Metadata-based Aggregation of Telemetry Flows

Sonia Fernández, Diego R. López

NMRG virtual meeting
14.04.2020
Index

1. Introduction

2. Context Information Management

3. Telemetry Metadata

4. Conclusions
Introduction
The Scenario

- Data is the essential intake for any closed-loop system
  - Avoid poisoning: correct metadata
  - Avoid starvation: sufficient streaming
- Deal with heterogeneity
  - Multi-technology
  - Multi-domain
  - Multi-vendor
  - Multi-…
- And the lessons learned can be applied to action flows in the future, leveraging SDN
  - Intent interfaces
  - Capability models
The goal

• Support the integration of different data flows
  • Open
  • Automated
  • Secure
  • Scalable

• Deal with heterogeneity at all levels
  • Data sources
  • Data models
  • Deployment styles
  • Supporting infrastructures

• A semantic metadata framework for telemetry data
  • Founded on the current results in data model space
Applying a Semantic Model to Telemetry

• Use the model to describe data flows
  • Sources
  • Consumers
  • Elements in the flow
• And including
  • The identification of the relationships to the flow data model
  • Provenance metadata
  • Security
• Note we are not talking about modeling the whole systems
  • Only the data they provide and/or consume
  • Usable to analyze and normalize flows
  • Without the need of explicit standard alignment
• Extend descriptors
  • Include a protocol for registration, announcements, etc.
Context Information Management
Introducing ISG CIM

• Focused on mechanisms to deal with context information from many different sources
  • Sharing that information through interoperable data publication platforms.
  • Agnostic to the architecture of the applications sharing information
  • Based on an information model describing entities and relationships
• Currently focused on IoT scenarios
  • Suitable for adaptation to other ones, already documented by the ISG
CIM Architecture

• CIM specification does not define a specific architecture
• Three prototypical architectures are presented: Centralized, Distributed and Federated
• Main components:
  • Context Consumer → Request context information from the Broker (e.g. Application)
  • Context Producer/Source → Produce context information (e.g. Router)
  • Central/Distributed/Federated Broker → Response queries / Stores context information
  • Context Registry → Stores the (context) source’s context information
CIM Architecture
CIM Information Model

Entity
- + id : URI
  - Has Subject
  - Has Property

Property
- + propertyid : URI
  - Has Subject
  - Has Value

Value
- + dataType : URI

Relationship
- + relationshipId : URI
  - Has Subject
  - Has Relationship

Has Relationship
- 1..*
- 0..*

Has Subject
- 1
- 0..*

Has Object
- 1
- 0..*

Has Property
- 0..*
- 1

Has Value
- 1
- 0..*

Has Relationship
- 0..*
- 1
- 0..*
Telemetry Metadata
Telemetry Metadata

Status

• Build general patterns for metadata definition
  • Collect multiple context sources (data sources)
  • Initial definition of some information models

• Use CIM standard as a reference
  • Apply the architectural recommendations in a general architecture

• Build a Semantic Aggregator that make use of new preprocessing/messaging tools (e.g. Kafka)
Architecture

Local’s Management System

Context Source
Local Network

Semantic Aggregator
Context Registry
Context Consumers
Orchestrator

NSP’s Management System

Interworking & Exposure

Core Network + Cloud
Access Network + Edge

Context Source
NSP Network

Internet

Local’s Management System

Semantic Aggregator
Context Registry
Context Consumers
AI/ML
Big Data
Analytics
Monitoring

Local Access Network

Context Source
Local Network
Architecture

**Context Sources** (Data Sources):

- Network-based probes:
  - ICMP, HTTP, IPFIX, DNS, etc...
- YANG-based network devices
- In-band Telemetry (by means of P4, etc)
- Telemetry for Cloud infrastructures
- Optical devices
- TSDB
Information Model: Network-based probes

- **Probe**: hasProperties (name, format, is_connected_to)
- **Endpoint**: hasProperties (id, name, is_executed_by)
- **URI**: hasProperties (schema, host, port, path, query)
- **Credentials**: hasProperties (user, method, password)
- **Agent**: hasProperties (id, hostname)
- **Event**: hasProperties (category, duration, start, end, transport)
- **Network**: hasProperties (type, id, direction, packets, bytes)
- **Source**: hasProperties (mac, ip, port, bytes, packets)
- **Destination**: hasProperties (mac, ip, port, bytes, packets)

**YAML Dictionary**:
```
icmp-event:
  - id
  - version
  - status
  - request:
    - type
    - code
    - message
  - response:
    - type
    - code
    - message
```
Information Model: YANG-based devices

Device
- name
- format
- type

isconnectedTo

isdescribedBy

YANG model
- name
- properties

Endpoint
- id
- name
- hasLogin
- hasURI

URI
- schema
- host
- port
- path
- query

Credentials
- user
- method
- password

YAML Dictionary
interface:
- name
- type
- mtu
- counters
  - in-octects
  - in-pkts
  - in-unicast-pkts

module: device-interfaces
  +++rw interfaces
  +++rw interface* [name]
  +++rw name
  +++rw config
  +++ro state
    | +++ro name?
    | +++ro type
    | +++ro mtu?
    | +++ro counters
    | +++ro in-octects?
    | +++ro in-pkts?
    | +++ro in-unicast-pkts?
Conclusions
Conclusions

• Data-driven network management requires the aggregation of heterogeneous sources of data

• A semantic, metadata-based model exists for IoT context data aggregation

• We are exploring the extension of this model to network management
  • Relying on YANG and telemetry frameworks
  • On various application scenarios

• If the RG is interested, we’d be happy to document these in an I-D
  • And look for its further consolidation