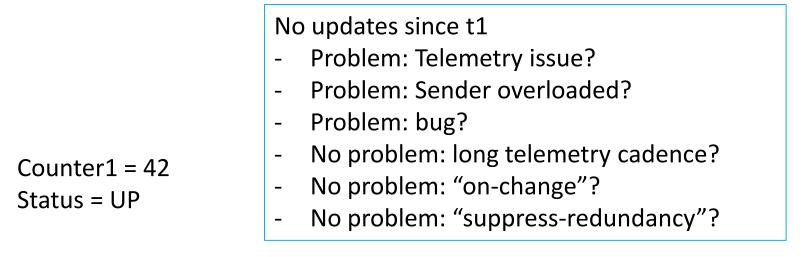# Data Manifest for Streaming Telemetry

draft-claise-opsawg-collected-data-manifest-01
B. Claise (Huawei), J. Quilbeuf (Huawei), D. Lopez (Telefonica), I. Dominguez (UPM), T. Graf (Swisscom)

IETF 113, OPSAWG

Goal & Problem Statement

- Goal is not to expose new information via YANG but rather to define what needs to be kept as metadata (or Data Manifest) to ensure that the data can still be interpreted correctly even:
  - if the device is not accessible (from the collection system)
  - If the device has been updated or has a new configuration
- End goal: analyze the data, from the data collection system, with the proper context, for anomaly detection and, in the end, closed loop automation
- Per-node <u>capability</u> discovery exists
  - YANG Modules describing Capabilities for Systems and Datastore Update Notifications, RFC9196 + YANG Instance Data File Format, RFC9195
  - Per-Node Capabilities for Optimum Operational Data Collection , draft-claise-netconf-metadata-forcollection-03
- But how were data <u>actually</u> metered, under which circumstances?

Data Collection Vantage Point

No updates since t1
- Problem: Telemetry issue?
- Problem: Sender overloaded?
- Problem: bug?
- No problem: long telemetry cadence?
- No problem: "on-change"?
- No problem: "suppress-redundancy"?

Counter1 = 42
Status = UP

t1 ——————————————————————————— Now → Time

End goal: analyze the data, from the data collection system, with the proper context, for anomaly detection and, in the end, closed loop automation
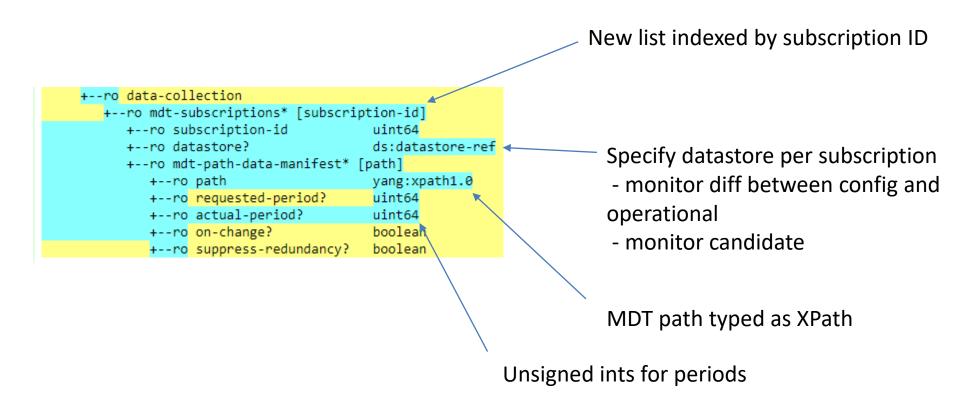
Proposal: Data Manifest

- Data Manifest composed of 2 YANG models for storing the context:
  - Platform manifest: part of the Data Manifest that completely characterizes the platform producing the data.
  - Data Collection manifest: part of the Data Manifest that completely characterizes how and when the telemetry was metered.
- "MUST be streamed all with the data and stored along with the collected data."
- "In case the data are moved to different place (typically a database), the data manifest MUST follow the collected data."

Platform Manifest

```
module: ietf-collected-data-platform-manifest
  +--ro platform
     +--ro name?                  string
     +--ro vendor?                string
     +--ro software-version?      string
     +--ro software-flavor?       string
     +--ro os-version?            string
     +--ro os-type?               string
     +--ro yang-library
     |  +--ro module-set* [name]
     |  |  +--ro name                   string
     |  |  +--ro module* [name]
     |  |  |  +--ro name          yang:yang-identifier
     |  |  |  +--ro revision?     revision-identifier
     |  |  |  +--ro namespace     inet:uri
     |  |  |  +--ro location*     inet:uri
     |  |  |  +--ro submodule* [name]
     |  |  |  |  +--ro name          yang:yang-identifier
     |  |  |  |  +--ro revision?     revision-identifier
     |  |  |  |  +--ro location*     inet:uri
     |  |  |  +--ro feature*      yang:yang-identifier
     |  |  |  +--ro deviation*    -> ../../module/name
     |  |  +--ro import-only-module* [name revision]
     |  |     +--ro name          yang:yang-identifier
     |  |     +--ro revision      union
     |  |     +--ro namespace     inet:uri
     |  |     +--ro location*     inet:uri
     |  |     +--ro submodule* [name]
     |  |        +--ro name          yang:yang-identifier
     |  |        +--ro revision?     revision-identifier
     |  |        +--ro location*     inet:uri
     |  +--ro schema* [name]
     |  |  +--ro name            string
     |  |  +--ro module-set*     -> ../../module-set/name
     |  +--ro datastore* [name]
     |     +--ro name        ds:datastore-ref
     |     +--ro schema      -> ../../schema/name
     +--ro packages-set
        +--ro package* [name version]
           +--ro name          -> /pkgs:packages/package/name
           +--ro version       -> /pkgs:packages/package[pkgs:name = current()/../name]/version
           +--ro checksum?     -> /pkgs:packages/package[pkgs:name = current()/../name][pkgs:version = current()/../version]/pkgs:checksum
```

New: Vendor

New: Include full yang-library, to have datastores

New: rw->ro for every node, Manifest is NOT configurable

New: Add packages-set

Data Collection Manifest

New list indexed by subscription ID

```
+--ro data-collection
   +--ro mdt-subscriptions* [subscription-id]
      +--ro subscription-id            uint64
      +--ro datastore?                 ds:datastore-ref
      +--ro mdt-path-data-manifest* [path]
         +--ro path                     yang:xpath1.0
         +--ro requested-period?        uint64
         +--ro actual-period?           uint64
         +--ro on-change?               boolean
         +--ro suppress-redundancy?     boolean
```

Specify datastore per subscription
 - monitor diff between config and operational
 - monitor candidate

MDT path typed as XPath

Unsigned ints for periods

Update Frequency

- Platform manifest:

  - **Update when**: platform changes (i.e. at reboot)

- Data collection manifest:

  - **Update when**: collection condition changes:

  - New subscription

  - Collection period is adjusted based on CPU availability

Collecting Manifests: Use Telemetry

- Platform and data collection manifests are about telemetry
    - we can assume a telemetry system is available
- Event-driven Telemetry/onChange well suited to stream the manifests updates only
- Collectors can choose or not to collect the manifests / header approach would force collector to have the manifests
- Collectors know IDs of their subscription -> YANG key for selecting the data-manifest they need

Mapping Data to Data Manifest

- Collected data needs the following metadata:
  - Source device -> for mapping to platform manifest
  - Subscription ID
  - MDT path

  For mapping data to data manifest

Specifying how to store metadata for collected data is out-of-scope of this draft.

Open Questions

- Identified Improvements:
  - Include the source of data somewhere and notably some self-assertiveness
  - Data integrity
- Open Questions:
  - Do we want to handle data manifest for non-telemetry values ? SNMP, IPFIX …?
  - How to properly specify devices / virtual devices as source?
  - Handle mis-collections ? More Counters ?
  - Include yang-library augments for yang-packages and revisions? how? (augment of an augment)

Conclusion

- New draft version (Thanks Med & Tianran & Ignacio for your review)

- Do you recognize the problem statement?

- We would like to request for WG adoption

Feedback, suggestions, issues, PRs:
https://github.com/JeanQuilbeufHuawei/draft-collected-data-manifest