# ETSI GR ENI 009 V1.2.1 (2023-05)

GROUP REPORT

# Experiential Networked Intelligence (ENI); Definition of data processing mechanisms

***Disclaimer***

The present document has been produced and approved by the Experiential Networked Intelligence (ENI) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

RGR/ENI-009v121_Data_Proc_Mech

Keywords

artificial intelligence, data collection, data management, data mechanism, data processing, data storage, network management

*ETSI*

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

## *Important notice*

The present document can be downloaded from:
https://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

## *Notice of disclaimer & limitation of liability*

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

## *Copyright Notification*

# Contents

Intellectual Property Rights ... 5

Foreword ... 5

Modal verbs terminology ... 5

Introduction ... 5

1 Scope ... 7

2 References ... 7
2.1 Normative references ... 7
2.2 Informative references ... 7

3 Definition of terms, symbols and abbreviations ... 8
3.1 Terms ... 8
3.2 Symbols ... 9
3.3 Abbreviations ... 10

4 Overview ... 11
4.1 Background ... 11
4.2 Data Precondition ... 11

5 Data Mechanism ... 11
5.1 Introduction ... 11
5.1.1 Data Mechanism Overview ... 11
5.2 Data Characteristics ... 12
5.2.1 Configuration Data ... 12
5.2.2 Sequential Data ... 12
5.2.3 Data Representation ... 13
5.2.4 Data Exchange Formats ... 13
5.2.5 Data Model from FBs in ENI System ... 14
5.2.5.1 Data model types ... 14
5.2.5.2 Data model template ... 15
5.2.5.2.1 Introduction ... 15
5.2.5.2.2 Common part of the data template ... 15
5.2.5.2.3 Specific part of the data template ... 15
5.3 Data Source ... 18
5.3.1 Introduction ... 18
5.4 Data Collection ... 19
5.4.1 Introduction ... 19
5.4.2 Data Acquisition Modes ... 19
5.4.3 Data Collection Techniques ... 19
5.4.3.1 Introduction ... 19
5.4.3.2 Data carried out in functional planes protocols ... 19
5.4.3.2.1 Data carried out in the Forwarding/User Plane ... 19
5.4.3.2.2 Data carried out in the Control Plane ... 20
5.4.3.2.3 Data carried out in the Management Plane ... 20
5.4.3.3 Specific data used to deploy telemetry ... 20
5.4.3.3.1 Network Telemetry ... 20
5.4.3.3.2 Resource Telemetry ... 21
5.4.3.3.3 Fault Telemetry ... 23
5.4.3.3.4 Streaming Telemetry ... 24
5.5 Hierarchical data storage ... 24
5.6 Data Processing ... 25
5.6.1 Data Correlation ... 25
5.6.2 Data Cleansing ... 26
5.7 Data Sharing ... 26
5.8 Data Management ... 27
5.8.1 Overview ... 27
5.8.2 Metadata Management ... 27

5.8.3 Data Security Management ....27
5.8.4 Data Quality Management ....27
5.9 Data Conversion ....27
5.9.1 Introduction ....27
5.9.2 Data Conversion between Functional Blocks ....27
5.9.3 Data Conversion between ENI AI Data Model and External System ....28
5.9.3.1 Introduction to Data Conversion between ENI AI Data Model and External System ....28
5.9.3.2 Introduction to AI Data models ....28
5.9.3.3 AI Data model types ....29
5.9.3.4 The corresponding framework of AI Data model ....29
5.9.3.5 External system data conversion ....30
5.9.3.5.1 Introduction ....30
5.9.3.5.2 Converting structured data into vectors ....31
5.9.3.5.3 Converting semi-structured data into vectors ....31
5.9.3.5.4 Converting unstructured data into vectors ....31
5.10 Data Security ....31
5.10.1 Introduction ....31
5.10.2 Artificial Intelligence Technology ....31
5.10.3 Cloud Computing Technology ....32
5.10.4 Firewall Technology ....32
5.10.5 Authentication and Authorization ....32
5.10.6 Data Encryption ....32
5.10.7 Data Masking ....32
5.10.8 Data Backup and Elasticity technology ....32
5.10.9 Data Erasure Technology ....32

6 Example Scenarios to Illustrate Data Mechanisms ....33
6.1 AI-enabled Traffic Classification Use Case ....33
6.1.1 Introduction ....33
6.1.2 Data Acquisition ....33
6.1.3 Data Processing for traffic classification ....33
6.2 Network Fault Root-Cause Analysis and Intelligent Recovery Use Case ....34
6.2.1 Introduction ....34
6.2.2 Data Acquisition ....34
6.2.3 Data Processing ....34
6.3 Intelligent Service Experience Evaluation Use Case ....35
6.3.1 Introduction ....35
6.3.2 Data Acquisition ....35

7 Example requirements ....36
7.1 Data format and interface ....36
7.2 Data Security ....36
7.2.1 Data Security introduction ....36
7.2.2 Data Collection Security example requirements ....36
7.2.3 Data Transmission Security example requirements ....37
7.2.4 Data Storage Security example requirements ....37
7.2.5 Data Processing Security example requirements ....37
7.2.5.1 Data Processing introduction ....37
7.2.5.2 Data Desensitization example requirements ....37
7.2.5.3 Data Analysis Security example requirements ....37
7.2.5.4 Data Import and Export security example requirements ....38
7.2.6 Data Exchange Security example requirements ....38
7.2.7 Data Destruction Security example requirements ....38

8 Recommendations ....38

**Annex A: Change History ....40**

History ....41

# Intellectual Property Rights

## Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

## Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

# Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Experiential Networked Intelligence (ENI).

# Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

# Introduction

The present document outlines a high-level reference framework that describes technical methods for producing high-quality actionable data efficiently and in a timely manner.

The organization of the present document is as follows:

- Clause 1 defines the scope of the present document.
- Clauses 2 and 3 provide informative references, terms, symbols and abbreviations.
- Clause 4 describes an overview of the data mechanism, including its motivation and challenges.
- Clause 5 defines components in the high-level framework of the data mechanism in terms of data acquiring and data processing.
- Clause 6 presents the data mechanisms in some example scenarios proposed in ETSI GR ENI 001 [i.1], Use Case specification.

- Clause 7 presents example requirements for data format, interface and data security.
- Clause 8 concludes possible contributions to other ENI group specifications of the present document.

Data Telemetry is used as an example for data mechanisms description and analysis.

# 1 Scope

The present document revises ETSI GR ENI 009 [i.30] (V1.1.1). The realization of intelligent network depends on the use of mechanisms related with: processing of the big data, AI algorithms and computing resources. Therefore, effective data management and operation is extremely important.

The present document is purposed to enhance the ETSI GR ENI 009 [i.30] (V1.1.1) on data operation requirements and mechanisms to better serve ENI system, especially within the following technical areas:

1) data format among the Functional Block of ENI system and towards the external world (internal Functional Blocks, Knowledge Representation);

2) data conversion and possibility to translate AI data model to be adapted to/from external system (external trained model imported into ENI);

3) consistency of data format and interface to accelerate the Autonomous Network (AN) evolution process; and

4) ensure that customer privacy is not disclosed in the entire lifecycle of data collection, processing and utilization (Federated Learning).

# 2 References

## 2.1 Normative references

Normative references are not applicable in the present document.

## 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GR ENI 001 (V3.2.1): "Experiential Networked Intelligence (ENI); ENI use cases".

[i.2] ETSI GR ENI 004 (V3.1.1): "Experiential Networked Intelligence (ENI); Terminology".

[i.3] ETSI GS ENI 005 (V3.1.1): "Experiential Networked Intelligence (ENI); System Architecture".

[i.4] IETF RFC 7011: "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information".

[i.5] IETF RFC 7950: "The YANG 1.1 Data Modeling Language".

[i.6] IETF RFC 4656: "A One-way Active Measurement Protocol (OWAMP)".

[i.7] IETF RFC 5357: "A Two-Way Active Measurement Protocol (TWAMP)".

[i.8] IETF RFC 9197: "Data Fields for In-situ OAM".

[i.9] IETF RFC 8321: "Alternate-Marking Method for Passive and Hybrid Performance Monitoring".

[i.10] IETF RFC 8889: "Multipoint Alternate Marking method for passive and hybrid performance monitoring".

[i.11] IETF RFC 7799: "Active and Passive Metrics and Methods (with Hybrid Types In-Between)".

[i.12] Recommendation ITU-T Y.1731: "OAM functions and mechanisms for Ethernet based networks".

[i.13] IETF RFC 6241: "Network Configuration Protocol (NETCONF)".

[i.14] IETF RFC 4271: "A Border Gateway Protocol 4 (BGP-4)".

[i.15] IETF RFC 7854: "BGP Monitoring Protocol (BMP)".

[i.16] IETF I-D.draft-kumar-rtgwg-grpc-protocol-00: "gRPC Protocol".

[i.17] IETF I.D.draft-zhou-ippm-enhanced-alternate-marking-12: "Enhanced Alternate Marking Method".

[i.18] IETF I.D.draft-song-ippm-postcard-based-telemetry-15: "Postcard-based On-Path Flow Data Telemetry using Packet Marking".

[i.19] IETF RFC 793: "Transmission Control Protocol (TCP)".

[i.20] IETF RFC 768: "User Datagram Protocol (UDP)".

[i.21] VNF Event Stream (VES).

[i.22] IETF RFC 3416: "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)".

[i.23] IETF RFC 959: "File Transport Protocol (FTP)".

[i.24] The Atlan Data wiki definition of structured data.

[i.25] The Atlan Data wiki definition of unstructured data.

[i.26] IETF RFC 4560: "Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations".

[i.27] Prometheus open source.

[i.28] NoSQL.

[i.29] Data model.

[i.30] ETSI GR ENI 009 (V1.1.1): "Experiential Networked Intelligence (ENI); Definition of data processing mechanisms".

# 3 Definition of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in ETSI GR ENI 004 [i.2], ETSI GS ENI 005 [i.3] and the following apply:

**column-oriented database:** database that organizes data by field

NOTE: This type of database keeps all of the data associated with a field next to each other in memory, and is optimized for online analytical processing. They are optimized for reading and computing on columnar data. Examples include Snowflake and BigQuery.

**data lake:** centralized storage repository that stores raw data that are in the form of structured, semi-structured and unstructured format

**data mart:** subset of a data warehouse focused on a particular line of business, department or subject area

**data warehouse:** repository used to connect, analyse, and report on historical and current data from heterogeneous sources

NOTE: A data warehouse is designed for query and analysis as opposed to transaction processing. It analyses and reports on data from operational systems as used in decision-support systems.

**ENI AI data model:** AI data model refers to the data involved in AI modeling process

**hadoop distributed file system:** distributed fault-tolerant file system that stores data on commodity machines and provides high throughput access

**massively parallel processing:** use of a large number of processing nodes that perform a set of coordinated tasks in parallel using a high-speed network

NOTE: The processing nodes typically are independent, and do not share memory, and typically each node runs its own instance of an operating system.

**Principal Component Analysis (PCA):** data dimensionality reduction algorithm

NOTE: The central idea of principal PCA is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set.

**prometheus:** open-source systems monitoring and alerting toolkit

NOTE: This open source is originally built at SoundCloud. Since its inception in 2012, many companies and organizations have adopted Prometheus, and the project has a very active developer and user community. It is now a standalone open source project and maintained independently of any company. To emphasize this, and to clarify the project's governance structure, Prometheus joined the Cloud Native Computing Foundation in 2016 as the second hosted project, after Kubernetes.

**protocol buffers (protobuf):** language-neutral, platform-neutral, extensible mechanism for serializing structured data

**reinforcement learning:** See ETSI GR ENI 004 [i.2] and ETSI GS ENI 005 [i.3].

**row-oriented database:** database that organizes data by record

NOTE: This type of database keeps all of the data associated with a record next to each other in memory, and is optimized for online transaction processing. An example is MySQL.

**semi-structured data:** information that does not conform to a formal data model, but does have some organizational properties that define key data (e.g. tags) that enable data to be self-describing

**software defined hardware:** software programmable hardware that is able to be reconfigured at runtime to enable near ASIC performance without sacrificing programmability for data-intensive algorithms

**structured data:** information organized in a predetermined way (a fixed format, data model or schema) within a record or a file

NOTE 1: As defined in [i.24].

NOTE 2: Structured data enables all elements to be individually addressable, and conform to a data model.

**unstructured data:** information that does not have a pre-defined data model, and does not contain properties that provide any organization or structure to its elements

NOTE: Unstructured data needs to be processed in order to find information by domain-specific applications.

**video stalling:** process during the video playback, the video is paused and waits for the buffer due to dragging or other reasons

## 3.2 Symbols

Void.

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GR ENI 004 [i.2], ETSI GS ENI 005 [i.3] and the following apply:

| | |
|---|---|
| 5G | Fifth Generation |
| AI | Artificial Intelligence |
| AS | Autonomous System |
| BGP | Border Gateway Protocol |
| BMP | BGP Monitoring Protocol |
| BSS | Business Support Systems |
| CPU | Central Processing Unit |
| CRM | Customer Relationship Management |
| EAM | Explicit Address Mapping |
| ENI | Experiential Networked Intelligence |
| FTP | File Transport Protocol |
| gNMI | gRPC Network Management Interface |
| IETF | Internet Engineering Task Force |
| IMS | Integrated Management System |
| IOAM | In-band OAM |
| IP | Internet Protocol |
| IPFIX | IP Flow Information eXport |
| IPFPM | IP Flow Performance Measurement |
| ITU | International Telecommunication Union |
| ITU-T | ITU Telecommunication standardization sector |
| JSON | JavaScript Object Notation |
| KPI | Key Performance Indicator |
| MS | Monitoring System |
| NE | Network Element |
| NMS | Network Management System |
| OAM | Operation, Administration and Maintenance |
| OMC | Operations and Maintenance Centre |
| OSS | Operations Support Systems |
| OWAMP | One-Way Active Measurement Protocol |
| PBT | Postcard-Based Telemetry |
| PCA | Principal Component Analysis |
| QoS | Quality of Service |
| SDK | Software Development Kit |
| SDN | Software-Defined Networking |
| SLA | Service Level Agreement |
| SNMP | Simple Network Management Protocol |
| SQL | Structured Query Language |
| SR-IOV | Single Root I/O Virtualization |
| TCP | Transmission Control Protocol |
| TWAMP | Two-Way Active Measurement Protocol |
| UDP | User Datagram Protocol |
| VES | VNF Event Stream |
| VNF | Virtual Network Function |
| XML | eXtensible Markup Language |
| YAML | Yet Another Markup Language |
| YANG | Yet Another Next Generation |

# 4 Overview

## 4.1 Background

Exploiting network data for intelligent network applications and use has been increasing in recent years. By combining AI and machine learning algorithms, network data is able to provide insights that help network operators better manage and optimize the network. Therefore, the quality of available sample data, for instance, time validity, diversity, volume, accuracy, plays an important role in learning from data. One challenge is that large amounts of data as well as data that meets the demands is able to be acquired. Additionally, the data collected from network equipment's from different vendors varies in the aspect of name, format, calculation rules, etc. Thus a large amount of time is often be spent to do the data normalizing, cleansing, and engineering before those data could be used to train the model. This blocks the deployment of actionable decisions, which are meant to improve ENI System performance and User Experience.

The present document describes data acquisition, sharing and processing mechanisms, as well as supports for data privacy in AI-enabled network Operation, Administration and Management (OAM). The present document identifies the sources and data to be extracted, however it does not intend to explain how the mechanisms work, or how data is processed in order to became used. This could be addressed in a later release.

## 4.2 Data Precondition

Different types of data are able to be analysed only and interpreted correctly in particular contexts. The following are examples of some of the types of data that the present document focuses on.

**Real-time data:** Typically, network data has to be continually monitored and dynamically processed in real-time. Example processing includes filtering, correlation, and cleansing. This is typically down locally and then aggregated results are distributed for further processing.

**Continuous data:** In some cases, continuous data over a long time span is required for analysis or model training. For example, historical traffic data are used to predict future traffic trends. In general, the longer the time span, the more representative it is, but the larger the data volume. Therefore, a way of efficiently processing and managing continuous data is needed.

NOTE: More consideration on "historical data" will be described in a future version in a later release.

# 5 Data Mechanism

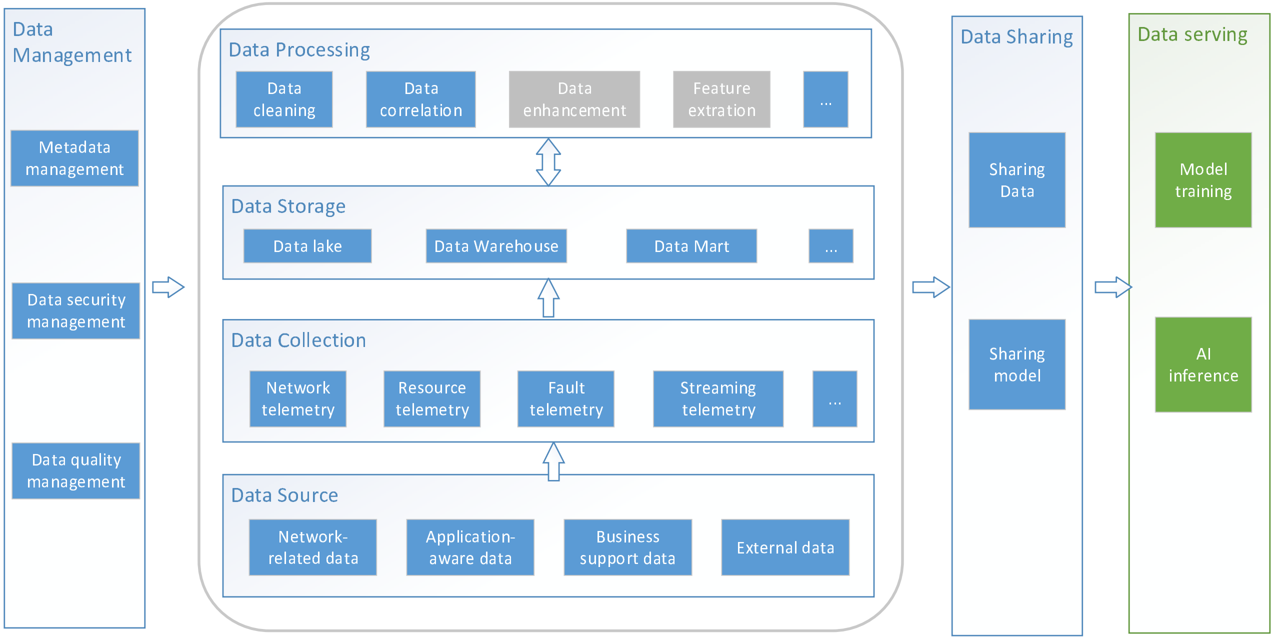
## 5.1 Introduction

### 5.1.1 Data Mechanism Overview

This clause defines components in a high-level overview for data acquisition and processing. Furthermore, this clause classifies different types of data in terms of their data sources, as well as describes data processing mechanisms, in order to support AI enabled network OAM and service management.

The Data Mechanism supports different data acquisition and processing mechanisms for data from different sources and for use by different network applications.

As shown in figure 5-1, the data mechanism overview is able to be partitioned into the following components.



NOTE: The content in grey box will be described in a future version in a later release.

**Figure 5-1: Data Mechanism Overview**

The main components above are thoroughly described and explained in the next clauses. However, before doing that, some information will be provided on the data contents characteristics, i.e. on the types of data that are able to be used to classify data as well as on the parameters that encompass each type and the scenarios where they could be found.

Telemetry is a service/application related to the collection of measurements, statistics, or other related data at pre-determined points, and the subsequent and automatic transmission of those data to appropriate devices. It will be used throughout this clause as an example of data source in order to provide some practical application to the descriptions presented in the main text.

## 5.2 Data Characteristics

### 5.2.1 Configuration Data

Configuration data are used to identify the context in which measurements are made. Table 5-1 lists some examples of configuration data that are required to be made per-user, per-service telemetry measurements, see clause 5.4.1.

**Table 5-1: Exemplary Configuration Data Characteristics**

| Configuration Data | Brief Description | Source | Scenario |
|---|---|---|---|
| Network device attribute information | Device ID, location, device model/version | Network Management Systems --> OSS | Network device alarm root cause analysis |
| Network device configuration information | Device IP, port, Vlan ID, IP Route Protocol | Network Management Systems | Intelligent traffic steering |
| Customer information | IMEI, IMSI, Terminal type, user name, user level (e.g. VIP user), register time, subscription service information | Network Management Systems --> BSS | Content Recommendation |

### 5.2.2 Sequential Data

Sequential data are a series of data recorded in time order. Table 5-2 shows some examples of sequential data.

**Table 5-2: Exemplary Temporal Data Characteristics**

| Sequential Data | Brief Description | Source | Scenario |
|---|---|---|---|
| Fault data | Alarm, log | Network Management Systems | Network device alarm root cause analysis |
| Performance data | CPU, memory, and I/O usage memory | Network infrastructure --> servers | KPI anomaly analysis |
| Network traffic data | Throughput, rate, delay | Network infrastructure --> switches, routers | Traffic prediction |
| External environment data | Temperature, humidity | External sources --> sensors | Device equipment energy saving |

## 5.2.3 Data Representation

Data is able to be classified into structured, semi-structured, and unstructured data formats according to whether the data can be expressed in a uniform structure.

Structured data is information organized in a predetermined way (a fixed format, data model or schema) within a record or a file [i.24]. Structured data enables all elements to be individually addressable, and conform to a data model. Table 5-3 shows some examples of structured data in the network.

**Table 5-3: Exemplary Structured Data Characteristics**

| Structured Data | Brief Description | Source | Scenario |
|---|---|---|---|
| Relational Data | Data structured that adheres to a pre-defined data model | SQL database | Customer information |

Semi-structured data is information that does not conform to a formal data model, but does have some organizational properties that define key data (e.g. tags) that enable data to be self-describing.

**Table 5-4: Exemplary Semi-structured Data Characteristics**

| Semi-Structured Data | Brief Description | Source | Scenario |
|---|---|---|---|
| XML Data | Data that has some organizational properties | XML Data Store | Some types of Network Data |

Unstructured data is information that does not have a pre-defined data model, and does not contain properties that provide any organization or structure to its elements. It will be pre-processed in order to find information by domain-specific applications [i.25]. Table 5-5 shows some examples of unstructured data in the network.

**Table 5-5: Exemplary Unstructured Data Characteristics**

| Unstructured Data | Brief Description | Source | Scenario |
|---|---|---|---|
| Word®, PDF®, or Text Documents, Media Files | Data that does not have a pre-defined data model | BSS | Content (e.g. streaming media) |

## 5.2.4 Data Exchange Formats

When using interfaces for data exchange between functional blocks, according to what is defined in ETSI GR ENI 004 [i.2] three data formats are usually used: JSON, XML and YAML:

- **JSON:** It is a lightweight text data exchange format, which is syntactically the same as the code for creating JavaScript objects and consists of key&value.

- **XML:** It is an extensible markup language, a subset of the standard universal markup language, and a markup language used to mark electronic files to make them structured.

- **YAML:** It is an intuitive data serialization format that can be recognized by the computer.

## 5.2.5 Data Model from FBs in ENI System

### 5.2.5.1 Data model types

A Data Model is an abstraction used to represent real world entities, the relationship between these entities and the operations that can be performed on the data. In ENI System, the data model determines how a FB encodes its data and then can be seen and understood by other FBs.

There are three basic data models in the process of data development, they are Hierarchical Model, Network model and Relational Model. The three models are named after their data structures. Hierarchical Model and Network model use structured data. Relational models are unstructured data structures. The basic structure of the hierarchical model is a tree structure; the basic structure of the network model is an undirected graph without any restrictions. The relational model is an unformatted structure, and a single two-dimensional table structure is used to represent the relationship between entities and entities. The relational model is a commonly used data model in the current database.

**Hierarchical Model:** Organize the data into a one-to-many relationship structure, and use a tree structure to represent entities and the connections between entities.

**Network Model:** Using connection instructions or pointers to determine the mesh connection relationship between data is a many-to-many type of data organization.

**Relational Model:** Organize data in the form of record groups or data tables, so as to use the relationship between various entities and attributes for storage and transformation, without hierarchy or pointers, it is a very effective way to establish the relationship between spatial data and attribute data. The data organization method.

The relational model, Key-Value model, Document model, Column-oriented model and Graph model are the main data model that the storage technology can support.

**The relational model:** It is the dominant data model. Records are stored in tables. Tables are defined by a schema, which determines what columns are in the table [i.29].

**Key-value model:** The data model consists of key and value pairs. The key is the unique identifier of the data received by the functional block, and the value represents different information of the data. A data message can only include one key and multiple values.

**Document model:** It assumes that documents encapsulate and encode data (or information) in some standard formats or encodings. Unlike a relational model, document model usually supports nested structures. Unlike a key-value model, document model is aware of the internal structure of the document [i.28].

**Column-oriented model:** It is like relational model, except that they flip the data around. Instead of storing records, column model stores all the values for a column together in a stream. An index provides a means to get column values for any particular record [i.28].

**Graph model:** This can be used to model things like social graphs (people are represented by vertices, and their relationships are the edges), or real-world objects (components are represented by vertices, and their connectedness is represented by edges) [i.28].

**Table 5-6: Comparison of data model [i.28]**

| Data Model | Performance | Scalability | Flexibility | Complexity | Functionality |
|---|---|---|---|---|---|
| Relational | Variable | Variable | Low | Moderate | Relational algebra |
| Key-value | High | High | High | Low | Variable (none) |
| Document | High | Variable (high) | High | Low | Variable (low) |
| Column-oriented | High | High | Moderate | Low | Minimal |
| Graph | Variable | Variable | High | High | Graph theory |

### 5.2.5.2 Data model template

#### 5.2.5.2.1 Introduction

The ENI system includes three types of functional modules: Input Processing, Analysis, and Output Generation. Input processing consists of two Functional Blocks: the Data Ingestion and the Normalization Functional Blocks. Analysis consists of six Functional Blocks: Knowledge Management, Context-Aware Management, Cognition Management, Situation Awareness, Model Driven Engineering, and Policy Management Functional Blocks. Output Generation consists of two Functional Blocks: Denormalization and Output Generation Functional Blocks.

This clause describes the possible template of the data from function modules blocks of the ENI system. The data template consists of a common part and Specific part. The common part describes the basic information of the data and the Specific part describes the actual content of the data. Different FBs have the same common part and different Specific parts.

#### 5.2.5.2.2 Common part of the data template

The common part is used to describe the basic information, including the owner, data abstract, timestamp, Time To Live, source and destination, etc. No matter which FB the data comes from, it should contain the common part, in order for the data to be properly routed to destination FB over the bus, and then be better understood and used by destination FB.

**Table 5-7: Common part of the data**

| Data Field | Brief Description | Field Type |
|---|---|---|
| Data Identifier | The unique identification of data information | Key/Character |
| Data Owner | Introduction to data Sources, like entity, domain, etc. | Value/String |
| Timestamp | Data acquisition time | Value/Date |
| Time To Live | How long the data is available for use | Value/Number |
| Source | The FB where data comes from | Value/String |
| Destination | The FB to which the data is going | Value/String |
| Data abstract | A brief introduction to the data and its purpose | Value/String |
| Option | Some other auxiliary information | Value/String |

#### 5.2.5.2.3 Specific part of the data template

##### 5.2.5.2.3.1 Specific part of the data from input Functional Block

The data model from the input function block is the data processed by the sub-function block of the input function block. The data normalization functional block [i.28] translates the data into a standardized form, using the pre-defined data structures. Therefore, the data processed by the data normalization functional block is structured data.

The key of data format between Input Processing and Analysis is data number, which is a string with a maximum length of 255 characters. Since the data used by the Functional Blocks in an ENI System typically comes from different sources, the data format of the Input Processing should indicate the data source.

**Table 5-8: Specific part of the data from Input Processing**

| Data Field | Brief Description | Field Type |
|---|---|---|
| Data Value | Description of accepted data value | Value/String |
| Data Label | A label added to data during Analysis | Value/String |

#### 5.2.5.2.3.2 Specific part of the data from Knowledge Management Functional Block

In the analysis function block, the Knowledge Management Functional Block is to represent information about both ENI System and the managed system, which includes the differentiation between acknowledged facts, hypotheses and inferences. Knowledge Management Functional Block is utilized by all other functional blocks of the ENI System. The proposals or commands generated by the other sub-function blocks of analysis function block need to be uniformly sent to the output function block through the knowledge management function block.

**Table 5-9: Specific part of the data from Knowledge Management FB**

| Data Field | Brief Description | Field Type |
|---|---|---|
| Data Value | Description of accepted data value | Value/String |
| Data Label | A label added to data during Analysis | Value/String |
| Knowledge | Knowledge stored in ENI system, including facts, axioms and inferences | Value/String |
| Model Information | Description of Model. Model information stored in ENI system | Value/String |

#### 5.2.5.2.3.3 Specific part of the data from Context-Aware Management Functional Block

Context-awareness enables a system to gather information about itself and its environment. This enables the system to provide personalized and customized services and resources corresponding to that context. Data model from Context-Aware Management Functional Block consists of measurement data, inferred knowledge and policy.

**Table 5-10: Specific part of the data from Context-Aware Management FB**

| Data Field | Brief Description | Field Type |
|---|---|---|
| Data Value | Description of accepted data value | Value/String |
| Data Label | A label added to data during Analysis | Value/String |
| Knowledge | Knowledge stored in ENI system, including facts, axioms and inferences | Value/String |
| Attributes Number | Number of data attributes | Value/Character |
| Attribute Characteristics | Data attribute description, such as type, range, missing, etc. | Value/String |

#### 5.2.5.2.3.4 Specific part of the data from Cognition Framework Functional Block

Cognition is defined to process new data, information and knowledge along with new inferences and compare those to previously stored knowledge. Cognition Framework Functional Block is to process data and information to update the existing knowledge or add new knowledge corresponding to the information. Thus, it can reflect configuration information and the specific processing method of data and information.

**Table 5-11: Specific part of the data from Cognition Framework FB**

| Data Field | Brief Description | Field Type |
|---|---|---|
| Data Value | Description of accepted data value | Value/String |
| Exist Knowledge | Existing knowledge or rules in Knowledge Management FB | Value/String |
| Computational Framework | Computational framework used to data analyse | Value/String |
| Computational Configuration | Computational parameters and operation mode | Value/String |
| New knowledge | New knowledge generated after processing | Value/String |

#### 5.2.5.2.3.5 Specific part of the data from Situational Awareness Functional Block

The function of situational awareness functional block is perception of data and behaviour that pertain to the relevant circumstances and/or conditions of a system or process, the comprehension of the meaning and significance of these data and behaviours, and how processes, actions, and new situations inferred from these data and processes are likely to evolve in the near future to enable more accurate and fruitful decision-making.

Table 5-12: Specific part of the data from Situational Awareness FB

| Data Field | Brief Description | Field Type |
|---|---|---|
| Data Value | Description of accepted data value | Value/String |
| Exist Knowledge | Existing knowledge or rules | Value/String |
| Situational prediction | Predicting the most likely evolution of the current situation | Value/String |

###### 5.2.5.2.3.6 Specific part of the data from Model Driven Engineering Functional Block

The function of the Model Driven Engineering Functional Block is to decide how to implement the selected actions from the Situational Awareness Functional Block. It uses model-driven engineering mechanisms to convert the actions into a form that enables imperative, declarative, and/or intent policies to be constructed by the Policy Management Functional Block. Model Driven Engineering Function Block realizes information and data processing based on one or more data models. Thus, it mainly includes data, model and policy.

NOTE: Further investigation is required on the presence of the Policy identity, Policy type and Policy information fields between various functional blocks and the model driven engineering.

Table 5-13: Specific part of the data from Model Driven Engineering FB

| Data Field | Brief Description | Field Type |
|---|---|---|
| Attributes Number | Number of data attributes from Context-Aware Management FB | Value/Character |
| Attribute Characteristics | Data attribute description, such as type, range, missing, etc. from Context-Aware Management FB | Value/String |
| Situational prediction | Predicting the most likely evolution of the current situation from Situational Awareness FB | Value/String |
| Model | It mainly refers to information model and data model | Value/String |
| Policy identify | The unique identification of Policy information | Key/Character |
| Policy type | Policies can represent as goals, recommendations or commands | Value/String |
| Policy Information | Description of Policy | Value/String |

###### 5.2.5.2.3.7 Specific part of the data from Policy Management Functional Block

The policy management function block is responsible for receiving the policies generated by model driven Engineering Functional Block and providing policies to other sub function blocks. The output from the analysis function block to the output function block is a single or integrated policy, including recommendations, commands, objectives, conclusions, etc.

Table 5-14: Specific part of the data from Policy Management FB

| Data Field | Brief Description | Field Type |
|---|---|---|
| Policy identify | The unique identification of Policy information | Key/Character |
| Policy type | Policies can represent as goals, recommendations, or commands | Value/String |
| Policy Information | Description of Policy | Value/String |

NOTE 1: Internal and external data Formats need transformation if not stated as the same format.

NOTE 2: Policy type enumeration is a recommendation.

###### 5.2.5.2.3.8 Specific part of the data from output Functional Block

The output from ENI system to the external system is the policy, which is processed by the output function block. The data model, which is output from the output function block, should be understood by external systems.

This clause describes the data model which is from ENI system towards the external system. The data model consists of key and value pairs, which is descripted in clause 5.2.3.2.1.

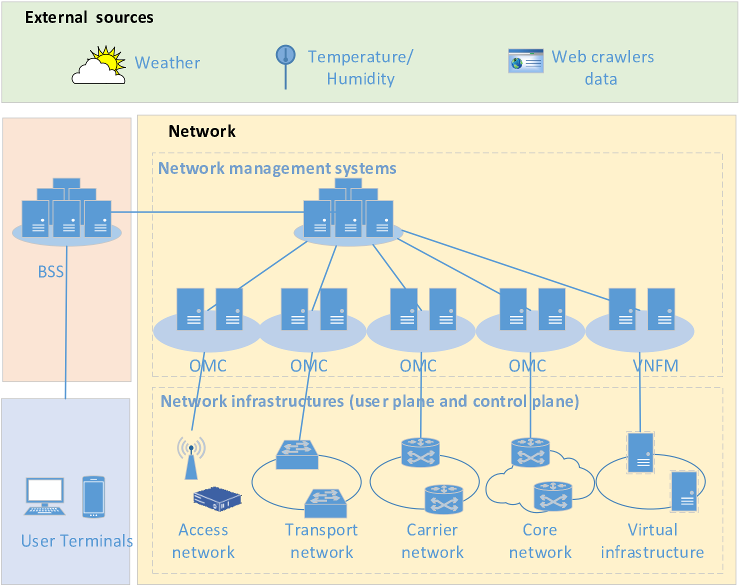
**Table 5-15: Specific part of the data from output Functional Block**

| Data Field | Brief Description | Field Type |
|---|---|---|
| Policy identify | The unique identification of Policy information | Key/Character |
| Policy type | Policies can represent as goals, recommendations, or commands | Value/String |
| Policy Information | Description of Policy | Value/String |

## 5.3 Data Source

### 5.3.1 Introduction

This clause describes the Data Source components, e.g. network management system, network elements, servers, terminals, external environment data, etc., see figure 5-2.



NOTE 1: Figure 5-2 should also contain an application domain that was not represented for the sake of clarity. That application-aware data includes user's experience data, e.g. initial buffing delay, freezing when a user is watching video.

NOTE 2: User devices will be described in a later release.

**Figure 5-2: Data Sources Categories, see notes 1 and 2**

The Data Source components have been categorized as follows:

**Network-related data:** includes data from user plane network infrastructure elements (e.g. base stations, routers, switches and virtual infrastructure) placed in the different segments of the network (e.g. access, transport carrier, core and cloud), control plane network elements (e.g. Software-Defined Networking (SDN) controllers)), as well as all levels of network management systems (e.g. OMC). The way of collecting data from network elements and network management systems will be described in the next clause.

**Business support data:** includes user management data, e.g. user name, user level (e.g. VIP user), register time, subscription service information, accounting data.

**External data:** auxiliary data that is generated outside the network or is unrelated to network behaviour but still relevant for understanding network operation state, including physical sensors that provide environment information (e.g. weather, temperature, humidity), external web/app-based information (e.g. web crawler that provides online news) and data derived from external software tools.

# 5.4 Data Collection

## 5.4.1 Introduction

Data Collection includes gathering and measuring data. The data gathered includes traffic by mirror, log, etc., whereas data measuring uses specific protocols which are examples of Telemetry, such as OWAMP [i.6], TWAMP [i.7], Traceroute [i.26] or IOAM [i.8].

## 5.4.2 Data Acquisition Modes

Data Collection is described in three modes, as follows:

- **Pull** is a mode where data is requested by a consumer and responded to by a producer.
- **Push** is a mode where data is sent by a producer to a customer.
- **Publish-Subscribe or Pub-Sub** is a messaging pattern where publishers of data are decoupled from subscribers of data. In other words, publishers do not send messages to specific entities, but rather, send messages to a pre-defined set of categories. Similarly, subscribers express interest in a set of categories, and have no knowledge of which publishers has delivered the data.

## 5.4.3 Data Collection Techniques

### 5.4.3.1 Introduction

Different types of network-related data could be collected during the telemetry service/application process execution. Those different types of data could be carried out by different protocols associated with different types of connection, e.g. the different functional planes are usually used to describe and deploy different services/applications by existing systems. In the following as an example, only the forwarding/user plane, the control plane, and the management plane are used to describe the data that could be collected in protocols running on them. The next clause deals with data extracted and carried out in those protocols in the above functional planes. The clause after that deals with specific data used to perform the telemetry service/application.

NOTE: The term user plane is applicable only to mobile networks. Forwarding plane (also known as "data plane") that could be used in other networks.

### 5.4.3.2 Data carried out in functional planes protocols

#### 5.4.3.2.1 Data carried out in the Forwarding/User Plane

On the forwarding /user plane, the main function of devices is traffic processing and forwarding. Various data objects (e.g. packet loss, packet received timestamp, queue status) could be collected from various network elements (e.g. routers, switches) as a result of the forwarding process. Various telemetry data could be exported from forwarding chips or line cards through making use of specific tools, such as the IPFIX protocol [i.4].

Examples of monitoring and collecting data from these objects include, for example, packet-level monitoring that could provide precise information for calculating statistics such as the instantaneous bitrate, packet loss, or round-trip latency experienced by individual flows.

According to the categorization specified in IETF RFC 7799 [i.11], techniques of forwarding plane telemetry could be divided into active, passive and hybrid methods. Usually, the IPFIX protocol (IETF RFC 7011 [i.4]) and traffic mirror are considered as passive methods, which are based on observations of an unmodified packet stream. On the other hand, the active methods include, One-way Active Measurement Protocol (OWAMP) IETF RFC 4656 [i.6] and Two-Way Active Measurement Protocol (TWAMP) IETF RFC 5357 [i.7], which generates packet streams as the basis of measurement. The hybrid methods include in-situ Operation, Administration and Maintenance (OAM) (I-D.ietf-ippm-ioam-data [i.8]), IP Flow Performance Measurement (IPFPM, IETF RFC 8321 [i.9]) and Multipoint Alternate Marking (I-D.ietf-ippm-multipoint-alt-mark [i.10]) which augments or modifies the stream of interest to collect metrics.

#### 5.4.3.2.2 Data carried out in the Control Plane

The main purpose of data acquisition in the control plane is to monitor the health of different network protocols. It is beneficial for detecting, localizing and predicting network issues/events through keeping track of the running status of protocols. Traditionally, approaches for control plane Key Performance Indicator (KPI) measurement include protocols such as PING for testing the reachability of a host in an IP network, Traceroute for displaying the path and measuring transit delays, and Recommendation ITU-T Y.1731 [i.12] for measurement of Ethernet frame delay, frame delay variation, frame loss, and frame throughput specified by the ITU Telecommunication standardization sector (ITU-T), which only measure the KPIs but do not reflect the actual running status of network protocols.

The control plane telemetry objects include control protocol or signalling objects, which could be exported from, for example, the main control Central Processing Unit (CPU). For example, the Border Gateway Protocol (BGP) [i.14] monitoring protocol (BMP [i.15]) could be used for monitoring the BGP routes to enable security analysis, Autonomous System (AS) analysis, PING, Traceroute [i.26] that could be used to determine the round-trip delay in communicating with the host and packet loss, etc. Ping uses a series of Internet Control Message Protocol (ICMP) Echo message to determine whether a remote host is active or inactive, whereas Traceroute shows an actual path.

#### 5.4.3.2.3 Data carried out in the Management Plane

The main functions associated to the Management plane are monitoring, configuration, and maintenance of devices. Information such as performance data, network logging data, network warning data, and network state data is collected from the management plane, which interacts with the Network Management System (NMS). Some legacy protocols (e.g. Simple Network Management Protocol (SNMP) and Syslog), are widely used in the management plane telemetry, where configuration and operation state could be exported from main control CPU. In addition, some network management protocols (e.g. gRPC [i.16] Network Management Interface (gNMI) [i.22], Network Configuration Protocol (NETCONF) [i.13], YANG Push [i.5]) have the ability to request telemetry information.

### 5.4.3.3 Specific data used to deploy telemetry

#### 5.4.3.3.1 Network Telemetry

An ENI System uses Big data analytics and machine learning technologies to analyse and produce actionable decisions from network telemetry data for improved network operator experience. A single-sourced and static data acquisition mechanism could not meet the volume, velocity, variety, and other requirements of these technologies. It is desirable to have a framework that integrates multiple telemetry and data collection approaches. This allows flexible combinations for different telemetry data acquisition from different applications.

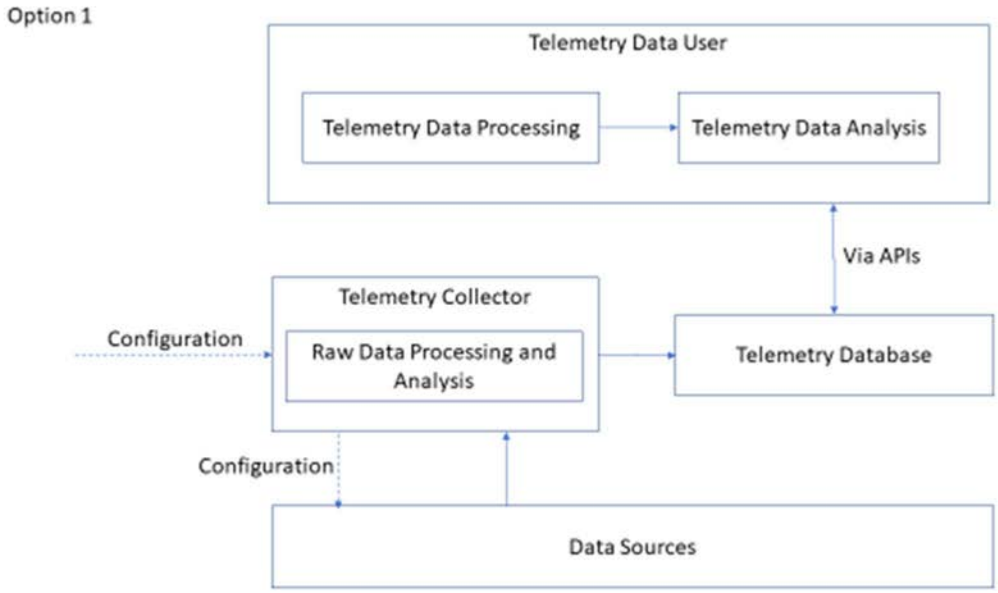


**Figure 5-3a: Components of Network Telemetry framework, API option**

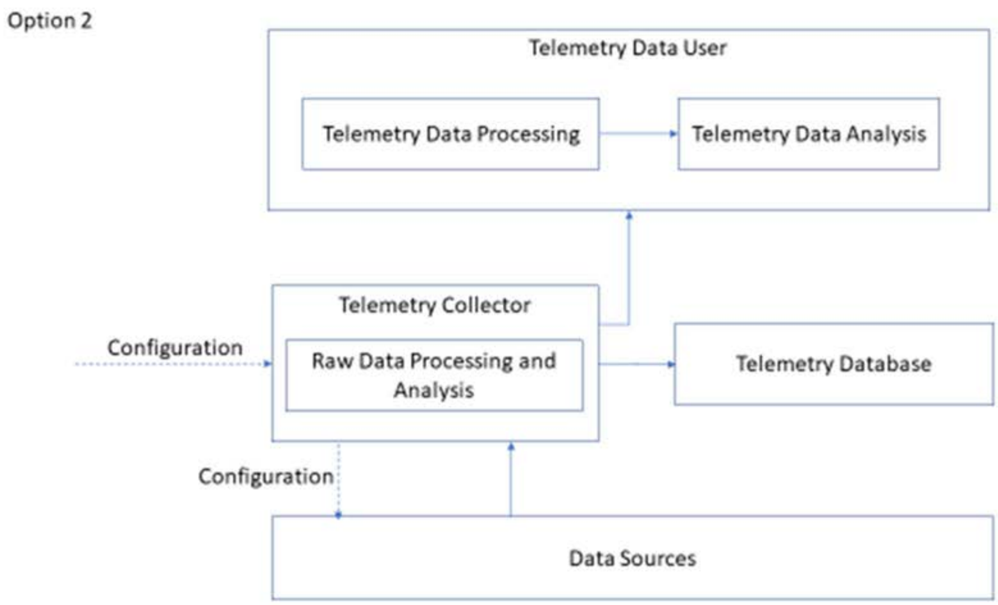


**Figure 5-3b: Components of Network Telemetry framework, Push Option**

The components of a network telemetry framework shown in figures 5-3a and 5-3b. There are two options for Telemetry Data User to get data:

1) deliver the data from Telemetry Collector directly; or

2) get data from Telemetry Database via APIs.

The components are defined as follows:

- **Telemetry Data User:** subscribe for the telemetry data, determine the telemetry data source and determine what types of telemetry the system needs. The Telemetry Data Processing and Analysis is responsible for analysing the feedback data from network devices. Based on data analysis results, further data requirements could be issued. Telemetry Data User could be part of network Operator Support System.

- **Telemetry Collector:** collect raw telemetry data, measurement metrics, such as delay, loss rate and jitter, could be generated and processed through In-situ OAM (IOAM) [i.8], Enhanced Alternate Marking (EAM) [i.17], Postcard- Based Telemetry (PBT) [i.18], IP Flow Performance Measurement IPFPM [i.9], etc. The Raw Data Processing and Analysis performs any filtering, aggregation, anonymization, and other functions. The Raw Data Processing and Analysis encompasses two functions processing on the raw data and analysis on formatted data. The Raw Data Processing could include filtering, aggregation and anonymization.

- **Telemetry database:** repository used to store the data collected from data sources.

- **Data Source:** provides the requested data to be captured, processed, and formatted in the network devices. For example, for the forwarding plane telemetry, the objects could include flow, packet and path and as an implementation option data encoding.

- **Configuration:** is expected to come from an External Source for example from other ENI or OSS functions. The OSS could include the Telemetry Data User hence there are two options for the input of the Configuration.

### 5.4.3.3.2 Resource Telemetry

This clause describes telemetry techniques for data coming from across various available resources.

In a lifecycle managed infrastructure, resources generate multiple data sets for machine learning and Big Data analytics-based systems. For data collection, extraction and subscription mechanisms to effectively retrieve resource telemetry, it is crucial to extract data from various sources of resource telemetry available across the infrastructure. Communication across the infrastructure resources is crucial for efficient operations, where correlation between the collected data across these resources is needed.

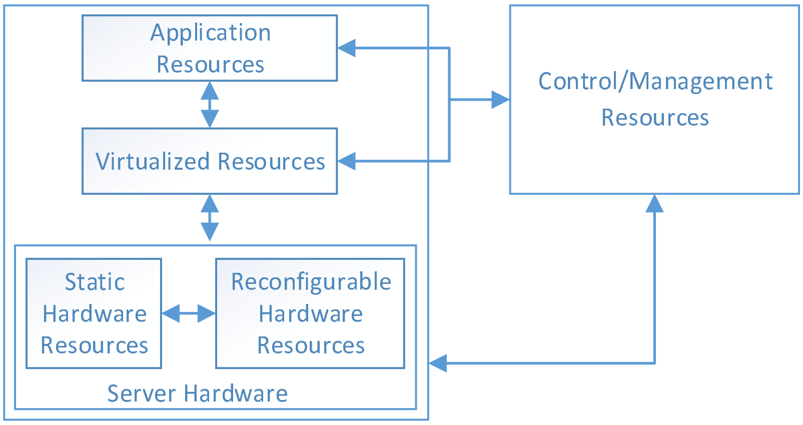

Application Resources
Virtualized Resources
Static Hardware Resources
Reconfigurable Hardware Resources
Server Hardware
Control/Management Resources

**Figure 5-4: Various infrastructure resources that generate telemetry data**

Below are the four resource descriptions for the forwarding (user) plane within the network device:

- **Data Collection from non-reconfigurable Hardware Resources:** telemetry agents that extract data from the hardware resources of a network device need to be efficient, performant and sensitive to latency in order to obtain the desired set of hardware metrics. To obtain actionable insights from static hardware resources requires the collector and analytics agent to be sensitive to performance. Use cases like power savings, last level cache management, etc., that impact network packet latency heavily rely on key hardware resource telemetry to be consumed and analyzed for actionable insights by analytics systems trough the use of machine learning algorithms which digest the telemetry. Event monitoring tools enable action to be taken on one or more of a series of points. These types of tools typically provide streaming data that could optionally be pre-processed (e.g. aggregated, correlated, or have calculations done on them). Another example would be a system that provides reports on memory and network usage, disk usage, and other counter-orientated statistics by polling at a specified time interval. In addition, event traces could also be provided.

NOTE: Agentless Telemetry is for study in a later release.

- **Data Collection from Reconfigurable Hardware Resources:** hardware accelerators, software defined hardware and real-time reconfigurable hardware are increasingly common across software defined infrastructure. Data collection from these resources requires the collectors to be sensitive to changes in hardware configuration, which could be conveyed in metadata, and export out the data in real time in order to facilitate actionable insights. Specifications for hardware such as the Intelligent Platform Management Interface could provide always available and real-time monitoring capabilities.

- **Data Collection from Virtualized Resources:** data extracted from network devices like virtual switches, virtual routers, virtual machines, etc., could provide key telemetry to feed into machine learning algorithms and Big Data analytics engines.

- **Data Collection from Application Resource:** applications that utilize hardware and virtual resources in the network are necessary to expose relevant telemetry for exporting to data storage mechanisms. Application monitoring tools are crucial to gather relevant telemetry. Protocols such as VNF Event Stream (VES) [i.21] help to export the metrics in standardized formats.

- **Data Collection from Control and Management Resources:** control and management also interact with network devices in order to accomplish the functionalities that are associated to them. However, in this case, these functionalities are not relevant per se, rather the data that could be collected from the protocols that carry it. Regarding those protocols as data resources it has to be considered that latency sensitive environments heavily rely on decision time spent by analytics algorithms and enforcement time. Moreover, relevant interfaces into control and management layers help to extract important data about state and performance of these layers. Most environments have control and management resources allocated separately.

Analytics components and artificial intelligence functions need to be aware of the interaction between various resources and resource telemetry involved. The nature of interactions between various types of resources are described below:

- **Application and Virtualized Resources Interaction:** network applications, deployed either in cloud native model or in virtual appliance model, need tight interaction and integration with virtual instances such as virtual switch, virtual router, virtual firewall, virtual load balancer, etc. Virtual resources heavily impact performance of software applications as software resources rely on virtualization layer to access various hardware resources and enforcement of control and management decisions.

- **Virtualized and Hardware Resources Interaction:** hypervisor plays an important role to provide tight integration between virtualized resources and various types of hardware resources. Leveraging telemetry from interactions across these two layers helps to provide secure and scalable solutions across various infrastructure management use cases. These interactions are often exposed via kernel calls, hypervisor metrics, kernel networking subsystem, etc.

- **Static Hardware and Reconfigurable Hardware Resources Interaction:** offloading the application functions at run time to reconfigurable hardware resources is often done via applications utilizing static hardware. Density and nature of interactions between these two types of hardware resources could help to uncover bottlenecks across infrastructure performance and application deployment, which could be often captured via monitoring traffic and memory sub-system accesses among these two components.

- **Software and Control/Management Resource Interaction:** life cycle management of applications utilize telemetry from software applications for control and management layers to infer and enforce appropriate decisions. It is imperative that software resources produce appropriate telemetry in a format that is best consumable by big data type agents in the control and management components.

- **Virtualized and Control/Management Resource Interaction:** telemetry from virtual switches and routers, virtual machines, etc., help control and management layers to obtain a snapshot of scale, performance and security of the infrastructure. Decisions to leverage Single Root I/O Virtualization (SR-IOV) or leverage accelerators for latency sensitive applications, are possible while consuming and assimilating telemetry from across virtualized layers.

- **Hardware and Control/Management Resource Interaction:** with advent of software defined hardware infrastructure, interaction between these two layers is crucial to help latency sensitive applications appropriate hardware configurations. Decisions to reconfigure hardware settings (such as for Field Programmable Gate Arrays (FPGAs), run time adjustment of hardware resources allocated to virtualized and software layers, hardware bandwidth control, etc. in the control layer requiring the management layer to consume the appropriate hardware telemetry.

### 5.4.3.3.3 Fault Telemetry

In a lifecycle managed infrastructure, data extracted from fault telemetry plays a crucial role in the detection and isolation of various faults across the lifecycle process stack. The life cycle of faults and fault processing in general could be broadly represented as indicated in figure 5-5.

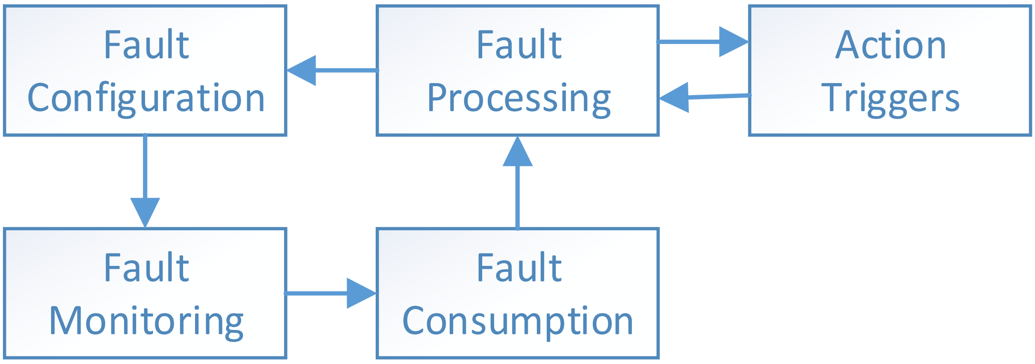


**Figure 5-5: Various stages of a Fault Processing lifecycle**

The five components of the fault telemetry lifecycle are defined as follows:

- **Fault configuration:** the nature of the infrastructure and applications deployed dictate what types of fault monitoring methods should be configured. This includes defining where a fault could be detected, how it is detected, and the permissions assigned to view faults.
- **Fault Monitoring:** faults could be broadly classified, based on the entity generating the faults, as:
    - Hardware faults, e.g. switch, and router faults, including routers programmed as firewalls.
    - Application faults.
    - Virtualization faults (incudes hypervisor faults).
    - Service faults.
    - Communication faults.

  These faults could be generated across various data sources as described in figure 5-5 and are consumed by appropriate agents.

NOTE: Agentless approaches will be considered in a future release.

- **Fault Consumption:** based on the nature of the fault, a software entity is required to consume the fault. Software entities could be generic or specific in nature (e.g. a tool that responds to SNMP [i.22] or YANG [i.5] data versus a tool that is written specifically for a particular type of fault monitoring application). General purpose monitoring software such as Prometheus [i.27] uses a time series database to capture and produce various alarms, notifications, etc.
- **Fault Processing:** data typically needs to be normalized before it is analysed. The normalization is similar in nature to that described in ETSI GS ENI 005 [i.3]. A variety of mechanisms could be employed to fine tuning fault detection and classification by using Machine Learning and other AI algorithms.
- **Action Triggers:** this is a consequential action taken or to be taken by external entities (i.e. NMSs or EMSs governance) in order to address the fault and mark the fault as resolved.

Fault information and fault telemetry results could be exposed via well know outputs, where the most commonly used are alarms, events, notifications, log messages and fault or failure reports.

#### 5.4.3.3.4 Streaming Telemetry

Modern cloud native deployments using microservices based platforms require telemetry and observation to be adapted to scale to the needs of microservices. Traditional telemetry mechanisms that generate data every 5 to 10 minutes are no longer applicable or scalable. Streaming telemetry is a modern method deployed in the cloud native context where microservices use push based mechanisms that continuously stream data. Network telemetry could leverage streaming telemetry by using existing data models such as YANG [i.5], OpenMetrics, etc., in order to enable a programmatic way to process and act on data in real time. Streaming telemetry frameworks provide a highly scalable mechanism for generating and consuming telemetry data sets that could be queried for analysis and help take appropriate actions in real-time automation.

## 5.5 Hierarchical data storage

This clause provides recommendations for the transformation process of different types of data, including raw data, feature data, training data and model data.

Different storage methods are used for different applications of data (e.g. data size, data popularity). Figure 5-6 shows these processes that transform data of one type to data of another type within a data storage hierarchy.

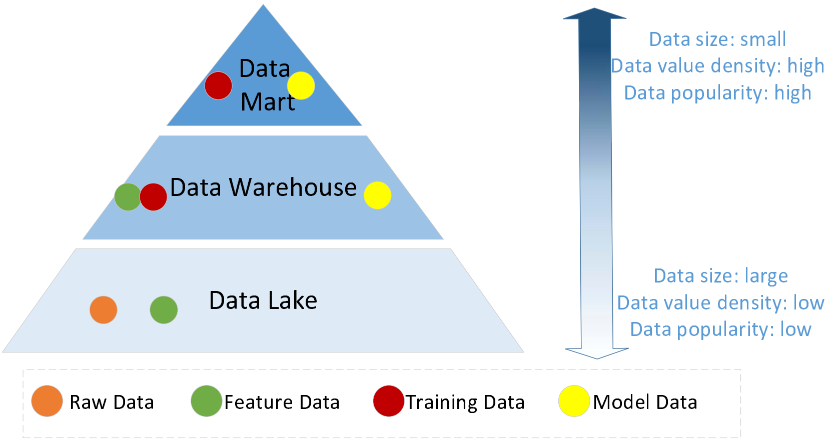


**Figure 5-6: Data Usage Hierarchy**

Figure 5-6 also shows a hierarchy of transformations on data to make it usable for machine learning features as training data sets. Also, transformations of training data into actual model data sets are expected.

**Raw data:** this includes all types of data that is of interest for producing high-quality actionable data. Examples include log files, raw traffic data, and service data. Knowledge and metadata are stored typically by the software entity performing the transformation from raw data into feature data to guide the transformation process. A large amount of raw data could create little value. Therefore, raw data is suggested to be stored in Data Lake.

**Feature data:** features are a component of an observation and are constituted by a set of attributes of data instances (e.g. the column of data in a relational database). Feature selection is the task of selecting which features to include in datasets [i.3]. Features could be extracted from raw data by means of e.g. data cleaning. Feature data is usually of high quality and used for analysis. Therefore, feature data is suggested to be stored in Data Lake or Data Warehouse. There are certain specific processing procedures to clean raw data and form feature data.

**Training dataset:** a training dataset is built by choosing a set of examples that fit the parameters of the model that will be used. Once a fitted model is obtained, a **validation dataset** is created. This enables the fitted model to be evaluated, and provides an opportunity to tune hyperparameters (i.e. the process of training models is the process of tuning hyperparameters). A **test dataset** could also be created; this is a dataset that is used to assess the performance of the model independently from the training dataset. Therefore, training data is suggested to be stored in Data Warehouse or Data Mart.

**Model data:** after model training, a model used by the selected algorithm is ready for use on real-world data in order to classify data and/or to make predictions. Model data is usually rarely written and read frequently. Therefore, model data is suggested to be stored in Data Warehouse or Data Mart.

In addition, deploying data, a data type not mentioned in figure 5-6, is generated when models are deployed and used by selected algorithms. Deploying data includes the configuration data, the interface data, interactive data, etc. It is suggested that this type of data is stored with Meta Database or Config Database.

# 5.6 Data Processing

## 5.6.1 Data Correlation

AI applications use a broad range of data from multiple domains. Analysis of cross-domain data provides better insights and a better and deeper understanding on the usage of how the system is operating. More data means more correlation, which leads to a better training model.

Data Correlation exemplary scenarios are described below:

- **Scenario #1: Correlate data from different geographic locations:** The geographic locations are able to be classified according to different granularity, including different provinces, different cities, and different cells. For example, the need to collect traffic data from different locations and combine it as training data, due to diversified traffic patterns, is important for creating a generalized model.

- **Scenario #2: Correlate data from different network domains:** An end-to-end service typically spans multiple network domains/segments, including access network, transmission network, core network, etc. Important KQIs such as end-to-end service quality and availability need to be trained using data from different domains.

- **Scenario #3: Correlate data from different professional systems:** Correlating data from different data sources that are external to the network, but relevant to goals that the network is trying to achieve, also helps the service provider to achieve a more targeted model. Examples include data from OSS, BSS and CRM systems.

### 5.6.2 Data Cleansing

Data cleansing is the process of detecting and correcting corrupt, inaccurate, incorrect, incomplete, irrelevant, or duplicated data. It produces consistent and well-formatted data for analysis, and endeavours to maximize the accuracy of the data, in Table 5-16 below.

**Table 5-16: Data Cleansing process**

| Purpose | Problems | How to detect | How to process |
|---|---|---|---|
| Data accuracy | Abnormal data | Deviation analysis by using rule libraries | Edit data based on the type of abnormality |
| Data completeness | Incomplete data | Missing or null values (see note) | Add the missing information by inference from other known stored values (see note) |
| Data measurement | Different data measurement methods | (see note) | (see note) |
| Data consistency | Different data formats | (see note) | (see note) |
| Data validity | data format checking | (see note) | (see note) |
| Data relevance | Evaluation of context awareness data | (see note) | (see note) |
| NOTE: This will be completed in a future release. Some evaluation of the pros & cons of showing the information content in a tabular form will be performed. | | | |

## 5.7 Data Sharing

Data sharing is a process where the data provider shares its data with a data consumer.

Based on the content of the shared data different sharing methods are used, for instance there are two types of data sharing that could be used:

a) the data provider shares the data with the data consumer;

b) the data provider shares the AI model or algorithm with the data consumer.

Examples of data formatting mechanisms in protocols are shown as follows, depending on types of data:

a) Subscription procedure data sharing (like subscribe/notify) could be based on formats for example Protobuf or JSON.

b) Periodic transmission data sharing uses File Transport Protocol (FTP) [i.23] to transmit static data to receiver in the format of file or table.

c) Request/response data sharing procedures encompasses transmission by the provider of small volumes of data: where a response follows a single request query. In this case, the format of JSON/XML generally is used.

For AI model or algorithm sharing, the shared model could use a supervised or semi-supervised algorithm. In the unsupervised learning case only the AI algorithm could be shared.

## 5.8 Data Management

### 5.8.1 Overview

Data management consists of metadata management, data security management and data quality management.

### 5.8.2 Metadata Management

Metadata management is the set of processes that ensure proper creation, storage, integration, and control to support the associated usage of metadata. In particular, it consists of rules, guidelines, or best practices in applying metadata to data and behaviour. The process of integration means that data could be provided as a service in an integrated way, while, the process of control means that who is able to decide where and how to use the data.

### 5.8.3 Data Security Management

Data security management includes the planning, development, and execution of security policies and procedures to provide proper authentication, authorization, access, and auditing of data, information assets, encryption, accounting, key management, data erasure and vulnerability assessment. It consists of definition of data security policy, access control of data security and auditing of data security.

NOTE: This topic will be expanded in a future release.

### 5.8.4 Data Quality Management

Data quality management is a set of processes that maximize the customer uniqueness, consistency, and completeness of data (e.g. product and service data). It involves Data Governance, Master Data Management, Digital Asset Management, and other disciplines.

Data quality analysis analyses and evaluates the data quality of a data set to determine if measured data is meeting the SLOs of their SLA.

## 5.9 Data Conversion

### 5.9.1 Introduction

Data transformation is the process of merging, cleaning and integrating data, changing from one form of expression to another form of expression, and achieving the semantic consistency of different source data.

### 5.9.2 Data Conversion between Functional Blocks

The data conversion types between ENI system function blocks include: data type conversion, data semantic conversion, data granularity conversion, table/data splitting, data discretization, refining new fields, attribute structure, etc.

**Data Type Conversion:** The data types of different data sources are uniformly converted into a compatible data type.

**Data semantic conversion:** In traditional data warehouses, there may be dimension tables, fact tables, etc. based on the third normal form. At this time, there will be many fields in the fact table that need to be combined with dimension tables for semantic analysis.

**Data Granularity Conversion:** Business systems generally store detailed data. Some systems even store data based on timestamps. The data in the data warehouse is used for analysis and does not require very detailed data. Under normal circumstances, business system data is Different granular requirements in the data warehouse are aggregated.

**Table/Data Splitting:** Some fields in the data table may store a variety of data information, for example, the time stamp contains information such as year, month, day, hour, minute, and second. In some rules, some or all of the time attributes need to be split to meet the requirements of data aggregation under multiple granularities. Similarly, multiple fields in a table may also have table field splits.

**Data Discretization:** Discretize continuously-valued attributes into several intervals to help reduce the number of values for a continuous attribute.

**Refining New Fields:** In many cases, new fields need to be extracted based on business rules. These fields are also called compound fields. These fields are usually generated based on a single field, but complex calculations and even complex algorithm models are required to obtain new indicators.

**Attribute Structure:** In the process of AI modeling, new attributes need to be constructed based on the existing attribute set.

For large-scale data, data compression is also required before data conversion to reduce the data size. Data compression is also conducive to reducing storage space, improving its transmission, storage and processing efficiency, and reducing data redundancy and storage space. Data compression can be achieved by data aggregation, dimension reduction, and data block reduction.

**Data Aggregation:** Combine data from different sources before use.

**Dimension Reduction:** Use relevant analysis methods to manually eliminate redundant data attributes and reduce the dimensions/fields involved in data analysis. In addition, Principal Component Analysis(PCA), factor analysis, etc. can also be used for dimensional aggregation.

NOTE: The main idea of PCA is to map the n-dimension feature to the k-dimension, which is a new orthogonal feature, also known as the main component.

**Data Block Reduction:** Using clustering or parametric models to replace the original data is common in multiple models for comprehensive machine learning and data mining.

## 5.9.3 Data Conversion between ENI AI Data Model and External System

### 5.9.3.1 Introduction to Data Conversion between ENI AI Data Model and External System

The data in the External Systems may be various. As mentioned in clause 5.2.3, the data may be structured, semi-structured, and unstructured data. The AI Data Model is essentially a mathematical function whose most direct input is a mathematical vector composed of numbers. Therefore, whatever the data format is, data is ultimately converted to a vector as input to the function, that is AI Data Model. Converting to the vector may be performed in the Data Ingestion and the Normalization Functional Blocks. Some data processing operations mentioned in clause 5.6, or other data converting operations mentioned in clause 5.9.2, may be performed before or after the converting to a vector to meet the input data quality requirements of the AI Data Model.

NOTE: The AI data model in this clause is not the Protocol Data Model.

AI Model files can be obtained by running machine learning algorithm in the training framework (for example, TensorFlow, Caffe), which constitute the whole model description according to certain rules, including directed graph information, weight information, connection relations and so on. The training framework usually has its own model file. Therefore, the AI model files are related to training frameworks.

### 5.9.3.2 Introduction to AI Data models

The data provided to the External System by ENI mainly consists of two forms, a trained AI Data model, or the output of the AI Data model. As stated in clause 5.9.3 on the AI Model Files the External Systems need to deploy the corresponding framework in order to perform inference with the model files. In some case that the External Systems do not have the corresponding framework, the model file needs to be converted into an appropriate model that can be used. The External Systems can also use the model file in the way of Docker or Web API.

### 5.9.3.3 AI Data model types

AI model is a mathematical function which learns a solution to a problem from sample data. According to the use of labels during training, AI models are divided into the following four categories:

**Supervised model:** A supervised model is a function that maps an input to an output using labelled inputs and outputs as examples. Each input consists of a tuple that includes an input object and a desired output value. The learning function examines the training data and generates a function that can determine new data's labels.

**Semi-supervised model:** Semi-supervised model is a combination of supervised and unsupervised learning, where the training data consists of both labelled and unlabelled data.

**Unsupervised model:** Unsupervised model defines a function that maps an input to an output without the benefit of the data being classified or labelled. Probability densities can be used to model the input data.

**Reinforcement learning model:** Reinforcement learning model uses software agents to take actions in an environment in order to maximize a cumulative reward. In this approach, the learning agent is not told which actions to take, but instead is responsible for determining which activities offer the biggest reward.

**Table 5-17: Comparison of the AI Data model**

| Model Type | Introduction | Classic Model | Usage Scenario |
|---|---|---|---|
| Supervised model | Supervised learning is a learning method that infers a function from input to output from a set of manually labelled data. | 1) k-Nearest Neighbours<br>2) Linear Regression<br>3) Logistic Regression<br>4) Support Vector Machines<br>5) Decision Trees and Random Forests<br>6) Neural networks | 1) Classification problems<br>2) Regression problems |
| Semi-supervised model | Semi-supervised learning is the combination of a large amount of unlabelled data and a small amount of labelled data. Compared with supervised learning, semi-supervised learning has higher accuracy and lower training cost. | 1) Graph Inference<br>2) Laplacian SVM | Prediction scenarios where sample labels are difficult to obtain or cost is high |
| Unsupervised model | Unsupervised learning is mainly used for mining implicit relationships between unlabelled datasets. | 1) k-Means<br>2) Hierarchical Cluster Analysis<br>3) Expectation Maximization<br>4) PCA<br>5) t-SNE | 1) Clustering<br>2) Abnormal detection |
| Reinforcement learning model | Reinforcement learning is the learning of the intelligent system from the environment to the behaviour mapping, so as to maximize the value of the reward signal function. Since there is little information given by the outside, the reinforcement learning system relies on its own experience to learn by itself. | 1) Q-Learning<br>2) Temporal Difference Learning | 1) Dynamic system processing<br>2) Robot control |

NOTE: Some description of how the model types relate to each other (e.g. a table showing Dynamic and Static compared to the later four models).

### 5.9.3.4 The corresponding framework of AI Data model

As stated in clause 5.9.3 on the AI Model Files the External Systems need to deploy the corresponding framework in order to perform inference with the model files. The mainstream frameworks corresponding to AI models include the following seven types:

- Theano.
- Tensorflow.

- Keras.
- Caffe.
- MXNet.
- Pytorch.

**Table 5-18: Comparison of the corresponding framework of AI Data model**

| Framework | Introduction | Advantage | Disadvantage |
|---|---|---|---|
| **Theano** | As the first Python deep learning framework, Theano laid the basic design direction for the subsequent development of deep learning frameworks. | 1) Combinate the Python and NumPy<br>2) The use of computational graphs<br>3) Low learning threshold | 1) It is more bloated than Torch<br>2) Distributed is not supported<br>3) Long compilation time<br>4) Development has been stopped |
| **TensorFlow** | It can be seen as the successor of Theano. TensorFlow, the most popular deep learning framework today, has a comprehensive and flexible ecosystem of tools, libraries, and community resources. | 1) Better visualization of computational graphs<br>2) Strong scalability<br>3) Excellent community support<br>4) Excellent performance | 1) The system design is complex and maintenance is difficult<br>2) The interface design is complex and changes frequently<br>3) The graph construction is static |
| **Keras** | It is a high-level neural network API written in Python that can run with TensorFlow or Theano as the backend. Keras development is focused on enabling rapid experimentation. | 1) Simple and easy to use<br>2) To provide standard-rich documents<br>3) Easy to debug and expand | 1) Lack of flexibility<br>2) Slow operation |
| **Caffe** | The core language is C++, which supports command line, python and MATLAB interfaces. It is an open source deep learning framework with expressiveness, speed and modularity. | 1) Lightweight, good scalability<br>2) Excellent performance<br>3) Almost all platforms are supported | 1) The official did not provide complete documentation<br>2) The installation is complicated |
| **MXNet** | MXNet is a deep learning library that supports a variety of common languages, borrowing ideas from Caffe, but with a cleaner implementation. | 1) Supports flexible dynamic graphs and efficient static graphs<br>2) Good scalability, powerful distributed performance, and strong portability<br>3) Supports multiple languages and platforms | 1) The entry threshold is high<br>2) The documentation is incomplete and the update is slow |
| **PyTorch** | PyTorch is essentially a GPU-enabled NumPy replacement, equipped with more advanced features that can be used to build and train deep neural networks. | 1) Concise, easy to understand code<br>2) Faster than TensorFlow and Keras<br>3) Flexible and easy to use<br>4) Active community and complete documentation | Mobile device deployment is not supported |

Except the External Systems have the corresponding framework, the External Systems can also use the model file in the way of Web API by Docker.

## 5.9.3.5 External system data conversion

### 5.9.3.5.1 Introduction

The conversion of external system data to AI data model can be divided into three categories: converting structured data into vectors, converting semi-structured data into vectors and converting unstructured data into vectors.

#### 5.9.3.5.2 Converting structured data into vectors

For structured data, if the types of data fields are all numeric, the data can be directly regarded as a two-dimensional vector. If the type of the data field is character, the characters can be converted into numbers by encoding and other methods.

EXAMPLE: code "male" as "0" and "female" as "1".

#### 5.9.3.5.3 Converting semi-structured data into vectors

For semi-structured data, it is first converted to structured data and then to a vector. Semi-structured data is information that does not conform to a formal data model, but has some organizational properties that define key data. Depending on the structure of the key data, semi-structured data can be transformed into structured data.

For example, it is first found that the converted field has obvious structure in semi-structured data, such as "character + colon + number" or "character + equal sign + number". Then get all the strings that satisfy the above structure, e.g. the string might be TimeStamp=20150701105747350". Third, grab the converted field and split it with an appropriate delimiter (e.g. colon or equal sign), e.g. grab Strings containing "TimeStamp=" and split them with "=". Finally, the characters before the delimiter are used as data fields in structured data, and the numeric value after the delimiter is used as the value.

#### 5.9.3.5.4 Converting unstructured data into vectors

Unstructured data needs to be processed in order to find information by. Unstructured data is information that does not have a pre-defined data model, and does not contain properties that provide any organization or structure to its elements. Unstructured data needs to be processed, such as extracting feature data according to domain-specific applications.

Different unstructured data are converted into vectors in different ways. For example, when converting text data to vector, tokenization is performed. Tokenization is a way of splitting text into smaller units called tokens. Tokens can be characters, words or terms. Then match the generated tokens with numeric vectors in the way of one-hot encoding or word embedding. When converting images to vectors, the vector elements represent the colour information of each pixel. An RGB image can be converted to a 3 - dimensional vector. The length of the three dimensions represents the height, height (in pixels) and number of channels of the image, respectively and each element is the quantized values of R (Red), G (Green) and B(Blue) channels.

## 5.10 Data Security

### 5.10.1 Introduction

Data security refers to taking necessary measures to ensure that the data in the data mechanism is effectively protected and legally used, and that customer privacy is not disclosed during the entire life cycle of data collection, processing and utilization.

Data security can be guaranteed through artificial intelligence technology, cloud computing, firewall, authentication and authorization, data encryption, data masking, data backup and elasticity technology, data erasure technology and other technical means.

### 5.10.2 Artificial Intelligence Technology

Federated machine learning/Federated learning in AI technology is usually used in data security.

Federated learning is a machine learning framework that can effectively help multiple institutions to model data use and machine learning while meeting the requirements of user privacy protection, data security and government regulations.

As a distributed machine learning paradigm, federated learning can effectively solve the problem of data islands, enable participants to jointly model on the basis of not sharing data, break data islands technically, and achieve AI cooperation.

### 5.10.3 Cloud Computing Technology

Cloud computing technology protects data privacy through key technology, new algorithms, encryption algorithms and other authentication methods, while enhancing the protection of data itself. Cloud computing technology can be used to encrypt data at all stages of data transmission, storage and processing to achieve information hiding and protect user data security.

### 5.10.4 Firewall Technology

Firewall is the initial security layer in the system. It is designed to prevent unauthorized sources from accessing enterprise data. Firewalls act as intermediaries between personal or corporate networks and the public Internet. The firewall uses pre-configured rules to check all data packets entering and leaving the network, thus helping to prevent malware and other unauthorized traffic from connecting to devices on the network.

Different types of firewalls include: basic packet filtering firewall, line level gateway, application level gateway, status check firewall, and next-generation firewall.

### 5.10.5 Authentication and Authorization

Authentication and authorization technology can be used to ensure that only authorized users can access data and prevent data leakage.

### 5.10.6 Data Encryption

Data encryption converts data into encoded ciphertext to ensure its security in static state and transmission between approvers. Encrypting data ensures that only those who have the correct decryption key can view the data in the original clear text.

### 5.10.7 Data Masking

Data masking is also known as data obfuscation, data anonymization, or pseudonymization. It is the process of replacing confidential data by using functionally fictitious data, such as characters or other data. The main purpose of data shielding is to protect sensitive private information when an enterprise shares data with a third party.

### 5.10.8 Data Backup and Elasticity technology

Data backup technology means that enterprises should save multiple copies of data, especially when they want to fully recover after data leakage or other disasters. With data backup, enterprises can restore normal business functions faster with fewer failures. To ensure data elasticity, organizations need to take appropriate protection measures to ensure the security and availability of backup data.

### 5.10.9 Data Erasure Technology

Data erasure technology has also become a data destruction technology, which is mainly used to ensure that the deleted data cannot be recovered after the enterprise has correctly deleted the data.

# 6 Example Scenarios to Illustrate Data Mechanisms

## 6.1 AI-enabled Traffic Classification Use Case

### 6.1.1 Introduction

Network traffic classification plays an important role in network operation and management, which supports numerous network closed-loop control activities in terms of network security, traffic engineering and Quality of Service (QoS). Traditional methods, such as port-based technique and payload-based technique, are inefficient and even fail to classify some types of network traffic, due to the increasing proportion of encapsulated traffic and enterprise private methods. Therefore, various techniques based on AI algorithms (e.g. pure machine learning) are used when training the algorithm to classify some types of network traffic. The training data set is composed of data stream features or extracted packet features with specific classification labels.

### 6.1.2 Data Acquisition

For the purpose of network traffic classification, the ENI System could collect the information as listed in table 6-1.

**Table 6-1: Data collected by ENI for network fault root-cause analysis and intelligent recovery**

| Information | Source | Characteristics | Data Format |
|---|---|---|---|
| Network Traffic data | Forwarding plane -->switches/routers | IP, port, packet length, inter arrival packet time, etc. extracted from TCP/UDP flows<br>See note 1 | See note 2 |
| NOTE 1: The indicated protocols are meant to be an example only within the use case.<br>NOTE 2: The contents of this column will be addressed in a future release. | | | |

### 6.1.3 Data Processing for traffic classification

This clause describes data processing procedures are performed in a sequential process within the ENI System as explained in the following steps, related to cleaning and labelling respectively:

1) Initially the data needs to be cleaned before it is available for processing traffic classification One or more packets could be lost during transmission. So some TCP [i.19]/UDP [i.20] flows could be incomplete. These flows that lack many packets or lack key packets (e.g. initial TCP 3-way handshake packets) are not recommended for model training or inference, and removed from the data set when they are proved to contain erroneous data. In addition, TCP flows could contain retransmission packets. Retransmission packets are recommended to be retained and some retransmission related information (e.g. retransmission packet numbers) should also be considered.

2) Secondly the cleaned data needs to be labelled for processing within traffic classification If supervised learning algorithms (e.g. Random Forests, Convolutional Neural Network) are applied for traffic classifier training, traffic data should be labelled. One way is to label the traffic types by analysing features (e.g. IP, special signatures in the packet payload). Another way is to assign labels to designated network traffic that is captured from network interfaces according to predefined rules, see note.

There are differences between classifying traffic using packet payloads, host behaviour, and flow features. This use case uses traffic features to classify the type of traffic with a pre-defined recipe.

NOTE: These topics will be expanded in a future release, including labelling of traffic data for supervised learning.

## 6.2 Network Fault Root-Cause Analysis and Intelligent Recovery Use Case

### 6.2.1 Introduction

Network fault root-cause analysis and intelligent recovery is an important use case for the ENI System.

Traditional network fault location and repair need manual processing, which typically has high cost, low efficiency and long implementation timer. Applying a machine learning algorithm to network fault root-cause analysis and intelligent recovery could become a more effective solution, which shortens the time for fault recovery and improves the efficiency of network maintenance. When faults occur, AI algorithms (e.g. Knowledge Graph, Reinforcement Learning) are used to calculate the fault self-recovery policy with alarms data, network topology data, network service data collected from the Monitoring System (MS). The fault self-recovery operation is then delivered to network through a multi-vendor command platform. Self-recoverable faults could be quickly recovered and do not affect the user experience. If a fault could not be rectified, accurate diagnosis is able to be performed to locate the root cause (e.g. by using Big Data Mining Algorithms, Deep Learning Algorithms) thus helping engineers to quickly rectify the fault.

### 6.2.2 Data Acquisition

For the purpose of this Use case, the ENI System could collect the information as listed in table 6-2.

**Table 6-2: Data collected by ENI for network fault root-cause analysis and intelligent recovery**

| Information | Source | Characteristics | File Format |
|---|---|---|---|
| Network Element data (NE data) | IMS, see note | NE name, NE type, NE IP address, physical location, running status, subnet, LSR ID | See note |
| Link data | IMS | Alarm level, link name, link type, source network element, source network element Internet Protocol (IP), source port, source port IP, destination network element, destination network element IP, destination port, destination port IP, link rate, creation time | See note |
| Tunnel data | IMS | Tunnel name, operation status, alarm status, enabling status, source node, destination node, creation time, tunnel ID, tunnel type | See note |
| Alarm data | IMS | Log serial number, Network Element Name, object ID, NE type, NE sub equipment, NE name, equipment alarm serial number, alarm module, alarm type, alarm level, alarm status, occurrence time, confirmation time, clearing time, positioning information | See note |
| NOTE: The contents of this column will be addressed in a future release. | | | |

NOTE: IMS refers to the Information Monitoring System that monitors the real-time network alarms and dispatches the alarms data and network infrastructure data (e.g. network element data, network topology data, network service data) to the ENI System.

### 6.2.3 Data Processing

Like for other Use Cases this clause also describes data processing procedures supporting analysis in the ENI System, e.g. data storage, data filtering, data cleansing, data sharing, etc.:

1) Users need to save data that belong to the same data source with strong data correlation to that data source.

2) Alarm data cleaning and reordering. The original alarm log data contains a large number of alarm logs with the same name at the same time of the same node. For example, if data mining is used, this part of duplicated data has a great impact on the effect of the associated use of rule mining. Independently of the choice of the method, data mining or otherwise, this part of data is cleaned out and only kept in a record with the smallest serial number (the earliest occurrence time) in the equipment log. In addition, the sequence of the gateway log serial number and the device log serial number is inconsistent, which leads to the wrong rules (e.g. if mining is used), while the device log serial number is more accurate. Therefore, the alarm log of the same device is rearranged according to the device log serial number to ensure that the gateway log serial number could accurately reflect the sequence of the alarm log on the same device.

3) Compress the repeated data from frequent alarms, the display of the frequency. The alarms with high frequency need to be taken into account by operation and maintenance. Frequent re-occurrence could give a measure as to how important or severe the alarm is considered.

# 6.3 Intelligent Service Experience Evaluation Use Case

## 6.3.1 Introduction

It is important for operators to evaluate the user's service experience based on the network data. According to the network data, operators could find bad user's service experience and solve problems quickly. However, the traditional service quality evaluation methods based on predefined or empirical functions are sometimes not able to reflect real user's service experience in realtime, essentially for video transmission. Therefore, an AI algorithm is applied to train an evaluation model which represents the non-linear relationship between the transmission parameters and user's service experience. The training set includes network data and user's service experience data, and the latter is used as labels. In table 6-3, the training data acquisition is shown. The service quality data acquisition could be performed by one of two options:

1) though a 3$^{rd}$ party service as defined in the 5G network case; or

2) by using the SDK service of client.

## 6.3.2 Data Acquisition

For the purpose of Intelligent Network Service Experience evaluation, the ENI System could collect the information as listed in table 6-3.

**Table 6-3: Data collected by ENI for service experience evaluation**

| Information | Source | Characteristics | File Format |
|---|---|---|---|
| Network data | Forwarding plane-->switches | TCP RTT, TCP retransmission rate, etc.<br>See note 1 | See note 2 |
| Service quality data | 1) through the third-party service open interface that is defined in the 5G network and oriented to the third-party Application Function (AF)<br>2) through the built-in Software Development Kit (SDK) of the third-party service client | initial buffering latency, video stalling duration, the number of video stalling events, etc. | See note 2 |
| NOTE 1: See next clause 6.3.3.<br>NOTE 2: The contents of this column will be addressed in a future release. | | | |

## 6.3.3 Data Processing

As for the other use cases, this clause describes data processing procedures supporting analysis in the ENI System:

1) A large amount of network data is generated when users access network. The fields of network data include e.g. TCP RTT, and TCP retransmission rate. Regarding the situation where large RTT and retransmission happens the service experience of the user is degraded. In some cases, fields of network data acquired by operators could be missing and that translates in service experience degradation. Also in other cases, fields of network data acquired by operators could be missing, so that this data could not be used directly in AI model training or inference. In order to increase the quality of the overall network data set, data addition and data cleaning techniques are applied to improve Training and Inference. In order to solve this, the deletion of network data entries that have too many fields could take place. Where entries are lost the missing data is filled in, if possible.

2) In order to create a training data set, user's service experience data should be associated with network data belonging to the same service session. User's service experience data includes initial buffing delay and video stalling. It equals to labels and reflects whether the quality is good or bad.

# 7 Example requirements

## 7.1 Data format and interface

In order to maintain the consistency of data format and interface, it is recommended that when using ENI architecture, the data format should conform to the recommendations in clause 5.2, and the interface should meet the following example requirements:

[DFI.1] The log interface transmission can support Kafka, Syslog and other methods, but not limited to the above methods.

[DFI.2] The application layer protocol can support http, https, MQTT and other types, but not limited to the above types.

[DFI.3] Support encryption algorithm call interface, including popular encryption algorithms such as AES, etc.

[DFI.4] It supports authentication when Web Service, SFTP and other applications or protocols are called, and can be accessed after successful authentication. However, it is not limited to the above methods, such as remote access to the security management platform through the software interface of the web service, which can be authenticated by using the token.

[DFI.5] Support different network interfaces for business interface and management interface.

[DFI.6] The application layer protocol can support MODBUS, OPC UA and other types, but not limited to the above types.

## 7.2 Data Security

### 7.2.1 Data Security introduction

The general example requirements of ENI data security include: data collection security, data transmission security, data storage security, data processing security, data exchange security, data destruction security.

### 7.2.2 Data Collection Security example requirements

In the process of collecting internal and external data, the purpose and use of the collected data should be clarified to ensure that the principles of authenticity, validity and minimum sufficiency of data sources are met, and data collection channels, standardized data formats, and related processes and procedures should be specified. In order to ensure the compliance, legitimacy and consistency of data collection. Specific example requirements are as follows:

[DCS.1] Data collection should be based on a unified data collection process to build data collection related tools to ensure the consistency of the organization's data collection process.

[DCS.2] Data collection should adopt technical means to ensure that personal information and important data are not leaked during the data collection process.

[DCS.3] Necessary technical means should be taken to verify the collected data.

[DCS.4] The data collection and acquisition process should be tracked and recorded to support the traceability of the data collection and acquisition operation process.

### 7.2.3 Data Transmission Security example requirements

Appropriate encryption protection measures should be adopted in the process of data transmission to ensure the security of transmission channels, transmission nodes and transmission data, and to prevent data leakage during transmission. Specific example requirements are as follows:

[DTS.1] Technical tools for auditing and monitoring protection measures such as channel security configuration, cryptographic algorithm configuration, and key management should be deployed.

[DTS.2] Nodes on each transmission link should deploy independent key pairs and digital certificates to ensure effective identity authentication of each node.

### 7.2.4 Data Storage Security example requirements

In the process of data storage, differentiated encrypted storage of data with different security levels should be performed, and integrity verification should be performed during use to prevent data from being tampered with and ensure data confidentiality and integrity. Specific example requirements are as follows:

[DSS.1] The performance of data storage devices should be monitored using technical tools, including data storage device usage history, performance indicators, errors or damage, and early warning of data storage devices that exceed safety thresholds.

[DSS.2] Establish a data backup and recovery mechanism, and implement redundant management of stored data to ensure data availability through regular data backup and recovery.

### 7.2.5 Data Processing Security example requirements

#### 7.2.5.1 Data Processing introduction

Specific data processing security example requirements include the following data desensitization example requirements, data analysis security example requirements, data import and export security example requirements.

#### 7.2.5.2 Data Desensitization example requirements

In this case desensitization refers to the process of removing sensitive information data into non-sensitive data:

[DPD.1] Desensitize sensitive data according to business needs to ensure a balance between data availability and security.

[DPD.2] Desensitization data identification and desensitization effect verification service components or technical means should be configured to ensure the effectiveness and compliance of data desensitization.

[DPD.3] Data desensitization components or technical means should be provided to support data desensitization technologies such as generalization, suppression, and pseudonymization.

[DPD.4] Deploy dynamic data masking solutions for specific data usage scenarios and data masking strategies.

#### 7.2.5.3 Data Analysis Security example requirements

[DAS.1] Appropriate security control measures should be taken during the data analysis process to prevent the security risks of valuable information and personal privacy leakage during the data mining and analysis process.

[DAS.2] Combine technical means to reduce security risks in the process of data analysis, such as automatic identification of important data based on machine learning, data security analysis algorithm design, etc.

[DAS.3] Necessary technical means (such as scanning the analysis result data and taking necessary control measures) and management measures should be taken to prevent the output data analysis results from containing recoverable personal information, important data and other data and structural identifiers ((such as user Identification) important signs of information and data structure), to prevent data analysis results from endangering personal privacy, company business value, social public interests and national security.

[DAS.4] A security risk monitoring system for the data analysis process should be established to conduct batch analysis and tracking of security risks that may be involved in data analysis.

[DAS.5] It should have data analysis security capabilities such as automatic identification of sensitive data based on machine learning, and data analysis algorithm security design.

#### 7.2.5.4 Data Import and Export security example requirements

[DIE.1] The security of data should be managed during the data import and export process to prevent possible harm to the availability and integrity of the data itself during the data import and export process, and to reduce the possible risk of data leakage.

[DIE.2] Multi-factor authentication technology should be used to identify the data import and export operators.

[DIE.3] Redundant backup capability should be provided for data import and export channels.

[DIE.4] Traffic overload monitoring should be carried out on the data import and export interfaces.

[DIE.5] A unified data import and export management system should be established to prompt the security risks of data import and export and conduct online audits.

[DIE.6] Standardized data import and export mechanisms or service components should be configured to clarify the minimum security protection example requirements for data import and export.

### 7.2.6 Data Exchange Security example requirements

Data exchange security example requirements include data sharing security Example requirements, data release security Example requirements, and data interface security example requirements.

### 7.2.7 Data Destruction Security example requirements

Data destruction security Example requirements include data destruction processing example requirements and storage device destruction processing example requirements.

# 8 Recommendations

The present document describes some technical methods to support data-driven intelligent network scenarios. The extraction of data and its treatment is a crucial aspect regarding the realization of intelligent networks. It assumes a fundamental role in the performance of every AI-based system, in particular to ensure that the appropriate data is collected and processed accordingly. Ingested data could be used both for network operations as well as model training. More data, and in particular more quality data, means better knowledge to feed the system during the training process, and better chances to improve its performance.

That is why the data mechanisms analysed in the present document, i.e. data acquisition, storage, processing, sharing and management, become so relevant for the creation of proposals for ETSI GS ENI 005 [i.3]. The results of the investigation are applied to the standard specification.

In particular, the present document enriches the Data Ingestion Functional Block and the Normalization Functional Block of the ENI Reference System Architecture ETSI GS ENI 005 [i.3] by describing some technologies inside these two FBs with more detail. Based on the output of this study, some normative work for ETSI GS ENI 005 [i.3] could be found as needed, after accurate evaluation, addressing the following aspects, in a future release of this area of work of the ENI System specified in ETSI GS ENI 005 [i.3]:

- What is the data format when data is used to interact between specific FBs? This could mean that $I_{norm-sem}$ could be structured data because normalized data is usually structured and structured data is more easy to use, $I_{sem-km}$ could be graph data because some knowledge is represented by using graphs.

- How is the data converted in order that it could be understood and used by other FBs or external systems? This could mean that the data from the Knowledge Management FB could be an AI model which is acquired by training based on the data from other FBs, the data from the Policy Management FB could be converted to appropriate commands or instructions that are adapted to external systems.

- How is data acquired and processed in a more efficient way in ENI closed control loops?

In order to enable network data to better support intelligent network data processing, some new technological trends are emerging. It is recommended that some of them are adopted in the future, e.g.:

- **Flexible data acquisition:** Data is able to be collected at a dynamic sampling frequency (e.g. data is collected with a higher or lower frequency) or dynamic granularity (e.g. more or less fine-grained) than the last collection. These are two sampling dimensions. The sampling is handled by the control loops, that decide the granularity and frequency of data collection. This is why data acquisition is required to become flexible.

- **Automatic data processing:** One or more closed control loops could decide to change the type of data to be collected based on system goals. The decision could be taken upon the reception of inputs based on data evaluation and optimization. As none of these tasks are done in a closed loop as they are too costly, and would prevent the loop from running in real-time, they are performed in the Data Ingestion Functional Block and the Normalization Functional Block This will be addressed in a future release.

- **Intelligent data management:** Different types of data require different types of processing. If rule based mechanisms are used, this implies a high level of complexity. However, mechanisms based on learning data characteristics and similar relationships between data are promising and it is recommended to be explored in a future release. These mechanisms enable automatic data label assignment. Label assignment will be addressed in a future release.

- **(Logical) Distributed data storage:** The decoupling of storage and computing resources is important to enable different types of horizontal and vertical storage scaling.

# Annex A:<br>Change History

| Date | Version | Information about changes |
|---|---|---|
| 2021-07 | V0.0.1 | New WI 0009v121 new scope |
| 2021-08 | V0.0.2 | ENI(21)000_190 New headings in new classes & subclauses |
| 2021-09 | V0.0.3 | ENI(21)019_041r1 added definition, abbreviations and 5.9.1 & 5.9.2 description 5.9.3 editor's notes |
| 2021-11 | V0.0.4 | ENI(21)000_233r1 and ENI(21)000234r2 added as approved in call#198 |
| 2021-12 | V0.0.5 | ENI(21)020_045 approved in ENI#20 |
| 2022-01 | V0.0.6 | ENI(22)000001 & ENI(22)000_001r1 & ENI(22)000_002r1 & ENI(22)000_003r1& ENI(22)000_004r1& ENI(22)000_005r1 |

# History

| Document history | | |
|---|---|---|
| V1.1.1 | June 2021 | Publication |
| V1.2.1 | May 2023 | Publication |
| | | |
| | | |
| | | |