

# An Architecture for a **Network Anomaly Detection** Framework

draft-netana-nmop-network-anomaly-architecture-00

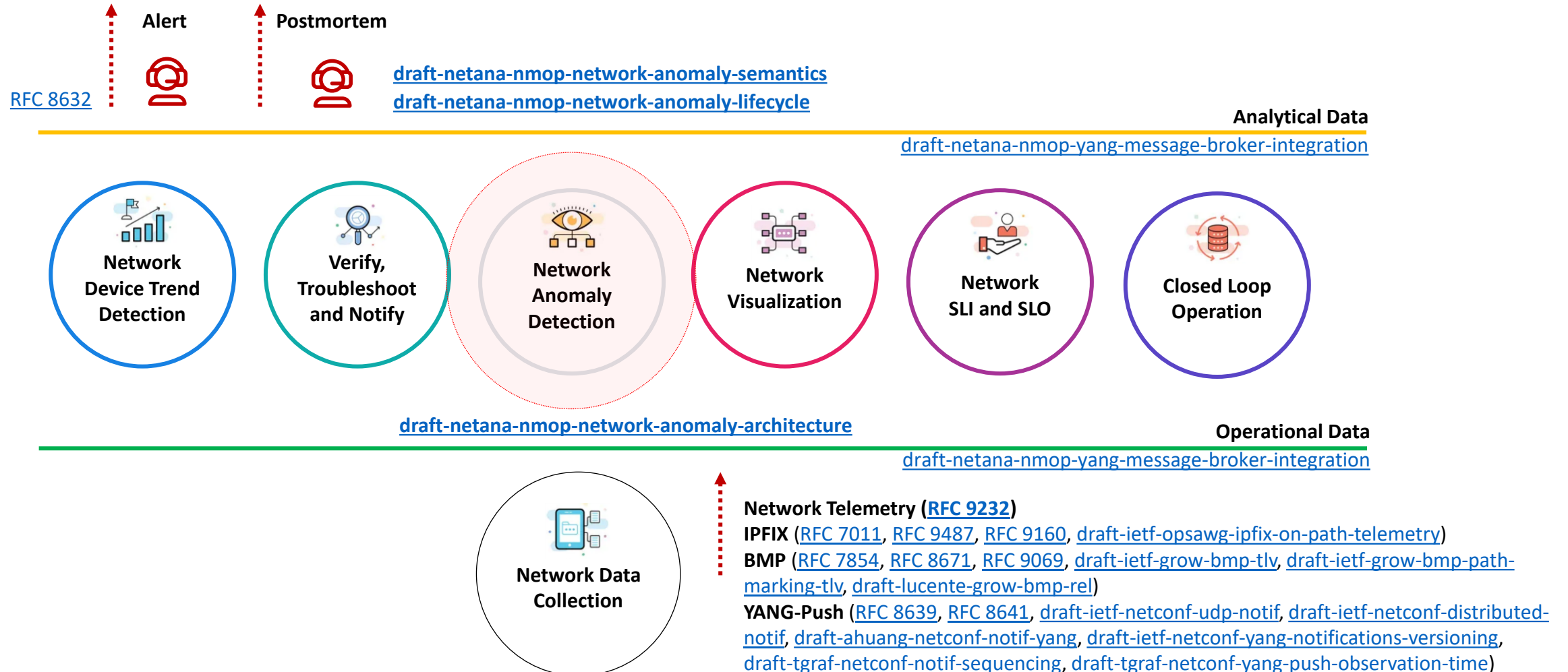
Motivation and architecture of a Network Anomaly Detection Framework  
and the relationship to other documents describing  
network symptom semantics and network incident lifecycle

wanting.du@swisscom.com  
pierre.francois@insa-lyon.fr  
thomas.graf@swisscom.com  
vincenzo.riccobene@huawei-partners.com

15. July 2024

# Data Mesh organizes Data in Organizations

Enables Network Analytics use cases

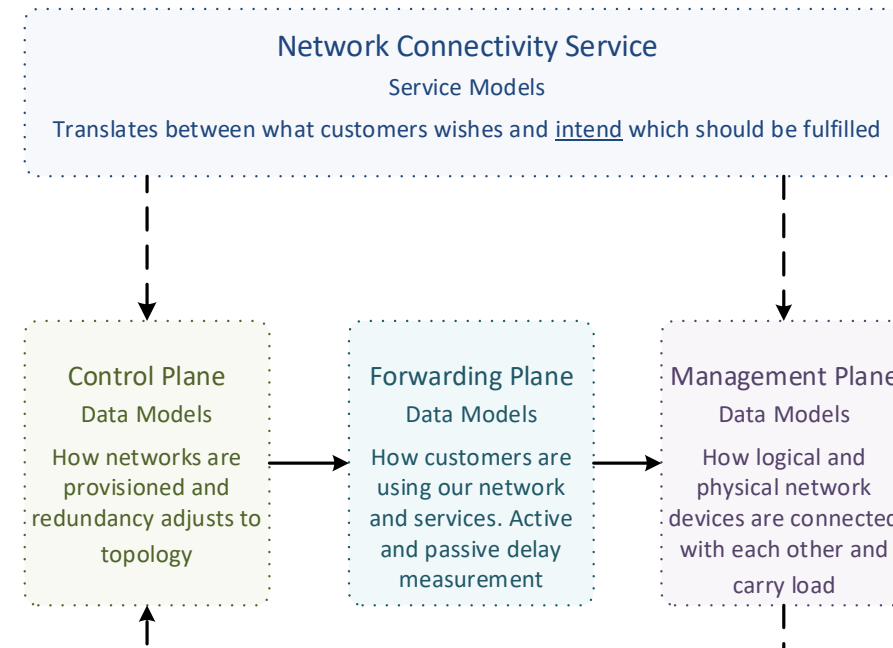
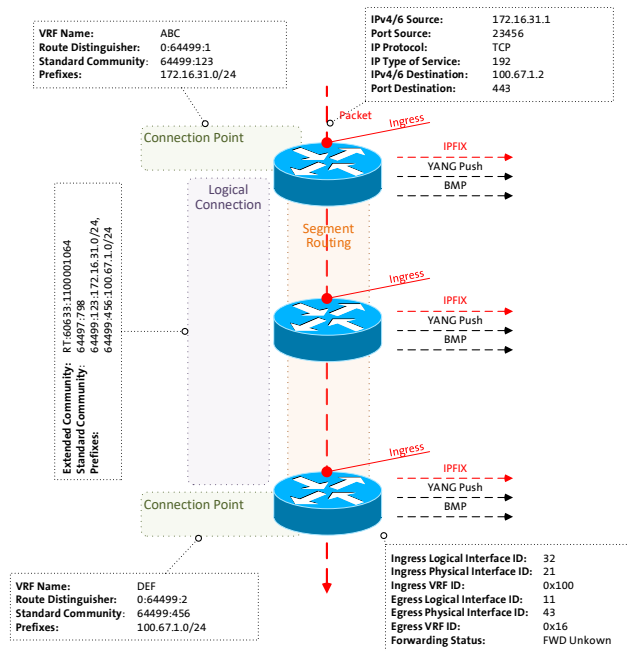


# What to monitor

Which operational metrics are collected

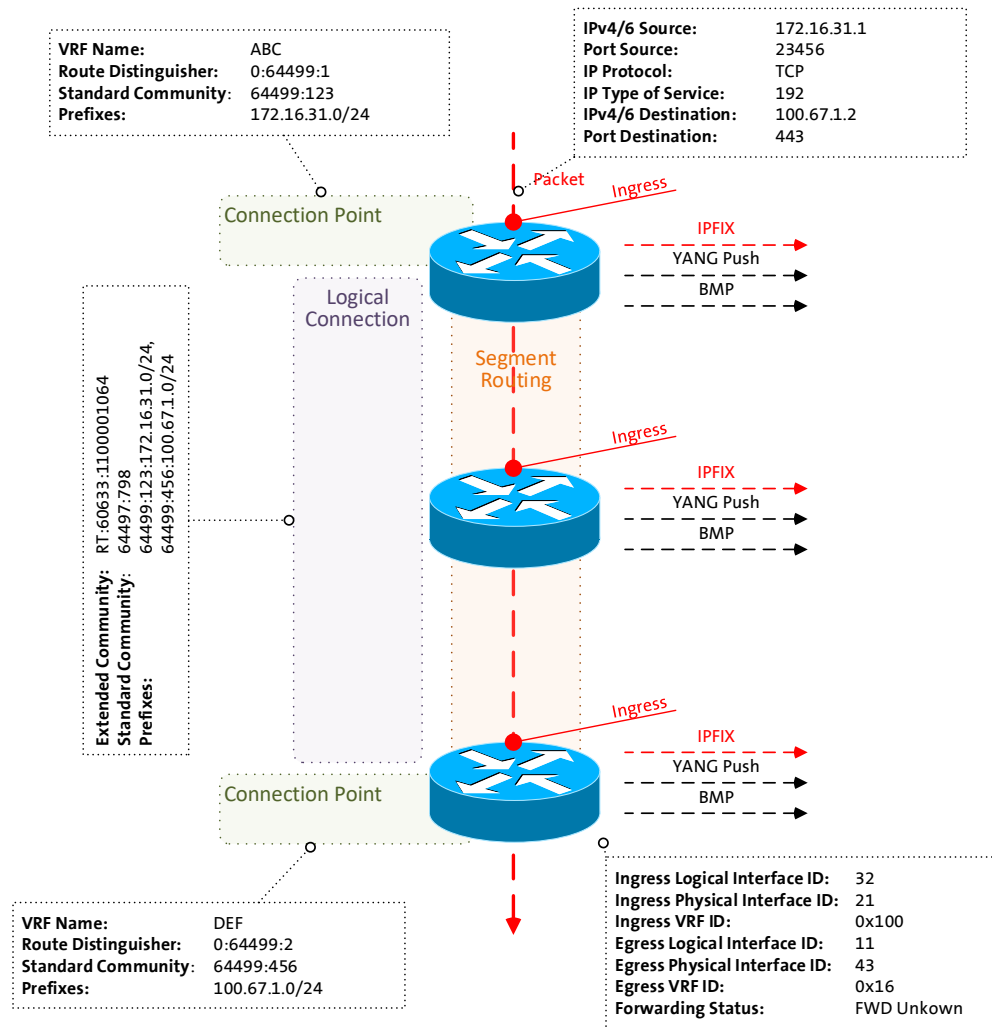
« Network operators **connect customers in** routing tables called **Connectivity Services** »

« Network Telemetry (RFC 9232) describes how to collect data from **all 3 network planes** efficiently »



# Monitoring L3 VPN's with IPFIX, BMP and YANG Push

## From Connectivity Service to Realtime Network Analytics



- > **Connectivity Service perspective**, Connection Points are connected through Logical Connections.
- > **From a BGP control-plane perspective**, IPv4/6 unicast prefixes in VRF's are tagged with BGP standard communities.
  - > One BGP standard community to identify the Logical Connection. One BGP standard community to identify each Connection Point.
  - > When IPv4/6 prefixes are exported from VRF's, a BGP route-distinguisher, BGP extended community route-targets and a SRv6 VPN SID for the IPv6 next-hop are allocated.
- > **From a forwarding plane perspective**, when IPv4/6 unicast traffic is received from the edge at the SRv6 PE, a lookup is performed, the SRv6 VPN SID is obtained and IPv6 next-hop is added when forwarded to the core.
- > **Swisscom collects** MPLS and SRv6 provider data plane, IPv4/6 unicast customer data-plane in IPFIX and at provider edge BGP VPNv4/6 unicast **in production** to perform real-time data correlation.

# Segment Routing IPv6 Encapsulation

RFC 9487 provides IPFIX Visibility

## > Provider data-plane

Divided into an IPv6 and Segment Routing Header.

The IPv6 header shows from which PE to which next-hop it is being forwarded. The Segment Routing Header the list of segments this packet needs to pass through and points to the active segment.

## > Customer data-plane

This is what we receive from the customer and encapsulate for transport through the SRv6 core.

No.	Time	Source IP	Source Port	Destination IP	Destination Port	Protocol
7	2022-12-22 13:50:12.823123	203.0.113.46		203.0.113.30		ICMP
8	2022-12-22 13:50:12.823197	203.0.113.30		203.0.113.46		ICMP

> Frame 7: 234 bytes on wire (1872 bits), 214 bytes captured (1712 bits)
> Ethernet II, Src: HuaweiTe_3a:2e:62 (f8:53:29:3a:2e:62), Dst: HuaweiTe_3a:33:a2 (f8:53:29:3a:33:a2)
> Internet Protocol Version 6, Src: 2001:db8:3::1, Dst: 2001:db8:18:0:10::
0110 .... = Version: 6
.... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
.... 0000 00.. .... = Differentiated Services Codepoint: Default (0)
.... ..00 .... = Explicit Congestion Notification: Not ECN-Capable
.... 0011 0001 1101 0001 1101 = Flow Label: 0x31d1d
Payload Length: 180
Next Header: Routing Header for IPv6 (43)
Hop Limit: 253
Source Address: 2001:db8:3::1
Destination Address: 2001:db8:18:0:10::
> Routing Header for IPv6 (Segment Routing)
Next Header: IPIP (4)
Length: 11
[Length: 96 bytes]
Type: Segment Routing (4)
Segments Left: 1
Last Entry: 3
Flags: 0x00
Tag: 0000
Address[0]: 2001:db8:2:0:40::
Address[1]: 2001:db8:18:0:10::
Address[2]: 2001:db8:17:0:10::
Address[3]: 2001:db8:14:0:10::
> Internet Protocol Version 4, Src: 203.0.113.46, Dst: 203.0.113.30
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 84
Identification: 0x0bdf (3039)
> 010. .... = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 63
Protocol: ICMP (1)
Header Checksum: 0xb77c [validation disabled]
[Header checksum status: Unverified]
Source Address: 203.0.113.46
Destination Address: 203.0.113.30
> Internet Control Message Protocol

# BMP – Address Family Agnostic

RFC 9069 provides Local RIB Visibility

## > BMP Per Peer Header

Shows at **which RIB** (Adj-RIB In, Local or Adj-RIB Out, Pre or Post Policy) and from **which Peering** the BGP PDU at **which time** was obtained.

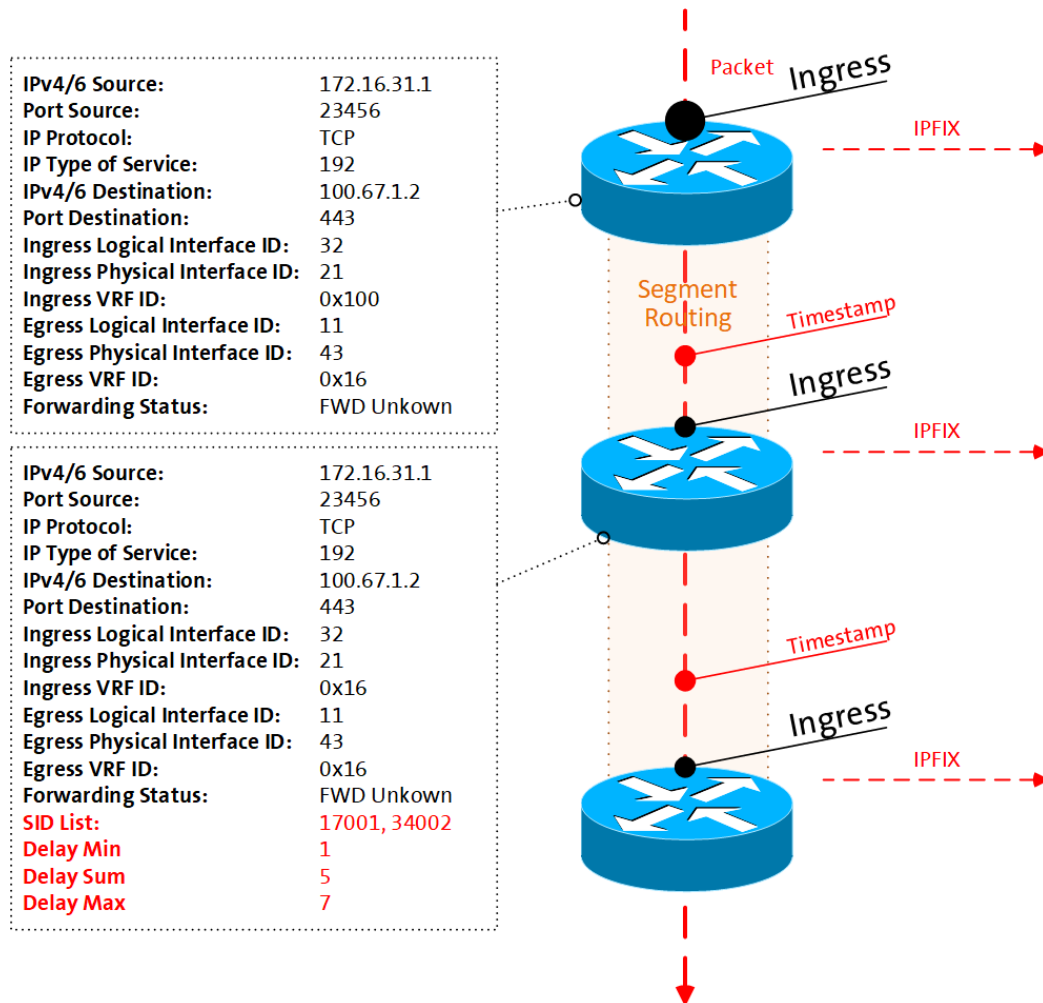
## > Encapsulated BGP PDU

Shows the encapsulated BGP PDU. In case of BMP route-monitoring, it describes whether it was a topology **update or withdrawal** and for **BGP community, NLRI and BGP Prefix SID** path attributes.

No.	Time	Source IP	Destination IP	Protocol	Length	Info
10	2023-11-06 22:12:33.943442	2001:db8:2::1	2a02:a90:4007::4:2	BGP	1294	UPDATE Message
<						
> Frame 10: 1294 bytes on wire (10352 bits), 1294 bytes captured (10352 bits)						
> Ethernet II, Src: Cisco_ff:dd:90 (40:06:d5:ff:dd:90), Dst: VMware_0e:d8:14 (00:0c:29:0e:d8:14)						
> Internet Protocol Version 6, Src: 2001:db8:2::1, Dst: 2a02:a90:4007::4:2						
> Transmission Control Protocol, Src Port: 39041, Dst Port: 1792, Seq: 746, Ack: 1, Len: 1220						
✓ BGP Monitoring Protocol, Type Route Monitoring						
Version: 3						
Length: 227						
Type: Route Monitoring (0)						
✓ Per Peer Header						
Type: Loc-RIB Instance Peer (3)						
> 0000 0000 = Flags: 0x00						
Peer Distinguisher: 0:0						
Unused: 000000000000000000000000						
Address: 0.0.0.0						
ASN: 65536						
BGP ID: 198.51.100.191						
Timestamp (sec): 1699272753						
Timestamp (msec): 942134						
✓ Border Gateway Protocol - UPDATE Message						
Marker: ffffffffffffffffffffffffffffffff						
Length: 179						
Type: UPDATE Message (2)						
Withdrawn Routes Length: 0						
Total Path Attribute Length: 156						
✓ Path attributes						
> Path Attribute - MP_REACH_NLRI						
> Path Attribute - ORIGIN: IGP						
> Path Attribute - AS_PATH: empty						
> Path Attribute - MULTI_EXIT_DISC: 0						
> Path Attribute - LOCAL_PREF: 16400						
> Path Attribute - COMMUNITIES: 64496:299 64496:1001 64497:1 64499:1						
> Path Attribute - EXTENDED_COMMUNITIES						
✓ Path Attribute - BGP Prefix-SID						
> Flags: 0xc0, Optional, Transitive, Complete						
Type Code: BGP Prefix-SID (40)						
Length: 37						
✓ SRV6 L3 Service						
Type: SRV6 L3 Service (5)						
Length: 34						
Reserved: 00						
✓ SRV6 Service Sub-TLVs						
✓ SRV6 Service Sub-TLV - SRV6 SID Information						
Type: SRV6 SID Information (1)						
Length: 30						
Reserved: 00						
SRV6 SID Value: 2001:db8:1::						
SRV6 SID Flags: 0x00						
SRV6 Endpoint Behavior: End.DT4 with NEXT-CSID (0x003f)						
Reserved: 00						
✓ SRV6 Service Data Sub-Sub-TLVs						
✓ SRV6 Service Data Sub-Sub-TLV - SRV6 SID Structure						
Type: SRV6 SID Structure (1)						
Length: 6						
Locator Block Length: 32						
Locator Node Length: 16						
Function Length: 16						
Argument Length: 0						
Transposition Length: 16						
Transposition Offset: 48						

# Measure On-Path Delay with Network Context

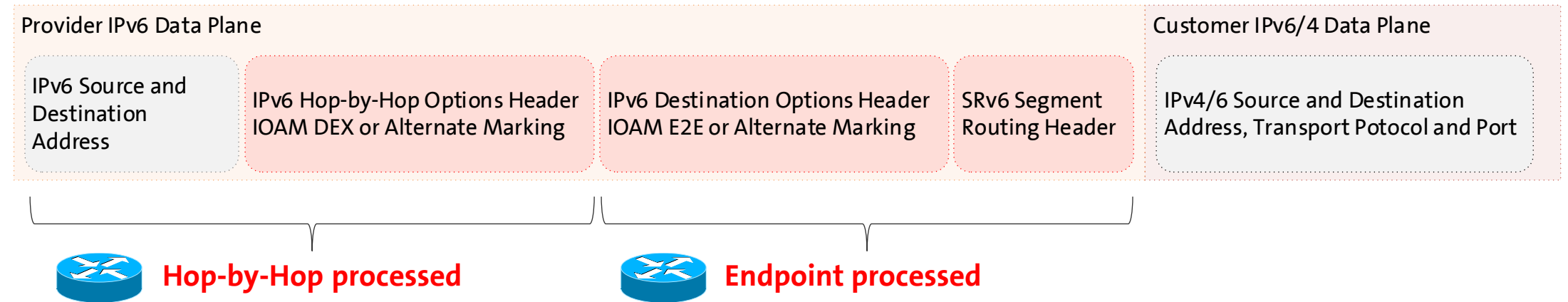
With draft-ietf-opsawg-ipfix-on-path-telemetry



- > Packets are **captured** ingress with an **optional sampler**, data plane dimensions **extracted, enriched** with management and control plane dimensions and added with a unique **flow ID** to a flow cache on the node for aggregation.
- > **A direct export marking bit and optionally a timestamp is added** to the packet when entering the OAM domain by leveraging Enhanced Alternate Marking ([RFC 9341](#), [draft-zhou-ippm-enhanced-alternate-marking](#)) or IOAM ([RFC 9378](#)) Direct Export or E2E Option Type.
- > Each subsequent packet for the same flow increases byte and packet count. Each new flow creates a new flow ID in the flow cache.
- > **At each node** in transit or only at the decapsulation node, **delay is calculated** by comparing the observation timestamp in the packet and when packet is received. **Delay is populated into the flow cache together with packet and byte count** as defined in [draft-ietf-opsawg-ipfix-on-path-telemetry](#).

# IPFIX provides statistical data plane visibility

Comprehensive IPv6 data plane coverage



- > [draft-ietf-opsawg-ipfix-on-path-telemetry](#) defines OAM agnostic on-path delay IPFIX entities.
- > [draft-ietf-opsawg-ipfix-alt-mark](#) defines Alternate Marking IPFIX entities.
- > [draft-spiegel-ippm-ioam-rawexport](#) defines IOAM IPFIX entities.
- > [RFC 9487](#) defines Segment Routing Header IPFIX entities





# Network Observability

## System Demo

Detect L3 VPN Topology changes  
and its Forwarding Plane impact  
in **Near Real-Time**

# What does Network Anomaly Detection mean

Monitor changes, called outliers, in networks



## Network Anomaly Detection

**For Connectivity Services**, Network Anomaly Detection **constantly monitors and detects any network or device topology change**, along with their associated forwarding consequences for customers as outliers. Notifications are sent to the Network Operation Center before the customer is aware of service disruptions. **It offers operational metrics for in-depth analysis**, allowing to understand in which platform the problem originates and facilitates problem resolution.



### Answers

What changed and when, on which connectivity service, and how does it impact the customers?



### Focuses

Provides meaningful connectivity service impact information before customer is aware of and support in root-cause analysis.



### Data Mesh

Consumes operational real-time Forwarding Plane, Control Plane and Management Plane metrics and produces analytical alerts.



### Direction

From connectivity service to network platform.

# What our motivation is

Automate learn and improve

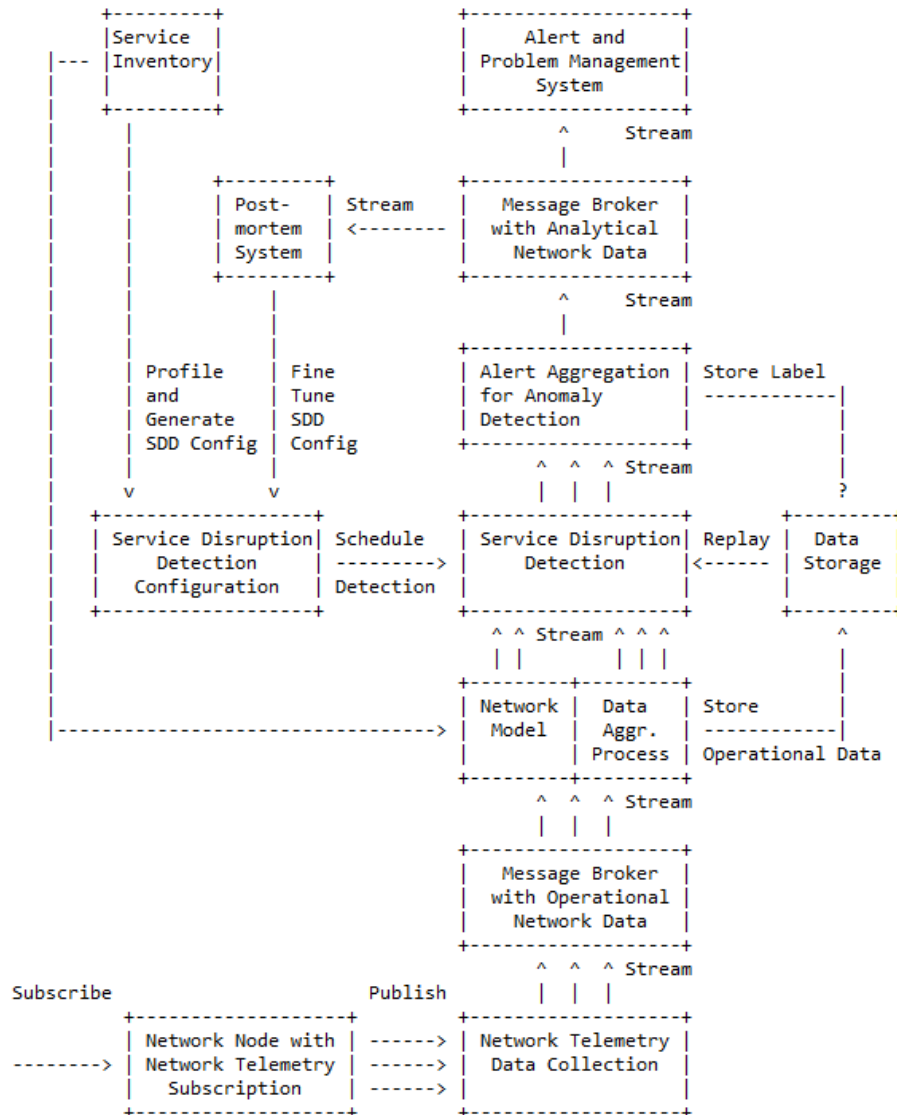
From network incidents postmortems we network operators **learn and improve** so does network anomaly detection and supervised and semi-supervised machine learning.

The more network incidents are observed, the more we can improve. With more incidents the **postmortem process needs be automated, let's get organized** first by defining human and machine-readable metadata semantics and annotate operational and analytical data.

Let's get further organized by exchanging standardized labeled network incident data among network operators, vendors and academia to **collaborate on academic research**.

« The community working on Network Anomaly Detection is probably the only group wishing for more network incidents »

# Elements of the Architecture



- **Service Inventory** contains list of the connectivity services.
- **Service Disruption Detection** processes aggregated network data to decide whether a service is degraded or not.
- **Service Disruption Detection Configuration** defines the set of approaches that need to be applied to perform SDD.
- **Operational Data Collection** manages network telemetry subscriptions and transforms data into message broker.
- **Operational Data Aggregation** produces data upon which detection of a service disruption can be performed.
- **Network Modeling** establishes knowledge of network relationships.
- **Data Profiling** categorizes nondeterministic customer related data.
- **Detection Strategies** for a profile a detection strategy is defined.
- **Machine Learning** is commonly used to detect outliers or anomalies.
- **Storage** some algorithms may relay on historical (aggregated) operational data to detect anomalies.
- **Alerting** consolidates analytical insights and notifies.
- **Postmortem** refines and stores the network anomaly and symptom labels into the Label Store.
- **Replaying** to validate refined anomaly and symptom labels, historical operational data is replayed.

# Semantic Metadata Annotation for Network Anomaly Detection

draft-netana-nmop-network-anomaly-semantics

```
module: ietf-symptom-semantic-metadata
```

```
  +--rw symptom
```

```
    +--rw id?                yang:uuid
    +--rw event-id?          yang:uuid
    +--rw description?        string
    +--rw start-time?         yang:date-and-time
    +--rw end-time?           yang:date-and-time
    +--rw confidence-score?   score
    +--rw concern-score?     score
```

```
    +--rw tags* [key]
```

```
      | +--rw key    string
      | +--rw value  string
```

```
    +--rw (pattern)?
```

```
      | +--:(drop)
      | | +--rw drop                empty
      | +--:(spike)
      | | +--rw spike                empty
      | +--:(mean-shift)
      | | +--rw mean-shift            empty
      | +--:(seasonality-shift)
      | | +--rw seasonality-shift    empty
      | +--:(trend)
      | | +--rw trend                empty
      | +--:(other)
      | +--rw other                  string
```

```
    +--rw annotator
```

```
      +--rw (annotator-type)
      | +--:(human)
      | | +--rw human                empty
      | +--:(algorithm)
      | | +--rw algorithm            empty
      +--rw name?                    string
```

- **Symptom ID and description** uniquely identifies the detected anomaly. **Event ID, start/end-time and confidence/concern-score** uniquely identifies the network event with its start and end time, how confident the system identified the anomaly and how concerned an operator should be.
- **Tags** allows to add customer information.
- **Pattern** describes the identified pattern of the anomaly.
- **Annotator Name, Type**, describes wherever the anomaly was detected by a human or algorithm and uniquely identifies the system who/which detected.

# Experiment: Network Anomaly Lifecycle

draft-netana-nmop-network-anomaly-lifecycle

« Network Anomaly Detection is an iterative process that requires continuous improvement »

## 4. Lifecycle of a Network Anomaly

The lifecycle of a network anomaly can be articulated in three phases, structured as a loop: Detection, Validation, Refinement.

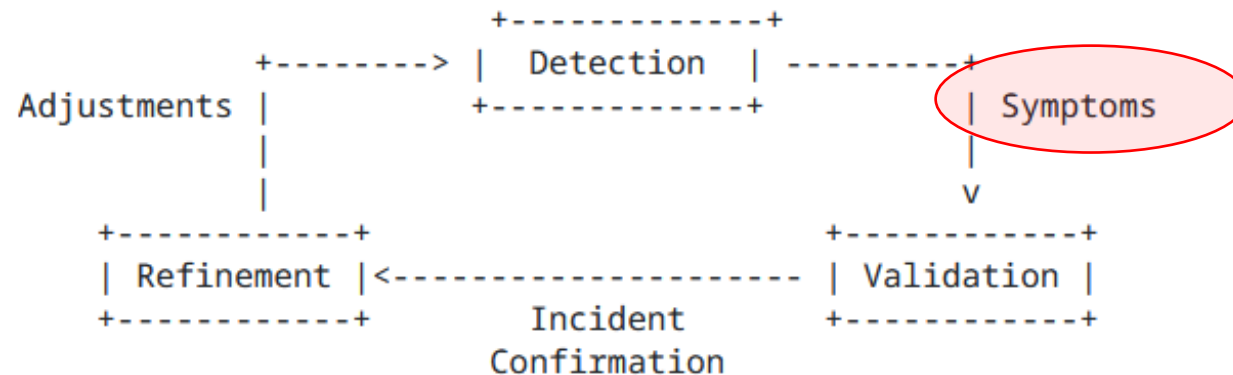


Figure 1: Anomaly Detection Refinement Lifecycle

Each of these phases can either be performed by a network expert or an algorithm or complementing each other.

**Detection:** The Network Anomaly Detection stage is about the continuous monitoring of the network through Network Telemetry [RFC9232] and the identification of symptoms.

**Validation:** Decides if the detected symptoms are signaling a real incident or if they are to be treated as false positives.

**Refinement:** Network operator performs detailed postmortem analysis of the network incident, collected Network Telemetry data and detected anomaly with the objective to identify useful adjustments in the Network Telemetry data collection and Anomaly Detection system.

# Experiment: Network Anomaly Lifecycle

draft-netana-nmop-network-anomaly-lifecycle

```
module: ietf-network-anomaly-metadata
```

```
  +--rw network-anomalies
```

```
    +--rw network-anomaly* [id version]
```

```
      +--rw id                yang:uuid
```

```
      +--rw version          uint32
```

```
      +--rw description?     string
```

```
      +--rw state             identityref
```

```
      +--rw annotator
```

```
        | +--rw (annotator-type)
```

```
        | | +--:(human)
```

```
        | | | +--rw human          empty
```

```
        | | +--:(algorithm)
```

```
        | |   +--rw algorithm      empty
```

```
        | +--rw name?              empty
```

```
      +--rw symptoms* [symptom_id]
```

```
        +--rw symptom_id          yang:uuid
```

- **ID and Description** uniquely identifies the detected network anomaly (as a container of symptoms).
- **Description and State** provide general information regarding the anomaly and .
- **Annotator** describes the entity that observed the network anomaly: this can be a human or an algorithm (anomaly detection system).
- **Symptoms** provides a list of symptoms (based on ietf-symptom-metadata) that are part of this network anomaly.

# Experiment: Network Anomaly Lifecycle

draft-netana-nmop-network-anomaly-lifecycle

**Network Anomaly Detection** is the art of understanding when something is not working as expected in the network.  
It is an **iterative process** that requires **continuous improvement**

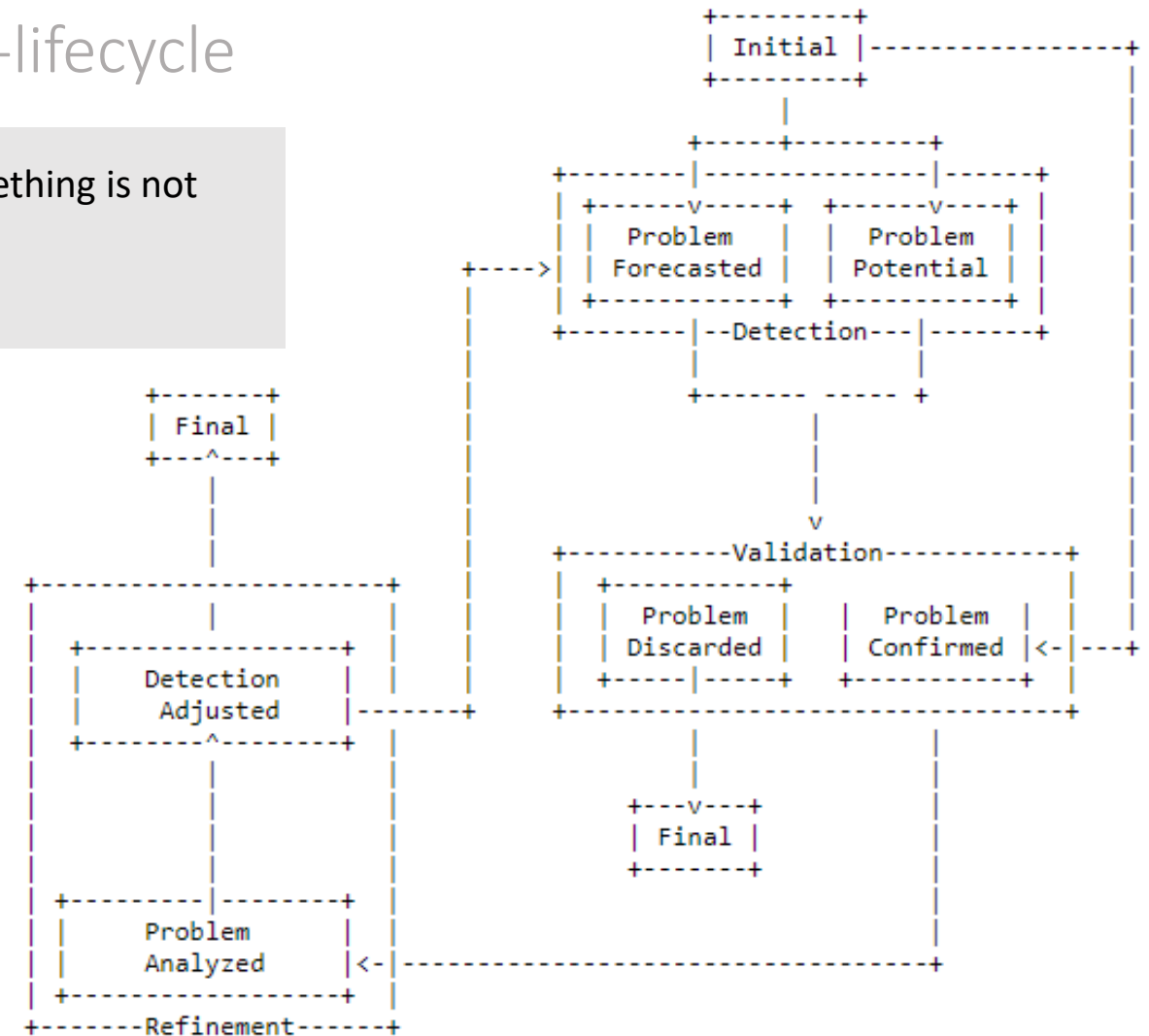
The detection in the different stages can either come from humans (network operation engineers) or algorithms (e.g. rule-based, AI-based).

→ **Having labels structured well is key**

It is crucial to make sure we can **audit the process** e2e and **involve network engineers** at any stage of the process to validate and

→ **Interoperability between teams and between “annotators”** is key

“annotator” = entity that produces the label - either human or algorithm

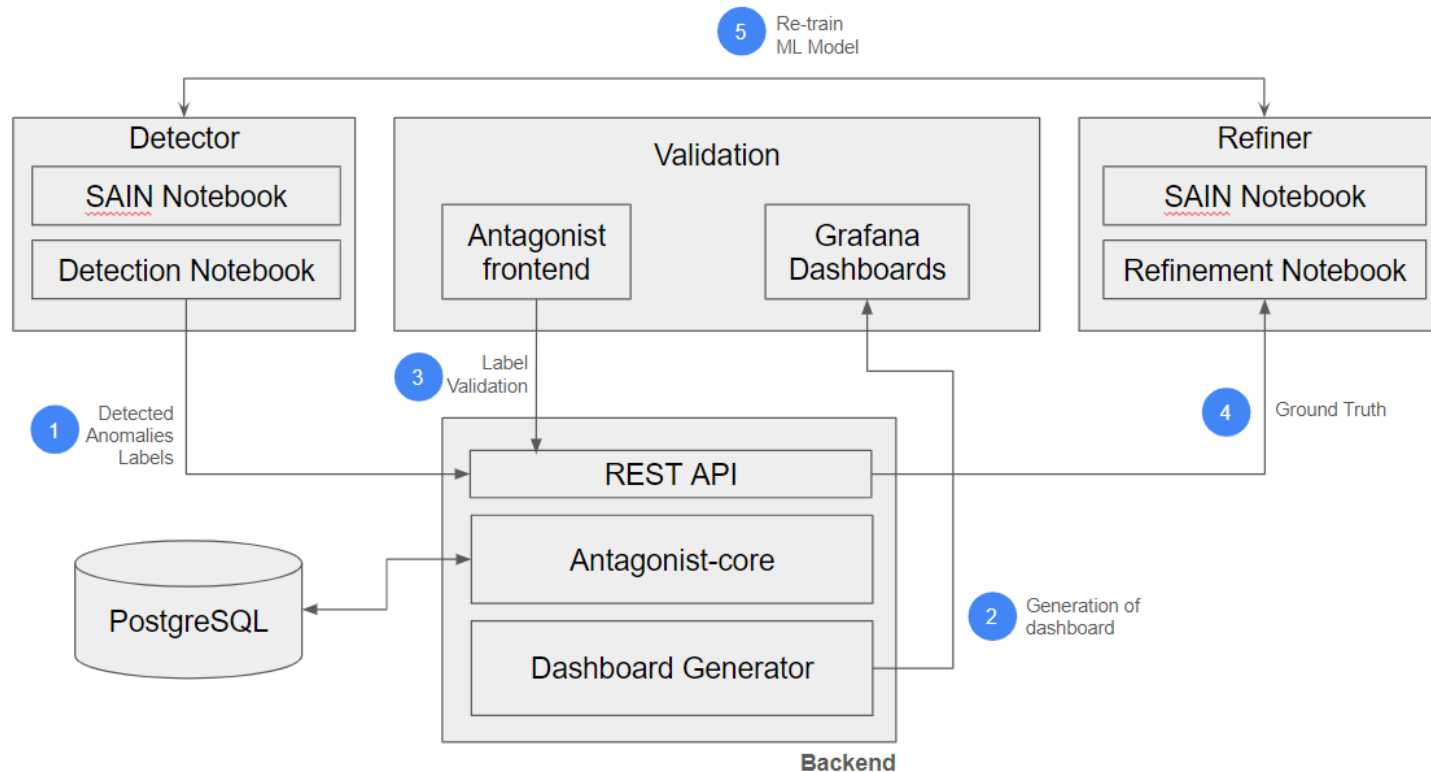


Identified three main stages of the life cycle



# Experiment: Antagonist

## anomaly tagging on historical data



### Next Steps:

- Improve scalability
- Integrate and Validate with Swisscom Data

### Goals:

- Prove that YANG models contain all the necessary information
- Validate models across a wide range of use-cases
- Show interoperability between

### Done so far:

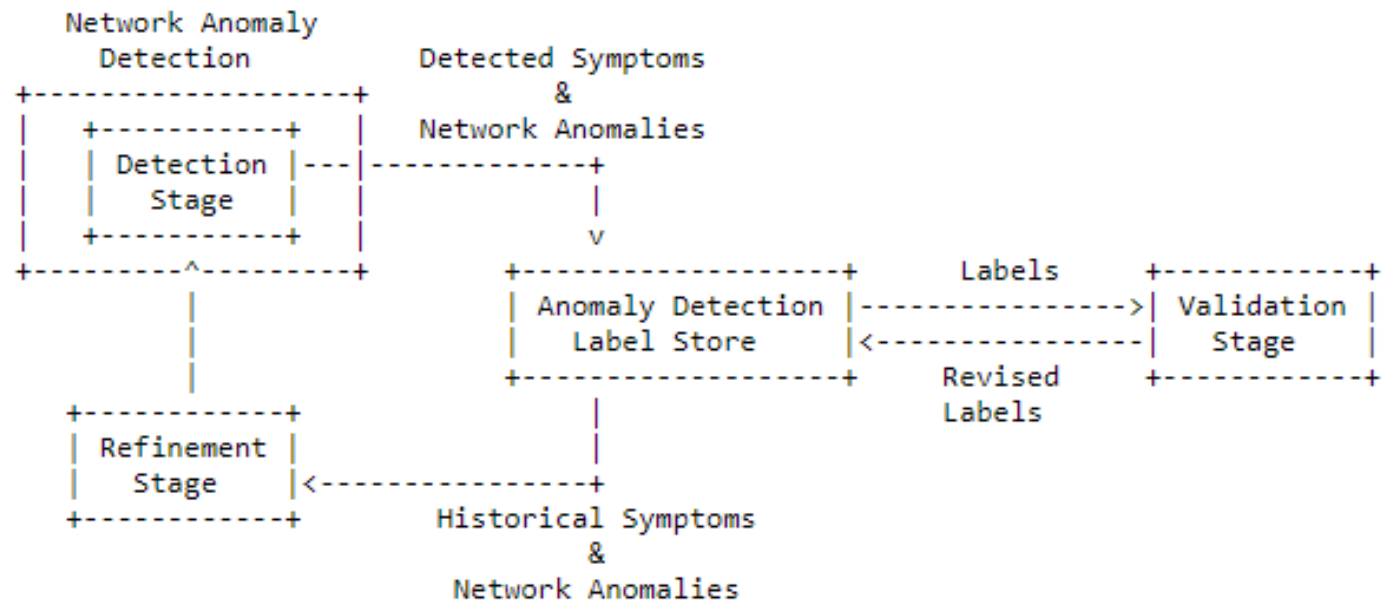
- ✓ Validation with real operational data (Cloud monitoring)
- ✓ Validation with rule-based Network Anomaly Detector (SAIN RFC9417/RFC9418)
- ✓ Validation with a ML-based Network Anomaly Detector (Autoencoder)
- ✓ Add support for Re-training of ML-based models
- ✓ Add partial support for Metadata Filtering and search
- ✓ YANG model refinements to reflect the results of the coding
- ✓ Automatic dashboard generation

## anomaly tagging on historical data

- Antagonist is a **Label Store** for Network Anomaly Detectors
- It's API is both human and machine readable and comes with a clear network specific semantic

Introducing a Label Store specific for network anomaly enables the following use cases:

- **Exchange of anomaly detection labels** between different operators or teams
- **Persistency** of anomaly detection labels generated by ML models
- **Enable network engineers to easily validate** the result of the detection and provide feedback to the model (active learning)
- **Provisioning of labels to ML** for dynamic retraining



# Experiment: Antagonist

## Supported YANG Models

In which state of the anomaly the label was generated

```
module: ietf-network-anomaly-metadata
+--rw network-anomalies
+--rw network-anomaly* [id version]
+--rw id yang:uuid
+--rw version uint32
+--rw description? string
+--rw state identityref
+--rw annotator
| +--rw (annotator-type)
| | +--:(human)
| | | +--rw human empty
| | +--:(algorithm)
| | | +--rw algorithm empty
| | +--rw name? empty
+--rw symptoms* [symptom_id]
+--rw symptom_id yang:uuid
```

Which entity generated the label

Provide labels to ML based on the specific action to perform  
(e.g. Active Learning, ML Retraining, )

```
module: ietf-symptom-metadata
+--ro ifws:symptom
+--ro ifws:id? yang:uuid
+--ro ifws:event-id? yang:uuid
+--ro ifws:description? string
+--ro ifws:start-time? yang:date-and-time
+--ro ifws:end-time? yang:date-and-time
+--ro ifws:confidence-score? score
+--ro ifws:concern-score? Score
```

Labels can be filtered based on the tags

```
+--ro (ifws:pattern)?
| +--:(ifws:drop)
| | +--ro ifws:drop empty
| +--:(ifws:spike)
| | +--ro ifws:spike empty
| +--:(ifws:mean-shift)
| | +--ro ifws:mean-shift empty
| +--:(ifws:seasonality-shift)
| | +--ro ifws:seasonality-shift empty
| +--:(ifws:trend)
| | +--ro ifws:trend empty
| +--:(ifws:other)
| | +--ro ifws:other string
+--ro ifws:tags* [key]
| +--ro ifws:key string
| +--ro ifws:value string
+--ro ifws:annotator
+--ro (ifws:annotator-type)
| +--:(ifws:human)
| | +--ro ifws:human empty
| +--:(ifws:algorithm)
| | +--ro ifws:algorithm empty
+--ro ifws:name? string
```

```
import sys
sys.path.append('.')
from demo_anomaly_detector import autoencoder_detector

# If a model has been pre-trained, it will be loaded automatically
anomaly_detector = autoencoder_detector.DemoAnomalyDetector()
```

Instantiate an  
anomaly  
detector

Detector  
(1 / 2)

```
network_anomalies = anomaly_detector.detect(telemetry_df)
```

Run detection on telemetry data

```
## Send the data to Antagonist
for network_anomaly in network_anomalies:

    # Create network anomaly label
    net_anomaly = {
        "annotator": {
            "name": anomaly_detector.get_model_name(),
            "annotator_type": "algorithm"
        },
        "description": f'Detected Network Anomaly on {machine_id} - {datetime.datetime.fromtimestamp(network_anomaly[0]).strftime("%Y-%m-%d at %H")}',
        "state": "incident-potential",
        "version": 1
    }
    response = requests.post(
        f"http://{ANTAGONIST_HOST}/api/rest/v1/network_anomaly", json=net_anomaly
    )
    response.raise_for_status()
    ni_uuid = response.json()
```

Store detected network anomalies on  
Antagonist

## Store detected symptoms on Antagonist

Detector  
(1 / 2)

```
# Create network symptoms labels and link with the network incident
for symptom in network_anomaly[2]:
```

```
    tags = {
        "machine": machine_id,
        "metric": db.get_metric_names()[symptom[0]],
        "group": group,
    }
```

Set the tags for the Symptoms

```
    net_sym = {
        'start-time': datetime.datetime.fromtimestamp(symptom[1]).strftime("%Y-%m-%dT%H:%M:%S"),
        'end-time': datetime.datetime.fromtimestamp(symptom[2]).strftime("%Y-%m-%dT%H:%M:%S"),
        "event-id": ni_uuid,
        "concern-score": symptom[3],
        "confidence-score": symptom[4],
        "description": "Symptom",
        "pattern": "",
        "tags": tags,
        "annotator": {
            "name": f"{anomaly_detector.get_model_name()}",
            "annotator_type": "algorithm"
        }
    }
```

Set the tags for the Symptoms

```
    # Persist the Symptom
    response = requests.post(
        f"http://{ANTAGONIST_HOST}/api/rest/v1/symptom", json=net_sym
    )
    response.raise_for_status()
    symptom_uuid = response.json()
```

Post the symptom to Antagonist

```
    # Link the Symptom to the network anomaly
    sym_to_net = {"symptom-id": symptom_uuid, "incident-id": ni_uuid}
    response = requests.post(
        f"http://{ANTAGONIST_HOST}/api/rest/v1/network_anomaly/symptom", json=sym_to_net
    )
    response.raise_for_status()
```

Connect the network anomaly with the Symptom

# Generation of Dashboards



The YANG model contains all the information that we need to automatically generate dashboards

# Label Validation - Tagging Network Anomalies

Home > Dashboards > Network Anomaly Tagging

2024-06-29 05:41:08 to 2024-07-06 20:57:43

Share

Network Anomalies

List of Network Anomalies

Spike that Lasted 1 Day on the 3rd of July

2024-07-02 20:35:47 - 2024-07-03 06:03:04

Network Anomaly

Network Anomaly Details

The chart displays various network metrics over a period from June 30 to July 6. A specific anomaly is highlighted with a blue shaded region on July 3rd, showing a sharp increase in the 'active\_opens' metric. The y-axis ranges from 0 to 0.5, and the x-axis shows dates and times.

active\_opens (group="Group-1", machine="machine-1-1", omnids="1")

cpu\_r (group="Group-1", machine="machine-1-1", omnids="1")

curr\_estab (group="Group-1", machine="machine-1-1", omnids="1")

disk\_q (group="Group-1", machine="machine-1-1", omnids="1")

disk\_r (group="Group-1", machine="machine-1-1", omnids="1")

disk\_rb (group="Group-1", machine="machine-1-1", omnids="1")

disk\_svc (group="Group-1", machine="machine-1-1", omnids="1")

disk\_u (group="Group-1", machine="machine-1-1", omnids="1")

disk\_w (group="Group-1", machine="machine-1-1", omnids="1")

disk\_wa (group="Group-1", machine="machine-1-1", omnids="1")

disk\_wb (group="Group-1", machine="machine-1-1", omnids="1")

eth1\_fl (group="Group-1", machine="machine-1-1", omnids="1")

eth1\_fo (group="Group-1", machine="machine-1-1", omnids="1")

eth1\_pi (group="Group-1", machine="machine-1-1", omnids="1")

eth1\_po (group="Group-1", machine="machine-1-1", omnids="1")

in\_errs (group="Group-1", machine="machine-1-1", omnids="1")

in\_segs (group="Group-1", machine="machine-1-1", omnids="1")

listen\_overflows (group="Group-1", machine="machine-1-1", omnids="1")

load\_1 (group="Group-1", machine="machine-1-1", omnids="1")

load\_15 (group="Group-1", machine="machine-1-1", omnids="1")

load\_5 (group="Group-1", machine="machine-1-1", omnids="1")

mem\_shmem (group="Group-1", machine="machine-1-1", omnids="1")

mem\_u (group="Group-1", machine="machine-1-1", omnids="1")

mem\_u\_p (group="Group-1", machine="machine-1-1", omnids="1")

Network Anomaly Tags

Network Anomaly

Description

Version

1

State

Choose

Annotator

Name

Type

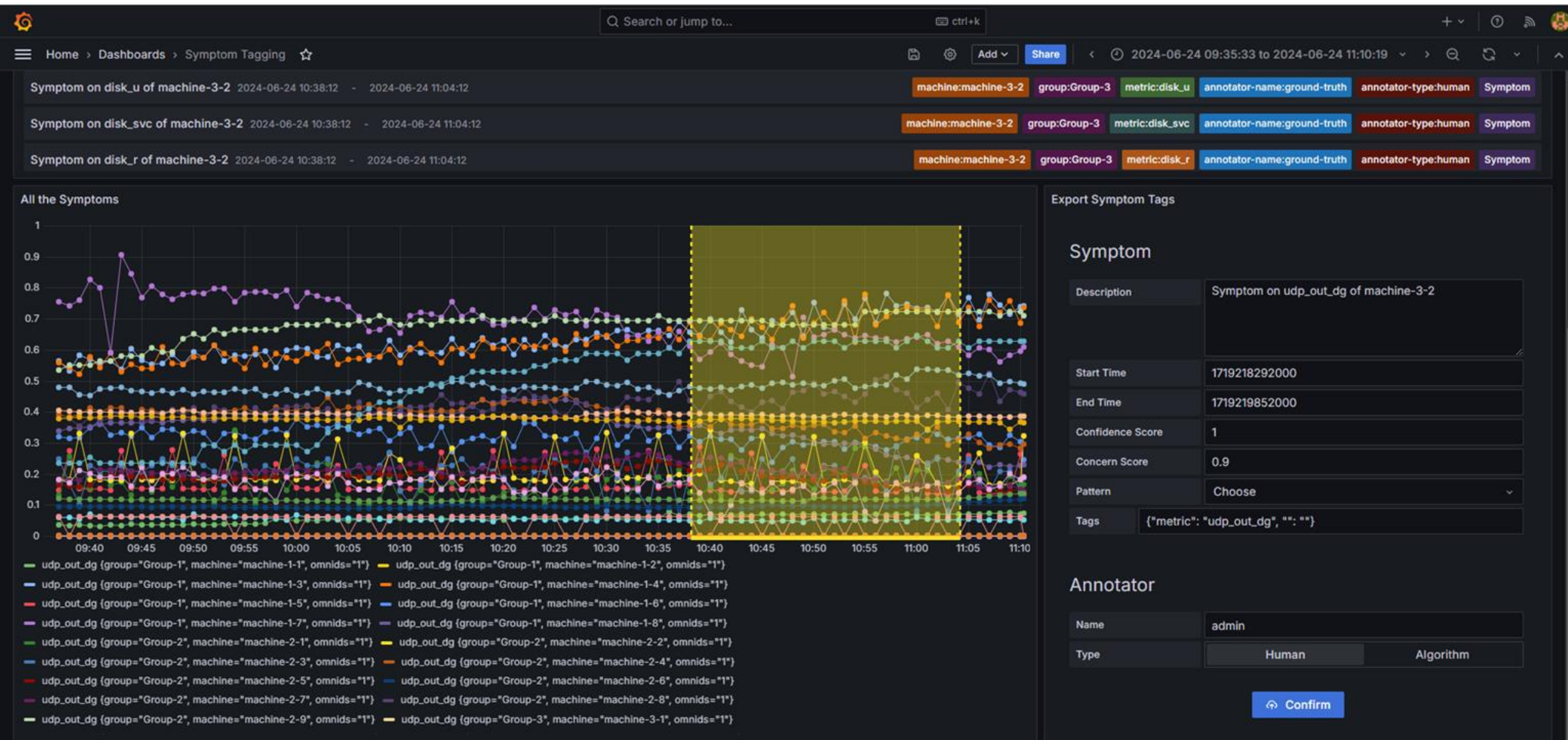
Human

Algorithm

Confirm



# Label Validation - Tagging Symptoms





# Label Validation - Analysing and editing network anomalies

Network Anomalies

Description

☒ Network Anomaly on machine-1-1 - 2024-07-01 at 22

☐ Network Anomaly on machine-1-1 - 2024-07-02 at 17

☐ Network Anomaly on machine-1-1 - 2024-07-03 at 11

☐ Network Anomaly on machine-1-1 - 2024-07-04 at 09

☐ Network Anomaly on machine-1-1 - 2024-07-05 at 09

☐ Network Anomaly on machine-1-1 - 2024-07-06 at 02

List of all the Network Anomalies

Visualize Details

Compare Versions

Network anomaly details

Network anomaly symptoms

Network anomaly versions comparison

Network anomaly stages

ID	Description	Annotator Name	Version	State
<input checked="" type="checkbox"/> d7e146a8-a96b-405a-bd13-a0...	Network Anomaly on machine-1-1 - ...	admin	2	Confirmed

List of all the Reviews of the network anomaly

Add New Version

Inspect

Network anomaly symptoms

Id	Description	Start-time	End-time	Confidence-score	Concern-score	Url
<input type="checkbox"/> d664102f-1b7e-4c...	Symptom on disk_wa of...	Mon, 01 Jul 2024 22:56:...	Tue, 02 Jul 2024 07:35:1...	1	0.9	<a href="http://localhost:3000/graf">http://localhost:3000/graf</a>
<input type="checkbox"/> 3e374c36-0bf4-45...	Symptom on disk_q of ...	Mon, 01 Jul 2024 22:56:...	Tue, 02 Jul 2024 07:35:1...	1	0.9	<a href="http://localhost:3000/graf">http://localhost:3000/graf</a>
<input type="checkbox"/> a643e6e9-e834-46...	Symptom on disk_svc o...	Mon, 01 Jul 2024 22:56:...	Tue, 02 Jul 2024 07:35:1...	1	0.9	<a href="http://localhost:3000/graf">http://localhost:3000/graf</a>
<input type="checkbox"/> d6f23c1a-1b3e-4d...	Symptom on disk_w of ...	Mon, 01 Jul 2024 22:56:...	Tue, 02 Jul 2024 07:35:1...	1	0.9	<a href="http://localhost:3000/graf">http://localhost:3000/graf</a>
<input type="checkbox"/> 21d7981c-aaa5-4e...	Symptom on disk_r of ...	Mon, 01 Jul 2024 22:56:...	Tue, 02 Jul 2024 07:35:1...	1	0.9	<a href="http://localhost:3000/graf">http://localhost:3000/graf</a>

# Label Validation - Analysing and editing network anomalies

## Network Anomalies

Description

☒ Network Anomaly on machine-1-1 - 2024-07-01 at 22

☐ Network Anomaly on machine-1-1 - 2024-07-02 at 17

☐ Network Anomaly on machine-1-1 - 2024-07-03 at 11

☐ Network Anomaly on machine-1-1 - 2024-07-04 at 09

☐ Network Anomaly on machine-1-1 - 2024-07-05 at 09

☐ Network Anomaly on machine-1-1 - 2024-07-08 at 02

Visualize Details

Compare Versions

Network anomaly details

Network anomaly symptoms

Network anomaly versions comparison

## New version

Annotator Name

admin

State

Confirmed

Id	Description	Start-time	End-time	Confidence-score
<input type="checkbox"/> d664102f-1b7e-4c6...	Symptom on disk_wa of ...	Mon, 01 Jul 2024 22:56:1...	Tue, 02 Jul 2024 07:35:1...	1
<input type="checkbox"/> 3e374c36-0bf4-459...	Symptom on disk_q of ...	Mon, 01 Jul 2024 22:56:1...	Tue, 02 Jul 2024 07:35:1...	1
<input type="checkbox"/> a643e6e9-e834-46c...	Symptom on disk_svc of ...	Mon, 01 Jul 2024 22:56:1...	Tue, 02 Jul 2024 07:35:1...	1
<input type="checkbox"/> d6f23c1a-1b3e-4d6...	Symptom on disk_w of ...	Mon, 01 Jul 2024 22:56:1...	Tue, 02 Jul 2024 07:35:1...	1
<input type="checkbox"/> 21d7981c-aaa5-4e1...	Symptom on disk_r of m...	Mon, 01 Jul 2024 22:56:1...	Tue, 02 Jul 2024 07:35:1...	1
<input type="checkbox"/> 86340f88-7d20-472...	Symptom on disk_u of ...	Mon, 01 Jul 2024 22:56:1...	Tue, 02 Jul 2024 07:35:1...	1

Add symptom

Delete symptom

Submit version

## Add symptoms

Start

02/07/2024

End

03/07/2024

Search

Start time

19:45

End time

03:44

Id	Description	Start-time	End-time	Confidence-score
<input type="checkbox"/> 17c17e26-ce8e-401...	Symptom on mem_u_e o...	Wed, 03 Jul 2024 03:05:1...	Wed, 03 Jul 2024 03:26:1...	1
<input type="checkbox"/> c673cf6a-cd6d-472...	Symptom on eth1_po of ...	Tue, 02 Jul 2024 17:30:1...	Wed, 03 Jul 2024 02:44:1...	1
<input type="checkbox"/> 245ae9ca-638d-469...	Symptom on out_segs o...	Wed, 03 Jul 2024 03:06:1...	Wed, 03 Jul 2024 03:17:1...	1
<input type="checkbox"/> 9f1db621-6941-437...	Symptom on tcp_use of ...	Wed, 03 Jul 2024 03:06:1...	Wed, 03 Jul 2024 03:17:1...	1
<input type="checkbox"/> 7bdc638b-3a26-469...	Symptom on disk_q of ...	Tue, 02 Jul 2024 17:30:1...	Wed, 03 Jul 2024 02:44:1...	1
<input type="checkbox"/> 450323f-70ad-465...	Symptom on disk_u of ...	Wed, 03 Jul 2024 03:06:1...	Wed, 03 Jul 2024 03:17:1...	1

# Refinement (Retraining)

```
params = {
    "start_time": start.strftime("%Y/%m/%dT%H:%M"),
    "end_time": end.strftime("%Y/%m/%dT%H:%M"),
    "annotator-type": "human"
}
response = requests.request(
    "GET", f"http://{ANTAGONIST_HOST}/api/rest/v1/symptom",
    headers=headers, data=payload, params=params)
response.raise_for_status()
# symptoms = format_symptoms(response.json(), start, end)
symptoms = response.json()
```

Get the symptom generated by humans

```
import sys
sys.path.append('.')

from demo_anomaly_detector import autoencoder_detector
new_anomaly_detector = autoencoder_detector.DemoAnomalyDetector()

annotation_df = json_to_df(symptoms)
new_anomaly_detector.train(historical_telemetry_df, annotation_df, force=True)
```

Train a new model  
(which can then be  
compared with  
previous one - or  
others)

## SAIN (1 / 2)

The SAIN Agent defines symptoms and pushes them into the Label Store

```
symptoms = [
    {
        'start-time': '2024-07-18T18:05:01',
        'end-time': '2024-07-18T23:12:42',
        "concern-score": 0.7, # impact_score
        "confidence-score": 1,
        "description": "Symptom",
        "pattern": "",
        "tags": {
            "subservice_id": '66409aa8-c0d2-4703-82f6-d18b7aaf3841',
            "subservice_type": 'Device',
            "expression": "device.cpu_util > 90%"
        },
        "annotator": {
            "name": f"SAIN-Agent-{sain_agent_id}",
            "annotator_type": "algorithm"
        }
    },
    {

```

A network anomaly is created to group those symptoms together

```
anomaly_dict = {
    "description": f'Device Under Stress',
    "state": "problem-potential",
    "version": 1,
    "symptoms": symptoms,
    "annotator": {
        "name": f"{sain_agent_id}",
        "annotator_type": "algorithm",
    }
}
```

## SAIN (2 / 2)

A Network Engineer does the validation and deems the network anomaly to actually be a problem

```
anomaly_dict = {
    "id": network_anomaly_id,
    "description": f'Device Under Stress',
    "state": "problem-confirmed",
    "version": 2,
    "symptoms": symptoms,
    "annotator": {
        "name": f"{sain_user_id}",
        "annotator_type": "human",
    }
}
```

A Network Engineer in the post-mortem analysis, discovers another symptom that would have helped identifying the root cause faster.  
This symptom is added to the list.

```
# Refinement of symptoms
new_symptom = {
    'start-time': '2024-07-18T18:05:01',
    'end-time': '2024-07-18T23:12:42',
    "concern-score": 1, # impact on health score
    "confidence-score": 1,
    "description": "Symptom",
    "tags": {
        "subservice_id": 'f803f55c-2967-4d24-a81a-f83267dc4239',
        "subservice_type": 'Interface',
        "expression": "interface.receive-packet > 800000"
    },
    "annotator": {
        "name": f"{sain_user_id}",
        "annotator_type": "human"
    }
}

symptoms.append(new_symptom)
anomaly_dict = {
    "id": network_anomaly_id,
    "description": f'Device Under Stress and High Network Traffic',
    "state": "problem-confirmed",
    "version": 3,
    "symptoms": symptoms,
    "annotator": {
        "name": f"{sain_user_id}",
        "annotator_type": "human",
    }
}
```

# Postmortem, L3 VPN Pilot Migration - Voice Over IP

## Post Maintenance Window Analysis

<div> <div>FILTER</div> <div> <div>Apr 9, 23:33-23:55</div> <div>Ip Prefix: 138.187.57.24...</div> <div>Log Type: 2 values</div> <div>Rd: 4 values</div> <div>+2</div> <div>+</div> </div> </div>	
<div> <div>SHOW</div> <div> <div>Time (5 Minutes)</div> <div>Log Type</div> <div>Rd</div> <div>Bgp Nexthop</div> <div>+2</div> <div>+</div> </div> </div>	
Time, Log Type, Rd, Bgp Nexthop, Platform Id, Node Id	Count
Overall	48
Apr 9, 23:30-23:35	36
withdraw	36
0:60633:4101214024	36
138.187.57.5	36
BNS_GW	36
bc001ro0101ss	6
bc001ro0101zhb	6
bc001ro0101zhh	6
bc001ro0101lss	6
bc001ro0101zhb	6
bc001ro011zhh	6
Apr 9, 23:35-23:40	6
withdraw	6
0:60633:4101214024	6
138.187.57.5	6
BNS_GW	6
bc001ro011olt	6
Apr 9, 23:50-23:55	6
withdraw	6
0:60633:4101214024	6
138.187.57.5	6
BNS_GW	6
bc001ro0101olt	6

Overall BGP Update/withdrawals Across Swisscom MPLS/SRv6 Cores



Maintenance window was scheduled to start on April 9th 22:00 with a total of 4 migration steps.



At 12:45 CE facing interfaces on first PE **node to be phased out was disabled.**

Triggering a VPNv4 withdrawal of 138.187.57.240/28 from 138.187.57.6 Lo0 towards route-reflectors and then to Inter-AS Option B ASBR. BMP route-monitoring and CLI show commands verified successful route propagation.



At 23:02 CE facing interfaces on first PE **node to be migrated to was enabled.**

Triggering a VPNv4 update of 138.187.57.240/28 from 138.190.129.180 Lo0 towards route-reflectors and then to Inter-AS option B ASBR. BMP route-monitoring and CLI show commands verified successful route propagation.



At 23:34 CE facing interfaces on second PE **node to be phased out was disabled.**

Triggering a VPNv4 withdrawal of 138.187.57.240/28 from 138.187.57.5 Lo0 towards route-reflectors and then to Inter-AS option B ASBR. BMP route-monitoring and CLI show commands verified successful route propagation to route-reflector **and on two Option B ASBR, but on other six after 20 mins delay.**

# Postmortem, L3 VPN Pilot Migration - Voice Over IP

## Show command drove to wrong conclusion

```
show route table bgp.l3vpn.0 protocol bgp 138.187.57.240/28 detail
```

```
60633:4101214024:138.187.57.240/28 (1 entry, 1 announced)
  BGP      Preference: 170/-101
           Route Distinguisher: 60633:4101214024
           Next hop type: Indirect, Next hop index: 0
           Address: 0x1a963a3c
           Next-hop reference count: 8
           Source: 138.190.128.116
           Protocol next hop: 138.187.57.5
           Label operation: Push 83714
           Label TTL action: prop-ttl
           Load balance label: Label 83714: None;
           Indirect next hop: 0x2 no-forward INH Session ID: 0x0
           State: <Delete Int Ext ProtectionPath ProtectionCand>
           Local AS: 64088.1116 Peer AS: 64088.1116
           Age: 5:28      Metric: 805      Metric2: 4
           Validation State: unverified
           Resolving-AIGP: 4
           Effective metric: 8 (IGP metric plus resolving AIGP)
           Task: BGP_64088.1116.138.190.128.116
           Announcement bits (1): 1-BMP
           AS path: 60633 64088.5 ?
           Communities: 60633:204 60633:208 60633:1002 64497:4965
64499:13338 target:60633:1100006314
  Accepted
  BMP: Pre: withdraw Station: DAISY_BMP_1
  BMP: Pre: withdraw Station: DAISY_BMP_2
  BMP: Station: <unassigned>
           Color: VPN Label: 83714
  Localpref: 100
  Router ID: 138.190.128.116
  Thread: junos-main
```

```
show route table bgp.l3vpn.0 protocol bgp 138.187.57.240/28 detail
```

```
60633:4103214024:138.187.57.240/28 (3 entries, 1 announced)
  *BGP     Preference: 170/-101
           Route Distinguisher: 60633:4103214024
           Next hop type: Indirect, Next hop index: 0
           Address: 0x1526757c
           Next-hop reference count: 4
           Source: 138.187.57.3
           Protocol next hop: 138.190.128.180
           Label operation: Push 83118
           Label TTL action: prop-ttl
           Load balance label: Label 83118: None;
           Indirect next hop: 0x2 no-forward INH Session ID: 0x0
           State: <Active Ext ProtectionPath ProtectionCand>
           Local AS: 64088.1116 Peer AS: 60633
           Age: 14:29:45      Metric: 800      Metric2: 4
           Validation State: unverified
           Resolving-AIGP: 4
           Effective metric: 8 (IGP metric plus resolving AIGP)
           Task: BGP_60633.138.187.57.3
           Announcement bits (2): 0-BGP_RT_Background 1-BMP
           AS path: 60633 64088.1180 ?
           Communities: 60633:204 60633:208 60633:1001 60633:1111
64497:4965 64499:13338 target:60633:1100006314
  Accepted
  BMP: Pre: advertise Station: DAISY_BMP_1
  BMP: Pre: advertise Station: DAISY_BMP_2
           Color: VPN Label: 83118
  Localpref: 100
  Router ID: 138.187.57.3
  Thread: junos-main
```



Juniper JunOS  
CLI show  
command  
shows that path  
is for 20min no  
longer primary  
active but still  
as backup path  
inactive. **Output  
mislead  
network  
engineer to  
believe that  
path is still  
installed.**



# Postmortem, L3 VPN Pilot Migration - Voice Over IP

## Data collection timestamp drove to wrong conclusion

```
"timestamp": "Tue Apr 09 2024 23:34:21",  
"writer_id": "bew03bmp45c 20240220-1 (45ae4201)",  
"peer_ip": "138.190.128.117",  
"string": "Tue Apr 09 2024 23:52:34"
```

```
"timestamp": "Tue Apr 09 2024 23:34:21",  
"writer_id": "bew03bmp45c 20240220-1 (45ae4201)",  
"peer_ip": "138.187.57.4",  
"string": "Tue Apr 09 2024 23:52:34"
```

```
"timestamp": "Tue Apr 09 2024 23:34:21",  
"writer_id": "bew03bmp45c 20240220-1 (45ae4201)",  
"peer_ip": "138.187.57.3",  
"string": "Tue Apr 09 2024 23:52:34"
```

```
"timestamp": "Tue Apr 09 2024 23:34:21",  
"writer_id": "bew03bmp45c 20240220-1 (45ae4201)",  
"peer_ip": "138.190.128.117",  
"string": "Wed Apr 10 2024 01:04:00"
```

```
"timestamp": "Tue Apr 09 2024 23:34:21",  
"writer_id": "zoi03bmp45c 20240220-1 (45ae4201)",  
"peer_ip": "138.187.57.3",  
"string": "Tue Apr 09 2024 23:54:09"
```

```
"timestamp": "Tue Apr 09 2024 23:34:21",  
"writer_id": "zoi03bmp45c 20240220-1 (45ae4201)",  
"peer_ip": "138.187.57.4",  
"string": "Tue Apr 09 2024 23:54:17"
```

```
"timestamp": "Tue Apr 09 2024 23:34:21",  
"writer_id": "zoi03bmp45c 20240220-1 (45ae4201)",  
"peer_ip": "138.190.128.117",  
"string": "Tue Apr 09 2024 23:54:24"
```

```
"timestamp": "Tue Apr 09 2024 23:34:21",  
"writer_id": "zoi03bmp45c 20240220-1 (45ae4201)",  
"peer_ip": "138.190.128.117",  
"string": "Wed Apr 10 2024 00:54:51"
```

The yellow marked timestamp shows the optional BMP per-peer header observation timestamp.

The blue marked timestamp shows the timestamp being augmented on the BMP data collection and **being used for the time series database.**

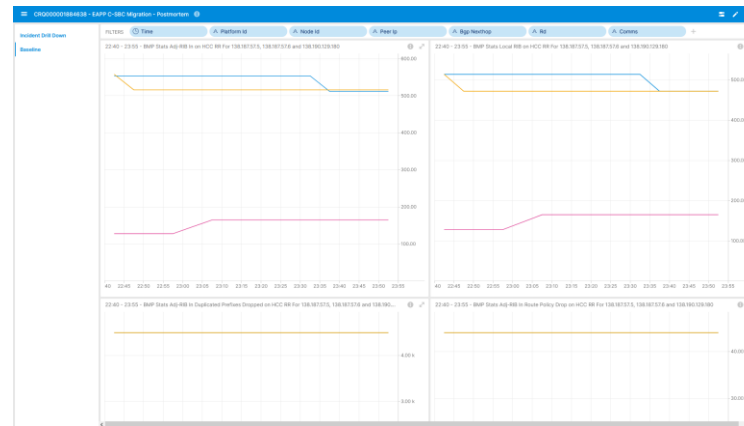


Because BMP per-peer timestamp is optional, in the time series database ingestion, the data collection augmentation timestamp is used instead. **Leading to false conclusions when the state change was observed.**

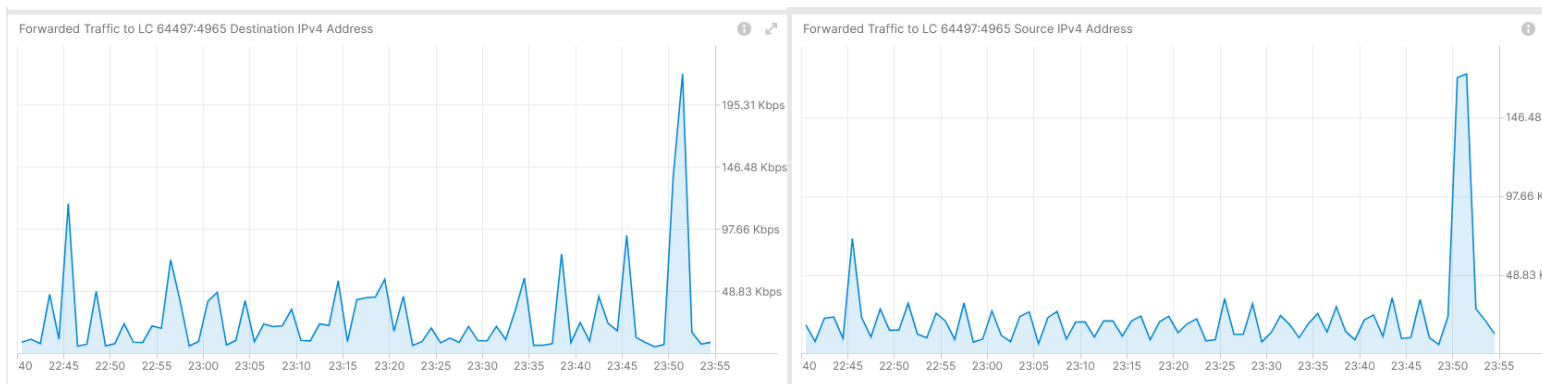


# Postmortem, L3 VPN Pilot Migration - Voice Over IP

## Route-Reflector Peering and L3 VPN Traffic View



BMP Peering Statistics on Route Reflectors



Traffic to Voice over IP Service on affected L3 VPN



IPFIX configured on PE **but not on involved MPLS Inter-AS option B ASBRs due to PR1567039.**



BMP ADJ-RIB In pre-policy on BGP VPNv4 /6 on MPLS PE's. BMP ADJ-RIB In pre-policy on BGP VPNv4 /6 on Route Reflectors and BMP ADJ-RIB In pre-policy and ADJ-RIB Out post-policy on Inter-AS Option B ASBR.



YANG Push on most nodes but not relevant for this use case.

Real-Time Streaming  
under Development

# Postmortem, L3 VPN Pilot Migration - Voice Over IP

## 64497:4965 - Anomaly Detection - Live

Max Concern Score: **NA**  
BMP Withdrawal Score: **0.19**



**BMP route-monitoring**  
Update/Withdraw check recognize withdrawal.



BMP peer Down/Up check did not apply.



**Interface Down/Up check did not recognize.**



Traffic Drop spike did not apply.



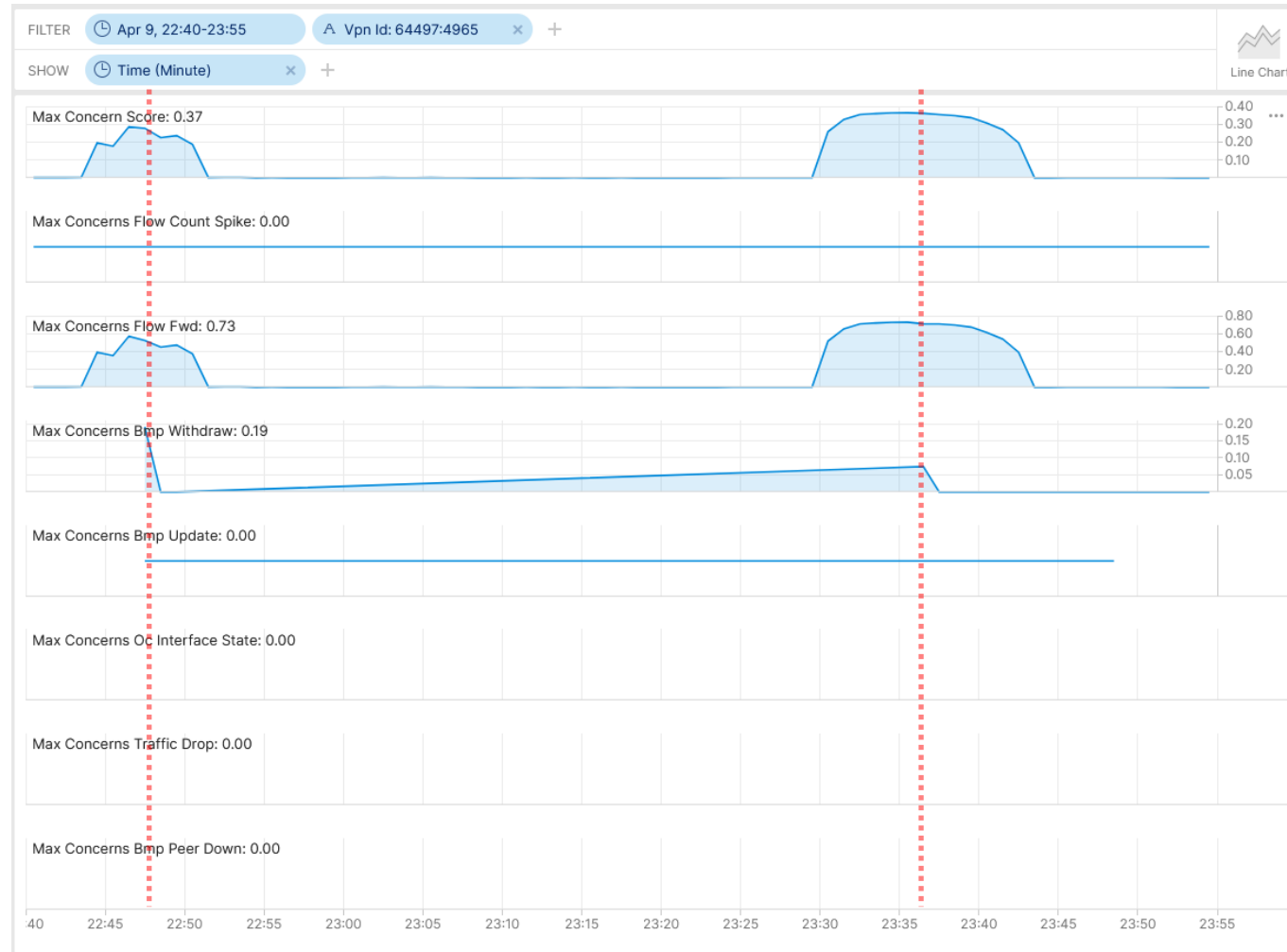
Missing Traffic check did not apply.  
(not fully implemented yet).



Increased or decreased Flow Count did not apply.



**Overall: 1 out of 6 checks have detected the BGP topology change.**  
Real-time streaming implementation work in progress as expected.



Cosmos Bright Lights Anomaly Detection – 64497:4965

# Postmortem

## What to do next?

- **Support on upcoming maintenance window with verification dashboard and active monitoring.**  
**-> Done**

### What went well?



Work in progress Cosmos Bright Lights real-time streaming Anomaly Detection BMP route-monitoring withdrawal rule detected topology change.



BMP collected metrics are consistent across multiple vendors vs. CLI show output is vendor dependent.

### What could be improved?



BMP per-peer observation timestamp should be mandatory. See <https://datatracker.ietf.org/doc/html/draft-boucadair-nmop-rfc3535-20years-later-02#section-4.7>. -> **To be addressed in GROW/NMOP.**

BMP per-peer header should have an export timestamp. See <https://datatracker.ietf.org/doc/html/draft-boucadair-nmop-rfc3535-20years-later-02#section-4.7>. -> **To be addressed in GROW/NMOP.**

With [RFC 8671](#) (Support for Adj-RIB-Out in BMP) path propagation could have been observed on route-reflectors.

With [draft-lucente-grow-bmp-rel](#) (Logging of routing events in BMP) path drops could be observed on Inter-AS option B ASBRs and route-reflectors.

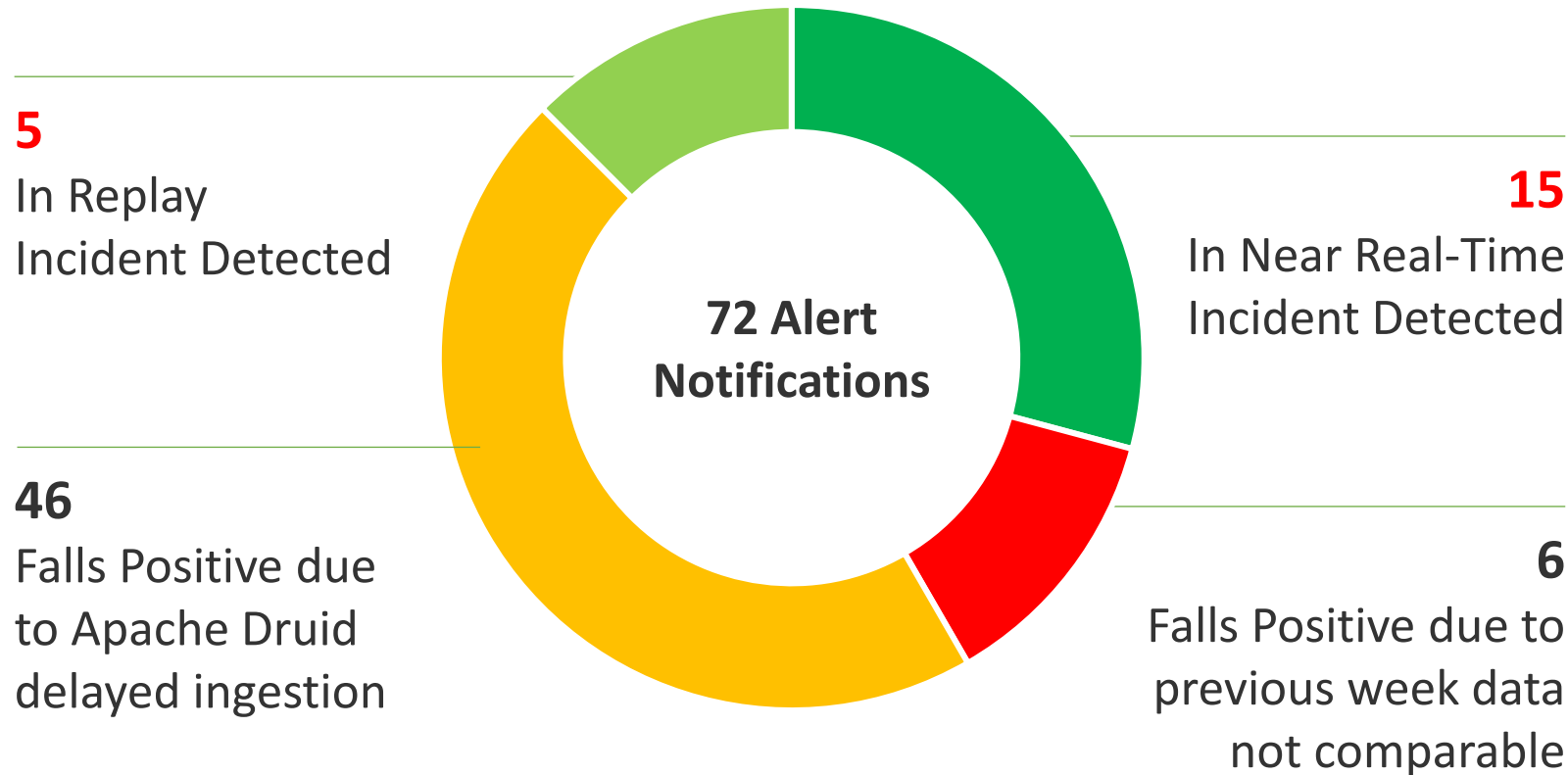
With [draft-ietf-grow-bmp-path-marking-tlv](#) path status changed could have been observed on Inter-AS option B ASBRs.

Clarify why Juniper JunOS delayed BMP export for 20 resp. 80 minutes. Due to fact that the path was still passive in the BGP RIB?

With IPFIX (**deconfigured due to PR1567039**) and support of IE90 ForwardingStatus (**not supported on Juniper JunOS**) forwarding drops could have been observed on Inter-AS option B ASBRs.

# Swisscom - Cosmos Bright Lights PoC Summary

After 20 Incidents and 18 Months Time

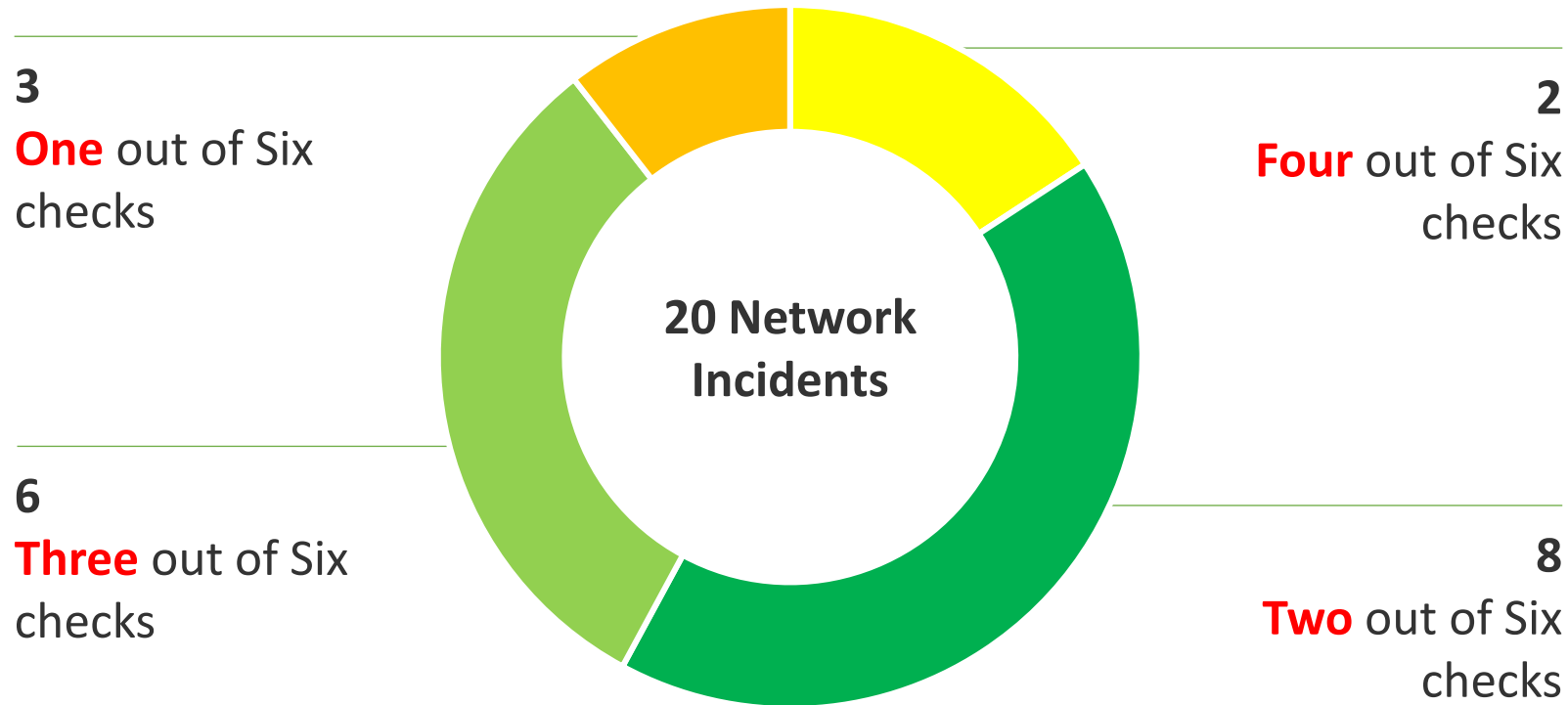


## Key Facts in V0 (2023-2024)

- 16 L3 VPNs proactively monitored.
- Individual Service Disruption Detection rule accuracy is beyond 90%. Summed accuracy is beyond 95%.
- Max Concern score ranged between 0.06 and 0.85. In average 0.46.
- In 4 cases additional YANG, in 13 cases additional BMP, in 2 cases Netconf Transaction-ID and 1 case additional L2 IPFIX metrics would have helped to gain more visibility.
- Key observability feature missing: BMP Local RIB with Path Marking.

# Swisscom - Cosmos Bright Lights PoC Detail

Multiple Perspectives increases Accuracy



## Key Improvements in V1 (2024)

- >12000 L3 VPNs proactively monitored since June 2024.
- Realtime Streaming eliminates delayed ingestion falls positives and scaling.
- Improved profiling. Compares to multiple previous weeks and discard largest deviation eliminates falls positives.  
-> Work In progress

## Key Improvements in V2 (2025)

- Annotate operational and analytical Network Incident data for reproduction.
- Enabling automated workflow. From PowerPoint slide decks to data driven actionable insights.

# Network Anomaly Detection Framework

Detect service interruption faster than humans can

## Incident and Problem Management:

- Some Key Terms for Network Incident and Problem Management

[draft-ietf-nmop-terminology](#)



## Network Anomaly Detection:

- An Architecture for a Network Anomaly Detection Framework

[draft-netana-nmop-network-anomaly-architecture](#)



- Semantic Metadata Annotation for Network Anomaly Detection

[draft-netana-nmop-network-anomaly-semantics](#)



- Experiment: Network Anomaly Lifecycle

[draft-netana-nmop-network-anomaly-lifecycle](#)



« Addressing the need to detect connectivity service interruption faster than humans can and facilitate collaboration by enabling exchange on labeled data with standardized semantics on a common framework.

**Please consider to attend IETF 120 NMOP working group session on Friday 13:00 – 15:00 or go onto the mailing list and contribute to the discussion. »**

# Relevant Papers for more Details

## Practical Anomaly Detection in Internet Services: An ISP centric approach

Alex Huang Feng\*, Pierre Francois\*, Kensuke Fukuda†, Wanting Du‡, Thomas Graf †, Paolo Lucente§, Stéphane Frénét\*

\*INSA Lyon, ‡National Institute of Informatics, †Swisscom, §pmacct.net

alex.huang-feng@insa-lyon.fr, pierre.francois@insa-lyon.fr, kensuke@nii.ac.jp, wanting.du@swisscom.com, thomas.graf@swisscom.com, paolo@pmacct.net, stephane.frenot@insa-lyon.fr

### Context

- ISPs provide multiple **IP connectivity services** such as:
  - BGP/MPLS VPNs
  - Internet Connectivity
- Network **disruptions and anomalies** degrade the reputation and impact the business of ISPs
- Network operators want to detect these anomalies
  - Comprehensively**: to understand the issue when alerted
  - Automatically**: to provide a notification if possible
- How can we detect these anomalies in **real world ISPs**?
- Which data can we use? Can we use **Shendata** only?
- Can a **rule-based approach** leveraging knowledge from operations be effective?

### Challenges of ISP networks

- Real world networks are heterogeneous:
  - Built with devices from multiple vendors
  - Devices have different network telemetry capabilities
  - Devices could not all be monitored
- State of the art focused mostly on:
  - Internet topology using BGP
  - Public data (Routeviews & RIPE NCC Archives)
  - Very few using production data & detecting anomalies in a single domain [1–3]

## Paper “**Practical Anomaly Detection in Internet Services: An ISP centric approach**”

accepted at AnNet’24

(in conjunction with IEEE NOMS’24)

Seoul, Korea (6–10 May 2024)

[Will be presented as a poster the May 6th 2024]

## Daisy: Practical Anomaly Detection in large BGP/MPLS and BGP/SRv6 VPN Networks

Alex Huang Feng alex.huang-feng@insa-lyon.fr Univ Lyon, INSA Lyon, Inria, CITI, EA3720 Villeurbanne, France	Pierre Francois pierre.francois@insa-lyon.fr Univ Lyon, INSA Lyon, Inria, CITI, EA3720 Villeurbanne, France	Stéphane Frénét stephane.frenot@insa-lyon.fr Univ Lyon, INSA Lyon, Inria, CITI, EA3720 Villeurbanne, France
Thomas Graf thomas.graf@swisscom.com Swisscom Zurich, Switzerland	Wanting Du wanting.du@swisscom.com Swisscom Zurich, Switzerland	Paolo Lucente paolo@pmacct.net pmacct.net Barcelona, Spain

### ABSTRACT

We present an architecture aimed at performing Anomaly Detection for BGP/MPLS VPN services, at scale. We describe the challenges associated with real time anomaly detection in modern, large BGP/MPLS VPN and BGP/IPv6 Segment Routing VPN deployments. We describe an architecture required to collect the necessary routing information at scale. We discuss the various dimensions which can be used to detect anomalies, and the caveats of the real world impacting the level of difficulty of such anomaly detection and network modeling. We argue that a rule-based anomaly detection approach, defined for each customer type, is best suited given the current state of the art. Finally, we review the current IETF contributions which are required to benefit from a fully open, standard, architecture.

### 1 INTRODUCTION

Customers subscribing to BGP/MPLS VPN services usually come along with stringent Service Level Agreements. Consequently, Service Providers must be capable of detecting anomalies in their services in a timely fashion, while accommodating for scale. Around 10 thousand L3 VPNs in our Swisscom use case. Long-lasting outages, detected by the customer before the service provider, are detrimental to the perception of service quality, and may dramatically impact the customer business.

The goal of the presented architecture is to provide an anomaly detection solution that scales while being flexible on the following aspects: (i) the dimensions that must be used to detect anomalies are multiple; (ii) VPN customers wear different profiles in terms of normal and abnormal values for such dimensions; (iii) the amount of information collected to produce values for such dimensions is extremely large in such deployments: around 175 thousand messages/second in our use case; (iv) the operating costs for managing an anomaly detection solution must be kept low; and (v) the networking platforms providing the service may come from different vendors and have different monitoring capabilities.

The remainder paper is structured as follows. In section 2, we define what is considered a network anomaly and present the associated challenges behind its detection. In Section 3, we describe the Daisy architecture. In Section 4, we review the ongoing IETF efforts aimed at filling the gaps for a fully open, standard, Anomaly Detection (AD) implementation. And finally, in section 5, we present the first results of Daisy deployment at Swisscom.

### 2 PROBLEM STATEMENT

We describe some of the challenges associated with customer diversity, and a non-exhaustive list of anomalies targeted by the base recipes from our limited proof of concept deployment setup.

### ACM Reference Format:

Alex Huang Feng, Pierre Francois, Stéphane Frénét, Thomas Graf, Wanting Du, and Paolo Lucente. 2023. Daisy: Practical Anomaly Detection in large BGP/MPLS and BGP/SRv6 VPN Networks. In *Applied Networking Research Workshop (ANRW ’23)*, July 24, 2023, San Francisco, CA, USA. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3606464.3606470>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

ANRW ’23, July 24, 2023, San Francisco, CA, USA  
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-4-4007-0274-7/23/007...\$15.00  
<https://doi.org/10.1145/3606464.3606470>

## Paper “**Daisy: Practical Anomaly Detection in large BGP/MPLS and BGP/SRv6 VPN Networks**”

published at ACM/IRTTF ANRW’23

San Francisco, USA (24 July 2023)

Open access: <http://hal.science/hal-04307611>







# Handling Operational YANG Modelled Data

## State of the Union

Nowadays network operators are using **machine and human readable YANG** [RFC 7950](#) to model their configurations and obtain YANG modelled data from their networks.

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

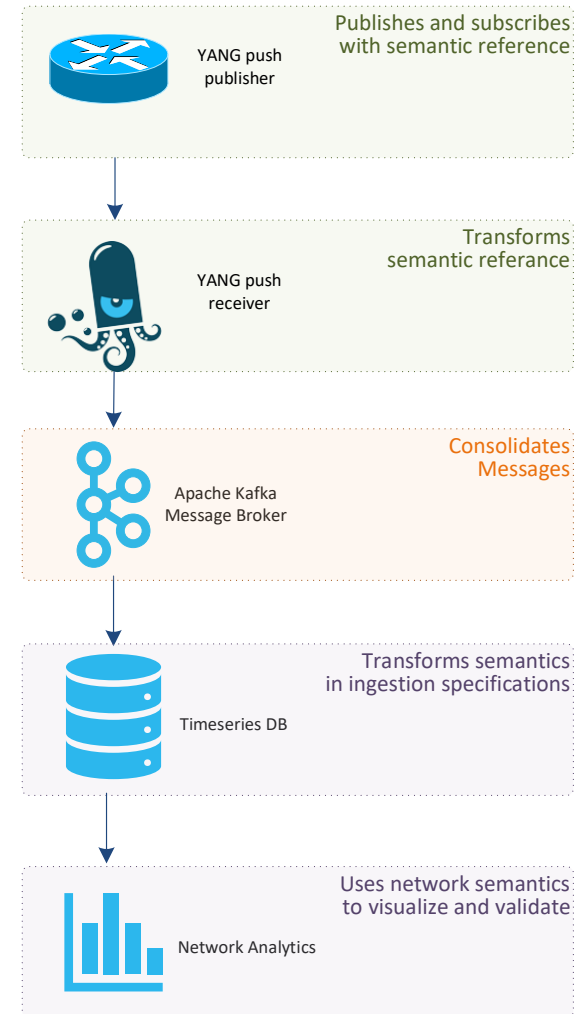
Today, subscribing to a YANG datastore, publishing a YANG modeled notifications message from the network and viewing the data in a time series database, **manual labor is needed to perform data transformation** to make a message broker and its data processing components with YANG notifications interoperable.

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

# From YANG-Push to Network Analytics

Aiming for an automated data processing pipeline

- **A network operator aims for:**
  - An **automated data processing pipeline** which starts with YANG-Push, consolidates at Data Mesh and ends at Network Analytics.
  - Operational metrics where **IETF defines the semantics.**
  - Analytical metrics where **network operators gain actionable insights.**
- **We achieve this by integrating YANG-Push into Data Mesh to:**
  - Produce metrics from networks **with timestamps when network events were observed.**
  - Hostname, publisher ID and sequence numbers help us to understand **from where metrics were exported and measure its delay and loss.**
  - Forward **metrics unchanged** from networks
  - **Learn semantics** from networks and validate messages.
  - **Control semantic** changes end to end.



# Elements of the Architecture

## Workflow Diagram

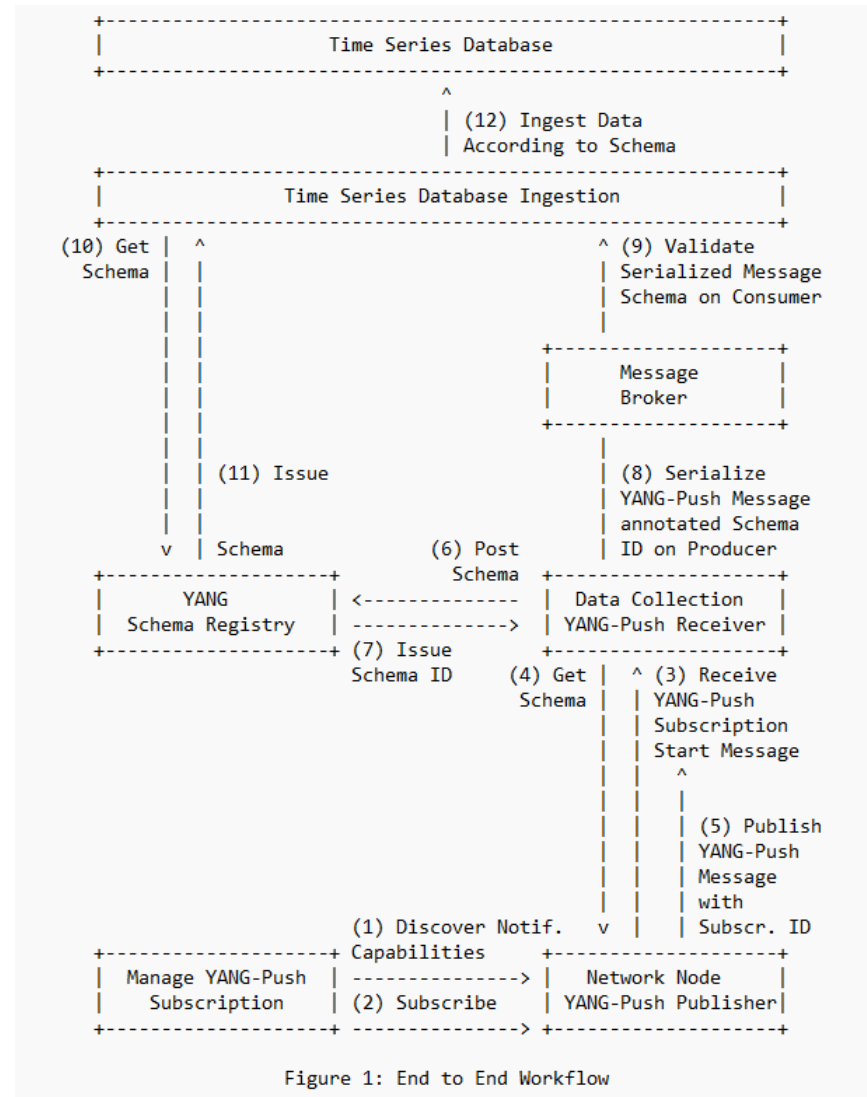


Figure 1: End to End Workflow

- **Network Orchestration** subscribes to YANG datastore.
- **Network Node** informs Data Collection on subscription state and publishes YANG metrics with YANG-Push.
- **Data Collection** obtains for each subscription the YANG module dependencies and the YANG modules on the network node, registers it in the YANG Schema Registry and prefixes the forwarded YANG notifications with the obtained schema ID.
- **YANG Schema Registry** issues for a Message Broker subject a schema ID for each new schema tree, compares a new schema tree with an existing and versions it.
- **Time Series Database Ingestion** consumes YANG-Push notifications from Message Broker, obtains schema tree from YANG schema registry, validates YANG notifications against schema and uses schema to populate into database table.

# Address YANG Specification and Integration Gaps

Aiming for an automated data processing pipeline

## YANG Specifications Gaps:

- YANG model for NETCONF Event Notifications  
[draft-ahuang-netconf-notif-yang](#) ----->
- Validating anydata in YANG Library context  
[draft-aelhassany-anydata-validation](#) ----->

## YANG Integration Gaps:

- Support of Network Observation Timestamping in YANG Notifications  
[draft-tgraf-netconf-yang-push-observation-time](#) ----->
- Support of Hostname and Sequencing in YANG Notifications  
[draft-tgraf-netconf-notif-sequencing](#) ----->
- Support of Versioning in YANG Notifications Subscription  
[draft-ietf-netconf-yang-notifications-versioning](#) ----->
- Augmented-by Addition into the IETF-YANG-Library  
[draft-linclanetconf-yang-library-augmentation](#) ----->

« Addressing those gaps are a prerequisite to enable an automated data processing chain as described in [draft-ietf-nmop-yang-message-broker-integration](#).

**Please consider to attend IETF 120 NMOP working group session on Friday 13:00 – 15:00 or go onto the mailing list and contribute to the discussion. »**