HAL
open science

TRAILS: Extending TOSCA NFV profiles for liability management in the Cloud-to-IoT continuum

Yacine Anser, Chrystel Gaber, Jean-Philippe Wary, Sara Nieves, Matheu García, Samia Bouzefrane

To cite this version:

Yacine Anser, Chrystel Gaber, Jean-Philippe Wary, Sara Nieves, Matheu García, et al.. TRAILS: Extending TOSCA NFV profiles for liability management in the Cloud-to-IoT continuum. IEEE International Conference on Network Softwarization (NetSoft 2022), Jun 2022, Milan, Italy. 10.1109/NetSoft54395.2022.9844027 . hal-03768388

HAL Id: hal-03768388

https://hal.science/hal-03768388v1

Submitted on 3 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TRAILS: Extending TOSCA NFV profiles for liability management in the Cloud-to-IoT continuum

Yacine Anser
Orange & Cnam
Caen, France
yacine.anser@orange.com

Chrystel Gaber, Jean-Philippe Wary
Orange
Châtillon, France
{chrystel.gaber, jeanphilippe.wary}
@orange.com

Sara Nieves
Matheu García
University of Murcia
Murcia, Spain
saranieves.matheu@um.es

Samia Bouzefrane
CEDRIC Lab
Cnam
Paris, France
samia.bouzefrane@cnam.fr

***Abstract*—To address the growing amount of data generated by the Internet of Things (IoT), Network Functions Virtualization (NFV), 5G, Fog and Edge computing converge to form a Cloud-to-IoT continuum. This complex multi-layer architecture involves several actors among which responsibilities may be blurred. Existing profiles mostly describe deployment aspects and elude responsibility, accountability or liability characteristics. Moreover, the multiplicity of component profiles prevents uniform service management. This paper proposes TRAILS (sTakeholder Responsibility, AccountabIity and Liability deScriptor), an extension of the TOSCA NFV profile that merges the existing profiles and adds a description of the responsibilities and accountabilities of supply chain actors. This allows a uniform and liability-aware management of services involving IoT devices, fog, edge and cloud nodes. To show the usability of our model, we discuss the ecosystem around the generation of the proposed extension as well as its application in an ontology-based referencing module of a liability-aware service manager that we designed.**

***Index Terms*—Liability, Responsibility, Accountability, IoT, NFV, 5G, Cloud-to-IoT continuum**

I. Introduction

Statista[1] forecasts that by 2030 more than 50 billion Internet of Things (IoT) devices worldwide will connect to the Internet and generate a huge amount of data. Managing all these data while ensuring service provision and meeting ubiquity, reliability, high-performance, efficiency, and scalability criteria is a great challenge.

Lingen *et al.* [1] and Biswas *et al.* [2] claim it is crucial for the evolution of IoT services that IoT, Fog, Cloud, Network Function Virtualization (NFV) and Software-Defined Networks (SDN) converge. We call this convergence the Cloud-to-IoT continuum and we illustrate it in Fig. 1. Examples of such architectures and uniform management have been proposed by Vilalta *et al.* [3] and Lingen *et al.* [1].

Sharif *et al.* [4], Pan *et al.* [5] and Atzori *et al.* [6] mentioned this convergence and underline the need of providing uniform service management involving cloud, IoT and NFV, which requires combining existing descriptors such as MUD profiles [7] and NFV descriptor [8].

In addition, Gaber *et al.* [9] and Biswas *et al.* [2] highlight the difficulty to distribute responsibilities, and therefore liabilities, among stakeholders of this multi-actor architecture. In [10], the INSPIRE-5Gplus project conceptualizes a manifest that formalizes responsibilities of 5G services. INSPIRE-5Gplus [9] also proposes the Liability-Aware Service Manager (LASM), an extension of the ETSI NFV security manager [11], which takes into account responsibility[2], accountability [3] and liability [4] in the orchestration of 5G services by exploiting INSPIRE-5Gplus manifest.

Contributions. To address both challenges, we propose TRAILS (sTakeholder Responsibility, AccountabIity and Liability deScriptor), an extension of the TOSCA NFV profile [8] which complies with the INSPIRE-5Gplus manifest by leveraging existing profiles. We then illustrate its usability and relevance to model a wide range of components and services. We also evaluate their impact on scalability before discussing their relevance for managing liabilities in the context 5G services.

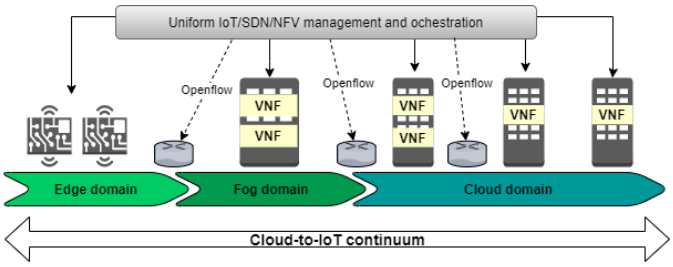

Fig. 1. Cloud-To-IoT Continuum

The authors would like to thank the anonymous reviewers of the paper and their shepherd Fluvio Valenza for their advices and comments on the article. The research leading to these results partly received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no 871808 (5G PPP project INSPIRE-5Gplus). The paper reflects only the authors' views. The Commission is not responsible for any use that may be made of the information it contains. We thank Kahina Lazri for her feedback on the ecosystem of VNF certification.

[1]https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/

[2]Responsibility refers to the notion of duty and corresponds to a task which needs to be performed while complying with a set of objectives.

[3]The accountability relationship binds an accountor to an accountee. The accountor is expected to perform a task and justify its outcomes to the accountee based on a set of technical evidence, design evidence and evidence at policy-level.

[4]Liability corresponds to accountability towards legislation. For example, in the case of contracts, both parties are required by the law to fulfil their end of the deal and to be able to demonstrate it. Penalties and Incentives can be decided when the provider fails or succeeds to comply with the set of objectives agreed between both parties.

Outline. This article is organized as follows. Section II gives an overview of the existing profiles and descriptors and highlights how they capture responsibilities, accountability and liability. Section III describes TRAILS, the proposed extension of TOSCA NFV profile that merges existing profiles and adds a description of the responsibilities, accountabilities and liabilities of supply chain actors. Section IV evaluates our contribution by comparing it to INSPIRE-5Gplus requirements, explaining how it can be used in a practical use case and by examining how TRAILS impacts scalability. Finally, section V discusses the advantages and shortcomings of TRAILS and VI concludes the paper.

II. Existing profiles

To our knowledge, no existing descriptor intends to modelize responsibility and liability relationships. Therefore, existing approaches do not fully cover responsibility, accountability and liability aspects. We demonstrate this by comparing them to INSPIRE-5Gplus manifest requirements, as illustrated in Section IV Table I.

A. INSPIRE-5Gplus liability manifest requirements

As defined in [10], a liability manifest should help supply chain stakeholders to explicitly express their *responsibilities* to which they commit and the usage conditions under which these commitments are valid (users responsibilities). Manifests should also allow users to assign themselves responsibilities through the definition of operation limitations. INSPIRE-5Gplus manifest requires supply chain stakeholders to sign their contribution to the manifest to materialize their commitment to their responsibilities. The clear attribution of responsibilities and their acceptance are binding in the same way as a contract, thus achieving *liability*. As such, INSPIRE-5Gplus manifest also contributes to *accountability* because they explicit what needs to be demonstrated by each stakeholder. INSPIRE-5Gplus also defines that a liability manifest should be *modular* enough to compose multiple components and capture accurately the relationships between stakeholders throughout the product's lifecycle, illustrated in Fig. 2. At each step of the product's lifecycle, the manifest is enriched with data useful for the next step. Typically, suppliers and validators fill in information regarding the product's content, guarantees, conditions of use. On their side, service providers add information related to the deployment and use in their own infrastructure. Finally, manifests should be *generic* for any type of network component, either IoT or VNF, in the Cloud or in the Edge.

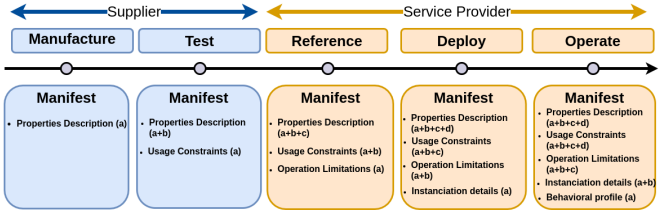

Fig. 2. INSPIRE-5Gplus manifest lifecycle

B. IoT profiles

With the IETF standard Manufacturer Usage Definition (MUD) [7], an IoT manufacturer can define the intended network behaviour of the device. In particular, MUD extends the Access Control Lists (ACLs) defined by the YANG data model as a means to deliver configuration to routers. The standard explicitly mentions that there is no manufacturer commitment and that it is the user's responsibility to ensure that IoT devices behave as intended.

Given that MUD does not offer the possibility of defining security properties in order to provide a more stringent approach, Sara Matheu *et al.* [12] [13] proposed an extended version of the MUD profile in which additional security aspects are taken into account. The format of the ACLs is extended to build an enhanced MUD profile with cryptographic algorithms, key usage, the maximum number of allowed connections and application-level authorization. With the MUD profile, the manufacturer does not take any responsibility for implementing controls on the components, it is up to the component's user to enforce them.

Based on the MUD standard, the National Institute of Standards and Technology (NIST) described a threat MUD file [14] which describes for a given threat the list of compromised domains that should be blocked to avoid possible attacks on IoT devices. Similarly to the original MUD, the thread MUD file assigns the responsibilities of implementing controls to the user.

The Software Updates for the Internet of Things (SUIT) [15] defines a way to deploy securely firmware updates in IoT devices. The SUIT manifest contains metadata of the firmware image, such as the date of the creation of the file, the behaviour of the IoT device while performing updates, the dependencies on other manifest files or the cryptographic information to validate the firmware.

C. VNF and NS descriptors

VNF and NS descriptors [8] [16] are used by VNF or NS providers to convey information related to their deployment or scaling such as requirements (resources, connectivity, interface), service topology (relationships and connections between virtual and physical network functions) or lifecycle. They can be modelled either with TOSCA [17] or YANG [18] but most available NFV MANagement and Orchestration (MANO) platforms support the TOSCA model. Both profiles describe how users can use VNFs and NSs but there is no notion of supply chain responsibility, accountability or liability.

III. Extension of the TOSCA NFV profile

We chose to extend TOSCA because it is a modelling language for defining portable deployment and automated management of services which is already commonly used to manage VNFs, NSs [19] and even IoT devices [20] [21]. It is also modular and enables to compose multiple components, thus achieving INSPIRE-5Gplus modularity requirement.

Background. The TOSCA metamodel defines a deployed service as an instance of a service template. A service template

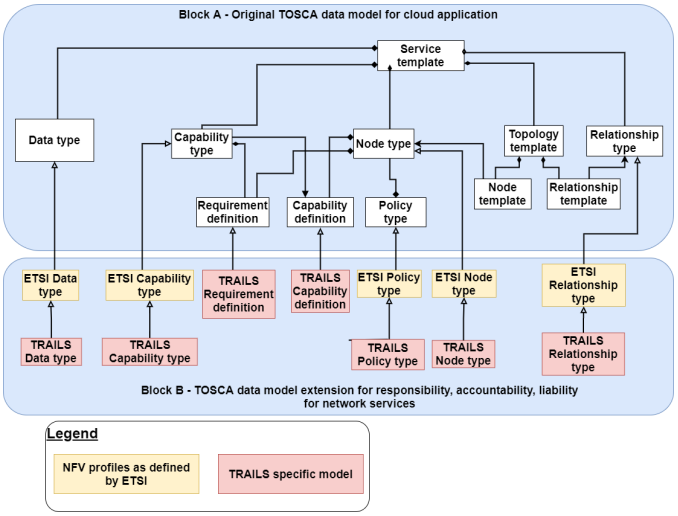

Fig. 3. Extension of the TOSCA NFV metamodel

is composed of a topology template, relationship template, node template and data type, as shown in Fig. 3 block A.

A topology is a directed graph where vertices represent components and arcs represent the network connections between components. A relationship template is an instance of relationship type that defines the semantics of a relationship (e.g. properties, valid capability target). Node templates are instances of node types that define the properties of a component, its requirements and capabilities. A node's lifecycle is managed with policies, they are defined through policy types and are evaluated to execute automatic triggers. All the characteristics mentioned above are defined using predefined data types.

The TOSCA topology template and its components are encapsulated in an archive called CSAR (Cloud Service ARchive). The archive contains at least two directories, the *TOSCA-Metadata* directory which includes entry information for processing a CSAR archive and the *Definition* directory which contains sources of the TOSCA application.

TRAILS extension. ETSI specifies an NFV specific data model using TOSCA metamodel. TRAILS extends this model to include responsibility, accountability and liability (Fig. 3 block B). For this, we introduce multiple elements. First, we updated *TRAILS Data type* to include the semantics required to describe a MUD profile, SUIT manifest or OpenAPI file. Second, *TRAILS Capability type*, which describes capabilities related to the security service. Third, *TRAILS policy type* describes the operation limitation which is a restriction imposed by an administrator before referencing the component. Fourth, *TRAILS Requirement definition*, which describes security requirements expressed thanks to security properties provided from the extended MUD profile. Fifth, *TRAILS Relationship type*, which binds two TRAILS's nodes through a security relationship and finally *TRAILS node*. To build a TRAILS CSAR archive, three new directories are required. The directory *Files* includes profiles and descriptors that can be referenced in the TRAILS data structure which facilitates the reuse of existing profiles. The directory *Certificates* contains all authors' certificates and *Signature* that includes file's signature. Finally, we add the file *Manifest.mf* file which lists all the files in the archive, the certificate of the LeadAuthor and the CSAR's signature.

Following Djikstra's separation of concern concept, we designed TRAILS node data structure, depicted in Fig. 4 so that each type of TRAILS node property describes a specific aspect of the component. The *header* provides an overview of the component or service by identifying its type, model and the entity which bears overall responsibility, the *LeadAuthor.Validation* indicates when the component was validated, by whom, the scope and the outcome of the validation. The Authors property lists all stakeholders of the component. The property *Commitment* describes the features promised by a given stakeholder, such as the Service Level Agreement (SLA). The property *UsageRecommendation* defines which conditions should be fulfilled to benefit at best of the component's features, such as the hardware and software dependencies, the way subservices should be combined or the component's expected Network behavior. Together, *Commitment* and *UsageRecommendation* describe complementary aspects of liability.

Attributability is ensured by the fact that properties are signed by their author / responsible party using a public/private key pair managed through a Public Key Infrastructure (PKI). We distinguish authors which take responsibility of a specific propriety and LeadAuthors which integrate multiple components and properties provided by other actors. As such, authors only sign the properties that it commits to whereas LeadAuthors sign all the properties in the scope of the integration it performed. To achieve this, we separate claims and properties in files that authors can sign individually. Then we regroup them in a CSAR archive that is signed by the relevant LeadAuthor.

The separation of concern is also demonstrated by the fact that TRAILS manifest can be used to study a network component under the angle of its topology (Fig. 12) or its responsibility chains (Fig. 13). The topology view is a directed graph where each component is a node and each link describes a connectivity link between two nodes. The responsibility view is a directed graph with a root (the final LeadAuthor which proposes the modeled service). Each vertex represents a couple of an Author and a Claim. Each directed edge represents a responsibility of an actor towards another one. Commitments are represented by an edge from a supplier towards its customer whereas *UsageRecommendations* are represented by an edge from a customer its supplier.

IV. Evaluation

To evaluate our proposal, we show that TRAILS complies with INSPIRE-5Gplus manifest requirements. Afterwards, we evaluate its semantics by describing how we used TRAILS to model existing network components and services. We also evaluate the impact of using TRAILS on scalability by evaluating its impact of convergence and stability.

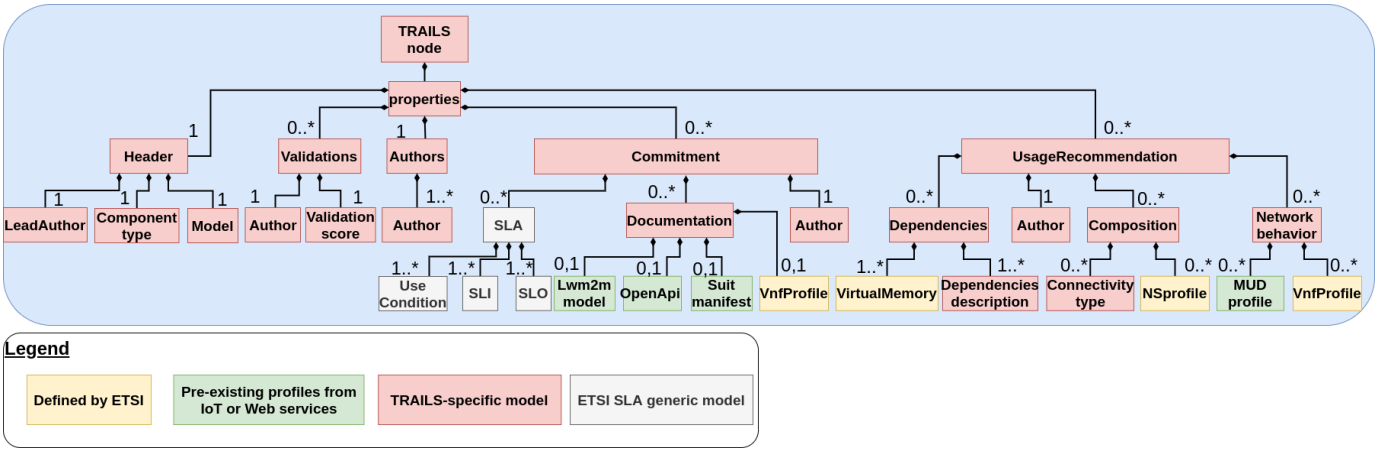

Fig. 4. High level structure of a TRAILS node

```
imports:
New_NFV_type.yaml
  topology_template:
     node_templates:
         component:
         type: tosca.nodes.nfv.trails
         properties:
          header:
           component_type: VNF
           model: VNF Firewall
           lead_author:
            name: Orange FR
            role: Operator
            ....
          authors :
          - name: Orange
            role: Validator
            country: France
            .....
            .....
          commitement:
          - $ref :
          '../VSRX-JuniperProperty.yaml
          #/commitement'
          usageRecommendation:
          - $ref :
          '../VSRX-JuniperUsageDescription.yaml
          #/usageRecommendation'
          requirements:
          - $ref : '....'
          capabilities :
          - $ref : '....'
```

Fig. 5. Extract of TRAILS topology template

A. Compliance with INSPIRE-5Gplus manifest

Table I compares TRAILS and the existing profiles based on the requirements of the INSPIRE-5Gplus manifest. Section II highlights that most of the existing models do not fulfil all the requirements such as responsibility, accountability, liability, modularity and genericity.

TRAILS fulfils the genericity criteria because it can be used for IoT devices, VNFs and NSs and leverages commonly used profiles that are relevant for each domain such as SUIT, MUD profiles, MUD extensions, VNF and NS descriptors.

TRAILS traces the responsibilities of each actor involved in the supply chain. Several stakeholders involved in the creation

Features	VNFD	NSD	MUD profile & extension	SUIT manifest	TRAILS
Responsibility	-	-	-	-	■
Accountability	□	□	□	□	■
Liability	□	□	□	□	■
Modularity	■	■	■	□	■
Genericity	□	□	□	□	■

■: the feature is supported
□: the feature is partially supported
-: the feature is not supported

TABLE I
COMPLIANCE WITH INSPIRE-5GPLUS MANIFEST REQUIREMENTS

of one service can define their responsibilities independently from each other. Supply chain providers can define responsibilities for themselves and their users. If users accept to use the service described by TRAILS, they can define responsibilities for themselves and include it as a new composite service. In this case, a composite TRAILS can be generated. As such, TRAILS fulfils at the same time the responsibility and modularity criteria. It should be noted that TRAILS also provides traceability of services. Liability is expressed in TRAILS by SLAs given their penalties and triggers. The signature of commitments, as well as the usage conditions, contribute to achieving the liability criteria.

At the same time, TRAILS ensures accountability by including SLA in the properties committed by each actor. In particular, the SLI, which comes up with metrics to monitor the service level, objectively measures performance against agreed objectives, and provides evidence that results have or have not been achieved before ensuring full accountability.

TRAILS complies with the manifest lifecycle described in Section II Fig. 2. During the manufacturing phase, the TRAILS profiles of multiple building blocks can be aggregated to form the profile of a new service. During the testing phase, validators can describe in TRAILS additional features, controls or usage conditions. During the referencing phase, a service operator can add operation limitations to comply with internal policies before adding the component to its catalogue. All these characteristics can then be used as shown in Section IV-B

to perform liability-aware service management. The version of the manifest presented in this paper does not standardize the data that service providers can add to the TRAILS profile in the deployment and operation phases.

B. Liability-Aware Service Management Use Case

1) Design of LASM Referencing and Ontology services: The LASM assists administrators in their management decisions to achieve the commitments of the administrated service [9]. The first module, *LASM Visualized Service* (LVS), deals with the presentation of services and data. The second one, *LASM Referecing Service* (LRS), catalogues the available network components and their TRAILS profiles. The third module, *LASM Ontology Service* (LOS) provides tools to evaluate a new component's TRAILS with regards to a referencing policy or research for a profile with specific features. The fourth, *LASM Analysis Service* (LAS), evaluates various metrics related to trust, responsibility or reputation of components and authors. Finally, the *LASM Orchestration & Deployment Service* (LODS) ensures the link with dedicated orchestrators or managers such as a MANO or an SD-IoT manager. Here, we present briefly our design of LRS and LOS.

LRS and LOS are complementary as LRS manages the catalogue of network components and LOS provides tools to reason on responsibility, accountability and liability aspects related to network components. LRS manages the synchronisation between the database and the ontology and therefore is the only one to expose an interface to external services.

As described by Fig. 6, both modules are implemented as REST web services using Django Rest framework. They communicate through a Kafka bus, a topic-based message broker. The ontology used by LOS is developed with owlready2, a module for ontology-oriented programming in python. We used Semantic Web Rule Language (SWRL) and Semantic Query Enhanced Web Rule Language (SQWRL) [22] to express respectively referencing policies and queries on the ontology content.

LRS centralizes all external requests and queries as illustrated in Fig. 6. For example, when the administrator adds a new component, LRS first validates the compliance of the profile to the TRAILS model by verifying the directory pattern, signatures, topology and syntax. Then, it requests LOS to evaluate the associated TRAILS profile with regards to a referencing policy. LRS stores TRAILS profiles in a database and associates them to a status, either "not evaluated", "Accepted" or "Rejected".

2) Use Case description: In this section, we illustrate how TRAILS, LRS and LOS can be used to take into account responsibility and accountability and liability in the management of a service in the Cloud-to-IoT continuum.

A Service Provider (SP) deploys a service on an infrastructure spanning from the Cloud to an IoT campus and managed by a Slice Provider (SLP). SLP subcontracts the management and monitoring of SP's IoT campus to the SubContractor (SC). Under normal conditions, SLP routes the packets collected from SP's devices in the IoT campus to SP's Cloud Delivery

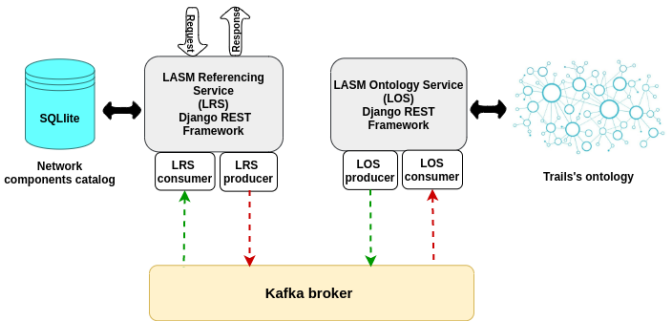

Fig. 6. LRS and LOS implementation

Network (CDN) application. SLP operates SP's slice with a basic assurance level where he commits for example to ensure a low loss of packets and an optimized level of energy consumption. In case an anomaly in the IoT devices is monitored, the contract between SP and SLP stipulates that SP shall put in place a video streaming service with a level of assurance high (e.g. providing proof of transit by specific nodes, high level of availability of the video streaming solution, guaranteed end-to-end isolation of the video streaming feed) to control and confirm the potential threat. Fig. 7 illustrates the scenario mentioned above.

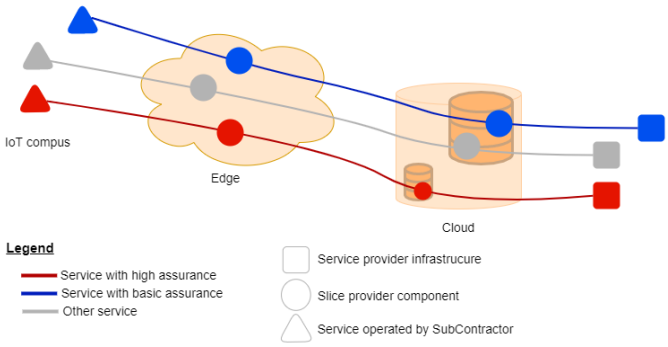

Fig. 7. SP's services

TRAILS and LASM assist SLP's administrator at three different stages, respectively the referencing of a network component, the component selection for orchestration or root cause analysis.

During the referencing stage of a network component, SLP can reference subcontracting solutions and ensure beforehand that they comply with its cybersecurity policies. In some cases, SLP cybersecurity policies will impose operational measures. If components are not compliant, SLP may decide to renegotiate a contract with the SC. The TRAILS profile of SC's IoT monitoring service can then be included in the contract between SLP and SC. Similarly, the TRAILS profiles of SLP's base service and video streaming service can be included in the contract between SP and SLP.

Fig.8 illustrates three rules from the operator's policy. The first rule indicates defines that any TRAILS for a component with a validation score "high" will be referenced.The second rule define a restriction about the energy consumption formu-

lated thanks to the energy-aware SLA proposed in [23]. It formally reflects the following statement: ”any network component that has as an energy consumption above 0.0018kw/h has the status Rejected”. In the last rule, the administrator assigns a scaling policy to a specific VNF model for which cybersecurity tests showed the need of scaling up resources such as CPU, RAM, energy. The scaling policy is based on the Anomaly Detection System (ADS) designed by *Lazri et al.* [24]. The system identifies the behaviours of a VNF before it leads to an SLA violation, which would enable to adopt proactive measures before a violation actually happens.This third rule will modify TRAILS by adding a new policy associated with the defined operation limitation as shown in Fig. 9.

```
- swrl_rule 1:
    name: R-High level of assurance
    src : "TRAILS(?t), validation(?v)
          , validation_score(?v,'high')
          , has_validation(?t,?v)
          -> value_Status(?t,'Accepted')"

 - swrl_rule 2:
    name: R-Restrictions on energy consumption
    src : "TRAILS(?t), SLA(?s)
          , has_slo_type(?s,'energy')
          , has_slo_value(?s,?x)
          , lessThan(?x,0.018)
          , has_sla(?t,?s)
          -> value_Status(?t,'Accepted')"

- swrl_rule 3:
    name: OL-Scaling policy
    src: "TRAILS(?t)
          , model(?m,'VSRX-Juniper')
          , has_model(?t,?m)
          -> value_Status(?t, 'Accepted')
          , policy(?p)
          , action(?a,'scaling_policy')
          , has_action(?p,?a)
          , has_policy(?t,?p)"
```

Fig. 8. Operator’s security policy

```
OperationLimitationPolicy :
 Description :
 ...
 ...
 Trigger:
     Event :
     ...
     Condition :
     ..
     Action :
     patch:
      description:
      implementation: /scripts/patch.sh
     scaling:
      description:
      implementation : /scripts/scaling_policy.sh
     configure:
      description:
      implementation: /scripts/black_list.sh
 ...
 ...
```

Fig. 9. TRAILS’s operation limitation policy

With LASM and TRAILs, SLP can select the components with the right characteristics to create services that comply with the contract binding SLP and SP. In our use case, SLP selects, using the query shown in Fig. 10, an IoT-camera with a high level of assurance, an SLA with an availability objective of 99%, and an SLI availability metric measured by the SC.

```
-sqwrl_rule 1 :
 name : Component selection
 src : "TRAILS(?t)
       , validation(?v)
       , SLA(?s)
       , validation_score(?v, 'high')
       , has_validation(?t,?v)
       , model(?m, 'IoT-camera')
       , has_model(?t,model)
       , has_sli_type(?s, 'Availability metric')
       , has_slo_type(?s, 'Availability')
       , has_slo_value(?t,99)
       , has_sla(?t,?s)
       -> sqwrl:select(?t)"
```

Fig. 10. Component selection query

LASM and TRAILS can complement a Root Cause Analysis (RCA) Service which identifies the most probable cause of an issue by estimating the liabilities with the help of TRAILS. LASM is not intended to impose automated penalties but to provide estimations for potential negotiations carried out by SLP’s jurists. For this, SLP can query the LOS model searching for a component, feature, responsible party involved in the issue. For example, the query represented in Fig. 11 search whether there is an actor which commits on the throughput.

```
-sqwrl_rule 2 :
name : Author identification
src : "TRAILS(?t), author(?a)
      , propertyDescription(?p)
      , features(?f,'debit')
      , has_features(?p,?f)
      , has_propertyDescription(?t,?p)
      -> sqwrl:select(?t,?a)"
```

Fig. 11. Author identification query

C. Semantic validation

To validate the semantic, we modeled a cellular blood pressure monitor IoT device from SmartMeter, an IoT Management Service (IMS) provided by Amazon Web Service (AWS), a Content Delivery Network Service (CNDS) provided by IBM and a Virtual Network Edge Service (VNES) provided by Equinix. Based on iBloodPressure’s user manual [25], we filled in the TRAILS *Header* with the full name of the device, the model and a description of the device which are indicated in the section *Introduction* of the manual. In TRAILS *validation* field, we list standards with which the device complies as stated in the section *Complied Standards List* of the manual. As usage recommendation, we referenced in the field *Network behavior* the MUD file of the device generated

by [26]. Based on AWS user and developer guide for the IMS [27], we retrieve general information to fill the TRAILS *Header* and listed in the section *validation* the complied standards indicated in the AWS IoT services and compliance such as International Standards Organization 27001 (ISO). The developer guide provide an openAPI file, we referenced it in the field *Documentation*. We then referenced the terms of the SLA indicated in [28] in the section *Commitment*. Similarly, we built a TRAILS for a CDN based on the information provided by IBM in [29]. We found the general information for *Header* under the section *About Content Delivery Networks*. We then listed the fact that IBM CND is PCI DSS compliant in the TRAILS *Validation* section. we filled in the TRAILS *commitment* with the openApi file provided in *CDN API reference* section.

Then, we built an example of TRAILS for an NS basing ourselves on the offer proposed by Equinix of a virtual network service provider [30]. We referenced Equinix *OpenApi* file and a SLA file written with WS-Agreement (Web Services Agreement Specification) language [31] in TRAILS *Commitment* section. The NSD of the service is referenced in the *UsageRecommendation* field to define usage condition and more particularly the protocol needed for the transport of the packets. For each example, there are at minimum two actors listed in the TRAILS, a validator and the Service Provider. For each author, we computed the signature of the claims to which it commits. Then, we generated a CSAR archive signed by each LeadAuthor.

Finally, we composed all of the services to build a new offer which corresponds to the use case described in section IV-B2. Figures 12 and 13 show respectively the topology of the composed service and its corresponding responsibilties share.

In comparison with TOSCA NFV profiles, TRAILS brings extra value such as the *Security Service* requirement and capability, which binds the *Virtual Gateway Service* and the *Virtual Firewall Service* through security relationship, the generic capacity that allows describing all the services provided by the SP using a unique model and the ability to highlight the responsibilities of each actor involved in the supply chain. In terms of memory size, the TOSCA NFV profiles of NS service reaches 36 bytes compared to 57 bytes for the TRAILS profiles, which represents an increase of 58.33%.

D. Scalability

As described in the use case, our target implementation requires the RCA module or the MANO to query the LRS in order to get a list of components which comply with specific criteria. So we expect that our impact on scalability will mostly correspond to the overhead required to perform a query. To quantify the impact of TRAILS on scalability, we break down this property into convergence, the time required to find a solution, and stability, how well the system performs when it is confronted to a large amount of data. All the experiments were performed five times on an Intel® Xeon® W-2133 Processor with 32 GBytes of available RAM and the results presented

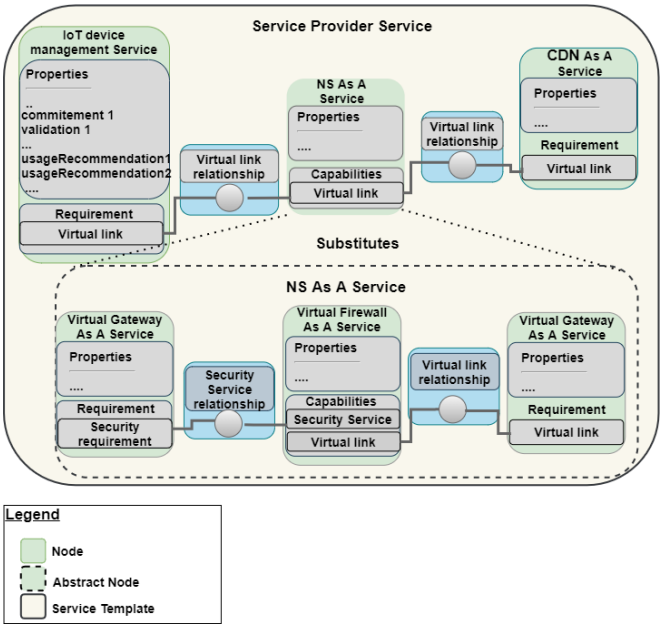

Fig. 12. TRAILS's topologic view

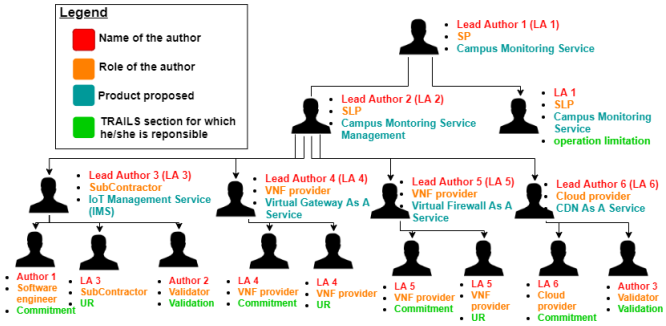

Fig. 13. TRAILS's responsability view

below corresponds to the average times measured over the 5 experiments.

We measure the impact of using TRAILS rather than TOSCA NFV by comparing the time required to query LOS-1, an ontology compatible with TOSCA NFV, and LOS-2, an ontology compatible with TRAILS. Each of them were populated with two individuals. LOS-1 containd two TOSCA NFV files which describe a VNF and an NS. LOS-2 contains two TRAILS that describes a VNF and a TRAILS that describes an NS. The required time to query LOS-1 to retrieve the VNF is 0.18 seconds and NS is 0.67 seconds whereas the same queries took respectively 0.21 seconds and 1.23 seconds o LOS-2. This represents an increase of 17% for the VNF and 84% for the NS.

To evaluate stability, we measure the evolution of the computation time to respond to a request which has a solution and a request without a solution depending on the size of the ontology. For this purpose, we progressively populated the ontology with clones of the TRAILS described in section IV-B that we modified so that the ontology considers they

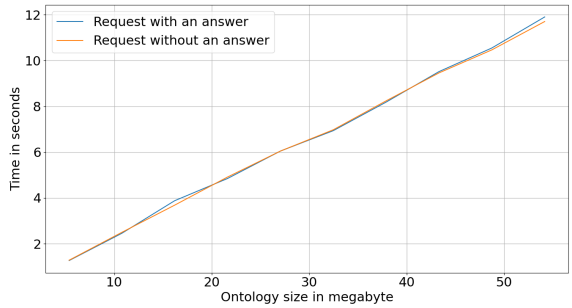

Fig. 14. Time evolution to perform a request to the regard to the size of the ontology

are unique individuals. At each step, we added 100 TRAILS until we reached 1000 TRAILS (54,18MB) since MANOs can have around this number of components in their catalog. The results displayed in figure 14 suggest that the time necessary to compute both types of requests follows linearly the size of the ontology.

V. Discussion

By design, TRAILS reflects the clauses of a contract as both are composed by obligations and conditions of use, measurable objectives, rewards and penalties.

TRAILS profiles can be used as a preventive measure, describing security policies and configurations to enhance the security of the device or to limit its attack surface, as the MUD, for example, is intended to do. Indeed, they can be generated as part of existing IoT certification process such as the industry-backed schemes GSMA IoT Security Assessment (IoTSA), the PSA Certified IoT Security Framework (PSA) or Eurosmart IoT Certification Scheme (IoTCS) or a state-backed certification frameworks such as Australia's IoT Security Trust Mark (STM). Indeed, the responses to the IoTSA or PSA questionnaires and IoTCS security profile can be mapped to the properties section of TRAILS and signed by vendors or evaluators depending on whether the certification relies on self or third party assessment. Current certification schemes mainly focus on the certification and evaluation processes, obtaining a security measurement of the device [32]. By linking the generation of the profile with the cybersecurity certification process, we benefit from the information obtained during the evaluation, recommending security measurements that could cope with the security issues detected.

The usage conditions and controls described in TRAILS can be also used as a preventive measure by analyzing during the operation time if the device is behaving as expected. In case of a deviation of the conditions imposed, it can be understood as a possible attack and appropriate measures should be applied.

In the same way, if a service depending on the device (services to which the device accesses or receives information) has been compromised by a threat, fast mitigation is a key to avoid major consequences. However, patches and updates delivered by the manufacturer can take days or even months. TRAILS provides a dynamic way to reconfigure the device, applying the needed countermeasures to protect it until the service is recovered from the attack. In particular, we can deny the access of the device to the compromised service and/or redirect the requests of the device to other similar and reliable services.

To our knowledge, no certification scheme evaluates the trustworthiness of network function validation. This task is traditionally performed by network operators through internally defined processes. Regarding performance evaluation, vendors provide for each network function, required resources that should be allocated to achieve a given service performance. For security assessment, security auditing is also conducted by operator security teams.

In the recent last years, the Network Equipment Security Assurance Scheme (NESAS) has been created by 3GPP and GSMA to accelerate the industrialization of network function security evaluation. NESAS group aims at defining a baseline security level that should be guaranteed by every network function vendor. Moreover, the group is responsible for defining test case scenarios to be validated by network vendors that belong to NESAS. While NESAS proved its benefit to both vendors and operators but also authorities as it provides an overall framework for security evaluation, an equivalent initiative that targets virtual network functions is still missing.

In addition to the software nature of virtual network functions, the multiplication of actors in the deployment of virtual networks makes it more challenging to define an overall framework for NFV validation. Indeed, in contrast to the legacy network ecosystem where the hardware is tightly coupled with the software, the operation of virtual network functions involves multiple actors including infrastructure providers, network vendors, and service operators. Security and performance evaluation in such a context requires strong liability management mechanisms.

Given that stakeholders sign their claims, TRAILS requires a Public Key Infrastructure and certificate. This is not the case today for ETSI NFV and would require setting up an organization to manage. Well-known and trusted organisms such as Global Platform, GSMA or ETSI could register supply chain actors and manage a Public Key Infrastructure. We propose to follow the example of the MUD file service hosted by Global Platform[5] or the eSIM certificate provisioning by GSMA[6].

TRAILS can be used in assurance continuity workflows as a way to rapidly share with users updated usage conditions in the case where vulnerabilities are disclosed. This scheme specifically includes an assurance continuity workflow in case a vulnerability is disclosed.

In terms of performance, we showed that the TRAILS model adds a significant amount of information, which may significantly impact convergence, especially if the ontology is populated with complex multi-actor and multi-layer network

[5]https://globalplatform.org/iotopia/mud-file-service/
[6]https://www.gsma.com/esim/gsma-root-ci/

services. However, we also showed that the system seems stable given that the time required to perform a query seems to follow linearly the size of the ontology. Further works could examine whether it is possible to optimize the ontology or the data structure to improve these performances.

VI. Conclusion

In this paper, we present TRAILS, an extension of the NFV TOSCA profile that complies with the INSPIRE5G-plus manifest by merging the existing profile that coexist in the Cloud-to-IoT continuum and by including responsibility, accountability and liability. We evaluated TRAILS by showing how it can be used with two implemented modules of a Liability-Aware Service Management system (LASM). Then, we demonstrated that the semantics defined to express TRAILS manifest are sufficient to model a wide range of components and services. Finally, we evaluated the impact of the proposal on the scalability of the tools which will query the catalog of TRAILS manifests exposed by the LASM. Finally, we discussed our observations and the place of TRAILS in the ecosystem. We concluded that the data structure adds a significant amount of information compared to TOSCA ETSI NFV profiles which may impact convergence depending on the complexity of the services considered. Yet, TRAILS can model a wide variety of situations and complex situations that cannot be model otherwise.

References

[1] F. van Lingen, M. Yannuzzi, A. Jain, R. Irons-Mclean, O. Lluch, D. Carrera, J. L. Perez, A. Gutierrez, D. Montero, J. Marti, R. Maso, Rodriguez, and J. Pedro, "The unavoidable convergence of nfv, 5g, and fog: A model-driven approach to bridge cloud and edge," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 28–35, 2017.

[2] A. R. Biswas and R. Giaffreda, "Iot and cloud convergence: Opportunities and challenges," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 375–376.

[3] R. Vilalta, V. Lopez, A. Giorgetti, S. Peng, V. Orsini, L. Velasco, R. Serral-Gracia, D. Morris, S. De Fina, F. Cugini, P. Castoldi, A. Mayoral, R. Casellas, R. Martinez, C. Verikoukis, and R. Munoz, "Telcofog: A unified flexible fog and cloud computing architecture for 5g networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 36–43, 2017.

[4] I. Alam, K. Sharif, F. Li, Z. Latif, M. M. Karim, S. Biswas, B. Nour, and Y. Wang, "A survey of network virtualization techniques for internet of things using sdn and nfv," *ACM Comput. Surv.*, vol. 53, no. 2, apr 2020.

[5] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2018.

[6] L. Atzori, J. L. Bellido, R. Bolla, G. Genovese, A. Iera, A. Jara, C. Lombardo, and G. Morabito, "SDN&NFV contribution to IoT objects virtualization," *Comput. Networks*, vol. 149, pp. 200–212, Feb 2019.

[7] E. Lear, R. Droms, and D. Romascanu, *RFC 8520 - Manufacturer Usage Description Specification*, IETF Std., Mar. 2019.

[8] "Etsi gs nfv-ifa 011 : Network functions virtualisation (nfv) management and orchestration vnf packaging specification."

[9] C. Gaber, J. S. Vilchez, G. Gür, M. Chopin, N. Perrot, J.-L. Grimault, and J.-P. Wary, "Liability-aware security management for 5g," in *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, pp. 133–138.

[10] J. Ortiz, R. Sanchez-Iborra, J. B. Bernabe, A. Skarmeta, C. Benzaid, T. Taleb, P. Alemany, R. Muñoz, R. Vilalta, C. Gaber, J.-P. Wary, D. Ayed, P. Bisson, M. Christopoulou, G. Xilouris, E. M. de Oca, G. Gür, G. Santinelli, V. Lefebvre, A. Pastor, and D. Lopez, "Inspire-5gplus: Intelligent security and pervasive trust for 5g and beyond networks," ser. ARES '20. New York, NY, USA: Association for Computing Machinery, 2020.

[11] "Network functions virtualisation (nfv) release 3; security; identity management and security specification."

[12] S. Matheu, J. Hernandez-Ramos, S. Perez, and A. Skarmeta, "Extending MUD profiles through an automated IoT Security Testing Methodology," *IEEE Access*, 2019.

[13] S. N. Matheu, A. Robles Enciso, A. Molina Zarca, D. Garcia-Carrillo, J. L. Hernández-Ramos, J. Bernal Bernabe, and A. F. Skarmeta, "Security architecture for defining and enforcing security profiles in DLT/SDN-Based IoT systems," *Sensors*, vol. 20, no. 7, p. 1882, 2020.

[14] M. Souppaya, D. Montgomery, W. Polk, M. Ranganathan, D. Dodson, W. Barker, S. Johnson, A. Kadam, C. Pratt, D. Thakore *et al.*, "Securing small-business and home internet of things (iot) devices: Mitigating network-based attacks using manufacturer usage description (mud)," *Special Publication (NIST SP) - 1800-15*, 2021.

[15] B. Moran, H. Tschofenig, and H. Birkholz, *A Concise Binary Object Representation (CBOR)-based Serialization Format for the Software Updates for Internet of Things (SUIT) Manifest*, IETF Std., Jul. 2021.

[16] "Network functions virtualisation (nfv) release 3; management and orchestration; network service templates specification."

[17] "Network functions virtualisation (nfv) release 3; protocols and data models; nfv descriptors based on tosca specification."

[18] "Network functions virtualisation (nfv) release 2 protocols and data models nfv descriptors based on yang specification."

[19] Y.-M. Hung, S.-C. Chien, and Y.-Y. Hsu, "Orchestration of nfv virtual applications based on tosca data models," in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2017, pp. 219–222.

[20] A. C. F. da Silva, U. Breitenbücher, K. Képes, O. Kopp, and F. Leymann, "OpenTOSCA for IoT: Automating the deployment of IoT applications based on the mosquitto message broker," in *Proceedings of the 6th International Conference on the Internet of Things*, ser. IoT'16. Association for Computing Machinery, pp. 181–182.

[21] F. Li, M. Vögler, M. Claeßens, and S. Dustdar, "Towards automated iot application deployment by a cloud-based approach," in *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications*, 2013, pp. 61–68.

[22] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "Swrl: A semantic web rule language combining owl and ruleml," World Wide Web Consortium, W3C Member Submission, 2004.

[23] Y. Anser, J.-L. Grimault, S. Bouzefrane, and C. Gaber, "Energy-aware service level agreements in 5g NFV architecture," in *8th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, pp. 377–382.

[24] C. Sauvanaud, M. Kaâniche, K. Kanoun, K. Lazri, and G. Da Silva Silvestre, "Anomaly detection and diagnosis for cloud services: Practical experiments and lessons learned," vol. 139, pp. 84–106.

[25] Resources. [Online]. Available: https://smartmeterrpm.com/resources/

[26] A. Hamza, D. Ranathunga, H. H. Gharakheili, M. Roughan, and V. Sivaraman, "Clear as MUD: Generating, validating and applying IoT behaviorial profiles (technical report)." [Online]. Available: http://arxiv.org/abs/1804.04358

[27] AWS IoT device management documentation.

[28] AWS IoT device management SLA. [Online]. Available: \url{https://aws.amazon.com/fr/iot-device-management/sla/}

[29] Content delivery network docs. [Online]. Available: \url{https://cloud.ibm.com/docs/CDN/cloud.ibm.com/docs/cdn/getting-started.html}

[30] Cloud edge services & edge network management | equinix. [Online]. Available: https://www.equinix.com/services/digital-infrastructure-services/network-edge

[31] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, "Web services agreement specification (ws-agreement)," *Global Grid Forum*, vol. 2, 01 2004.

[32] S. N. Matheu, J. L. Hernández-Ramos, A. F. Skarmeta, and G. Baldini, "A Survey of Cybersecurity Certification for the Internet of Things," *ACM Computing Surveys*, vol. 53, no. 6, pp. 1–36, 2 2021.