF/SQL data generation and all the datasets that are used are available in the public GitHub repository [18].

E. Performance evaluation

The performance of the described SPARQL requests execution comparison analysis results are shown in Table III. The requests execution time was analyzed for TN models with different sizes. Requests #1 and #2 are used to make full statistical traffic reports across all the devices and users. This type of requests is commonly executed quite rare. Taking this into consideration the time is acceptable. Requests #3 and #4 allow receive data for a selected device or user. This type of requests have high performance for all the model sizes and can be executed often.

TABLE III. PERFORMANCE EVALUATION

	Model #1 **Users per segment: 1 500** **Devices per segment: 1 000** **RDF triples: 5.193M**	**Model #2** **Users per segment: 15 000** **Devices per segment: 10 000** **RDF triples: 51.9M**	**Model #3** **Users per segment: 45 000** **Devices per segment: 30 000** **RDF triples: 155.57M**
RDF/XML load time, sec.	51.9	862.3	4948.9
Request #1 execution time	6sec, 186ms	2min, 4sec, 781ms	13min, 47sec, 219ms
Request #2 execution time	4sec, 528ms	1min, 59sec, 780ms	11min, 38sec, 355ms
Request #3 execution time	55ms	62ms	152ms
Request #4 execution time	53ms	61ms	187ms

V. CONCLUSION

The article proposes the new cloud platform for TN monitoring based on knowledge graphs. The architecture of the platform is designed to build operators' networks monitoring systems that solve a wide class of monitoring tasks, including modern ones that require use of integrated TN graph models [2]. The developed platform makes it possible to provide up-to-date analytics based on monitoring data for all network segments without expensive modernization of currently functioning monitoring systems. Low expenses are provided by using ontologies, that help to combine data of various network segments within one monitoring platform. Methods for different TN ontologies integration and TNMO ontology enrichment are proposed.

In the case study section, the practical task of combining different operator's network segments which use different monitoring systems and different ontologies is presented. In purpose of the task solving, the KG structure and the ontology were developed. The KG based on the developed ontology make it possible to build a single semantic model for different network segments and obtain a single report on monitoring data by executing one SPARQL query. For performance

evaluation, the proposed SPARQL queries have been executed on three KGs of different size. The performance analysis results show that the performance of generation of reports over all the TN devices population is acceptable and the performance of retrieving data for selected TN devices is high. Further development of the proposed monitoring platform assumes development of methods for automatic extending of the ontology when new networks or their segments are connected.

ACKNOWLEDGMENT

The research was supported by the state budget (project No. 0060-2019-0011).

REFERENCES

[1] Stanford University, “Network Monitoring Tools” Stanford University. [Online]. Available: http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html. [Accessed: Aug. 25, 2021]

[2] K. Krinkin, I. Kulikov, A. Vodyaho and N. Zhukova, "Architecture of a Telecommunications Network Monitoring System Based on a Knowledge Graph," 2020 26th Conference of Open Innovations Association (FRUCT), Yaroslavl, Russia, 2020, pp. 231-239, doi: 10.23919/FRUCT48808.2020.9087429.

[3] 7. Jeroen Johannes van der Ham. A semantic model for complex computer networks: the network description language, volume 3. Citeseer, 2010.

[4] David Cleary, Boris Danev, and Diarmuid O’Donoghue. Using ontologies to simplify wireless network configuration. In FOMI, 2005.

[5] Claudia Villalonga, Martin Strohbach, Niels Snoeck, Michael Sutterer, Mariano Belaunde, Ern¨o Kovacs, Anna V Zhdanova, Laurent Walter Goix, and Olaf Droegehorn. Mobile ontology: Towards a standardized semantic model for the mobile domain. In International Conference on Service-Oriented Computing, pages 248–257. Springer, 2007.

[6] Qianru Zhou, Ontology-driven knowledge based autonomic management for telecommunication networks: theory, implementation, and applications. Heriot-Watt University, 2018.

[7] Pedro PF Barcelos, Maxwell E Monteiro, Ricardo de M Sim˜oes, Anilton S Garcia, and Marcelo EV Segatto. Ootn-an ontology proposal for optical transport networks. In IEEE ICUMT, pages 1–7, 2009.

[8] Abdulbaki Uzun and Axel K¨upper. Openmobilenetwork: extending the web of data by a dataset for mobile networks and devices. In ACM ICSS, pages 17–24, 2012.

[9] K. Krinkin, A. Vodyaho, I. Kulikov and N. Zhukova, "Models of Telecommunications Network Monitoring Based on Knowledge Graphs," 2020 9th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2020, pp. 1-7, doi: 10.1109/MECO49872.2020.9134148.

[10] Kulikov I., Wohlgenannt G., Shichkina Y., Zhukova N. (2020) An Analytical Computing Infrastructure for Monitoring Dynamic Networks Based on Knowledge Graphs. In: Gervasi O. et al. (eds) Computational Science and Its Applications – ICCSA 2020. ICCSA 2020. Lecture Notes in Computer Science, vol 12254. Springer, Cham. https://doi.org/10.1007/978-3-030-58817-5_15.

[11] Basic Geo (WGS84 lat/long) Vocabulary. https://www.w3.org/2003/01/geo/

[12] Friend of a Friend (FOAF) ontology. http://xmlns.com/foaf/spec/

[13] Ontology of units of Measure (OM). http://purl.oclc.org/net/unis/ontology/sensordata.owl

[14] Units of measurement ontology (UO). http://purl.obolibrary.org/obo/uo.owl

[15] Zhou Q., Gray A.J.G., McLaughlin S. (2019) ToCo: An Ontology for Representing Hy-brid Telecommunication Networks. In: Hitzler P. et al. (eds) The Semantic Web. ESWC 2019. Lecture Notes in Computer Science, vol 11503. Springer, Cham. https://doi.org/10.1007/978-3-030-21348-0_33

[16] Xiuquan Qiao, Xiaofeng Li and Junliang Chen (March 30th 2012). Telecommunications Service Domain Ontology: Semantic Interoperation Foundation of Intelligent Integrated Services, Telecommunications Networks - Current Status and Future Trends, Jesus Hamilton Ortiz, IntechOpen, DOI: 10.5772/36794

[17] Telecommunication Networks Monitoring Ontology (TNMO): https://github.com/kulikovia/TNMO

[18] The GitHub repository: https://github.com/kulikovia/FRUCT-2021-2