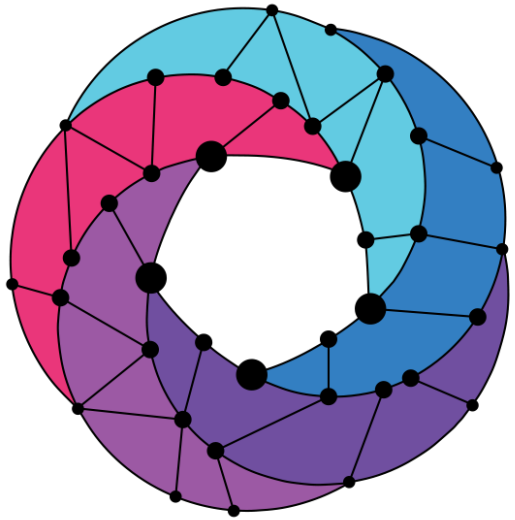


Open Source MANO

Subscription And Notification Support In OSM

K Sai Kiran

Zero Touch Specialist – TATA ELXSI

© ETSI 2020

Agenda



- Problem Statement
- Design Principle
- Design Principle Realization
- High Level Flow
  - Subscription
  - Notification
- Current Support And Future Road Map
- Conclusion

© ETSI 2020

Problem Statement

- ETSI NFV SOL005 defines a class of Northbound APIs through which entities can subscribe for changes in the Network Service (NS) life-cycle. The entities get notified via REST APIs which those entities expose.
- The entities which are interested to know the life-cycle changes of network service are called Subscribers. Subscribers receive messages called notifications when an event of their interest occurs.
- SOL005 specifies usage of filters in the registration phase, through which subscribers can select events and NS they are interested in.
- Subscribers can choose the authentication mechanism of their APIs.
- Events need to be notified with very little latency and make them near real-time.
- Deregistration of subscription should be possible however subscribers cannot modify existing subscriptions as per SOL005. (Allowed verbs: GET, POST, DELETE; Invalid verbs: PUT, PATCH)

© ETSI 2020

Design Principle

Near Real time Notification :Faster/Quick Notification To External System On Event Occurrence

Data Transformation at Source



Transform the data to format as it will be consumed. Subscription data is cleaned & transformed during subscription registration.

Pre-check On End Point Health



Broken end point will add extra latency during notification processing pipeline.

No Redundant Information



Redundant subscriber information will add latency in query processing and may cause duplicate notification to external system.

Minimal Access to Database



Query to database each time when event occurs will add latency to notification pipeline.

Large Volume handling



Huge volume of events can cause delay in notification post

Notification Prioritization



1st notification posting and retries shall carry different priority

© ETSI 2020

Design Principle Realization

Near Real time Notification :Faster/Quick Notification To External System On Event Occurrence

Data Transformation At Source



Transform the data to format as it will be consumed. Subscription data is cleaned & transformed during subscription registration.

Fast look-up tables are created based on ingested data with faster mongo indexing

Pre-check On End Point Health



Broken end point will add extra latency during notification processing pipeline.

End point health check as pre-requisite for successful subscription registration

No Redundant Information



Redundant subscriber information will add latency in query processing and may cause duplicate notification to external system.

Duplicate subscription request detection and rejection

Minimal Access to Database



Query to database each time when event occurs will add latency to notification pipeline.

LRU based cache or redis

Large Volume Handling



Huge volume of events can cause delay in notification post

Kafka load balances events to each NBI

Notification Prioritization

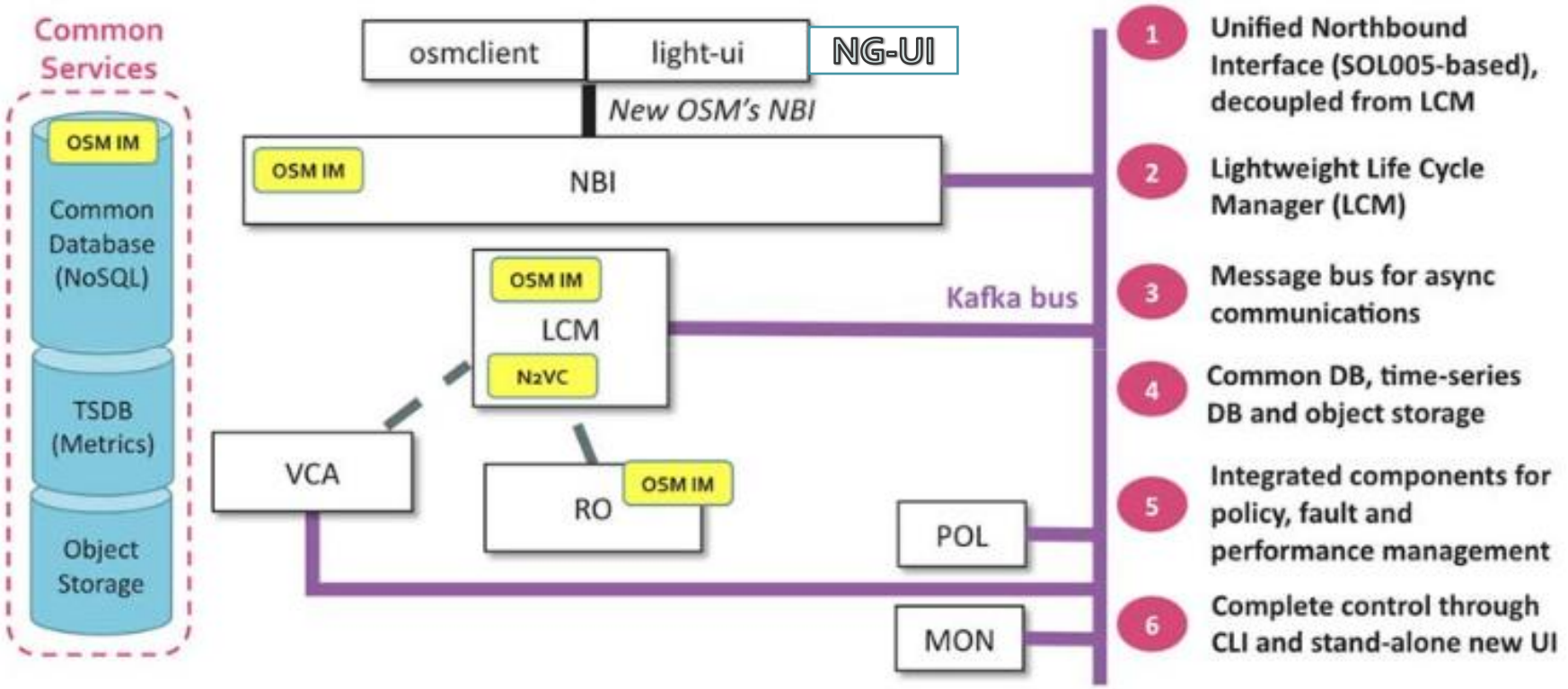


1st notification posting and retries shall carry different priority

New notifications have high priority, failed notification will be retired after back off delay

© ETSI 2020

High level flow



Source: OSM Whitepaper

© ETSI 2020

High level flow - Subscription Steps



1. Subscriber's input is validated against JSON Schema.
2. The receiver endpoint is validated; the receiver should send SOL005 compliant acknowledgment.
3. Detect duplicate subscriptions.
    1. Redundant subscriber information will add latency in query processing and may cause a duplicate notification to the external system. This may overwhelm the external system.
    2. Each subscriber has a filter object, these filter objects are complex JSON, with multiple nesting and can be in any order.
    3. Since filters are complex JSON objects, we have engineered a recursive algorithm which will bring lexicographic order at all nesting levels. This simplifies duplicate subscriber detection.

© ETSI 2020

High level flow - Subscription Steps




```
{
   "filter":{
      "nsInstanceSubscriptionFilter":{
         "nsdIds":[
            "f445b11a-63d8-44b3-85a8-b4b864ccccd6"
         ]
      },
      "notificationTypes":[
         "NsLcmOperationOccurrenceNotification"
      ],
      "operationTypes":[
         "INSTANTIATE",
         "TERMINATE"
      ],
      "operationStates":[
         "PROCESSING",
         "COMPLETED",
         "FAILED"
      ]
   },
   "CallbackUri":"http://localhost:3030/notify",
   "authentication":{
      "authType":"basic",
      "paramsBasic":{
         "userName":"user",
         "password":"user"
      }
   }
}
```


Sample Subscription Data

4. Data reformatting
    1. The subscription filter data is transformed as it is begin consumed.
    2. This step helps to perform fast query, which is done when an event occurs.

© ETSI 2020

High level flow - Notification Steps

- Kafka consumer running in OSM NBI with consumer group id nbi-server will get data of any NS lifecycle change.
  - In case of multiple NBI exists due to scaling, and then Kafka will balance messages sent to each NBI.
- Validate the message from Kafka and check if its relevant to NS life-cycle. If it is relevant to NS lifecycle then we filter and format the message for further processing.
- Generate an optimal query that will return a list of subscribers interested in this event.
- Generate payload and authentication headers in the SOL005 compliant structure for each subscriber according to their filters and authentication types.
- Asynchronously perform an HTTP POST to send notifications to all relevant subscribers.
- Notifications which have failed will be retires 5 times, with exponential back-off delay.

© ETSI 2020

Current Support And Future Road Map

Current Support OSM Rel 8

- Subscriptions for NS lifecycle
  - JSON schema validation.
  - Pre-chech of notification endpoint.
  - Duplicate subscription detection.
  - Data reformatting at source.
- Notifications for NS lifecycle
  - Optimal query to get list of subscribers.
  - SOL005 compliant structure for each subscriber according to their filters and authentication types.
  - POST events to notification endpoints.
  - Retry and backoff for failed notifications.

Future Release

- Subscription and notification support for NSD.
- Subscription and notification support for VNFD.
- Support for OAuth and TLS authentication types for notification endpoint.
- Cache to store subscribers.
- Improved messages in Kafka regarding events.

© ETSI 2020

Conclusion

- By subscribing to the dynamic changes in the network, operators can now take the necessary steps to assure customers service continuity in real time.
- The subscription mechanism is completely configurable from the end-user perspective, in terms of the choice of services to follow and the choice of lifecycle events.
- Subscription & Notification feature is enabler for E2E service assurance.

© ETSI 2020

Thank You



Open Source MANO

Subscription And Notification Support In OSM

© ETSI