

An Architecture for YANG-Push to Apache Kafka **Integration**

draft-netana-nmop-yang-kafka-integration-01

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

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

15. March 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.

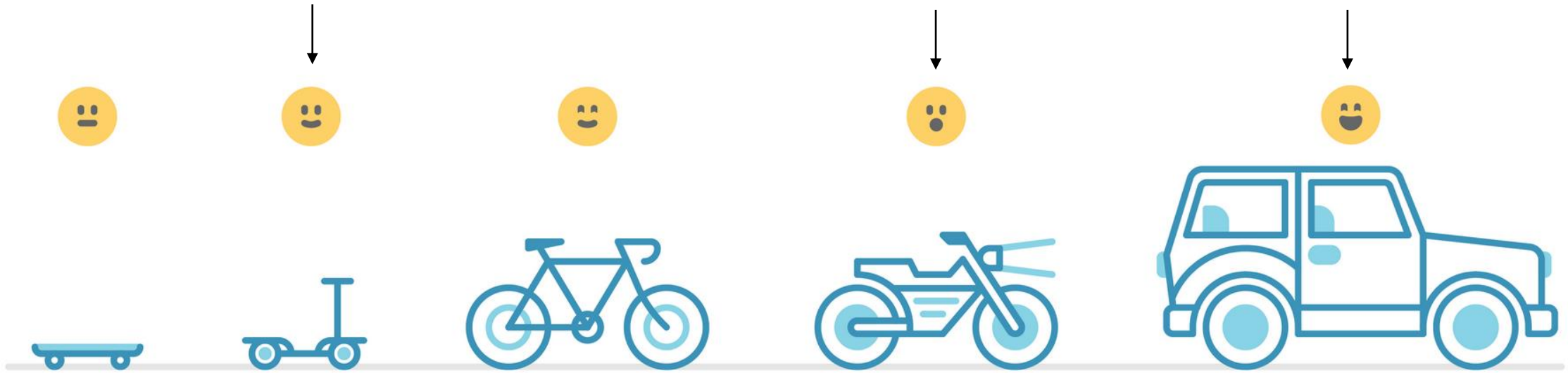
Today, subscribing to a YANG datastore, publishing a YANG modeled notifications message from the network and viewing the data in a time series database, **manual labor is needed to perform data transformation** to make a message broker and its data processing components with YANG notifications interoperable.

« Even though YANG is intend to ease the handling of data, **this promise has not yet been fulfilled** for Network Telemetry [RFC 9232](#) »

Network
Vendor/Operator

IETF

Data
Industry

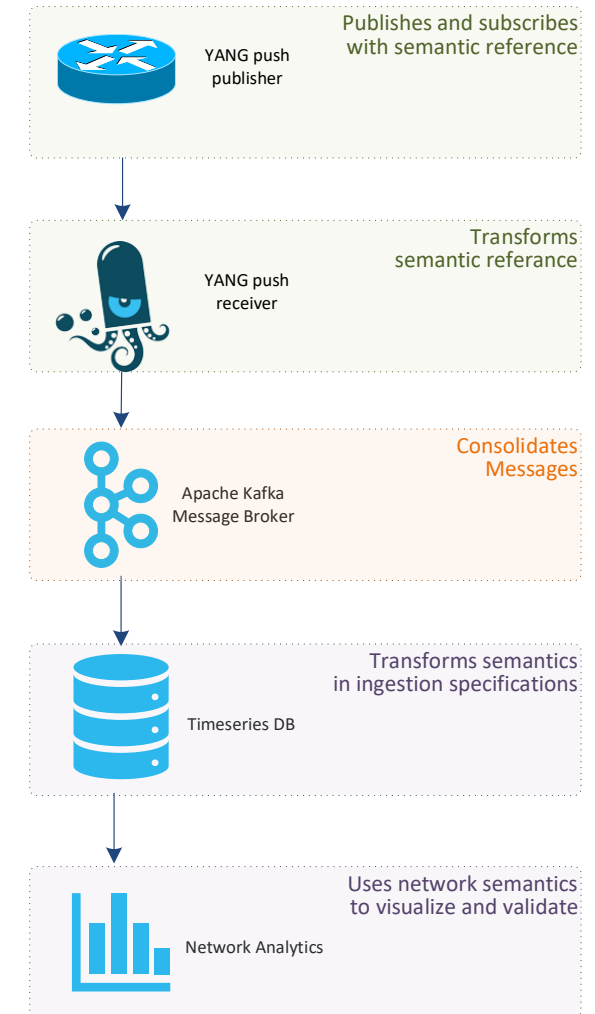


State of the Union
From data **mess** to data **mesh**

From YANG-Push to Network Analytics

Aiming for an automated 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

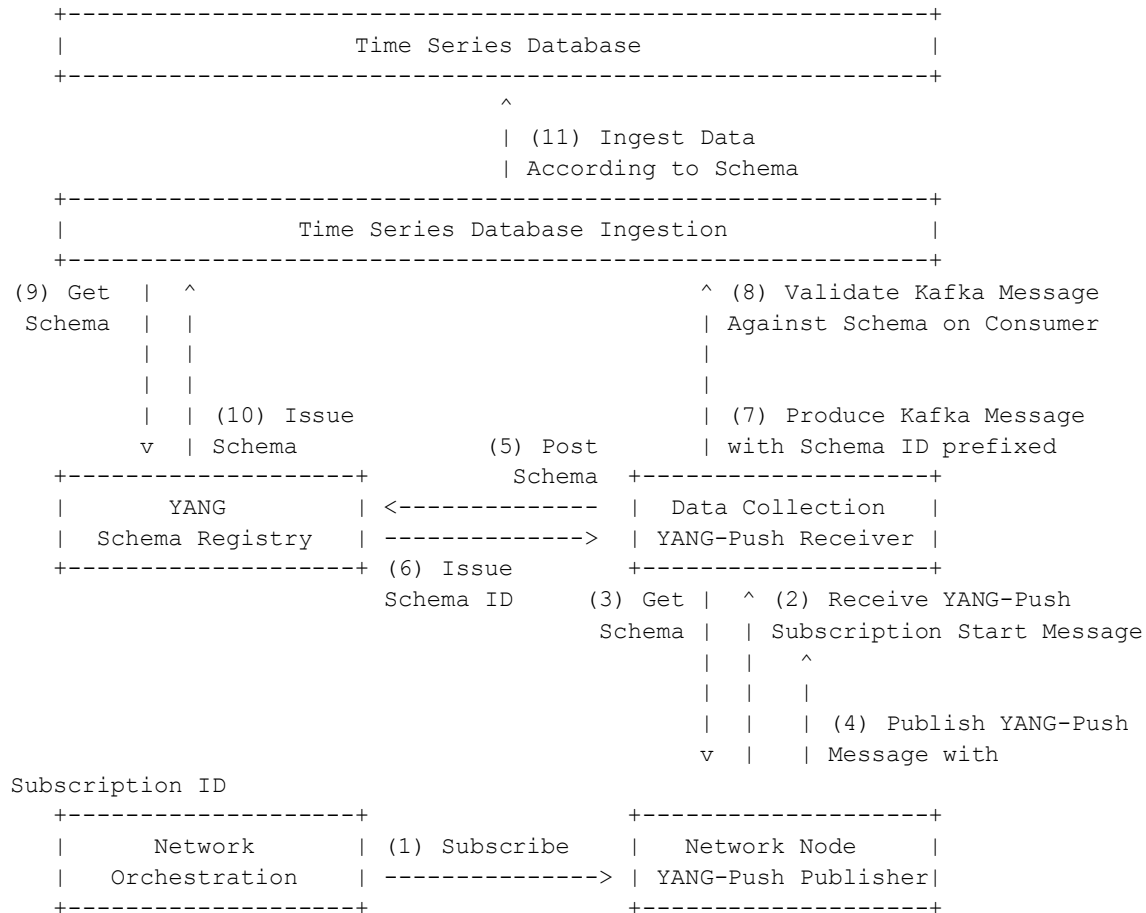


Figure 1: End to End Workflow

- **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 Kafka 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 Apache Kafka, obtains schema tree from YANG schema registry, validates YANG notifications against schema and uses schema to populate into database table.

Netconf Notifications

Define YANG module

```
module: ietf-notification

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

<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2023-02-04T16:30:11.22Z</eventTime>
  <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <id>1011</id>
    <datastore-contents>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-
interfaces">
        <interface>
          <name>eth0</name>
          <oper-status>up</oper-status>
        </interface>
      </interfaces>
    </datastore-contents>
  </push-update>
</notification>
```

- **YANG model for NETCONF Event Notifications**, [draft-ahuang-netconf-notif-yang](#), updates RFC 5277 by defining the schema as a YANG module.
- This enables YANG-push to define semantics for the entire YANG push message and use other encodings than XML such as YANG-JSON RFC 7951 or YANG-CBOR RFC 9264.

Subscription State Change Notifications

RFC 8641 Extensions

```
{
  "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
      }
    }
  }
}
```

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

- **Support of Versioning in YANG Notifications Subscription**, [draft-ietf-netconf-yang-notifications-versioning](#), adds the ability to subscribe to a specific revision or latest-compatible-semversion. Extends the YANG-Push Subscription State Change Notifications so that the receiver learns on top of xpath and the sub-tree filter also the YANG module name, revision and revision-label.
- **Support of Hostname and Sequencing in YANG Notifications**, [draft-tgraf-netconf-notif-sequencing](#), extends the NETCONF notification defined in RFC5277 with sysName, publisherId and sequenceNumber.

Push-Update and Push-Change-Update Notifications

RFC 8641 Extensions

```
{
  "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-yang-push-netobs-timestamping:observation-time": "2023-03-25T08:30:11.22Z",
      "datastore-contents": {
        "ietf-interfaces:interfaces": [
          {
            "interface": {
              "name": "eth0",
              "type": "iana-if-type:ethernetCsmacd",
              "oper-status": "up",
              "mtu": 1500
            }
          }
        ]
      }
    }
  }
}
```

- **Support of Hostname and Sequencing in YANG Notifications**, [draft-tgraf-netconf-notif-sequencing](#), , extends the NETCONF notification defined in RFC5277 with sysName, publisherId and sequenceNumber.
- **Support of Network Observation Timestamping in YANG Notifications**, [draft-tgraf-netconf-yang-push-observation-time](#), extends YANG-Push push-update notifications with observation-time and state-changed-observation-time.

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

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
```

- **Augmented-by Addition into the IETF-YANG-Library, [draft-lincla-netconf-yang-library-augmentation](#), enables that augmented-by YANG modules can now be discovered in YANG Library [RFC 8525](#).**

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"
          }
        ]
      }
    }
  }
}
```

- **Validating anydata in YANG Library context,** [draft-aelhassany-anydata-validation](#), enables that anydata modeled nodes can be validated with YANG Library [RFC 8525](#).

[RFC 7950](#)

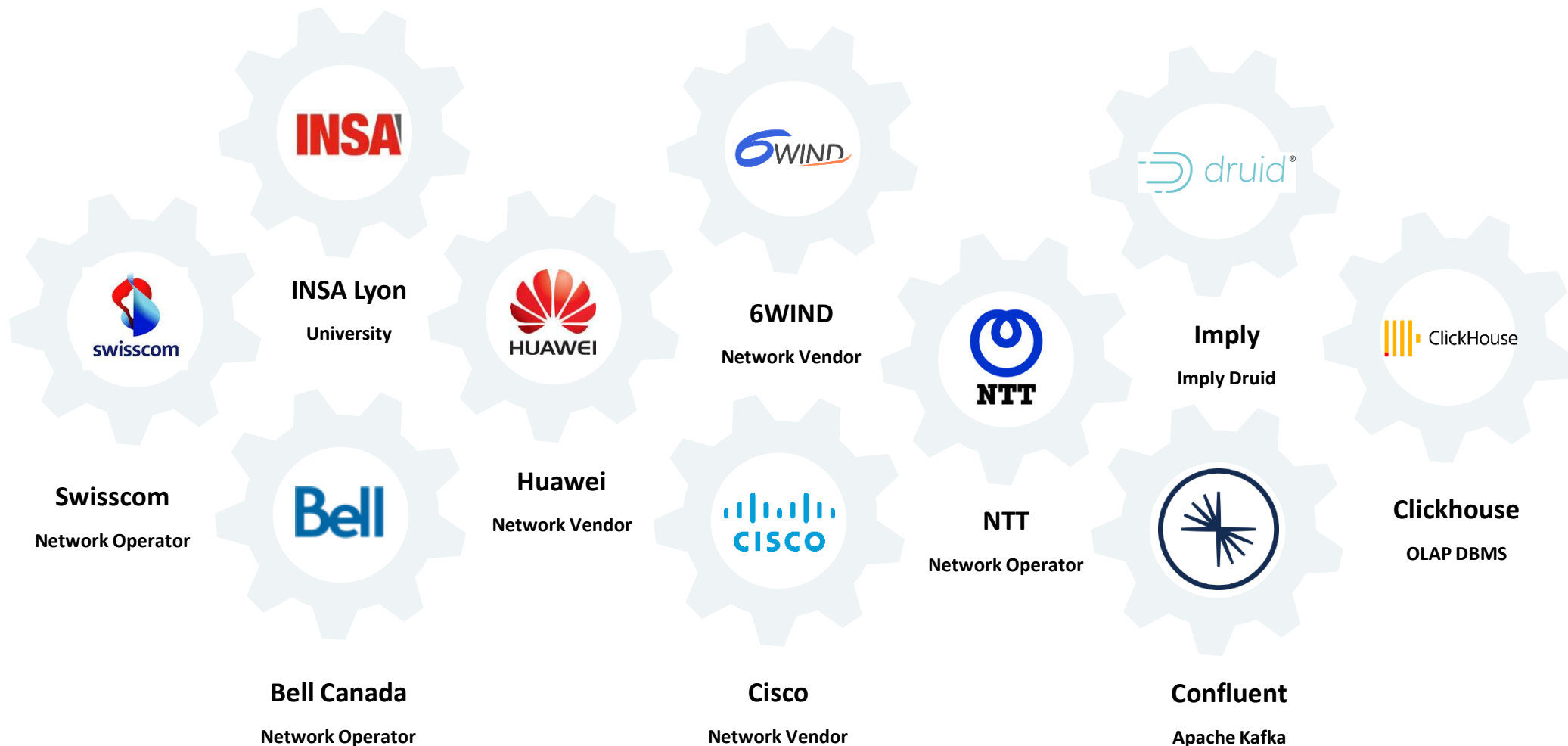
7.10. The "anydata" Statement

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.**

Industry Colaboration

On YANG Push to Apache Kafka integration



Milestones

IETF 115 - 119

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.

Open Points

Currently ongoing

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: 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. How a YANG-Push receiver **validates the content of anydata nodes** needs to be addressed.

Open Point 3: 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 4: 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.

An Architecture for YANG-Push to Apache Kafka Integration

Status, Summary and Next steps

Status

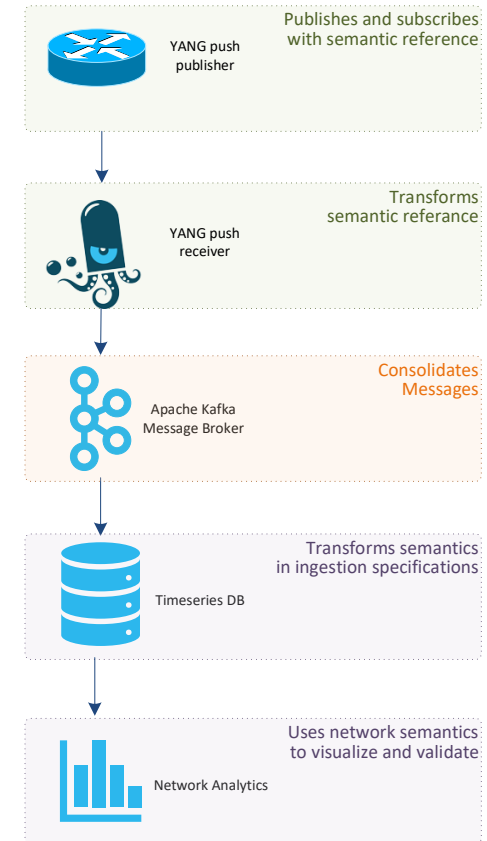
- IETF Community requested to have an architecture overview document describing motivation, architecture and dependencies on YANG-Push, YANG Library and YANG enhancements.

Summary

- Describes motivation, architecture and dependencies on YANG-Push, YANG Library and YANG enhancements.
- Describes project milestones with status and open points which are either addressed in this IETF or are going to be addressed next.
- **Do you realize the benefit of having YANG-Push natively integrated into Apache Kafka? -> What are your thoughts and comments?**
- **Do you agree on the shortcomings in YANG-Push, YANG Library and YANG and how they are addressed? -> What are your thoughts and comments?**

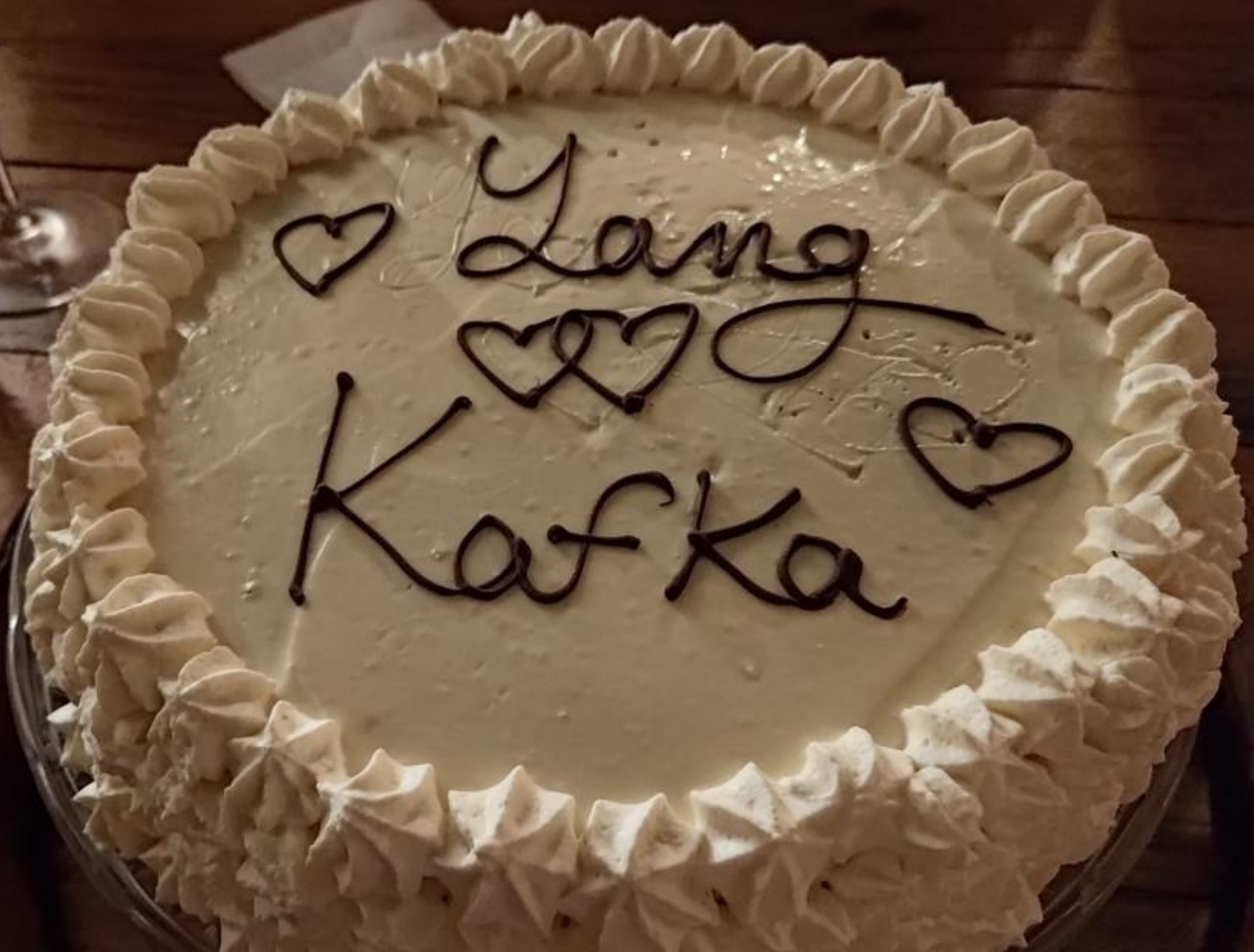
Next Steps

- **-> We request NMOP working group adoption.**



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

15. March 2024



♥ Yang ♥♥♥
Kafka ♥