

NMOP
Internet-Draft
Intended status: Informational
Expires: 2 September 2026

T. Graf
A. Elhassany
Swisscom
1 March 2026

An Architecture for YANG-Push to Message Broker Integration
draft-ietf-nmop-yang-message-broker-integration-11

Abstract

This document describes the motivation and architecture of a native YANG-Push notifications and YANG Schema integration into a Message Broker and YANG Schema Registry.

Discussion Venues

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

Discussion of this document takes place on the Operations and Management Area Working Group Working Group mailing list (nmop@ietf.org), which is archived at <https://mailarchive.ietf.org/arch/browse/nmop/>.

Source for this draft and an issue tracker can be found at <https://github.com/network-analytics/draft-ietf-nmop-yang-message-broker-integration/>.

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 2 September 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Origins of YANG-Push	3
1.2. Origins of Apache Kafka	4
1.3. Document Scope	4
2. Conventions and Definitions	4
2.1. Terminology	5
3. Motivation	6
3.1. Automatic Onboarding	7
3.2. Preserve Schema	7
3.3. Preserve Semantic Information	7
3.4. Standardize Data Processing Integration	7
4. Elements of the Architecture	7
4.1. YANG-Push Subscription	9
4.2. YANG-Push Publisher	11
4.3. YANG-Push Receiver	15
4.4. YANG Schema Registry	15
4.5. Stream Catalog	17
4.6. YANG Message Broker Producer	17
4.7. YANG Message Broker Consumer	17
4.8. YANG Data Consumer	18
4.9. YANG Data Storage	18
5. Implementation Status	18
5.1. YANG Schema Registry Extension	19
5.2. YANG-Push Receiver Parsing Library	19
5.3. YANG Library Augmented-by Addition	19
6. Security Considerations	20
7. Acknowledgements	20
8. References	20
8.1. Normative References	20
8.2. Informative References	22
Appendix A. Project Milestones	25
Authors' Addresses	27

1. Introduction

Nowadays network operators are using YANG [RFC7950] to model their configurations and obtain YANG modelled data from their networks. It is well understood that plain text are initially intended for humans and need effort to make it machine readable due to the lack of semantics. YANG modeled data is addressing most of these needs.

Increasingly more network operators organizing their data in a Data Mesh [Deh22] where a Message Broker such as Apache Kafka [Kaf11] or Apache Pulsar [Pull6] facilitates the exchange of messages among data processing components like a stream processor to filter, enrich, correlate or aggregate, or a time series database to store data.

Even though YANG is intended to ease the handling of data, this promise has not yet been fulfilled for Network Telemetry [RFC9232]. From subscribing on a YANG datastore, publishing a YANG modeled notifications message from the network and viewing the data in a time series database, manual labor, such as obtaining the YANG schema from the network and creating a transformation or ingestion specification into a time series database, is needed to make a Message Broker and its data processing components with YANG notifications interoperable. Since YANG modules can change over time, for example when a router is being upgraded to a newer software release, this process needs to be adjusted continuously, leading often to errors in the data chain if dependencies are not properly tracked and schema changes adjusted simultaneously.

1.1. Origins of YANG-Push

With [RFC3535] the IAB set the requirements for Network Management in 2003. From these requirements NETCONF [RFC6241], NETCONF Notifications [RFC5277] and RESTCONF [RFC8040] have been defined to configure through <edit-config> and retrieve operational data through <get> and NETCONF notifications through <notification> from a YANG datastore on a network node.

With YANG-Push, as defined in [RFC8639], [RFC8640] and [RFC8641], periodical and on-change subscriptions to the YANG datastore can be dynamically or statically configured. When notifications are dynamically configured, messages are published over the initially established NETCONF session, while when it is statically configured messages are published through HTTPS-based [I-D.ietf-netconf-https-notif] or UDP-based [I-D.ietf-netconf-udp-notif] transport. Section 3.7 of [RFC8641] describes push-update messages where the YANG subscribed data is being published, where Section 2.7 of [RFC8639] describes the subscription state change notifications where changes in the Subscription are being described.

1.2. Origins of Apache Kafka

Apache Kafka [Kaf11] is a Message Broker that supports producing and consuming messages from so called topics. Each topic has one or more partitions where messages are replicated or load balanced to scale out. With the introduction of Confluent Schema Registry [Con18] a topic can contain one or more subjects. A subject refers to a Schema defining the structure of the message. The Schema then is used to validate messages sent through topics and are identified by a Schema ID. The Schema ID is issued when the Schema is registered to the Confluent Schema Registry. Once the Schema ID is obtained, it can be prefixed to the message with a Confluent Schema Registry compatible serializer. Messages can then be validated against Schema at the producer or at the consumer from a topic to ensure Schema integrity of the message. The type of Schema evolution scheme can be defined per subject, whether non backward compatibility changes are allowed or not.

1.3. Document Scope

This document focuses on YANG-Push [RFC8641] as the messaging protocol between network nodes and Network Telemetry [RFC9232] data collection. It describes the main components and the aimed architecture for deploying such solution in a production network. Then, it illustrates the integration of the YANG 1.1 [RFC7950] as a schema modeling language into the Apache Kafka Message Broker and Confluent Schema Registry [Con18].

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.

2.1. Terminology

This document defines the following terms:

Message Broker: is an intermediary software component that translates messages from the formal messaging protocol of the sender to the formal messaging protocol of the receiver routed in topics [Kaf11]. Message brokers are elements in Data Mesh where software applications communicate by exchanging formally-defined messages.

Stream Catalog: provides a single point of access that allows users to centrally search semantics for information across a Message Broker.

Additionally it makes use of the terms defined in [RFC8639], [RFC7950], Apache Kafka [Kaf11], Confluent Schema Registry Documentation [ConDoc18] and Data Mesh [Deh22].

The following terms are used as defined in [RFC8639]:

- * Publisher
- * Receiver
- * Subscription
- * Subscription ID
- * Event stream filter
- * Notification message

The following terms are used as defined in Apache Kafka Message Broker [Kaf11]:

- * Producer
- * Consumer
- * Topic
- * Partition

The following terms are used as defined in Confluent Schema Registry Documentation [ConDoc18]:

- * Schema

- * Schema ID
- * Schema Registry
- * Subject

The following terms are used as defined in Data Mesh [Deh22]:

- * **Data Product:** A logical unit that contains all components to publish, process and store domain data for analytical or data-intensive use cases.
- * **Service Level Objective:** Is a target value or range of values for a service level that is measured by a service level indicator.
- * **Service Level Indicator:** Is a measure of a service level provided by a service provider to a customer.

This document defines the following terms:

- * **YANG Schema Registry:** Stores YANG schemas and has an API for registering and retrieving YANG schemas for YANG Message Broker Producers and Consumers. See Section 4.4.
- * **YANG Message Broker Producer:** Serializes YANG telemetry messages from previously retrieved YANG notifications From YANG-Push Receiver and registers YANG schemas at YANG Schema Registry for the serialized telemetry messages. See Section 4.6.
- * **YANG Message Broker Consumer:** De-Serializes YANG telemetry messages from Message Broker and retrieve YANG schemas for the serialized telemetry messages from YANG Schema Registry. See Section 4.7.
- * **YANG Data Consumer:** Transforms and ingests previously obtained YANG notifications from Message Broker into YANG Data Storage by leveraging YANG schemas from YANG Schema Registry. See Section 4.8.
- * **YANG Data Storage:** Stores and indexes YANG data retrieved through YANG notifications. See Section 4.9.

3. Motivation

There are four main objectives for native YANG-Push notifications and YANG Schema integration into a Message Broker.

3.1. Automatic Onboarding

Automate the Data Mesh onboarding of newly subscribed YANG metrics.

3.2. Preserve Schema

The preservation of the YANG schema, that includes the YANG data types as defined in [RFC6991] and the nested structure of the YANG module, the data taxonomy, throughout the data processing chain ensures that metrics can be processed and visualized as they were originally intended. Not only for users but also for automated closed loop operation actions.

3.3. Preserve Semantic Information

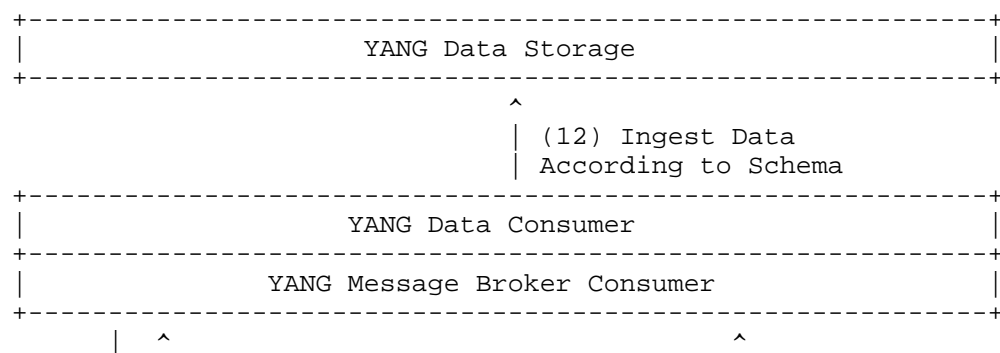
[RFC7950] defines in Section 7.21.3 and 7.21.4 the description and reference statement. This information is intended for the user, describing in a human-readable fashion the meaning of a definition. In Data Mesh, this information can be imported from the YANG Schema Registry into a Stream Catalog where subjects within Message Broker are identifiable and searchable. An example of a Stream Catalog is Apache Atlas [Atl15]. It can also be applied for time series data visualization in a similar fashion.

3.4. Standardize Data Processing Integration

Since the YANG Schema is preserved for operational metrics in the Message Broker, a standardization for integration between network data collection and stream processors or time series databases is implied.

4. Elements of the Architecture

The architecture consists of 10 elements. Figure 1 gives an overview on the workflow.



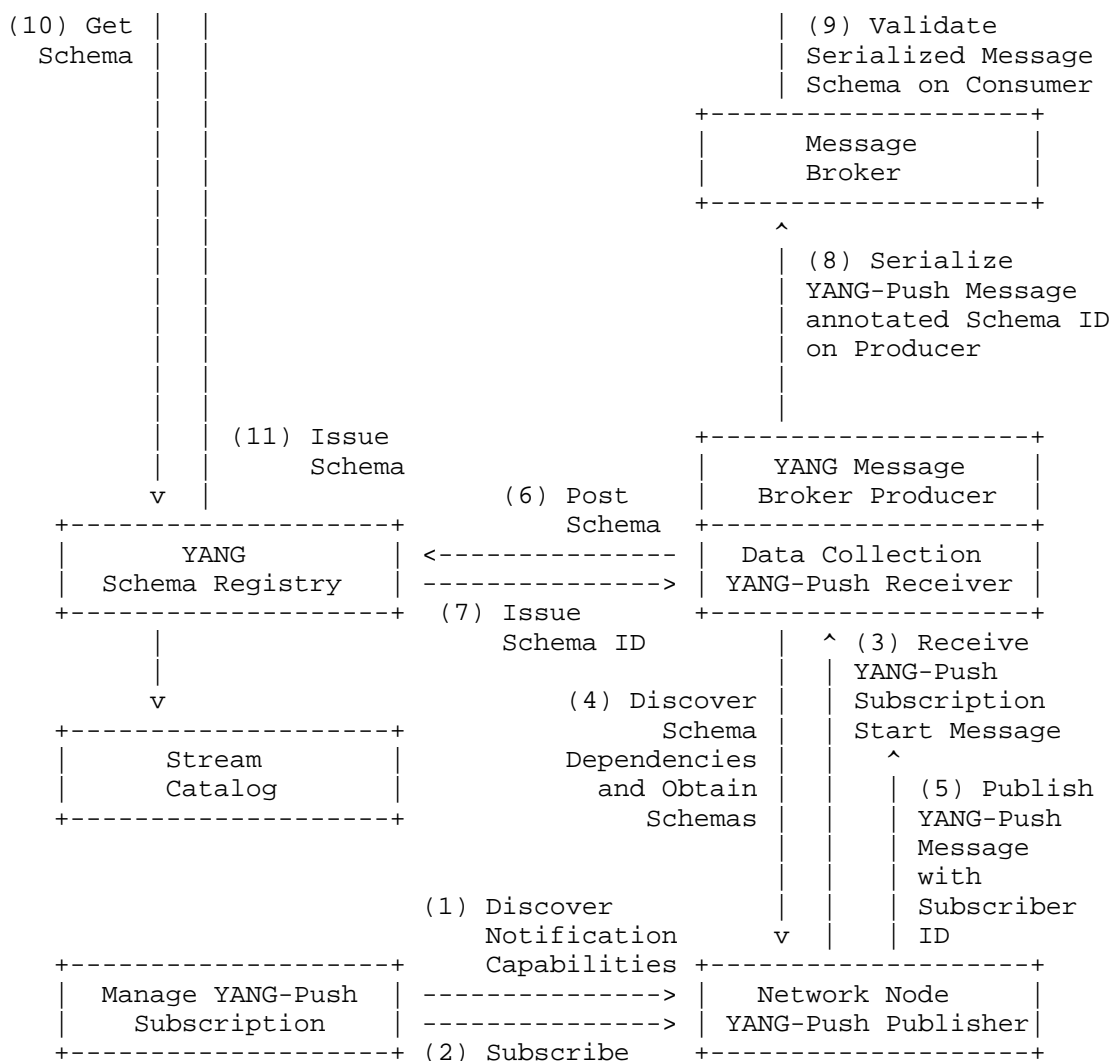


Figure 1: End to End Workflow

The workflow diagram (Figure 1) describes the steps from establishing the YANG-Push subscription to Data Storage ingestion.

4.1. YANG-Push Subscription

With step number (1) in the workflow diagram, the YANG-Push notification transport capabilities are being discovered according to [I-D.ietf-netconf-yp-transport-capabilities], notification capabilities according to [I-D.ietf-netconf-notif-envelope] and notification subscription capabilities according to Section 3 of [RFC9196].

With step (2) a YANG-Push subscription according to Section 2.4 and 2.5 of [RFC8639] is dynamically or statically configured.

With step (3) subscription start notifications are sent according to Section 2.7 of [RFC8639] from the YANG-Push Publisher to the Receiver to inform which event stream filter has been applied to which Subscription ID.

When the YANG-Push Subscription is managed dynamically, the YANG data is being received on the same NETCONF session where the Subscription is being maintained. With configured Subscription the YANG data is sent to the YANG-Push Receiver through a separate transport session.

[I-D.ietf-netconf-yang-notifications-versioning] adds the capability to subscribe to a specific YANG module revision or backward compatible YANG module in step (2) and adds the module name, revision, revision-label and yang-library-content-id information into the subscription state change notifications in step (3).

Figure 2 provides an example how to create a YANG-Push configured Subscription with NETCONF in XML [W3C.REC-xml-20081126] with UDP-based [I-D.ietf-netconf-udp-notif] transport

===== NOTE: '\ ' line wrapping per RFC 8792) =====

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <subscriptions xmlns="urn:ietf:params:xml:ns:yang:ietf\
        -subscribed-notifications">
        <subscription>
          <id>6666</id>
          <datastore xmlns="urn:ietf:params:xml:ns:yang:ietf\
            -yang-push"
            xmlns:ds="urn:ietf:params:xml:ns:yang:ietf\
```

```
-datastores">ds:operational</datastore>
<datastore-xpath-filter xmlns="urn:ietf:params:xml:ns\
:yang:ietf-yang-push"
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-inter\
  faces">/if:interfaces</datastore-xpath-filter>
<revision xmlns="urn:ietf:params:xml:ns:yang:ietf-yang\
-push-revision">2018-02-20</revision>
<transport xmlns:unt="urn:ietf:params:xml:ns:yang:ietf\
-udp-notif-transport">unt:udp-notif</transport>
<encoding>encode-json</encoding>
<receivers>
  <receiver>
    <name>subscription-specific-receiver-def</name>
    <receiver-instance-ref xmlns="urn:ietf:params:xml\
:ns:yang:ietf-subscribed-notif-receivers">\
    global-udp-notif-receiver-def</receiver-instance-ref>
  </receiver>
</receivers>
<periodic xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
  <period>6000</period>
</periodic>
</subscription>
<receiver-instances xmlns="urn:ietf:params:xml:ns:yang:ietf\
-subscribed-notif-receivers">
  <receiver-instance>
    <name>global-udp-notif-receiver-def</name>
    <udp-notif-receiver xmlns="urn:ietf:params:xml:ns:yang\
:ietf-udp-notif-transport">
      <address>203.0.113.1</address>
      <port>12345</port>
      <enable-segmentation>false</enable-segmentation>
      <max-segment-size/>
    </udp-notif-receiver>
  </receiver-instance>
</receiver-instances>
</subscriptions>
</config>
</edit-config>
</rpc>
```

Figure 2: NETCONF Example to establish configured Subscription

Figure 3 provides an example of a JSON encoded, [RFC7951], subscription-started state change notification message over HTTPS-based [I-D.ietf-netconf-https-notif] or UDP-based [I-D.ietf-netconf-udp-notif] transport with [I-D.ietf-netconf-notif-envelope] and [I-D.ietf-netconf-yang-notifications-versioning] as extensions for the same Subscription.

```
{
  "ietf-yp-notification:envelope": {
    "event-time": "2023-03-25T08:30:11.22Z",
    "hostname": "example-router",
    "sequence-number": 1,
    "notification-contents": {
      "ietf-subscribed-notification:subscription-started": {
        "id": 6666,
        "ietf-yang-push:datastore": "ietf-datastores:operational",
        "ietf-yang-push:datastore-xpath-filter": "/if:interfaces",
        "ietf-yang-push-revision:revision": "2014-05-08",
        "ietf-yang-push-revision:module-name": "ietf-interfaces",
        "ietf-yang-push-revision:revision-label": "",
        "ietf-yang-push-revision:yang-library-content-id": "1",
        "ietf-distributed-notif:message-publisher-ids": [1,2],
        "transport": "ietf-udp-notif-transport:udp-notif",
        "encoding": "encode-json",
        "ietf-yang-push:periodic": {
          "ietf-yang-push:period": 100
        }
      }
    }
  }
}
```

Figure 3: JSON YANG-Push Example for a subscription-started notification message

4.2. YANG-Push Publisher

With step number (4) in the workflow diagram, a YANG-Push push-update or push-change-update message, depending on wherever periodical or on-change Subscription has been established, is sent from the YANG-Push Publisher to the receiver according to Section 3.7 of [RFC8639].

[I-D.ietf-netconf-notif-envelope] defines the YANG-Push notification header in YANG with event-time, hostname and sequence-number to enable a YANG data consumer to use the hostname to recognize which network node the message was published from and with sequence-number message reordering or loss.

[I-D.ietf-netconf-distributed-notif] adds a message-publisher-id for network nodes with distributed architecture where multiple YANG-Push publishing processes are publishing messages. The message-publisher-id enables a YANG data consumer to recognize from which YANG-Push Publisher process the message was published from.

Section 3.5.1 of [I-D.ietf-netconf-notif-envelope] adds observation timestamp and point-in-time in the YANG-Push push-update or push-change-update message. The observation timestamp contains the timestamp and point-in-time when the metrics were observed.

Figure 4 provides an example of a JSON encoded, [RFC7951], push-update notification message over HTTPS-based [I-D.ietf-netconf-https-notif] or UDP-based [I-D.ietf-netconf-udp-notif] transport and [I-D.ietf-netconf-notif-envelope] with hostname, sequence-number, observation timestamp and point-in-time as extensions.

===== NOTE: '\ ' line wrapping per RFC 8792) =====

```
{
  "ietf-yp-notification:envelope": {
    "event-time": "2023-03-25T08:30:11.22Z",
    "hostname": "example-router",
    "sequence-number": 1,
    "notification-contents": {
      "ietf-yang-push:push-update": {
        "id": 6666,
        "ietf-yp-observation:timestamp": \
        "2023-03-25T08:30:11.22Z",
        "ietf-yp-observation:point-in-time": \
        "current-accounting",
        "datastore-contents": {
          "ietf-interfaces:interfaces": [
            {
              "interface": {
                "name": "eth0",
                "type": "iana-if-type:ethernetCsmacd",
                "oper-status": "up",
                "mtu": 1500
              }
            }
          ]
        }
      }
    }
  }
}
```

Figure 4: JSON YANG-Push Example for a push-update notification message

Figure 5 provides an example of a JSON encoded, [RFC7951], push-change-update notification message over HTTPS-based [I-D.ietf-netconf-https-notif] or UDP-based [I-D.ietf-netconf-udp-notif] transport and [I-D.ietf-netconf-notif-envelope] with hostname, sequence-number, observation timestamp and point-in-time as extensions.

```

===== NOTE: '\ ' line wrapping per RFC 8792) =====

{
  "ietf-yp-notification:envelope": {
    "event-time": "2023-03-25T08:30:11.22Z",
    "hostname": "example-router",
    "sequence-number": 1,
    "notification-contents": {
      "ietf-yang-push:push-change-update": {
        "id": 2222,
        "ietf-yp-observation:timestamp": \
        "2023-03-25T08:30:11.22Z",
        "ietf-yp-observation:point-in-time": \
        "state-changed",
        "datastore-contents": {
          "yang-patch": {
            "patch-id": "patch_54",
            "comment": "Changing encoding to JSON and increasing \
            the period to 10 minutes",
            "edit": [
              {
                "edit-id": "id_change_1",
                "operation": "merge",
                "target": "/ietf-subscribed-notifications\:subs\
                criptions/subscription[id=2222]",
                "value": {
                  "ietf-subscribed-notifications:encoding": \
                  "ietf-subscribed-notifications:encode-json",
                  "ietf-yang-push:periodic": {
                    "period": 60000
                  }
                }
              }
            ]
          }
        }
      }
    }
  }
}

```

Figure 5: JSON YANG-Push Example for a push-change-update notification message

4.3. YANG-Push Receiver

For all the YANG modules and revisions of each hostname and Subscription ID in the subscription state change notification received in step number (3) in the workflow diagram, all the YANG module dependencies need to be determined through the YANG Library [RFC8525] and the import statements in the YANG modules. The dependencies include: imports, augments, deviations and features being used for each YANG module. Through NETCONF <get-schema> RPC calls, [RFC6022], all YANG modules need to be retrieved as described in step (4) in the workflow diagram.

YANG Library [RFC8525] already supports the discovery of submodules, deviations and features but misses augments. [I-D.ietf-netconf-yang-library-augmentedby] extends the YANG Library so that also the list of YANG modules which have direct augmentations are listed.

A change in xpath or sub-tree filter or module name in the subscription state change notifications indicates that the subscribed path has changed. A change in revision indicates that the version of the YANG module of the subscribed path has changed, where revision-label indicates wherever the revision was backward compatible or not. YANG Library content-id changes indicates that there were potential semantic changes in the augmented or imported YANG modules of the subscribed xpath or sub-tree. If no change to previous subscription state is observed, step 4 can be omitted.

Figure 9 in Section 4.1 and YANG module in Section 5 of [RFC8641] defines the payload of YANG-push notifications where "datastore-contents" or the "value" of a "push-change-update") is "anydata". [I-D.netana-nmop-yang-anydata-validation] extends Section 7.10 of [RFC7950] by describing how validate Schema integrity of the anydata content.

4.4. YANG Schema Registry

The schema registry SHOULD support YANG as the format for defining schema and has the ability to store metadata along with the schema. For each schema registered into the schema registry, a schema ID is returned. That schema ID can be used when interacting with the Message Broker to indicate the schema to use with the message.

Confluent Schema Registry is pluggable. Currently Supports AVRO, JSON Schema and Protobuf. The YANG support is being developed at [Yak24] as part of this architecture. Enable to register, obtain and compare [YSR24] YANG Schemas. One YANG Schema with all its augmentations is being registered per YANG-Push subscription ID along

with metadata describing which YANG features were enabled for the registered YANG schema. For each YANG Schema a locally significant Schema ID is being issued as described in step (7) in the workflow diagram.

```
curl -X POST -H "Content Type: application/vnd.schemaregistry.v1+json"
-d @ietf-interfaces@2018-02-20.json
http://localhost:8081/subjects/ietf-interfaces/
```

```
{
  "schemaType": "YANG",
  "references": [
    {
      "name": "ietf-interfaces",
      "subject": "ietf-interfaces",
      "version": "2018-02-20",
    }
  ],
  "metadata": {
    "tags": {
      "features": ["arbitrary-names", "if-mib"]
    }
  },
  "schema": "... yang schema text"
}
```

Figure 6: Register ietf-interfaces.yang into YANG Schema Registry

```
curl http://localhost:8081/subjects/ | jq
```

Figure 7: List all subjects YANG Schema Registry

```
curl http://localhost:8081/subjects/ietf-interfaces/versions
```

Figure 8: List versions of a given subject in YANG Schema Registry

```
curl http://localhost:8081/subjects/ietf-interfaces/versions/1
```



```
{
  "schemaType": "YANG",
  "subject": "ietf-interfaces",
  "version": 2018-02-20,
  "id": 1,
  "references": [
    {
      "name": "ietf-yang-types",
      "subject": "ietf-yang-types",
      "version": "2013-07-15",
    }
  ],
  "metadata": {
    "tags": {
      "features": ["arbitrary-names", "if-mib"]
    }
  },
  "schema": "... yang schema text"
}
```

Figure 9: Retrieve schema of a specific subject and version in YANG Schema Registry

4.5. Stream Catalog

The YANG schemas in the YANG schema registry are made accessible to the Stream Catalog where users have an overview and search on available semantics in the Message Broker.

4.6. YANG Message Broker Producer

The previously issued Schema ID is prefixed to the metadata augmented YANG push push-update message previously described in Section 4.3 before serialized to a Message Broker topic in step (8) of the workflow diagram.

Section 2 of [I-D.ietf-nmop-message-broker-telemetry-message] defines the envelope schema of the message facilitating the YANG-Push and different types of provenance metadata.

4.7. YANG Message Broker Consumer

From the Message Broker topic the message is being consumed and the prefixed Schema ID is being used in step (10) of the workflow diagram to retrieve the YANG Schema to validate the Schema integrity of the message.

In order to validate Schema integrity, the event stream filter of the Subscription ID in the subscription started notification Section 4.3.1 of [RFC8641] or telemetry message Section 3 of [I-D.ietf-nmop-message-broker-telemetry-message] is needed to apply [I-D.netana-nmop-yang-anydata-validation] on the YANG schema tree.

4.8. YANG Data Consumer

The data storage ingestion specifications are being derived with the in Section 4.7 already retrieved Schema ID and YANG-Push push-update messages can be now ingested and indexed into the data storage table according to their schema in step (12).

By tracking the YANG-Push notification sequence-number for a given hostname as defined in Section 3.4 of [I-D.ietf-netconf-notif-envelope] loss can be recognized accross the YANG data processing chain. By taking the event-time in the YANG-Push notification header and the time the YANG-Push notification messages was consumed into account, the serialization delay between YANG-Push publisher and YANG Data Consumer can be measured.

The loss rate and delay for a given hostname can be used as a Service Level Indicator for the YANG data product in the Data Mesh [Deh22].

4.9. YANG Data Storage

The YANG data is ingested in step (12) according to the previously defined ingestion specification and indexed with the observation timestamp as defined in Section 3.5.1 of [I-D.ietf-netconf-notif-envelope]. A network operator is now able to query the previously subscribed YANG data.

5. Implementation Status

Note to the RFC-Editor: Please remove this section and its corresponding references before publishing.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

5.1. YANG Schema Registry Extension

Ahmed Elhassany developed a YANG Schema Extension in Confluent Schema Registry.

The source code can be obtained here: [YSR24], the progress report here: [YSRPR24], and was validated at the IETF 117 and 123 hackathon.

5.2. YANG-Push Receiver Parsing Library

Zhuoyao Lin developed as part of her internship a library to parse YANG-Push subscription notifications, identify YANG module dependencises with YANG Library [RFC8525] and obtain with NETCONF <get-schema> rpc calls [RFC6022] all YANG modules from YANG-Push publisher.

The source code can be obtained here: [LYP23] and was validated at the IETF 117 hackathon.

5.3. YANG Library Augmented-by Addition

Zhuoyao Lin implemented [I-D.ietf-netconf-yang-library-augmentedby] in order to discover augmented-by YANG modules in YANG Library [RFC8525].

The source code can be obtained here: [YLA24] and was validated at the IETF 119 hackathon.

6. Security Considerations

TBD

7. Acknowledgements

The authors would like to thank Yannick Buchs, Benoit Claise, Mohamed Boucadair, Dhruv Dhody, Qin Wu, Andy Bierman, Feng Chong, Holger Keller, Kristian Larsson, Ignacio Dominguez Martinez-Casanueva, Paul Aitken and Reshad Rahman for their review and valuable comments. Alex Huang Feng, Jean Quilbeuf and Huoyao Lin for review and contributing code and providing examples and inputs to the open points.

8. References

8.1. Normative References

[I-D.ietf-netconf-notif-envelope]

Feng, A. H., Francois, P., Graf, T., and B. Claise,
"Extensible YANG Model for YANG-Push Notifications", Work
in Progress, Internet-Draft, draft-ietf-netconf-notif-
envelope-04, 6 February 2026,
<<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-notif-envelope-04>>.

[I-D.ietf-netconf-yang-library-augmentedby]

Lin, Z., Claise, B., and I. D. Martinez-Casanueva,
"Augmented-by Addition to the YANG Library", Work in
Progress, Internet-Draft, draft-ietf-netconf-yang-library-
augmentedby-17, 20 February 2026,
<<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-yang-library-augmentedby-17>>.

[I-D.ietf-netconf-yang-notifications-versioning]

Graf, T., Claise, B., and A. H. Feng, "Support of
Versioning in YANG Notifications Subscription", Work in
Progress, Internet-Draft, draft-ietf-netconf-yang-
notifications-versioning-11, 12 February 2026,
<<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-yang-notifications-versioning-11>>.

[I-D.ietf-netconf-yp-transport-capabilities]

Wu, Q., Ma, Q., Feng, A. H., and T. Graf, "YANG Notification Transport Capabilities", Work in Progress, Internet-Draft, draft-ietf-netconf-yp-transport-capabilities-05, 16 June 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-yp-transport-capabilities-05>>.

[I-D.ietf-nmop-message-broker-telemetry-message]

Elhassany, A., Graf, T., and P. Lucente, "Extensible YANG Model for Network Telemetry Messages", Work in Progress, Internet-Draft, draft-ietf-nmop-message-broker-telemetry-message-04, 18 January 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-nmop-message-broker-telemetry-message-04>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC6022] Scott, M. and M. Bjorklund, "YANG Module for NETCONF Monitoring", RFC 6022, DOI 10.17487/RFC6022, October 2010, <<https://www.rfc-editor.org/info/rfc6022>>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/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/info/rfc8174>>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <<https://www.rfc-editor.org/info/rfc8639>>.

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

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <<https://www.rfc-editor.org/info/rfc8792>>.

- [RFC9196] Lengyel, B., Clemm, A., and B. Claise, "YANG Modules Describing Capabilities for Systems and Datastore Update Notifications", RFC 9196, DOI 10.17487/RFC9196, February 2022, <<https://www.rfc-editor.org/info/rfc9196>>.
- [RFC9254] Veillette, M., Ed., Petrov, I., Ed., Pelov, A., Bormann, C., and M. Richardson, "Encoding of Data Modeled with YANG in the Concise Binary Object Representation (CBOR)", RFC 9254, DOI 10.17487/RFC9254, July 2022, <<https://www.rfc-editor.org/info/rfc9254>>.

8.2. Informative References

- [Atl15] Hortonworks, "Apache Atlas", Apache Software Foundation, May 2015, <<https://atlas.apache.org/>>.
- [Con18] Yokota, R., "Confluent Schema Registry", Confluent Community and Apache Software Foundation, December 2018, <<https://github.com/confluentinc/schema-registry/>>.
- [ConDoc18] Yokota, R., "Confluent Schema Registry Documentation", Confluent Community and Apache Software Foundation, December 2018, <<https://docs.confluent.io/platform/current/schema-registry/>>.
- [Deh22] Dehghani, Z., "Data Mesh", O'Reilly Media, ISBN 9781492092391, March 2022, <<https://www.oreilly.com/library/view/data-mesh/9781492092384/>>.
- [I-D.ietf-netconf-distributed-notif]
Zhou, T., Zheng, G., Voit, E., Graf, T., and P. Francois, "Subscription to Notifications in a Distributed Architecture", Work in Progress, Internet-Draft, draft-ietf-netconf-distributed-notif-18, 1 March 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-distributed-notif-18>>.
- [I-D.ietf-netconf-https-notif]
Jethanandani, M. and K. Watsen, "An HTTPS-based Transport for YANG Notifications", Work in Progress, Internet-Draft, draft-ietf-netconf-https-notif-15, 1 February 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-https-notif-15>>.

- [I-D.ietf-netconf-udp-notif]
Feng, A. H., Francois, P., Zhou, T., Graf, T., and P. Lucente, "UDP-based Transport for Configured Subscriptions", Work in Progress, Internet-Draft, draft-ietf-netconf-udp-notif-25, 28 January 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-udp-notif-25>>.
- [I-D.netana-nmop-yang-anydata-validation]
Elhassany, A. and T. Graf, "Validating anydata in YANG Library context", Work in Progress, Internet-Draft, draft-netana-nmop-yang-anydata-validation-01, 20 October 2025, <<https://datatracker.ietf.org/doc/html/draft-netana-nmop-yang-anydata-validation-01>>.
- [Kaf11] Narkhede, N., "Apache Kafka", Apache Software Foundation, January 2011, <<https://kafka.apache.org/>>.
- [LYP23] Lin, Z., "libyangpush", Apache Software Foundation, September 2023, <<https://github.com/network-analytics/libyangpush/>>.
- [Netgauze_Github]
"Netgauze open source Network Telemetry Data Collection", <<https://github.com/NetGauze/NetGauze/>>.
- [Pul16] Guo, S. and M. Merli, "Apache Pulsar", Apache Software Foundation, January 2016, <<https://pulsar.apache.org/>>.
- [RFC3535] Schoenwaelder, J., "Overview of the 2002 IAB Network Management Workshop", RFC 3535, DOI 10.17487/RFC3535, May 2003, <<https://www.rfc-editor.org/info/rfc3535>>.
- [RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <<https://www.rfc-editor.org/info/rfc5277>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.

- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <<https://www.rfc-editor.org/info/rfc8525>>.
- [RFC8526] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "NETCONF Extensions to Support the Network Management Datastore Architecture", RFC 8526, DOI 10.17487/RFC8526, March 2019, <<https://www.rfc-editor.org/info/rfc8526>>.
- [RFC8528] Bjorklund, M. and L. Lhotka, "YANG Schema Mount", RFC 8528, DOI 10.17487/RFC8528, March 2019, <<https://www.rfc-editor.org/info/rfc8528>>.
- [RFC8640] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Dynamic Subscription to YANG Events and Datastores over NETCONF", RFC 8640, DOI 10.17487/RFC8640, September 2019, <<https://www.rfc-editor.org/info/rfc8640>>.
- [RFC9144] Clemm, A., Qu, Y., Tantsura, J., and A. Bierman, "Comparison of Network Management Datastore Architecture (NMDA) Datastores", RFC 9144, DOI 10.17487/RFC9144, December 2021, <<https://www.rfc-editor.org/info/rfc9144>>.
- [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/info/rfc9232>>.

[W3C.REC-xml-20081126]

Bray, T., Paoli, J., Sperberg-McQueen, M., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <<https://www.w3.org/TR/2008/REC-xml-20081126>>.

[Yak24] Feng, F., "Yangkit", Apache Software Foundation, February 2024, <<https://github.com/network-analytics/yang-kafka-integration>>.

[YLA24] Lin, Z., "libyangpush", Apache Software Foundation, March 2024, <<https://github.com/Zephyre777/draft-lincla-netconf-yang-library-augmentation/>>.

[YSR24] Elhassany, A. and A. Huang-Feng, "YANG Schema Registry Extension", Apache Software Foundation, February 2024, <<https://github.com/network-analytics/yang-kafka-integration>>.

[YSRPR24] Elhassany, A., "YANG Schema Registry Extension Progress Report", July 2025, <<https://github.com/network-analytics/ietf-network-analytics-document-status/blob/main/123/Hackathon/ietf-123-hackathon-validate-yang-push-publisher.pdf>>.

Appendix A. Project Milestones

IETF 115:

- * Official Project Kickoff.
- * [I-D.ietf-netconf-yang-notifications-versioning] extends schema reference in subscription state change notification.

IETF 116:

- * YANG module with augmentations can be registered in Confluent Schema Registry with YANG extension [Yak24].
- * draft-tgraf-netconf-notif-sequencing extends NETCONF notification header with sysName, messagePublisherId and sequencedraft-tgraf-netconf-notif-sequencing Number.
- * draft-tgraf-netconf-yang-push-observation-time extends YANG-Push push-update or push-change-update message with observation-time or state-changed-observation-time.

- * draft-ahuang-netconf-notif-yang defines the NETCONF notification header specified in [RFC5277] in YANG.

IETF 118:

- * All relevant YANG modules for a subscribed xpath can be determined through the YANG Library [RFC8525] and retrieved through NETCONF <get-schema> rpc calls according to [RFC6022]. Gap in YANG library addressed in [I-D.ietf-netconf-yang-library-augmentedby].

IETF 119:

- * [I-D.netana-nmop-yang-anydata-validation] addresses that anydata modeled nodes can be validated with YANG Library RFC 8525. 6WIND VSR and Huawei VRP YANG-Push publisher and open-source [I-D.ietf-netconf-yang-library-augmentedby] implementation validated at hackathon.

IETF 120:

- * 6WIND VSR, Huawei VRP and Cisco IOS XR YANG-Push Publisher (Section 4.2) implementations validated at hackathon.
- * draft-tgraf-netconf-yang-push-observation-time merges both timestamps for periodical and on-change YANG-Push subscriptions into one observation timestamp and adding a point-in-time declaration to describe when the observation was observed.

IETF 121:

- * 6WIND VSR, Huawei VRP and Cisco IOS XR YANG-Push Publisher (Section 4.2) implementations supporting, [I-D.ietf-netconf-udp-notif], [I-D.ietf-netconf-distributed-notif], draft-tgraf-netconf-yang-push-observation-time and [I-D.ietf-netconf-yang-notifications-versioning] validated at hackathon.
- * [I-D.netana-nmop-yang-anydata-validation] with YANG schema registry and message broker YANG data producer and consumer implementation validated at hackathon.
- * Validated at hackathon based on [I-D.ietf-netconf-yang-notifications-versioning] that with [I-D.ietf-netconf-yang-library-augmentedby] all datastore-subtree-filter or datastore-xpath-filter referenced YANG modules and their dependencies can be fully identified.

IETF 122,123 and 124:

- * 6WIND VSR, Huawei VRP and Cisco IOS XR YANG-Push Publisher (Section 4.2) implementations supporting, [I-D.ietf-netconf-udp-notif], [I-D.ietf-netconf-distributed-notif], [I-D.ietf-netconf-yp-transport-capabilities], [I-D.ietf-netconf-notif-envelope], [I-D.ietf-netconf-yang-notifications-versioning] and [I-D.ietf-netconf-yang-library-augmentedby] validated at hackathon.
- * Validate from Manage YANG-Push Subscription to YANG Data Consumer end to end all components of this architecture.

IETF 125:

- * 6WIND VSR, Huawei VRP and Cisco IOS XR YANG-Push Publisher (Section 4.2) implementations supporting, [I-D.ietf-netconf-udp-notif], [I-D.ietf-netconf-distributed-notif], [I-D.ietf-netconf-yp-transport-capabilities], [I-D.ietf-netconf-notif-envelope], [I-D.ietf-netconf-yang-notifications-versioning] and [I-D.ietf-netconf-yang-library-augmentedby].
- * Arrcus Arcos YANG-Push Publisher (Section 4.2) implementation supporting, [I-D.ietf-netconf-udp-notif], [I-D.ietf-netconf-yp-transport-capabilities] and [I-D.ietf-netconf-notif-envelope].
- * NetGauze Network Telemetry data collection [Netgauze_Github] supporting YANG-Push Receiver (Section 4.3), YANG Message Broker Producer (Section 4.6) and YANG Message Broker Consumer (Section 4.7).
- * Ciena Blue Planet UAA supporting YANG Message Broker Consumer (Section 4.7).
- * Confluent YANG Schema Registry plugin [Yak24] supporting YANG Schema Registry (Section 4.4).
- * Validate from Manage YANG-Push Subscription to YANG Data Consumer end to end all components of this architecture.

Authors' Addresses

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

Ahmed Elhassany
Swisscom
Binzring 17
CH- Zuerich 8045
Switzerland
Email: ahmed.elhassany@swisscom.com