Network Anomaly Detection Lifecycle and Semantics

draft-ietf-nmop-network-anomaly-lifecycle-02
draft-ietf-nmop-network-anomaly-semantics-02

NMOP WG, Monday 17th March 2025

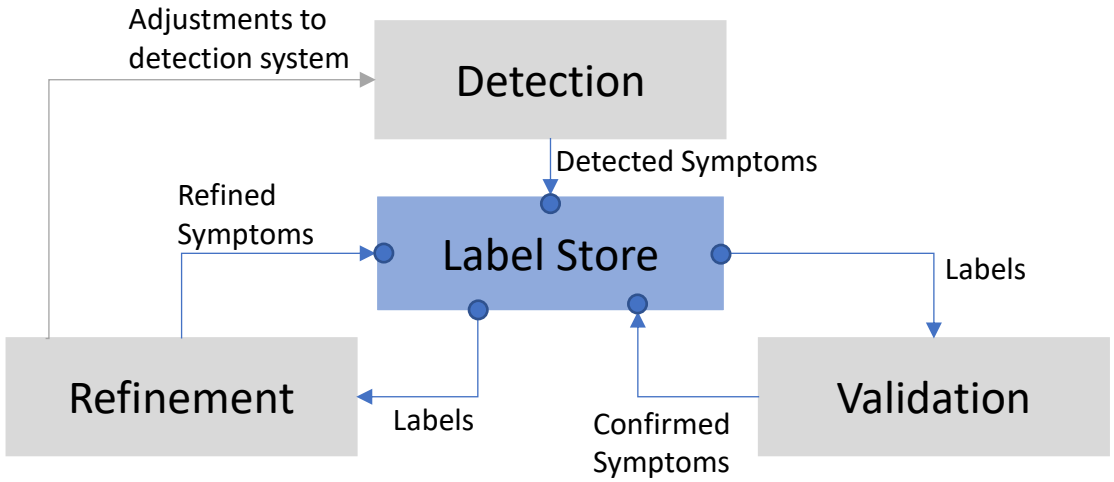
IETF 122 - Bangkok

Vincenzo Riccobene (Huawei), Thomas Graf and Wanting Du (Swisscom), Alex Huang Feng (Insa Lyon)

Summary: “An Experiment: Network Anomaly Lifecycle”

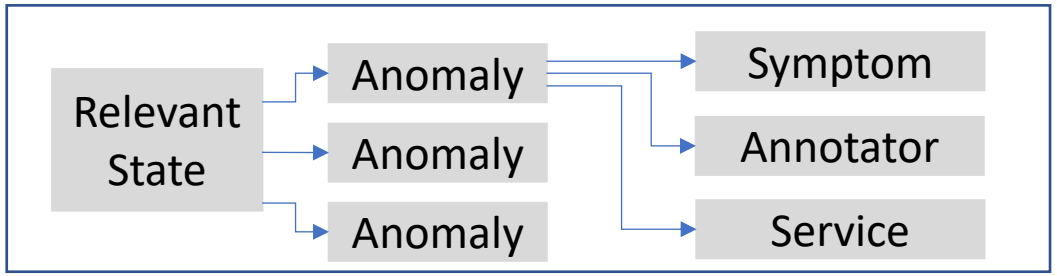
https://datatracker.ietf.org/doc/draft-ietf-nmop-network-anomaly-lifecycle/

- This draft defines the **lifecycle, generic data models and APIs** to be used for Network Anomaly Detection Post-mortem analysis of network incidents
- The lifecycle consists of **three stages**. Data is collected and revised across the three stages by using a **label store**, for which a data model and an API is defined in the draft



- **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 operators perform postmortem analysis of incidents, analyze the telemetry data and detected anomalies with the objective to identify useful adjustments in the data collection and Anomaly Detection system.

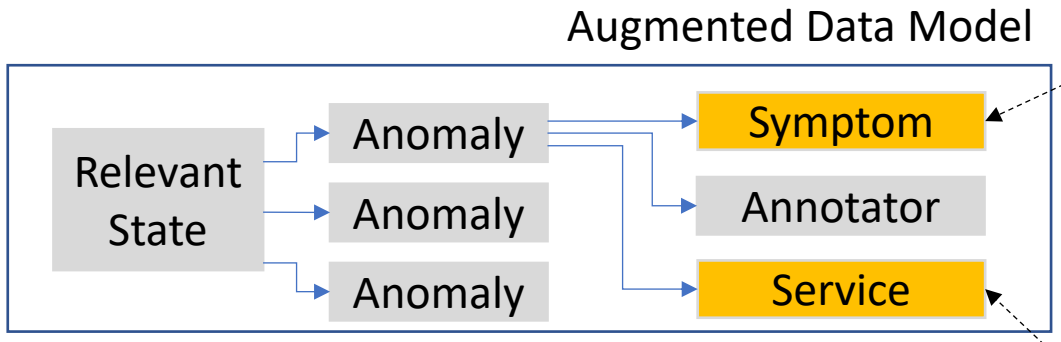
Data Model Representation



Summary: “Semantic Metadata Annotation for Network Anomaly Detection”

https://datatracker.ietf.org/doc/draft-ietf-nmop-network-anomaly-lifecycle/

- This draft provides a **detailed semantic** to describe the **output of Service Disruption Detection** in a way that can be easy for Network Engineers to understand the underlying symptoms in a deterministic and semantically structured way.
- The result is an **augmentation of the Lifecycle data model**, supporting the characterization of symptoms for connectivity services and the determinist mapping of operational and analytical data for those services



Define the symptom as a combination of:

- **Action**: What action has the network node performed, in connection to the symptom (e.g. a packet was dropped)
- **Reason**: Why the network node performed that action (e.g. the destination is unreachable)
- **Cause**: What caused that reason to happen in the first place (e.g. Time To Live expired)

Define details related to the service, to enable the mapping to operational data

Main updates to the documents

- Removed Antonio Roberto from the author list, as requested from him
- Fixed most of the YANG errors from previous version

Yang Validation:

The remaining errors require some more drastic changes to the models. We want to validate those with code for the next iteration

- Made changes to the YANG models, as necessary, based on the work we did on the model validation (see next slides)

Lifecycle Data Model – Main updates

```
module: ietf-relevant-state
  +--rw relevant-state
     +--rw id                  yang:uuid
     +--rw description?        string
     +--rw start-time          yang:date-and-time
     +--rw end-time?           yang:date-and-time
     +--rw concern-score       score
     +--rw anomalies* [id version]
        +--rw id                    yang:uuid
        +--rw uri?                  inet:uri
        +--rw version               yang:counter32
        +--rw state                 identityref
        +--rw description?          string
        +--rw start-time            yang:date-and-time
        +--rw end-time?             yang:date-and-time
        +--rw confidence-score      score
        +--rw pattern?              identityref
        +--rw annotator!
        |  +--rw name                 string
        |  +--rw (annotator-type)?
        |     +--:(human)
        |     |  +--rw human?         empty
        |     +--:(algorithm)
        |        +--rw algorithm?     empty
        +--rw symptom!
        |  +--rw id                 yang:uuid
        |  +--rw concern-score      score
        +--rw service!
           +--rw id    yang:uuid
```

Having the concern score only at the symptom level is not enough. We also need a “global” concern score for the Relevant State

We need a link to the visualization of the symptom

Defining the pattern as an identity ref allows the user of the model to augment it easily, as needed.

Semantics Data Model – Main updates

```
+--rw symptom!
|  +--rw id                                yang:uuid
|  +--rw concern-score                     score
|  +--rw smcblsymptom:action?              string
|  +--rw smcblsymptom:reason?              string
|  +--rw smcblsymptom:cause?               string
|  +--rw (smcblsymptom:plane)?
|     +--:(smcblsymptom:forwarding)
|     |  +--rw smcblsymptom:forwarding?    empty
|     +--:(smcblsymptom:control)
|     |  +--rw smcblsymptom:control?       empty
|     +--:(smcblsymptom:management)
|        +--rw smcblsymptom:management?    empty
```

```
+--rw service!
   +--rw id
   |       yang:uuid
   +--rw smtopology:vpn-service-container
   |  +--rw smtopology:vpn-service* [vpn-id]
   |     +--rw smtopology:uri?                inet:uri
   |     +--rw smtopology:vpn-id              string
   |     +--rw smtopology:vpn-name?           string
   |     +--rw smtopology:site-ids*           string
   |     +--rw smtopology:change-start-time?
   |     |       yang:date-and-time
   |     +--rw smtopology:change-end-time?
   |     |       yang:date-and-time
   |     +--rw smtopology:change-id?          yang:uuid
   +--rw smtopology:vpn-node-termination-container
      +--rw smtopology:vpn-node-termination*
               [hostname route-distinguisher]
         +--rw smtopology:hostname              inet:host
         +--rw smtopology:route-distinguisher   string
         +--rw smtopology:peer-ip*
         |       inet:ip-address
         +--rw smtopology:next-hop*
         |       inet:ip-address
         +--rw smtopology:interface-id*         int32
```

We need a link to the visualization of the symptom

In some cases, changes to the network inventory and maintenance windows can generate some false alarms. These fields provide a reference to any change that needs to be tracked, to make the information available to the user and/or tag them during post-mortem analysis.

“hackathon” - Integration work

Mapping data model on real data from Network Anomaly Detection

- We are developing Antagonist (https://github.com/vriccobene/antagonist), an open source label store to persist and expose network anomaly detection labels, from the various actors involved in the lifecycle (both humans and algorithms).
- Current stage: a detailed analysis of the mapping between the network incident data and the defined YANG data model is on going, to identify any further missing or misplaced fields
  - The changes proposed in this iteration are the result of the first part of this analysis, which will continue in the next few months
- Next stage: test the Label Store with the actual data for network incidents from the network
- We plan to provide more insights at the next IETF 123, and to get closer to the final version of the models

Mapping between network data and current Lifecycle model

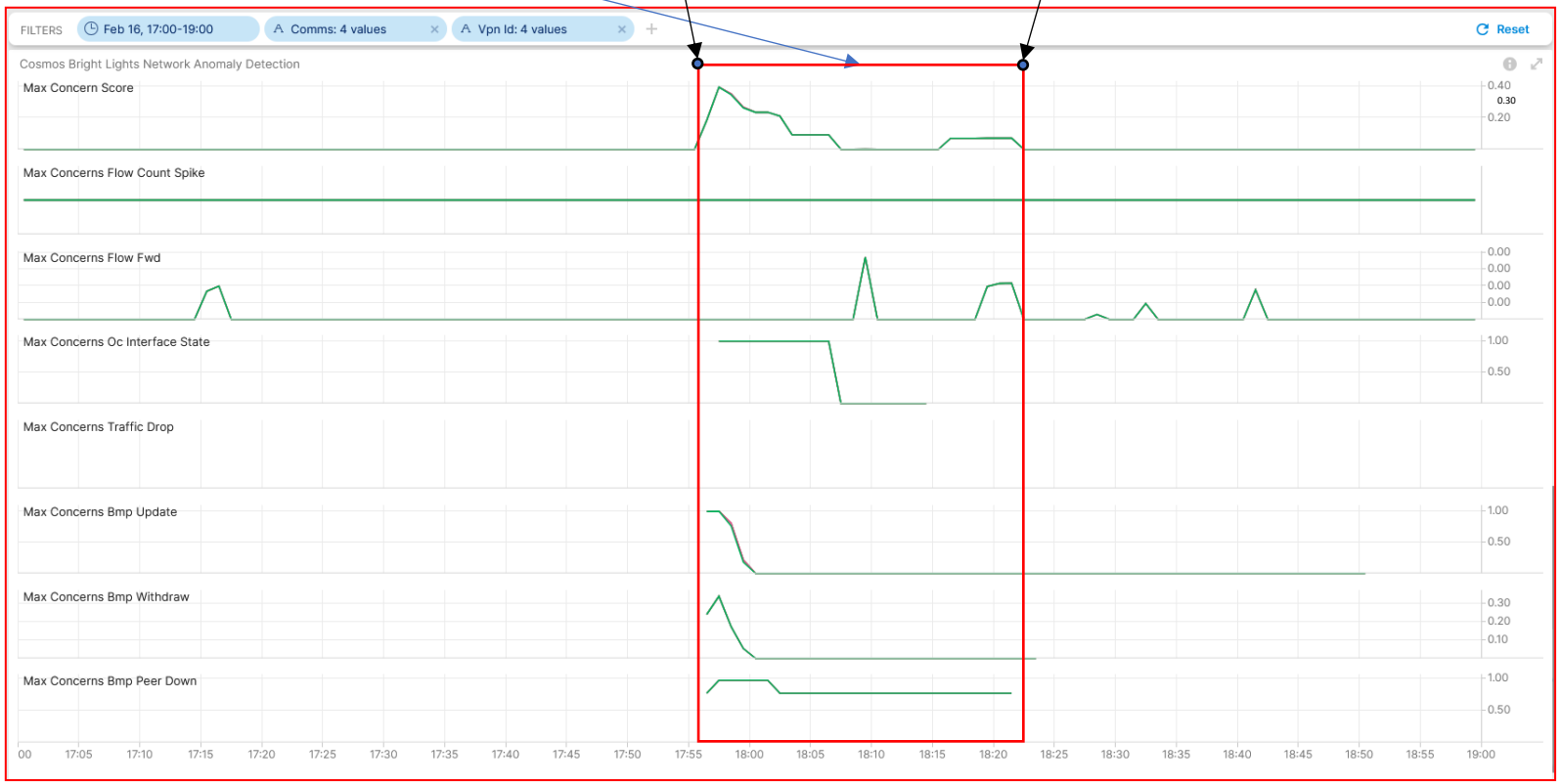
Network Anomaly Lifecycle Management YANG Model

```
module: ietf-relevant-state
  +--rw relevant-state
     +--rw id                    yang:uuid
     +--rw description?          string
     +--rw start-time            yang:date-and-time
     +--rw end-time?             yang:date-and-time
     +--rw concern-score         score
     +--rw anomalies* [id version]
        +--rw id                       yang:uuid
        +--rw uri?                     inet:uri
        +--rw version                  yang:counter32
        +--rw state                    identityref
        +--rw description?             string
        +--rw start-time               yang:date-and-time
        +--rw end-time?                yang:date-and-time
        +--rw confidence-score         score
        +--rw pattern?                 identityref
        +--rw annotator!
        |  +--rw name                     string
        |  +--rw (annotator-type)?
        |     +--:(human)
        |     |  +--rw human?             empty
        |     +--:(algorithm)
        |        +--rw algorithm?         empty
        +--rw symptom!
        |  +--rw id                     yang:uuid
        |  +--rw concern-score          score
        +--rw service!
           +--rw id    yang:uuid
```

Relevant State

start-time

end-time

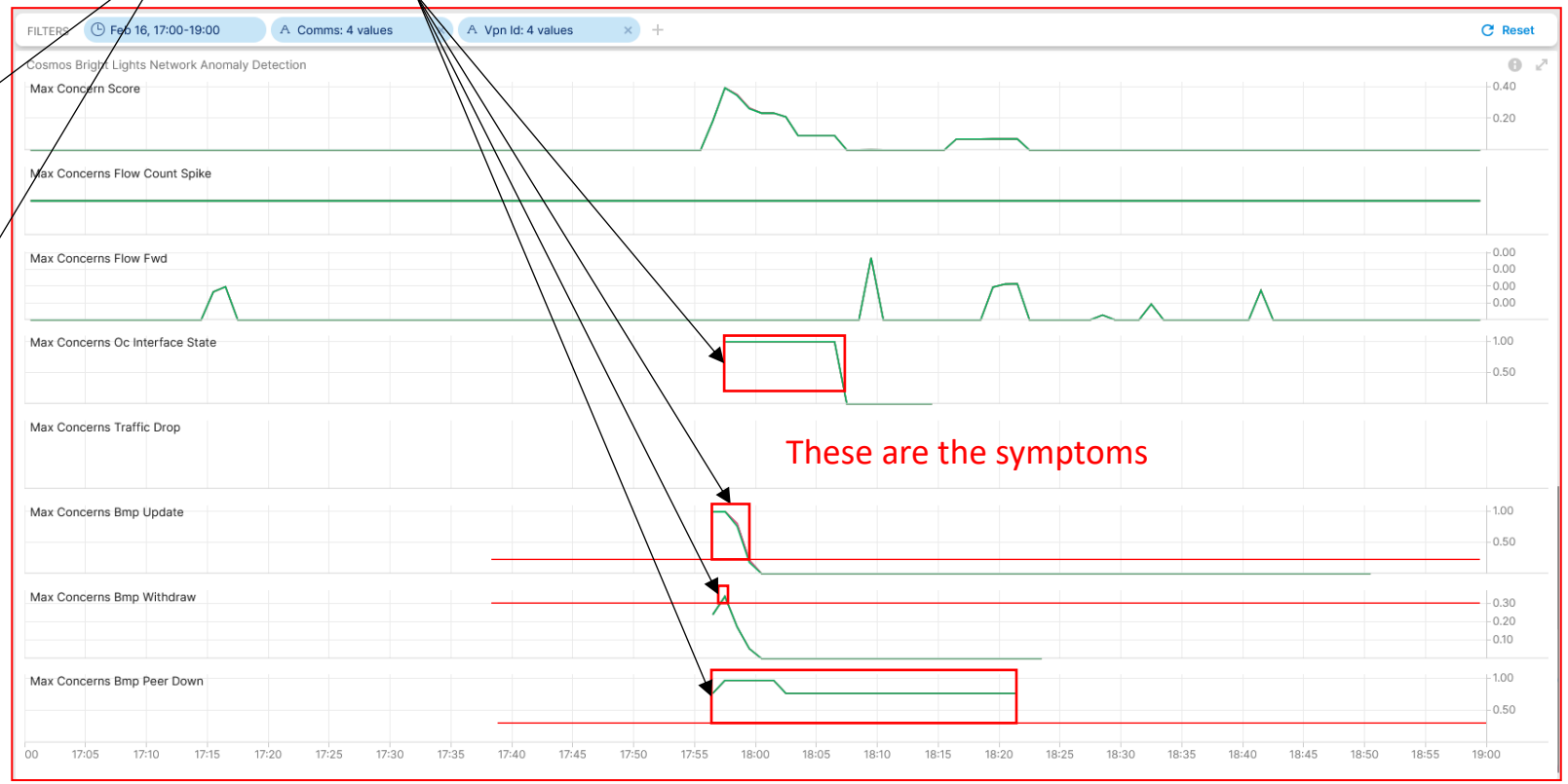


Mapping between network data and current Lifecycle model

Network Anomaly Lifecycle Management YANG Model

```
module: ietf-relevant-state
  +--rw relevant-state
     +--rw id                    yang:uuid
     +--rw description?          string
     +--rw start-time            yang:date-and-time
     +--rw end-time?             yang:date-and-time
     +--rw concern-score         score
     +--rw anomalies* [id version]
        +--rw id                       yang:uuid
        +--rw uri?                     inet:uri
        +--rw version                  yang:counter32
        +--rw state                    identityref
        +--rw description?             string
        +--rw start-time               yang:date-and-time
        +--rw end-time?                yang:date-and-time
        +--rw confidence-score         score
        +--rw pattern?                 identityref
        +--rw annotator!
        |  +--rw name                     string
        |  +--rw (annotator-type)?
        |     +--:(human)
        |     |  +--rw human?             empty
        |     +--:(algorithm)
        |        +--rw algorithm?         empty
        +--rw symptom!
        |  +--rw id                     yang:uuid
        |  +--rw concern-score          score
        +--rw service!
           +--rw id    yang:uuid
```

Anomalies & Symptoms



Next Steps

- ❑ Finalize the mapping of the YANG model with data from other real network incidents
- ❑ Update YANG models as needed, based on the analysis
- ❑ Finalize open source implementation of Antagonist, based on findings from previous steps
- ❑ Integrate and run Antagonist on the Swisscom Lab and validate the data models and the APIs
- ❑ Finalize the YANG models, based on the evidence collected by the hackathon activities

Discussion Points

- Any Feedback on the documents?
- Would any other operator like to test any new Network Anomaly Detection use case with Antagonist?

Thanks!