

OPSAWG  
Internet-Draft  
Intended status: Standards Track  
Expires: 24 April 2025

B. Claise  
J. Quilbeuf  
Huawei  
D. Lopez  
I. Dominguez  
Telefonica I+D  
T. Graf  
Swisscom  
21 October 2024

## A Data Manifest for Contextualized Telemetry Data draft-ietf-opsawg-collected-data-manifest-05

### Abstract

Network platforms use Model-driven Telemetry, ~~and in particular such as -YANG-~~ Push<sub>r</sub> to continuously stream information, including both counters and state information. This document ~~documents describes~~ the metadata that ensure that the collected data can be interpreted correctly. This document specifies the Data Manifest, composed of two YANG data models (the Platform Manifest and the Data Collection Manifest). These YANG modules are specified at the network (~~e.g.~~, controller) level to provide a model that encompasses several network platforms. The Data Manifest must be streamed and stored along with the data, up to the collection and analytics ~~systems~~ in order to keep the collected data fully exploitable by the data scientists ~~and relevant tools~~.

**Commenté [MB1]:** Other tools are widely used as well.

**Commenté [MB2]:** This is only an example and not the definite place to make use of the models

**Commenté [MB3]:** There might be many in a network

### Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 24 April 2025.

### Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Use Cases . . . . .	4
1.1.1. Network Analytics . . . . .	4
1.1.2. New Device Onboarding . . . . .	5
1.1.3. Data Mesh Principles in Networking . . . . .	5
2. Terminology . . . . .	6
3. Platform Manifest . . . . .	6
3.1. Overview of the Model . . . . .	6
3.2. YANG module ietf-platform-manifest . . . . .	9
4. Data Collection Manifest . . . . .	13
4.1. Overview of the Model . . . . .	13
4.2. YANG module ietf-data-collection-manifest-statistics . . . . .	15
4.3. YANG module ietf-data-collection-manifest . . . . .	17
5. Data Manifest and the Collected Data . . . . .	21
5.1. Collecting the Data Manifest . . . . .	22
5.2. Mapping Collected Data to the Data Manifest . . . . .	23
5.3. Operational Considerations . . . . .	24
6. Example . . . . .	24
7. Security Considerations . . . . .	26
8. IANA Considerations . . . . .	26
9. Contributors . . . . .	26
10. Open Issues . . . . .	26
11. Normative References . . . . .	27
12. Informative References . . . . .	28
Appendix A. title="Example of use based on MDT" . . . . .	29
Appendix B. Changes between revisions . . . . .	32
Appendix C. Generating YANG Tree Diagrams . . . . .	35
C.1. YANG Tree Diagram of the Platform Manifest . . . . .	35
C.2. YANG Tree diagram of the Data Collection Manifest . . . . .	37
Acknowledgements . . . . .	38
Authors' Addresses . . . . .	38

## 1. Introduction

Network platforms use Model-driven Telemetry (MDT), and in e.g., particular YANG-Push [RFC8641], to continuously stream information, including both counters and state information.

**Commenté [MB4]:** Consider introducing this first as this term is not used in, e.g., rfc9232, etc.

This document specifies what needs to be kept as metadata (i.e., the Data Manifest) to ensure that the collected data can still be interpreted correctly throughout the collection and network analytics toolchain. When streaming YANG-structured data with YANG-Push [RFC8641], there is a semantic definition in the corresponding YANG module definition. This is the semantic information for the collected data nodes: While this semantic is absolutely required to correctly decode and interpret the data, understanding the network

**Commenté [MB5]:** Not introduced at this stage. The next para introduces the concept.

platform and collection environment contexts information is equally important to interpret the data.

This document ~~proposes-introduces~~ the Data Manifest, which is composed of two YANG data models, namely, the Platform Manifest and the Data Collection Manifest, in order to keep the collected data exploitable by the data scientists ~~and relevant tools~~.

The Platform Manifest contains information characterizing the platform streaming the telemetry information, while the ~~the~~ Data Collection Manifest contains the required information to characterize how and when the telemetry information was metered.

The two ~~proposed~~ YANG modules ~~in the Data Manifest~~ do not expose many new information but rather define what should be exposed by a platform streaming or storing telemetry data. Some related YANG modules have been specified to retrieve the platform capabilities ~~such as~~:

- \* [The IETF-YANG Library](#) [RFC8525].
- \* ["YANG Modules Describing Capabilities for Systems and Datastore Update Notifications"](#) [RFC9196] for the platform capabilities regarding the production and export of telemetry data.
- \* [\[I-D.claise-netconf-metadata-for-collection\]](#), which is based on the previous draft to define the optimal settings to stream specific items (i.e., per path).

**Commenté [MB6]:** To match the title used in these RFCs

**Commenté [MB7]:** Which one?

**Commenté [MB8]:** I would remove this individual draft, especially that it is expired.

These related YANG modules are important to discover the capabilities before applying the telemetry configuration (such as on-change ~~subscription~~).

Some of their content is part of the context for the streamed data.

This ~~documents-document~~ covers only metadata about the collection context for the telemetry. The collected data is likely to be transformed into usable indicators for the network. The list of such transformation ~~operationseparation~~ applied to the data is often called data lineage. Supplying the data lineage for the computed indicators is out of scope of this document.

We ~~This document first presents~~ the module for the Platform Manifest in Section 3 and then the module for the Data Collection Manifest in Section 4. The full Data Manifest is obtained by combining these two modules. We ~~explain in~~ Section 5 ~~explains~~ how the Data Manifest can be retrieved and how collected data is mapped to the Data Manifest.

**Commenté [MB9]:** Already stated when introduced these modules above. May be simpler to cite these sections in that text and the delete this sentence.

## 1.1. Use Cases

### 1.1.1. Network Analytics

**Commenté [MB10]:** I would position this as a standalone section out of the intro.

Streamed information from network platforms is used for network

~~analytics, incident detections, and in the closed control loop for network automation~~ end closed loop automation. This streamed data can be stored in a database (sometimes called a big data lake) for further analysis.

**Commenté [MB11]:** Consider adding an informative ref to draft-ietf-nmop-terminology where these terms are defined

**Commenté [MB12]:** To be consistent with the use in RFC9232

As an example, a database could store a time series representing the evolution of a specific counter collected from a network platform. When analyzing the data, ~~the-a~~ network operator/data scientist must understand the context information for these data:

- \* This counter definition, ~~typically as defined~~ in the YANG model.
- \* The network platform ~~specific~~ vendor, model, and OS.
- \* The collection parameters.

Characterizing the source used for producing the data (vendor, platform, and OS) is useful to complement the data. As an example, knowing the exact data source software specification might reveal a particularity in the observed data, explained by a specific bug, a specific bug fix, or simply a particular specific behavior. This is also necessary to ensure the reliability of the collected data. On top of that, in particular for YANG-Push [RFC8641], it is crucial to know the set of YANG modules supported by the platform, along with their deviations. In some cases, there might even be some backwards incompatible changes in native modules ~~(i.e., vendor proprietary modules)~~ between one OS version to the next one. This information is captured by ~~means of~~ the proposed Platform Manifest [\(Section 3\)](#).

From a collection parameters point of view, the data scientists analyzing the collected data must know ~~that whether~~ the counter was requested

from the network platform as on-change or at specific cadence [\[RFC8641\]](#).

Indeed, an on-change collection explains why there ~~might be is~~ a single value as opposed to a time series. In case of periodic collection, this exact cadence might not be observable in the time series. Indeed, this time series might report some values as 0 or might even omit some values. The reason for this behavior might be diverse: the network platform was under stress, with a too small observation period, compared to the minimum-observed-period [I-D.claise-netconf-metadata-for-collection]. Again, ~~k~~Knowing the conditions under which the counter was collected and streamed (along with the platform details) help drawing the right conclusions. As an example, taking into account the value of 0 might lead to a wrong conclusion that the counter dropped to zero.

This document specifies the Data Collection Manifest, which contains the required information to characterize how and when the telemetry information was metered.

The goal of the ~~current~~ document is to define what needs to be kept as metadata (i.e., the Data Manifest) to ensure that the collected data can still be interpreted correctly.

**Commenté [MB13]:** This is already captured in the introduction. I would simplify and delete this redundant text.

### 1.1.2. New Device Onboarding

When a new device is onboarded, operators ~~must have to make checksure~~ that the new device streams data, ~~e.g.,~~ with YANG-Push, that the telemetry data is the right ones, that the data is correctly ingested in the collection system, and finally that the data can be analyzed (compared with other similar devices). For the last point, the Data Manifest, which must be linked to the data up to the collection and analytics systems, contains ~~all~~ the relevant information.

**Commenté [MB14]:** «All» or the «minimum required»?

### 1.1.3. Data Mesh ~~Principles~~ in Networking

The concept behind the data mesh <https://www.datamesh-architecture.com/> are:

**Commenté [MB15]:** Is this really a «use case»?

The text below says that data manifest an data collection adheres to the DaP principle.

- \* ~~Principle of~~-Domain Ownership: Architecturally and organizationally align business, technology, and analytical data, following the line of responsibility. ~~Here, t~~The Data Mesh principles adopt the boundary of bounded context to individual data products where each domain is responsible for (and owns) its data and models.
- \* ~~Principle of~~-Data as a Product: The “Domain” owners are responsible to provide the data in useful way (discoverable through a catalog, addressable with a permanent and unique address, understandable with ~~well-defined~~well-defined semantics, trustworthy and truthful, self-describing for easy consumption, interoperable by supporting standards, secure, self-contained, etc.) and should treat consumers of that data as customers. It requires and relies on the “Domain Ownership” principle.
- \* ~~Principle of~~-Self-serve Data Platform: This fosters the sharing of cross-domain data in order to create extra value.
- \* ~~Principle of~~-Federated Computational Governance: Describes the operating model and approach to establishing global policies across a mesh of data products.

**Commenté [MB16]:** I would transform this to a REF entry

The most relevant concept for this document is the “Data as a Product” principle. The Data Manifest fulfills this principle as the two YANG data models, Platform Manifest and the Data Collection Manifest, along with the data, provide all the necessary information in a self-describing way for easy consumption.

## 2. Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “NOT RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Data Manifest: ~~all t~~The necessary data required to interpret ~~thea~~ telemetry information.

Platform Manifest: part of the Data Manifest that completely characterizes the platform producing the telemetry information.

Data Collection Manifest: part of the Data Manifest that completely characterizes how and when the telemetry information was metered.

### 3. Platform Manifest

#### 3.1. Overview of the Model

Figure 1 contains the YANG tree diagram [RFC8340] of the `"ietf-platform-manifest"` module. The tree diagram is obtained by reusing existing modules, such as `"ietf-yang-library"` [RFC8525] using the YANG mount mechanism [RFC8528]. In Additionally addition to the YANG module in

Section 3.2, we this document specify specifies the YANG-library instance to be mounted. We explain in Appendix C explains how the YANG tree is obtained.

```
module: ietf-platform-manifest
  +-ro platforms
    +-mp platform* [id]
      +-ro id                  string
      +-ro name?               string
      +-ro vendor?              string
      +-ro vendor-pen?          uint32
      +-ro software-version?    string
      +-ro software-flavor?    string
      +-ro os-version?          string
      +-ro os-type?              string
      +-ro yang-library/
        | +-ro module-set* [name]
        | | +-ro name      string
        | | +-ro module* [name]
        | | | +-ro name      yang:yang-identifier
        | | | +-ro revision?  revision-identifier
        | | | +-ro namespace   inet:uri
        | | | +-ro location*  inet:uri
        | | | +-ro submodule* [name]
        | | | | +-ro name      yang:yang-identifier
        | | | | +-ro revision?  revision-identifier
        | | | | +-ro location*  inet:uri
        | | | | +-ro feature*   yang:yang-identifier
        | | | | +-ro deviation* -> ../../module/name
        | | | +-ro import-only-module* [name revision]
        | | | | +-ro name      yang:yang-identifier
        | | | | +-ro revision  union
        | | | | +-ro namespace  inet:uri
        | | | | +-ro location*  inet:uri
        | | | | +-ro submodule* [name]
        | | | | | +-ro name      yang:yang-identifier
        | | | | | +-ro revision?  revision-identifier
        | | | | | +-ro location*  inet:uri
      | +-ro schema* [name]
```

**Commenté [MB17]:** As this may change for a given platform, how versions are tracked over time (upon change, etc.). The id is not sufficient to identify the collection context.

**Commenté [MB18]:** Seems like a mix of hw/system info.

RFC7317

```
  +-ro platform
    +-ro os-name?    string
    +-ro os-release? string
    +-ro os-version? string
    +-ro machine?    string
RFC8348
  +-ro hardware-rev?  string
  +-ro firmware-rev? string
  +-ro software-rev? string
```

You may explain why reusing these modules or parts of these modules is not an option?

Grabbing data nodes from system/hw models has the merit to ease capturing required context info.

```

|   |   +-+ro name                      string
|   |   +-+ro module-set*                -> ../../module-
set/name
|   |   +-+ro deprecated-nodes-implemented? boolean
|   |   +-+ro obsolete-nodes-absent?    boolean
|   +-+ro datastore* [name]
|   |   +-+ro name      ds: datastore-ref
|   |   +-+ro schema     -> ../../schema/name
|   +-+ro content-id   string
x--+ro modules-state/
    +-+ro module-set-id   string
    x--+ro module* [name revision]
        x--+ro name          yang:yang-identifier
        x--+ro revision       union
        x--+ro schema?        inet:uri
        x--+ro namespace       inet:uri
        x--+ro feature*       yang:yang-identifier
        x--+ro deviation* [name revision]
            | x--+ro name      yang:yang-identifier
            | x--+ro revision   union
        x--+ro conformance-type enumeration
        x--+ro submodule* [name revision]
            x--+ro name          yang:yang-identifier
            x--+ro revision       union
            x--+ro schema?        inet:uri

```

Figure 1: YANG tree diagram for ietf-platform-manifest module

The YANG module actually contains a list of Platform Manifests (in 'platforms/platform'), indexed by the identifier of the platform. That identifier should be defined by the network manager so that each platform has a unique ididentifier. There are several ongoing drafts documents about managing the inventory of the network

[I-D.ietf-ivy-network-inventory-yang], [I-D.havell-nmop-digital-map]  
based on [RFC8345]. The platform -identifer should be the same as

**Commenté [MB19]:** Not sure about this one

the identifier used in these inventoriesdrafts or the node-'node-id' in [RFC8345]. As an example, the identifier could be the 'sysname' from the "ietf-notification" module presented in [I-D.tgraf-netconf-notif-sequencing]. The scope of this module is the scope of the data collection, i.e., a given network, therefore it contains a collection of Platform Manifests, as opposed to the device scope, which would contain a single Platform Manifest.

**Commenté [MB20]:** This is not an inventory doc.

I would delete the mention of this individual draft.

**Commenté [MB21]:** Please explicit which one

The Platform Manifest is identified-characterized by a set of parameters ('name', 'software-version', 'software-flavor', 'os-version', 'os-type') that are aligned with the YANG Catalog [www.yangcatalog.org](http://www.yangcatalog.org) [I-D.clacla-netmod-model-catalog] so that the YANG Catalog could be used to retrieve the YANG modules a posteriori. The vendor of the platform can be identified via its name 'vendor' or its PEN number 'vendor-pen', as described in [RFC9371].

The Platform Manifest also includes the contents of the YANG Library [RFC8525]. That module set is particularly useful to analyze the xpath filters, as they are based on module names (see [RFC8639], page

47) .

The Platform Manifest is obtained by specifying the new fields defined above and mounting the YANG library module, along with the YANG Revisions augmentations. Thus, the YANG Library part is not repeated in the YANG module for the Platform Manifest.

### 3.2. YANG module ietf-platform-manifest

We provide in this Section defines the code of the "ietf-platform manifest" YANG module. Additionally, we this section provides the extension data file for YANG schema mount. The platform manifest MUST conform to the model obtained by combining these two specifications.

```
<CODE BEGINS> file "ietf-platform-manifest@2024-07-02.yang"
module ietf-platform-manifest {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-platform-manifest";
    prefix p-mf;

    import ietf-yang-schema-mount {
        prefix yangmnt;
        reference
            "RFC_8528: YANG Schema Mount";
    }

    organization
        "IETF OPSAWG (Network Configuration) Working Group";
    contact
        "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
        WG List: <mailto:opsawg@ietf.org>
        Author: Benoit Claise <mailto:benoit.claise@huawei.com>
        Author: Jean Quilbeuf <mailto:jean.quilbeuf@huawei.com>
        Author: Diego R. Lopez <diego.r.lopez@telefonica.com>
        Author: Ignacio Dominguez
                <ignacio.dominguezmartinez@telefonica.com>
        Author: Thomas Graf <thomas.graf@swisscom.com>";
    description
        "This module describes the platform information to be used as
         context of data collection from a given network element. The
         contents of this model must be streamed along with the data
         streamed from the network element so that the platform context
         of the data collection can be retrieved later.

The data content of this model should not change except on
upgrade or patching of the-a device.
```

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',  
'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED',  
'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document  
are to be interpreted as described in BCP 14 (RFC 2119)  
(RFC 8174) when, and only when, they appear in all  
capital letters, as shown here.

**Commenté [MB22]:** I don't see an relevant discussion in that page. Please double check.

**Commenté [MB23]:** As mount is not supported at design time, this module should be defined as an example and follow the guidance in 8407bis for examples (prefix choice, etc.).

I suggest to extract the main manifest grouping in a separate module and register that one.

Copyright (c) 2022-2025 IETF Trust and the persons identified as

**Commenté [MB24]:** Not used in the module

authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Revised BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>). This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices. ";

```
revision 2024-07-02 {
    description
        "Initial revision";
    reference
        "RFC xxxx: A Data Manifest for Contextualized Telemetry Data
Title to be completed";
}

container platforms {
    config false;
    description
        "Top container including all platforms in scope. If this model
is
        hosted on a single device, it should contain a single entry in
        the list. At the network level, it should contain an entry for
        every monitored platform.";
    list platform {
        key "id";
        description
            "Contains information about the platform that allows to
identify
            and understand the individual data collection information.
";
        leaf id {
            type string;
            description
                "Identifies a given platform on the network, for instance
the
                '_sysname_' of the _platform_. The '_id_' has to be
unique on _within_ the
                Network _scope._";
        }
        leaf name {
            type string;
            description
                "Model of the platform from which data is collected.";
        }
        leaf vendor {
            type string;
            description
                "Organization that implements that platform.";
        }
        leaf vendor-pen {
            type uint32;
            description
                "_Vendor's Vendor's registered Private Enterprise Number as
described in [RFC9371]";
```

Commenté [MB25]: Move this to a a reference statement

```

}
leaf software-version {
    type string;
    description
        "Name of the version of software. With respect to most
network
device appliances, this will be the operating system
version.

But for other YANG module implementations, this would be a
version of appliance software. Ultimately, this should
correspond to a version string that will be recognizable by
the consumers of the platform.";
}
leaf software-flavor {
    type string;
    description
        "A variation of a specific version where YANG model support
may be different. Depending on the vendor, this could be a
license, additional software component, or a feature set.";
}
leaf os-version {
    type string;
    description
        "Version of the operating system using this module. This is
primarily useful if the software implementing the module is
an application that requires a specific operating system
version.";
}
leaf os-type {
    type string;
    description
        "Type of the operating system using this module. This is
primarily useful if the software implementing the module is
an application that requires a specific operating system
type.";
}
yangmnt:mount-point "yang-library";
}
}
<CODE ENDS>

```

```

<CODE BEGINS> file "platform-extension-data.xml"
<yang-library xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library"
               xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
<module-set>
    <name>mountee-set</name>
    <module>
        <name>ietf-yang-status-conformance</name>
        <revision>2024-02-14</revision>
        <namespace>
            urn:ietf:params:xml:ns:yang:ietf-yang-status-conformance
        </namespace>
    </module>
    <module>
        <name>ietf-datastores</name>
        <revision>2018-02-14</revision>
        <namespace>

```

**Commenté [MB26]:** This is an example. I would state that

```

        urn:ietf:params:xml:ns.yang:ietf-datastores
    </namespace>
</module>
<module>
    <name>ietf-yang-library</name>
    <revision>2019-01-04</revision>
    <namespace>
        urn:ietf:params:xml:ns.yang:ietf-yang-library
    </namespace>
</module>
<import-only-module>
    <name>ietf-yang-types</name>
    <revision>2013-07-15</revision>
    <namespace>
        urn:ietf:params:xml:ns.yang:ietf-yang-types
    </namespace>
</import-only-module>
<import-only-module>
    <name>ietf-inet-types</name>
    <revision>2013-07-15</revision>
    <namespace>
        urn:ietf:params:xml:ns.yang:ietf-inet-types
    </namespace>
</import-only-module>
</module-set>
<schema>
    <name>test-schema</name>
    <module-set>mountee-set</module-set>
</schema>
<datastore>
    <name>ds:running</name>
    <schema>test-schema</schema>
</datastore>
<datastore>
    <name>ds:operational</name>
    <schema>test-schema</schema>
</datastore>
<content-id>2</content-id>
</yang-library>
<modules-state
    xmlns="urn:ietf:params:xml:ns.yang:ietf-yang-library">
    <module-set-id>2</module-set-id>
</modules-state>
<schema-mounts
    xmlns="urn:ietf:params:xml:ns.yang:ietf-yang-schema-mount">
    <mount-point>
        <module>ietf-platform-manifest</module>
        <label>yang-library</label>
        <shared-schema/>
    </mount-point>
</schema-mounts>
<CODE ENDS>
```

#### 4. Data Collection Manifest

##### 4.1. Overview of the Model

| Figure 2 contains the YANG tree diagram [RFC8340] of the ["ietf-data-](#)

collection-manifest” module. The module relies upon the YANG Schema mount [RFC8528] to reuse existing YANG modules describing the current data collection status. We explain in Appendix C explains how the YANG tree is obtained.

```
module: ietf-data-collection-manifest
  +-ro data-collections
    +-mp data-collection* [platform-id]
      +-ro platform-id    -> /p-mf:platforms/p-mf:platform/p-mf:id
      +-ro streams/
        |  +-ro stream* [name]
        |    +-ro name          string
        |    +-ro description?  string
      +-ro filters/
        |  +-ro stream-filter* [name]
        |    +-ro name          string
        |    +-ro (filter-spec)?
        |      +-:(stream-subtree-filter)
        |      +-:(stream-xpath-filter)
        |  +-ro selection-filter* [filter-id]
        |    +-ro filter-id     string
        |    +-ro (filter-spec)?
        |      +-:(datastore-subtree-filter)
        |      +-:(datastore-xpath-filter)
      +-ro subscriptions/
        +-ro subscription* [id]
          +-ro id                  subscription-id
          +-ro (target)
            +-:(stream)
              +-ro (stream-filter)?
                +-:(by-reference)
                  +-ro stream-filter-name   stream-filter-ref
                +-:(within-subscription)
                  +-ro (filter-spec)?
                    +-:(stream-subtree-filter)
                    +-:(stream-xpath-filter)
              +-ro stream      stream-ref
            +-:(datastore)
              +-ro datastore      identityref
              +-ro (selection-filter)?
                +-:(by-reference)
                  +-ro selection-filter-ref selection-filter-
ref
          +-:(within-subscription)
            +-ro (filter-spec)?
              +-:(datastore-subtree-filter)
              +-:(datastore-xpath-filter)
        +-ro stop-time?      yang:date-and-time
        +-ro encoding?       encoding
        +-ro receivers
          +-ro receiver* [name]
            +-ro name          string
            +-ro sent-event-records? yang:zero-based-
counter64
          +-ro excluded-event-records? yang:zero-based-
counter64
          +-ro state         enumeration
```

```

    +-ro current-period?      yp:centiseconds
    +-ro (update-trigger)?
        +-:(periodic)
            +-ro periodic!
                +-ro period      centiseconds
                +-ro anchor-time?  yang:date-and-time

```

Figure 2: YANG tree diagram for ietf-data-collection-manifest module

The 'data-collections' container contains the information related to each YANG-Push subscription. As for the Platform Manifest, these subscriptions are indexed by the platform-'platform-id', so that all subscriptions in the network can be represented in the module at the network level without any conflict.

As most of the information related to YANG-push subscription [RFC8639] and [RFC8641] is stored in the "ietf-yang-push" model module, we mount these modules are mounted. These modules have a part common to all subscriptions of the-a platform, stored in the 'streams' and 'filters' containerseentainer. The information about subscriptions themselves are stored in the 'subscriptions/subscription' list, indexed by a subscription id.

**Commenté [MB27]:** Which one is not stored?

In the subscription object, the 'current-period' indicates the period currently used between two updates. That leaf can only be present when the subscription is periodic. The current period might differ from the requested period if the platform implements a mechanism to increase the collection period when it is overloaded. Having the current period information is crucial to understand if telemetry is missing because of a bug or a packet loss or simply because it was dynamically adjusted by the platform.

The 'current-period' data node is added by the module 'ietf-data-collection-manifest-statistics' presented in Section 4.2. This module augments the subscription list from the module 'ietf-subscribed-notifications'. It is mounted as well via the YANG Schema Mount mechanism. The module for the Data Collection Manifest is presented in Section 4.3.

#### 4.2. The "ietf-data-collection-manifest-statistics" YANG module ietf-data-collection-manifest-statistics

Below is the code of theThe "ietf-data-collection-manifest-statistics" module, which augments the "ietf-subscribed-notification" modules module to provide information needed for the Data Collection Manifest.

```

<CODE BEGINS>
file "ietf-data-collection-manifest-statistics@2024-07-02.yang"
module ietf-data-collection-manifest-statistics {
    yang-version 1.1;
    namespace
        "urn:ietf:params:xml:ns:yang:ietf-data-collection-manifest-
statistics";

```

**Commenté [MB28]:** A more generic name (and shorter 😊) can be used as this is not specific to the manifest discussion.

**Commenté [MB29]:** Why this module is not registered?

```

prefix dcs;

import ietf-subscribed-notifications {
    prefix sn;
    reference
        "RFC 8639: A YANG Data Model for Subscriptions to
         Event Notifications";
}
import ietf-yang-push {
    prefix yp;
    reference
        "RFC 8641: Subscriptions to YANG Datastores.";
}

organization
    "IETF OPSAWG (Network Configuration) Working Group";
contact
    "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
     WG List: <mailto:opsawg@ietf.org>
     Author: Benoit Claise <mailto:benoit.claise@huawei.com>
     Author: Jean Quilbeuf <mailto:jean.quilbeuf@huawei.com>
     Author: Diego R. Lopez <diego.r.lopez@telefonica.com>
     Author: Ignacio Dominguez
     <ignacio.dominguezmartinez@telefonica.com>
     Author: Thomas Graf <thomas.graf@swisscom.com>";
description
    "This module augments subscribed notification with the
     current-period statistics reporting the actual collection
     period.

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',
' SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED',
'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document
are to be interpreted as described in BCP 14 (RFC 2119)
(RFC 8174) when, and only when, they appear in all
capital letters, as shown here."
Copyright (c) 2022-2025 IETF Trust and the persons identified as
authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, is permitted pursuant to, and subject
to the license terms contained in, the Revised BSD License
set forth in Section 4.c of the IETF Trust's Legal Provisions
Relating to IETF Documents
(https://trustee.ietf.org/license-info).
This version of this YANG module is part of RFC XXXX; see the
RFC itself for full legal notices. ";

revision 2024-07-02 {
    description
        "Initial revision";
    reference
        "RFC ****: A Data Manifest for Contextualized Telemetry Data
Title to be completed";
}

augment "/sn:subscriptions/sn:subscription" {

```

**Commenté [MB30]:** Not used

```

|     description
|       "Adds extra-current-period statistics about data collection.";
leaf current-period {
    when '../yp:periodic';
    type yp:centiseconds;
    description
        "Period during two succesivesuccessive data collections, in
the
        current state. Might differ from the configured period
when the platformplatform might increase the period
automatically when it is overloaded.";
}
}
<CODE ENDS>
```

#### 4.3. The "ietf-data-collection-manifest" YANG module ietf-data-collection-manifest

a mis en forme : Anglais (États-Unis)

~~We provide in this~~This section includes the code of the "ietf-data-collection-manifest" YANG module. Additionally, we\_it provides the extension data file for YANG schema mount. The Data Collection Manifest MUST conform to the model obtained by combining these two specifications.

a mis en forme : Surlignage

```

<CODE BEGINS> file "ietf-data-collection-manifest@2024-07-02.yang"
module ietf-data-collection-manifest {
    yang-version 1.1;
    namespace
        "urn:ietf:params:xml:ns:yang:ietf-data-collection-manifest";
    prefix d-mf;

    import ietf-platform-manifest {
        prefix p-mf;
        reference
            "RFC XXXX: A Data Manifest for Contextualized Telemetry Data
Title to be completed";
    }
    import ietf-yang-schema-mount {
        prefix yangmnt;
        reference
            "RFC 8528: YANG Schema Mount";
    }

    organization
        "IETF OPSAWG (Network Configuration) Working Group";
    contact
        "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
        WG List: <mailto:opsawg@ietf.org>
        Author: Benoit Claise <mailto:benoit.claise@huawei.com>
        Author: Jean Quilbeuf <mailto:jean.quilbeuf@huawei.com>
        Author: Diego R. Lopez <mailto:diego.r.lopez@telefonica.com>
        Author: Ignacio Dominguez
                <mailto:ignacio.dominguezmartinez@telefonica.com>
        Author: Thomas Graf <mailto:thomas.graf@swisscom.com>";

    description
        "This module describes the context of data collection from a
```

Commenté [MB31]: This is an example

given network element. The contents of this model must be streamed along with the data streamed from the network element so that the context of the data collection can be retrieved later.

This module must be completed with  
'ietf-platform-manifest'  
to capture the whole context of a data collection session.

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',  
'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED',  
'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document  
are to be interpreted as described in BCP 14 (RFC 2119)  
(RFC 8174) when, and only when, they appear in all  
capitals, as shown here.

Commenté [MB32]: Not used

Copyright (c) 2022-2025 IETF Trust and the persons identified as  
authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or  
without modification, is permitted pursuant to, and subject  
to the license terms contained in, the Revised BSD License  
set forth in Section 4.c of the IETF Trust's Legal Provisions  
Relating to IETF Documents  
(<https://trustee.ietf.org/license-info>).  
This version of this YANG module is part of RFC XXXX; see the  
RFC itself for full legal notices. ";

```
revision 2024-07-02 {  
    description  
        "Initial revision";  
    reference  
        "RFC xxxxXXXX: A Data Manifest for Contextualized Telemetry Data  
Title to be completed";  
}  
  
container data-collections {  
    config false;  
    description  
        "Contains the configuration and statistics for the collected  
data,  
        per node in the network.";  
    list data-collection {  
        key "platform-id";  
        description  
            "Defines the information for each collected object.  
leaf platform-id {  
    type leafref {  
        path "/p-mf:platforms/p-mf:platform/p-mf:id";  
    }  
    description  
        "Identifier of the platform collecting the data. This id is  
the same  
        as the one in the platform manifest.";  
    }  
    yangmnt:mount-point "yang-push-collection";  
    // augment here with other kind of collection items  
}
```

```

        }
    }
<CODE ENDS>

<CODE BEGINS> file "data-collection-extension-data.xml"
<yang-library xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library"
               xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
    <module-set>
        <name>mountee-set</name>
        <module>
            <name>ietf-subscribed-notifications</name>
            <revision>2019-09-09</revision>
            <namespace>
                urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications
                </namespace>
            </module>
        <module>
            <name>ietf-yang-push</name>
            <revision>2019-09-09</revision>
            <namespace>
                urn:ietf:params:xml:ns:yang:ietf-yang-push
                </namespace>
            </module>
        <module>
            <name>ietf-data-collection-manifest-statistics</name>
            <revision>2024-07-02</revision>
            <namespace>
                urn:ietf:params:xml:ns:yang:ietf-data-collection-manifest-statistics
                </namespace>
            </module>
        <module>
            <name>ietf-datastores</name>
            <revision>2018-02-14</revision>
            <namespace>
                urn:ietf:params:xml:ns:yang:ietf-datastores
                </namespace>
            </module>
        <module>
            <name>ietf-yang-library</name>
            <revision>2019-01-04</revision>
            <namespace>
                urn:ietf:params:xml:ns:yang:ietf-yang-library
                </namespace>
            </module>
        <import-only-module>
            <name>ietf-inet-types</name>
            <revision>2013-07-15</revision>
            <namespace>
                urn:ietf:params:xml:ns:yang:ietf-inet-types
                </namespace>
            </import-only-module>
        <import-only-module>
            <name>ietf-interfaces</name>
            <revision>2018-02-20</revision>
            <namespace>
                urn:ietf:params:xml:ns:yang:ietf-interfaces
                </namespace>
            </import-only-module>

```

**Commenté [MB33]:** This is an example

```
<import-only-module>
  <name>ietf-ip</name>
  <revision>2018-02-22</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-ip
  </namespace>
</import-only-module>
<import-only-module>
  <name>ietf-netconf-acm</name>
  <revision>2018-02-14</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-netconf-acm
  </namespace>
</import-only-module>
<import-only-module>
  <name>ietf-network-instance</name>
  <revision>2019-01-21</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-network-instance
  </namespace>
</import-only-module>
<import-only-module>
  <name>ietf-restconf</name>
  <revision>2017-01-26</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-restconf
  </namespace>
</import-only-module>
<import-only-module>
  <name>ietf-yang-patch</name>
  <revision>2017-02-22</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-yang-patch
  </namespace>
</import-only-module>
<import-only-module>
  <name>ietf-yang-types</name>
  <revision>2023-01-23</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-yang-types
  </namespace>
</import-only-module>
</module-set>
<schema>
  <name>test-schema</name>
  <module-set>mountee-set</module-set>
</schema>
<datastore>
  <name>ds:running</name>
  <schema>test-schema</schema>
</datastore>
<datastore>
  <name>ds:operational</name>
  <schema>test-schema</schema>
</datastore>
<content-id>2</content-id>
</yang-library>
<modules-state
```

```

xmlns="urn:ietf:params:xml:ns.yang:ietf-yang-library">
<module-set-id>2</module-set-id>
</modules-state>
<schema-mounts
  xmlns="urn:ietf:params:xml:ns.yang:ietf-yang-schema-mount">
  <mount-point>
    <module>ietf-data-collection-manifest</module>
    <label>yang-push-collection</label>
    <shared-schema/>
  </mount-point>
</schema-mounts>
<CODE ENDS>
```

## 5. Data Manifest and the Collected Data

### 5.1. Collecting the Data Manifest

The Data Manifest MUST be streamed and stored along with the collected data. In case the collected data are moved to a different place (typically a database), the companion Data Manifest MUST follow the

collected data. This can render the collected data unusable if that context is lost, for instance when the data is stored without the relevant information. The Data Manifest MUST be updated when the Data Manifest information changes, for example, when a router is upgraded, when a new telemetry subscription is configured, or when the telemetry subscription parameters change. The Data Manifest can itself be considered as a time series, and stored in a similar fashion to the collected data.

The collected data should be mapped to the Data Manifest. Since the Data Manifest will not change as frequently as the collected data itself, it makes sense to map several data to the same Data Manifest. Somehow, the collected data must include a metadata pointing to the corresponding Data Manifest. [In case of Data Manifest change, the system should keep the mapping between the data collected so far and the old Data Manifest, and not assume that the latest Data Manifest is valid for the entire time series.]

The A Platform Manifest is likely to remain the same until the platform is updated. Thus, the Platform Manifest only needs to be collected once per streaming session and updated after a platform reboot.

Similarly, the elements common to all subscriptions, such as the stream definitions and the common filters might be updated less frequently than the subscriptions.

As this draft-document specifically focuses on giving context on data collected via streamed telemetry, we emit is assumed that a streaming telemetry system is available. Retrieving the Data Collection Manifest and Platform Manifest can be done either by reusing that streaming telemetry system (in-band) or using another system (out-of-band), for instance by adding headers or saving manifests into a YANG instance file [RFC9195].

**Commenté [MB34]:** This is an important aspect that needs further elaboration about how this will be implemented.

~~We propose to~~ This document recommends reusing the existing telemetry system (in-band approach) in order to lower the efforts for implementing this ~~draft~~ approach. To enable a platform supporting streaming telemetry to also support the Data Manifest, it is sufficient that this platform supports the models from Sections ~~3~~ and ~~Section~~ 4. Recall that each type of manifest has its own rough frequency update, i.e., at reboot for the Platform Manifest and at new subscription or CPU load variation for the Data Collection Manifest. The Data Manifest MUST be streamed with the YANG-Push on-change feature [RFC8641] (also called event-driven telemetry). Appendix A shows how the in-band approach would work while storing to a time-series database (TSDB).

The underlying time series database should accommodate the various rates at which different part of the Data Manifest are updated. In particular, storing the Platform Manifest should be optimized to avoid duplicating repeated content and only storing a new version when there is a change in the manifest.

### 5.2. Mapping Collected Data to the Data Manifest

With YANG-push, each notification sent by ~~the-a~~ device is part of a subscription, which is also one of the YANG keys used to retrieve the Data Manifest, the other key being the platform ~~ID's id'~~. In order to enable a posteriori retrieval of the Data Manifest associated to a ~~datapoint~~, the ~~collector~~ must:

- \* Keep the subscription id and platform id in the metadata of the collected values
- \* Collect as well the Data Manifest for the subscription associated to the datapoint.

**Commenté [MB35]:** To be added to the definition section

**Commenté [MB36]:** To be added to the definition section

With this information, to retrieve the Data Manifest from ~~thea~~ datapoint, the following happens:

- \* The subscription id and platform id are retrieved from the datapoint metadata
- \* The Data Manifest for that datapoint is obtained by using the values above as keys.

We don't focus on the timing aspect as storing both the data and their manifest in a time series database (TSDB) will allow the data scientists to look for the Data Manifest corresponding to the timestamp of the datapoint. More precisely, given the timestamp of a collected datapoint, the query to the TSDB would be to get the last version of the Data Manifest before that timestamp. In that scenario, the reliability of the collection of the Data Manifest is the same as the reliability of the data collection itself, since the Data Manifest is like any other data.

### 5.3. Operational Considerations

It is expected that the Data Manifest is streamed directly from the network equipment, along with YANG-Push [RFC8641] data. However, if the network equipment streaming telemetry does not support yet the

YANG modules from the Data Manifest specified in this document, the telemetry collector could populate the Data Manifest from available information collected from the platform. However, this option requires efforts on the telemetry collector side, as the information gathered in the Data Manifest proposed in this document could be scattered among various standard and vendor-specific YANG modules [RFC8199], that depend on the platform.

That Data Manifest should be kept and available even if the source platform is not accessible (from the collection system), or if the platform has been updated (new operating system or new configuration). The Platform Manifest is "pretty" stable and should change only when the platform is updated or patched. On the other hand, the Data Collection Manifest is likely to change each time a new YANG-Push subscription [RFC8641] is requested and might even change if the platform load increases and collection periods are updated. To separate these two parts, ~~we enclose~~ each of them are managed using its own module.

## 6. Example

Figure 3 shows Below is—an example of both a Platform manifest and corresponding Data Collection Manifests. The list of YANG modules in the yang-library container is kept empty for brevity.

```
<CODE BEGINS> file "manifests-example.json"
{
    "ietf-platform-manifest:platforms": [
        "platform": [
            {
                "id": "PE1",
                "name": "PE1",
                "vendor": "ACME",
                "vendor-pen": 32473,
                "software-version": "3.14",
                "os-version": "2.79",
                "os-type": "ACME OS",
                "yang-library": {...}
            }
        ]
    },
    "ietf-data-collection-manifest:data-collections": {
        "data-collection": [
            {
                "platform-id": "PE1",
                "subscriptions": {
                    "subscription": [
                        {
                            "id": 4242,
                            "datastore": "ietf-datastores:operational",
                            "datastore-xpath-filter": "/ietf-interfaces:interfaces/interface/enabled",
                            "on-change": {},
                            "receivers": {
                                ...
                            }
                        }
                    ]
                }
            }
        ]
    }
}
```

**Commenté [MB37]:** Any considerations about when it is safe to flush out the manifest data.

```

        "receiver": [
            {
                "name": "yp-collector",
                "state": "active"
            }
        ]
    },
    {
        "id": 4243,
        "datastore": "ietf-datastores:operational",
        "datastore-xpath-filter":
            "/ietf-interfaces:interfaces/interface/statistics/in-octets",
        "periodic": {
            "period": 10000
        },
        "current-period": 20000,
        "receivers": {
            "receiver": [
                {
                    "name": "yp-collector",
                    "state": "active"
                }
            ]
        }
    }
]
}
}

```

**Commenté [MB38]:** Should be prefixed with the new module in this document

Figure 3

CODE ENDS

Figure 3 The file above contains the Data Collection Manifest for two XPaths. With the Data Collection Manifest for the first one, with subscription id 4242, the exact semantics of the collected path, here the administrative status of the network interfaces, can be obtained by looking up the module in the yang-library of the corresponding Platform Manifest, in order to obtain the exact revision of ietf-interfaces used at collection time. Also, the "on-change" container indicates that data will be sent only if there is a change, thus not receiving data indicates that the administrative status of the interface did not change.

The other example of Data Collection Manifest, with subscription id 4243, shows how a periodic subscription is reported. In that example, the 'current-period' indicates that the requested period of 10s (1000 centiseconds) could not be attained and is now of 20s, for instance because the device is overloaded.

## 7. Security Considerations

As we are reusing an existing telemetry system, the security

**Commenté [MB39]:** The security template in draft 8407bis is needed for at least the augmented module.

considerations lies with the new content divulged in the new manifests. Appropriate access control filters must be associated to the corresponding leafs and containers.

The integrity and provenance of the data of the collection manifest can be ensured by a signing mechanism such as [I-D.lopez-opsawg-yang-provenance].

## 8. IANA Considerations

This document includes no request to IANA.

**Commenté [MB40]:** I think there is at least one module to register

## 9. Contributors

## 10. Open Issues

- \* Do we want to handle the absence of values, i.e. add information about missed collection or errors in the collection context ? It could also explain why some values are missing. On the other hand, this might also be out scope. CLOSED: the goal of the manifest is to be able to detect miscollection a posteriori. Assurance of the metric collection is out of scope and could be done via an external mechanism such as SAIN.
- \* Henk: how does this interact with SBOM effort? CLOSED: SBOM is another kind of manifest, we are focusing here on data collection.
- \* What is the link with the RFC8345 NodeId and IVY? CLOSED: added text.
- \* Handling of deletion in [I-D.kll-yang-label-tsdb].

**Commenté [MB41]:** Is this still an open issue?

## 11. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <<https://www.rfc-editor.org/info/rfc8525>>.
- [RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <<https://www.rfc-editor.org/info/rfc8639>>.

**Commenté [MB42]:** Should be listed as informative per 8407bis

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.

[RFC9195] Lengyel, B. and B. Claise, "A File Format for YANG Instance Data", RFC 9195, DOI 10.17487/RFC9195, February 2022, <<https://www.rfc-editor.org/info/rfc9195>>.

Commenté [MB43]: Should be listed as informative

## 12. Informative References

[I-D.clacla-netmod-model-catalog]  
Clarke, J. and B. Claise, "YANG module for yangcatalog.org", Work in Progress, Internet-Draft, draft-clacla-netmod-model-catalog-03, 3 April 2018, <<https://datatracker.ietf.org/doc/html/draft-clacla-netmod-model-catalog-03>>.

[I-D.claise-netconf-metadata-for-collection]  
Claise, B., Nayyar, M., and A. R. Sesani, "Per-Node Capabilities for Optimum Operational Data Collection", Work in Progress, Internet-Draft, draft-claise-netconf-metadata-for-collection-02, 12 July 2021, <<https://datatracker.ietf.org/doc/html/draft-claise-netconf-metadata-for-collection-02>>.

[I-D.havel-nmop-digital-map]  
Havel, O., Claise, B., de Dios, O. G., Elhassany, A., and T. Graf, "Modeling the Digital Map based on RFC 8345: Sharing Experience and Perspectives", Work in Progress, Internet-Draft, draft-havel-nmop-digital-map-01, 5 July 2024, <<https://datatracker.ietf.org/doc/html/draft-havel-nmop-digital-map-01>>.

[I-D.ietf-ivy-network-inventory-yang]  
Yu, C., Belotti, S., Bouquier, J., Peruzzini, F., and P. Bedard, "A YANG Data Model for Network Inventory", Work in Progress, Internet-Draft, draft-ietf-ivy-network-inventory-yang-03, 7 July 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-ivy-network-inventory-yang-03>>.

[I-D.kll-yang-label-tsdb]  
Larsson, K., "Mapping YANG Data to Label-Set Time Series", Work in Progress, Internet-Draft, draft-kll-yang-label-tsdb-00, 18 October 2023, <<https://datatracker.ietf.org/doc/html/draft-kll-yang-label-tsdb-00>>.

[I-D.lopez-opsawg-yang-provenance]  
Lopez, D., Pastor, A., Feng, A. H., Birkholz, H., and S. Garcia, "Applying COSE Signatures for YANG Data Provenance", Work in Progress, Internet-Draft, draft-lopez-opsawg-yang-provenance-03, 6 July 2024, <<https://datatracker.ietf.org/doc/html/draft-lopez-opsawg-yang-provenance-03>>.

- [I-D.tgraf-netconf-notif-sequencing]  
Graf, T., Quilbeuf, J., and A. H. Feng, "Support of Hostname and Sequencing in YANG Notifications", Work in Progress, Internet-Draft, draft-tgraf-netconf-notif-sequencing-06, 29 June 2024, <<https://datatracker.ietf.org/doc/html/draft-tgraf-netconf-notif-sequencing-06>>.
- [RFC8199] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", RFC 8199, DOI 10.17487/RFC8199, July 2017, <<https://www.rfc-editor.org/info/rfc8199>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8528] Bjorklund, M. and L. Lhotka, "YANG Schema Mount", RFC 8528, DOI 10.17487/RFC8528, March 2019, <<https://www.rfc-editor.org/info/rfc8528>>.

**Commenté [MB44]:** Should be normative

#### Appendix A. An Example of use-based Based on MDT

In this example, the goal is to collect the administrative status and number of received bytes for the interfaces of a fictional ACME device, and store the result in a time-series database (TSDB). The metrics are collected via using YANG-Push, which is configured by specifying their XPaths and when they should be collected (periodically or on-change). More precisely, we want collect "ietf-interfaces:interfaces/interface/enabled" on every change and "ietf-interfaces:interfaces/interface/statistics/in-octets" every 100 milliseconds. The paths here are referring to the YANG module from [RFC8343]. The configuration of YANG push is out of scope for this document. Since they don't have the same trigger, each of the path must be collected in its own subscription. Figure 4 presents an example for such a collection.

**Commenté [MB45]:** Already introduced



```

+
+-----+
| Device |
+-----+

```

Figure 4: Example of Collection from From a device-Device to a TSDB

In the scenario from depicted in Figure 4, the collector receives YANG-push data from the device and stores it into a TSDB. We first present a version without Data Manifest and then how to enrich it with the Data Manifest.

We use the notation from [I-D.kll-yang-label-tsdb] to represent how the data is stored in the TSDB. Without the data manifest, the result of the collection would be stored as showed in Figure 5. The "host" label indicates the devices from which the data is collected and the YANG keys are included as well. Here the interface "eth0" is enabled and received 1234 octets. In that case, the value is stored, without any way to know how the value was obtained.

```

* Metric: interfaces_interface_enabled
* Value: True
* Labels:
  - host: "PE1"
  - interfaces_interface_name: "eth0"
--
* Metric: interfaces_interface_statistics_in_octets
* Value: 1234
* Labels:
  - host: "PE1"
  - interfaces_interface_name: "eth0"

```

Figure 5: Storing Datapoints without Without Data Manifest

An option possibility for keeping the Data Manifest with the data is to store it directly into the TSDB. In that case, the collector can subscribe to the data exported by the module presented in this draft document and store

it as other metrics. For the Platform Manifest, assuming the platform ID-id' is "PE1", the collector subscribes to the path "ietf-platform-manifest:platforms/platform[platform[id=PE1]]". For the Data Collection Manifests, the collector subscribes to the path "ietf-data-collection-manifest:data-collections/data-collection[platform-id="PE1"]/yang-push-collection/subscriptions/subscription[id=X]" where X is the subscription id of existing subscriptions. With the approach from [I-D.kll-yang-label-tsdb], the corresponding subtrees would be split into a set of datapoints, one per leaf. Figure 6 shows two examples of storing leaves in a TSDB. The first leaf is the vendor PEN number, which is part of the Platform Manifest. The second leaf is the Xpath filter used for subscription to the interface status.

```

* Metric: platforms_platform_vendor_pen
* Value: 32473
* Labels:

```

```

- host: "PE1"
- platforms_platform_id: "PE1"
--
* Metric: data_collections_data_collection_yang_push_collection_
    subscriptions_subscription_datastore_xpath_filter
* Value: "ietf-interfaces:interfaces/interface/enabled"
* Labels:
- host: "PE1"
- data_collections_data_collection_platform_id: "PE1"
- data_collections_data_collection_yang_push_collection_
    subscriptions_subscription_id: 4242

```

Figure 6: Example of storing Platform and Data Collection Manifest: Vendor PEN and Xpath filter.

In the labels, the "host" might be different from the "platforms\_platform\_id" in case the collector is the one assembling it, i.e. for devices that do not natively support the Data Manifest. In that case, the value of this label could be the hostname of the collector. The host value does not matter for retrieving the Data Manifest as the platform id is the meaningful field.

In our example, we can retrieve the Platform Manifest associated to a collected datapoint by looking for datapoints that have the label "platforms\_platform\_id" equal to the value of the host for that collected datapoint. In order to link a datapoint with the corresponding Data Collection Manifest, we need to add an additional label for the subscription id. For instance, the same datapoints as in Figure 5 could be stored as in Figure 7.

```

* Metric: interfaces_interface_enabled
* Value: True
* Labels:
- host: "PE1"
- interfaces_interface_name: "eth0"
- data_collections_data_collection_yang_push_subscriptions_
    subscription_id: 4242
--
* Metric: interfaces_interface_statistics_in_octets
* Value: 1234
* Labels:
- host: "PE1"
- interfaces_interface_name: "eth0"
- data_collections_data_collection_yang_push_subscriptions_
    subscription_id: 4243

```

Figure 7: Storing datapoints with information to retrieve the Data Manifest

From the "interfaces\_interface\_enabled" datapoint, one can retrieve the corresponding Data Collection Manifest by looking for datapoints that have the label data\_collections\_data\_collection\_yang\_push\_collection\_subscriptions\_subscription\_id equal to 4242.

Various optimizations could be done, such as relying on on-change subscription to modify only the leaves that changed. In that way, the amount of data needed for updating and storing the Data Manifest in the TSDB would be limited.

## Appendix B. Changes between revisions

v04 -> v05

- \* Remove references to full-include draft, use schema mount.
- \* Explain link with schema node id

v03 -> v04

- \* State that data lineage is out of scope
- \* Replace copy-pasted version of the modules with schema mount version, use full-embed for the "real" one
- \* Schema mount version is the fallback plan if full:embed is not there fast enough.
- \* Update examples accordingly

v02 -> v03

- \* Explicit that modules are network (Controller) level
- \* InfluxDB example changed to TSDB example aligned with [I-D.kll-yang-label-tsdb]
- \* Minor edits i.e. network element -> platform , object -> data node

v01 -> v02

- \* Updated example with latest version of the model.

v00 (WG adoption) - v01

- \* Solve integrity issue by delegating to [I-D.lopez-opsawg-yang-provenance].

v05 -> v06

- \* Remove YANG packages
- \* Switch YANG models from device view to network view
- \* Add PEN number to identify vendors
- \* Intro rewritten with uses cases
- \* Added an "Operational Considerations" section
- \* Switch from MDT to YANG-push

v04 -> v05

- \* First version of example scenario
- \* Updated affiliation

- \* Updated YANG module names to ietf-platform-manifest and ietf-data-collection-manifest
  - \* Unify used terms as defined in the terminology section
  - \* Replaced 'device' with 'platform'
  - \* Split Section 5 into two sections for better readability
- v03 -> v04
- \* Fix xym error
  - \* Moved terminology after introduction
  - \* Clarified the role of the module
- v02 -> v03
- \* Add when clause in YANG model
  - \* Fix validation errors on YANG modules
  - \* Augment YANG library to handle semantic versioning
- v01 -> v02
- \* Alignment with YANGCatalog YANG module: name, vendor
  - \* Clarify the use of YANG instance file
  - \* Editorial improvements
- v00 -> v01
- \* Adding more into data platform: yang packages, whole yanglib module to specify datastores
  - \* Setting the right type for periods: int64 -> uint64
  - \* Specify the origin datastore for mdt subscription
  - \* Set both models to config false
  - \* Applying text comments from Mohamed Boucadair
  - \* Adding an example of data-manifest file
  - \* Adding rationale for reusing telemetry system for collection of the manifests
  - \* Export manifest with on change telemetry as opposed to YANG instance file
- v00
- \* Initial version

## Appendix C. Generating YANG Tree Diagrams

This section provides the files needed to generate the YANG tree diagrams [RFC8340] from Figure 1 and Figure 2. The diagrams were obtained using yanglint <https://github.com/CESNET/libyang> version 2.1.80, using the YANG Schema Mount [RFC8528]. They were manually edited to remove parts irrelevant to this document such as data nodes from imported modules, notifications and RPCs.

In order to get a tree diagram involving YANG Schema Mount with yanglint, two data files are required, in addition to the YANG module, its dependencies and the YANG modules to be mounted. First we need the extension data, containing the YANG library to use at the mount point. Then we need the YANG library to use at the top-level context. We provide below the commands used to get the "raw" YANG Tree diagrams from these files.

### C.1. YANG Tree Diagram of the Platform Manifest

For the Platform Manifest, the extension data is provided in Section 3.2 as "platform-extension-data.xml". The top-level YANG library is included below as "platform-toplevel-yanglib.xml". The following command was used to obtain the YANG Tree diagram (before manual edition).

```
yanglint -f tree \
-x platform-extension-data.xml \
-Y platform-toplevel-yanglib.xml \
ietf-platform-manifest@2024-07-02.yang

<CODE BEGINS> file "platform-toplevel-yanglib.xml"
<yang-library xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library"
  xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
  <module-set>
    <name>main-set</name>
    <module>
      <name>ietf-datastores</name>
      <revision>2018-02-14</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-datastores
      </namespace>
    </module>
    <module>
      <name>ietf-yang-library</name>
      <revision>2019-01-04</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-yang-library
      </namespace>
    </module>
    <module>
      <name>ietf-yang-schema-mount</name>
      <revision>2019-01-14</revision>
      <namespace>
```

a mis en forme : Surlignage

```

        urn:ietf:params:xml:ns.yang:ietf-yang-schema-mount
    </namespace>
</module>
<module>
    <name>ietf-platform-manifest</name>
    <revision>2024-07-02</revision>
    <namespace>urn:ietf:params:xml:ns.yang:ietf-platform-
manifest</namespace>
</module>
<import-only-module>
    <name>ietf-yang-types</name>
    <revision>2023-01-23</revision>
    <namespace>
        urn:ietf:params:xml:ns.yang:ietf-yang-types
    </namespace>
</import-only-module>
<import-only-module>
    <name>ietf-inet-types</name>
    <revision>2013-07-15</revision>
    <namespace>
        urn:ietf:params:xml:ns.yang:ietf-inet-types
    </namespace>
</import-only-module>
</module-set>
<schema>
    <name>main-schema</name>
    <module-set>main-set</module-set>
</schema>
<datastore>
    <name>ds:running</name>
    <schema>main-schema</schema>
</datastore>
<datastore>
    <name>ds:operational</name>
    <schema>main-schema</schema>
</datastore>
<content-id>1</content-id>
</yang-library>
<modules-state
xmlns="urn:ietf:params:xml:ns.yang:ietf-yang-library">
<module-set-id>2</module-set-id>
</modules-state>
<CODE ENDS>

```

#### C.2. YANG Tree Diagram of the Data Collection Manifest

For the Data Collection Manifest, the extension data is provided in Section 4.3 as "data-collection-extension-data.xml". The top-level YANG library is included below as "data-collection-toplevel-yanglib.xml". The following command was used to obtain the YANG Tree diagram (before manual edition).

```

yanglint -f tree \
-x data-collection-extension-data.xml \
-Y data-collection-toplevel-yanglib.xml \
ietf-data-collection-manifest@2024-07-02.yang

```

```
<CODE BEGINS> file "data-collection-toplevel-yanglib.xml"
<yang-library xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library"
               xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
  <module-set>
    <name>main-set</name>
    <module>
      <name>ietf-datastores</name>
      <revision>2018-02-14</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-datastores
      </namespace>
    </module>
    <module>
      <name>ietf-yang-library</name>
      <revision>2019-01-04</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-yang-library
      </namespace>
    </module>
    <module>
      <name>ietf-yang-schema-mount</name>
      <revision>2019-01-14</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount
      </namespace>
    </module>
    <module>
      <name>ietf-data-collection-manifest</name>
      <revision>2024-07-02</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-data-collection-manifest
      </namespace>
    </module>
    <module>
      <name>ietf-platform-manifest</name>
      <revision>2024-07-02</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-platform-manifest
      </namespace>
    </module>
    <import-only-module>
      <name>ietf-inet-types</name>
      <revision>2013-07-15</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-inet-types
      </namespace>
    </import-only-module>
    <import-only-module>
      <name>ietf-yang-types</name>
      <revision>2023-01-23</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-yang-types
      </namespace>
    </import-only-module>
  </module-set>
  <schema>
    <name>main-schema</name>
    <module-set>main-set</module-set>
```

```
</schema>
<datastore>
  <name>ds:running</name>
  <schema>main-schema</schema>
</datastore>
<datastore>
  <name>ds:operational</name>
  <schema>main-schema</schema>
</datastore>
<content-id>1</content-id>
</yang-library>
<modules-state
  xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
  <module-set-id>2</module-set-id>
</modules-state>
<CODE ENDS>
```

#### Acknowledgements

Thanks to Mohamed Boucadair, Tianran Zhou and Jan Lindblad for their reviews and comments.

#### Authors' Addresses

Benoit Claise  
Huawei  
Email: benoit.claise@huawei.com

Jean Quilbeuf  
Huawei  
Email: jean.quilbeuf@huawei.com

Diego R. Lopez  
Telefonica I+D  
Don Ramon de la Cruz, 82  
Madrid 28006  
Spain  
Email: diego.r.lopez@telefonica.com

Ignacio Dominguez  
Telefonica I+D  
Ronda de la Comunicacion, S/N  
Madrid 28050  
Spain  
Email: ignacio.dominguezmartinez@telefonica.com

Thomas Graf  
Swisscom  
Binzring 17  
CH-8045 Zurich  
Switzerland  
Email: thomas.graf@swisscom.com

