An Architecture for YANG-Push to Message Broker Integration

draft-ietf-nmop-yang-message-broker-integration-11

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

19. March 2026

An Architecture for YANG-Push to Message Broker Integration

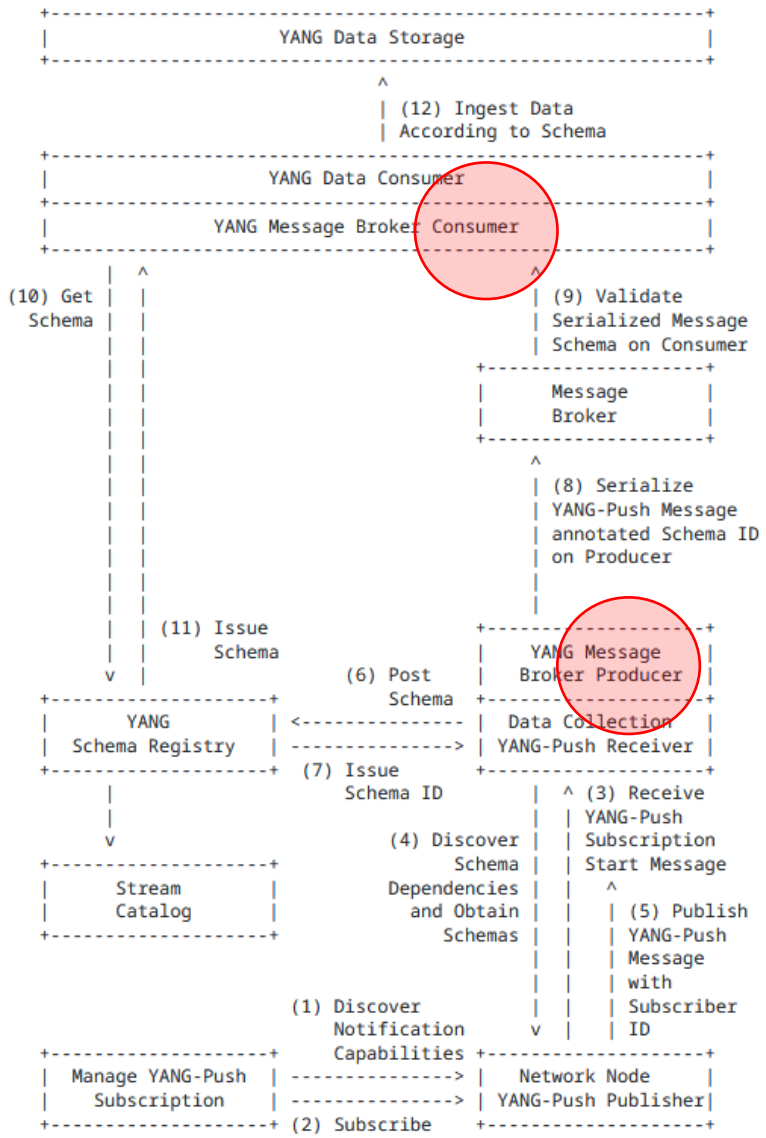
Status from -10 and -11 and Next Steps

Status

- Expanded system architecture picture with YANG Message Broker Producer and Consumer
- Expanded terminology section with terms defined by the document.
- Detailed YANG-Push Receiver and YANG Schema Registry sections to describe and example how YANG features are being covered in schema registration.
- Added IETF 125 in the milestone appendix to describe the current implementation states and the planned hackathon items.
- Minor editorial edits.

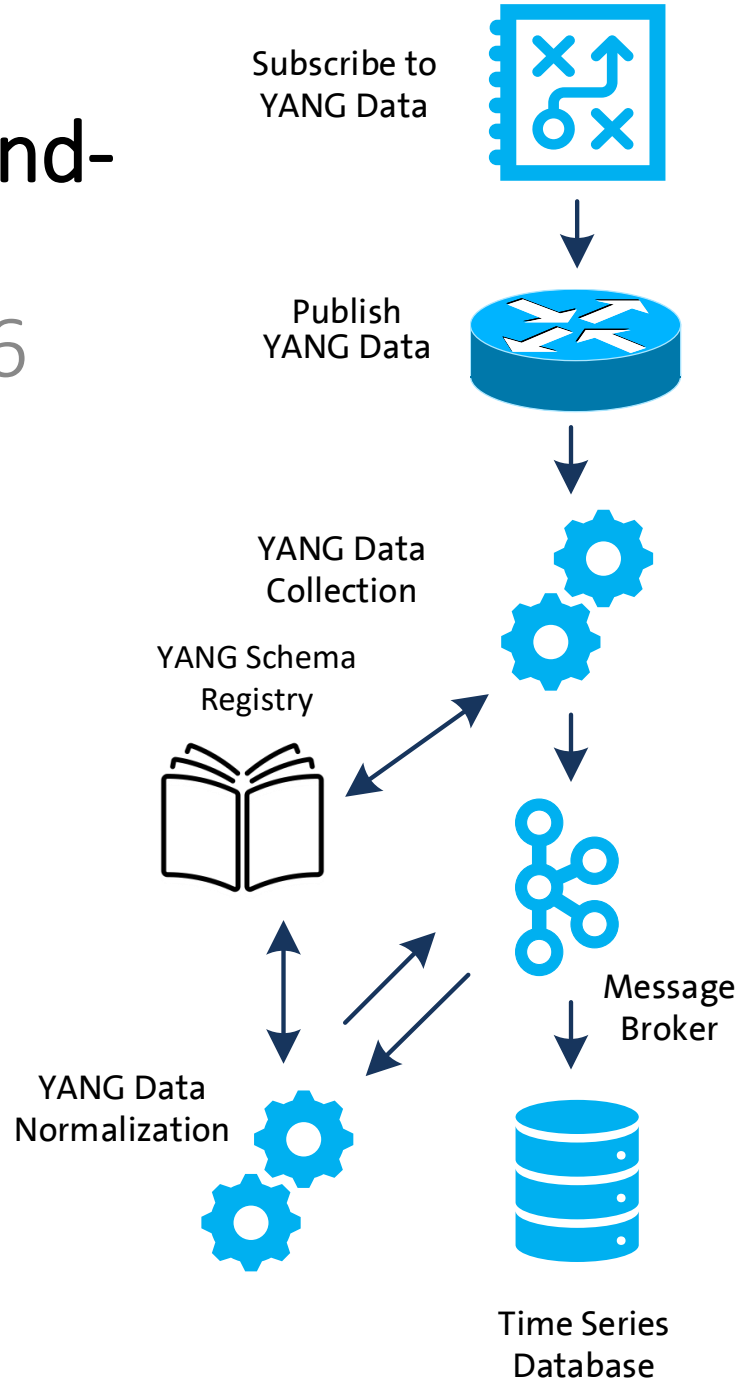
Next Steps

➢ Add security and operations considerations section.



Validate YANG-Push to Message Broker End-To-End Data Processing Chain

IETF 125 Hackathon, March 14-15th 2026



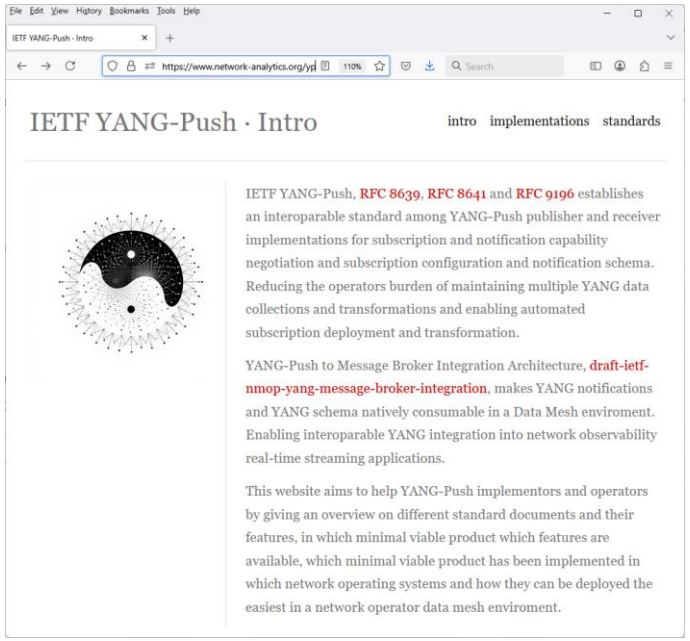
Hackathon Plan, Software and Website

Test Plan

Validate and verify

- 5 YANG-Push Publishers
- 2 YANG-Push Receivers
- 2 YANG-Push Network Telemetry Message
- 1 YANG Message Broker Producer and Schema Registry
- 3 YANG Message Broker Consumers

implementation in YANG data integration automation. Subscribe to YANG data on YANG-Publisher, obtain and register all YANG modules necessary to build YANG schema tree, register YANG schemas to Schema Registry and verify YANG notifications against scheme trees and produce and consume from Message Broker.



Software

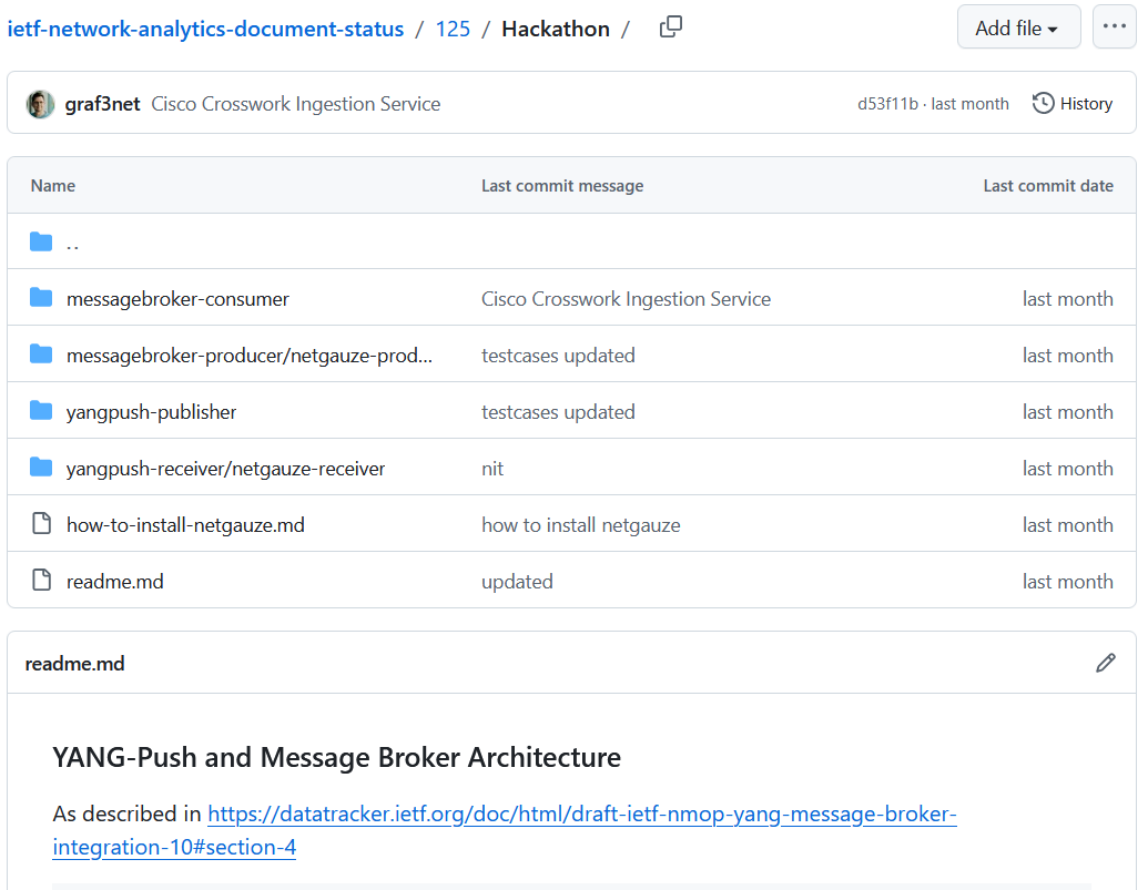
https://www.network-analytics.org/yp/how-to-deploy.html

- YANG-Push Publisher - Cisco IOS XR, 6WIND VSR, Huawei NE (Router) and MA (OLT) and Arrcus Arcos
- YANG-Push Receiver – NetGauze and Pmacct
- YANG Message Broker Producer – NetGauze
- YANG Message Broker Consumer – NetGauze, Ciena Blueplanet UAA Workflow Engine, Cisco Crosswork Ingestion Service
- YANG Schema Registry – Apache Kafka YANG Plugin
- Libyang - https://github.com/CESNET/libyang/compare/master...devel
- Yangkit - https://github.com/yang-central/yangkit/pull/41
- Wireshark - udp-notif dissector

Hackathon – Repositories

The results of the end-to-end tests

- Netconf RPCs and YANG-Push JSON encoded notifications with wire captures from receiver
- YANG-Push Receiver cache content generation with YANG library context, YANG modules, metadata and notification schema validation logs
- YANG Message Broker Producer serialization logs
- YANG Schema Registry schema examples
- YANG Message Broker Consumer, serialization logs, YANG library context, YANG modules and message schema validation

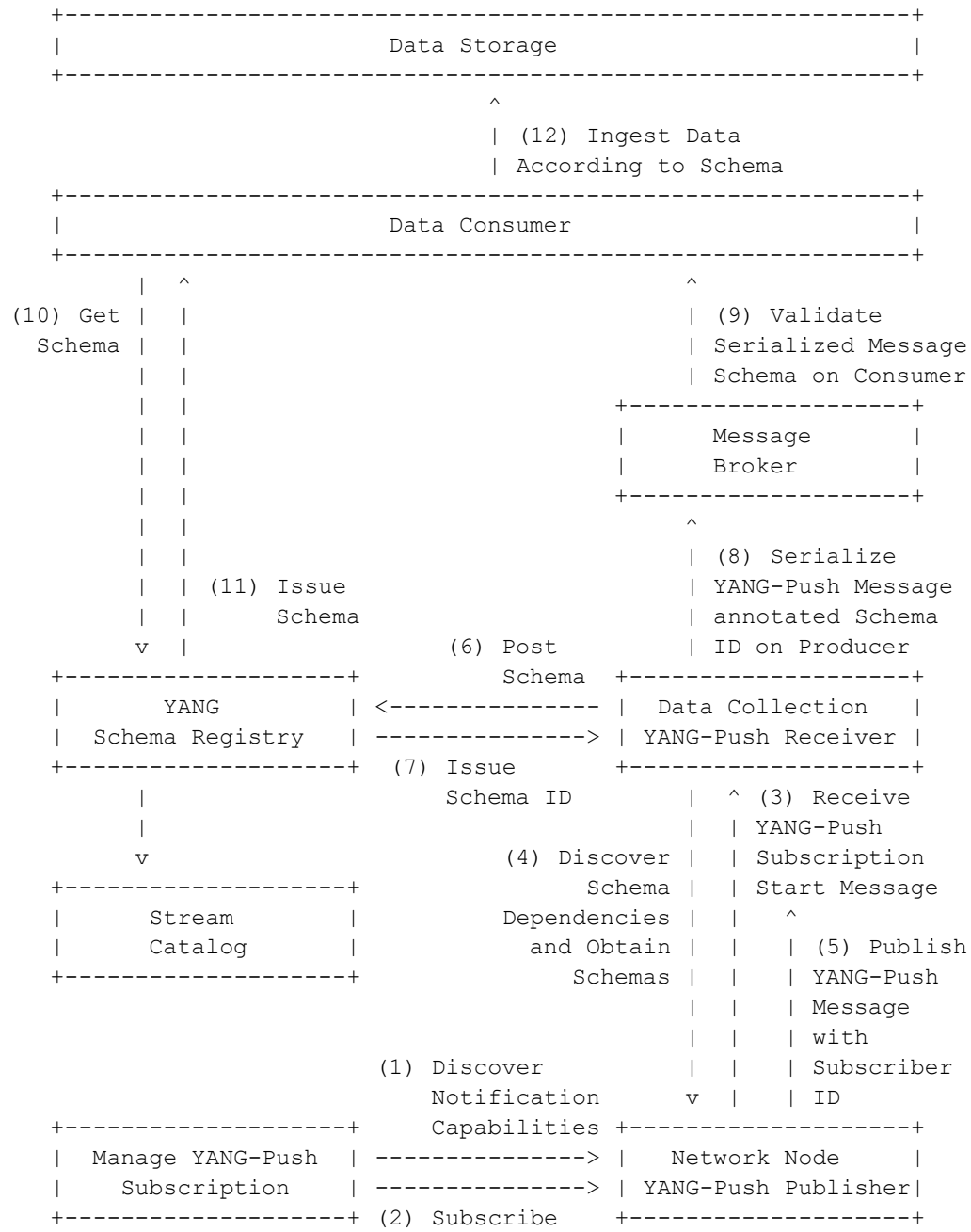


https://github.com/network-analytics/ietf-network-analytics-document-status/tree/main/125/Hackathon/

An Architecture for YANG-Push to Message Broker Integration

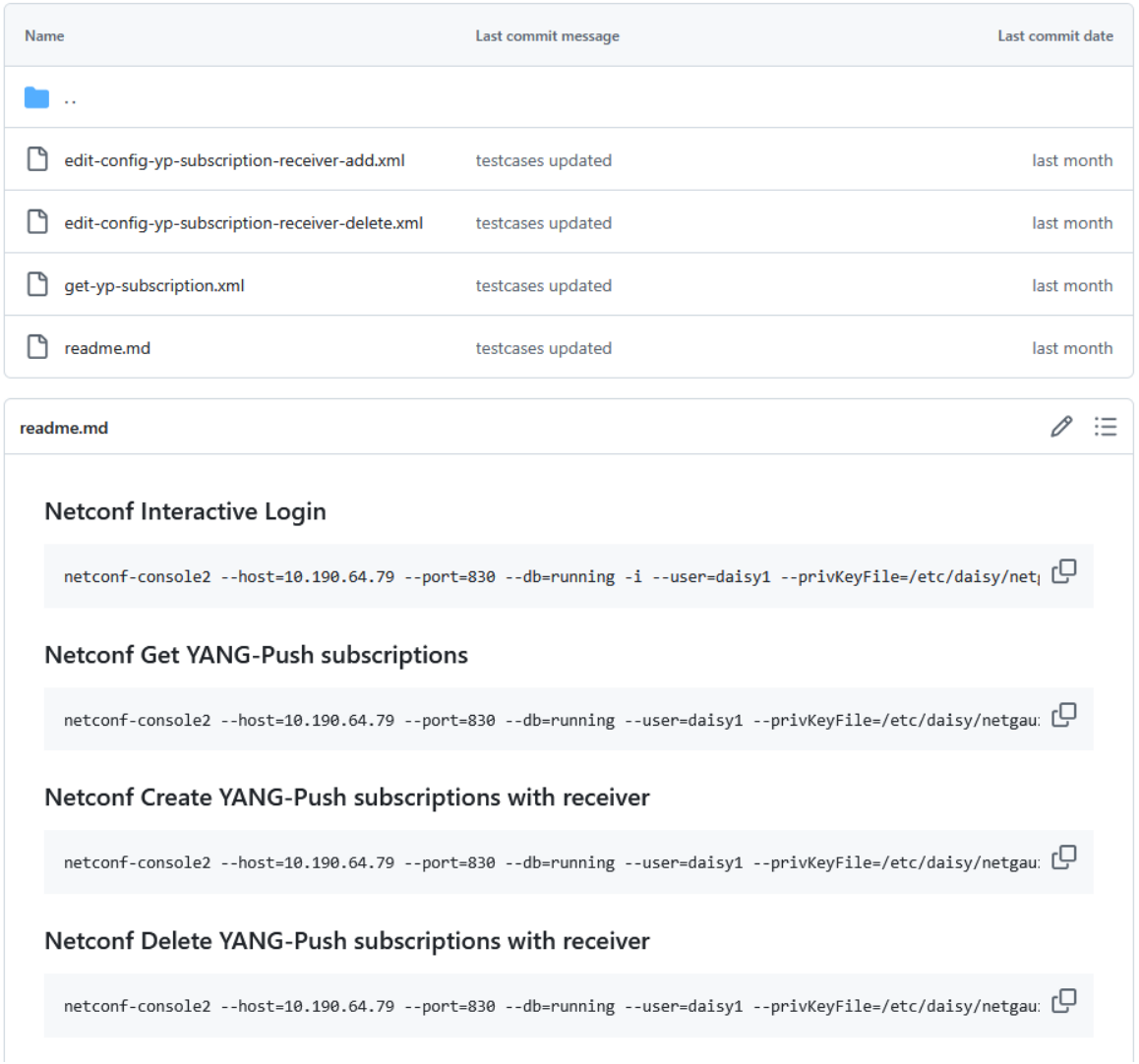
draft-ietf-nmop-yang-message-broker-integration
draft-ietf-nmop-message-broker-telemetry-message
draft-ietf-nmop-yang-message-broker-message-key
draft-netana-nmop-message-broker-bmp-telemetry-message

- Subscription to YANG Notifications
  RFC 8639
- Subscription to YANG Notifications for Datastore Updates
  RFC 8641
- UDP-based Transport for Configured Subscriptions
  draft-ietf-netconf-udp-notif
- Subscription to Distributed Notifications
  draft-ietf-netconf-distributed-notif
- Extensible YANG Model for YANG-Push Notifications
  draft-ietf-netconf-notif-envelope
- Support of Versioning in YANG Notifications Subscription
  draft-ietf-netconf-yang-notifications-versioning
- YANG Modules Describing Capabilities for Systems and Datastore Update Notifications
  RFC 9196
- YANG Notification Transport Capabilities
  draft-ietf-netconf-yp-transport-capabilities
- YANG Library
  RFC 8525
- Augmented-by Addition into the IETF-YANG-Library
  draft-ietf-netconf-yang-library-augmentation
- Validating anydata in YANG Library context
  draft-ietf-netmod-yang-anydata-validation
- Encoding of Data Modeled with YANG in the CBOR
  RFC 9254



YANG-Push Publisher - Subscription

- Netconf RPC calls to create, delete and obtain YANG-Push subscriptions and receiver configurations.



Name	Last commit message	Last commit date
..		
edit-config-yp-subscription-receiver-add.xml	testcases updated	last month
edit-config-yp-subscription-receiver-delete.xml	testcases updated	last month
get-yp-subscription.xml	testcases updated	last month
readme.md	testcases updated	last month

readme.md

Netconf Interactive Login

```
netconf-console2 --host=10.190.64.79 --port=830 --db=running -i --user=daisy1 --privKeyFile=/etc/daisy/net
```

Netconf Get YANG-Push subscriptions

```
netconf-console2 --host=10.190.64.79 --port=830 --db=running --user=daisy1 --privKeyFile=/etc/daisy/netgau
```

Netconf Create YANG-Push subscriptions with receiver

```
netconf-console2 --host=10.190.64.79 --port=830 --db=running --user=daisy1 --privKeyFile=/etc/daisy/netgau
```

Netconf Delete YANG-Push subscriptions with receiver

```
netconf-console2 --host=10.190.64.79 --port=830 --db=running --user=daisy1 --privKeyFile=/etc/daisy/netgau
```

YANG-Push Publisher - Notifications

- Subscription state change notifications as JSON and PCAP.
- Push update and change-update notifications as JSON and PCAP.

Name	Last commit message	Last commit date
..		
1-push-update.json	testcases updated	last month
1-subscription-started.json	testcases updated	last month
1-yp-subscripton.pcap	testcases updated	last month
2-push-update.json	testcases updated	last month
2-subscription-started.json	testcases updated	last month
2-yp-subscripton.pcap	testcases updated	last month
27-push-update.json	testcases updated	last month
27-subscription-started.json	testcases updated	last month
27-yp-subscripton.pcap	testcases updated	last month
readme.md	testcases updated	last month

readme.md

YANG-Push Notifications

subscription-started, push-update and push-change-update notifications in raw and json sorted by subscription id. pcap requires https://github.com/network-analytics/udp-notif-dissector Wireshark dissector.

YANG-Push

IETF 125 Publisher Implementation Status

	6WIND VSR	Huawei NE	Huawei MA	Cisco IOS XR	Arrcus Arcos	Open-Source
RFC 8639 YANG-Push Subscription	✓	✓	✓	✓	✓	
RFC 8641 YANG-Push Notification	✓	✓	✓	✓	✓	
draft-ietf-netconf-udp-notif	✓	✓	✓	✓	✓	✓
draft-ietf-netconf-distributed-notif	✓	✓	✓		na	
draft-ietf-netconf-notif-envelope	✓	✓	✓	✓	✓	
draft-ietf-netconf-yang-notifications-versioning	✓	✓	✓	✓	✓	
RFC 8525 YANG Library	✓	✓	✓	✓	✓	
draft-ietf-netconf-yang-library-augmentation	✓	✓	✓	✓		✓
RFC 9196 System and Notification Capabilities			✓	P	✓	
draft-netana-netconf-yp-transport-capabilities			✓	✓	✓	
RFC 9254 CBOR Named Identifiers	✓					

[NetGauze](#)
[Pmacct](#)
[udp-notif-c-collector](#)
[yang-library-augmentedby](#)

Green marked describes new capability at IETF 125. "P" to partially implemented.

Operational IETF/IEEE YANG

IETF 125 Module Implementation Status

	6WIND VSR	Huawei NE	Huawei MA	Cisco IOS XR	Arrcus Arcos
ietf-interfaces.yang (inventory, state and statistic)	✓	P	✓		
ietf-hardware.yang (inventory, state and statistic)	(✓)		✓		
ietf-alarms.yang (state)	✓		✓		
ieee802-dot1ab-lldp.yang (inventory and state)	(✓)				
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)					

Green marked describes new capability at IETF 125. "P" to partially implemented.

YANG-Push Receiver - NetGauze

- Describes how NetGauze organizes and caches YANG modules for a set of YANG-Push Subscriptions with the same YANG schema tree.
- Includes generated YANG library and YANG-Push Subscription metadata.
- Generates unique cache contend-id. Same principle as with YANG library contend-id on YANG-Push Publisher.
- Repository examples with two sets of YANG-Push Subscriptions in two tar.gz files.

1. YANG-Push Receiver - Per YANG-Push Subscription YANG Schema Retrieval

https://datatracker.ietf.org/doc/html/draft-ietf-nmop-yang-message-broker-integration-10#section-4.3 describes that once a new YANG-Push subscription is established all subscription related YANG schemas are obtained from YANG-Push publisher and cached locally. The dependencies include: imports, augments, deviations and features being used for each YANG module. YANG library is used to determine dependencies on augments, deviations and features where YANG modules are used to determine their imports.

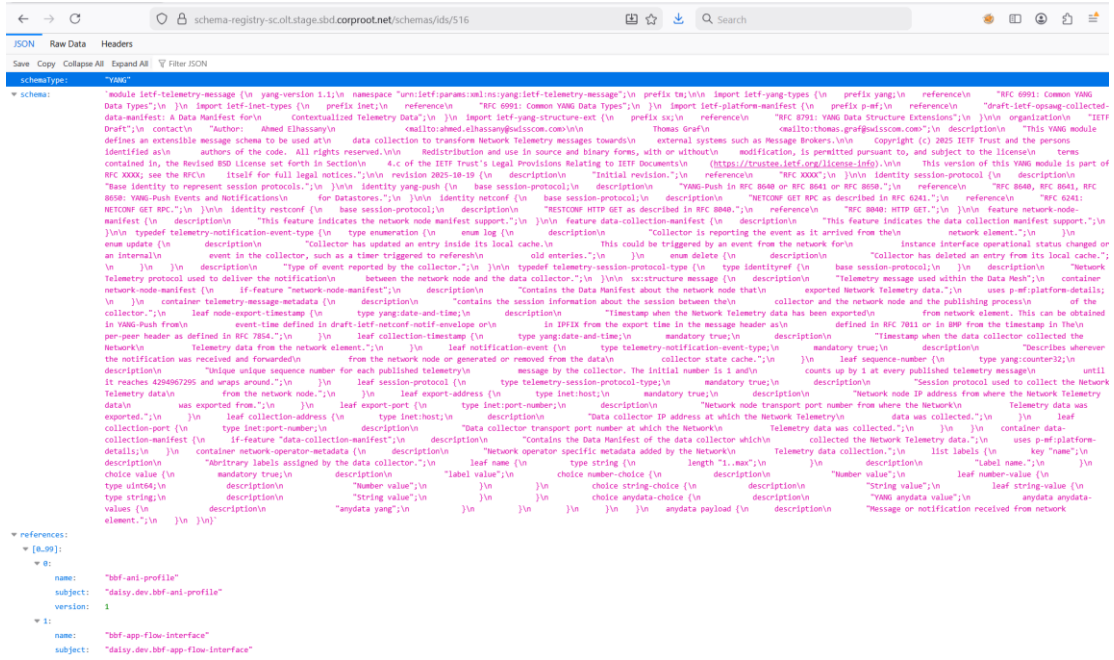
1.1 NetGauze Implementation

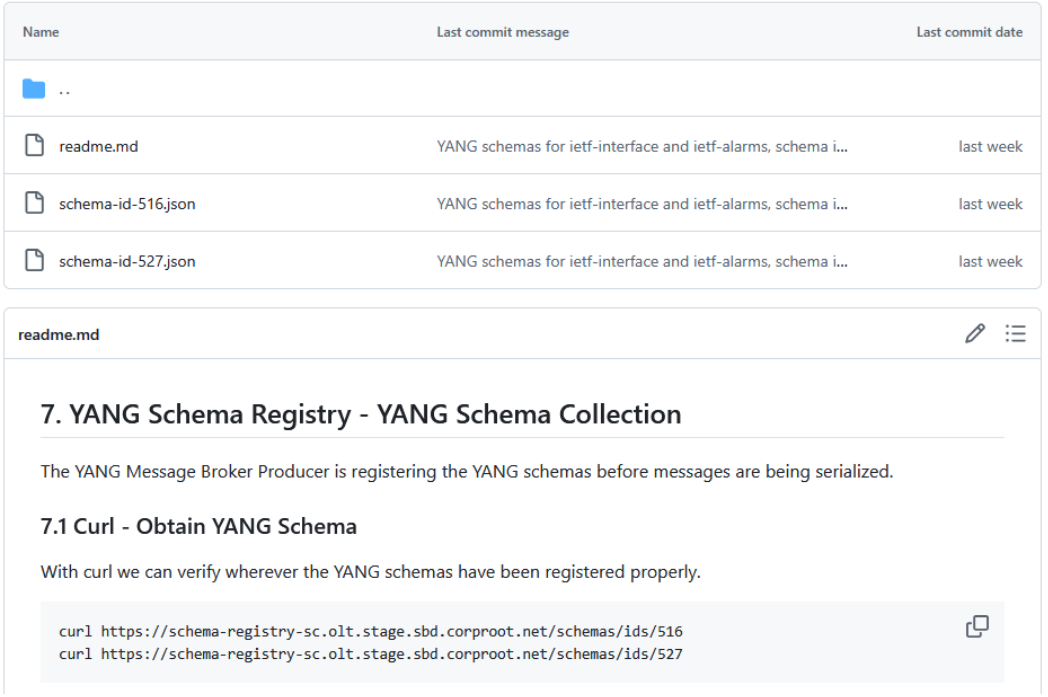
In `/var/cache/NetGauze/yang-schemas/` for each unique YANG schema tree a new directory is being automatically created and named after a locally generated cache contend-id. Within the subdirectory modules all netconf `<get-schema>` collected YANG modules are stored and with `yang-lib.xml` a YANG library context is being generated automatically. The `subscription-info.json` describes the relation between YANG-Push publisher node, YANG-Push subscription id, YANG library content-id of the YANG-Push publisher and the receiver and the YANG-Push subscription started metadata describing which xpath in which datastore and list of directly imported YANG modules. See below an example for YANG library cache contend-id `06bb7b0eac2786553a446740ed61659405d4e6d5fb4242f306234ee472c85827`.

```
[root@dev-col-left01 ~]# cd  /var/cache/NetGauze/yang_schemas/06bb7b0eac2786553a446740ed61659405d4e6d5fb42
[root@dev-col-left01 06bb7b0eac2786553a446740ed61659405d4e6d5fb4242f306234ee472c85827]# ll
total 160
drwxr-xr-x 2 root root  20480 Jan 22 14:43 modules
-rw-r--r-- 1 root root   3588 Jan 30 15:56 subscriptions-info.json
-rw-r--r-- 1 root root 133250 Jan 22 14:43 yang-lib.xml
[root@dev-col-left01 06bb7b0eac2786553a446740ed61659405d4e6d5fb4242f306234ee472c85827]# cat subscriptions-
[
  {
    "peer": "10.190.64.79:10100",
    "id": 1,
    "content_id": "22851",
    "target": {
      "ietf-yang-push:datastore": "ietf-datastores:operational",
      "ietf-yang-push:datastore-xpath-filter": "/ietf-interfaces:interfaces/interface[type='iana-if-type:e
    },
    "models": [
      "iana-if-type",
      "huawei-interface-statistics-mgt",
      "ietf-interfaces",
      "ietf-subscribed-notifications"
    ]
```

YANG Schema Registry

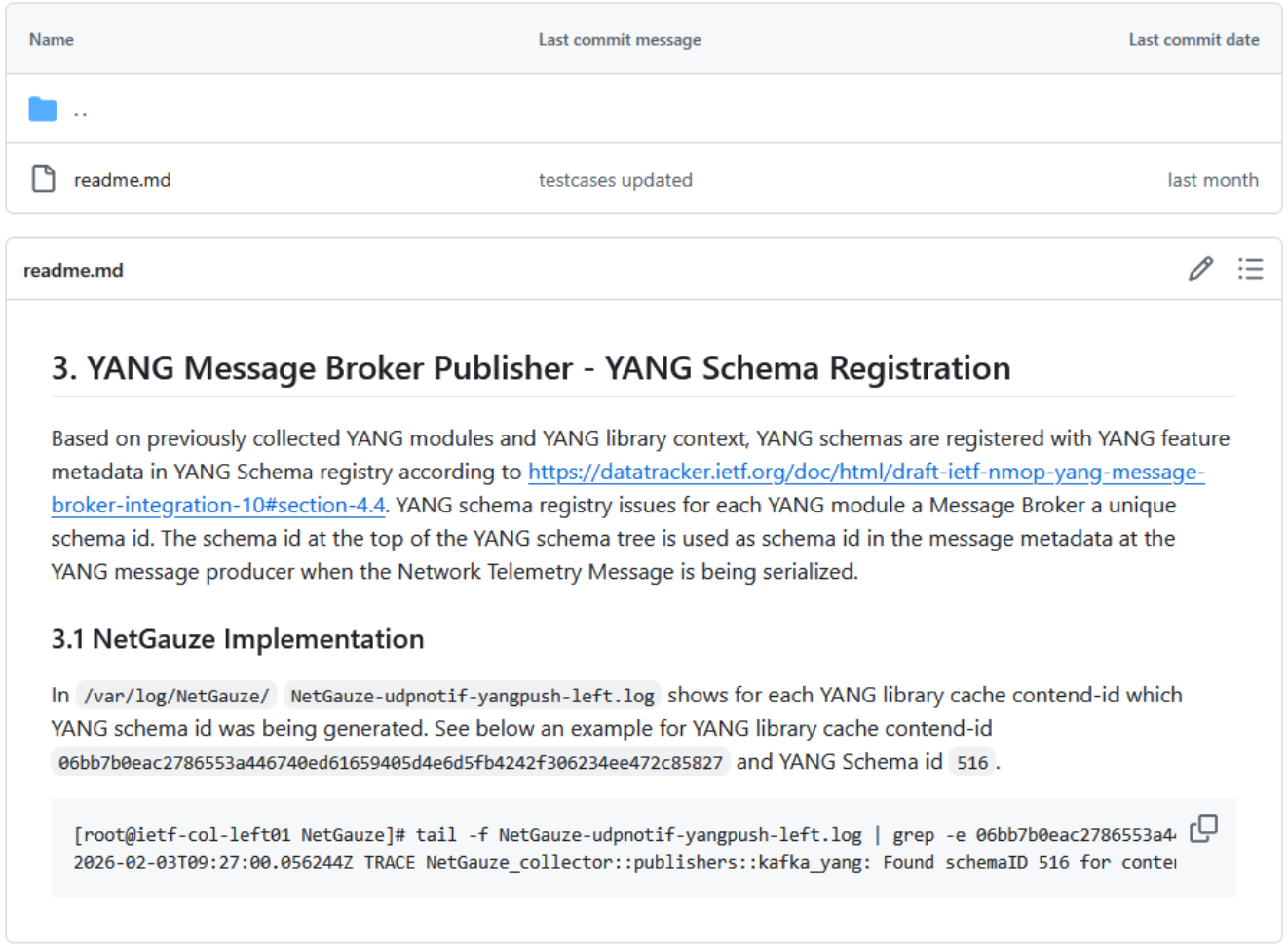
- Examples two YANG schema trees registgered as schema id 516 and 527.
- Shows ietf-telemetry-message as the top YANG module and its references.





Message Broker Producer - NetGauze

- Shows in tracelogs how NetGauze maps YANG cache contend-id to YANG schema-id.
- Troubleshooting can be performed with notificication sequence number per Message Broker telemetry message.



Name	Last commit message	Last commit date
..		
readme.md	testcases updated	last month

readme.md

3. YANG Message Broker Publisher - YANG Schema Registration

Based on previously collected YANG modules and YANG library context, YANG schemas are registered with YANG feature metadata in YANG Schema registry according to https://datatracker.ietf.org/doc/html/draft-ietf-nmop-yang-message-broker-integration-10#section-4.4. YANG schema registry issues for each YANG module a Message Broker a unique schema id. The schema id at the top of the YANG schema tree is used as schema id in the message metadata at the YANG message producer when the Network Telemetry Message is being serialized.

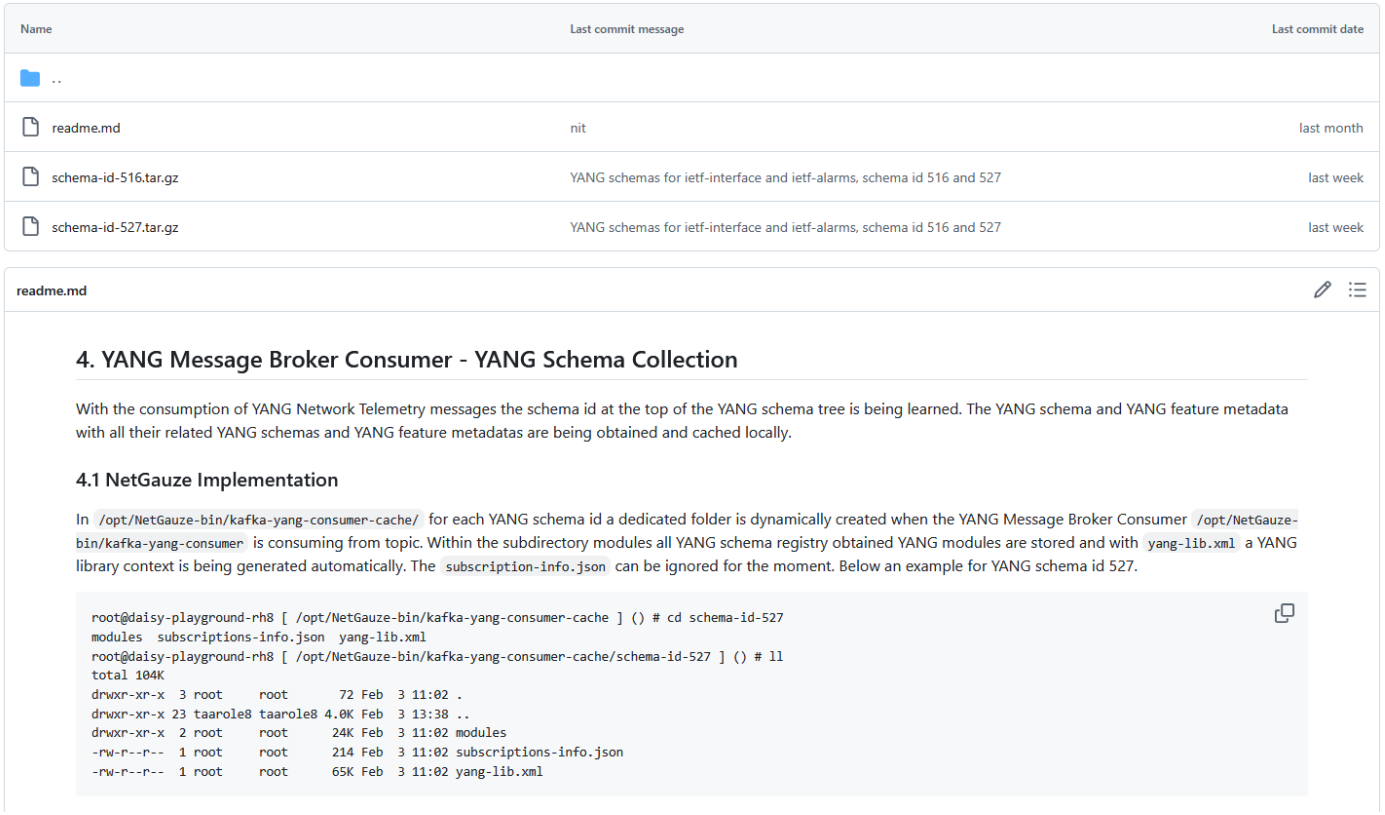
3.1 NetGauze Implementation

In `/var/log/NetGauze/` `NetGauze-udpnotif-yangpush-left.log` shows for each YANG library cache contend-id which YANG schema id was being generated. See below an example for YANG library cache contend-id `06bb7b0eac2786553a446740ed61659405d4e6d5fb4242f306234ee472c85827` and YANG Schema id `516`.

```
[root@ietf-col-left01 NetGauze]# tail -f NetGauze-udpnotif-yangpush-left.log | grep -e 06bb7b0eac2786553a4
2026-02-03T09:27:00.056244Z TRACE NetGauze_collector::publishers::kafka_yang: Found schemaID 516 for conte
```

Message Broker Consumer – NetGauze (1)

- Describes how NetGauze organizes and caches YANG modules for a set of YANG-Push Subscriptions with the same YANG schema tree with schema-id.
- Includes generated YANG library.
- Repository examples with two sets of YANG-Push Subscriptions in two tar.gz files.



Name	Last commit message	Last commit date
..		
readme.md	nit	last month
schema-id-516.tar.gz	YANG schemas for ietf-interface and ietf-alarms, schema id 516 and 527	last week
schema-id-527.tar.gz	YANG schemas for ietf-interface and ietf-alarms, schema id 516 and 527	last week

readme.md

4. YANG Message Broker Consumer - YANG Schema Collection

With the consumption of YANG Network Telemetry messages the schema id at the top of the YANG schema tree is being learned. The YANG schema and YANG feature metadata with all their related YANG schemas and YANG feature metadatas are being obtained and cached locally.

4.1 NetGauze Implementation

In `/opt/NetGauze-bin/kafka-yang-consumer-cache/` for each YANG schema id a dedicated folder is dynamically created when the YANG Message Broker Consumer `/opt/NetGauze-bin/kafka-yang-consumer` is consuming from topic. Within the subdirectory modules all YANG schema registry obtained YANG modules are stored and with `yang-lib.xml` a YANG library context is being generated automatically. The `subscription-info.json` can be ignored for the moment. Below an example for YANG schema id 527.

```
root@daisy-playground-rh8 [ /opt/NetGauze-bin/kafka-yang-consumer-cache ] () # cd schema-id-527
modules  subscriptions-info.json  yang-lib.xml
root@daisy-playground-rh8 [ /opt/NetGauze-bin/kafka-yang-consumer-cache/schema-id-527 ] () # ll
total 104K
drwxr-xr-x  3 root     root       72 Feb  3 11:02 .
drwxr-xr-x 23 taarole8 taarole8 4.0K Feb  3 13:38 ..
drwxr-xr-x  2 root     root      24K Feb  3 11:02 modules
-rw-r--r--  1 root     root      214 Feb  3 11:02 subscriptions-info.json
-rw-r--r--  1 root     root      65K Feb  3 11:02 yang-lib.xml
```

Message Broker Consumer – NetGauze (2)

- Shows in tracelogs which Message Broker telemetry message, kafka offset, and notificication sequence number passed YANG schema validation or not.

6. YANG Message Broker Consumer - YANG Telemetry Message Schema Validation

With the previously generated YANG schema tree, YANG Telemetry Message schema validation is performed.

6.1 NetGauze Implementation

Below shows the YANG Message Broker message consumption for schema id `527`. It details the topic offset, YANG schema id and message key (currently the YANG-Push publisher node IP address), the message itself and wherever the message was validated successfully or not.

```
root@daisy-playground-rh8 [ /opt/NetGauze-bin ] () # RUST_LOG=kafka_yang_consumer=debug,info ./ka
2026-02-03T10:02:28.021308Z DEBUG kafka_yang_consumer: Message Payload [partition: 0, offset: 960
2026-02-03T10:02:28.662404Z DEBUG kafka_yang_consumer: Message validation PASSED - partition: 0,
```

6.1 NetGauze Impediments

Schema validation only supports validation of first YANG structure, in this case the telemetry-message message structure and doesn't perform andydata validation if it is in such a nested YANG structure. These constrains are with libyang 4.2.2 and has been addressed in libyang 4.9.1 but not yet integrated into NetGauze.

Message Broker Consumer – Cisco Crosswork (1)

- Describes how Cisco Crosswork Ingestion Service organizes and caches YANG modules for a set of YANG-Push Subscriptions with the same YANG schema tree with schema-id.

4. YANG Message Broker Consumer - YANG Schema Collection

With the consumption of YANG Network Telemetry messages the schema id at the top of the YANG schema tree is being learned. The YANG schema and YANG feature metadata with all their related YANG schemas and YANG feature metadatas are being obtained and cached locally.

4.1 Cisco Crosswork Ingestion Service Implementation

The Cisco Crosswork Ingestion Service learns the YANG schema id from the Confluent wire-format header of each consumed Kafka message (`byte[0]=0x00` magic, `byte[1-4]=schema_id`). For each schema id, all related YANG modules are fetched from the Confluent Schema Registry via `GET /schemas/ids/{schemaId}` and cached locally.

In `/opt/cisco/dg/cache/` for each YANG schema id a dedicated folder is dynamically created when the YANG Message Broker Consumer is consuming from the topic. The extracted YANG modules are saved as `{module-name}@{revision}.yang` files.

The resolved schema context is cached in-memory per schema id so that subsequent messages with the same schema id do not require additional Schema Registry lookups.

Message Broker Consumer – Cisco Crosswork (2)

- Describes how Cisco Crosswork Ingestion Service validates Message Broker telemetry messages against YANG schema tree.

6.1 Cisco Crosswork Ingestion Service Implementation

Validation is performed using `YangDataDocumentJsonParser` from yangkit, which implements RFC 7951 (JSON Encoding of Data Modeled with YANG). Both `push-update` (periodic) and `push-change-update` (on-change) message types are supported.

The validator checks:

- **RFC 7951 format compliance** — e.g., `int64` / `uint64` values must be quoted as strings
- **YANG type constraints** — type mismatches are rejected
- **Mandatory leaf enforcement** — missing mandatory leaves produce validation errors

Validation traverses the YANG schema tree hierarchically, starting from each top-level object and walking down through containers, lists, and leaves:

```
Validating field 'ietf-telemetry-message:message'...
   ✓ Found container: data-collection-manifest
   ✓ Found container: network-operator-metadata
   ✓ Found container: payload
   ✓ Found container: telemetry-message-metadata
   ✓ Field 'ietf-telemetry-message:message' is valid.
   ✓ JSON successfully validated against YANG schema
```

For `push-change-update` messages, yang-patch edit values are reconstructed from their XPath target paths and validated individually against the schema tree.

Validation Result:

- **Valid** → message is forwarded for downstream ingestion
- **Invalid** → message is dropped, counter incremented

Below an example of Kafka consumer validation output processing 76 YANG-Push messages with schema id 140:

```
$ make kafka-consumer-validate KAFKA_GROUP=scott-group
[2026-03-06 14:32:15] Starting YANG-validated consumer (group=scott-group, schema-id=140)
[2026-03-06 14:32:16] Downloaded 35 YANG modules for schema-id 140
[2026-03-06 14:32:16] Built YANG schema context: 35 modules loaded
...
[2026-03-06 14:32:18] === Validation Summary ===
[2026-03-06 14:32:18] Total messages:   76
[2026-03-06 14:32:18] Valid messages:    76
[2026-03-06 14:32:18] Invalid messages:  0
[2026-03-06 14:32:18] Dropped messages:  0
```

Message Broker Consumer – Cisco Crosswork (3)

- YANG structure (RFC 8791) support has been merged with PR 41 to yangkit.
- YANG anydata validation (draft-ietf-netmod-yang-anydata-validation) is noted as impediment for next IETF 126 hackathon.
- Describes how Cisco Crosswork Ingestion Service validates Message Broker telemetry messages against YANG schema tree.

6.2 Impediments

The upstream yangkit (v1.5.0) `YangDataDocumentJsonParser` did not support `sx:structure` (RFC 8791) definitions. JSON documents rooted at an `sx:structure` node (e.g., `ietf-notification:notification`) could not be parsed — the parser returned `null` because it only searched the normal YANG data tree.

A fix was contributed via PR #41, which adds a fallback BFS lookup in `YangDataUtil.getSchemaNodeContainerForDocument()` to search `sx:structure` definitions when the normal tree lookup fails. This fix is required for validating YANG-Push telemetry messages that use the `ietf-notification` envelope defined as `sx:structure notification` in draft-ietf-netconf-notif-envelope.

YANG Message Broker

IETF 125 Implementation Status

	NetGauze	Pmacct	Apache Kafka	Ciena Blueplanet	Cisco Crosswork
draft-ietf-nmop-message-broker-telemetry-message	✓	✓			
YANG-Push Receiver	✓	✓			
YANG Schema Retrieval	✓				
YANG Message Broker Producer	✓				
YANG Schema Registry	na	na	✓	na	na
YANG Message Broker Consumer	✓			(✓)	✓
Validate Telemetry Message against YANG Schema	✓			(✓)	✓
draft-ietf-netmod-yang-anydata-validation	✓				
YANG Schema Comparison	na	na	✓	na	na

[NetGauze](#)
[Pmacct](#)

Green marked describes new capability at IETF 125. "P" to partially implemented.

Next Steps from IETF 124 Hackathon…

- Yanglint, misses anydata validation support within YANG structures
- Pmacct corrects collection-timestamp to be typedef date-and-time compliant.
- Pmacct corrects encoding value to include ietf-subscribed-notifications namespace in JSON encoding.
- Pmacct corrects network-operator-metadata/labels/anydata-values anydata statement to be valid I-JSON.
- YANG schema registry supporting YANG features
- NetGauze supporting
  - netconf <get> to obtain YANG library content,
  - YANG validation for YANG-Push notifications,
  - YANG schema registration.

Next Steps for IETF 126 and 127 Hackathon…

- Re-do YANG Message Broker Consumer tests on updated testbed with Ciena Blue Planet UAA Workflow Engine

- Complete IETF YANG module subscription tests with 6Wind VSR

- Compare YANG feature coverage between libyang, yangkit and yangtools

- Support draft-ietf-netmod-yang-anydata-validation in yangkit

- Revert import only modules for augmented-by workaround

- Preserve cached YANG-Push subscription state across receiver reload

- Obtain YANG-Push subscription state from publisher if missing

- Support Netconf session queuing for schema retrieval

- Support YANG-Push Message Broker Topic Naming with dynamic Apache Kafka topic creation API

Thanks to…

- Paolo Lucente – Pmacct
- Qin Wu – Huawei
- Michael Mackey – Huawei
- Scott Huang - Cisco
- Rob Wilton – Cisco
- Dan Voyer – Cisco
- Sivakumar Sundaravadivel – Ciena
- IJsbrand Wijnands – Arrcus
- Irfan Mohammad – Arrcus
- Benoit Claise - Everything OPS
- Wanting Du – Swisscom
- Rafael Julio – Swisscom
- Leonardo Rodoni - Swisscom
- Ahmed Elhassany – Swisscom
- Thomas Graf – Swisscom

Want to know more then join us at…

- NMOP on **Friday 09:00 – 11:00** at the Grand Ballroom 1 for Message Broker, Telemetry Message and Message Key document updates and hackathon related details.

Backup

An Architecture for YANG-Push to Message Broker Integration

Normative References, State and Reviewers

Working Group	Document Name	Document State	Reviews
NETCONF	draft-ietf-netconf-notif-envelope-04	WG Document (passed working group last call) and is looking for a shepherd	OPSDIR Early review by Joe Clarke YANGDOCTORS Early review (of -02) by Jürgen Schönwälder
NETCONF	draft-ietf-netconf-yang-library-augmentedby-17	RFC editor queue	OPSDIR Early review (of -08 and -14) by Sheng Jiang, Xiao Min YANGDOCTORS Early review (of -07 and -13) by Andy Bierman SECDIR review (of -14 and -15) by Rich Salz
NETCONF	draft-ietf-netconf-yang-notifications-versioning-11	Working group last call past- Pending shepherd review.	YANGDOCTORS Early Review due 2025-10-29 OPSDIR review (of -10) by Gabriele Galimberti
NETCONF	draft-ietf-netconf-yp-transport-capabilities-05	Submitted to IESG for Publication	OPSDIR IETF Last Call review by Xiao Min TSVART IETF Last Call review by Tommy Pauly YANGDOCTORS IETF Last Call review by Jan Lindblad OPSDIR Early review (of -03) by Xiao Min YANGDOCTORS Early review (of -01) by Jan Lindblad SECDIR IETF Last Call Review due 2025-10-18
NMOP	draft-ietf-nmop-yang-message-broker-integration-11	WG Document (request for working group last call at IETF 124)	
NMOP	draft-ietf-nmop-message-broker-telemetry-message-04	WG Document (request for working group last call at IETF 125)	YANGDOCTORS Early review (of -02) by Martin Björklund

An Architecture for YANG-Push to Message Broker Integration

Informative References, State and Reviewers

Working Group	Document Name	Document State	Reviews
NETCONF	draft-ietf-netconf-udp-notif-25	AD review received. Pending confirmation.	OPSDIR Early review (of -21) by Tina Tsou YANGDOCTORS Early review (of -20) by Jürgen Schönwälder TSVART Early review (of -11) by Michael Tüxen TSVART Early Review due 2025-10-23
NETCONF	draft-ietf-netconf-udp-client-server-10	Publication Requested	OPSDIR Early review (of -07) by Ran Chen TSVART Early review (of -06) by Joerg Ott YANGDOCTORS Early review (of -06) by Jürgen Schönwälder GENART review (of -09) by Elwin Davies DNSDIR review (of -09) by Peter Van Dijk
NETCONF	draft-ietf-netconf-distributed-notif-18	AD review received. Pending confirmation.	OPSDIR Early review (of -14) by Jürgen Schönwälder YANGDOCTORS Early review (of -13 and -17) by Martin Björklund
NETMOD	draft-ietf-nmop-yang-anydata-validation-00	Working group adopted at IETF 124.	