An Architecture for YANG-Push to Message Broker Integration

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

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

zhenghaomian@huawei.com
david.zhujian@huawei.com
thomas.graf@swisscom.com

8. March 2025

IETF YANG-Push

A 22 years journey

2002

IAB Workshop

Defines operators' requirements in RFC 3535 to lifecycle CLI and SNMP. YANG, Netconf and Restconf development started.

2010

YANG 1.0

Specified in RFC 6020. 1.1 in RFC 7950.

2015

IETF YANG-Push Specification Started

Development of RFC 8639 and RFC 8641 started at IETF NETCONF.

2017

gNMI

gNMI was presented to IETF NETCONF and implementations started at major network vendors.

2019

IETF YANG-Push Specification Finished

Development of RFC 8639 and RFC 8641 concluded at IETF NETCONF without any major network vendor implementations.

2022

Data Mesh Integration

Vendor-specific implementations and IETF YANG-Push are hard to manage. New requirements emerged for integrating with the message broker and an automated data processing chain. New specifications are proposed to resolve these challenges.

2024

IETF YANG-Push Major Implementations Started

Questions arise. Proposing a simplified IETF YANG-Push and an Agile Incremental Driven Development.

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

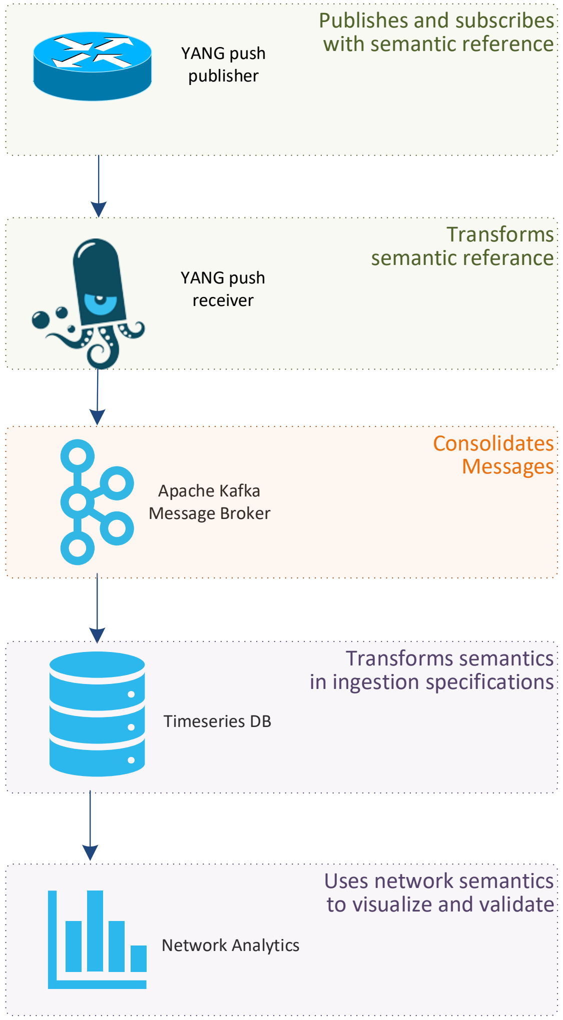
Today, subscribing to a YANG datastore, publishing a YANG modeled notifications message from the network and viewing the data in a time series database, **manual labor is needed to perform data transformation** to make a message broker and its data processing components with YANG notifications interoperable.

« Even though YANG is intented to ease data management, 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.



YANG push publisher

Publishes and subscribes with semantic reference

YANG push receiver

Transforms semantic referance

Apache Kafka Message Broker

Consolidates Messages

Timeseries DB

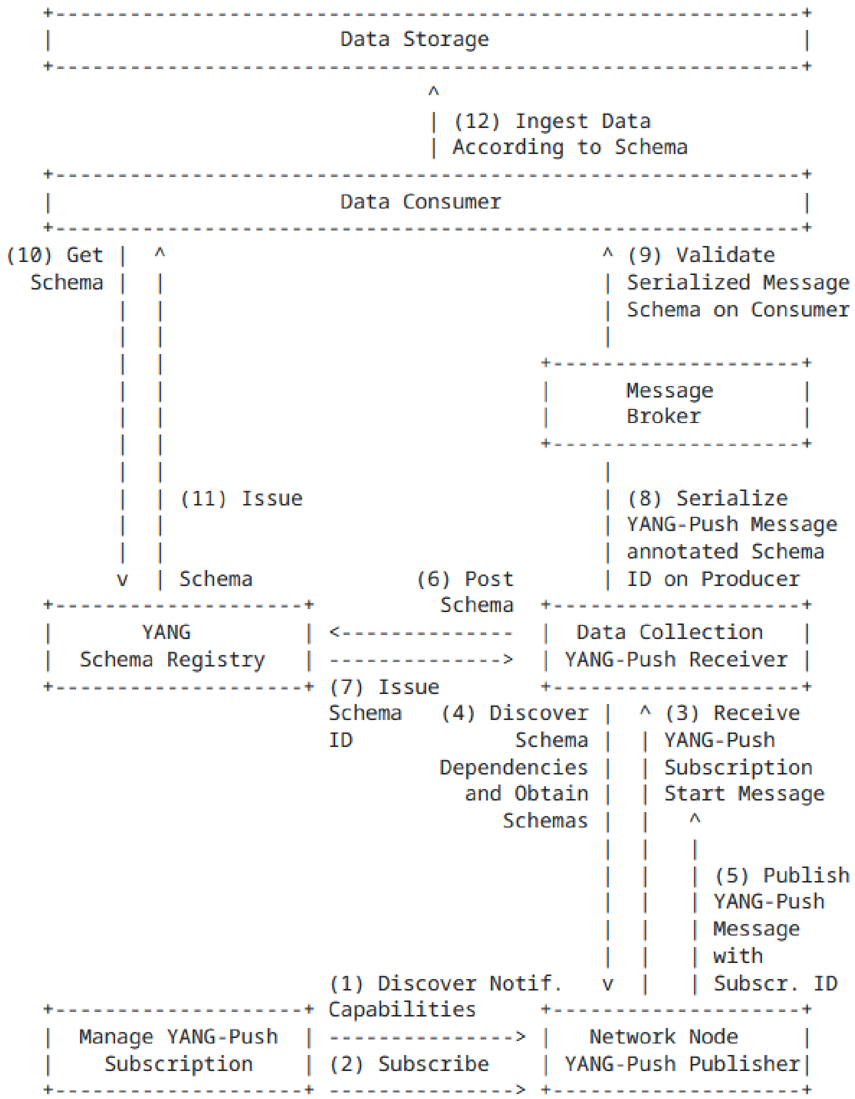
Transforms semantics in ingestion specifications

Network Analytics

Uses network semantics to visualize and validate

Elements of the Architecture

And Document Relationships



```
   +------------------------------------------------------------+
   |                       Data Storage                         |
   +------------------------------------------------------------+
                                  ^
                                  | (12) Ingest Data
                                  | According to Schema
   +------------------------------------------------------------+
   |                       Data Consumer                        |
   +------------------------------------------------------------+
(10) Get |  ^                                   ^ (9) Validate
  Schema |  |                                   | Serialized Message
         |  |                                   | Schema on Consumer
         |  |                                   |
         |  |                              +--------------------+
         |  |                              |      Message       |
         |  |                              |      Broker        |
         |  |                              +--------------------+
         |  |                                   |
         |  | (11) Issue                        | (8) Serialize
         |  |                                   | YANG-Push Message
         |  |                                   | annotated Schema
         v  | Schema            (6) Post        | ID on Producer
   +--------------------+          Schema  +--------------------+
   |       YANG         | <--------------  |  Data Collection   |
   |  Schema Registry   | -------------->  | YANG-Push Receiver |
   +--------------------+ (7) Issue        +--------------------+
                          Schema   (4) Discover |  ^ (3) Receive
                          ID             Schema |  | YANG-Push
                                   Dependencies |  | Subscription
                                     and Obtain |  | Start Message
                                        Schemas |  |   ^
                                                |  |   |
                                                |  |   | (5) Publish
                                                |  |   | YANG-Push
                                                |  |   | Message
                                                |  |   | with
                          (1) Discover Notif.   v  |   | Subscr. ID
   +--------------------+ Capabilities     +--------------------+
   | Manage YANG-Push   | ---------------> |    Network Node    |
   |   Subscription     | (2) Subscribe    | YANG-Push Publisher|
   +--------------------+ ---------------> +--------------------+
```

- **4.1. YANG-Push Subscription**
  Thanks to RFC 9196 based YANG modules describing capabilities for systems and notifications, transport (draft-netana-netconf-yp-transport-capabilities), notification extensions (draft-netana-netconf-notif-envelope) and supported xpaths can be discovered.
- **4.2. YANG-Push Publisher**
  For configured subscriptions, udp-notif, distributed-notif and https-notif are the RFC 8639, RFC 8641 based YANG-Push transport protocols. draft-netana-netconf-notif-envelope provides the notification header with hostname, sequence and observation timestamp extensions.
- **4.3. YANG-Push Receiver**
  Discovers subscription state changes by detecting changes in xpath/sub-tree, module name, revision, revision-label and YANG library content-id defined in draft-ietf-netconf-yang-notifications-versioning and discovers YANG schema tree through YANG library with ietf-netconf-yang-library-augmentedby extension.
- **4.6. YANG Message Broker Consumer**
  Validates YANG notifications against schema. Anydata described in Section 7.10 of RFC 7950, validation described in draft-aelhassany-anydata-validation.
- **4.7. YANG Data Consumer**
  Thanks to hostname, sequence-number and event-time from draft-netana-netconf-notif-envelope, loss and serialization delay can be measured.
- **4.8. YANG Data Storage**
  Metrics can be indexed in timeseries database thanks to observation timestamp defined in draft-netana-netconf-notif-envelope.

Addressing YANG Specification and Integration Gaps

9 documents at NMOP, NETCONF and NETMOD

YANG-Push Transport Gaps:

Passed working group last call. Moving to IESG

- UDP-based Transport for Configured Subscriptions
  draft-ietf-netconf-udp-notif
- Subscription to Distributed Notifications
  draft-ietf-netconf-distributed-notif

YANG-Push Specifications Gaps:

Passed working group adoption call.

- Extensible YANG model for YANG-Push Notifications
  draft-netana-netconf-notif-envelope
- YANG Notification Transport Capabilities
  draft-netana-netconf-yp-transport-capabilities
- Validating anydata in YANG Library context
  draft-aelhassany-anydata-validation

YANG-Push Integration Gaps and Arch:

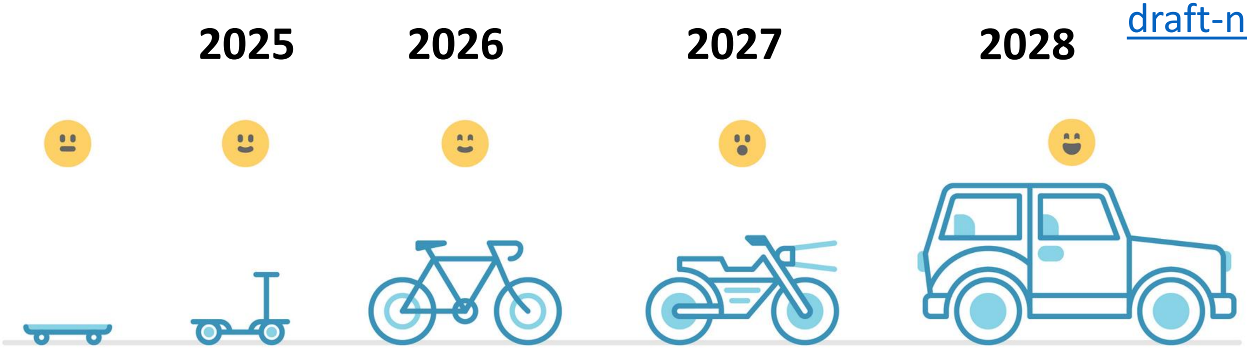
- Support of Versioning in YANG Notifications Subscription
  draft-ietf-netconf-yang-notifications-versioning
- Augmented-by Addition into the IETF-YANG-Library
  draft-ietf-netconf-yang-library-augmentation

YANG-Push Simplification:

- YANG-Push Operational Data Observability Enhancements
  draft-wilton-netconf-yp-observability

YANG-Push Message Broker:

- An Architecture for YANG-Push to Message Broker Integration
  draft-ietf-nmop-yang-message-broker-integration
- Extensible YANG Model for Network Telemetry Notifications
  draft-netana-nmop-message-broker-telemetry-message



IETF YANG-Push Implementations and Next Steps

Who we are and what we like…

The group consisting of: **34 colleagues** from Bell Canada, Deutsche Telekom, NTT International, Swisscom, Huawei, Cisco, 6Wind, Ciena Blueplanet, Juniper, Nokia, and INSA Lyon.

Development on first major vendor implementations started at IETF 118. Throughout IETF 119 - 121, vendor implementation and network operator testing scope and interest from other vendors and operators is steadily growing.

This is where we noticed WT-508 at BBF and its potential synergies.

What we like about IETF YANG-Push

- Interoperable
- Unified with Netconf and Restconf
- Transport independent
- Push based

What is unique to IETF YANG-Push

- Capabilities are discoverable which enables subscription automation
- Subscription state change notifications enables the discovery of schema changes
- YANG Library enables discovery of schema dependencies
- Standardized notification metadata

IETF YANG-Push Implementations

Incremental development…

MVP 1 - Works

4 major vendor implementations finalized.

- draft-ietf-netconf-udp-notif transport (with segmentation option) and notifications encoded in RFC 7951 IETF-JSON.
- RFC 8641 periodic subscription with anchor-time configurable in ietf-subscribed-notifications.
- YANG notifications as defined in draft-netana-netconf-notif-envelope, YANG module name, version and yang-library-content-id as in draft-ietf-netconf-yang-notifications-versioning and notification capabilities (RFC 9196) discoverable as defined in their documents.
- YANG-Library as defined in RFC 8525 and draft-ietf-netconf-yang-library-augmentedby.

MVP 2 – Scales and Secures

2 major vendor implementations started.

- Distributed notifications as defined in draft-ietf-netconf-distributed-notif.
- YANG notifications encoded in CBOR (named identifiers) as defined in RFC 9254.
- YANG notifications DTLS (1.2 MUST RFC 6347, 1.3 SHOULD RFC 9147) encrypted as defined in draft-ietf-netconf-udp-notif.

MVP 3 - Optimizes

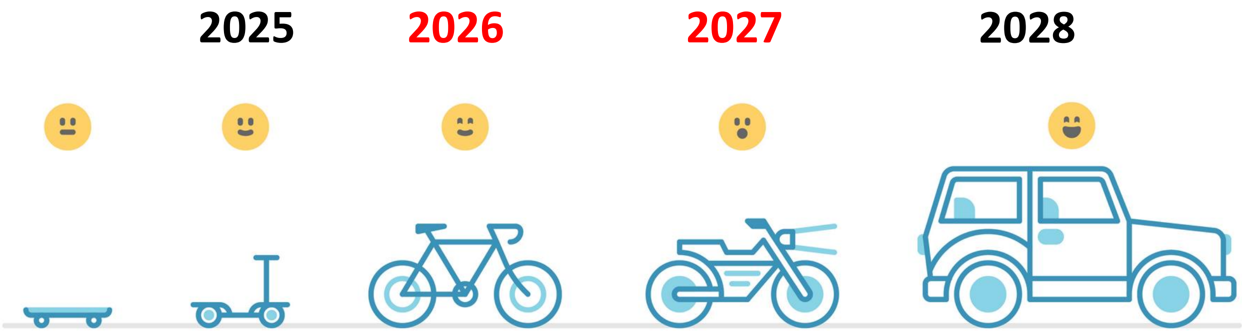
- RFC 8641 on-change subscription configurable in ietf-subscribed-notifications.
- RFC 8641 on-change and periodical subscription in RFC 9196 defined capabilities discoverable.
- draft-ietf-netconf-udp-notif transport, encoding and encryption capabilities (RFC 9196) discoverable with draft-netana-netconf-yp-transport-capabilities.

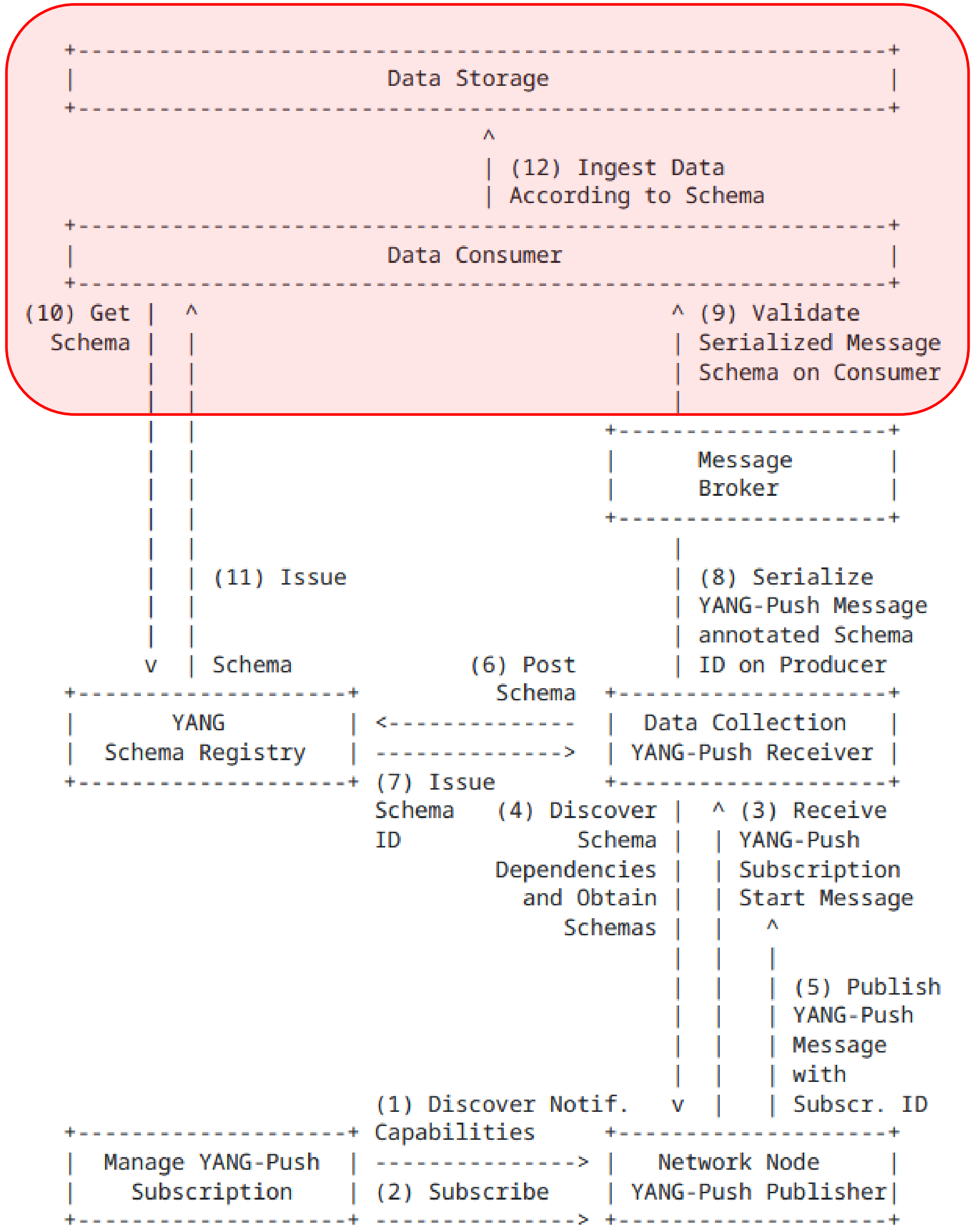
Part 2

Our collaboration goes into the next step in 2026

This allows Swisscom

to transition from SNMP to YANG





```
   +------------------------------------------------------------+
   |                       Data Storage                         |
   +------------------------------------------------------------+
                                  ^
                                  | (12) Ingest Data
                                  | According to Schema
   +------------------------------------------------------------+
   |                       Data Consumer                        |
   +------------------------------------------------------------+
(10) Get |  ^                                   ^ (9) Validate
  Schema |  |                                   | Serialized Message
         |  |                                   | Schema on Consumer
         |  |                                   |
         |  |                              +--------------------+
         |  |                              |      Message       |
         |  |                              |      Broker        |
         |  |                              +--------------------+
         |  |                                   |
         |  | (11) Issue                        | (8) Serialize
         |  |                                   | YANG-Push Message
         |  |                                   | annotated Schema
         v  | Schema             (6) Post       | ID on Producer
   +--------------------+          Schema  +--------------------+
   |       YANG         | <--------------  |  Data Collection   |
   |  Schema Registry   | -------------->  | YANG-Push Receiver |
   +--------------------+ (7) Issue        +--------------------+
                          Schema   (4) Discover |  ^ (3) Receive
                          ID             Schema |  | YANG-Push
                                   Dependencies |  | Subscription
                                     and Obtain |  | Start Message
                                        Schemas |  |   ^
                                                |  |   |
                                                |  |   | (5) Publish
                                                |  |   | YANG-Push
                                                |  |   | Message
                                                |  |   | with
                          (1) Discover Notif.   v  |   | Subscr. ID
   +--------------------+ Capabilities     +--------------------+
   |  Manage YANG-Push  | ---------------> |   Network Node     |
   |    Subscription    | (2) Subscribe    | YANG-Push Publisher|
   +--------------------+ ---------------> +--------------------+
```

IETF YANG Model Implementations

Incremental development…

"Starting 2026, Swisscom onboards new network platforms with YANG-Push only. Initiating the SNMP to YANG lifecycle and the standardization of YANG models."

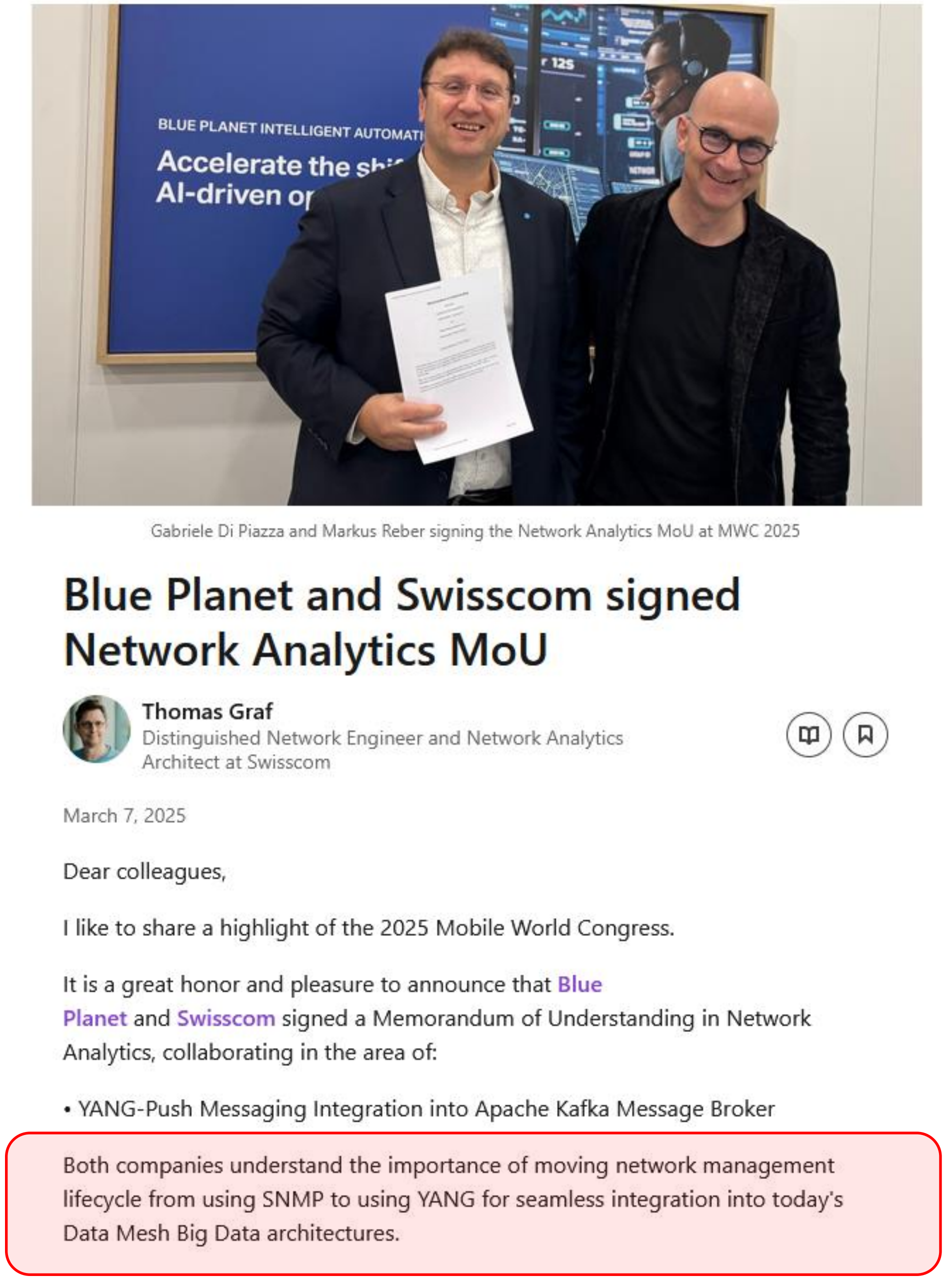
MVP 2 – Physical & Essential

Already supported by OLT MA5800.

- ietf-interfaces.yang (inventory, state and statistic)
- ietf-hardware.yang (inventory, state and statistic)
- ietf-alarms.yang (state)
- ieee802-dot1ab-lldp.yang (inventory and state)

MVP 3 – Logical Relationships

- ieee802-dot1ax.yang (inventory, state and statistic)
- ietf-bfd-ip-sh.yang (state)
- ietf-bfd-ip-mh.yang (state)
- ietf-bfd-lag.yang (state)
- ietf-isis.yang (state)



Gabriele Di Piazza and Markus Reber signing the Network Analytics MoU at MWC 2025

Blue Planet and Swisscom signed Network Analytics MoU

Thomas Graf
Distinguished Network Engineer and Network Analytics Architect at Swisscom

March 7, 2025

Dear colleagues,

I like to share a highlight of the 2025 Mobile World Congress.

It is a great honor and pleasure to announce that Blue Planet and Swisscom signed a Memorandum of Understanding in Network Analytics, collaborating in the area of:

- YANG-Push Messaging Integration into Apache Kafka Message Broker

Both companies understand the importance of moving network management lifecycle from using SNMP to using YANG for seamless integration into today's Data Mesh Big Data architectures.

RFC 8343 – ietf-interfaces.yang

YANG Data Model for Interface Management

```
module: ietf-interfaces
  +--rw interfaces
     +--rw interface* [name]
        +--rw name                        string
        +--rw description?                string
        +--rw type                        identityref
        +--rw enabled?                    boolean
        +--rw link-up-down-trap-enable?   enumeration {if-mib}?
        +--ro admin-status                enumeration {if-mib}?
        +--ro oper-status                 enumeration
        +--ro last-change?                yang:date-and-time
        +--ro if-index                    int32 {if-mib}?
        +--ro phys-address?               yang:phys-address
        +--ro higher-layer-if*            interface-ref
        +--ro lower-layer-if*             interface-ref
        +--ro speed?                      yang:gauge64
        +--ro statistics
           +--ro discontinuity-time    yang:date-and-time
           +--ro in-octets?            yang:counter64
           +--ro in-unicast-pkts?      yang:counter64
           +--ro in-broadcast-pkts?    yang:counter64
           +--ro in-multicast-pkts?    yang:counter64
           +--ro in-discards?          yang:counter32
           +--ro in-errors?            yang:counter32
           +--ro in-unknown-protos?    yang:counter32
           +--ro out-octets?           yang:counter64
           +--ro out-unicast-pkts?     yang:counter64
           +--ro out-broadcast-pkts?   yang:counter64
           +--ro out-multicast-pkts?   yang:counter64
           +--ro out-discards?         yang:counter32
           +--ro out-errors?           yang:counter32
```

With IETF YANG-Push we like to subscribe

with on-change sync-on-start

- For inventory and alert use cases the state of the interface.

and periodically

- For performance measurement use cases the interface statistics.

RFC 8348 – ietf-hardware.yang

YANG Data Model for Hardware Management

```
module: ietf-hardware
  +--rw hardware
     +--ro last-change?   yang:date-and-time
     +--rw component* [name]
        +--rw name              string
        +--rw class             identityref
        +--ro physical-index?   int32 {entity-mib}?
        +--ro description?      string
        +--rw parent?           -> ../../component/name
        +--rw parent-rel-pos?   int32
        +--ro contains-child*   -> ../../component/name
        +--ro hardware-rev?     string
        +--ro firmware-rev?     string
        +--ro software-rev?     string
        +--ro serial-num?       string
        +--ro mfg-name?         string
        +--ro model-name?       string
        +--rw alias?            string
        +--rw asset-id?         string
        +--ro is-fru?           boolean
        +--ro mfg-date?         yang:date-and-time
        +--rw uri*              inet:uri
        +--ro uuid?             yang:uuid
        +--rw state {hardware-state}?
        |  +--ro state-last-changed?   yang:date-and-time
        |  +--rw admin-state?          admin-state
        |  +--ro oper-state?           oper-state
        |  +--ro usage-state?          usage-state
        |  +--ro alarm-state?          alarm-state
        |  +--ro standby-state?        standby-state
        +--ro sensor-data {hardware-sensor}?
           +--ro value?               sensor-value
           +--ro value-type?          sensor-value-type
           +--ro value-scale?         sensor-value-scale
           +--ro value-precision?     sensor-value-precision
           +--ro oper-status?         sensor-status
           +--ro units-display?       string
           +--ro value-timestamp?     yang:date-and-time
           +--ro value-update-rate?   uint32
```

With IETF YANG-Push we like to subscribe

with on-change sync-on-start

- For inventory use cases the state of the hardware.
- For alert use cases the hardware state notifications.

and periodically

- For performance measurement use cases the hardware sensor-data.

```
notifications:
  +---n hardware-state-change
  +---n hardware-state-oper-enabled {hardware-state}?
  |  +--ro name?          -> /hardware/component/name
  |  +--ro admin-state?   -> /hardware/component/state/admin-state
  |  +--ro alarm-state?   -> /hardware/component/state/alarm-state
  +---n hardware-state-oper-disabled {hardware-state}?
     +--ro name?          -> /hardware/component/name
     +--ro admin-state?   -> /hardware/component/state/admin-state
     +--ro alarm-state?   -> /hardware/component/state/alarm-state
```

RFC 8632 – ietf-alarms.yang

YANG Data Model for Alarm Management

```
 notifications:
    +---n alarm-notification
    |  +--ro resource                 resource
    |  +--ro alarm-type-id            alarm-type-id
    |  +--ro alarm-type-qualifier?    alarm-type-qualifier
    |  +--ro alt-resource*            resource
    |  +--ro related-alarm* [resource alarm-type-id alarm-type-
qualifier] {alarm-correlation}?
    |  |  +--ro resource                 -> /alarms/alarm-
list/alarm/resource
    |  |  +--ro alarm-type-id            -> /alarms/alarm-
list/alarm[resource=current()/../resource]/alarm-type-id
    |  |  +--ro alarm-type-qualifier     -> /alarms/alarm-
list/alarm[resource=current()/../resource][alarm-type-
id=current()/../alarm-type-id]/alarm-type-qualifier
    |  +--ro impacted-resource*      resource {service-impact-
analysis}?
    |  +--ro root-cause-resource*    resource {root-cause-analysis}?
    |  +--ro time                    yang:date-and-time
    |  +--ro perceived-severity      severity-with-clear
    |  +--ro alarm-text              alarm-text
    +---n alarm-inventory-changed
```

With IETF YANG-Push we like to subscribe

with on-change

- For alert use cases the alarm notifications.

IEEE LLDP – ieee802-dot1ab-lldp.yang

YANG Data Model for LLDP Neighbor Discovery

```
module: ieee802-dot1ab-lldp
  +--rw lldp
     +--ro local-system-data
     |  +--ro chassis-id-subtype?              ieee:chassis-id-subtype-type
     |  +--ro chassis-id?                      ieee:chassis-id-type
     |  +--ro system-name?                     string
     |  +--ro system-description?              string
     |  +--ro system-capabilities-supported?   lldp-types:system-capabilities-map
     |  +--ro system-capabilities-enabled?     lldp-types:system-capabilities-map
     +--rw port* [name dest-mac-address]
        +--rw name                          if:interface-ref
        +--rw dest-mac-address              ieee:mac-address
        +--ro port-id?                      ieee:port-id-type
        +--ro remote-systems-data* [time-mark remote-index]
           +--ro time-mark                        yang:timeticks
           +--ro remote-index                     uint32
           +--ro remote-too-many-neighbors?       boolean
           +--ro remote-changes?                  boolean
           +--ro chassis-id-subtype?              ieee:chassis-id-subtype-type
           +--ro chassis-id?                      ieee:chassis-id-type
           +--ro port-id-subtype?                 ieee:port-id-subtype-type
           +--ro port-id?                         ieee:port-id-type
           +--ro port-desc?                       string
           +--ro system-name?                     string
           +--ro system-description?              string
           +--ro system-capabilities-supported?   lldp-types:system-capabilities-map
           +--ro system-capabilities-enabled?     lldp-types:system-capabilities-map
```

With IETF YANG-Push we like to subscribe

with on-change sync-on-start

- For inventory use cases the local and remote-systems-data.

IEEE dot1ax – ieee802-dot1ax.yang

YANG Data Model for LAG Interface Management

```
module: ieee802-dot1ax
  +--rw lag-system
     +--rw aggregating-system* [agg-system]
        +--rw agg-system          string
        +--rw system-id?          ieee:mac-address
        +--rw system-priority?    uint32

  augment /if:interfaces/if:interface:
    +--rw aggregator
       +--rw name?                           string
       +--rw agg-system-name?                string
       +--rw admin-state?                    enumeration
       +--rw link-up-down-notification?      enumeration
       +--rw collector-max-delay?            int16
       +--rw aggregator-lacp
          +--rw actor-admin-key?    int16

  augment /if:interfaces/if:interface:
    +--rw aggregation-port
       +--rw aggregation-port-lacp
          +--rw actor-system-priority?            int16
          +--rw actor-admin-key?                  int16
          +--rw partner-admin-system-priority?    int16
          +--rw partner-admin-system-id?          ieee:mac-address
          +--rw partner-admin-key?                int16
          +--rw actor-port-priority?              int16
          +--rw partner-admin-port?               int16
          +--rw partner-admin-port-priority?      int16
          +--rw actor-admin-state?                bits
          +--rw partner-admin-state?              bits
```

With IETF YANG-Push we like to subscribe

with on-change sync-on-start

- For inventory use cases the state of the hardware.

RFC 9127 – ietf-bfd.yang

YANG Data Model for Bidirectional Forwarding Detection

BFD IP Single-Hop Hierarchy

https://datatracker.ietf.org/doc/html/rfc9127#section-2.6

```
  notifications:
    +---n singlehop-notification
       +--ro local-discr?                   discriminator
       +--ro remote-discr?                  discriminator
       +--ro new-state?                     state
       +--ro state-change-reason?           iana-bfd-types:diagnostic
       +--ro time-of-last-state-change?     yang:date-and-time
       +--ro dest-addr?                     inet:ip-address
       +--ro source-addr?                   inet:ip-address
       +--ro session-index?                 uint32
       +--ro path-type?                     identityref
       +--ro interface?                     if:interface-ref
       +--ro echo-enabled?                  boolean
```

BFD-over-LAG Hierarchy

https://datatracker.ietf.org/doc/html/rfc9127#section-2

```
  notifications:
    +---n lag-notification
       +--ro local-discr?                   discriminator
       +--ro remote-discr?                  discriminator
       +--ro new-state?                     state
       +--ro state-change-reason?           iana-bfd-types:diagnostic
       +--ro time-of-last-state-change?     yang:date-and-time
       +--ro dest-addr?                     inet:ip-address
       +--ro source-addr?                   inet:ip-address
       +--ro session-index?                 uint32
       +--ro path-type?                     identityref
       +--ro lag-name?                      if:interface-ref
       +--ro member-link?                   if:interface-ref
```

With IETF YANG-Push we like to subscribe

with on-change sync-on-start

- For alert use cases the hardware state notifications.

BFD IP Multihop Hierarchy

https://datatracker.ietf.org/doc/html/rfc9127#section-2.7

```
  notifications:
    +---n multihop-notification
       +--ro local-discr?                   discriminator
       +--ro remote-discr?                  discriminator
       +--ro new-state?                     state
       +--ro state-change-reason?           iana-bfd-types:diagnostic
       +--ro time-of-last-state-change?     yang:date-and-time
       +--ro dest-addr?                     inet:ip-address
       +--ro source-addr?                   inet:ip-address
       +--ro session-index?                 uint32
       +--ro path-type?                     identityref
```

RFC 9130 – ietf-isis.yang

YANG Data Model for the IS-IS Protocol

```
  notifications:
    +---n database-overload
    |  +--ro routing-protocol-name?   -> /rt:routing/
    |  |                                 control-plane-protocols/
    |  |                                 control-plane-protocol/name
    |  +--ro isis-level?              level
    |  +--ro overload?                enumeration
    +---n if-state-change
    |  +--ro routing-protocol-name?   -> /rt:routing/
    |  |                                 control-plane-protocols/
    |  |                                 control-plane-protocol/name
    |  +--ro isis-level?              level
    |  +--ro interface-name?          if:interface-ref
    |  +--ro interface-level?         level
    |  +--ro extended-circuit-id?     extended-circuit-id
    |  +--ro state?                   if-state-type
    +---n adjacency-state-change
    |  +--ro routing-protocol-name?   -> /rt:routing/
    |  |                                 control-plane-protocols/
    |  |                                 control-plane-protocol/name
    |  +--ro isis-level?              level
    |  +--ro interface-name?          if:interface-ref
    |  +--ro interface-level?         level
    |  +--ro extended-circuit-id?     extended-circuit-id
    |  +--ro neighbor?                string
    |  +--ro neighbor-system-id?      system-id
    |  +--ro state?                   adj-state-type
    |  +--ro reason?                  string
```

With IETF YANG-Push we like to subscribe

with on-change sync-on-start

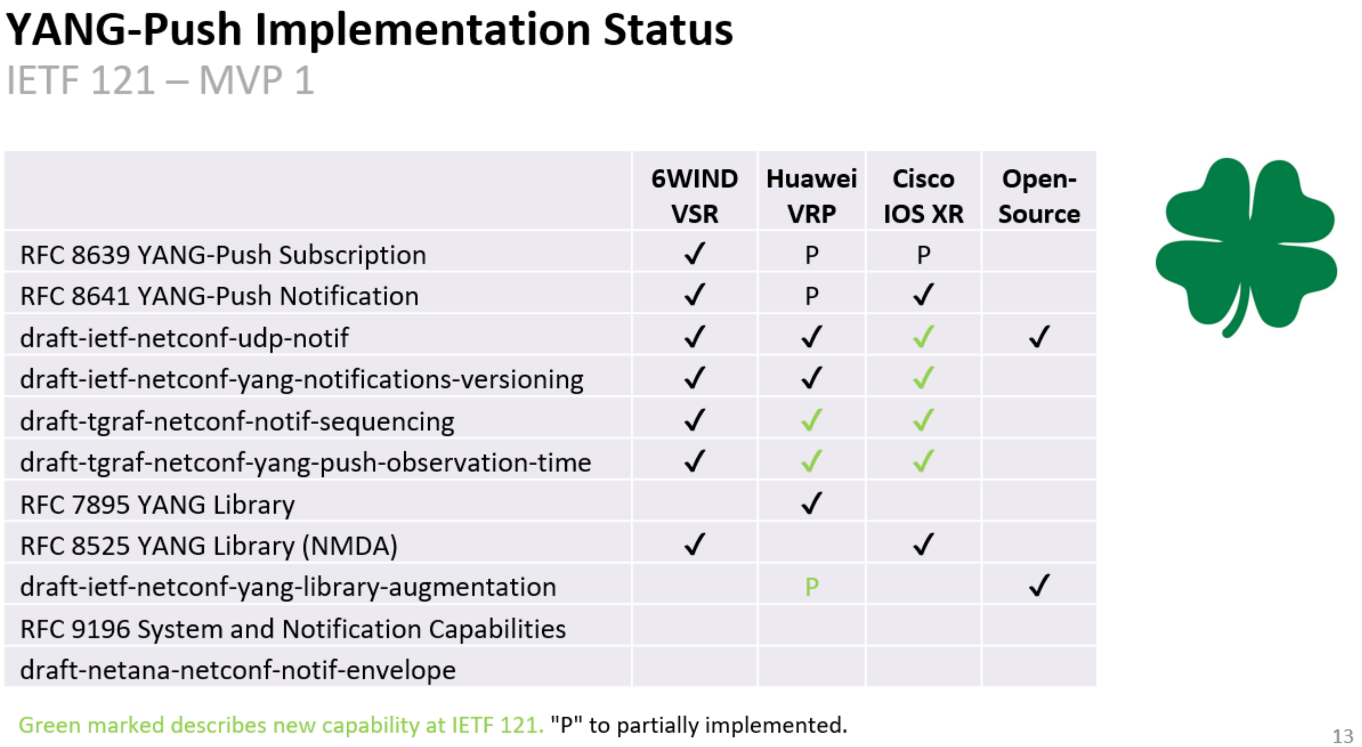
- For alert use cases the hardware state notifications.

An Architecture for YANG-Push to Message Broker Integration

IETF 122 Hackathon

Validate Configured Subscription YANG-Push Publisher Implementations

- **Champion(s)**
  Thomas Graf (thomas.graf @ swisscom.com)
  Yannick Buchs (yannick.buchs @ swisscom.com)
  Daniel Voyer (danvoyerwork @ gmail.com)
  Holger Keller (holger.keller @ telekom.de)
  Rob Wilton (rwilton @ cisco.com)
  Benoit Claise (benoit.claise@huawei.com)
  Qiufang Ma (maqiufang1 @ huawei.com)
  Jérémie Leska (jeremie.leska @ 6wind.com)
  Samuel Gauthier (samuel.gauthier @ 6wind.com)
- **Draft Specifications**
  https://datatracker.ietf.org/doc/html/rfc8639
  https://datatracker.ietf.org/doc/html/rfc8641
  https://datatracker.ietf.org/doc/html/rfc9196
  https://datatracker.ietf.org/doc/html/draft-netana-netconf-notif-envelope
  https://datatracker.ietf.org/doc/html/draft-ietf-netconf-yang-notifications-versioning
  https://datatracker.ietf.org/doc/html/draft-ietf-netconf-udp-notif
  https://datatracker.ietf.org/doc/html/draft-ietf-netconf-distributed-notif
  https://datatracker.ietf.org/doc/html/draft-netana-netconf-yp-transport-capabilities
- **Project Info**
  Validates and verify 3 YANG-Push vendor implementations in the area of:
- *Subscription automation*
  Discover YANG-Push systems and notifications capabilities and configure periodical and on-change subscriptions with netconf.
- *Notification integration*
  Validate subscription state change and push-update and push-change-update notifications for draft-ietf-nmop-yang-message-broker-integration integration.
- *Configured subscription transport integration*
  Validate draft-ietf-netconf-udp-notif and draft-ietf-netconf-distributed-notif packet format on the wire.
- **Repository**
  https://github.com/network-analytics/ietf-network-analytics-document-status/tree/main/122/Hackathon



YANG-Push Implementation Status

IETF 121 – MVP 1

	6WIND VSR	Huawei VRP	Cisco IOS XR	Open-Source
RFC 8639 YANG-Push Subscription	✓	P	P	
RFC 8641 YANG-Push Notification	✓	P	✓	
draft-ietf-netconf-udp-notif	✓	✓	✓	✓
draft-ietf-netconf-yang-notifications-versioning	✓	✓	✓	
draft-tgraf-netconf-notif-sequencing	✓	✓	✓	
draft-tgraf-netconf-yang-push-observation-time	✓	✓	✓	
RFC 7895 YANG Library		✓		
RFC 8525 YANG Library (NMDA)	✓		✓	
draft-ietf-netconf-yang-library-augmentation		P		✓
RFC 9196 System and Notification Capabilities				
draft-netana-netconf-notif-envelope				

Green marked describes new capability at IETF 121. "P" to partially implemented.

13

Yang
Kafka

Extensible YANG model for YANG-Push Notifications

For XML, JSON or CBOR encoded messages with hostname, sequence-number and observation-time

```
notifications:
  +---n envelope
     +--ro event-time                   yang:date-and-time
     +--ro hostname?                    inet:host
     |       {notification-hostname-sequence-number}?
     +--ro sequence-number?             yang:counter32
     |       {notification-hostname-sequence-number}?
     +--ro notification-contents?       <anydata>

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-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-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
```

- draft-netana-netconf-notif-envelope defines new extensible notification structure, defined in YANG, for use in YANG-Push Notification messages enabling any YANG compatible encodings such as XML RFC 7950, JSON RFC 7951 or CBOR RFC 9264.
- New notification envelope can be enabled in "ietf-subscribed-notification" RFC 8639.
- Capability can be discovered through 'ietf-notification-capabilities' RFC 9196.
- Supports the following notification metadata extensions
  - **hostname:** Describes the node's hostname according to the 'sysName' object definition in RFC 1213 from where the message was published from. This value is usually configured on the node by the administrator to uniquely identify the node in the network.
  - **sequence-number:** Generates a unique sequence number for each published message by the publisher process. The number counts up at every published notification message as described in RFC 9187.
  - **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.

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]
    |       {yang-push-revision-supported}?
    |  +--ro module-name       yang:yang-identifier
    |  +--ro revision          rev:revision-date
    |  +--ro revision-label?   ysver:version
    +--ro yang-library-content-id?       -> /yanglib:yang-library/content-id
            {yang-push-revision-supported}?
 {
     "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-yang-push-revision:yang-library-content-id": "1",
         "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 sub-tree filter also the yang module name, revision, revision-label and the yang-library-content-id.

    With YANG Library content-id a YANG-Push receiver is now able to detect changes in the YANG library. This includes also the imported YANG modules of the subscribed xpath.

  - Extends RFC 9196 defined subscription-capabilities with a **yang-push-module-revision-supported** leaf.

YANG Notification Transport Capabilities

Extending System Capabilities for YANG-Push Configured Subscription Transport

```
module: ietf-notification-transport-capabilities

  augment /sysc:system-capabilities/notc:subscription-capabilities:
    +--ro transport-capabilities
       +--ro transport-capability* [transport-protocol]
          +--ro transport-protocol    identityref
          +--ro security-protocol?    identityref
          +--ro encoding-format*      identityref
```

```
augment "/sysc:system-capabilities/notc:subscription-capabilities" {
    description "Add system level capability.";
    container transport-capabilities {
      description "Capabilities related to YANG-Push transports.";
      list transport-capability {
        key "transport-protocol";
        description "Capability list related to notification transport capabilities.";
        leaf transport-protocol {
          type identityref {
            base sn:transport;
          }
          description "Supported transport protocol for YANG-Push.";
        }
        leaf security-protocol {
          type identityref {
            base security-protocol;
          }
          description "Type of secure transport.";
        }
        leaf-list encoding-format {
          type identityref {
            base sn:encoding;
          }
          description "Supported encoding formats.";
        }
      }
    }
```

- draft-netana-netconf-yp-transport-capabilities augments System Capabilities model and provides additional transport related attributes associated with system capabilities:
  - Specification of transport protocols the client can request to establish a draft-ietf-netconf-udp-notif or draft-ietf-netconf-https-notif configured transport connection;
  - Specification of transport encoding, such as JSON or XML as defined in RFC 8040 or CBOR as defined in RFC 9254 the client can request to encode YANG notifications;
  - Specification of secure transport mechanisms that are needed by the client to communicate with the server such as DTLS as defined in RFC 9147 TLS as defined in RFC 8446 or SSH as defined in RFC 4254;

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, 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-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.
```

YANG-Push Operational Data Observability Enhancements

Simplifies by combining periodic and on-change subscription

```
module: ietf-yp-ext

  augment /sn:subscription-started/yp:update-trigger:
    +--:(periodic-and-on-change) {yp:on-change}?
       +-- periodic-and-on-change!
          +-- period               yp:centiseconds
          +-- anchor-time?         yang:date-and-time
          +-- dampening-period?    yp:centiseconds
          +-- sync-on-start?       boolean
          +-- excluded-change*     yp:change-type
  augment /sn:subscription-started:
    +--ro common-notification-format?   boolean
  augment /sn:subscription-modified/yp:update-trigger:
    +--:(periodic-and-on-change) {yp:on-change}?
       +-- periodic-and-on-change!
          +-- period               yp:centiseconds
          +-- anchor-time?         yang:date-and-time
          +-- dampening-period?    yp:centiseconds
          +-- sync-on-start?       boolean
          +-- excluded-change*     yp:change-type
  augment /sn:subscription-modified:
    +--ro common-notification-format?   boolean
  augment /sn:subscriptions/sn:subscription/yp:update-trigger:
    +--:(periodic-and-on-change) {yp:on-change}?
       +--rw periodic-and-on-change!
          +--rw period               yp:centiseconds
          +--rw anchor-time?         yang:date-and-time
          +--rw dampening-period?    yp:centiseconds
          +--rw sync-on-start?       boolean
          +--rw excluded-change*     yp:change-type
  augment /sn:subscriptions/sn:subscription:
    +--rw common-notification-format?   boolean
```

- To reduce complexities in modelling the operational state, the following two YANG-Push enhancements are proposed:
  - A new YANG-Push encoding format that can be used for both on-change and periodic subscriptions that reports the data from the subscription filter point.
  - A combined periodic and on-change subscription that reports events on a periodical cadence and also if changes to the data have occurred.

```
notifications:
  +---n update
     +--ro id?                    sn:subscription-id
     +--ro subscription-path?     yang:xpath1.0
     +--ro target-path?           string
     +--ro snapshot-type?         enumeration
     +--ro observation-time?      yang:date-and-time
     +--ro datastore-snapshot?    <anydata>
     +--ro incomplete?            empty
```

- This removes the YANG Patch format RFC 8072 dependency and eases the message broker integration.
- Allows the YANG-Push publisher to split a subscription into smaller child subscriptions for more efficient independent and concurrent processing. Reuses the ideas from draft-ietf-netconf-distributed-notif. Child subscriptions remain encoded from the same subscription point.

IETF YANG-Push Implementations and Next Steps

Challenges and how to solve…

What in IETF YANG-Push could have been defined differently and why?

What prevents IETF YANG-Push for being integrated/used efficiently?

What in IETF YANG-Push is missing and for which purpose?

- On-change notification schema different than periodical
- Patch-id in On-Change complex to implement.
- Reduce YANG complexity (example: augmentations, deviations, xpath, lists)
- Each subscribed xpath needs normalization. High effort with many vendor specific YANG modules
- Missing end to end open-source implementations

- Extensible YANG-Push header combing notification and subscription. Separation of header and subscribed content is needed to allow partial parsing of message in binary encoding for data processing chain.
- On-change and periodical notification schema should have identical schema and contain the entire schema tree below subscription and represent current state.
- Common alignment on what should be supported in xpath and what not.
- **Agile incremental driven development. Deployment guide describing implementers and operators what is/should be supported at which MVP stage.**