GREEN WG
Internet-Draft
Intended status: Standards Track
Expires: 1 September 2025

J. Lindblad
All For Eco

28 February 2025

Philatelist, YANG-based Network Controller collection and aggregation framework integrating Telemetry data and Time Series Databases
draft-lindblad-tlm-philatelist-03

Abstract

Timestamped telemetry data is collected en masse today. Mature tools are typically used, but the data is often collected in an ad hoc manner. While the dashboard graphs look great, the resulting data is often of questionable quality, not well defined, and hard to compare with seemingly similar data from other organizations.

This document proposes a standard, extensible, cross domain framework for collecting and aggregating timestamped telemetry data in a way that combines YANG, metadata and Time Series Databases to produce more transparent, dependable and comparable results. This framework is implemented in the Network Controller layer, but is rooted in data that is collected from all kinds of Network Elements and related systems.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://janlindblad.github.io/draft-tlm-philatelist/draft-lindblad-tlm-philatelist.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-lindblad-tlm-philatelist/.

Source for this draft and an issue tracker can be found at https://github.com/janlindblad/draft-tlm-philatelist.

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 1 September 2025.

Copyright Notice

Copyright (c) 2025 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

1.1. The Problem

Many organizations today are collecting large amounts of telemetry data from their networks and data centers for a variety of purposes. Much (most?) of this data is funneled into a Time Series Database (TSDB) for display in a graphical dashboard or further (AI-backed) processing and decision making.

While this data collection is often handled using existing and stable tools, there generally seems to be little commonality when it comes to what is meaured, how the data is aggregated, or definitions of the measured quantities (if any).

Data science issues like adding overlapping quantities, adding quantities of different units of measurement, or quantities with different scopes, are likely common. Such errors are hard to detect given the ad hoc nature of the collection. This often leads to uncertainty regarding the quality of the conclusions drawn from the collected data.

1.2. The Solution

The Philatelist framework proposes to standardize the collection, definitions of the quantities measured and meta data handling to provide a robust ground layer for telemetry collection on the Controller side. The architecture defines a few interfaces, but allows great freedom in the implementations with its plug-in architecture. This allows flexibility enough that any kind of quantity can be measured, any kind of collection protocol and mechanism employed, and the telemetry data flows aggregated using any kind of operation.

To do this, YANG is used both to describe the quantities being measured, as well as act as the framework for the metadata management. Note that the use of YANG here does not limit the

architecture to devices or sources supporting traditional YANG-based transport protocols. YANG is used to describe the data, regardless of which format, protocol or source it arrives from.

Initially developed in context of the Power and Energy Efficiency work (POWEFF), the authors realized both the potential and the need for this collection and aggregation architecture to become a general framework for collection of a variety of metrics.

There is not much point in knowing the "cost side" of a running system (as in energy consumption or CO2e-emissions) if that cannot be weighed against the "value side" delivered by the system (as in transported bytes, VPN connections, music streaming hours, or number of cat videos, etc.), which means traditional performance metrics will play an equally important role in the collection.

1.3. YANG-based Telemetry Outlook

Much work has already gone into the area of telemetry, YANG, and their intersection. E.g. [I-D.draft-ietf-opsawg-collected-data-manifest-05], [I-D.draft-claise-netconf-metadata-for-collection-03] and [I-D.draft-ietf-nmop-yang-message-broker-integration-06] come to mind. We (the POWEFF authoring team) would like to work with the authoring teams of these drafts to align our joint work. We believe this work generally fits well with the principles outlined in the Network Telemetry Framework [RFC9232].

Many essential data sources in real world deployments do not support any YANG-based interfaces, and that situation is expected to remain for the forseable future, which is why we find it important to be able to ingest data from free form (often REST-based) interfaces, and then add the necessary rigor on the Collector level. Then output the datastreams in formats that existing, mature tools can consume directly.

A couple of collection source examples, just to stimulate the imagination:

* SYSLOG [RFC5424]

* EMAN MIBs over SNMP [RFC7603]

* DMTF’s Redfish REST API [Redfish] (no endorsement)

* Proprietary REST APIs, e.g. [Sensor_Service_Methods] (no endorsement)

* YANG-Push [RFC8641]

In particular, this draft depends on the mapping of YANG-based structures to the typical TSDB tag-based formats described in [I-D.draft-kll-yang-label-tsdb-00].

For the evolution of the YANG-based telemetry area, we believe this approach, combining pragmatism in the telemetry data stream interfaces with rigor and transparency regarding the data content, is key. We would like to make this work fit in with the works of others in the field.

1.4. The Philatelist Name

This specification is about a framework for collection, aggregation and interpretation of timestamped telemetry data. The definition of "philatelist" seems close enough.

1. philatelist

noun. [’fノェヒ☒テヲtノ冤ノェst’] [fih-LAT-uh-list]
A collector and student of postage stamps.

Synonyms
- collector
- aggregator

Figure 1: Source: https://www.synonym.com/synonyms/philatelist

2. Conventions and Definitions

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.

This document uses the terminology defined in [RFC7950].

In addition, this document defines the following terms:

TSDB Time Series Database.

Sensor An entity in a system that delivers a snapshot value of some quantity pertaining to the system. Sensors are identified by their Sensor Path.

Sensor Path A textual representation of the sensor’s address within

the system.

3. Architecture Overview

3.1. Functional Role Diagram

The following role diagram explains the basic concepts of the architecture. Many of the functional units would exist in many instances in a real deployment. For example, in a real deployment there would be lots of Network Elements with the Provider role.

On top we have a Network Orchestrator that ensures the Network Controller functions get a suitable configuration. The Collector, Index and Aggregator functions run as part of one or more Network Controllers. Collectors and Aggregators are responsible for ensuring there is a TSDB Partition (also known as bucket, interval, segment, etc. in various TSDB implementations) to receive the potentially large volumes of telemetry data that they produce themselves, or in the case of the Collector, may configure a Provider Network Element to send the collected data to the TSDB, or to the Collector itself, which then passes it on to the TSDB.

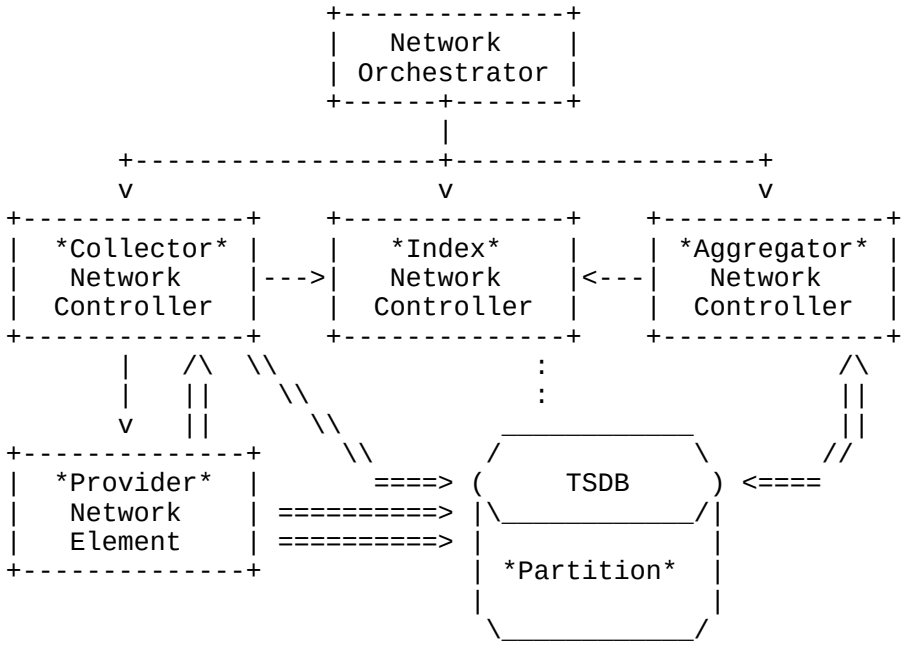


Figure 2: Philatelist Functional Role Diagram.

In the figure, single line arrows indicate control/configuration flow. Double line arrows indicate telemetry data flow. The dotted line between the Index and the TSDB indicates that the index reflects the TSDB partition contents.

3.2. Dashboards

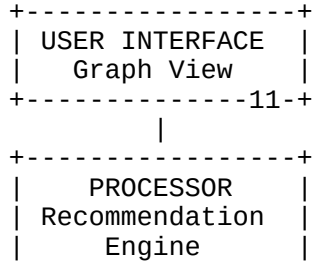
In addition to the functional roles, there is a concept of Dashboards, which is used both within Providers and Collectors. A Dashboard is a particular collection of sensors and controls that have been predefined for particular use cases. Network Elements may implement and publish one or more of these predefined Dashboards, and Controllers may know how to interpret one or more of them. Dashboards contain one or more dashboard items, each one a sensor or control.

For example, a particular Network Element might publish two dashboards. One Dashboard might be called "Current Power Draw" and contain only a single dashboard item which allows the Controller to read out the Network Element's total power draw at this instance. The second Dashboard might be called "Subsystem Power" and contain a tree of dashboard items, which allows the Controller to read out the current power draw of the Network Element's various subsystems. The contents of each Dashboard is defined as a YANG structure in some standards document, or might be a vendor specific YANG definition.

A key point of this architecture is that Dashboard descriptions (in YANG) can be provided also for Network Elements that offer no YANG-based management interfaces at all, or for Network Elements hosting a YANG-based interface, but that were released prior to this document being written.

3.3. Collection Data Flow Tree Diagram

The deployment of a Philatelist framework consists of a collection of Controller plug-in components with well defined interfaces. Here is an example of a deployment. Each box is numbered in the lower right for easy reference.



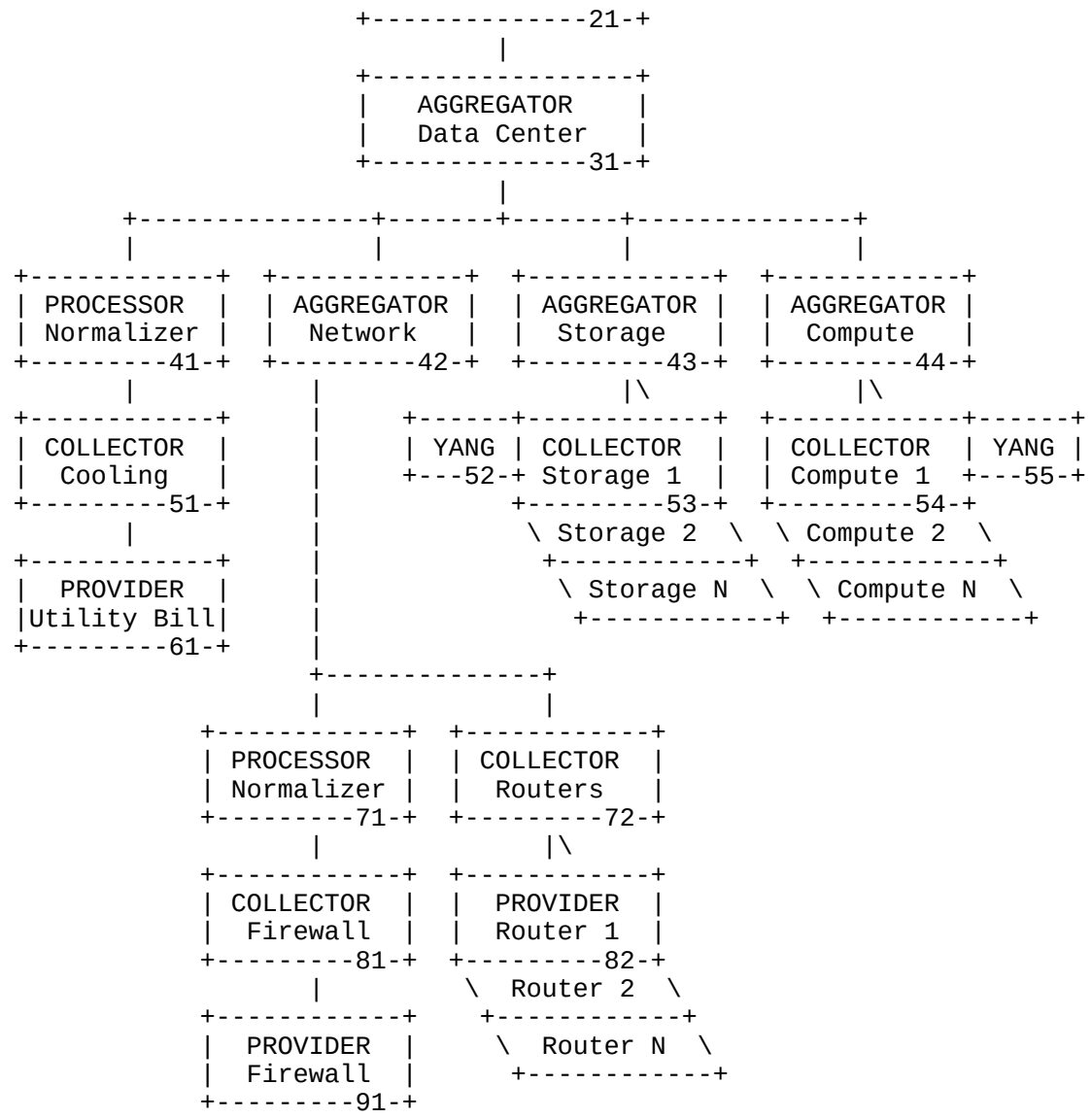


Figure 3: Example Philatelist component deployment.

Each component in the above diagram, represents a logical function. Many of the functions represented by these boxes could be running within a single server, or they could be fully distributed, or, perhaps more likely, something in between.

Provider components (61, 82, 91) are running on a Network Element system that supports a YANG-based telemetry data server. The telemetry data flows from the telemetry source system to a Time Series Database (TSDB).

Collector components (51, 72, 81) ensure the Providers are programmed properly to deliver the telemetry data to the TSDB Partition designated by the Collector. In some cases this flow may be direct (e.g. via a message bus) from the Provider to the TSDB, in other cases, it may be going through the Collector. In some cases the collector may be polling the Provider, in others it may have set up an automatic, periodic subscription.

Many telemetry Provider systems will not have any on-board YANG-based telemetry server. Such servers will instead be managed by a Collector capable of handling a particular kind of Provider (53, 54). Such a Collector is still responsible to set up a telemetry data stream to the Collector’s TSDB. In this case, the Collector will also supply a YANG description (52, 55) of the incoming data stream.

Processor components (21, 41, 71) are transforming the data stream in some way, e.g. converting from one unit of measurement to another, or adjusting the data values recorded to also include some aspect that this particular sensor is not taking into account.

Aggregator components (31, 42, 43, 44) combine the time series telemetry data flows using some operation, e.g. summing, averaging or computing the max or min over them. In this example there are aggregators for Network, Storage, Compute and the entire Data Center

On top of the stack, we may often find a (graphical) user interface (11), for human consumption of the intelligence acquired by the system. Equally relevant is of course an (AI) application making decisions based on findings in the aggregated telemetry flow.

3.4. The Provider Component

A Provider is a Network Element, or any other kind of relevant sensor, that is the source of telemetry data that may or may not offer a YANG-based management interface. It may, for example, provide an SNMP, Redfish or proprietary REST API.

Each Provider typically has a large number of "sensors" that can be polled or in some cases subscribed to. It may also offer some controls (configurables or actionables).

One problem with the sensors is that the sensors relevant for a given use case are often spread around inside the Provider system, and many may not know about all of them. Also, the metadata associated with each sensor is often only missing or only available in human readable form (free form strings), rather than in a strict machine parsable format.

```
/hardware/component[name="psu3"]/.../sensor-data/value
...
/interfaces/interface[name="eth0/0"]/.../out-broadcast-packets
...
/routing/mpls/mpls-label-blocks/.../inuse-labels-count
...
```

Figure 4: Example of scattered potential sensors in a device.

To solve these problems, the Provider YANG module contains a list of Dashboards. Each dashboard contains the sensors and controls useful for some particular use case. The contents of the Dashboards is often defined by a standard, but could also be defined in proprietary YANG modules. Each dashboard item is listed with their sensor paths and machine parsable units, definition and any other metadata.

An admin user or application can then copy the sensor definition from the Dashboard and insert into the Collector’s configuration with items to collect and send to the TSDB.

```
module: ietf-tlm-philatelist-provider
  +--rw tlm-provider
     +--rw dashboards
     |  +--rw dashboard* [id]
     |     +--rw id                        identityref
     |     +--rw items* [tsdb-path]
     |        +--rw tsdb-path              -> ../../../../dash-items/
     |                                       dash-item/tsdb-path
     +--rw dash-items
     |  +--rw dash-item* [tsdb-path]
     |     +--rw tsdb-path                 string
     |     +--rw item-type                 identityref
     |     +--rw accuracy
     |     |  +--rw max-error-relative?    something
     |     |  +--rw max-error-offset?      something
     |     +--rw label* [name]
     |     |  +--rw name                   string
     |     |  +--rw (value-source)?
     |     |     +--:(static-values)
     |     |     |  +--rw static-values*   string
     |     |     +--:(runtime-values)
     |     |        +--rw runtime-values* -> ../../../dash-item/
     |     |                                 tsdb-path
     |     +--rw access-path?              string
     |     +--rw access-params?            -> ../../../accesses/
                                             access/id
```

Figure 5: YANG tree diagram of the Provider Dashboard list.

Note: The "something" YANG-type is used in many places in this document. That is just a temporary placeholder we use until we have figured out what the appropriate type should be.

Each Dashboard in the dashboard list has a name that is an identityref. That makes it possible to define particular dashboards with well known names and contents in YANG, so that Providers and Collectors know what to expect. Each dashboard refers to a set of dashboard items (some of which may be the same in multiple Dashboards). Each dashboard item has a type that is defined as a YANG identity, making them maximally extensible. Examples of sensor types might be a sensor for energy measured in kWh, or energy measured in J, or temperature measured in F, or in C, or in K.

Each dashboard item has an access path and access parameters. These are a mapping into the access mechanism the Collector must use to poll or subscribe to the sensor value.

```
module: ietf-tlm-philatelist-provider
  +--rw tlm-provider
     +--rw accesses
     |  +--rw access* [id]
     |     +--rw id                                  string
     |     +--rw method?                             identityref
     |     +--rw get-local-file-once
     |     |  +--rw filename?                        string
     |     +--rw get-static-url-once
     |     |  +--rw url?                             something
     |     +--rw gnmi-polling
     |     |  +--rw encoding?                        something
     |     |  +--rw protocol?                        something
     |     +--rw restconf-get-polling
     |     |  +--rw xxx?                             string
     |     +--rw netconf-get-polling
     |     |  +--rw xxx?                             string
     |     +--rw restconf-yang-push-subscription
     |     |  +--rw xxx?                             string
     |     +--rw netconf-yang-push-subscription
     |     |  +--rw xxx?                             string
     |     +--rw redfish-polling
     |     |  +--rw xxx?                             string
     |     +--rw frequency?                          sample-frequency
```

Figure 6: YANG tree diagram of the Provider Accesses list.

The list of access methods contains a number of YANG-based and non-YANG based access methods, but this set of access methods can also be extended by YANG-augmentation. The get-local-file-once access method allows reading fixed values from a data sheet encoded in YANG-instance data file format [RFC9195], and the get-static-url-once access method does the same from a given URL. That URL may be served from the Network Element, or from the Collector itself or anywhere else on the network or even internet.

The access-path leaf discussed above (/tlm-provider/dash-items/dash-item/access-path) contains the path to the item that should be read or subscribed to. If the dash-item in question is for a YANG-based interface, then that path would be an XPath expression, with prefixes. Those prefixes need to be mapped to XML namespaces (NETCONF) or YANG module names (RESTCONF). That mapping is provided by the prefix-mappings list.

```
module: ietf-tlm-philatelist-provider
  +--rw tlm-provider
     +--rw prefix-mappings
        +--rw prefix-mapping* [prefix]
           +--rw prefix         string
           +--rw namespace?     string
           +--rw module-name?   string
```

Figure 7: YANG tree diagram of the Provider Prefix Mapping list.

3.5. The Collector Component

The Collector component is part of a Network Controller that collects telemetry data from Providers, typically by periodic polling or subscriptions, and ensures the collected data is stored in a Time Series Database (TSDB). The actual data stream may or may not be passing through the collector component; the collector is responsible for ensuring data flows from the Provider to the destination TSDB, and that the data has a YANG description and is tagged with necessary metadata. How the Collector agrees with a Provider to deliver data in a timely manner is beyond the scope of this document.

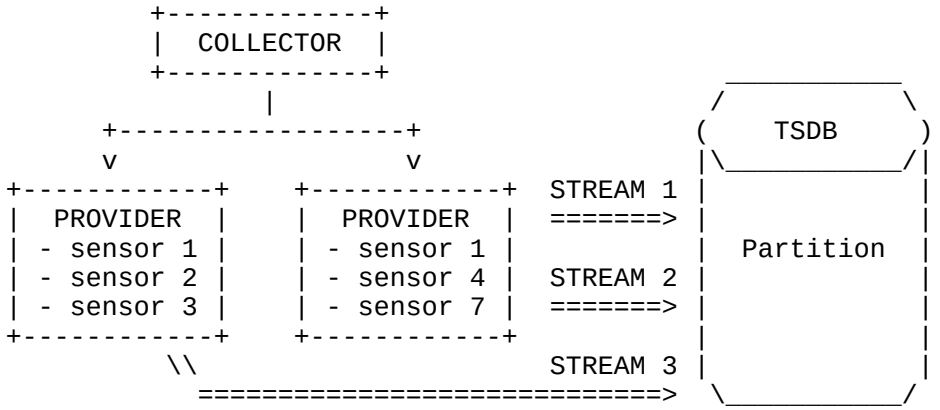


Figure 8: Example of Collector setting up three streams of telemetry data from two Providers to one TSDB Partition.

The top of the Collector model contains a list of organizations, as a single Collector component might be doing collection work for different organizations (customers, departments, scopes) that must not be intermixed. Each organization has a list of device-groups pointing out specific Network Elements. Each group has a list of dashboards that will be queried. Since each Provider has a list of supported dashboards, the Collector simply copies the dashboards it is interested in to its own collection list.

```
module: ietf-tlm-philatelist-collector
  +--rw tlm-collector
     +--rw organizations
        +--rw organization* [name]
           +--rw name                    string
           +--rw device-groups
           |  +--rw device-group* [name]
           |     +--rw name              string
           |     +--rw devices*          string
           |     +--rw dashboards
           |     |  +--rw dashboard* [id]
           |     |     +--rw id          identityref
           |     |     +--rw items* [tsdb-path]
           |     |        +--rw tsdb-path  -> ../../../../
                                               dash-items/
                                               dash-item/
                                               tsdb-path
```

Figure 9: YANG tree diagram of the top part of the Collector model.

The Collector model also contains the same dash-item list as shown above in the Provider model. This allows the Collector configuration to hold a local copy of the dashboards and dash-items it finds relevant, to guide its own collection work.

Finally, the Collector keeps a list of streams to work with, pointing out the sources (device-group and dash-name) and destination (a TSDB Partition).

```
module: ietf-tlm-philatelist-collector
  +--rw tlm-collector
     +--rw organizations
        +--rw organization* [name]
           +--rw tlm-streams
              +--rw tlm-stream* [id]
                 +--rw id                  something
                 +--rw sources* [device-group dash-name]
                 |  +--rw device-group     -> ../../../../
                 |  |                        device-groups/
                 |  |                        device-group/name
                 |  +--rw dash-name        -> ../../../../
                 |                           device-groups/
                 |                           device-group/
                 |                           dashboards/dashboard/id
                 +--rw destination?        partition-ref-t
```

Figure 10: YANG tree diagram of the Collector tlm-streams.

The sensor groups are then arranged into streams from a collection of sources (that support the same set of sensor groups) to a destination. This structure has been chosen with the assumption that there will be many source devices with the same set of sensor groups, and we want to minimize repetition.

3.6. The Index Component

When Collectors collect, and when Aggregators process and aggregate telemetry data, they need to send this data to a TSDB Partition as destination. To keep track of which data is sent where, and what the connection details are for that partition, the Network Controller implements the Index YANG module. Both the Collector and Aggregator modules reference this module.

```
module: ietf-tlm-philatelist-index
  +--rw tlm-index
     +--rw partitions
        +--rw partition* [id]
           +--rw id                    something
           +--rw url?                  something
           +--rw organization?         something
           +--rw partition?            something
           +--rw impl-specific
              +--rw binding* [key]
                 +--rw key             string
                 +--rw (value-type)?
                    +--:(value)
                    |  +--rw value?    string
                    +--:(values)
                    |  +--rw values*   string
                    +--:(env-var)
                       +--rw env-var?  string
```

Figure 11: YANG tree diagram of the Index TSDB Partitions.

The implementation specific part of the model is for integration with specific TSDB implementations. Each such integration may need a specific sef of key-value bindings, that can be provided in this list.

3.7. The Processor and Aggregator Components

Processor components are parts of a Network Controller that take an incoming data flow and transforms it somehow, and possibly augments it with a flow of derived information. The purpose of the transformation could be to convert between different units of measurement, correct for known errors in the input data, or fill in approximate values where there are holes in the input data.

Aggregator components take multiple incoming data flows and combine them, typically by adding them together, taking possible differences in cadence in the input data flows into account.

Processor and Aggregator components provide a YANG model of their output data, just like the Collector components, so that all data flowing in the system has a YANG description and is associated with metadata.

Note: In the current version of the YANG modules, a Processor is simply an Aggregator with a single input and output. Unless we see a need to keep these two component types separate, we might remove the Processor component and keep it baked in with the Aggregator.

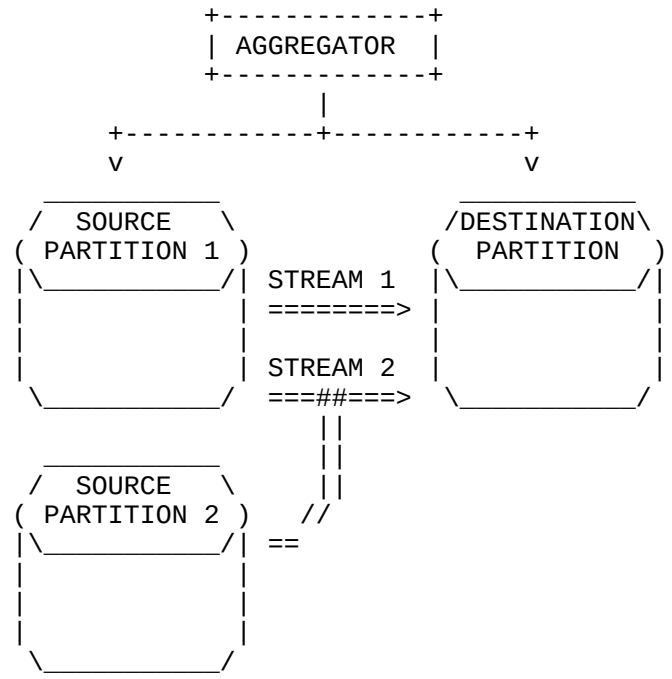


Figure 12: Example of an Aggregator setting up two streams of telemetry data from two sources to one destination.

In this diagram, the sources and destination look like separate TSDBs, which they might be. They may also be different partitions within the same TSDB.

Each stream is associated with one or more inputs, one output and a processing operation. All the input streams are combined using one or more aggregation operations. Some basic operations have been defined in the Aggregator YANG module, but the set of operations has been designed to be maximally extensible.

```
module: ietf-tlm-philatelist-aggregator
  +--rw tlm-aggregator
     +--rw aggregations
     |  +--rw aggregation* [id]
     |     +--rw id           string
     |     +--rw input* [source]
     |     |  +--rw source    partition-ref-t
     |     +--rw operation?   -> ../../../operations/operation/id
     |     +--rw output
     |        +--rw destination?   partition-ref-t
```

Figure 13: YANG tree diagram of the top Aggregator model.

The operations listed below are basic aggregation operations. Linear-sum is just adding all the input sources together, with linear interpolation when their data points don’t align perfectly in time. Rolling average is averaging the input flows over a given length of time. The filter-age drops all data points that are outside the min to max age. The function allows plugging in any other function the Aggregator may have defined, but more importantly, the operations choice is easily extended using YANG augment to include any other IETF or vendor specified augmentations.

```
module: ietf-tlm-philatelist-aggregator
  +--rw tlm-aggregator
     +--rw operations
        +--rw operation* [id]
           +--rw id                        something
           +--rw (op-type)?
              +--:(linear-sum)
              |  +--rw linear-sum
              +--:(linear-average)
              |  +--rw linear-average
              +--:(linear-max)
              |  +--rw linear-max
              +--:(linear-min)
              |  +--rw linear-min
              +--:(rolling-average)
              |  +--rw rolling-average
              |     +--rw timespan?       something
              +--:(filter-age)
              |  +--rw filter-age
              |     +--rw min-age?        something
              |     +--rw max-age?        something
              +--:(function)
                 +--rw function
                    +--rw name?           something
```

Figure 14: YANG tree diagram of the Aggregator operations list.

3.8. The Link to Assets

In [I-D.draft-palmero-ivy-ps-almo-01], the DMLMO team has built an inventory structure that describes systems, subsystems and their soft- and hardware components. They are called assets in the DMLMO YANG models. Some of the collected telemetry data streams may pertain to quite precisely to these assets, and it may be interesting to see the linkage. For this reason, there is an optional module, ietf-tlm-philatelist-assets, that augments the Philatelist Index structure and adds the possibility to point to a DMLMO asset that the TSDB Partition pertains to.

4. YANG Modules

4.1. Base types module for Philatelist

```
<CODE BEGINS>
module ietf-tlm-philatelist-types {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-tlm-philatelist-types";
  prefix ietf-tlm-philatelist-types;

  organization
    "IETF GREEN (Getting Ready for Energy-Efficient Networking)
     Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/group/green/about/>
     WG List:  <mailto:green@ietf.org>
     Editor:  Jan Lindblad
              <mailto:jan.lindblad+ietf@for.eco>
     Editor:  Snezana Mitrovic
              <mailto:snmitrov@cisco.com>
     Editor:  Marisol Palmero
              <mailto:mpalmero@cisco.com>";
  description
    "This YANG module defines base identities for quantities,
    measurement units, connection methods, sensor and control types
    for the Telemetry Philatelist framework.

    Copyright (c) 2024 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
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.

    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.
    ";

  revision 2024-04-15 {
    description
```

```
        "Restructured as part of the Telemetry Philatelist framework";
      reference
        "RFC XXXX: ...";
    }

    typedef something {
      type string;
      description
        "FIXME: Used when we haven't decided the type yet
        ";
    }
    typedef xpath {
      type string;
      description
        "FIXME: Proper type needed
        ";
    }
    typedef sample-frequency {
      type string;
      description
        "FIXME: Proper type needed
        ";
    }

    // ========== SENSOR-CLASS ==============================
    identity sensor-class {
      description
        "Sensor's relation to the asset it sits on.
        ";
    }
    identity sc-input {
      base sensor-class;
      description
        "Sensor reports input quantity of the asset it sits on.
        ";
    }
    identity sc-output {
      base sensor-class;
      description
        "Sensor reports output quantity of the asset it sits on.
        ";
    }
    identity sc-allocated {
      base sensor-class;
      description
        "Sensor reports (maximum) allocated quantity of the asset
        it sits on.
        ";
```

```
}

// ========== SENSOR-QUANTITY ==============================
identity sensor-quantity {
  description
    "Sensor's quantity being measured.
    ";
}

// ========== SENSOR-UNIT ==============================
identity sensor-unit {
  description
    "Sensor's unit of reporting.
    ";
}

// ========== DASH-TYPE ==============================
identity dash-type {
  description
    "The base identity for all dashboard types.
    Dashboards are predefined collections of sensors and/or
    controls that a Network Element publishes towards a Network
    Controller, which uses the dashboard to find use case relevant
    telemetry collection and control points.
    ";
}

// ========== DASH-ITEM-TYPE ==============================
identity dash-item-type {
  description
    "The base identity for an individual item on a dashboard.
    This is further subdivided into controls and sensors, which
    are even further subdivided into sensors measuring e.g.
    temperature in Celsius.
    ";
}
identity sensor-type {
  base dash-item-type;
  description
    "Sensor's type, i.e. combination of class, quantity and unit.
    Sensor class tells whether the sensor measures some quantity
    on the inside or outside of the component it pertains to.
    E.g. whether it is measuring the incoming our outgoing current
    from a power supply.
    Sensor quantity tells what the sensor is measuring.
    E.g. temperature, current or number of packets
    Sensor unit tells about the sensor measurement unit.
    E.g. Celsius, Farenheit or Kelvin. Or energy in kWh or J.
```

```
        ";
    }

    // ========== CONNECTION-METHOD ==============================

    identity connection-method {
      description
        "Base identity for all kinds of connection methods.
        ";
    }
    identity get-local-file-once {
      base connection-method;
      description
        "Connection method identity for the case where a file contains
        the dashboard information in lieu of the Network Element
        itself.
        ";
    }
    identity get-static-url-once {
      base connection-method;
      description
        "Connection method identity for the case where a URL contains
        the dashboard information in lieu of the Network Element
        itself. The URL may or may not be hosted on the Network
        Element.
        ";
    }
    identity cm-polled {
      base connection-method;
      description
        "Connection method identity for all mechanisms that are based
        on the client regularly polling the server.
        ";
    }
  }
  <CODE ENDS>
```

4.2. Dashboard abstract interface module for Philatelist

```
  <CODE BEGINS>
  module ietf-tlm-philatelist-dashboard {
    yang-version 1.1;
    namespace
      "urn:ietf:params:xml:ns:yang:ietf-tlm-philatelist-dashboard";
    prefix ietf-tlm-philatelist-dashboard;

    import ietf-tlm-philatelist-types {
      prefix ietf-tlm-philatelist-types;
```

```
    }

    organization
      "IETF GREEN (Getting Ready for Energy-Efficient Networking)
       Working Group";
    contact
      "WG Web:   <https://datatracker.ietf.org/group/green/about/>
       WG List:  <mailto:green@ietf.org>
       Editor:  Jan Lindblad
                <mailto:jan.lindblad+ietf@for.eco>
       Editor:  Snezana Mitrovic
                <mailto:snmitrov@cisco.com>
       Editor:  Marisol Palmero
                <mailto:mpalmero@cisco.com>";
    description
      "This YANG module defines the Telemetry Philatelist Dashboard.

      These definitions are used both by Philatelist Network
      Controllers, and Philatelist Network Elements. Network Elements
      that are unaware of the Philatelist framework may also be
      covered when a 'proxy mechanism' for the Philatelist information
      is added.

      Copyright (c) 2024 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
      (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
      for full legal notices.

      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.
      ";

    revision 2024-04-15 {
      description
        "Restructured as part of the Telemetry Philatelist framework";
```

```
      reference
        "RFC XXXX: ...";
    }

    identity cm-gnmi {
      base ietf-tlm-philatelist-types:connection-method;
      description
        "Connection method identity for gNMI-based mechanisms.
        ";
    }
    identity cm-restconf {
      base ietf-tlm-philatelist-types:connection-method;
      description
        "Connection method identity for RESTCONF-based mechanisms.
        ";
    }
    identity cm-netconf {
      base ietf-tlm-philatelist-types:connection-method;
      description
        "Connection method identity for NETCONF-based mechanisms.
        ";
    }
    identity cm-redfish {
      base ietf-tlm-philatelist-types:connection-method;
      description
        "Connection method identity for Redfish-based mechanisms.
        ";
    }
    identity gnmi-polling {
      base cm-gnmi;
      base ietf-tlm-philatelist-types:cm-polled;
      description
        "Connection method identity for gNMI-based polling mechanisms.
        ";
    }
    identity restconf-get-polling {
      base cm-restconf;
      base ietf-tlm-philatelist-types:cm-polled;
      description
        "Connection method identity for RESTCONF-based polling
        mechanisms.
        ";
    }
    identity netconf-get-polling {
      base cm-netconf;
      base ietf-tlm-philatelist-types:cm-polled;
      description
        "Connection method identity for NETCONF-based polling
```

```
        mechanisms.
        ";
    }
    identity restconf-yang-push-subscription {
      base cm-restconf;
      description
        "Connection method identity for RESTCONF-based
        subscription mechanisms.
        ";
    }
    identity netconf-yang-push-subscription {
      base cm-netconf;
      description
        "Connection method identity for NETCONF-based
        subscription mechanisms.
        ";
    }
    identity redfish-polling {
      base cm-redfish;
      description
        "Connection method identity for Redfish-based polling
        mechanisms.
        ";
    }
    grouping access-g {
      description
        "Grouping describing the basic set of access methods offered by
        Philatelist servers, i.e. Network Elements. The set of access
        mechanisms may be extended by servers offering additional
        mechanisms.  This grouping is also used by Network Controllers,
        when they need to find a way to interact with the
        Network Elements.
        ";

      leaf method {
        type identityref {
          base ietf-tlm-philatelist-types:connection-method;
        }
        default ietf-tlm-philatelist-types:get-local-file-once;
        description
          "Discriminator pointing out which access mechanism is
          offered. This value controls which detailed configuration
          nodes will be available.
          ";
      }
      container get-local-file-once {
        when "derived-from-or-self(../method,
              'ietf-tlm-philatelist-types:get-local-file-once')";
```

```
          description
            "The server itself does not offer any access mechanism for
            this dashboard item. Instead, philatelist controllers will
            need to read static values from a local on the controller.
            How the file appears in a relevant location on the
            controller is outside the scope of this specification.
            The file format used MUST adhere to RFC9195
            (https://datatracker.ietf.org/doc/rfc9195/)
            ";
          leaf filename {
            type string;
            description
              "The name of the file containing the static data for the
              dashboard item. If the filename is not specified, the
              controller should look for a file with the same name as the
              connection name.
              ";
          }
        }
        container get-static-url-once {
          when "derived-from-or-self(../method,
                'ietf-tlm-philatelist-types:get-static-url-once')";
          description
            "The server points to a URL the controller can use to
            download a file with static values for this dashboard item.
            The URL may or may not be pointing to the server itself, and
            could potentially even point to a URL on the controller
            itself. How the static file contents pointed to by the URL
            appears on the webserver is outside the scope of this
            specification.
            ";
          leaf url {
            type ietf-tlm-philatelist-types:something;
            description
              "The URL that the controller should read in order to
              get the static data about the dashboard item.
              The file format used MUST adhere to RFC9195
              (https://datatracker.ietf.org/doc/rfc9195/)
              ";
          }
        }
        container gnmi-polling {
          when "derived-from-or-self(../method, 'gnmi-polling')";
          description
            "The server points to a gNMI interface the controller can
            poll at regular intervals to read the current sensor value
            for this dashboard item.
            ";
```

```
          leaf encoding {
            type ietf-tlm-philatelist-types:something;
            description
              "The encoding of the data provided by this mechanism,
              such as self-describing-gpb
              ";
          }
          leaf protocol {
            type ietf-tlm-philatelist-types:something;
            description
              "The exact protocol parameters for this gNMI endpoint,
              such as grpc no-tls
              ";
          }
        }
        container restconf-get-polling {
          when "derived-from-or-self(../method, 'restconf-get-polling')";
          description "FIXME";
          leaf xxx { type string; description "FIXME"; }
        }
        container netconf-get-polling {
          when "derived-from-or-self(../method, 'netconf-get-polling')";
          leaf xxx { type string; description "FIXME"; }
          description "FIXME";
        }
        container restconf-yang-push-subscription {
          when "derived-from-or-self(../method,
                'restconf-yang-push-subscription')";
          leaf xxx { type string; description "FIXME"; }
          description "FIXME";
        }
        container netconf-yang-push-subscription {
          when "derived-from-or-self(../method,
                'netconf-yang-push-subscription')";
          leaf xxx { type string; description "FIXME"; }
          description "FIXME";
        }
        container redfish-polling {
          when "derived-from-or-self(../method, 'redfish-polling')";
          leaf xxx { type string; description "FIXME"; }
          description "FIXME";
        }
        leaf frequency {
          when "derived-from(../method,
                'ietf-tlm-philatelist-types:cm-polled')";
          type ietf-tlm-philatelist-types:sample-frequency;
          description
            "The frequency with which the sensor data value collection
```

```
          should happen. E.g. once per 30 minutes, once per 5 minutes,
          once per 30 seconds or on-change.
          ";
      }
    }

    grouping provider-g {
      description
        "Top-level provider grouping.  Devices will implement
        this as a config false container, or as a piece of instance
        data that a controller can read. Controllers implement this
        data structure as config true, configurable and readable by
        the operators.
        ";
      container dashboards {
        description
          "Each device may support one or more dashboards.
          Controllers can then choose the most advanced dashboard they
          are aware of and interested in. A dashboard contains
          a list of sensors and/or controls that a controller may find
          useful for some particular use case.
          ";
        list dashboard {
          key id;
          description
            "List of dashboards supported by a given device or
            controller.
            ";
          leaf id {
            type identityref {
              base ietf-tlm-philatelist-types:dash-type;
            }
            description
              "Formal dashboard id. The dashboard-items in this
              dashboard are found in ../items, and what items are
              available there depends on this dashboard id .
              ";
          }
          list items {
            key tsdb-path;
            description
              "List of dashboard items. Some of the items are sensors
              which provide data to be read, some may be controls that
              the controller may use.
              ";
            leaf tsdb-path {
              type leafref {
                path ../../../../dash-items/dash-item/tsdb-path;
```

```
            }
            description
              "Path to the sensor or control item on the dashboard.
              The format of the path is TSDB style, i.e. used MUST
              conform to the I-D.draft-kll-yang-label-tsdb, e.g.
              interfaces_interface_statistics_in_unicast_pkts
              Each dashboard item may point to multiple sensors,
              as in the example above, where each interface would
              have a counter of incoming unicase packets.
              Each dashboard item may occur in more than one
              dashboard, and is therefore further desribed in
              ../../../../dash-items/dash-item
              ";
          }
        }
      }
    }
    container dash-items {
      description
        "Container for all dashboard items. Each item may be part of
        (referenced by) multiple dashboards.
        ";
      list dash-item {
        key tsdb-path;
        description
          "Dashboard item, a sensor or control item that is part
          of a dashboard, i.e. a collection of sensors and controls
          relevant for a particular use case.
          Each dashboard item may occur in more than one dashboard.
          ";
        leaf tsdb-path {
          type string;
          description
            "Path to the sensor or control item on the dashboard.
            The format of the path is TSDB style, i.e. used MUST
            conform to the I-D.draft-kll-yang-label-tsdb, e.g.
            interfaces_interface_statistics_in_unicast_pkts
            Each dashboard item may point to multiple sensors, as in
            the example above, where each interface would have
            a counter of incoming unicase packets.
            ";
        }
        leaf item-type {
          type identityref {
            base ietf-tlm-philatelist-types:dash-item-type;
          }
          mandatory true;
          description
```

```
        "The item type describes the type of dashboard item this
        is, including if it is a sensor or contol, sitting on
        the inside or outside of the parent component, the
        measurement quantity (e.g. temperature) and unit
        (e.g. Celsius).

        See the ietf-tlm-philatelist-types:dash-item-type for a
        more detailed explanation, or if you intend to define a
        new type of dashboard item.
        ";
    }
    container accuracy {
      when "derived-from(../item-type,
            'ietf-tlm-philatelist-types:sensor-type')";
      description
        "The accuracy of the dasboard item (sensor). The
        accuracy described using two parameters. One is the max
        deviation relative to the sensor reading, the other one
        is a constant deviation offset.

        Example 1: if a sensor might produce values between
        0 and 1000, and the sensor currently reports 555, but
        the actual value is 531, that could be reported as a
        5% relative error with offset 0.

        Example 2: if a sensor might produce values between
        0 and 1000, and the sensor currently reports 0, but
        the actual value is 2, that could be reported as a
        0% relative error with offset 2 (or some slightly
        highger value, e.g. 5).

        The accuracy MUST be described such that there is a 96%
        confidence that the actual value V is within the reported
        value R so that:

          (measured-error) =  | R - V |
          (error-margin)   =  | R * (max-error-relative) |
                               + (max-error-offset)
          p( (measured-error) < (error-margin) ) >= 0.96
        ";
      leaf max-error-relative {
        type ietf-tlm-philatelist-types:something;
        description
          "The part of the accuracy claim that depends on (varies
          in proportion to) the reported value.
          ";
      }
      leaf max-error-offset {
```

```
            type ietf-tlm-philatelist-types:something;
            description
              "The part of the accuracy claim that does not depend on
              (does not vary with) the reported value.
              ";
          }
        }
        list label {
          key name;
          description
            "List of TSDB path labels. A single TSDB path often refer
            to a whole collection of readable sensors. Each such
            sensor is distinguished by the values associated with
            labels in the path. Those labels are listed here.

            Example: interfaces_interface_statistics_in_unicast_pkts
            might have one label interfaces_interface_name. Concrete
            readouts going into the TSDB might have values like

            Metric: interfaces_interface_statistics_in_unicast_pkts
            Value: 5432100
            Labels:
              host = router-01
              interfaces_interface_name = eth0

            As can be seen in the example, a controller might inject
            labels that are not part of the tsdb-path (host), but
            helps the controller keep track of the incoming data
            streams.
            ";
          reference
            I-D.draft-kll-yang-label-tsdb;
          leaf name {
            type string;
            description
              "The name of the label, in TSDB path notation.
              E.g. interfaces_interface_name
              ";
          }
          choice value-source {
            description
              "The source of the values associated with the labels.
              Label values may be provided in several ways:
              + the server may provide values in runtime, in which
                case the runtime-values points to the dashboard item
                to read to get the actual values.
              + the controller or operator may pick one or more
                 values by configuration, in which case they are
```

```
                listed under static-values .
            + the controller may pick one or more values using some
              internal mechanism of its own, in which case they
              will not be visible here.
            ";
          leaf-list static-values {
            type string;
            description
              "One or more values configured by the controller or
              operator designating which instances of this TSDB
              path to collect data about.

              Example: if the label name for this label entry is
              interfaces_interface_name, this could be a configured
              list of interface names to collect data about.
              ";
          }
          leaf-list runtime-values {
            type leafref {
              path ../../../dash-item/tsdb-path;
            }
            description
              "One or more dashboard items to read label values
              from.

              Example: if the label name for this label entry is
              interfaces_interface_name, this could point to a
              dashboard item that collects the names of all
              interfaces, or all interfaces relevant for a
              particular purpose or customer.
              ";
          }
        }
      }
      choice access-path {
        description
        "This is the path used by the client (Network Controller)
        when asking the server (Network Element) for data.
        The format of the access-path depends on the specific
        access mechanism. If the access mechanism is YANG-based,
        this access path would be an XPath string. If the access
        mechanism is SNMP, it would be an OID. A redfish access
        mechanism would use a endpoint-local URL.
        The actual path may be constructed in one of several ways.
        ";
        leaf plain-string {
          type string;
          description
```

```
                "The given string is used as is, without substitution
                of any labels or special characters.
                ";
            }
            leaf string-with-labels {
              type string;
              description
                "The given string contains references of the form

                $(label_name)

                i.e. a dollar-sign, a left-parenthesis,
                the name of the label - case sensitive and without
                whitespace - and finally a right-parenthesis.

                This expression refers to the labels-value pairs from

                ../label/name  and  ../label/value-source

                The reference expression is substituted with the
                label value before use.

                Example: The access path for an SNMP GET-based node
                interfaces_interface_statistics_in_unicast_pkts
                might have access path

                1.3.6.1.2.1.2.2.1.11.$(interfaces_interface_num)

                which gives the final OID string

                1.3.6.1.2.1.2.2.1.11.4711

                if the ../label[name='interfaces_interface_num'] value
                is 4711.
                ";
            }
            leaf url-with-labels {
              type string;
              description
                "The given string contains references of the forms

                {label_name}
                {label_name:3}
                {?label_name}

                etc. as defined in RFC6570.

                This expression refers to the labels-value pairs from
```

```
                ../label/name  and  ../label/value-source

                The reference expression is substituted with the
                label value before use according to the RFC6570 rules.

                Example 1: The access path for a NETCONF get-based node
                interfaces_interface_statistics_in_unicast_pkts
                might have access path

                /if:interfaces/interface[name='{interfaces_interface_na
                me}']/statistics/in-unicast-pkts

                Any prefixes used (as if: above) are mapped to the
                corresponding namespace (NETCONF) or module name
                (RESTCONF) using the ../../../prefix-mappings list.

                Example 2: The access path for a Redfish-based node

                Systems_EthernetInterfaces_EthernetInterfaceMetrics_Dro
                ppedPackets

                might have access path

                /redfish/v1/Systems/VM1/EthernetInterfaces/{interfaces_
                interface_num}/EthernetInterfaceMetrics/DroppedPackets
                ";
            }
          }
          leaf access-params {
            type leafref {
              path ../../../accesses/access/id;
            }
            description
              "Points to the specific access method to be used with
              this dashboard item.
              ";
          }
        }
      }
      container accesses {
        description
          "Holds a list of all the access methods that have been
          defined for dashboard items on the Network Element.
          ";
        list access {
          key id;
          description
            "One specific access method to be used with some dashboard
```

```
            items.
            ";
          leaf id {
            type string;
            description
              "Name for this access method.
              ";
          }
          uses access-g;
        }
      }
      container prefix-mappings {
        description
          "Contains the mappings for prefixes used in the access-path
          to the XML namespace (NETCONF) or module name (RESTCONF).
          ";
        list prefix-mapping {
          key prefix;
          description
            "List of access-path prefixes and their mapping to
            namespace and module name.
            ";
          leaf prefix {
            type string;
            description
              "Prefix, as used in the dash-item/access-path.
              The prefix is case sensitive, and should not contain
              the ending colon (:).
              ";
          }
          leaf namespace {
            type string;
            description
              "XML namespace corresponding to the prefix name.
              Used by NETCONF-based access mechanisms.
              ";
          }
          leaf module-name {
            type string;
            description
              "YANG module name corresponding to the prefix name.
              Used by RESTCONF-based access mechanisms.
              ";
          }
        }
      }
    }
  }
```

```
   <CODE ENDS>
```

4.3. Provider interface module for Philatelist

```
   <CODE BEGINS>
   module ietf-tlm-philatelist-provider {
     yang-version 1.1;
     namespace
       "urn:ietf:params:xml:ns:yang:ietf-tlm-philatelist-provider";
     prefix ietf-tlm-philatelist-provider;

     import ietf-tlm-philatelist-dashboard {
       prefix ietf-tlm-philatelist-dashboard;
     }

     organization
       "IETF GREEN (Getting Ready for Energy-Efficient Networking)
        Working Group";
     contact
       "WG Web:   <https://datatracker.ietf.org/group/green/about/>
        WG List:  <mailto:green@ietf.org>
        Editor:  Jan Lindblad
                 <mailto:jan.lindblad+ietf@for.eco>
        Editor:  Snezana Mitrovic
                 <mailto:snmitrov@cisco.com>
        Editor:  Marisol Palmero
                 <mailto:mpalmero@cisco.com>";
     description
       "This YANG module defines the Telemetry Philatelist Provider
       interface.

       This module is used by Network Elements that have built-in
       support for the Philatelist provider interface, or by a proxy
       mechanism representing some/all of them, when not supported
       directly.

       Copyright (c) 2024 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
       (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
```

```
      for full legal notices.

      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.
      ";

    revision 2024-04-15 {
      description
        "Restructured as part of the Telemetry Philatelist framework";
      reference
        "RFC XXXX: ...";
    }

    container tlm-provider {
      description
        "Container with telemetry collection dashboards for
        Network Elements with built-in support for the Philatelist
        collection framework.
        ";
      uses ietf-tlm-philatelist-dashboard:provider-g;
    }
  }
  <CODE ENDS>
```

4.4. Index interface module for Philatelist

```
  <CODE BEGINS>
  module ietf-tlm-philatelist-index {
    yang-version 1.1;
    namespace
      "urn:ietf:params:xml:ns:yang:ietf-tlm-philatelist-index";
    prefix ietf-tlm-philatelist-index;

    import ietf-tlm-philatelist-types {
      prefix ietf-tlm-philatelist-types;
    }

    organization
      "IETF GREEN (Getting Ready for Energy-Efficient Networking)
       Working Group";
    contact
      "WG Web:   <https://datatracker.ietf.org/group/green/about/>
       WG List:  <mailto:green@ietf.org>
       Editor:  Jan Lindblad
```

```
              <mailto:jan.lindblad+ietf@for.eco>
         Editor:  Snezana Mitrovic
              <mailto:snmitrov@cisco.com>
         Editor:  Marisol Palmero
              <mailto:mpalmero@cisco.com>";
     description
       'This YANG module defines the Telemetry Philatelist Index.
       These definitions are for Philatelist Network Controllers.

       A Network Controller with the Collector role programs one or more
       SOURCES (typically Network Elements) to generate a STREAM of
       telemetry data. The STREAM is sent to a specific DESTINATION
       in a Time Series Database (TSDB).

       Each STREAM consists of timestamped sensor values from each
       sensor in a sensor group.

                 +-------------+
                 |  COLLECTOR  |
                 +-------------+                          ____________
                        |                               /DESTINATION\
               +------------------+                    (  PARTITION  )
               v                  v                   |\____________/|
         +------------+    +------------+  STREAM 1 |              |
         |   SOURCE   |    |   SOURCE   |  =======> |              |
         | - sensor 1 |    | - sensor 1 |           |              |
         | - sensor 2 |    | - sensor 4 |  STREAM 2 |              |
         | - sensor 3 |    | - sensor 7 |  =======> |              |
         +------------+    +------------+           |              |
                  \\                       STREAM 3 |              |
                    ============================>   \____________/

       '+"
       Copyright (c) 2024 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
       (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
       for full legal notices.

       The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',
```

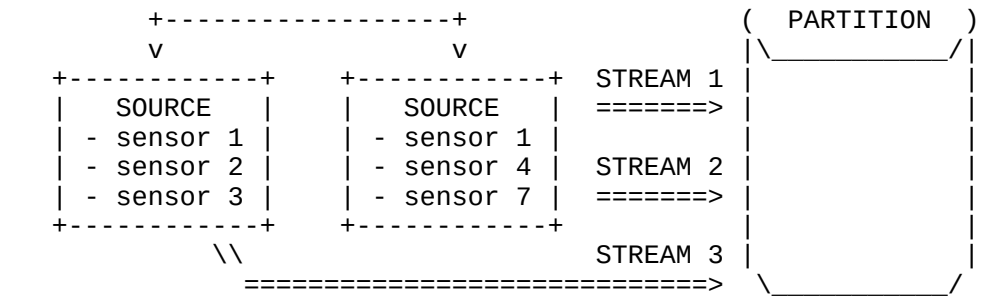
```
    'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED',
    'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document
    are to be interpreted as described in BCP 14 (RFC 2119)
    (RFC 8174) when, and only when, they appear in all
    capitals, as shown here.
    ";

  revision 2024-04-15 {
    description
      "Restructured as part of the Telemetry Philatelist framework";
    reference
      "RFC XXXX: ...";
  }

  typedef partition-ref-t {
    type leafref {
      path
        "/ietf-tlm-philatelist-index:tlm-index"+
        "/ietf-tlm-philatelist-index:partitions"+
        "/ietf-tlm-philatelist-index:partition"+
        "/ietf-tlm-philatelist-index:id";
    }
    description
      "Pointer to a specific TSDB partition
      (aka. bucket, interval, segment, etc.)
      ";
  }
  grouping tsdb-partition-g {
    description
      "Grouping for identifying and connecting to a specific TSDB
      partition (aka. bucket, interval, segment, etc.)
      ";
    leaf url {
      type ietf-tlm-philatelist-types:something;
      description
        "The URL to use to connect to the TSDB.
        ";
    }
    leaf organization {
      type ietf-tlm-philatelist-types:something;
      description
        "The organization this partition belongs to.
        Leaving this unset means the 'default' organization.
        ";
    }
    leaf partition {
      type ietf-tlm-philatelist-types:something;
      description
```

```
          "The TSDB partition (aka. bucket, interval, segment, etc.)
          that the collected data is stored in.
          ";
      }
      container impl-specific {
        description
          "Implementation specific key-valye pairs for establising and
          maintaining the collection connections.
          ";
        list binding {
          key key;
          description
            "List of key-value bindings. The meaning of these key-value
            pairs is implementation dependent.
            ";
          leaf key {
            type string;
            description
              "The key part of the key-value pair.
              The set of key values that are defined is implementation
              dependent.
              ";
          }
          choice value-type {
            description
              "The value part of the key-value pair. The value part may
              have several different formats, and implementations may
              augment yet other formats into this choice.
              ";
            leaf value {
              type string;
              description
                "The value part of the key-value pair as a simple
                string value.
                ";
            }
            leaf-list values {
              type string;
              ordered-by user;
              description
                "The value part of the key-value pair as a collection
                of string values.
                ";
            }
            leaf env-var {
              type string;
              description
                "The value part of the key-value pair. The actual
```

```
                value is provided by an operating system environment
                variable. The name of that environment variable is
                given by this leaf. Case sensitive.

                The set of environment variable names that are defined
                is implementation dependent.
                ";
            }
          }
        }
      }
    }

    container tlm-index {
      description
        "List of TSDB Partitions referenced by this Network Controller.
        ";
      container partitions {
        description
          "Container for all the TSDB Partition access information.
          ";
        list partition {
          key id;
          description
            "TSDB Partition access information for the Partitions that
            this Network Controller is aware of.
            ";
          leaf id {
            type ietf-tlm-philatelist-types:something;
            description
              "The Network Controller's internal identifier for this
              TSDB Partition.
              ";
          }
          uses tsdb-partition-g;
        }
      }
    }
  }
  <CODE ENDS>
```

4.5. Collector interface module for Philatelist

```
<CODE BEGINS>
module ietf-tlm-philatelist-collector {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-tlm-philatelist-collector";
  prefix ietf-tlm-philatelist-collector;

  import ietf-tlm-philatelist-types {
    prefix ietf-tlm-philatelist-types;
  }
  import ietf-tlm-philatelist-dashboard {
    prefix ietf-tlm-philatelist-dashboard;
  }
  import ietf-tlm-philatelist-index {
    prefix ietf-tlm-philatelist-index;
  }

  organization
    "IETF GREEN (Getting Ready for Energy-Efficient Networking)
     Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/group/green/about/>
     WG List:  <mailto:green@ietf.org>
     Editor:  Jan Lindblad
              <mailto:jan.lindblad+ietf@for.eco>
     Editor:  Snezana Mitrovic
              <mailto:snmitrov@cisco.com>
     Editor:  Marisol Palmero
              <mailto:mpalmero@cisco.com>";
  description
    'This YANG module defines the Telemetry Philatelist Collector.
    These definitions are for Philatelist Network Controllers.

    A Network Controller with the Collector role programs one or more
    SOURCES (typically Network Elements) to generate a STREAM of
    telemetry data. A SOURCE may be a Provider, or a proxy mechanism
    standing in for the Provider.
    The STREAM is then sent to a specific DESTINATION PARTITION
    in a Time Series Database (TSDB). Partitions are also known as
    buckets, intervals, segments, etc. in various implementations.

    Each STREAM consists of timestamped sensor values from each
    sensor in a sensor group.

              +-------------+
              |  COLLECTOR  |
              +-------------+                    ____________
                     |                         /DESTINATION\
```




```
    '+"
    Copyright (c) 2024 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
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.

    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.
    ";

  revision 2024-04-15 {
    description
      "Restructured as part of the Telemetry Philatelist framework";
    reference
      "RFC XXXX: ...";
  }

  container tlm-collector {
    description
      "Root container for the Philatelist Collector function.
      ";
    container organizations {
      description
        "List of organizations that the collected data pertains to.
```

```
          Data belonging to one organization will not mix with that of
          another.
          ";
        list organization {
          key name;
          description
            "Organization that this collection process pertains to.
            ";

          leaf name {
            type string;
            description
              "Collector's name of the organization.
              ";
          }
          container device-groups {
            description
              "List of device-groups.
              ";
            list device-group {
              key name;
              description
                "A device-group contains a group of similar devices
                that will have collection performed in the same way.
                ";
              leaf name {
                type string;
                description
                  "Name of the device-group.
                  ";
              }
              leaf-list devices {
                type string;
                description
                  "Points to the devices members of this device-group.
                  The exact meaning of these names is implementation
                  specific.
                  ";
              }
              uses ietf-tlm-philatelist-dashboard:provider-g;
            }
          }

          container tlm-streams {
            description
              "List of telemetry streams pertainin to this
              organization.
              ";
```

```
            list tlm-stream {
              key id;
              description
                "A stream of telemetry data that is collected from a
                device-group that share a particular dashboard.
                ";
              leaf id {
                type ietf-tlm-philatelist-types:something;
                description
                  "Identifier of the telemetry stream.
                  ";
              }
              list sources {
                key "device-group dash-name";
                description
                  "List of sources to collect from. Each source points
                  to a device-group and a dashboard name to collect
                  from each device in the device-group.
                  ";
                leaf device-group {
                  type leafref {
                    path ../../../../device-groups/device-group/name;
                  }
                  description
                    "The device-group to collect from.
                    ";
                }
                leaf dash-name {
                  type leafref {
                    path "../../../../device-groups/device-group/"+
                      "dashboards/dashboard/id";
                  }
                  description
                    "The name of the dashboard
                    ";
                }
              }
              leaf destination {
                type ietf-tlm-philatelist-index:partition-ref-t;
                description
                  "The TSDB Partition (bucket, interval, segment, etc.)
                  that the collected telemetry data is sent to.
                  ";
              }
            }
          }
        }
      }
```

```
    }

    augment "/tlm-collector/organizations/organization/device-groups/"+
      "device-group/dash-items/dash-item/label/value-source" {
      description
        "Some additional value-sources that the collector enables
        (that are not available to providers).
        ";
      case controller-managed-value {
        leaf controller-managed-value {
          type empty;
          description
            "The Collector will determine the label value by its
            own discretion.
            ";
        }
      }
      case controller-provided-value {
        leaf controller-provided-value {
          type string;
          description
            "The Collector will determine the label value by its
            own discretion, and that value is configured here.
            ";
        }
      }
    }
  }
  <CODE ENDS>
```

4.6. Aggregator interface module for Philatelist

```
  <CODE BEGINS>
  module ietf-tlm-philatelist-aggregator {
    yang-version 1.1;
    namespace
      "urn:ietf:params:xml:ns:yang:ietf-tlm-philatelist-aggregator";
    prefix ietf-tlm-philatelist-aggregator;

    import ietf-tlm-philatelist-types {
      prefix ietf-tlm-philatelist-types;
    }
    import ietf-tlm-philatelist-index {
      prefix ietf-tlm-philatelist-index;
    }

    organization
      "IETF GREEN (Getting Ready for Energy-Efficient Networking)
```

```
   Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/group/green/about/>
     WG List:  <mailto:green@ietf.org>
     Editor:  Jan Lindblad
              <mailto:jan.lindblad+ietf@for.eco>
     Editor:  Snezana Mitrovic
              <mailto:snmitrov@cisco.com>
     Editor:  Marisol Palmero
              <mailto:mpalmero@cisco.com>";
  description
    'This YANG module defines the Telemetry Philatelist Aggregator.
    These definitions are for Philatelist Network Controllers.

    An AGGREGATOR ensures data from one or more SOURCE(s) are
    combined into a FLOW using a (sequence of) OPERATIONs (OPs)
    to generate a new data set in the DESTINATION (which could
    be a new collection in the same data storage system as the
    SOURCE).

                  +-------------+
                  | AGGREGATOR  |
                  +-------------+
                         |
              +----------+-------------+
              v                        v
          ___________                 ___________
         /  SOURCE   \               /DESTINATION\
        ( PARTITION 1 )             (  PARTITION  )
        |\___________/| STREAM 1    |\___________/|
        |             | ========>   |             |
        |             |             |             |
        |             | STREAM 2    |             |
         \___________/  ===##===>    \___________/
                           ||
          ___________      ||
         /  SOURCE   \     ||
        ( PARTITION 2 )   //
        |\___________/| ==
        |             |
        |             |
        |             |
         \___________/

    '+"
    Copyright (c) 2024 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
      (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
      for full legal notices.

      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.
      ";

    revision 2024-04-15 {
      description
        "Restructured as part of the Telemetry Philatelist framework";
      reference
        "RFC XXXX: ...";
    }

    container tlm-aggregator {
      description
        "Root container for the Philatelist Aggregator function.
        ";
      container aggregations {
        description
          "List of aggregation operations that are applied to a set of
          input telemetry streams in a TSDB Partition to form an output
          stream in a different TSDB Partition.
          ";
        list aggregation {
          key id;
          description
            "Each aggregation takes one or more input streams from a
            TSDB Partition, an operation to apply to them, and points
            to an output stream an another TSDB Partition.
            ";
          leaf id {
            type string;
            description
              "The internal id of this aggregation operation.
              ";
```

```
          }
          list input {
            key source;
            description
              "The list of sources/input streams for the aggregation.
              ";
            leaf source {
              type ietf-tlm-philatelist-index:partition-ref-t;
              description
                "The TSDB Partition (bucket, interval, segment, etc.)
                that the input telemetry data is read from.
                ";
            }
          }
          leaf operation {
            type leafref {
              path ../../../operations/operation/id;
            }
            description
              "The operation to apply to the input stream(s) in order
              to compute the output stream.
              ";
          }
          container output {
            description
              "The TSDB Partition to send the computed output to.
              ";
            leaf destination {
              type ietf-tlm-philatelist-index:partition-ref-t;
              description
                "The TSDB Partition (bucket, interval, segment, etc.)
                that the aggregated telemetry data is sent to.
                ";
            }
          }
        }
      }
      container operations {
        description
          "The operations that may be applied during the aggregation.
          ";
        list operation {
          key id;
          description
            "Details about which operation to apply and how.
            ";
          leaf id {
            type ietf-tlm-philatelist-types:something;
```

```
        description
          "The internal id of the operation to apply.
          ";
      }
      choice op-type {
        description
          "A choice of basic operation types. This set of choices
          may be extended by other modules agumenting the choice.
          ";
        container linear-sum {
          description
          "This operation produces the sum of the input streams.

          Since the data points in the input stream are not
          likely to be perfectly aligned in time, this linear-sum
          operation produces a linear interpolation between each
          point in the time series.

          This works well when all inputs are continuously
          receiving additional data points, and when their
          frequency is roughly the same. A different summing
          operation should be chosen if this is not the case,
          e.g. most-recent-sum.
          ";
        }
        container linear-average { description "FIXME"; }
        container linear-max { description "FIXME"; }
        container linear-min { description "FIXME"; }
        container rolling-average {
          description "FIXME";
          leaf timespan {
            type ietf-tlm-philatelist-types:something;
            description "FIXME";
          }
        }
        container filter-age {
          description "FIXME";
          leaf min-age {
            type ietf-tlm-philatelist-types:something;
            description "FIXME";
          }
          leaf max-age {
            type ietf-tlm-philatelist-types:something;
            description "FIXME";
          }
        }
        container function {
          description "FIXME";
```

```
                leaf name {
                  type ietf-tlm-philatelist-types:something;
                  description "FIXME";
                }
              }
            }
          }
        }
      }
    }
    <CODE ENDS>
```

4.7. Assets interface module for Philatelist

```
    <CODE BEGINS>
    module ietf-tlm-philatelist-assets {
      yang-version 1.1;
      namespace
        "urn:ietf:params:xml:ns:yang:ietf-tlm-philatelist-assets";
      prefix ietf-tlm-philatelist-assets;

      import ietf-lmo {
        prefix ietf-lmo;
      }
      import ietf-lmo-assets {
        prefix ietf-lmo-assets;
      }
      import ietf-tlm-philatelist-index {
        prefix ietf-tlm-philatelist-index;
      }

      organization
        "IETF GREEN (Getting Ready for Energy-Efficient Networking)
         Working Group";
      contact
        "WG Web:   <https://datatracker.ietf.org/group/green/about/>
         WG List:  <mailto:green@ietf.org>
         Editor:  Jan Lindblad
                  <mailto:jan.lindblad+ietf@for.eco>
         Editor:  Snezana Mitrovic
                  <mailto:snmitrov@cisco.com>
         Editor:  Marisol Palmero
                  <mailto:mpalmero@cisco.com>";
      description
        "This YANG module defines the Telemetry Philatelist linkage to
        DMLMO Assets.
        These definitions are for Philatelist Network Controllers.
```

```
      Copyright (c) 2024 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
      (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
      for full legal notices.

      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.
      ";

    revision 2024-04-15 {
      description
        "Initial revision.";
      reference
        "RFC XXXX: ...";
    }

    grouping asset-pointer-g {
        description
          "Pointer to an LMO asset.
          ";
      leaf pertains-to-asset-class {
        type leafref {
          path /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:lmo-class;
        }
        must
          "derived-from-or-self(current(), 'ietf-lmo-assets:asset')";
        must "../pertains-to-asset-id";
        description
          "The LMO Asset class of the asset.
          ";
      }
      leaf pertains-to-asset-id {
        type leafref {
          path "/ietf-lmo:lmos/ietf-lmo:lmo"+
            "[ietf-lmo:lmo-class=current()/../pertains-to-asset-class]"
```

```
            +"/ietf-lmo:inst/ietf-lmo:id";
        }
        description
          "The LMO Asset id (within the class) of the asset.
          ";
      }
    }

    augment "/ietf-tlm-philatelist-index:tlm-index"+
            "/ietf-tlm-philatelist-index:partitions"+
            "/ietf-tlm-philatelist-index:partition" {
      description
        "By augmenting an asset pointer into the TSDB Partition Index,
        controller may clarify which LMO Asset the data in the
        Partition pertains to.
        ";
      uses asset-pointer-g;
    }
  }
  <CODE ENDS>
```

5. Security Considerations

TODO Security

6. IANA Considerations

This document has no IANA actions.

7. Changes (to be deleted by RFC Editor)

7.1. From version -02 to -03

* Updated author affiliation and document working group

* Updated references to other documents

7.2. From version -01 to -02

* Adopted [RFC6570] style URI Templates in ietf-tlm-philatelist-dashboard

* Moved up the outlook section, and clarified the relation to existing device side telemetry solutions.

* Added several informative references to prior work

* Reformatted YANG modules for shorter lines to fit IETF layout

7.3. From version -00 to -01

* Introduced Dashboard and Index concepts
* Restructured YANG into three controller modules: collector, index, aggregator
* Restructured YANG into one device module: provider
* Restructured YANG common parts into one abstract module: dashboard
* Split YANG modules, some contents going into poweff-specific modules
* Renamed remaining YANG modules from -poweff- to -tlm-philatelist-
* Updated text to reflect new module organization
* Added optional linkage to DMLMO assets

7.4. Version -00

* Initial version.

8. References

8.1. Normative References

[I-D.draft-kll-yang-label-tsdb-00]
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.draft-palmero-ivy-ps-almo-01]
Palmero, M., Brockners, F., Kumar, S., Cardona, C., and D. Lopez, "Asset Lifecycle Management and Operations: A Problem Statement", Work in Progress, Internet-Draft, draft-palmero-ivy-ps-almo-01, 24 April 2024, <https://datatracker.ietf.org/doc/html/draft-palmero-ivy-ps-almo-01>.

[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/rfc/rfc2119>.

[RFC6570] Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., and D. Orchard, "URI Template", RFC 6570, DOI 10.17487/RFC6570, March 2012, <https://www.rfc-editor.org/rfc/rfc6570>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/rfc/rfc7950>.

[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/rfc/rfc8174>.

[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/rfc/rfc9195>.

8.2. Informative References

[I-D.draft-claise-netconf-metadata-for-collection-03] 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-03, 25 January 2022, <https://datatracker.ietf.org/doc/html/draft-claise-netconf-metadata-for-collection-03>.

[I-D.draft-ietf-nmop-yang-message-broker-integration-06] Graf, T. and A. Elhassany, "An Architecture for YANG-Push to Message Broker Integration", Work in Progress, Internet-Draft, draft-ietf-nmop-yang-message-broker-integration-06, 5 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-yang-message-broker-integration-06>.

[I-D.draft-ietf-opsawg-collected-data-manifest-05] Claise, B., Quilbeuf, J., Lopez, D., Martinez-Casanueva, I. D., and T. Graf, "A Data Manifest for Contextualized Telemetry Data", Work in Progress, Internet-Draft, draft-ietf-opsawg-collected-data-manifest-05, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-collected-data-manifest-05>.

[Redfish] "DMTF Redfish", 13 January 2025, <https://www.dmtf.org/standards/redfish>.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, DOI 10.17487/RFC5424, March 2009, <https://www.rfc-editor.org/rfc/rfc5424>.

[RFC7603] Schoening, B., Chandramouli, M., and B. Nordman, "Energy Management (EMAN) Applicability Statement", RFC 7603, DOI 10.17487/RFC7603, August 2015, <https://www.rfc-editor.org/rfc/rfc7603>.

[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/rfc/rfc8641>.

[RFC9232] Song, H., Qin, F., Martinez-Julia, P., Ciavaglia, L., and A. Wang, "Network Telemetry Framework", RFC 9232, DOI 10.17487/RFC9232, May 2022, <https://www.rfc-editor.org/rfc/rfc9232>.

[Sensor_Service_Methods] "Schneider Electric Sensor Service Methods", 20 June 2024, <https://community.se.com/t5/DCE-web-services-API/Sensor-Service-Methods/ta-p/446584>.

Acknowledgments

Kristian Larsson has provided invaluable insights, experience and validation of the design. Many thanks to the entire POWEFF team for their committment, flexibility and hard work behind this. Thanks to James Henderson for the review, a number of small fixes and several good susggestions. Hat off to Benoテョt Claise, who inspires by the extensive work produced in IETF over the years, and in this area in particular.

Author’s Address

Jan Lindblad
All For Eco
Email: jan.lindblad+ietf@for.eco