Addressing Network Operator Challenges in YANG push Data Mesh Integration

thomas.graf@swisscom.com
alex.huang-feng@insa-lyon.fr

18. March 2023

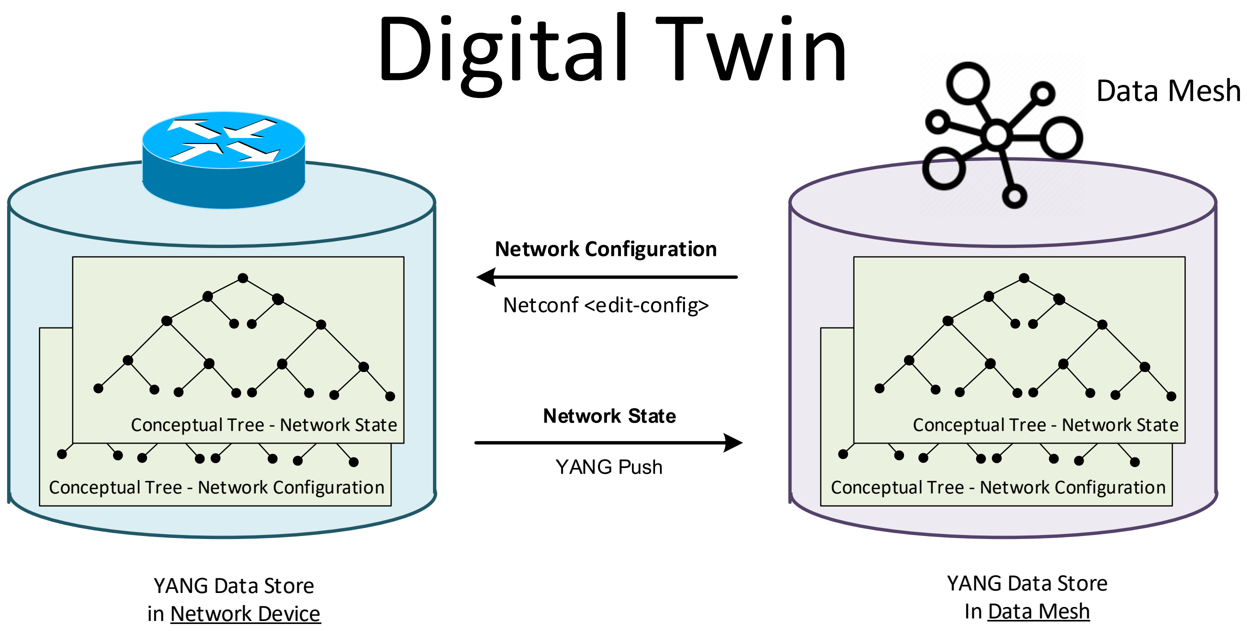
Evolving YANG Push

Missing puzzle pieces

YANG Push	Today at Network Operators	Today at IETF
Transport Protocol	Many and non-standard	netconf-https-notif and netconf-udp-notif
Encoding	JSON widely adopted. Propriety protobuf in various variants. CBOR not implemented yet.	JSON and XML in RFC8040, CBOR in RFC9254
Subscription	Non-standard, periodical widely adopted. On-change sparse.	RFC8639 and RFC8641
Metadata	Non-standard. Partially among message content.	netconf-yang-notifications-versioning, draft-tgraf-netconf-notif-sequencing, draft-tgraf-yang-push-observation-time, draft-claise-opsawg-collected-data-manifest, draft-claise-netconf-metadata-for-collection
Versioning	Neither covered in subscription nor in publishing.	netmod-yang-module-versioning
YANG module	Non-standard widely adopted. IETF coverage non-existent.	Many RFC's defined

YANG datastores enabling Closed Loop Operation

Automated data onboarding with bounded context



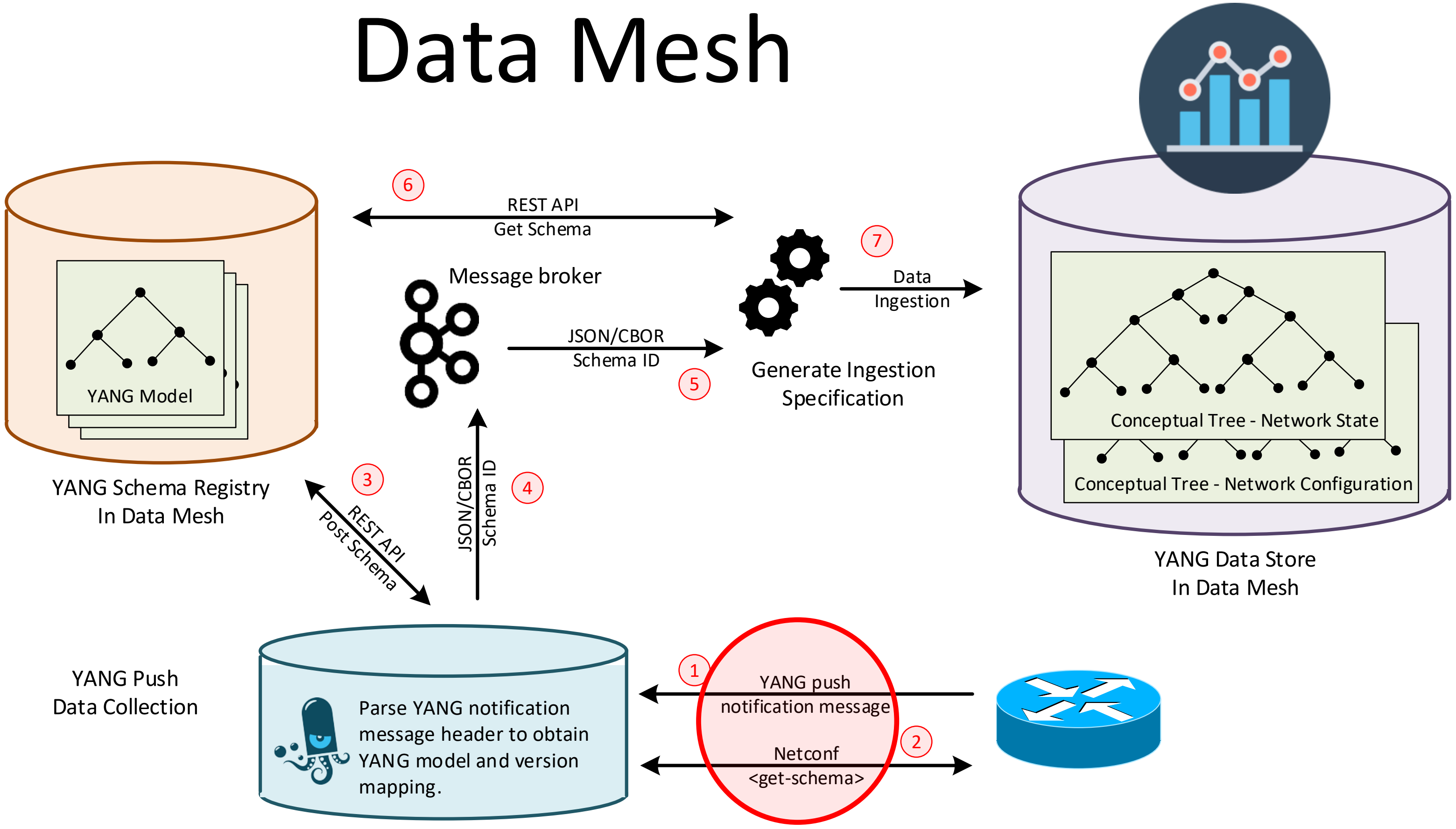
YANG is a data modelling language which will not only transform how we managed our networks; it will transform also how we manage our services.

17 industry leading colleagues from 4 network operators, 2 network and 3 analytics providers, and 3 universities **commit on a project to integrate YANG and CBOR into data mesh. IETF 116 public side meeting on Wednesday March 29th 12:00 – 12:45.**

Automated networks can only run with a common data model. A digital twin YANG data store enables a comparison between intent and reality. Schema preservation enables closed loop operation. **Closed Loop is like an autopilot on an airplane.** We need to understand what the flight envelope is to keep the airplane within. Without, we crash.

When Big Data and Network becomes one

Marrying two messaging protocols



- **Data Mesh** is a big data architecture where different domains can exchange data with a **bounded context and SLO's** are defined in Data Products. **Same principle as in networks.**
- **Semantics** are needed to describe the data. **A gauge32 is not the same as counter32.** Values can increase or decrease. Needs monotonic increasing counter normalization or not.
- **Versioning** is needed to not only understand that the semantic has changed, but also wherever the new semantic is backward compatible or not. **Preventing to break the data processing pipeline.**
- **Hostname, sequence numbers and observation timestamping** are needed to **measure loss and delay for SLO's.**
- **YANG push as defined in RFC8641 is missing** hostname, sequence numbers, observation timestamping and versioning. draft-ahuang-netconf-notif-yang, draft-tgraf-netconf-notif-sequencing, draft-tgraf-yang-push-observation-time and draft-tgraf-netconf-yang-notifications-versioning addresses this.

Define YANG module for Netconf Notifications

Closing the semantic gap

```
module: ietf-notification

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

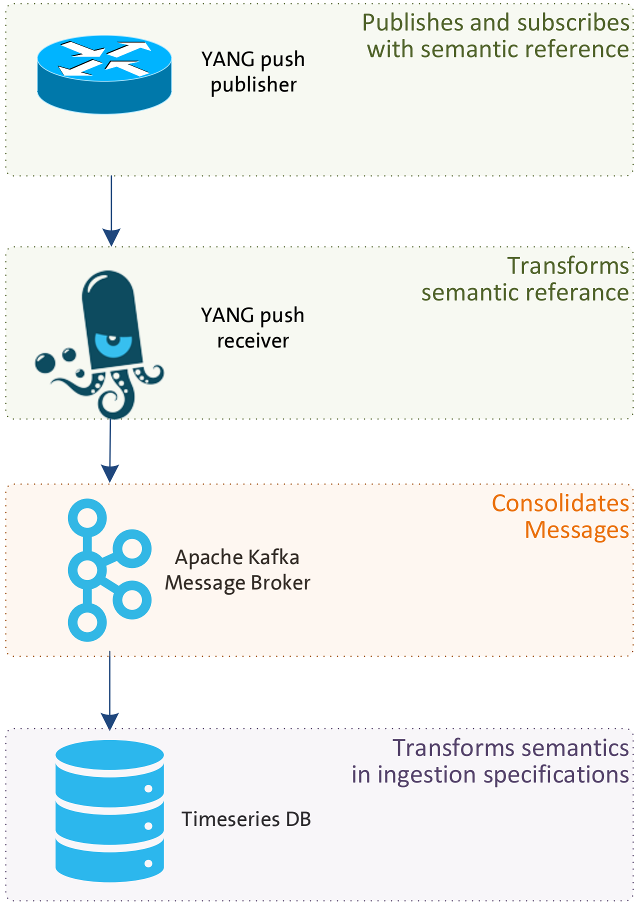
```
<notification 
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
 <eventTime>2023-02-04T16:30:11.22Z</eventTime>
<push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
   <id>1011</id>
   <datastore-contents>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-
interfaces">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

- **With RFC 5277 the XML schema for NETCONF event notification was defined.**
- With **draft-ahuang-netconf-notif-yang** the schema is also defined as a YANG module. **Enabling now to define semantics for the entire YANG push message.**

Define YANG module for Netconf Notifications

Status

- The yang module prefix has changed to “inotif” to be more explicit.
- The namespace is changed to the one used in RFC5277 : urn:ietf:params:xml:ns:netconf:notification:1.0
- In IANA section, instead of asking for a new URI, we ask IANA to add this document as a reference to the URI from RFC5277
- **Requesting NETCONG working group adoption.**


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

Extend Streaming Update Notifications with Hostname and Sequencing

For push-update and push-change-update

```
module: ietf-notification-sequencing

  augment-structure /inotif:notification:
    +-- sysName             inet:host
    +-- sequenceNumber      yang:counter32
```

```
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
 <eventTime>2023-02-04T16:30:11.22Z</eventTime>
 <sysName xmlns="urn:ietf:params:xml:ns:yang:ietf-notification-
sequencing">
  example-router
 </sysName>
 <sequenceNumber xmlns="urn:ietf:params:xml:ns:yang:ietf-
notification-sequencing">
  187653
 </sequenceNumber>
 <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>
```

- When the **NETCONF event notification message is forwarded from the YANG push receiver to another system**, such as a messaging system or a time series database where the message is 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 RFC1213 from where the message was published from.
  - **sequenceNumber:** Generates a unique sequence number as described in RFC9187 for each published message.

Extend Streaming Update Notifications with Observation Timestamping

For push-update and push-change-update

```
module: ietf-yang-push-netobs-timestamping

    augment /yp:push-update:
      +--ro observation-time?   yang:date-and-time
    augment /yp:push-change-update:
      +--ro state-changed-observation-time?   yang:date-and-time
```

```
{
   "ietf-notification:notification": {
      "eventTime": "2023-02-04T16:30:11.22Z",
      "sysName": "example-router",
      "sequenceNumber": 187653,
      "ietf-yang-push:push-update": {
         "id": 1011,
         "observation-time": "2023-02-04T16:30:09.44Z",
         "datastore-xpath-filter": "ietf-interfaces:interfaces",
         "datastore-contents": {
            "ietf-interfaces:interface": {
               "name": {
                  "eth0": {
                     "oper-status": "up"
                  }
                }
            }
         }
      }
   }
}
```

- **To correlate network data** among different Network Telemetry planes as described in Section 3.1 of RFC9232 or among different YANG push subscription types defined in Section 3.1 of RFC8641, **network observation timestamping is needed to understand the timely relationship among these different planes and YANG push subscription types.**
- **draft-tgraf-yang-push-observation-time** extends the YANG push streaming update notification defined in RFC8641 with:
  - **observation-time:** Describes the measurement observation time for the "push-update" notification in a "periodical" subscription.
  - **state-changed-observation-time:** Describes in the "push-change-update" notification in an "on-change" subscription the time when the network state change was observed after the subscription was initially established. In case of an "on-change sync on start" subscription it describes the time when the network state change was observed before the subscription was established.

Extend Datastore Selection and Subscription State Change Notifications with revision and revision-label

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

  augment /sn:establish-subscription/sn:input/sn:target:
    +--rw revision?          rev:revision-date-or-label
    +-- revision-label?    ysver:version
  augment /sn:modify-subscription/sn:input/sn:target:
    +--rw revision?          rev:revision-date-or-label
    +-- revision-label?    ysver:version
  augment /sn:subscription-started/sn:target:
    +--ro revision           rev:revision-date-or-label
    +-- revision-label?    ysver:version
  augment /sn:subscription-modified/sn:target:
    +--ro revision           rev:revision-date-or-label
    +-- revision-label?    ysver:version
  augment /sn:subscriptions/sn:subscription/sn:target:
    +--ro revision           rev:revision-date-or-label
    +--rw revision-label?    ysver:version
```

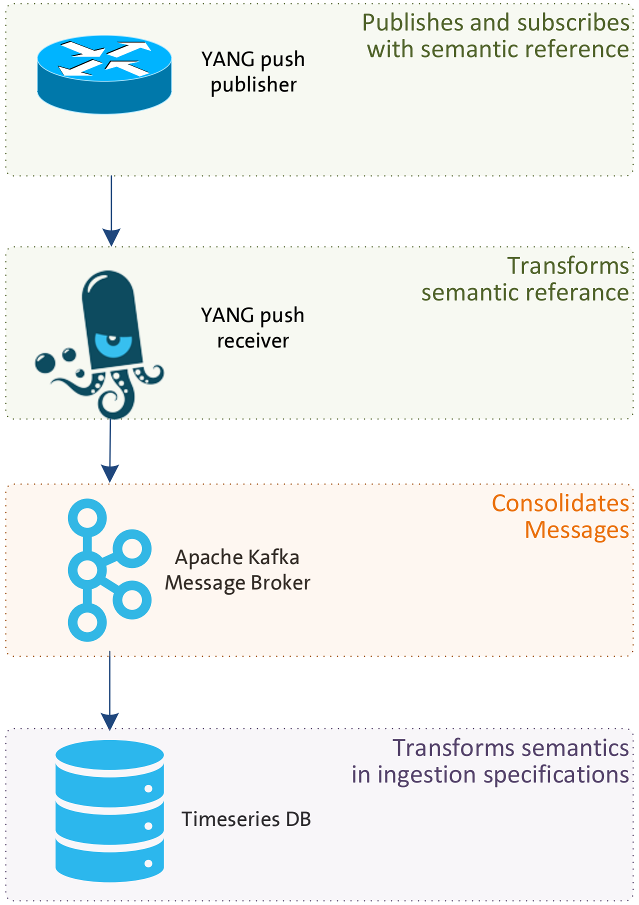
```
{
  "ietf-restconf:notification" : {
    "eventTime": "2023-01-03T10:00:00Z",
    "ietf-subscribed-notifications:subscription-modified": {
      "id": 101,
      "revision": "2014-05-08",
      "revision-label": "1.0.0",
      "stream-xpath-filter": "/ietf-interfaces:interfaces",
      "stream": {
         "ietf-netconf-subscribed-notifications" : "NETCONF"
      }
    }
  }
}
```

- **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 messages are being published for configured subscription.
- **draft-tgraf-netconf-yang-notifications-versioning** extends the YANG push subscription and publishing mechanism defined in RFC8641:
  - **By adding the ability to subscribe to a specific revision** or latest-compatible-semversion.
  - **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 revision and revision-label.

Extend Datastore Selection and Subscription State Change Notifications with revision and revision-label

Status

- **Moved revision and revision-label from push-update to subscription-start and subscription-modified Subscription State change notification** message to reduce message overhead as per suggestion from Jason Stern and Rob Wilton.
- **YANG full tree view added in section 4.1.2**, added descriptions and resolved some issues in the YANG module raised by the YANG validation.
- Fengchong and Rob questioned that revision and revision-label for the subscribed xpath, sub-tree, might be not enough to include all use cases. The authors agree that **when the subcribed xpath revision did not change after software upgrade, but the included yang modules did,** this needs to be covered as well. **pkg-name and pkg-version from draft-ietf-netmod-yang-packages** might be the best answer.

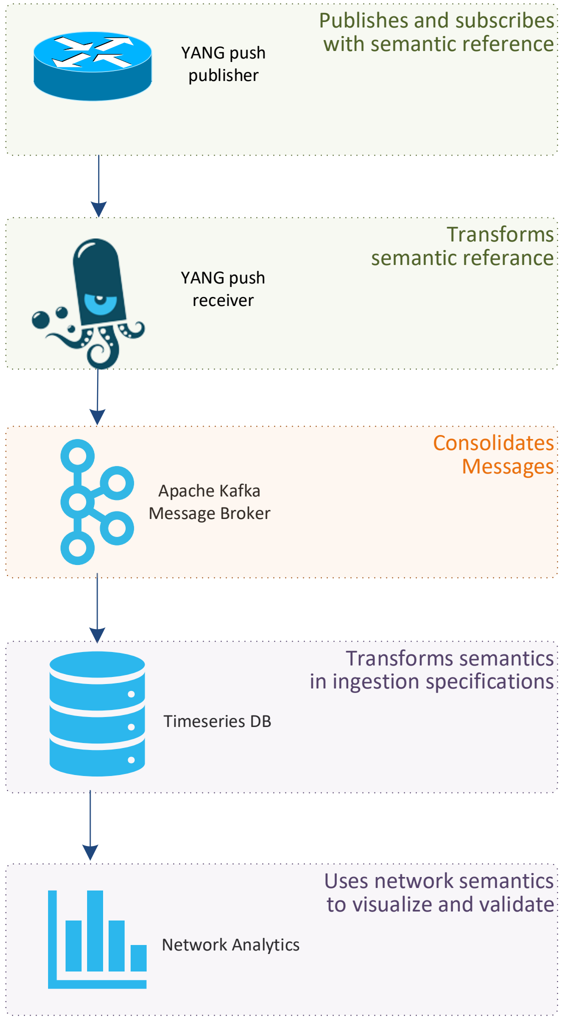

YANG push publisher
Publishes and subscribes with semantic reference
YANG push receiver
Transforms semantic referance
Apache Kafka Message Broker
Consolidates Messages
Timeseries DB
Transforms semantics in ingestion specifications

From YANG push to Analytics

Next steps

- **Do you realize the gaps and how it could be resolved?**
  - By defining a YANG module for NETCONF notification and adding hostname, sequence number, observation time, revision and revision-label into YANG push-update and Subscription State Change notification messages an **automated data processing pipeline** which starts with YANG push, consolidates at Data Mesh and ends at Network Analytics would become at reach.
- **-> What are your thoughts and comments?**
- **-> Interested to learn more? Join the IETF 116 public side meeting on Wednesday March29th 12:00-12:45 in room G301 or look at the project page:**

  https://github.com/graf3net/draft-daisy-kafka-yang-integration/blob/main/draft-daisy-kafka-yang-integration-03.md



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

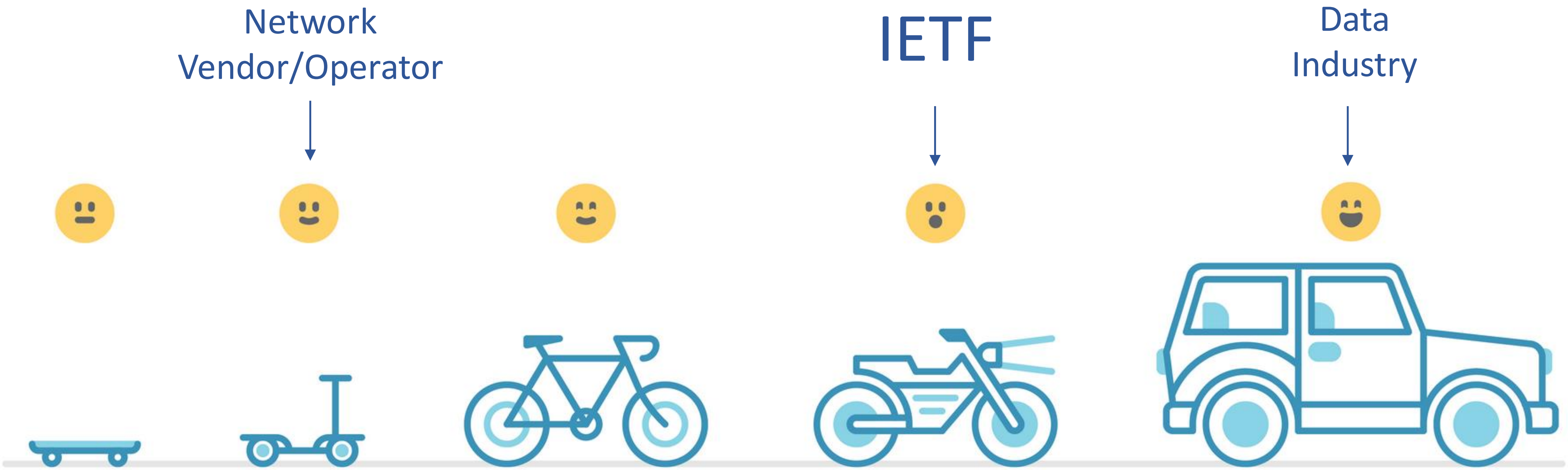
Network Analytics

Uses network semantics to visualize and validate

alex.huang-feng@insa-lyon.fr, pierre.francois@insa-lyon.fr
thomas.graf@swisscom.com
benoit.claise@huawei.com, jean.quilbeuf@huawei.com

18. March 2023

Backup



Network Vendor/Operator

IETF

Data Industry

State of the Union

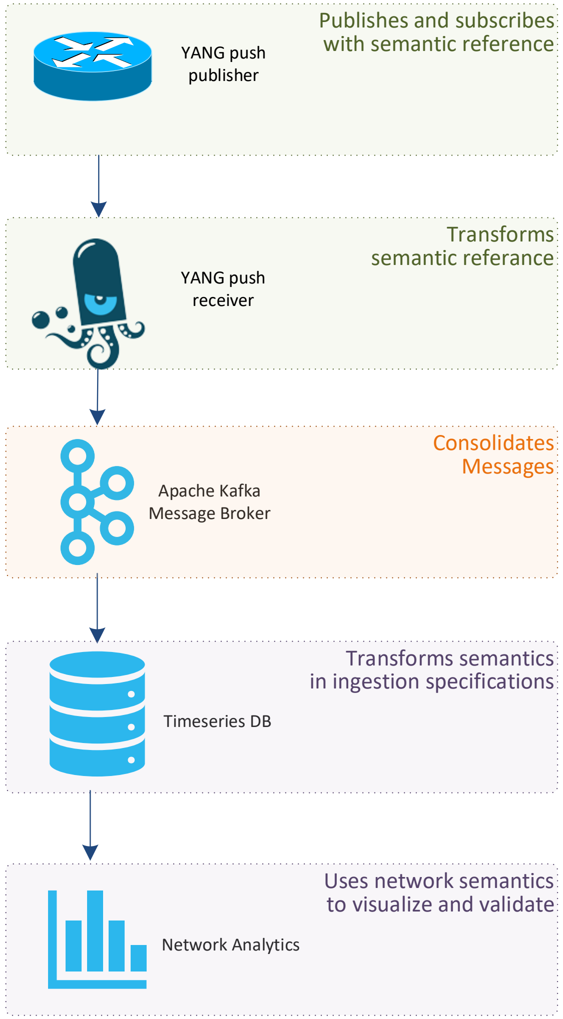
From data mess to data mesh

From YANG push to 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 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

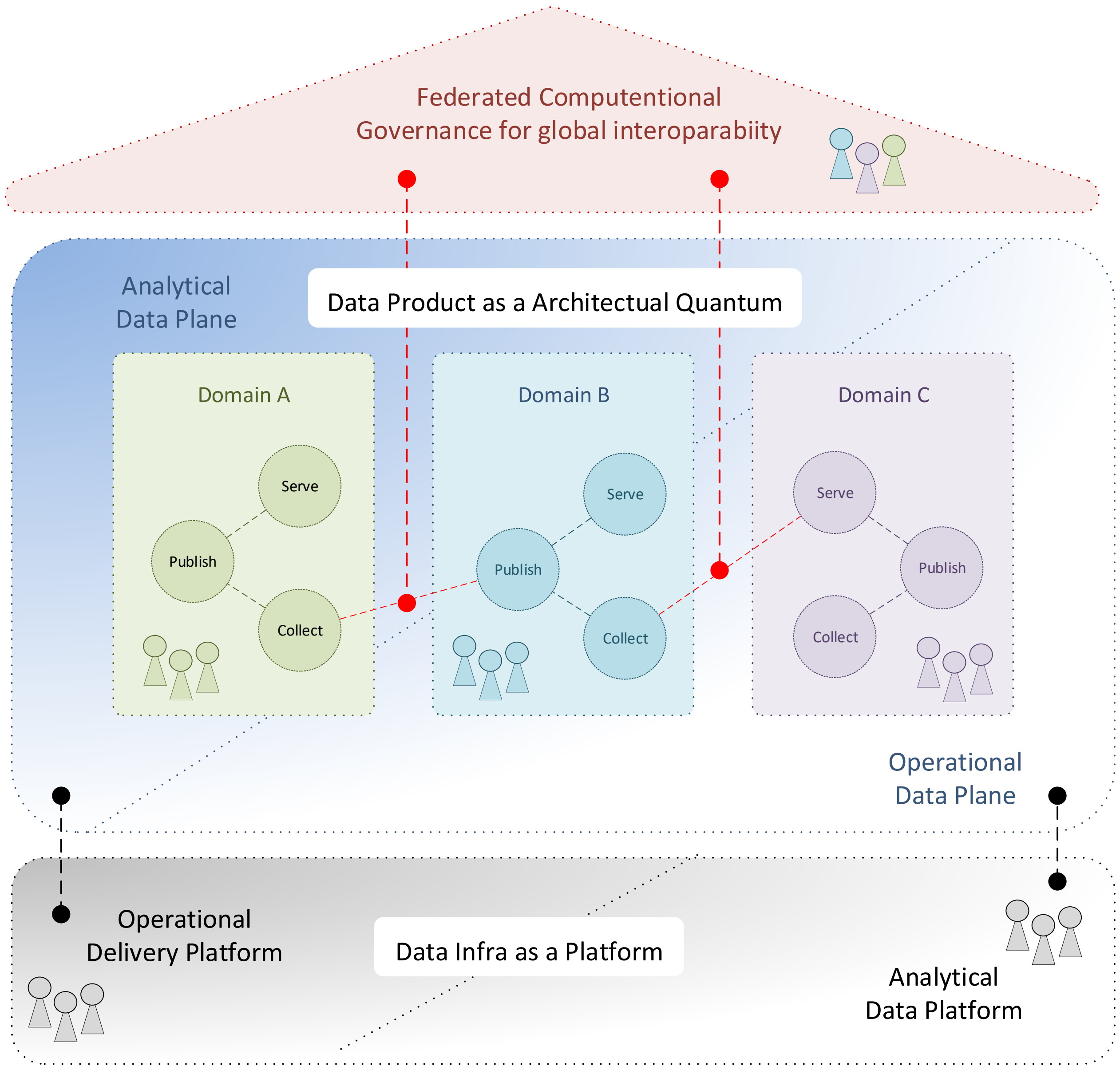
Network Analytics

Uses network semantics to visualize and validate

Evolving Big Data Architecture

Domain oriented, like networks

1st Generation	**Proprietary** Enterprise Data Warehouse
2nd Generation	**Data lake** Big data ecosystem
3rd Generation current	**Kappa** Adds streaming for real-time data
4th Generation next-step	**Data Mesh** Distributed and organized in domains.

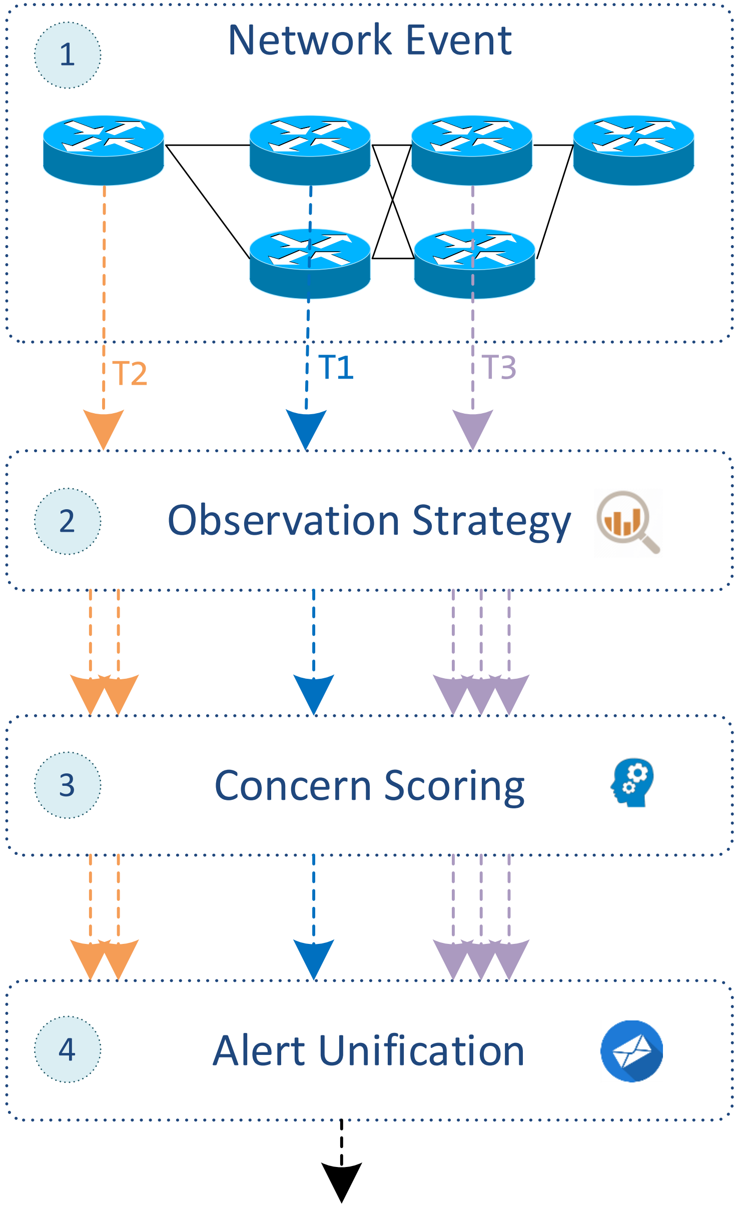


From Principles to Logical Architecture

From Network to Alert Event

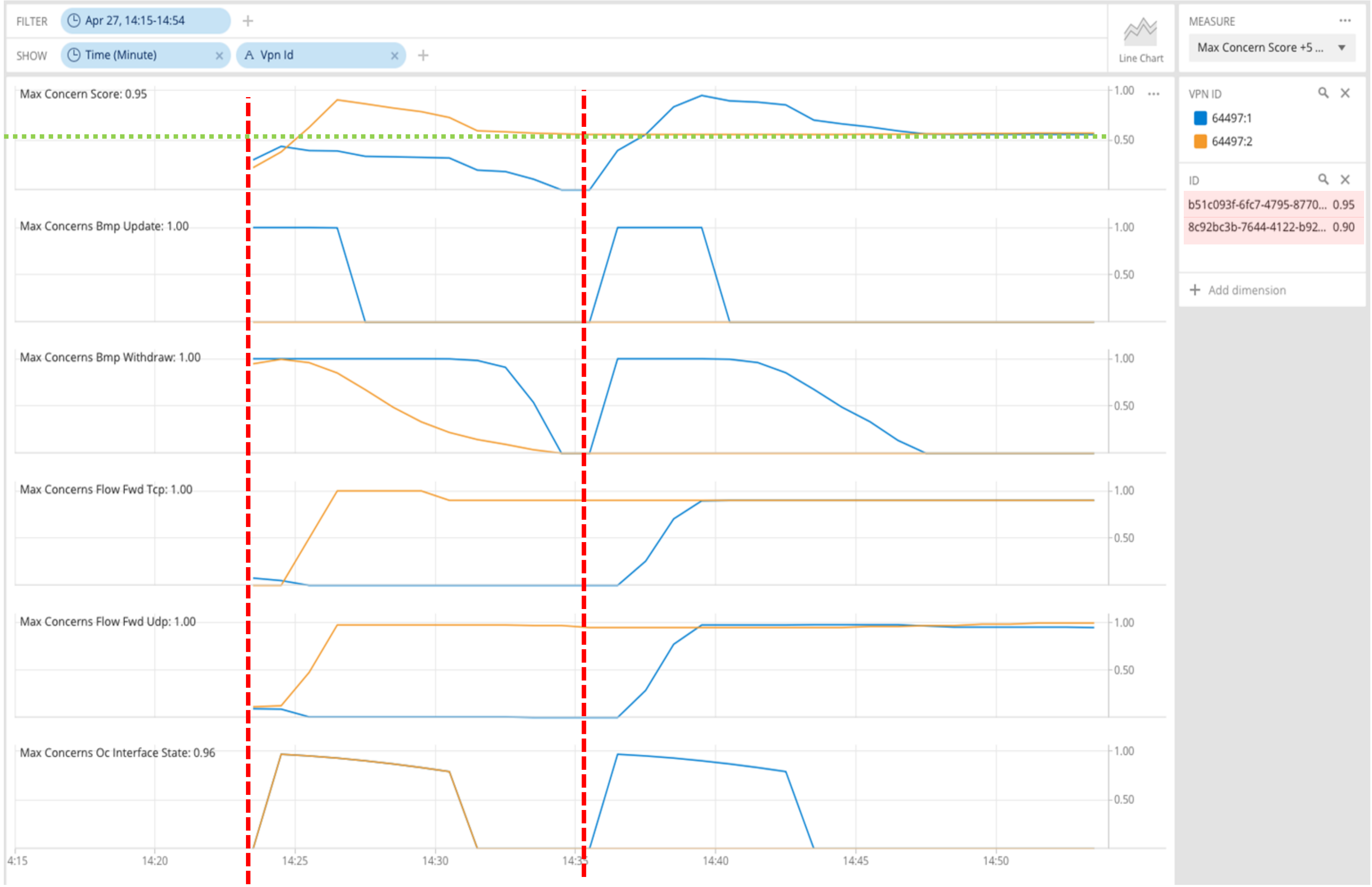
Observe multiple perspectives at different times

1. **A single link down** result in multiple device topology, control-plane and forwarding-plane events being exposed at different times.
2. **Determine** which interfaces and BGP peerings are being used first and then observe state. **Observe** BGP withdrawals and updates, traffic drop spikes and missing traffic. Generate multiple concerns.
3. **Calculate** for each observation a concern score between 0 and 1. **The higher, the more probable** the changes impacted forwarding.
4. **Unify** several concerns for one VPN connectivity service to one alert identifier.



L3 VPN Network Anomaly Detection

Verify operational changes automatically



FILTER: Apr 27, 14:15-14:54

SHOW: Time (Minute), Vpn Id

Line Chart

MEASURE: Max Concern Score +5 ...

VPN ID

🟦 64497:1

🟧 64497:2

ID	
b51c093f-6fc7-4795-8770...	0.95
8c92bc3b-7644-4122-b92...	0.90

Add dimension
Max Concern Score: 0.95
Max Concerns Bmp Update: 1.00
Max Concerns Bmp Withdraw: 1.00
Max Concerns Flow Fwd Tcp: 1.00
Max Concerns Flow Fwd Udp: 1.00
Max Concerns Oc Interface State: 0.96
1.00
0.50
4:15
14:20
14:25
14:30
14:3
14:40
14:45
14:50

Analytical Perspectives

Monitors the network service and wherever it is congested or not.

- BGP updates and withdrawals.
- UDP vs. TCP missing traffic.
- Interface state changes.

Network Events

1. VPN orange lost connectivity. VPN blue lost redundancy.
2. VPN blue lost connectivity.

Key Point

- AI/ML **requires** network intent and network modelled data to deliver dependable results.