

# An Architecture for YANG-Push to Message Broker **Integration**

draft-ietf-nmop-yang-message-broker-integration-01

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

thomas.graf@swisscom.com  
ahmed.elhassany@swisscom.com

16. June 2024

# Handling Operational YANG Modelled Data

## State of the Union

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.

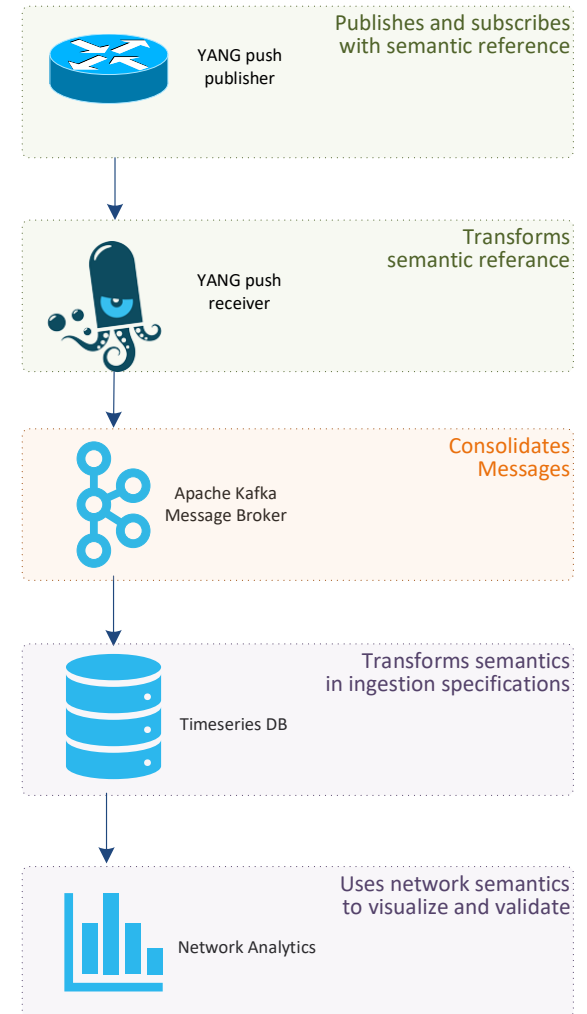
Today, subscribing to a YANG datastore, publishing a YANG modeled notifications message from the network and viewing the data in a time series database, **manual labor is needed to perform data transformation** to make a message broker and its data processing components with YANG notifications interoperable.

« 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

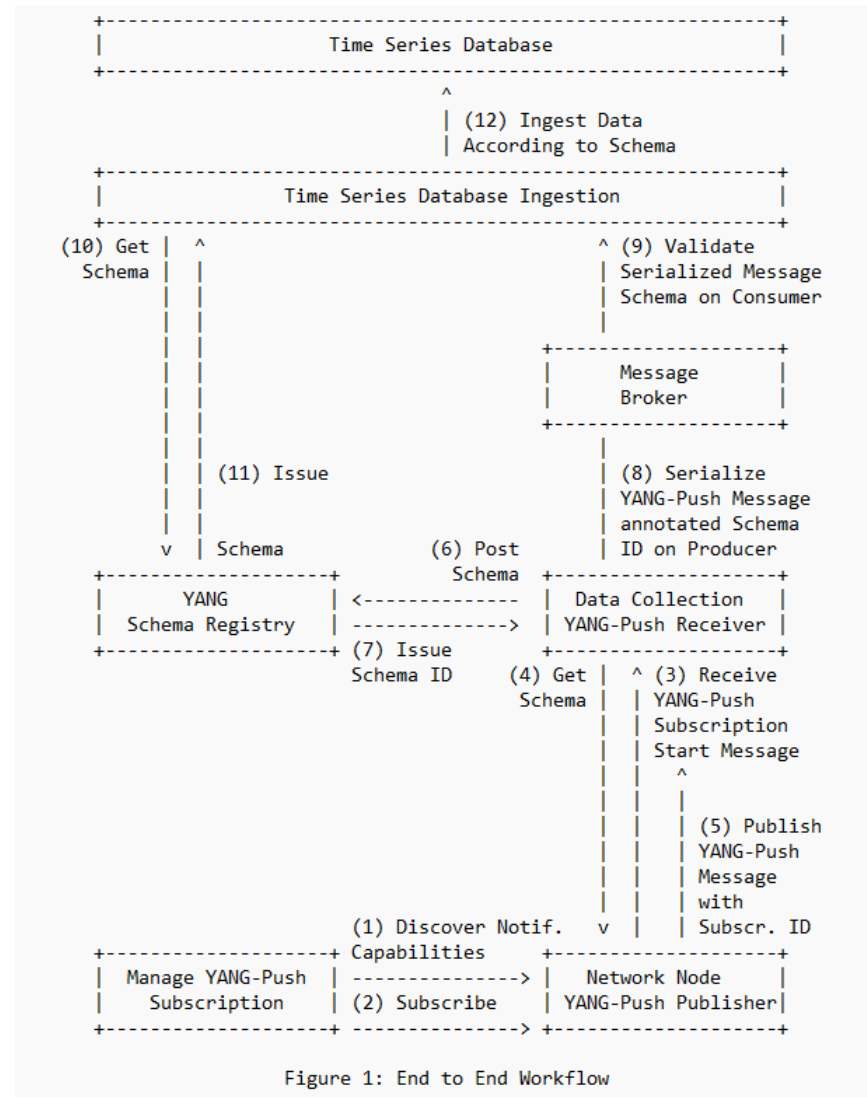


Figure 1: End to End Workflow

- **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.
- **Time Series Database Ingestion** 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

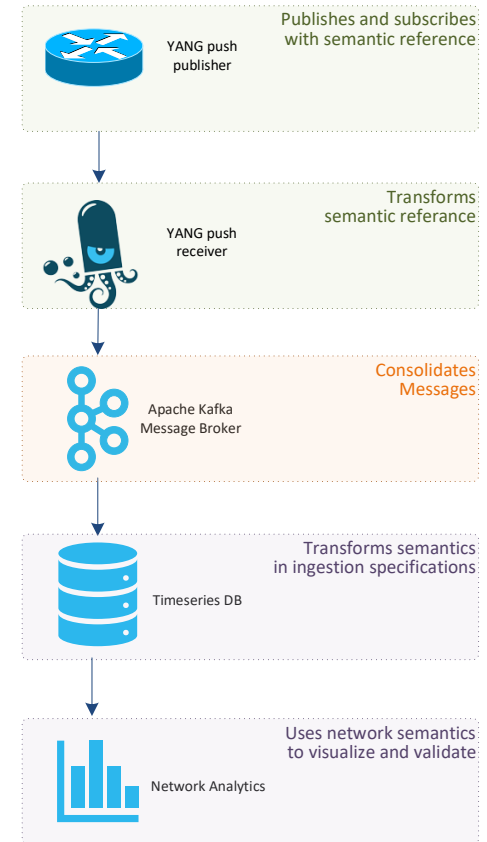
- Document adopted at NMOP.
- Feedback from Dhruv, Andy, Feng and Qin addressed in -01. Thank you very much!

### Changes in -01

- 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
- Used the boiler plate from RFC 7942 and moved section before Security Considerations
- Applied RFC 8792 to handle long lines
- 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.

### Next Steps

- **Looking forward for review and comments.**



# Address YANG Specification and Integration Gaps

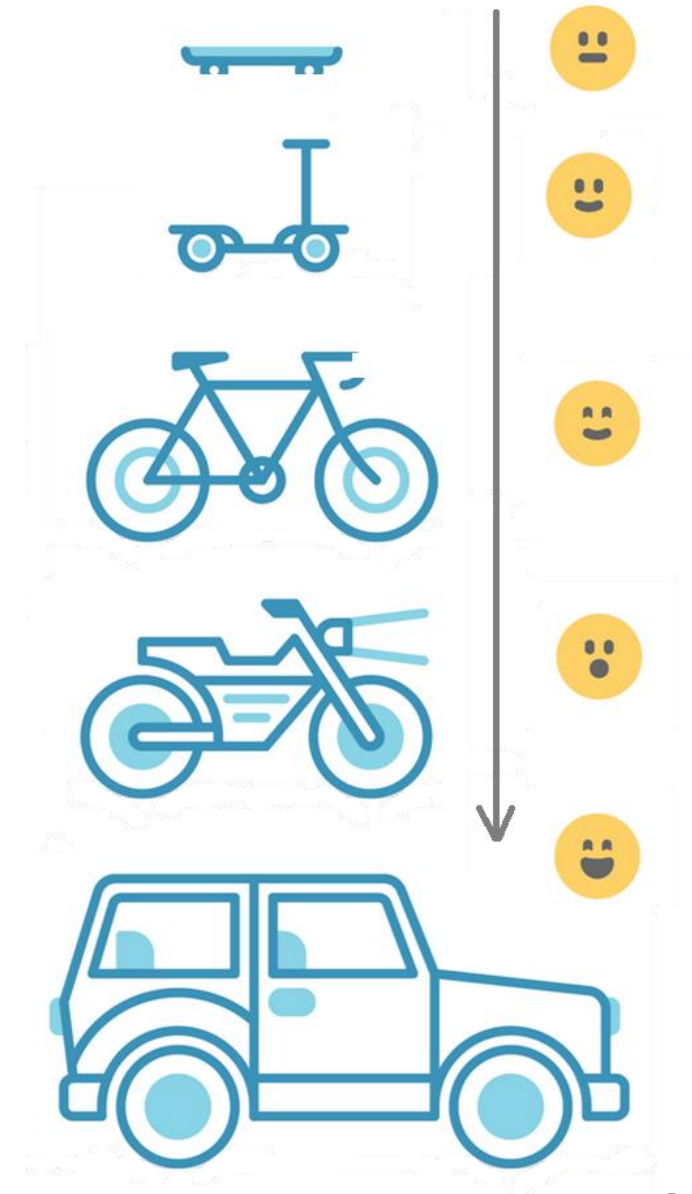
Aiming for an automated data processing pipeline

## 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-linclanetconf-yang-library-augmentation](#)



# YANG model for NETCONF Event Notifications

Entire YANG-Push messages is modeled in YANG

```
module: ietf-notification

structure notification:
  +-- eventTime      yang:date-and-time

{
  "ietf-notification:notification": {
    "eventTime": "2023-02-10T08:00:11.22Z",
    "ietf-yang-push:push-update": {
      "id": 1011,
      "datastore-contents": {
        "ietf-interfaces:interfaces": [
          {
            "interface": {
              "name": "eth0",
              "oper-status": "up"
            }
          }
        ]
      }
    }
  }
}
```

- **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](#).

```
notification push-update {
  description
    "This notification contains a push update that in turn contains data
    subscribed to via a subscription. In the case of a periodic subscription,
    this notification is sent for periodic updates. It can also be used for
    synchronization updates of an on-change subscription. This notification
    shall only be sent to receivers of a subscription. It does not constitute
    a general-purpose notification that would be subscribable as part of the
    NETCONF event stream by any receiver.";
  leaf id {
    type sn:subscription-id;
    description
      "This references the subscription that drove the
      notification to be sent.";
  }
}
```

Notification groupings defined in ietf-yang-push.yang of [RFC 8641](#)

# **YANG model** for NETCONF Event Notifications

draft-ahuang-netconf-notif-yang-05 - Status and Next Steps

## **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.**



# Extend Netconf Notifications with **Hostname and Sequence Number**

For push-update and push-change-update

```
module: ietf-notification

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

{
  "ietf-notification:notification": {
    "eventTime": "2023-03-25T08:30:11.22Z",
    "ietf-notification-sequencing:sysName": "example-router",
    "ietf-notification-sequencing:sequenceNumber": 1,
    "ietf-yang-push:push-update": {
      "id": 6666,
      "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
            }
          }
        ]
      }
    }
  }
}
```

- 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 Netconf Notifications with **Hostname and Sequence Number**

draft-tgraf-netconf-notif-sequencing-05 - Status and Next Steps

## 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

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?
      inotifseq: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-yang-push:push-update": {
      "id": 6666,
      "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.

# Extend YANG-Push Notifications with **Observation Timestamping**

draft-tgraf-netconf-yang-push-observation-time-01 - 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 either **"state-changed", when the state change was observed in real-time** with or without sync on start option. Or it is **"current-state" when it was observed after the YANG-Push subscription was established and the time since the state changed to current state is unknown**.
- YANG-Push observation timestamping capability is now discoverable by extending YANG-related system capabilities defined in RFC 9196.
- Minor editorial changes and operational considerations 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

For subscription state change notification messages

```
module: ietf-yang-push-revision

augment /sn:establish-subscription/sn:input:
  +---w module-version-config* [module-name]
    +---w module-name          yang:yang-identifier
    +---w revision?            rev:revision-date-or-label
    +---w revision-label?      ysver:version

augment /sn:subscription-started:
  +--ro module-version* [module-name]
    +--ro module-name          yang:yang-identifier
    +--ro revision              rev:revision-date-or-label
    +--ro revision-label?      ysver:version

{
  "ietf-notification:notification": {
    "eventTime": "2023-03-25T08:30:11.22Z",
    "ietf-notification-sequencing:sysName": "example-router",
    "ietf-notification-sequencing:sequenceNumber": 1,
    "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-distributed-notif:message-observation-domain-id": [1,2],
      "transport": "ietf-udp-notif-transport:udp-notif",
      "encoding": "encode-json",
      "ietf-yang-push:periodic": {
        "ietf-yang-push:period": 100
      }
    }
  }
}
```

- **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.

# 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.**

# 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.**
- [draft-lincla-netconf-yang-library-augmentation](#) extends the YANG library defined in [RFC 8525](#):
  - By adding augmented-by YANG module relation.

# Validate anydata schema subtree with YANG Library

## RFC 7950 Extension

### notifications:

```
+---n push-update
|   +--ro id?                sn:subscription-id
|   +--ro datastore-contents? <anydata>

{
  "ietf-yang-push:push-update": {
    "id": 89,
    "datastore-contents": {
      "ietf-interfaces:interfaces": {
        "interface": [
          {
            "name": "eth0",
            "oper-status": "down"
          }
        ]
      }
    }
  }
}
```

- 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, even 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-aelhassany-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.



# 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 115:** Official Project Kickoff. Introduced [draft-ietf-netconf-yang-notifications-versioning](#).

**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-lincl-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-lincl-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 proofed that with [draft-ietf-netconf-yang-notifications-versioning](#) and [draft-lincl-netconf-yang-library-augmentation](#) all datastore-subtree-filter or datastore-xpath-filter referenced YANG modules and their dependencies can be fully identified.

# YANG-Push Implementation Status

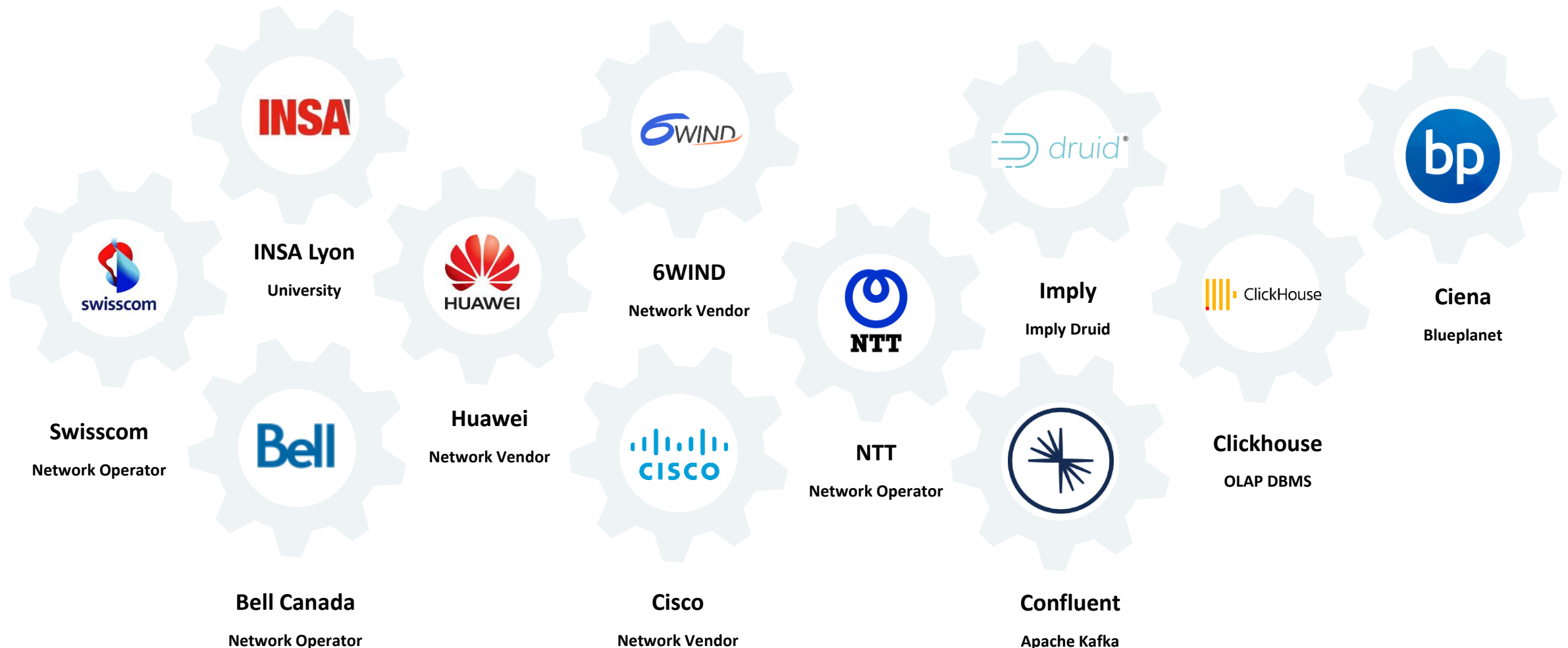
IETF 120

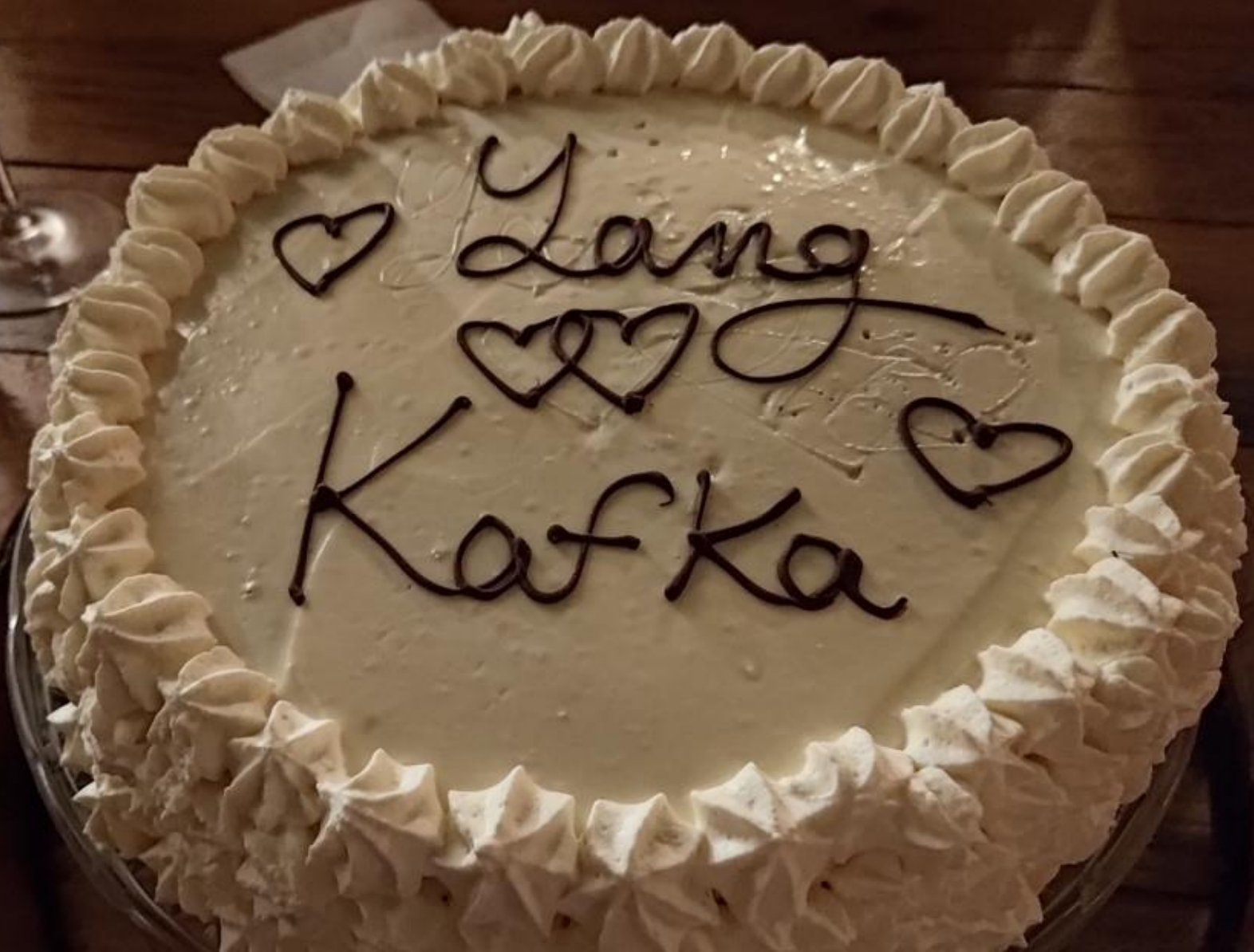
	6WIND VSR	Huawei VRP	Cisco IOS XR
RFC 8641 YANG-Push	x	x	x
draft-ietf-netconf-udp-notif	x	x	
draft-ietf-netconf-distributed-notif	x	x	
draft-ietf-netconf-yang-notifications-versioning	x	x	
draft-tgraf-netconf-notif-sequencing	x		
draft-tgraf-netconf-yang-push-observation-time	x		
RFC 7895 YANG Module Library		x	
RFC 8525 YANG Library	x		x
draft-lincla-netconf-yang-library-augmentation			



# Industry Colaboration

On YANG Push to Apache Kafka integration





♥ Yang ♥♥♥  
Kafka ♥