An Architecture for YANG-Push to Message Broker Integration
draft-ietf-nmop-yang-message-broker-integration-04

Motivation and architecture of a native YANG-Push notifications and YANG Schema integration into Message Broker and YANG Schema Registry
Agenda Items

➢ Document Introduction
➢ Status and next steps
➢ Related YANG-Push integration and specification gaps, introduction and status
Nowadays network operators are using machine and human readable YANG RFC 7950 to model their configurations and obtain YANG modelled data from their networks. 

Network operators organizing their data in a Data Mesh where a message broker such as Apache Kafka facilitates the exchange of messages among data processing components. 

Today, subscribing to a YANG datastore, publishing a YANG modeled notifications message from the network and viewing the data in a time series database, manual labor is needed to perform data transformation to make a message broker and its data processing components with YANG notifications interoperable.

« Even though YANG is intend to ease the handling of data, this promise has not yet been fulfilled for Network Telemetry RFC 9232 »
From YANG-Push to Network Analytics
Aiming for an automated data processing pipeline

• A network operator aims for:
  • An **automated data processing pipeline** which starts with YANG-Push, consolidates at Data Mesh and ends at Network Analytics.
  • Operational metrics where **IETF defines the semantics**.
  • Analytical metrics where network operators gain actionable insights.

• We achieve this by integrating YANG-Push into Data Mesh to:
  • Produce metrics from networks **with timestamps when network events were observed**.
  • Hostname, publisher ID and sequence numbers help us to understand from where metrics were exported and measure its delay and loss.
  • Forward **metrics unchanged** from networks
  • **Learn semantics** from networks and validate messages.
  • **Control semantic** changes end to end.
Elements of the Architecture

Workflow Diagram

- **Network Orchestration** subscribes to YANG datastore.

- **Network Node** informs Data Collection on subscription state and publishes YANG metrics with YANG-Push.

- **Data Collection** obtains for each subscription the YANG module dependencies and the YANG modules on the network node, registers it in the YANG Schema Registry and prefixes the forwarded YANG notifications with the obtained schema ID.

- **YANG Schema Registry** issues for a Message Broker subject a schema ID for each new schema tree, compares a new schema tree with an existing and versions it.

- **Data Consumer** consumes YANG-Push notifications from Message Broker, obtains schema tree from YANG schema registry, validates YANG notifications against schema and uses schema to populate into database table.
An Architecture for YANG-Push to Apache Kafka Integration
Status, Summary and Next steps

Status
• Document adopted at NMOP.
• Feedback from Dhruv, Andy, Feng, Qin, Kristian and Nacho addressed in -04. Thank you!

Changes in -04
• Expanded last paragraph in introduction section to detail manual work currently needed in the end-to-end data processing chain due to missing YANG schema
• Figure 1 in Section 3 and Section 3.1 now considers in step 1 to perform the YANG-Push notification capabilities described in Section 3 of RFC 9196
• Added the message broker component in figure 1 in Section 3
• Added section 3.8 describing that observation-time is used for times series metric indexing
• Moved Section 4 and 5 to appendix and Applied RFC 8792 to handle long lines
• Used the boiler plate from RFC 7942 and moved section before Security Considerations
• Section 3.1 describes that in the described architecture both, dynamic and configured YANG-Push subscriptions are supported. To add clarity, an additional paragraph was added detailing on how being subscribed, messages are published in same or different transport session.
• Changed terminology from Time Series Database and Time Series Database Ingestion to Data Storage and Data Consumer.

Next Steps
➢ Looking forward for review and comments.
Address YANG Specification and Integration Gaps
Aiming for an automated data processing pipeline

YANG Specifications Gaps:
• YANG model for NETCONF Event Notifications
draft-ahuang-netconf-notif-yang
• Validating anydata in YANG Library context
draft-aelhassany-anydata-validation

YANG Integration Gaps:
• Support of Network Observation Timestamping in YANG Notifications
draft-tgraf-netconf-yang-push-observation-time
• Support of Hostname and Sequencing in YANG Notifications
draft-tgraf-netconf-notif-sequencing
• Support of Versioning in YANG Notifications Subscription
draft-ietf-netconf-yang-notifications-versioning
• Augmented-by Addition into the IETF-YANG-Library
draft-lincl-netconf-yang-library-augmentation
YANG model for NETCONF Event Notifications

Entire YANG-Push messages is modeled in YANG

- YANG model for NETCONF Event Notifications, draft-ahuang-netconf-notif-yang, updates RFC 5277 by defining the schema as a YANG module.

- Enables YANG-push to define YANG semantics for the entire YANG-push message to support other encodings than XML such as YANG-JSON RFC 7951 or YANG-CBOR RFC 9264.

➢ Changes in -05: Updates and describes relationship to RFC 5277, RFC 8639, RFC 7951 and RFC 9254 in terms of notification structure.
Extend Netconf Notifications with **Hostname and Sequence Number**

For push-update and push-change-update

---

**module:** ietf-notification

```yaml
structure notification:
    +-- eventTime       yang:date-and-time
    +-- inotifseq:sysName inet:host
    +-- inotifseq:sequenceNumber yang:counter32

```

---

**Changes in -06:** Defined new NETCONF and YANG-Push notification capabilities and described how a systems discovers them.

**When NETCONF event notification messages are forwarded from a YANG push receiver to another system,** a message broker or a time series database where the messages are stored, the transport context is lost since it is not part of the NETCONF event notification message metadata. Therefore, the downstream system is unable to associate the message to the publishing process (the exporting router), nor able to detect message loss or reordering.

**draft-tgraf-netconf-notif-sequencing** extends the NETCONF notification defined in **RFC5277** with:

- **sysName:** Describes the hostname following the 'sysName' object definition in **RFC 1213** from where the message was published from.

- **sequenceNumber:** Generates a unique sequence number as described in **RFC 9187** for each published message.
Extend YANG-Push Notifications with Observation Timestamping

For push-update and push-change-update

module: ietf-yp-observation-time

augment /yp:push-update:
  +--ro observation-time?  yang:date-and-time
  +--ro point-in-time?    enumeration

augment /yp:push-change-update:
  +--ro observation-time?  yang:date-and-time
  +--ro point-in-time?    enumeration

augment /sysc:system-capabilities/notc:subscription-capabilities:
  +--ro yang-push-observation-supported?
    notifseq:notification-support
    (yang-push-observation-timestamp)?

    
    "ietf-notification:notification": {
      "eventTime": "2023-03-25T08:30:11.22Z",
      "ietf-notification-sequencing:sysName": "example-router",
      "ietf-notification-sequencing:sequenceNumber": 1,
      "ietf-yp-observation-time:observation-time": "2023-02-04T16:30:09.44Z",
      "ietf-yp-observation-time:point-in-time": "current-accounting",
      "datastore-contents": {
        "ietf-interfaces:interfaces": [
          {
            "interface": {
              "name": "eth0",
              "type": "iana-if-type:ethernetCsmacd",
              "oper-status": "up",
              "mtu": 1500
            }
          }
        ]
      }
    }

- To correlate network data among different Network Telemetry planes as described in Section 3.1 of RFC 9232 or among different YANG push subscription types defined in Section 3.1 of RFC 8641, network observation timestamping is needed to understand the timely relationship among these different planes and YANG push subscription types.

- draft-tgraf-netconf-yang-push-observation-time extends the YANG push streaming update notification defined in RFC 8641 with:
  - **observation-time**: Describes the measurement observation time for the "push-update" notification in a "periodical" and for the "push-change-update" notification in a "on-change" subscription.
  - **point-in-time**: Describes at which point in time the value of observation-time was observed.

- Changes in -02: Changed semantics; observation-time describes when and point-in-time at which point in time. Added new YANG-Push notification capabilities.
Support of Versioning in YANG Notifications Subscription

For subscription state change notification messages

- Network operators need to control semantics in its data processing pipeline. That includes YANG-Push.
- This is today only possible during YANG-Push subscription but not when nodes are being upgraded or when messages are being published for configured subscription.
- draft-ietf-netconf-yang-notifications-versioning extends the YANG push subscription and publishing mechanism defined in RFC 8641:
  - By adding the ability to subscribe to a specific revision or latest-compatible-semversion of one or more yang modules.
  - By extending the YANG push Subscription State Change Notifications Message so that the YANG push receiver learns beside the xpath and the subtree filter also the yang module name, revision and revision-label.

➢ Changes in -05: Changed ietf-yang-push.yang augmentation to resolve YANG issue that within a "case" statement identifiers need to be unique.
Augmented-by Addition
YANG Library Extension

module: ietf-yang-library
  +--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 yanglib-aug:augmented-by* => ../../../yanglib:module/name

- With YANG-Push xpath or sub-tree a part of the YANG-Push data tree is subscribed.
- With YANG Library the relationship among the subscribed YANG modules can be determined from the top of the YANG tree. **What is missing is the ability to discover dependencies within the YANG tree.**
  - **By adding augmented-by YANG module relation.**
- **Changes in -01:** Section 3.1 is now detailing the relationship to YANG-Push to Message Broker Integration architecture and how its being applied.
Validate anydata schema subtree with YANG Library
RFC 7950 Extension

With YANG-Push xpath or sub-tree a part of the YANG-Push data tree is subscribed. The subscribed YANG datastore content is published as anydata, event though the content has a valid schema.

RFC 7950 lacks specification how the data model of anydata content is exposed through YANG library defined in RFC 8525.

draft-aehassany-anydata-validation extends RFC 7950 by describing:

- How anydata can be validated with YANG Library.

RFC 7950

7.10. The "anydata" Statement

The "anydata" statement defines an interior node in the schema tree. It takes one argument, which is an identifier, followed by a block of substatements that holds detailed anydata information.

The "anydata" statement is used to represent an unknown set of nodes that can be modeled with YANG, except anyxml, but for which the data model is not known at module design time. It is possible, though not required, for the data model for anydata content to become known through protocol signaling or other means that are outside the scope of this document.
Address YANG Specification and Integration Gaps
Aiming for an automated data processing pipeline

YANG Specifications Gaps:

• YANG model for NETCONF Event Notifications
draft-ahuang-netconf-notif-yang

• Validating anydata in YANG Library context
draft-aelhassany-anydata-validation

YANG Integration Gaps:

• Support of Network Observation Timestamping in YANG Notifications
draft-tgraf-netconf-yang-push-observation-time

• Support of Hostname and Sequencing in YANG Notifications
draft-tgraf-netconf-notif-sequencing

• Support of Versioning in YANG Notifications Subscription
draft-ietf-netconf-yang-notifications-versioning

• Augmented-by Addition into the IETF-YANG-Library
draft-lincla-netconf-yang-library-augmentation

« Do you recognize that addressing those gaps are a prerequisite to enable an automated data processing chain? If yes, please consider to attend IETF 120 NETCONF working group session on Thursday 09:30 – 11:30 or go onto the mailing list and contribute to the discussion. »
Open Points from IETF 119
Addressed at IETF 120

Open Point 1: datastore-contents in push-update or the value in push-change-update uses anydata as data type which contents does not have a schema defined. draft-aelhassany-anydata-validation addresses that anydata modeled nodes can be validated with YANG Library RFC 8525.

Open Point 2: Definitions how NOTIFICATIONS are encoded in NETCONF are defined in Section 4.2.10 of RFC 7950. However, specifications for encoding in JSON and CBOR are missing RFC 7951. Confirm finding and propose how this needs to be addressed.

Open Point 3: Test with running code wherever with draft-ietf-netconf-yang-notifications-versioning and draft-lincla-netconf-yang-library-augmentation all datastore-subtree-filter or datastore-xpath-filter referenced YANG modules and their dependencies can be fully identified.
Milestones
IETF 115 – 120


**IETF 116**: YANG module with augmentations can be registered in Confluent Schema Registry with YANG extension. draft-tgraf-netconf-notif-sequencing, draft-tgraf-netconf-yang-push-observation-time and draft-ahuang-netconf-notif-yang introduced.

**IETF 118**: All relevant YANG modules for a subscribed xpath can be determined through the YANG Library RFC 8525 and retrieved through NETCONF <get-schema> rpc calls according to RFC 6022. Gap in YANG library addressed in draft-lincla-netconf-yang-library-augmentation.

**IETF 119**: draft-aelhassany-anydata-validation addresses that anydata modeled nodes can be validated with YANG Library RFC 8525. 6WIND VSR and Huawei VRP YANG-Push and open-source draft-lincla-netconf-yang-library-augmentation implementation validated at hackathon.

**IETF 120**: 6WIND VSR, Huawei VRP and Cisco IOS XR YANG-Push publisher and draft-aelhassany-anydata-validation implementation validated at hackathon. Running code proved that with draft-ietf-netconf-yang-notifications-versioning and draft-lincla-netconf-yang-library-augmentation all datastore-subtree-filter or datastore-xpath-filter referenced YANG modules and their dependencies can be fully identified.

**Sep 2025**: Submit Architecture for YANG-Push to Message Broker Integration to the IESG.
## YANG-Push Implementation Status

**IETF 120**

<table>
<thead>
<tr>
<th>RFC 8639 YANG-Push Subscription</th>
<th>6WIND VSR</th>
<th>Huawei VRP</th>
<th>Cisco IOS XR</th>
<th>Open-Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RFC 8641 YANG-Push Notification</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>draft-ietf-netconf-udp-notif</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>draft-ietf-netconf-distributed-notif</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft-ietf-netconf-yang-notifications-versioning</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft-tgraf-netconf-notif-sequencing</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft-tgraf-netconf-yang-push-observation-time</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RFC 7895 YANG Module Library</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RFC 8525 YANG Library</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft-lincla-netconf-yang-library-augmentation</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Red marked describes new capability at IETF 120
Industry Collaboration
On YANG Push to Apache Kafka integration

Swisscom
Network Operator

INA
University

6WIND
Network Vendor

Imply
Imply Druid

ClickHouse
OLAP DBMS

Huawei
Network Vendor

NTT
Network Operator

Bell Canada
Network Operator

Cisco
Network Vendor

Confluent
Apache Kafka
Yang Kafka
Current Status

• Push back from Mohamed Boucadair on -04 working group adoption call.
• -05 addresses the following points:
  • Document updates besides RFC 5277 now also RFC 8639, RFC 7951 and RFC 9254 since RFC 8639 applies the notification statement in YANG-Push and RFC 7951 and RFC 9254 misses the description how to encode the notification statement in JSON and CBOR.
  • Describes the relationship to RFC 5277, RFC 8639, RFC 7951 and RFC 9254 and excludes scoping for Restconf since Section 6 of RFC 8040 describes encoding in JSON.
  • Editorial changes such as examples are moved from the appendix to section 4.

Next Steps

➢ Requesting feedback from the netconf working group and YANG-Push implementers.
Current Status

- Addresses feedback at NMOP that notification changes should be discoverable.
  - Section 2.1 describes new netconf notification with hostname and sequence capability.
  - Section 2.2 describes new YANG-related system capabilities. Netconf notification with hostname and sequence capability is now discoverable through extended YANG-related system capabilities defined in RFC 9196.
- Minor editorial changes and implementation status section added.

Next Steps

- Requesting feedback from the netconf working group and YANG-Push implementers.
Extend YANG-Push Notifications with **Observation Timestamping**
draft-tgraf-netconf-yang-push-observation-time-02 - Status and Next Steps

**Current Status**

- Changed semantics:
  - One observation-time timestamp describing when the metric was observed eases end to end integration into streaming processor and time series database.
  - Point-in-time describes at which point in time the value of observation-time was observed.
    - For "periodical" subscription, the "current-accounting" describes the **point in time where the metrics were polled and observed**.
    - For "on-change" subscriptions, the value of point-in-time is **"state-changed", when the state change was observed in real-time**.
    - For "on-change" subscriptions with the "sync on start option", the value of point-in-time **for the initial state is "initial-state"**.
  - YANG-Push observation timestamping capability is now discoverable by extending YANG-related system capabilities defined in RFC 9196.
  - Minor editorial changes and implementation status section added.

**Next Steps**

- **Requesting feedback from the netconf working group and YANG-Push implementers.**
Support of **Versioning** in YANG Notifications Subscription
draft-ietf-netconf-yang-notifications-versioning-04 - Status and Next Steps

**Current Status**

- Resolved issue that within a "case" statement identifiers need to be unique.
- Thanks to Jérémie Leska from 6Wind for reporting this issue.

**Next Steps**

- Validate implementation at IETF 120 hackathon.
- Requesting feedback from the netconf working group and YANG-Push implementers.