# ETSI GS ENI 001 V4.1.1 (2026-05)

**GROUP SPECIFICATION**

# Experiential Networked Intelligence (ENI); ENI Use Cases

***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
RGS/ENI-001v411_Use_Cases

Keywords
artificial intelligence, management, network,
use case

*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

# Contents

# 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 IPR online database.

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™**, **LTE™** and **5G™** logo 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 Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Experiential Networked Intelligence (ENI).

# Modal verbs terminology

In the present document "**shall**", "**shall not**", "**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.

# 1 Scope

The present document specifies a collection of use cases from a variety of stakeholders, where the use of an Experiential Networked Intelligence (ENI) system can be applied to the fixed network, the mobile network, and cloud-based network, to enhance the operator experience through the use of network intelligence. The present document is a revision of ETSI GS ENI 001 [1]. It identifies and describes additional use cases and scenarios and gives the baseline on how the studies in ENI can be applied as solutions to some identified use cases in accordance with the ENI Reference Architecture and will substantially benefit the operators and other stakeholders.

# 2 References

## 2.1 Normative 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.

Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

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 necessary for the application of the present document.

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

[2] ETSI GS ENI 002 (V3.2.1): "Experiential Networked Intelligence (ENI); ENI requirements".

[3] ETSI GS ENI 005 (V4.1.1): "Experiential Networked Intelligence (ENI); ENI System Architecture".

## 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 may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] NGMN Alliance: "Description of Network Slicing Concept", Version 1.0, January 13, 2016.

[i.2] 3GPP TR 23.799 (V14.0.0): "Study on Architecture for Next Generation System (Release 14)", December 2016.

[i.3] ETSI TS 132 101 (V11.4.0): "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Principles and high level requirements (3GPP TS 32.101 version 11.4.0 Release 11)".

[i.4] ETSI TS 128 530 (V15.1.0): "5G; Management and orchestration; Concepts, use cases and requirements (3GPP TS 28.530 version 15.1.0 Release 15)".

[i.5] ETSI GR NFV-EVE 012 (V3.1.1): "Network Functions Virtualisation (NFV) Release 3; Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework".

[i.6] ETSI GR ENI 004 (V4.1.1): "Experiential Networked Intelligence (ENI); ENI terminology".

[i.7] IETF RFC 6645: "IP Flow Information Accounting and Export Benchmarking Methodology".

[i.8] Y. Hu, D. Li, P. Sun, P. Yi and J. Wu: "Polymorphic Smart Network: An Open, Flexible and Universal Architecture for Future Heterogeneous Networks," in IEEE™ Transactions on Network Science and Engineering, vol. 7, no. 4, pp. 2515-2525, 1 October-December 2020, doi: 10.1109/TNSE.2020.3006249.

[i.9] Goldman Sachs: "The global market for humanoid robots could reach $38 billion by 2035".

[i.10] Precedence Research: "Intelligent Virtual Assistant Market Size, Share, and Trends 2025 to 2034".

[i.11] IEEE 802.11™: "IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks--Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

[i.12] Recommendation ITU-R M.2160-0 (11/2023): "Framework and overall objectives of the future development of IMT for 2030 and beyond".


# 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.6] apply.

## 3.2 Symbols

Void.

## 3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

| | |
|---|---|
| 2G | 2nd Generation GSM |
| 3G | 3rd Generation |
| 3GPP | 3rd Generation Partnership Group |
| 4G | 4th Generation |
| 5G | 5th Generation |
| ACNO | Algorithmic Centre for Network Operation |
| AI | Artificial Intelligence |
| AI/ML | Artificial Intelligence/Machine Learning |
| AI-MC | AI-based Mobile Caching |
| AIML | AI Mark-up Language |
| AI-NTC | AI enabled Network Traffic Classifier |
| AN | Access Network |
| AP | Access Point |
| API | Application Programming Interface |
| APP | APPlication |
| AT | AnTenna |
| BBU | BaseBand Unit |
| BRAS | Broadband Remote Access Server |
| BSS | Business Support System |
| CA | Context Awareness |
| CA-KM | Context Awareness-Knowledge Management |
| C-AM | Context-Aware Management (functional block) |
| CAPEX | CAPital EXpenditure |
| CCO | Capacity and Coverage Optimization |
| CFD | Computational Fluid Dynamics |

| | |
|---|---|
| CGN | Carrier Grade Network address translation |
| CMS | Cloud Management System |
| CN | Core Network |
| CogM | Cognitive Management |
| COVID | Coronavirus Disease |
| CP | Cloud Provider |
| CPRI | Common Public Radio Interface |
| CPU | Computing Processing Unit |
| CS | Control System |
| CSMF | Communication Service Management Function |
| CSP | Cloud Service Provider |
| CT | Client/Tenant |
| DaaS | Desktop as a Service |
| DC | Data Centre |
| DDoS | Distributed Denial of Service |
| DevOps | Development and Operations |
| DHCP | Dynamic Host Configuration Protocol |
| DIN | Data Ingestion and Normalization |
| DL | DownLink |
| DNS | Domain Name System |
| DOG | Denormalization and Output Generation |
| DPI | Deep Packet Inspection |
| D-RAN | Distributed RAN |
| E2E | End-to-End |
| eCPRI | electronic Common Public Radio Interface |
| eMBB | electronic Mobile Broad Band |
| EMS | Element Management System |
| ENI | Experiential Networked Intelligence |
| ETL | Extract-Transform-Load |
| FB | Functional Block |
| FCAPS | Fault, Configuration, Accounting, Performance and Security |
| FPS | Frames Per Second |
| FTP | File Transfer Protocol |
| GPU | Graphic Processing Unit |
| GUI | Graphical User Interface |
| HTTP | HyperText Transfer Protocol |
| I/O | Input and Output |
| IANA | Internet Assign Number Authority |
| IBCM | Intent-Based Cloud Management |
| ID | IDentity |
| IDC | Internet Data Centre |
| IDS | Intrusion Detection Systems |
| INFP | Intelligent Network Failure Prevention |
| IoT | Internet of Things |
| IP | Internet Protocol |
| ISP | Internet Service Provider |
| IT | Information Technology |
| KG | Knowledge Graph |
| KM | Knowledge Management (functional block) |
| KPI | Key Performance Indicator |
| KVM | Kernel-based Virtual Machine |
| LLM | Large Language Model |
| LTE | Long Term Architecture |
| MAN | Metro Area Network |
| MANO | MANagement and Orchestration |
| MC | Mobile Caching |
| MCP | Multi-vendor Command Platform |
| MDE | Model Driven Engineering (functional block) |
| MEC | Multi-access Edge Computing |
| MIMO | Multiple Input Multiple Output |
| ML | Machine Learning |
| MOP | Mode Of Operations |

| | |
|---|---|
| MOS | Mean Opinion Score |
| MPLS | Multi-Protocol Label Switching |
| MS | Monitoring System |
| N/A | Non Applicable |
| NF | Network Function |
| NFV | Network Functions Virtualisation |
| NFVI | NFV Infrastructure |
| NFVO | NFV Orchestrator |
| NGFI | Next Generation Fronthaul Interface |
| NGMN | Next Generation Mobile Networks |
| NLP | Natural Language Processing |
| NPO | Network Planning and Optimization |
| NRM | Network Resource Management |
| NSI | Network Slice Instances |
| NSMF | Network Service Management Function |
| NSSMF | Network Sub-Slicing Management Function |
| NWDAF | NetWork Data Analytical Function |
| O&M | Operation and Maintenance |
| OMC | Operation and Maintenance Centre |
| ONNX | Open Neural Network eXchange |
| OPEX | OPerational EXpenditure |
| OPN | Operations/Parameters in the Network |
| OS | Operating Systems |
| OSS | Operations Support System |
| OTN | Optical Transport Network |
| P2P | Point to Point |
| PHY | PHYsical layer |
| PINet | PolymorphIc Network |
| PM | Policy Management (functional block) |
| PoC | Proof of Concept |
| PoP | Point of Presence |
| PTN | Packet Transport Network |
| PTP | Precision Time Protocol |
| PUE | Power Usage Effectiveness |
| QCI | Quality of service Class Identifiers |
| QoE | Quality of Experience |
| QoS | Quality of Service |
| RAM | Random Access Memory |
| C-RAN | Centralized RAN |
| RAN | Radio Access Network |
| RAU | Remote Aggregation Unit |
| RCA | Root Cause Analysis |
| RCC | Radio Cloud Centre |
| RF | Radio Frequency |
| RP | Reference Point |
| RRU | Remote Radio Units |
| RSRP | Reference Signal Received Power |
| RSSI | Received Signal Strength Indicator |
| SA | Service Assurance |
| SDN | Software Defined Networking |
| SD-WAN | Software Defined - Wide Area Network |
| SIA | Service Impact Analysis |
| SINR | Signal to Interference plus Noise Ratio |
| SIP | Session Initiation Protocol |
| SLA | Service-Level Agreement |
| SM | Session Management |
| SNR | Signal-to-Noise Ratio |
| SON | Self-Organizing Network |
| SP | Service Provider |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| TN | Transport Network |

| | |
|---|---|
| TT | Trouble Ticket |
| UC | Use Case |
| UE | User Equipment |
| UL | UpLink |
| UPF | User Plane Function in 5G |
| V2I | Vehicle to Infrastructure |
| V2N | Vehicle to Network |
| V2P | Vehicle to People |
| V2V | Vehicle to Vehicle |
| vBRAS | virtual Broadband Remote Access Server |
| vCPU | virtual CPU |
| VDI | Virtual Desktop Interface |
| VM | Virtual Machines |
| VNF | Virtualised Network Functions |
| VoIP | Voice over IP (Internet Protocol) |
| VoLTE | Voice over LTE in 4G |
| WAN | Wireless Access Network |
| WLAN | Wireless Local Area Network |
| XDR | eXternal Data Representation |
| XML | eXtensible Markup Language |
| YAML | YAML Ain't Markup Language |

# 4 Overview

## 4.1 Background

Operators see human-machine interaction as slow, error-prone, expensive, and cumbersome. For example, operators are worried about the increasing complexity of integration of different standardization platforms in their network and operational environment; this is due to the vast differences inherent in programming different devices as well as the difficulty in building agile, personalized services that can be easily created and disassembled. These human-machine interaction challenges are considered by operators as barriers to reducing the time to market of innovative and advanced services. Moreover, there is no efficient and extensible standards-based mechanism to provide contextually-aware services (e.g. services that adapt to changes in user needs, business goals, or environmental conditions).

These and other factors contribute to a very high OPerational EXpenditure (OPEX) for network management. Operators need the ability to automate their network configuration and monitoring processes to reduce OPEX. More importantly, operators need to improve the use and maintenance of their networks. In particular, this requires the ability to visualize services and their underlying operations so that the proper changes can be applied to protect offered services and resources (e.g. ensure that their Quality of Service (QoS) and Quality of Experience (QoE) requirements are not violated). If such visualization could be provided, then operators would be better able to maintain their networks.

The associated challenges may be stated as:

a) automating complex human-dependent decision-making processes;

b) determining which services should be offered, and which services are in danger of not meeting their Service-Level Agreement (SLA), as a function of changing context;

c) defining how best to visualize how network services are provided and managed to improve network maintenance and operation; and

d) providing an experiential architecture (i.e. an architecture that uses various mechanisms to observe and learn from the experience an operator has in managing the network) to improve its understanding of the operator experience, over time.

The aforementioned challenges will require advances in network telemetry, big data mechanisms to gather appropriate data at speed and scale, machine learning for intelligent analysis and decision making, and applying innovative, policy-based, model-driven functionality to simplify and scale complex device configuration and monitoring. The present document was published as ETSI GS ENI 001 [1].

## 4.2 Overview of the ENI System

### 4.2.1 Brief Description

The purpose of the ETSI ISG ENI is to define a Cognitive Network Management architecture that improves on the operator experience. The operator experience is improved by adding closed-loop mechanisms (including AI functions) based on context-aware, metadata-driven policies for recognizing and integrating new and evolving knowledge, thereby enabling quicker actionable decision-making.

The ENI System is an innovative, policy-based, model-driven functional architecture that improves operator experience. In addition to network automation, the ENI System assists decision-making of humans as well as machines, to enable a more maintainable and reliable system that provides context-aware services that more efficiently meet the needs of the business. For example, the ENI System enables the network to change its behaviour (e.g. the set of services offered) in accordance with changes in context, including business goals, environmental conditions, and the varying needs of end-users.

Examples of the possible functionalities of an ENI System are given in Figure 4-1.

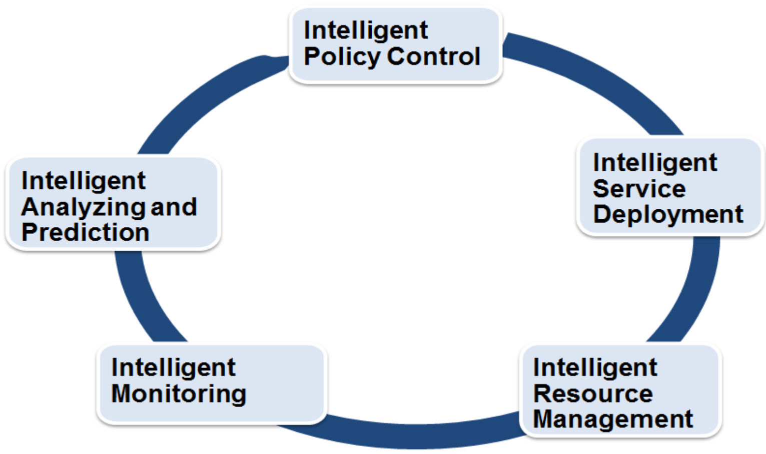


**Figure 4-1: Example of functionalities of ENI System**

### 4.2.2 Expected Benefits

The ENI System provides the following important benefits:

1) to measure and quantify the operation and performance of the resources and services of an operator;
2) to enable personalized services to be provided to customers;
3) to learn from the operation of the network, and decisions made by the operator;
4) to automate the operator's complex human-dependent decision-making processes by translating changing user needs, business goals, and environmental conditions into closed-loop configuration and monitoring;
5) to enable the optimization and adjustment of resources and services managed by the operator, as well as associated tools and applications needed by the operator to conduct business.

ENI System delivers enhanced customer experience by allowing operators to understand the operating status of their network and networked applications in near-real-time, and reconfigure their network. The ENI System automatically collects network status and associated metrics, faults, and errors, and then uses artificial intelligence to ensure network performance and quality of service are met at the highest possible efficiency (e.g. with the minimum required resources). An ENI System can also be used to find bottlenecks of service and/or failure of network. Both of these benefits are done on-demand, in response to changing contextual information.

The ENI System helps to increase the value of services provided by an operator to its customers by rapidly on-boarding new services, enabling the creation of a new ecosystem of cloud consumer and enterprise services, reducing Capital and Operational Expenditures, and providing efficient operations.

# 5 Use cases

## 5.1 Introduction

This clause describes the use cases and scenarios identified by the ETSI ENI ISG. Each use case describes how an ENI System can be applied, and the benefits it provides. Examples of mapping the ENI Reference Architecture (specified in ETSI GS ENI 005 [3]) to specific use cases are also provided. It is noted that such mapping, including the reference points and roles of functional blocks, should only be seen as examples and based on ETSI GS ENI 005 [3]. It is also noted that the applicability of each functional block in terms of what is its role in the overall implementation of the Use Case can only be seen as an example, based on ETSI GS ENI 005 [3]. When ETSI GS ENI 005 [3] changes, and overall doubts have been solved by architecture experts, text on quotes and particular interpretation of the contents may be modified.

The use cases included in the present document are categorized into the following five categories (Table 5-1):

1) Infrastructure Management: This category of use cases covers the processes related to the management of the network infrastructure (e.g. adjustment of allocated and provided services, maintenance, capability specification, and planning). In particular, it is about using policies for managing the network infrastructure, enabled by placing analytics in the control loop and using the results of the analytics as part of the input to policy-based management of the infrastructure.

2) Network Operations: Use cases described in this category are concerned with running the network, where the runtime contexts of the network are extracted and analysed, and the management operations are performed and optimized dynamically at runtime.

3) Service Orchestration and Management: This category of use cases relates to the service and order management, covering processes such as activation using the operator's business channels or customer portals. It is about providing differentiated SLAs for different applications, including vertical applications, through the application of machine learning in an intelligent entity, i.e. ENI. For example, services can be differentiated based on level (e.g. gold vs. silver vs. bronze classes of service) as well as based on the type of application within a level (e.g. a video streaming service has a different service than FTP, even though both are applications that a particular customer has).

4) Assurance: Use cases described in this category are concerned with the functionality of network monitoring, trending, and prediction, as well as taking policy-based actions using knowledge learned from the network to facilitate network maintenance. This includes service runtime operations dedicated to guarantee continuous service delivery.

5) Network Security: Use cases described in this category are related with using AI to tackle network security.

**Table 5-1: Summary of ENI Use Cases**

| **Category** | | | | | |
|---|---|---|---|---|---|
| **1 - Infrastructure Management （clause 5.2）** | Use Case #1-1: Policy-driven IDC Traffic Steering | Use Case #1-2: Handling of Peak Planned Occurrences | Use Case #1-3: DC Energy Saving using AI | Use Case #1-4: Intelligent Optimization for Transmission Network | Use Case #1-5: Energy saving in radio network |
| | Use Case #1-6: Health Management of Data Centre Optical Modules | Use Case #1-7: LLM for Network OAM Application on Generic Computing Platform | | | |

| Category | | | | | |
|---|---|---|---|---|---|
| **2 - Network Operations (clause 5.3)** | Use Case #2-1: Policy-driven IP Managed Networks | Use Case #2-2: Radio Coverage and Capacity Optimization | Use Case #2-3: Intelligent Software Rollouts | Use Case#2-4: Intelligent Fronthaul Management and Orchestration | Use Case #2-5: Elastic Resource Management and Orchestration |
| | Use Case #2-6: Application Characteristic based Network Operation | Use Case #2-7: AI enabled network traffic classification | Use Case #2-8: Automatic service and resource design framework for cloud service | Use Case #2-9: Intelligent time synchronization of network | Use Case #2-10: Intelligent Content-Aware Real-Time Gaming Network |
| | Use Case #2-11: Intent-driven operating for user-centric cloud-network convergence services | Use Case #2-12: Green Energy Efficiency Evaluation for Artificial Intelligent Computing Centres | Use Case #2-13: Intelligent Satellite-Terrestrial Network Optimization | Use Case #2-14: Space-Ground Cooperative Network Slicing | |
| **3 - Service Orchestration and Management (clause 5.4)** | Use Case #3-1: Context-Aware VoLTE Service Experience Optimization | Use Case #3-2: Intelligent Network Slicing Management | Use Case #3-3: Intelligent Carrier-Managed SD-WAN | Use Case #3-4: Intelligent caching based on prediction of content popularity | Use Case #3-6: Intent-based Cloud Management for VDI service |
| | Use Case #3-7: Intelligent vehicle diversified service fulfilment based on polymorphic network | Use Case #3-8: AI based family broadband network user experience optimization | Use Case #3-9: Intent-Driven Home Intranet Management | | |
| **4 - Assurance (clause 5.5)** | Use Case #4-1: Network Fault Identification and Prediction | Use Case #4-2: Assurance of Service Requirements | Use Case #4-3: Network fault root-cause analysis and intelligent recovery | Use Case #4-4: IP Network Congestion Prediction and Prevention | #4-5: Fault detection and diagnosis for IDC infrastructure |
| **5 - Network Security (clause 5.6)** | Use Case #5-1: Policy-based network slicing for IoT security | Use Case #5-2: Limiting profit in cyber-attacks | | | |
| **6 - AI Agents Use Cases Consumer Use Cases** | Use Case #6-1: AI Agents to Enable Smart Life | Use Case #6-2 on Network-Assisted Collaborative Robots | Use Case #6-3 on AI Phone | | |
| **7 - AI Agents Use Cases Business Use Cases** | Use Case #7-1 on AI Agent-based Customized Network for Smart City Traffic Monitoring | Use Case #7-2 on AI Agents-Based Customized Network for Smart Construction Sites | Use Case #7-3 on AI Agents-Based Customized Network for Smart Construction Sites | Use Case #7-4 on AI Agent-Assisted Collaborative Energy Distribution in Power Enterprises | |
| **8 - AI Agents Use Cases Telecom Operator Use Cases** | Use Case #8-1 on AI Agent-Based Autonomous Network Management | Use Case #8-2 on AI Agent-Based Disaster Handling Network Management | Use Case #8-3 on AI Agent-Based Time-Sensitive Network Management | Use Case #8-4 on AI Agent-Driven Core Network Signalling Optimization | Use Case #8-5 on AI Agent-Based Core Networks to Enhance User Experience |

## 5.2 Infrastructure Management

### 5.2.1 Use Case #1-1: Policy-driven IDC Traffic Steering

#### 5.2.1.1 Use case context

This use case relates to intelligent link load balancing and bandwidth allocation between Internet Data Centres (IDCs). The tenants of IDCs are enterprises that have requirements that dynamically adjust service and/or resource behaviour (e.g. reliable network connectivity and changes to an offered service based on network load).

There are a number of problems with how current traffic steering is performed between IDCs. These include the use of multiple possible links between IDCs (e.g. which link is the best to use at a given time). Currently, the link for a tenant is normally determined as the shortest path between the IDC that the tenant resides in and the IDC that the tenant is connecting to. In addition, the link load is not considered when calculating the traffic path. Furthermore, the bandwidth allocated to a tenant is not always fully used.

#### 5.2.1.2 Description of the use case

##### 5.2.1.2.1 Overview

Operators are deploying IDCs in Metropolitan Area Networks (MANs) to provide network access with load-balancing and resiliency. Current network configuration practices include:

- In order to provide service assurance for important tenants, network administrators typically schedule the traffic in specific periods. Traditional network management is usually complex, with a long cycle caused by manual actions, so it is difficult to meet the requirement of real-time traffic optimization.
- Large service provider's traffic usually is sensitive to the events of a day. For example, online big sales and usage of social media with video steaming cause a significant increase in traffic. This means that the network administrator cannot provide bandwidth assurance for some important tenants.
- The bandwidth requirements of tenants tend to change dynamically. Traditional static bandwidth allocation leads to low bandwidth utilization and redundancy.
- The imbalance across multiple links leads to inefficient resource utilization. For example, it is possible that the utilization of a link reaches a certain threshold, while other links' loads remain low.

##### 5.2.1.2.2 Motivation

The ENI System can be used to achieve intelligent link load balancing and intelligent bandwidth allocation. In ENI, policies can be modified by using machine learning to fill in important parameters, such as available links, link bandwidth, real-time link utilization, and other predefined constraints. Three examples of the predefined constraints to be considered before modifying the policies are:

1) each link is predefined with a threshold of the maximum bandwidth and cannot be exceeded;
2) flow of a client at a specific service level (e.g. gold) cannot be switched;
3) the maximum times of switching specific service from one link to another link is predefined and cannot be exceeded.

Such policies can be used to better manage the network and achieve autonomous service traffic monitoring and network resource optimization. It can also be used to adjust the service along different links of an IDC, thus improving the operator's experience through enhanced network resilience and service QoS and QoE.

The ENI System also:

- predicts changes by using AI in the tenant's service requirements based on historical data (e.g. the type of QoS to be provided for a given service based on the type of application and metadata);

- collects and analyses real-time data, given the service adjustment recommendations (e.g. which metadata and metrics to monitor based on the type of service and the type of changes applied);
- corrects the prediction result according to the adjustment recommendations, and converges to an ideal service management policy;
- analyses QoS and other applicable data and metadata to make the final service policy modifications; this is then stored as a reusable set of objects.

By using the above intelligent service adjustment policy provided by the ENI System, real-time, dynamic, and automated resource allocation and adjustment to the service can be achieved. The bandwidth utilization is improved. Meanwhile, it provides bandwidth assurance for important tenants according to the service level.

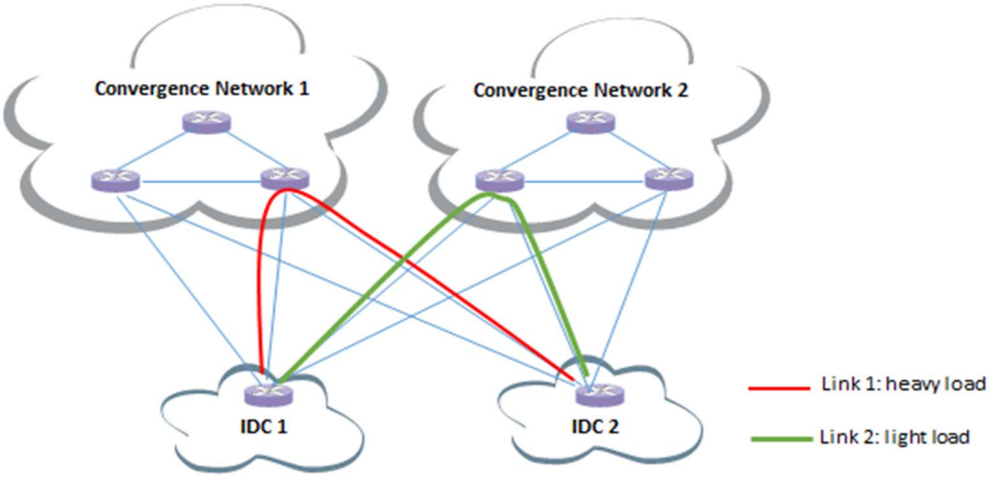


**Figure 5-1: Policy-driven, automatic IDC traffic steering**

As shown in left portion of Figure 5-1, two IDCs can connect to each other via two different paths. There are multiple links between the two IDCs. When link 1 is heavily loaded, as much traffic as necessary can be moved to link 2.

### 5.2.1.2.3 Actors and Roles

- IDC network.
- ENI System.
- Network manager (City Level).
- Operators.

### 5.2.1.2.4 Initial context configuration

- the network administrator's inputs the policies;
- IDCs connect to each other via different links;
- network traffic routed via different links is defined according to policies;
- bandwidth of tenant may need to be adjusted in real time according to the dynamic needs of the tenant and the operational context.

### 5.2.1.2.5 Trigger conditions

- Utilization of a link or bandwidth of a tenant exceeds the configured threshold (e.g. as defined in an SLA).
- Change in operational requirements.

#### 5.2.1.2.6 Operational Flow of the actions

For intelligent link load balancing:

1) network administrator pre-configures the threshold/constraint of link utilization and appropriate metadata and metrics to monitor link loads;

2) ENI System uses the network administrator's input to modify policies considering, for example, available links, bandwidth, link utilization, and constraints;

3) network administrator executes policies and ensures they all execute correctly (i.e. without error);

4) the ENI System adapts to monitor metrics, metadata, and other information (as defined by the above generated policies) to achieve measured improvements;

5) IDC network uses the policies to manage the network behaviour.

For intelligent bandwidth allocations:

- network administrator pre-configures the threshold of bandwidth utilization and appropriate metadata and metrics to monitor bandwidth;
- ENI System collects the bandwidth usage data for each tenant for a specified time period;
- ENI System pre-processes the data to extract the appropriate characteristics of the tenant's service to determine if the allocated bandwidth is sufficient or not;
- ENI System establishes an appropriate mathematical model to predict the bandwidth requirements of tenants at different times in the coming year;
- ENI System collects and analyses real-time bandwidth usage data for tenants;
- when the configured bandwidth utilization threshold is danger of being reached, the ENI System proactively adjusts the bandwidth allocation policies for the affected tenants, taking into account the QoS policy and other SLA policies of each tenant.

#### 5.2.1.2.7 Post-conditions

The impact of dynamic polices:

- Network traffic is balanced.
- Appropriate metrics and metadata are continually gathered to ensure that the service requirements are met or exceeded.
- Bandwidth for the tenant is assured and automatically adjusted in real-time.
- Bandwidth utilization is improved.

## 5.2.2 Use Case #1-2: Handling of Peak Planned Occurrences

### 5.2.2.1 Use case context

Currently, most services share a common infrastructure where resource allocation is a very critical process. When a network operator extends its infrastructure to a new area or upgrades an already existent, it makes an assessment on the number of customers and services the infrastructure will serve under normal operation scenarios. Then, when provisioning services, the configuration of the infrastructure is performed once and usually does not change during the service lifecycle. Although advanced QoS strategies help to mitigate peak resources usage scenarios, it still constitutes a very static and slow process for today's services, which imply the need for the process to become more and more dynamic. Moreover, when considered as adequate and feasible, a network operator may make use of mobile stations for such temporary increases on network capacity.

Typical peak scenarios may be characterized by the occurrence of localized and temporary bursts of network traffic caused by planned events, e.g. soccer games, or unplanned, e.g. natural catastrophes, which may lead to critical service level degradation or even service disruption along with the subsequent impact in operators, in services as well as in end user's experience. In particular, for network operators, service degradation and/or disruption constitutes something that is to be avoided no matter at what cost as it jeopardizes network operator's image as a service provider among its customers.

In the present Use Case, only planned events will be taken into account.

#### 5.2.2.2 Description of the use case

##### 5.2.2.2.1 Overview

Service prioritization and management of resource sharing infrastructures are very complex processes for operators, which take a considerable amount of time for planning, and are normally performed only once in a given area. When dealing with temporary planned events, it is necessary to calculate the stress on the network infrastructure and define backup action plans to mitigate potential service degradation or even disruptions. Additionally, after the end of the event it is necessary to revert the temporary changes to the normal usage conditions.

An example of such a temporary planned event could be the case of a certain area, served by a network infrastructure for telecommunication services, which will be hosting a music event that will be broadcasted by live television. Currently, the network infrastructure is providing resources to several service instances in a shared manner. A relatively large crowd is expected at the event and if no actions are taken by the network operator there is the possibility of degradation on some of the services that make use of that region's infrastructure.

Analysis performed during the planning of events may also encompass the ability to extend the current infrastructure capacity of that area.

The current Use Case is further described by the following set of components and features.

##### 5.2.2.2.2 Motivation

With the ENI System, the use of AI methods on helping to understand the context dynamicity and on predicting potential peak traffic scenarios becomes quite important. More specifically, the AI can perform the calculation of possible scenarios for planned events by making use of machine learning, e.g. by taking into account events history. On the other hand, it can also assist on the calculation based on the expected response of network equipment under stress, which can also help on the preparation and definition of the necessary backup action plans. In addition, the ENI System can also evaluate, for all these scenarios, if the use of resource sharing techniques is enough to support the increase of network traffic or if there is the need for additional measures, e.g. mobile stations that provide additional physical resources.

Still another benefit related to the AI capability to provide more realistic predictions lies in the possibility for network operators to use narrower margins of the total amount of resources when they wish to extend the current resource capacity of a given region. With these new tools, network operators may enforce pre-defined policies to govern the responses of the ENI System, e.g. do not use mobile stations if the peak consumption is not expected to exceed 90 % of the current network capacity.

##### 5.2.2.2.3 Actors and Roles

The presence of the following actors/entities as well as their associated roles are envisaged in the current use case:

- Customers/clients: end users that enjoy the delivery of a service.
- Network Administrator: entity/person responsible for the initial policy design that encompasses the planning of the Network Infrastructure regarding the mitigation impact of planned events, which may involve the extension of the infrastructure capacity. With the assistance of the ENI System regarding these planning activities, this entity/person is in a position to choose the most suitable backup action/plan to be enforced.
- Network Infrastructure: network elements and resources that participate in service fulfilment procedures.
- Network Operator: owner of the Network Infrastructure that is used to provide services to customers/clients.

- Operations Support System/Business Support System (OSS/BSS): operational and business systems that belong to the management system of network operators. In this case they are providing, among others, monitoring, actuation, internal records of very different items that may range from products to resources, as well as other business interfaces dedicated to external entities.

- ENI System: component that governs service fulfilment and participates in planning and configuration procedures upon occurrence of planned scenarios that may impact service delivery, which may encompass situations involving extension of infrastructure capacity.

#### 5.2.2.2.4 Initial context configuration

The network is operating in perfect conditions with all its components working in good shape.

#### 5.2.2.2.5 Triggering conditions

A music event is scheduled for a certain area and may lead to local service degradation or disruption. On occurrence of the planned event, backup actions, previously calculated by the ENI System and validated by the Network Administrator, are triggered. Those backup actions may encompass extensions on infrastructure capacity.

#### 5.2.2.2.6 Operational flow of actions

The following sequence of actions may be identified:

1) After receiving a notification of a new event e.g. a music festival, the ENI System makes use of AI methods to calculate and produce a report containing several scenarios and their respective outcome depending on the size of the crowd and expected local resource consumption. For each scenario, it also produces a backup action plan, i.e. possible changes to local QoS profiles or additional resources needed, taking into account previously defined policies.

2) In its notification report, the ENI System identifies the most suitable scenario and asks the Network Administrator for validation.

3) The Network Administrator evaluates the proposed scenarios and backup plans, validates one of them, and notifies the ENI System about its choice.

4) Upon receiving the Network Administrator's reply, the ENI System elaborates a schedule containing a roadmap of the backup actions to be subsequently performed.

5) On occurrence of the planned event, the ENI System triggers the proper configuration operations via OSS components on impacted network infrastructure resources, including possible redundant resources that may be reserved for any deviation on the predicted consumption.

6) During the event, the ENI System increases the monitoring resolution on the previously mentioned resources.

7) If found as necessary, it may activate additional resources, if available, previously reserved for any deviation on the predicted consumption.

8) At the end of the event the ENI System triggers the rollback of the network resources configuration to the state where it was immediately before the event.

#### 5.2.2.2.7 Post-conditions

The local network infrastructure is operating according to the planned deployment prior to the event. All information regarding network infrastructure during the event is stored to increase the prediction capabilities of the ENI System.

## 5.2.3 Use Case #1-3: Energy optimization using AI

### 5.2.3.1 Use case context

By introducing Network Functions Virtualisation (NFV) different virtual networks can be deployed on the same NFV Infrastructure (NFVI) for different network services. The Virtual Network Function (VNF) instances are implemented on Virtual Machines (VMs) or Containers. And the VNF instances can be instantiated, scaled in/out, or terminated on demand by using Management and Orchestration (MANO) system or any other form of orchestrator. The VNF instances can be easily moved from one server to another server by using VM/Container migration technologies. Therefore, the services provided by the VNFs can be steered from one server to another server along with the VM/Container migration, while the network services provided by the VNFs are uninterrupted.

With the trend of NFV, more and more DCs will be deployed to replace the traditional Central Offices in the operators' network. The Data Centres (DC) are made up of many servers with huge power consumption. Typically, the servers in a DC take 70 % of the total power consumption. The other equipment including switches, routers, storage equipment and air conditioners take the other 30 % of the total power consumption. The servers are deployed and running to meet the requirement of peak hour service, which means the servers are normally at high power-up state at full time even in non-peak hours. It is however possible to move the services to some of the servers and turn the other servers to idle or underclocking state in non-peak hours, with the aim of optimizing the power usage at the DC. It should be noted that such mechanism of energy optimization can be applied widely to other network resources in addition to data centres.

As another important energy-consuming component in the data centre, cooling system usually has the problem with cooling capacity redundancy. The temperature of the air conditioner is often adjusted to a very low level to ensure that the server will not overheat and even trigger a temperature alarm. This will result in a high energy consumption and carbon emission. With the growing number and size of DCs, it is difficult for traditional manual tuned instruction based policy to achieve the desired energy saving goal. An ENI System could identify and act on these requirements. By using AI technology, the best parameters combination can be found and adjusted to realize the lowest Power Usage Effectiveness (PUE) under the condition of meeting the temperature requirements. Optimized refrigeration equipment strategies can be further generated according to the real-time data and prediction of IT load and energy consumption. Intelligent energy optimization management will help Communication Service Providers (CSPs) reduce operating costs and increase efficiency of Operations and Maintenance (O&M) without sacrificing customer service quality.

In the following, reducing waste energy in individual DC is used as an example for this use case to elaborate how NFV and AI can be combined to optimize usage of the energy in networks.

### 5.2.3.2 Description of the use case

#### 5.2.3.2.1 Overview

Traditional ways of DC energy saving are normally done manually and the effect is not obvious. Power consumption of the DCs, same as the other network physical resources, represents a large portion of the cost for operators, and causes environmental concerns. Consisting primarily of a homologous architecture/resource pool, the scope of what can be optimized in an intra-DC context is limited. It is however, a necessary first step towards greater AI-driven improvements that are realizable with the additional consideration of both inter-DC orchestration and/or the exploitation of heterogeneous network resource pools (such as edge or IoT devices). The consideration of these additional factors will enable the minimization of the carbon footprint through intelligent resource management. For example, by relying solely on edge device compute resources in periods of low demand and adjusting the cooling equipment settings when the cooling capacity is redundant in the data centre, an ENI System could identify and act on these requirements in an autonomous fashion ensuring that OPEX is optimized, among other Key Performance Indicators (KPIs).

#### 5.2.3.2.2 Motivation

By using ENI System, the usage pattern of the services can be learned from historical data and updated in real-time way. The ENI System can help to trigger the movement of the services and turn the spare servers to idle state. As shown in Figure 5-2, if the actual load of service in one day is represented by a curve, then the shadow between the peak and the curve is potential energy saving for the DC. The optimization may take information from multiple sources and predict and analyse in an autonomous way.

In addition, the ENI System can predict the peak hours by using artificial intelligence techniques such as deep learning or machine learning, and then wake up necessary number of servers into full load state. If an unexpected event is detected, more servers can be woken up to support this burst. With the help of simulation techniques such as Computational Fluid Dynamics (CFD), cooling and heating air distribution flow can be analysed and optimized. E.g. CFD combined with the server workload can provide a basis for predicting a future hot point and thus provide adequate cooling and steer towards where needed most. By using ENI System and AI techniques, the energy saving for DCs can be achieved and OPEX can be saved.

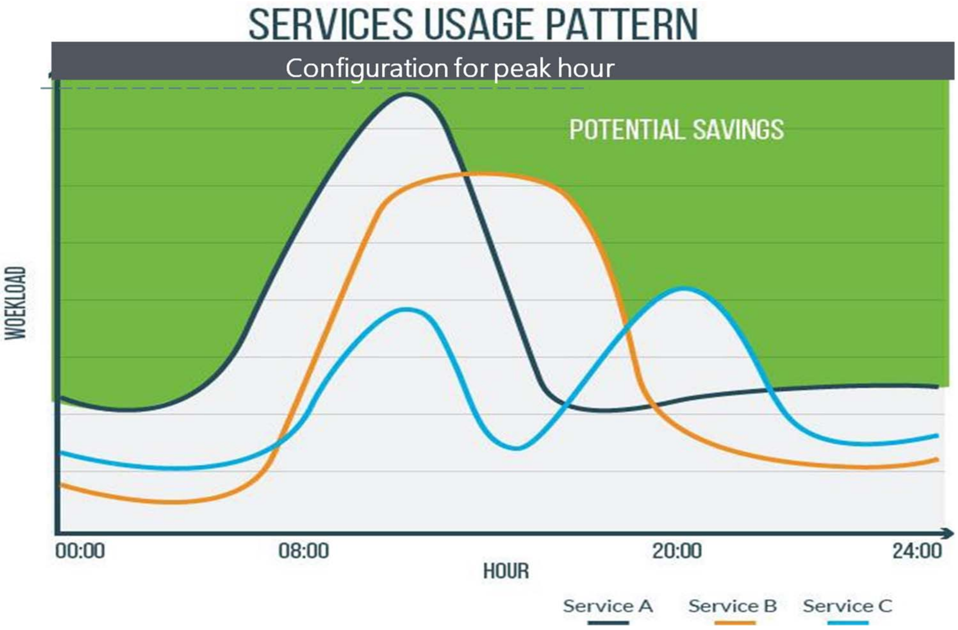


**Figure 5-2: Potential DC energy saving by AI**

#### 5.2.3.2.3 Actors and Roles

- Operator: manages the DCs and confirms the VM/Container migration policies and scale in/out policies, as well as other energy optimization related policies.
- ENI System: collects and learns service pattern from the data collected from the DC servers or collects operating parameters from DC environmental monitoring and control system and learn the relationship between PUE and collected characteristic parameters; determines the VM migration policies and scale in/out policies according to prediction of the service requirements or determines the energy saving policies for cooling system according to the inferred best combination of profile parameters; triggers steering of the service flows from one VM to another VM or distributing the parameter configuration strategies to the data centre control system.
- DC servers: provide the required information to the ENI System, execute the VM migration and VNF scale in/out according to the policies.
- DC environmental monitoring and control system: provide the required information (such as air conditioning setting parameter data, cabinet environmental data, equipment energy consumption data, etc.) to the ENI System, and execute the operation of environmental adjustment.
- NFV MANO: executes the lifecycle management operation of the VNFs according to policies.

#### 5.2.3.2.4 Initial context configuration

All servers in the DC are running all time and the energy consumption is high. The ENI System performs some initial actions related to the collection of information, use of AI algorithms and service patterns learning.

#### 5.2.3.2.5 Triggering conditions

The following trigger types associated with the ENI System may be identified:

- The ENI System predicts that the required resources of a service will fall below a certain threshold in a certain period.
- The ENI System predicts that the required resources of a service will grow up higher than a certain threshold in a certain period.
- The ENI System predicts that the IT load will change beyond a certain threshold.
- The ENI System predicts that the ambient temperature will be higher than a certain threshold.
- The ENI System decides to change the DC environmental settings.
- The ENI System detects a change of the service pattern learned before.

#### 5.2.3.2.6 Operational flow of actions

The following initial sequence of actions may be identified:

1) The ENI System collects and stores information of the virtual networks, including CPU usage, storage usage, and network usage for each VNF, etc. as well as the power consumption information and environmental information.

2) The ENI System uses AI algorithm to build the relations between the network service and its required resources, and the relations between the power consumption and the environment settings including e.g. the location of the running servers, the setting of the cooling system, etc.

3) The ENI System learns the service pattern and predicts the required resources of the service and the IT load changes in a certain period in the future, e.g. the next hour.

The following triggers and subsequent actions may be identified:

1) When the ENI System predicts that the required resources of a service will fall below a certain threshold in a certain period, and the service configured by the operator as able to be moved, the ENI System triggers, directly or indirectly, the NFV MANO system to migrate the services and VMs/Containers providing this service to another selected server:

   a) If the VMs/Containers on one server are all migrated to another server, the spare server is turned into idle mode.

2) When the ENI System predicts that the required resources of a service will grow up higher than a certain threshold in a certain period, the ENI System triggers the scale out of the existing VNF and bring up new VMs/Containers:

   a) If the running servers cannot provide the required resources of a new VM/Container according to prediction, the ENI System wakes up a selected idle mode server.

3) When the ENI System predicts that the system load will change beyond a certain threshold, the ENI System will start the next round of reasoning and optimization. It infers the optimal strategies based on the ambient temperature, load rate and cooling system operation data, selects the best optimal policies to deliver, and executes the policies through the control system.

4) When the ENI System predicts that the ambient temperature will be higher than a certain threshold, the ENI System will also start the next round of reasoning and optimization and deliver optimal policies.

5) The ENI System may decide to change the DC environmental monitoring and control system to adjust the environmental settings when a server is woken up or turned into the idle mode.

6) When the ENI System detects a change of the service pattern learned before, the ENI System will adjust the VM/Container migration policies and scale in/out policies.

#### 5.2.3.2.7 Post-conditions

Servers in the DC are dynamically turned to idle and waken up according to the service pattern; Refrigeration equipment in the DC are dynamically adjusted configuration parameters according to the IT load and environment parameters; therefore the cost of power consumption is reduced as much as possible.

### 5.2.3.3 Mapping to ENI reference architecture

#### 5.2.3.3.1 Functional blocks

The mapping to ENI architecture for energy optimization using AI is shown in Figure 5-3.

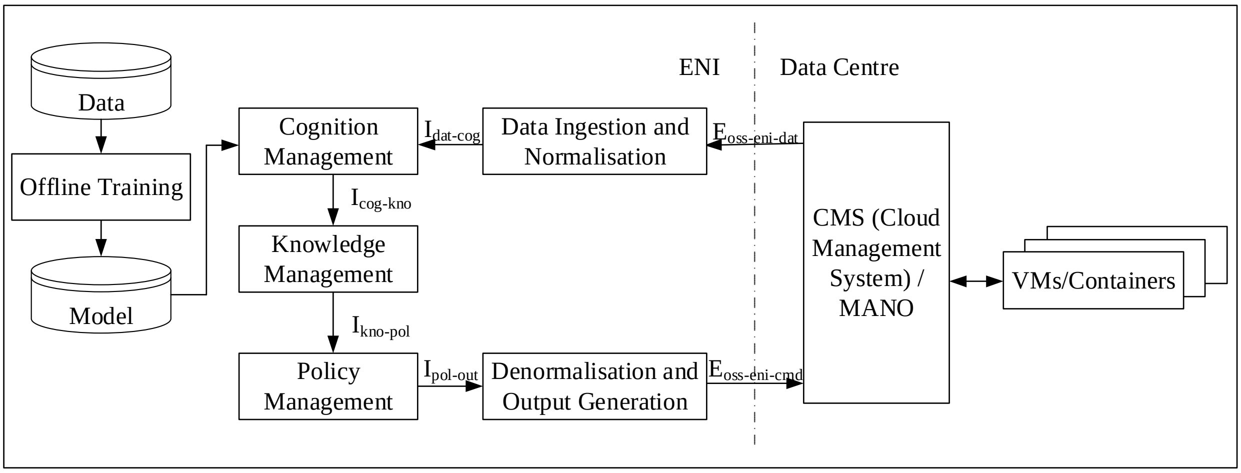


**Figure 5-3: Mapping to ENI reference architecture**

Data Ingestion and Normalization Functional Block converts the data collected from the DC servers into normalized form so that the ENI System can analyse and understand.

Cognition Management Functional Block evaluates the existing knowledge and performs inferences using the trained model and ingested data to make predictions about the future service requests, then determines if any operation should be taken to achieve the goal of energy saving.

Knowledge Management Functional Block stores the generated knowledge and defines a formal and consensual representation of knowledge so that the computer system could implement the machine learning algorithms and perform reasoning using the knowledge representation.

Policy Management Functional Block provides decisions about server consolidation and live migration, as well as decisions about air conditioning operating parameters adjustment to optimize energy utilization and guarantee the Key Performance Indicators (KPIs).

Denormalization and Output Generation Functional Block converts information (recommendations and decisions) generated by the ENI System to a form that the Cloud Management System (CMS)/MANO/Control System can understand and execute.

#### 5.2.3.3.2 Interfaces

$E_{oss\text{-}eni\text{-}dat}$ defines data exchange between the ENI System and the Data Centre (Assisted System). The Cloud Management System in the data centre collects data from virtual machines and send it to ENI System.

$E_{oss\text{-}eni\text{-}cmd}$ defines recommendations and/or commands and acknowledgements exchanged between the ENI System and the Data Centre (Assisted System). The ENI System provides decisions about server consolidation, live migration and air conditioning operating parameters adjustment so that the Cloud Management System and Control System could take actions accordingly to save the energy and OPEX for DCs.

$I_{dat\text{-}cog}$ defines the internal interface between Data Ingestion and Normalization Functional Block and Cognition Management Functional Block, which passes the normalized data and information to the Cognition Management Functional Block to perform inferences and generate new knowledge.

$I_{cog\text{-}kno}$ defines the internal interface between Cognition Management Functional Block and Knowledge Management Functional Block, which passes the generated new knowledge (e.g. predictions of future service requirements or IT load change) to the Knowledge Management Functional Block for storage.

$I_{pol\text{-}dat}$ defines the internal interface between Policy Management Functional Block and Denormalization and Output Generation Functional Block, which passes the recommendations/commands so that it can be converted to a form that CMS/MANO/CS can execute and implement.

#### 5.2.3.3.3 Flow of information

The flow of information is given in Figure 5-4.

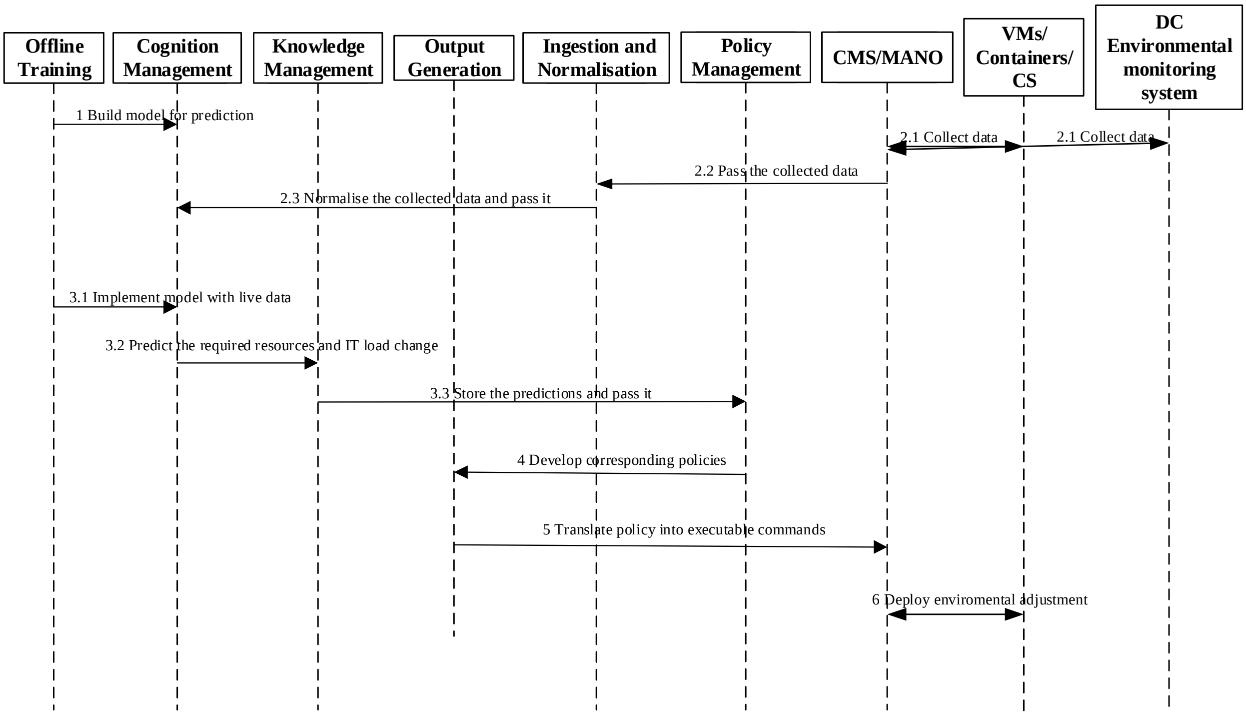


**Figure 5-4: Procedure for energy optimization using AI**

Step 1: Choose the appropriate model and train the model with historical data.

Step 2.1: The Cloud Management System collect data (e.g. CPU usage rate, storage usage rate, network throughput, air conditioning setting parameter data, cabinet environmental data, equipment energy consumption data) from virtual machines/containers and DC environmental monitoring system.

Step 2.2: The collected data is sent back to the ENI System.

Step 2.3: The Ingestion and Normalization Functional Block converts data from multiple input sources into a normalized form and feed the information to the pre-trained model.

Step 3.1: Fine-tune and implement the model with live data.

Step 3.2: The Cognition Management Functional Block analyses the existing knowledge and generate new knowledge (e.g. predictions about the required resources of the service or predictions about IT load change).

Step 3.3: The Knowledge Management Functional Block stores the prediction and send it to the Policy Management Functional Block.

Step 4: Develop policies about server consolidation and live migration or air conditioning operating parameters adjustment according to the predictions of the future service flow and IT load change.

Step 5: Convert policy into a form that Cloud Management System could understand and take actions.

Step 6: The Cloud Management System implements the related adjustment on the virtual machines/containers or Control System.

## 5.2.4 Use Case #1-4: Intelligent Optimization for Transmission Network

### 5.2.4.1 Use case context

Nodes of transmission network can be logically divided into three types: Access Layer Node, Convergent Layer Node and Backbone Layer Node. Access Layer Nodes are generally deployed as user access points. Convergent Layer Nodes are usually deployed at sites with convenient optical routing and a large number of optical cables, see Figure 5-5 below. Multiple Access Layer Nodes converge at the same Convergent Layer Node. And multiple Convergent Layer Nodes converge at the same Backbone Layer Node.

The topology of the traditional transmission network is chain architecture, which is easy for extension. However, this type of transmission network will be split into multiple isolated areas when a single node or link fails, which seriously affects the transmission capacity and performance. In view of this disadvantage, current transmission network is deployed in ring architecture. This type of transmission network has high robustness and reliability where failure of single node or link will not lead to network partitions.

Based on the characteristics of ring architecture, in the traditional capacity management approach, the Virtual Networks Function (VNF) among the transmission ring will be expanded to ensure ring capacity requirement of business transmission, when the capacity usage rate of transmission ring reaches a certain threshold. However, this management mode lacks information linkage between transmission rings, which means different transmission rings cannot perceive the service load among each other. This mode will lead the redundant expansion operation, resulting in low utilization rate of the overall capacity of the transmission network, and increasing the construction cost. In order to utilize the network resources more reasonable, operator puts forward related manual-decision optimization plans. However, the formulation of the final optimization plan is highly dependent on the manual analysis and decision based on expert experience in related fields, leading the process tedious and time-consuming.

With the advent of 5G and saturation of telecom industry market, higher requirements are put forward for the guarantee of transmission network capacity, that the guarantee of capacity is bound to be carried out with lower cost and higher efficiency.

### 5.2.4.2 Description of the use case

#### 5.2.4.2.1 Overview

In order to ensure the optimum capacity utilization of transmission network and avoid unnecessary capacity expansion operations, operator conducts optimization analysis of the transmission network first. Then operator decides the optimization method and formulates the optimization plan. After that, operator will optimize the transmission network according to the optimization plan. Generally, optimization methods include:

1) Keep the link connection between VNFs unchanged, and expand the capacity of targeted VNF(s) among transmission ring with excessive capacity usage rate.

2) Keep the link connection between VNFs and the capacity of each VNF unchanged, then add VNFs to form new transmission ring.

3) Keep the overall topology of transmission network and capacity of each VNF unchanged, then adjust the service load on each transmission ring.

Due to the lack of information linkage between transmission rings, the capacity usage rate of each transmission ring cannot be dynamically perceived. However, when the capacity usage rate of a ring exceeds the pre-determined health threshold, the other rings are usually in a state of idle or low usage rate. Therefore, in order to reduce the optimization investment as much as possible, the following method is preferred:

- Keep the capacity of each VNF unchanged, then adjust the link connection between VNFs to achieve the globally optimum utilization. If the capacity usage rate of the transmission ring is still over specific threshold, then carry out capacity expansion operation on the relevant VNFs among this transmission ring.

This method adjusts the network topology to improve capacity utilization and service load-balance effect. After topology optimization, if the capacity of transmission ring is still unable to meet the service requirement, then the relevant VNFs can be expanded.

### 5.2.4.2.2 Motivation

Currently, the formulation process of optimization plan, with high degree of manual involvement, is complicated and time-consuming, leading to slow progress of transmission network optimization and increased labour cost. These characteristics are not conducive to the operators to increase efficiency and revenue with reduced cost.

In this use case, the ENI System makes intelligent analysis based on AI/graph theory algorithm first, and then starts to decide topology optimization plan by reorganizing the link connections between VNFs and forming new transmission rings. After that, the link connections between VNFs of transmission network shall be globally optimum, reaching the optimum capacity utilization rate. As a result, the investment can be reduced to the greatest extent.

In Figure 5-5, as shown in the Original Transmission Network Link Architecture, the overall capacity usage rate of Access Ring 1 has exceeded the health threshold. On the premise of keeping the existing VNF capacity unchanged, the link connections between VNFs shall be adjusted to ensure capacity usage rate within the health interval. Based on intelligent analysis, ENI System outputs the optimization plan according to pre-determined optimization policy, shown as the Optimized Transmission Network Link Architecture. The process of scheme decision involves two steps:

1) ENI System removes the link connections between VNFs among Access Ring 1 and Access Ring 2.

2) ENI System moves the VNF with red circle shown in Figure 5-5 into Access Ring 2, and reorganizes the link connections between VNFs among the optimized Access Ring 1 and Access Ring 2.

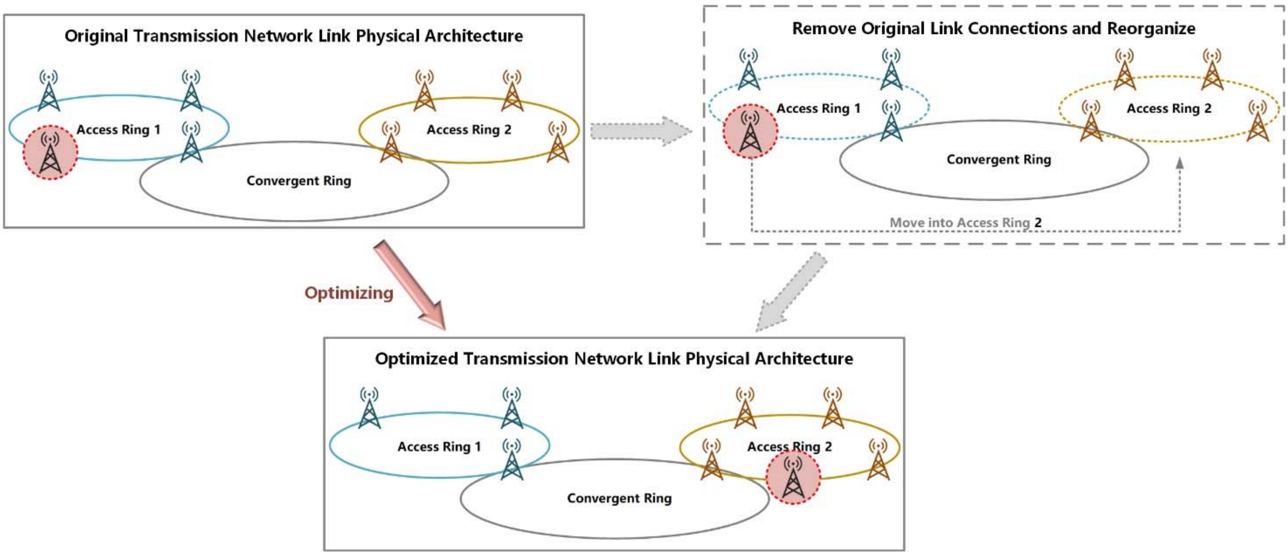


**Figure 5-5: Original and Optimized Transmission Network Link Physical Architecture**

### 5.2.4.2.3 Actors and Roles

- Transmission Network: provides signal transmission and conversion (e.g.: Transmission network options may include OTN, PTN, etc.).
- ENI System: collects the detailed information of transmission rings which are needed to be optimized, then intelligently implements analysis based on AI/ Graph theory algorithm, and determines the optimization plans.
- Network Operator: manages the network topology and the configuration of VNFs.

### 5.2.4.2.4 Initial context configuration

- Both topology of transmission network and information of Virtual Network Function (VNF), such as VNF capacity and the peak-hour data flow, have been input into ENI System.

- Optimization policy has been input into ENI System.

#### 5.2.4.2.5 Trigger conditions

- The capacity usage rate of one transmission ring exceeds the specific threshold.

#### 5.2.4.2.6 Operational Flow of the actions

1) Operator inputs the transmission network topology and detailed information of VNFs among the transmission ring needed to be optimized.

2) Operator pre-configures the optimization policy.

3) The trigger condition is met.

4) ENI System analyses the relation among capacity of transmission ring, topology of transmission ring and the capacity of VNF, based on the transmission network topology and detailed VNF information.

5) ENI System determines and outputs the optimization plan based on AI/ Graph theory algorithm, according to the optimization policy.

6) Operator optimizes the transmission network according to optimization plan.

#### 5.2.4.2.7 Post-conditions

Based on the predetermined optimization policy, the transmission network in the optimization plan shall achieve the globally optimum capacity utilization.

## 5.2.5 Use Case #1-5: Energy saving in radio network

### 5.2.5.1 Use case context

Compared with 4G radio network, 5G radio network has higher frequency band and weaker coverage, so the number of base stations needed increases greatly, and the energy consumption also increases disproportionately as a result if not addressed. The power consumption of 5G base station is about 2,5 - 3,5 times of 4G base stations. At the same time, the complexity of network and the difficulty of operation and management make OPEX increase rapidly.

### 5.2.5.2 Description of the use case

#### 5.2.5.2.1 Overview

This use case is to use AI capability from ENI System to realize energy saving for radio network. It will collect some data like Fault, Configuration, Accounting, Performance and Security (FCAPS) from Operation and Maintenance Centre (OMC), weather/environment information from external system and service/core network performance statistics from Deep Packet Inspection (DPI) system. ENI System uses these data to evaluate the cell efficiency of radio network and generate energy saving policies. The whole procedure is illustrated in Figure 5-6 as a block diagram, where the used acronyms stand for:

- PM: Performance Management;
- NRM: Network Resource Management;
- data lake:
  - an entity to store and collect data;
  - CN: Core Network;
  - AN: Access Network.

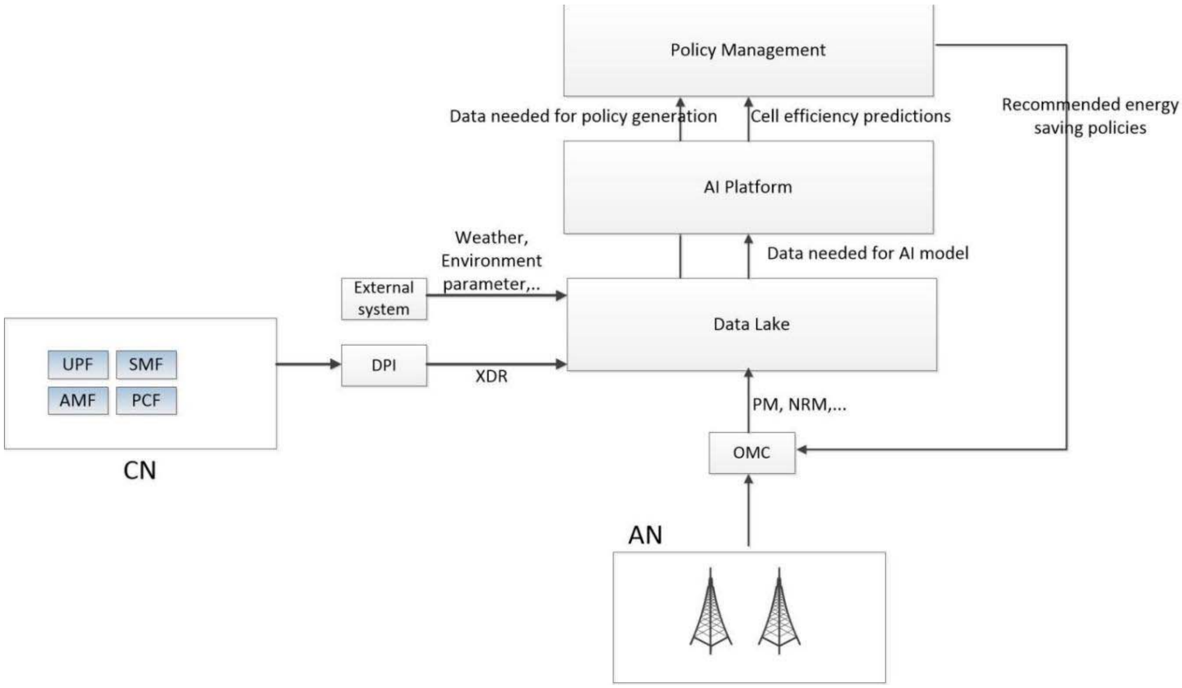


**Figure 5-6: Energy saving for radio network**

- XDR: eXternal Data Representation as a data representation is expected to be used in many places in the diagram above.

### 5.2.5.2.2 Motivation

The 5G energy saving is the most urgent problem to be solved by operators in 5G commercial deployment and belongs to one issue of cost management. It is expected that ENI System can provide intelligent energy saving services by leveraging AI capability and save the energy of radio network accordingly.

### 5.2.5.2.3 Actors and Roles

The key actors are listed below:

| | |
|---|---|
| Operator: | trigger the energy saving procedure/request. |
| ENI System: | collect the data, predict cell efficiency of radio network, provide energy saving polices. |
| DPI system: | provide XDR data to ENI System for cell efficiency prediction. |
| OMC: | provide FCAPS data to ENI System cell efficiency prediction. |
| External system: | provide weather/environment information to ENI System cell efficiency prediction. |
| Base station: | the targeted object for energy saving. |

### 5.2.5.2.4 Initial context configuration

- The energy saving policy is implemented in ENI System.
- The ENI System has trained the AI model continuously and learned how to utilize the data for cell efficiency prediction in order to provide energy saving policies to radio network.

### 5.2.5.2.5 Triggering conditions

- The operator triggers the energy saving procedure to ENI System which collects the relevant data and carries out energy saving procedures.

#### 5.2.5.2.6 Operational flow of actions

1) The operator sends the energy saving requests to ENI System.
2) The ENI System collects the data from different sources.
3) The ENI System predicts the cell efficiency and provide the energy saving policies to OMC.
4) OMC sends the energy saving policies to base stations.

#### 5.2.5.2.7 Post-conditions

The Base Stations set of the AN that makes part of the 5G Radio Network is operating according to the new energy savings predicted by the ENI System.

## 5.2.6 Use Case #1-6: Health Management of Data Centre Optical Modules

### 5.2.6.1 Use case context

This use case involves monitoring and predicting the health status of optical modules in a Data Centre (DC).With the implementation and application of emerging technologies such as cloud computing, big models, and AIGC, there is a demand for higher data transmission rates and lower latency in new data centres, further increasing the number and performance of high-speed optical modules. The optical module, as the key to data centre transmission, provides a guarantee for stable and reliable data transmission in the data centre. There are a large number of optical modules in the data centre, and the failure of optical modules can have a significant impact on the operation of the data centre. Traditional operation and maintenance methods are no longer sufficient to meet the management needs of data centres for massive optical modules, resulting in low efficiency and difficulty in troubleshooting. Therefore, it is necessary to carry out refined management of data centre optical modules.

### 5.2.6.2 Description of the use case

#### 5.2.6.2.1 Overview

Operators are laying out a large number of data centres, which require a large amount of data transmission and exchange. Optical modules are the core components for data transmission within and between data centres. The current data centre optical module management includes:

1) In the past, a large number of optical modules existed in traditional transmission rooms. When there was a fault or alarm in the transmission line, it was necessary to locate whether the optical module was abnormal. With the establishment of a large number of data centres, the number and scale of optical modules are large, and traditional management cannot meet the operation and maintenance needs of data centres.
2) One of the common problems in data centres is the abnormal calculation and transmission caused by the failure of optical modules.
3) The traditional management efficiency of optical modules is low, and the means are single, which cannot meet the high-speed transmission needs of data centres.

#### 5.2.6.2.2 Motivation

The ENI System can be used to evaluate and predict the health of optical modules. In ENI, machine learning can be used to judge and predict the status of optical modules, such as health, sub-health, and faults. The judgment is based on the historical working status data of the optical module. More specifically, the ENI System can classify the current working state of optical modules and predict the working state based on time series through machine learning.

This can be used to better manage data centre optical modules, avoid data transmission performance degradation caused by optical module degradation, and better solve the loss of data centre caused by optical module failures. With the assistance of ENI, data centres can operate and maintain optical modules based on their health status, and promptly replace sub-healthy optical modules.

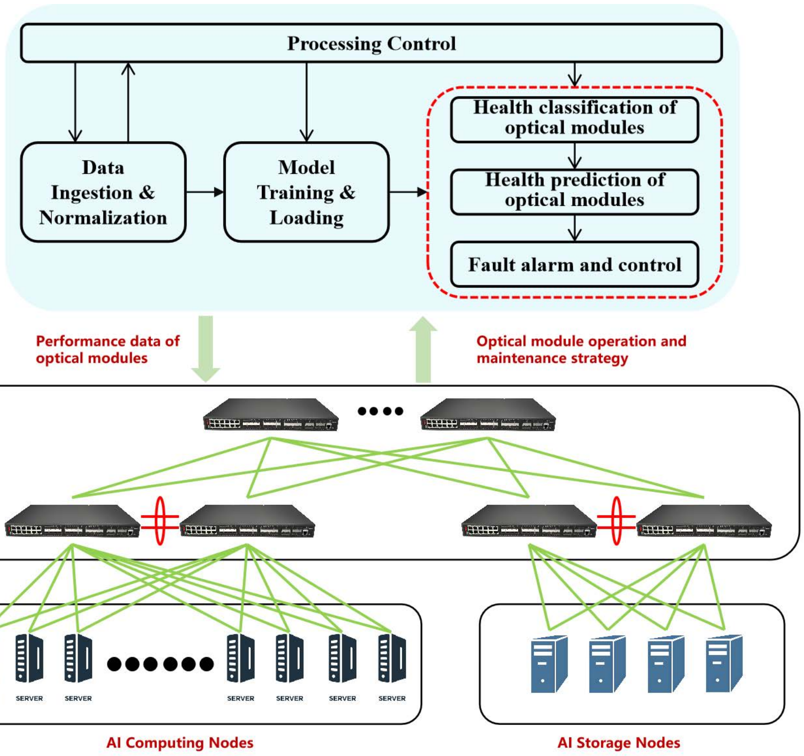


**Figure 5-7: A general architecture of the use case**

Figure 5-8 shows the framework of the use case mapping to the ENI reference architecture.

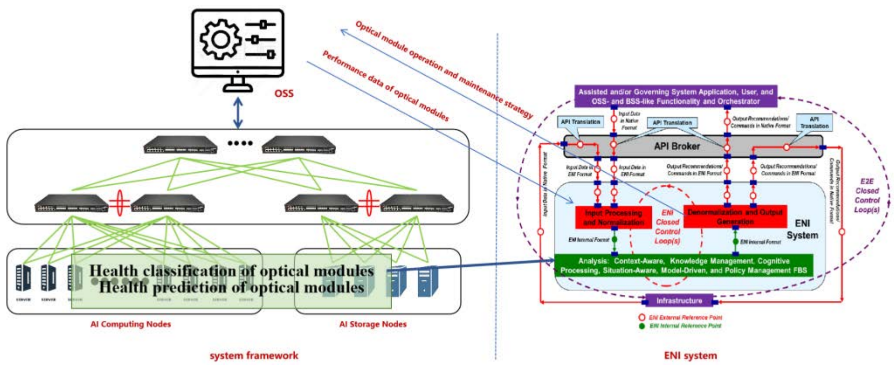


**Figure 5-8: Use case architecture mapped to ENI reference architecture**

### 5.2.6.2.3 Actors and Roles

1) ENI System: Collect the operating parameters of the optical module from the network management or environment, learn the relationship between the working status of the optical module and the performance data of the optical module. Classify the working status of the optical module based on its current performance data, and predict the performance changes of the optical module in the future.

2) Operator: Manage the data centre and confirm the replacement strategy for optical modules.

3) Data Centre network management: Provide the ENI System with the required information (such as bias current, bias voltage, power, temperature, etc.) and report the health status of the optical module.

### 5.2.6.2.4 Initial context configuration

1) The relationship between historical performance data of optical modules and business quality.

2) Classification criteria for the health status of optical modules.

#### 5.2.6.2.5 Trigger conditions

Periodic execution of optical module health prediction operation.

#### 5.2.6.2.6 Operational Flow of the actions

For the classification of the working status of optical modules:

1) The operator defines the health status of the optical module in advance and associates the historical status with the health status of the optical module.
2) The ENI System uses the operator's definition of the health of the optical module to train the model, learning the relationship between the performance data of the optical module and its working state.
3) The network management reports the current performance data of the optical module.
4) The ENI System classifies the health status of optical modules based on the current reported performance.
5) The operator determines the operation and maintenance strategy based on the classification results of ENI.

For predicting the health of optical modules:

1) Train a time series based optical module performance data model for the ENI System.
2) The network management reports the current performance data of the optical module.
3) The ENI System predicts the future performance data of the optical module based on the current reported performance.
4) The operator determines the operation and maintenance strategy based on the prediction results of ENI.

#### 5.2.6.2.7 Post-conditions

Network management can obtain historical performance data of optical modules.

## 5.2.7 Use Case #1-7: LLM for Network OAM Application on Generic Computing Platform

### 5.2.7.1 Use case context

This use case intends to describe network OAM LLM application running on generic computing platform instead of a GPU platform, with special attention to the lower cost and power consumption aspects, in the context defined by ENI. This use case solves the adaptation of generic computing platform to replace part or all a GPU platform for a reduced power and cost consumption.

### 5.2.7.2 Description of the use case

#### 5.2.7.2.1 Overview

As demand for GPU increase dramatically due to the rise of AIGC applications, service providers having difficulties to provide sufficient computing power for AIGC related applications. The CPU platform on the other hand, accumulates significant computing power by stable service provider investment. The overall CPU workload for service provider often very low, often less than 50 %. To solve the problem of computing power and demand, it is possible to employ AIGC applications on CPU resources, realizes the balance between the CPU workload and lowering the power and budget cost:

#### 5.2.7.2.2 Motivation

Traditionally LLM related application is GPU friendly, including learning and processing tasks on computers with CPU server. The processors are widely used in personal computers, servers, and embedded systems.

An LLM application on this architecture leverages the capabilities of CPU processors to perform complex Natural Language Processing (NLP) tasks, including but not limited to language translation, sentiment analysis, text summarization, and language generation. These applications utilize advanced machine learning models, particularly those in the realm of deep learning, to understand, interpret, and generate human language in a way that is meaningful and contextually relevant.

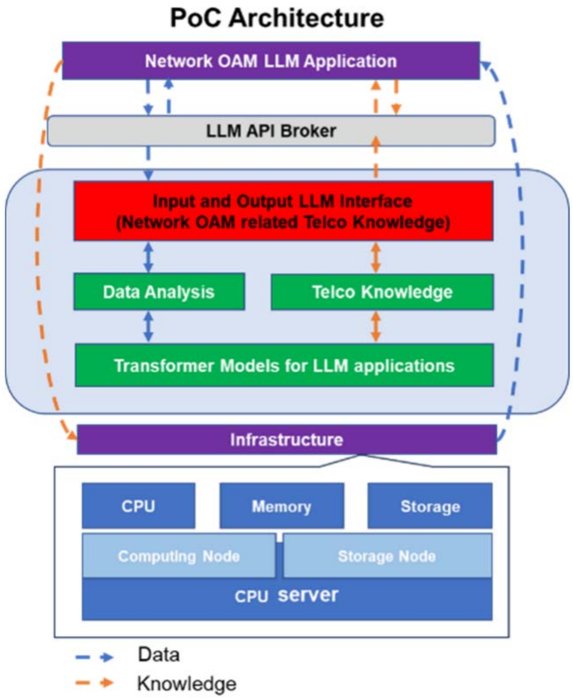


**Figure 5-9: Architecture of the use case**

#### 5.2.7.2.3 Actors and Roles

1) Language Models: At the core of an LLM application are the language models themselves. These models are trained on vast amounts of text data to learn the statistical properties of languages. They can predict the likelihood of a sequence of words or generate new text based on a given prompt.

2) Processing Unit: Given the intensive computational demands of LLMs, especially those involving deep learning, the CPUs, and optionally Graphics Processing Units (GPUs), are utilized to process the data efficiently. Modern CPU processors are capable of parallel processing, significantly speeding up the computations required for training and inference.

3) Software Frameworks: The application relies on software frameworks and libraries optimized for machine learning, such as TensorFlow, PyTorch, or ONNX. These frameworks provide the tools and functions necessary for model development, training, and deployment, taking full advantage of the CPU architecture for performance optimization.

4) Data Preprocessing and Management: Effective language learning requires clean and structured data. The application includes components for data preprocessing, such as tokenization, stemming, and lemmatization, as well as data management tools for handling large datasets.

5) User Interface (UI): An intuitive UI is crucial for facilitating interaction between the user and the LLM application. This could range from simple command-line interfaces to sophisticated Graphical User Interfaces (GUIs), depending on the application's complexity and target audience.

6) Security and Privacy: Given the sensitive nature of language data, CPU-based LLM applications incorporate security features to protect user data and ensure privacy. This includes data encryption, secure data storage, and compliance with data protection regulations.

##### 5.2.7.2.4 Initial context configuration

1) Data processing of the LLM, the infrastructure layer contains two types of server structure: the high performance and storage. The high-performance server will perform the inference LLM tasks, and the storage server perform less computing exhausting tasks e.g. backup, data exchange.

2) The second stage is model pre-training, which maps to cognition framework, knowledge management and context-aware management FBs of the ENI System. In this stage, training datasets are fed into LLM algorithms (e.g. transformer based) to generate the models used to provide Telco OAM applications.

##### 5.2.7.2.5 Trigger conditions

When the LLM related application complete the fine-tuning process.

##### 5.2.7.2.6 Operational Flow of the actions

1) Model tuning start, the CPU servers can be scaled to meet the demands of the tuning process.

2) Adding more servers to form a cluster or enhancing existing servers with more powerful CPUs.

3) Scalability ensures that the infrastructure can keep up with the increasing computational demands of model tuning, particularly as models grow in size and complexity.

4) Model inferencing. the CPU servers are used in inferencing tasks, the task of request handling from upper system layers can be performed using CPU server and software.

##### 5.2.7.2.7 Post-conditions

The results from inference can be used for external request.

## 5.3 Network Operations

### 5.3.1 Use Case #2-1: Policy-driven IP managed networks

#### 5.3.1.1 Use case context

There are some types of network nodes that need to allocate IP addresses to end users. Examples include Broadband Remote Access Server (BRAS), Dynamic Host Configuration Protocol (DHCP) server, and Carrier Grade Network Address Translation (CGN). Each of these network nodes needs to be configured with IP addresses (i.e. from an IP address pool), which they can use to allocate to the end users. Currently, the plan and configuration of IP address pools rely on manual configuration that is fundamentally static in nature.

#### 5.3.1.2 Description of the use case

##### 5.3.1.2.1 Overview

In a common scenario of Home Access, the client sends an access request to a BRAS. The BRAS picks one IP address from its pre-configured IP address pool and allocates that IP address to this client; this enables this client to access the network using this IP address. CGN translates private address into public address. CGNs are configured with several public IP address pools. When there is a need for a CGN to translate the private IP address of one session from the client side to a public IP address for network side, the CGN picks one public IP address in its pre-configured public IP address pools, replaces the private IP address by the selected public IP address, and records this mapping.

#### 5.3.1.2.2 Motivation

The traditional IP management approach suffers from low utilization of IP addresses and poor sharing among equipment. Manual address allocation is cumbersome, and scripts are fragile and cannot adjust to dynamic network conditions. There are several disadvantages:

- Currently certain operators do not have sufficient IP address resources, especially for IPv4.
- IP address resource utilization ratio is low in general: some network nodes have low utilization ratio of internal addresses; some devices suffer from tidal effect (i.e. high in peak period and low in idle period).
- Address resources are not shared among equipment, which leads to inefficiencies in deployment.

In this use case, the ENI System learns the pattern of user sessions, which consumes the IP addresses, and classifies the users accordingly. The ENI System generates IP address pool configuration policies and IP address allocation policies to improve the efficiency of the utilization of IP addresses.

Policy enables more intelligent usage of address pools and automates the address allocation. With policy-driven network resource optimization and network resource monitoring, it is possible to automatically adjust address allocation on different equipment using policies. Such policies may consider factors such as demand on address, utilization ratio, address usage lifecycle, and constraints (e.g. the rejection rate of a BRAS or CGN, or thresholds that apply to address utilization). This allows more intelligent usage of address pools and automates the address allocation process, where improved operator experience can be expected. It also ensures more consistent operation of address allocation, which also improves the operator experience.

Such a use case is illustrated in Figure 5-10.

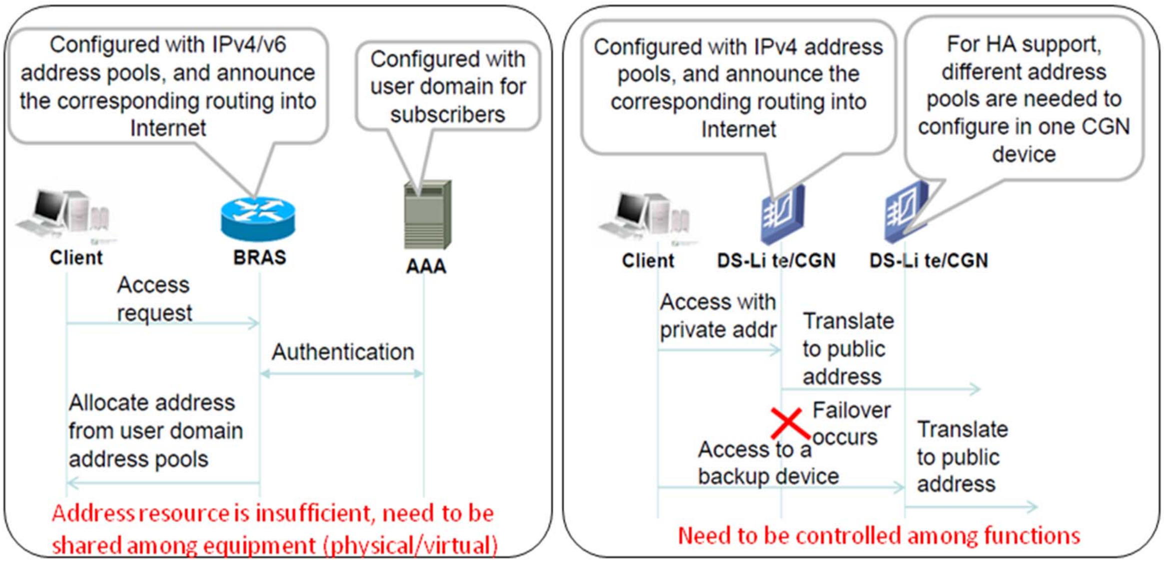


**Figure 5-10: Current Problems in Operator IP Managed Networks**

#### 5.3.1.2.3 Actors and Roles

- ENI System with the IP address allocation algorithm system and data collection system.
- Network Functions which need IP address pool configuration (e.g. vBRAS).
- Network administrator.

Stakeholders managing the above:

- Operators.

#### 5.3.1.2.4 Initial context configuration

- The network administrator's inputs the policies to configure the number and size of IP address blocks.
- One vBRAS is configured with a predefined number of IP address blocks, where each block contains a predefined number of IP addresses.

vBRAS allocates an IP address to the users randomly; current solutions suffer from many IP addresses in each IP address block not being used, with at least one IP address in use.

#### 5.3.1.2.5 Triggering conditions

- Trigger 1 for IP address allocation policy adjustment: when a user's IP address usage does not align with the current allocation policy more than a predefined number of times in one measurement time period, the ENI System will adjust the IP address allocation policy according to the latest information from the user.
- Trigger 2 for IP address allocation policy adjustment: when one or more users change their behaviour, the ENI System will those users based on an appropriate classification or clustering algorithm, and adjust the IP address allocation policy accordingly.

#### 5.3.1.2.6 Operational flow of actions

1) The ENI System collects and stores the information of the users' usage of IP addresses, in a normalized format with user ID, location number, daily IP address usage time, holiday IP address usage, weekdays and weekends IP address usage, etc.
2) The ENI System uses one or more classification or clustering algorithms to build an appropriate model. Users are labelled based on their behaviour characteristics by using an appropriate algorithm, according to their historical and contextual information (e.g. location information, time of attachment and detachment, types of applications used, and amount of data transferred).
3) The ENI System modifies policies to re-configure the number and size of IP address blocks to be allocated to each user group, as well as the IP address allocation mechanisms.
4) IP address blocks and IP address allocation policies are sent to the BRAS for processing:
    a) When a user attaches to the BRAS, the BRAS allocates an IP address to the user's corresponding IP address block, according to the IP address allocation policy and the user information including his/her equipment identifier.
    b) When the current usage of one IP address block reaches a threshold, the BRAS will select another IP address block with the same characteristics for further IP address allocation to the same type of users.
    c) If all IP addresses in an IP address block are not in use, and the IP addresses are not kept for redundancy purposes, this IP address block will be recycled.
5) When triggered, the ENI System will regroup the users and adjust the IP address allocation policy accordingly.
6) When a user attached to the BRAS requesting for an IP address, the BRAS will select a most frequently used IP address block among the ones mapping to the user label, and allocate an IP address in this block to the user.

#### 5.3.1.2.7 Post-conditions

All current users have the minimum number of IP addresses allocated or reserved. IP address pools are optimized.

## 5.3.2 Use Case #2-2: Radio Coverage and capacity optimization

### 5.3.2.1 Use case context

Coverage and Capacity Optimization (CCO) is one of the typical operational tasks of the Radio Access Network (RAN). CCO aims to provide the required capacity in the targeted coverage areas, to minimize the interference and maintain an acceptable quality of service in an autonomous way. To achieve these targets, antenna power and configuration (pilot power, antenna down tilt, antenna azimuth, or massive MIMO pattern in 5G) play a critical role, as they affect the direction of the antenna radiation pattern, therefore can be used to improve the received signal strength in the own cell as well as to reduce the interference to neighbouring cells.

The CCO task also exists in enterprise Wireless Local Area Network (WLAN) scenario. In enterprise WLAN, an Access Point (AP) controller sets multiple APs' RF parameters (e.g. channel frequency, bandwidth, power) to provide full coverage and minimize the inter-cell interference (namely dynamic channel allocation and transmit power control).

### 5.3.2.2 Description of the use case

#### 5.3.2.2.1 Overview

CCO allows the system to periodically adapt to traffic fluctuation (i.e. load and location) and the radio environment by adjusting the key Radio Frequency (RF) parameters (e.g. excitation amplitude and phase of the single element). For the online CCO task, it is not possible to find definite function to map between the RF parameters and the target coverage and capacity performance. The main reason is that the set of configurable RF parameters is multi-dimensional, and each RF parameter has wide range of values, leading to very large number of possible options.

#### 5.3.2.2.2 Motivation

Performing exhaustive search to find optimal RF parameter combination and associated value can be extremely complex. Today's network lacks efficient way of find the optimal combination of RF parameters for the changing network environment. An intelligent entity (e.g. ENI System) can leverage machine learning to analyse and learn what the proper action is for each current network state (e.g. current RF parameters together with geometrical configuration of overall array, relative pattern of the individual element and relative displacement of the elements; or code book (provided as an index into a set of predefined excitation matrices) with corresponding antenna power pattern, User Equipment (UE) location, traffic load, Spectrum allocation, current antenna engineering parameters e.g. direction angle, tilt angle, base station location and pilot power, etc.). Based on the learnt model (which can be continuously optimized), the ENI System can then instruct the Operations System (OS) the base station the proper action to adjust the RF parameters for optimizing coverage and capacity.

In WLAN scenario, the ever-changing radio environment (e.g. external AP interference and non-Wi-Fi-type interference) requires the system to adjust their RF parameters to achieve best performance. Using collected RF parameters, signal strength and throughput data, an intelligent entity (e.g. ENI System) can use machine learning to learn the mapping relationship, and instruct the AP controller to set proper RF parameters for those managed APs to optimize coverage and capacity.

The Use Case is illustrated in Figure 5-11.

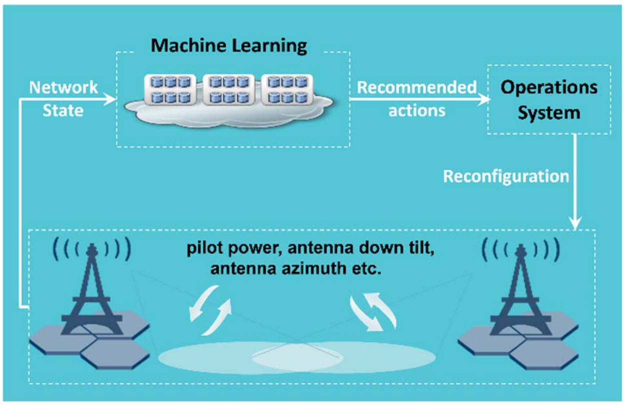


**Figure 5-11: Coverage and Capacity Optimization**

#### 5.3.2.2.3 Actors and Roles

- Operator: defines the target coverage (e.g. RSRP and SINR for users) and capacity performance (e.g. maximize the traffic and Transmission Control Protocol (TCP) load) of managed areas.
- ENI Engine: collects and analyses the state and performance of radio access network, dynamically determines what RF parameters should be configured according to them.
- Operations System: adjusts RF parameters, code book, or pilot power, according to the policies generated by ENI System.

#### 5.3.2.2.4 Initial context configuration

- The configurations of RF parameters are fixed.
- The ENI System is learning how to configure the RF parameters in order to achieve the target coverage and capacity in certain network state through its machine learning capacities.

#### 5.3.2.2.5 Triggering conditions

Current RF parameters configurations do not meet the target coverage and capacity performance.

#### 5.3.2.2.6 Operational flow of actions

1) Operator pre-configures the target coverage and capacity performance.
2) ENI System collects and analyses the radio environment information to be aware of the state and performance of current network.
3) ENI System determines the RF parameters configuration according to the current network state and target coverage and capacity performance.
4) Operations system reconfigures the RF parameters according to the output of ENI System.

#### 5.3.2.2.7 Post-conditions

- The RF parameters, code book, or pilot power dynamically adjust according to the changing radio environment.
- The target coverage and capacity performance is met.

### 5.3.2.3 Mapping to ENI reference architecture

#### 5.3.2.3.1 Functional blocks

The mapping to ENI architecture for radio coverage and optimization using AI is shown in Figure 5-12.

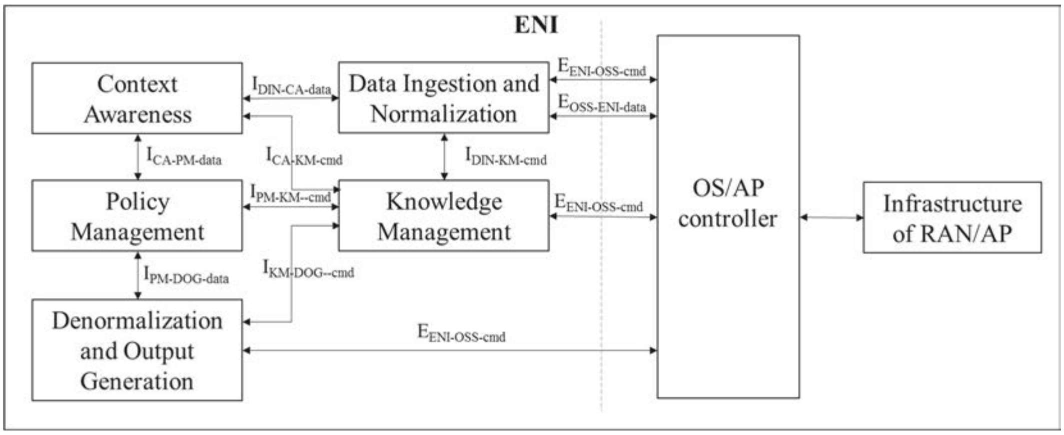


**Figure 5-12: Mapping to ENI reference architecture**

The knowledge management functional block holds the goals of the operator, the OS expert knowledge such as the rules of RF reconfiguration, the source data which should be collected for the CCO, the method of data process, the details of models that used for sensing the real-time state of RAN, the rules of model optimization and updating, and the policy learned by the ENI System.

The data ingestion and normalization function block transfers the raw data provided by the OS/AP controller into a form that can be understood by the ENI System.

The context awareness functional block analyses the real-time coverage and capacity performance of RAN by using AI models, meanwhile, it retrains and updates the model periodically.

The policy management functional block determines the actions should be taken, such as increase the down tilt angle by one degree. Based on the performance after the action taken by OS/AP controller, the block determines the next action should be taken. If the performance got worse, the block determines to roll back the configuration, and if the state got better, the block determines to future adjust the configuration. Finally, the coverage and capacity performance met the goals, and the block learned the optimal action should be taken under certain state.

The denormalization and output generation functional block converts recommendations generated by the ENI System, to a form that the OS/AP controller can understand.

### 5.3.2.3.2 Interfaces

$E_{OSS\text{-}ENI\text{-}data}$ defines data exchanged between ENI and the OS/AP controller. The OS/AP controller collects data from RAN infrastructure or AP and send it to ENI System. The data includes history traffic statistics, measurement report and configuration data, etc.

$E_{OSS\text{-}ENI\text{-}cmd}$ defines recommendations and/or commands and acknowledgements exchanged between ENI and the OS/AP controller. The commands include the down tilt angle parameter should be configured provided by ENI System, the request for RAN/AP infrastructure data sent from ENI System and the expert knowledge provided by OS/AP controller.

$I_{DIN\text{-}KM\text{-}cmd}$ defined recommendations and/or commands and acknowledgements exchanged between knowledge management functional block and data ingestion and normalization functional block. The recommendations include the data need to collect and the method of data ingestion and normalization provided by knowledge management functional block, which includes feature extraction and feature engineering, etc.

$I_{CA\text{-}KM\text{-}cmd}$ defined recommendations and/or commands and acknowledgements exchanged between knowledge management functional block and context awareness functional block. The commands include the model information used for context awareness, and the rules for model retraining and updating provides by knowledge management functional block, and the updated model information provided by context awareness functional block.

$I_{PM\text{-}KM\text{-}cmd}$ defined recommendations and/or commands and acknowledgements exchanged between knowledge management functional block and policy management functional block. The commands include the RF configure rules provided by OS/AP controller and the new rules learned by ENI System.

$I_{KM\text{-}DOG\text{-}cmd}$ defined recommendations and/or commands and acknowledgements exchanged between knowledge management functional block and denormalization and output generation functional block. The commands include methods of transfer the command generated by ENI to the form that OS/AP controller can understand.

$I_{DIN\text{-}CA\text{-}data}$ defines data exchanged between data ingestion and normalization functional block and context awareness functional block.

$I_{CA\text{-}PM\text{-}data}$ defines data exchanged between context awareness functional block and policy management functional block, which includes the real-time coverage and capacity performance.

$I_{PM\text{-}DOG\text{-}data}$ defines data exchanged between policy management functional block and denormalization and output generation functional block, which include the recommend action generated by policy management.

### 5.3.2.3.3 Flow of information

The flow of information for this use case is given in Figure 5-13.

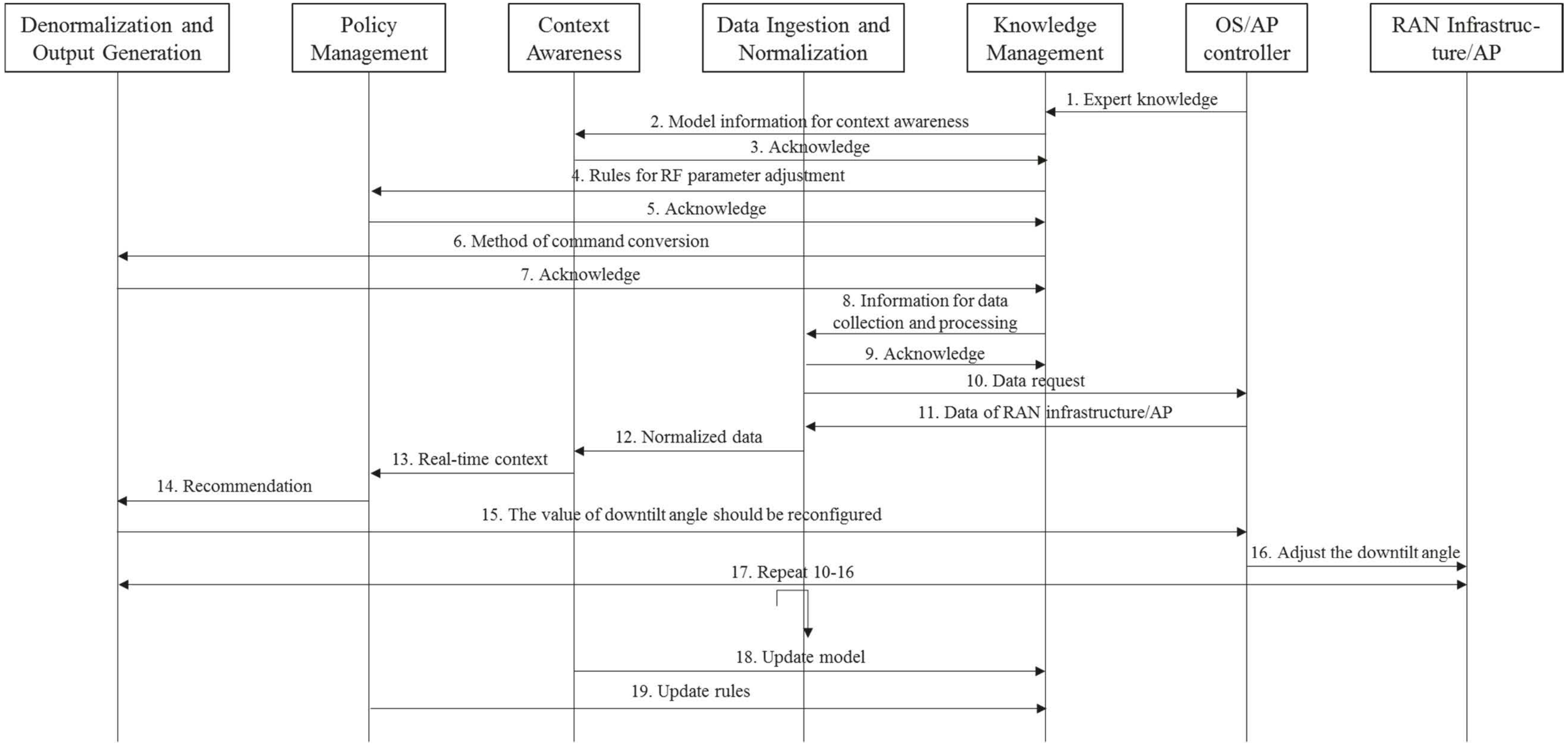


**Figure 5-13: Procedure for coverage and capacity optimization**

Step 1: OS/AP controller transfers expert knowledge to knowledge management functional block.

Step 2: Knowledge management functional block transfers model information for context awareness to context awareness functional block.

Step 3: Context awareness functional block transfers acknowledge to knowledge management functional block.

Step 4: Knowledge management functional block transfers rules for RF parameter adjustment to policy management functional block.

Step 5: Policy management functional block transfers acknowledge to knowledge management functional block.

Step 6: Knowledge management functional block transfers method of command conversion to denormalization and output generation functional block.

Step 7: Denormalization and output generation functional block transfers acknowledge to knowledge management functional block.

Step 8: Knowledge management functional block transfers information for data collection and processing to data ingestion and normalization functional block including what data needs to be collected and the method for data ingestion and normalization.

Step 9: Data ingestion and normalization functional block transfers acknowledge to knowledge management functional block.

Step 10: Data ingestion and normalization functional block transfers request for RAN infrastructure/AP data to OS/AP controller.

Step 11: OS/AP controller transfers the collected data based on the requirement of step10 to data ingestion and normalization functional block, such as history traffic statistics, measurement report and configuration data.

Step 12: Data ingestion and normalization functional block processes the raw data and then transfers the processed data to context-awareness functional block.

Step 13: Context-awareness functional block gets the real-time coverage and capacity performance to policy management functional block.

Step 14: Policy management functional block determines how to adjust the down tilt angle according to the state of network, and then transfers the recommendation to denormalization and output generation functional block.

Step 15: Denormalization and output generation functional block gets the command that OS/AP controller can understand, and then transfer it to OS/AP controller.

Step 16: OS/AP controller adjusts the configuration of RAN infrastructure/AP.

Step 17: ENI System continue monitors the mobile network performance and adjusts the RF parameters.

Step 18: Context-awareness functional block retraining and updating model, and then replace the model saved in knowledge management functional block.

Step 19: Policy management functional block learned the optimal action should be taken under certain state, and then transfers the knowledge to knowledge management functional block.

### 5.3.3 Use Case #2-3: Intelligent Software Rollouts

#### 5.3.3.1 Use Case context

Physical resources such as routers, during their lifetime, need to have their firmware updated, not only for the support of new services or functionalities, but also to fix existent impairments. In some cases a firmware rollout can take several months to plan and enforce.

Indeed, updating a physical resource firmware constitutes a particularly delicate use case since it involves service disruption, potential bugs on the new version or in the worst case scenario the need to use workforce for equipment replacement. Thus, operators are very cautious when they need to perform a firmware rollout for a given resource, usually by dividing the complete process in different phases, either by geographical locations or different classes of clients.

With the arrival of new paradigms such as NFV or Multi-access Edge Computing (MEC) into the marketplace, this problem can become even worst as more (virtual) software-based resources are being dealt with and there is less time between releases.

### 5.3.3.2 Description of the use case

#### 5.3.3.2.1 Overview

As just stated above, this rollout use case may become even worst when dealing with (virtual) software-based resources, in particular if dynamic on boarding of VNFs or of other type of applications is supported, in which case automatized and intelligent software rollout becomes vital for operators. With dynamic on boarding, common in DevOps environments, automatic tests to benchmark and building of a profile for a given application is possible and recommended. The subject of performing tests to benchmark network functions is very relevant for network operators and is a common procedure with their physical counterparts.

The flow of actions for both physical and Virtualised is similar and should take into account the best practises from Cloud Computing and DevOps. However, since the rollout of Virtualised equipment is considered to be more challenging due to the fact that the number of updates for software-based components is performed much more times, this type of update will be the only one considered in the present use case.

The current Use Case is further described by the following set of components and features.

#### 5.3.3.2.2 Motivation

By making use of the ENI System, operators can define different policies for different types of rollouts and for different types of resources. One example could be the definition of a hierarchy of parameters for phasing out the rollout, e.g. client class, geographical location, or time of the day. In addition, and also taking dynamic on boarding and DevOps environments into consideration, different types of policies can be defined by using the ENI System, such as:

- Development, e.g. tests should provide a correlation between network function performance (throughput, jitter, delay) and resource utilization (CPU, RAM, I/O).
- Update schedule, e.g. for enterprise customers schedule updates outside business hours.
- Update procedures, e.g. create backup of current versions of software instances when updating instances from platinum level services in order to prevent service disruption in case of occurrence of significant errors.
- Failure procedures, e.g. considering two types of errors where the response would be defined by policies:
  - i) critical errors, which make the ENI System stop the update movement process, and initiate the rollback to an already updated instance; and
  - ii) minor errors, which makes the ENI System retry the update.

NOTE: In this use case, only type ii) errors will be considered.

Thus, the use of AI methods becomes more important when moving software from testing to production by using automatized procedures.

#### 5.3.3.2.3 Actors and Roles

The presence of the following actors/entities as well as their associated roles are envisaged in the current Use Case:

- Customers/clients: the operators themselves.
- Network Administrator: entity/person responsible for the initial policy design that encompasses the definition of different activities during rollouts.

- Network Infrastructure: infrastructure that includes resources that are meant to be upgraded or, in the worst case, replaced.
- ENI System: system solution that makes use of AI methods when upgrading or moving software from testing to production, and that enables the use of policies to govern updates to software instances. This solution also participates in software tests and builds a profile with information, e.g. correlation between network function performance (throughput, jitter, delay) and resource utilization (CPU, RAM, I/O), that can be used to improve fulfilment and assurance procedures.
- OSS/BSS: operational and business systems that belong to the management system of network operators. In this case they are providing, among others, monitoring, actuation, internal records of very different items that may range from products to resources, as well as other business interfaces dedicated to external entities.

##### 5.3.3.2.4 Initial context configuration

The network is operating in perfect conditions with all its components in good shape. Moreover, the network operator already has a development environment that is specified to mimic the production environment. This development environment is used to run automatized tests in order to validate new software versions and build the respective software profile, where the series of tests are defined by network operator policies. Finally, the move of software from development to a production environment is also conditioned by network operator policies, thus governing the phased deployment of the new version.

##### 5.3.3.2.5 Triggering conditions

A new software version of a virtual component is released by the vendor and is on boarded on a network operator infrastructure. The upload of a new software version to software repository triggers the start of automatic tests pre-defined by policies also previously enforced in the ENI System.

##### 5.3.3.2.6 Operational flow of actions

The following sequence of actions may be identified:

1) A new software version is instantiated on the network operator development environment.
2) Within the new environment, the software is subject to a series of tests determined by pre-defined policies in the ENI System, which results, will be used to create a software profile.
3) During the tests, the ENI System starts analysing the behaviour of all the collected data and compares it with the profile of previous versions of software.
4) Since the results of the tests are conformant to previous versions, the ENI System is in position to allow the triggers for moving the new version from the development to the production environment.
5) The ENI System takes into account the pre-defined operator policies for the new software rollout and performs the scheduling of updates for the software instances.
6) The ENI System triggers the movement of the new version from the development to the production environment.
7) During the update, all platinum SLA customers of software instances are using a redundant software instance to avoid any service disruption.
8) Some instances monitoring data may detect an inconsistency with the application profile indicating a problem with the update. Since it is considered a minor error, the ENI System retries the update on failed instances.
9) At the end of the process, the ENI System may notify relevant software components, e.g. OSS/BSS, that the software rollout has been carried out successfully.

##### 5.3.3.2.7 Post-conditions

The new software version has been updated on all deployed instances and inventories. The network and corresponding services are running steady.

## 5.3.4 Use Case #2-4: Intelligent Fronthaul Management and Orchestration

### 5.3.4.1 Use Case context

Centralized Radio Access Network (C-RAN) has been extensively considered for emerging and future cellular networks. In C-RAN, centralized RAN functions are located in an entity termed as Centralized Baseband Unit (BBU), and the remainder of radio access connectivity between the UE and the network are handled by Remote Radio Units (RRUs). Such an architecture enables functional split between BBU and RRUs. For example, RF and some Physical layer (PHY) level functionalities can be handled at RRUs, while the rest can be moved to centralized BBUs.

Such functional split versus classical Distributed RAN (D-RAN) brings several advantages including accelerated network deployment on RRU side, reduced operating costs (although Capital Expenditure can be high in short term), support for richer multi-node network cooperation and coordination (e.g. on Coordinated Multi-Point systems or Carrier Aggregation) and improved network performance, in particular at the cell edge.

To support such functional split, a Common Public Radio Interface (CPRI) has been proposed to support Fronthaul connectivity, which is the connection between BBU and RRUs. However, CPRI is believed to require strict high bandwidth, low delay, tight synchronization and additional transmission equipment partly attributed to its Point-to-Point connectivity paradigm.

To address the above issues, Next Generation Fronthaul Interface (NGFI) including envisioned new variants of CPRI (eCPRI) target redefining interface flexibility and network functional split between remote and centralized units. Such an interface enables statistical multiplexing on Fronthaul bandwidth, decoupling interface traffic from some RF-level attributes (e.g. number of antennas) and results in more flexible remote unit connectivity to a centralized unit.

In line with recent advancement on NGFI, RRUs are divided into clusters; each cluster may possess one logical entity termed as Remote Aggregation Unit (RAU) that can be physically located as part of one of RRUs per cluster or as a separate individual entity. The RAU is in charge of radio resource management per cluster.

As the functional split can dynamically switch between remote RAU and the centralized entity, the new centralized entity is redefined as Radio Cloud Centre (RCC) to convey multitude of functionalities beyond conventional BBU. RAU and RCC may also refer to the general remote and central entities in any of the next generation Fronthaul technologies.

### 5.3.4.2 Description of the use case

#### 5.3.4.2.1 Overview

Flexible NGFI opens a new network design paradigm where nodes connectivity between centralized and remote units transforms from Point-to-Point or Point-to-Multi Point into Many-to-Many connectivity comprising hybrid of wired and wireless solutions. In other words, a multi-tier shared network forms the Fronthaul where the slicing of network resources between centralized and remote units can be dynamically tuned in an on-demand fashion.

The new use case is concerned with applying AI technologies, and the resulting interfaces and network components, at the next generation flexible Fronthaul, to facilitate the flexible and dynamic slicing of network resources at the Fronthaul.

#### 5.3.4.2.2 Motivation

Slicing of network resource (especially in a dynamic and flexible manner) at the Fronthaul between remote and centralized units can be complex, as it is affected by multiple factors and their changing contexts - factors such as, the clustering on the remote units (the size, how are they clustered, etc.), the functional split between remote and centralized entities and dimensionality of solution space on network resources to be reserved on the Fronthaul (power, processing capability, radio resources, buffering memory, route to be selected across multiple Fronthaul nodes, etc.).

The application of AI under such context will bring efficient optimization framework, balancing the multiple aspects considered on network resources slicing mentioned above. It will also enable flexible and dynamic resource slicing and functional split at the Fronthaul, considering the changing contexts of the network, such as the changing traffic demand, at the RAU and RCC. As an example of such an application, through load estimation and prediction using the state-of-the-art AI algorithms, the Fronthaul management and orchestration can also be designed in an 'on-demand' manner.

Figure 5-14 shows the concept of the proposed Fronthaul use case.

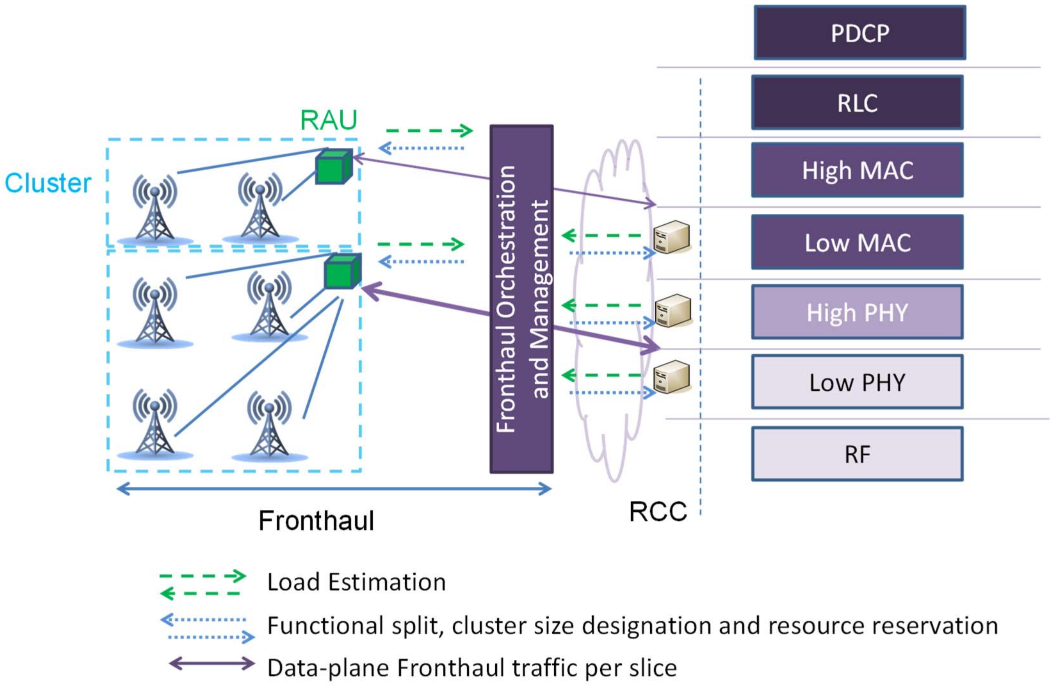


**Figure 5-14: Concept of the proposed Fronthaul use case**

#### 5.3.4.2.3 Actors and Roles

- Operator: Provides interfaces to convey signalling on load estimation, service level agreements, slice-level requirements and traffic profiles (if any) to the ENI System.
- ENI System: Collects load estimation data (traffic demand, current configuration) from different RAU/RCC units; determines optimal Fronthaul parameters (e.g. functional split, clustering size per RAU); and reserves Fronthaul network resources accordingly between different RAU and RCC units.
- RAU/RCC units: Provide relevant input data (e.g. on load estimation) via Operator's interfaces to the ENI System and readjust their Fronthaul parameters (e.g. functional split, clustering size per RAU) according to output data from ENI System.

#### 5.3.4.2.4 Initial context configuration

Network is configured with a default set of parameters on, e.g. a target Fronthaul KPI, the functional split, the cluster size, and the corresponding reservation of Fronthaul resources. The default parameters can be set by the network Operator.

#### 5.3.4.2.5 Triggering conditions

When the Fronthaul KPI is below the target.

#### 5.3.4.2.6 Operational flow of actions

1) ENI System collects Fronthaul parameters, current configurations and past and current traffic from different RAU/RCC Units.
2) ENI System makes decision on the Fronthaul parameters and feedbacks the parameters to the RAU/RCC units. Such decision can be made based on experiential learning of the ENI System given the current context.
3) RAU/RCC units receive and readjust their respective Fronthaul parameters.
4) ENI System reserves Fronthaul network resources. Fronthaul operation KPI is fed back to ENI System.
5) When triggered, the ENI System will reconfigure the parameters and reallocate the resources.

#### 5.3.4.2.7 Post-conditions

Target Fronthaul KPI is guaranteed.

## 5.3.5 Use Case #2-5: Elastic Resource Management and Orchestration

### 5.3.5.1 Use case context

Vertical markets and industries are addressing a large diversity of heterogeneous services, use cases, and applications in 5G cellular networks. It is currently common understanding that for networks to be able to satisfy those needs, a flexible, adaptable, and programmable architecture based on network slicing is required. Moreover, a softwarization and cloudification of the communications networks is already happening, where Network Functions (NFs) are transformed from monolithic pieces of equipment to programs running over a shared pool of computational and communication resources. However, this novel architecture paradigm requires new solutions to exploit its inherent flexibility.

This use case addresses mechanisms to exploit the abovementioned flexibility and the concept of resource elasticity, which is a key means to provide an efficient management and orchestration of the computational resources of Virtualised and cloudified networks. Elasticity can thus be understood as the ability to gracefully adapt to load changes in an automatic manner such that at each point in time the available resources match the demand as closely and efficiently as possible. These automation mechanisms can greatly benefit from the employment of AI techniques in general and the integration of an ENI System in particular, which would allow optimized decisions to be made based on real data.

### 5.3.5.2 Description of the use case

#### 5.3.5.2.1 Overview

An elastic management and orchestration of resources can be achieved in different ways. Three different set of elasticity mechanisms can be differentiated, each of them addressing a specific challenge in the overall use case context:

- The computational aspects of network functions have not been taken into account in their original design, hence computationally elastic VNFs can be redesigned to account for those in their operation.
- Flexible mechanisms for orchestration and placement of NFs across central and edge clouds should be designed, considering source and destination hosts resources, migration costs and services' requirements. In particular, latency requirement is a key driver for placement of VNFs at the edge.
- Slicing multiplexing gains due to the sharing of the infrastructure and physical resources need to be fully exploited. Moreover, an efficient network management has to capitalize on the possibility of sharing and re-using the same virtual resources for network slices with similar or identical requirements and shared VNFs.

The three above described challenges are the target of the proposed elastic management and orchestration of resources. To that aim, AI and the ENI System may play an important role as a tool to enhance the performance of elasticity algorithms. Prominent examples of performance-boosting capabilities that could be provided by the ENI System are the following:

i) speeding the service deployment process by realizing an AI-based, automatic, accurate, and reliable mapping from service requests to network slice instantiations;

ii) identification of similarities (in terms of requirements or shared VNFs) across slices to facilitate resource sharing, thus increasing the system resource utilization efficiency;

iii) learning and profiling the computational utilization patterns of VNFs, thus relating performance and resource availability;

iv) traffic prediction models for proactive resource allocation and relocation;

v) optimized VNF migration mechanisms for orchestration using multiple resource utilization data (CPU, RAM, storage, bandwidth), and vi) optimized elastic resource provisioning to network slices based on data analytics. In the following, details are provided to the description of this use case.

#### 5.3.5.2.2 Motivation

As previously mentioned, there is a need to design mechanisms that allow the network infrastructure to become flexible enough to host the heterogeneous set of verticals that 5G is meant to address, where the flexibility enabled by this use case is indeed beneficial not only for wireless networks, but also for fixed networks. An elastic resource management and orchestration increases the flexibility of the network by allowing a very efficient utilization of the resources that gracefully adapts its behaviour to the load and the available resources at every time. Furthermore, networks and network slices get currently over-dimensioned in their computational capabilities for cases of peak load. With this use case, a more autonomous and intelligent self-dimensioning of the network is targeted, along with a smart redistribution of the computational resources.

#### 5.3.5.2.3 Actors and Roles

The AI-assisted "elastic" network management and orchestration is enabled by a predisposition to elasticity of the whole network infrastructure that provides end-to-end services through network slicing. However, this predisposition can be achieved with the standard 3GPP and ETSI NFV architecture, where management and orchestration functionalities of several architectural elements would be enhanced with elastic capabilities. In particular, the following architectural elements and elements play an active role in the current Use Case:

- Management and Orchestration System: it is composed of the functions from different network, technology, and administration domains (such as 3GPP public mobile network management, ETSI ISG NFV Orchestration) that manage network slices and related communications services across multiple management and orchestration domains in a seamless manner.

- Network Slice Management Function (NSMF): it is part of the Management and Orchestration System, e.g. 3GPP public mobile network management ETSI TS 128 530 [i.4], or it is an external entity in systems compliant with ETSI GR NFV-EVE 012 [i.5]). A possible architectural design, compliant with 3GPP and ETSI ISG NFV standards and including ENI engine, is depicted in Figure 5-15. NSMF would use AI to extend the 3GPP NSMF/NSSMF functionalities, in order to support the elastic intra-slice (or cross-domain) orchestration and the elastic cross-slice orchestration. The former deals with the orchestration of the different VNFs part of the same slice across multiple domains, while the latter addresses the joint orchestration of the multiple slices deployed on a common architecture. The NSMF also includes functions related to performance monitoring, measurement, and alarm. It is also in charge of defining and instantiating elastic slices, creating first the slice blueprint based on the service-related resource requirements and then defining the appropriated Network Slice Instance.

- Elastic Slice: a set of VNFs and the associated resources to support a mobile service with elastic (non-stringent) requirements that admit graceful performance degradation. This allows e.g. more flexibility in the allocation of resources and in the deployment of the associated VNFs.

- Elastic VNFs: they can be (re-)designed with elastic principles in mind such that the computational resources available for its execution are taken into account, or its temporal and/or spatial interdependencies with other VNFs are mitigated.

- ENI System: system solution that provides a set of AI methods (e.g. supervised/unsupervised and reinforcement learning schemes) to the Elastic Network Slice Management Function.

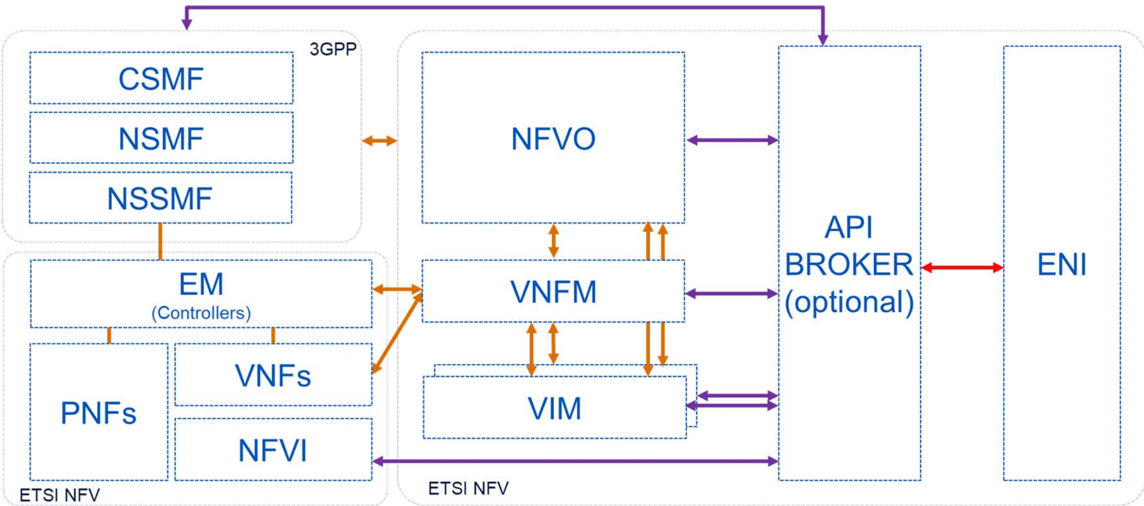


**Figure 5-15: An architectural design including ETSI NFV MANO, 3GPP management functions, and ENI engine**

#### 5.3.5.2.4 Initial context configuration

Consumer-facing service descriptions are mapped to network slice "blueprints". Based on the slice blueprints, a running Network Slice Instance (NSI) is selected or created. Once the NSI deployed, the AI schemes can be used to predict network loads, estimate resource usages, and react accordingly by activating elastic Cross-slice (or Intra-slice) Orchestrator functions in order to optimize the resource usage across slices and prevent system faults.

#### 5.3.5.2.5 Triggering conditions

The ENI System may recommend or enforce the application of one or more algorithms for an elastic (re-)orchestration of resources when at least one of the following events happens:

- A new service request arrives.
- The resource requirements of a new slice cannot be satisfied in the current system configuration.
- The amount of resources allocated to one instantiated slice exceeds a given "efficiency" threshold.
- The requirements of running services change (or is predicted to change) and become substantially more stringent.
- A risk of imminent resource shortage is detected.

#### 5.3.5.2.6 Operational flow of actions

During the slice setup process, the ENI System may be used first to define the slice blueprint; then, based on the slice blueprint, to identify whether it exists one deployed NSI that can support the new service, with a minimum amount of additional resources. Based on this, the resource required are allocated, the slice is instantiated and managed during its lifecycle.

If there are not enough resources available prior to the slice instantiation or an alarm notifies congestions, the ENI System may be used to support the following "elastic" system adaptation functions:

1) Elasticity solutions at the VNF level: VNF computational resource scaling and graceful degradation of performance.

2) Elasticity solutions at the intra-slice level: migration of VNFs to different clouds, to create room for other VNFs with tighter (latency or computational) requirements or enhance the performance of the migrated VNFs.

3) Elasticity solutions at the cross-slice level: cross-slice resource management to maximize resource sharing and optimize the resource utilization efficiency.

The three (families of) elasticity functions mentioned above can be jointly executed and are not mutually exclusive. Nonetheless, in general, they act at different time scales and involve different hierarchical elements of the network architecture (e.g. cross-domain or per-domain).

#### 5.3.5.2.7 Post-conditions

The elastic Network Slice Management Function entails an improvement in the exploitation of the network resources. On the one hand, less resources are employed to guarantee the same QoS. On the other hand, more service requests can be accepted and treated at the same time, improving the network efficiency and reducing redundancy in resource exploitation. Network slicing is re-organized still meeting non-elastic slice requirements.

### 5.3.5.3 Mapping to ENI reference architecture

#### 5.3.5.3.1 Functional blocks

As discussed in ETSI GS ENI 005 [3], the ENI System can enhance the current operator policies for network management with AI. Specifically, this use case relates to the elastic management and orchestration procedures, as described above. The elastic (re-)orchestration algorithms needed to implement the aforementioned operations will reside in the management and orchestration functional blocks of the Assisted System. The ENI System continuously analyses the network performance and, whenever it considers it necessary, may trigger the elastic network management and orchestration mechanisms. The Assisted System receives inputs and instructions from the ENI System that are based on its learning and inferring capabilities. The communication between the elastic Management and Orchestration System of the network and the ENI System happens through the Reference Points specified in ETSI GS ENI 005 [3] and their relevant Interfaces. In the following, the Functional Blocks of the ENI architecture involved in this use case is described.

All the Functional Blocks of the ENI System described in ETSI GS ENI 005 [3] play a role in this use case, according to their defining functions and tasks. The present document of the following blocks is of particular importance:

- Data Ingestion and Normalization Functional Block: it shall collect from all the layers of the network data on performance KPIs, resource availability and state, network slice instances, etc. The more efficiently these data are gathered, the more effective can be the optimization recommended or enforced by the ENI System.
- Context-Aware Management Functional Block: this block is continuously used to update the context in which the ENI System's decisions are made.
- Situation Awareness Functional Block: this block has the goal to estimate and evaluate how the ENI System's decisions impact the assisted network. It should prevent that poor decisions lead to violations of the requirements and SLAs of network slices.
- Cognition Framework Functional Block: this block concretely applies the AI algorithms that lead to an elastic management and orchestration of resources and to the optimization of the network activities. As also proposed in ETSI ISG ENI's PoC#2 "Elastic Network Slice Management", the different kinds of elasticity are enabled by the decisions taken by the Cognition Framework Functional Block, at different levels. More in detail:
    - **Intelligent admission control:** when a new service request arrives, the Operator Management Systems (OSS and BSS) should check the feasibility of serving an additional network slice in the system according to very different viewpoints, e.g.: the kind and amount of resources available in the system, the kind and amount of already provisioned slice in the system or the forecast of future load. After this evaluation, the intelligent admission control system shall take a decision on whether accept the slice (and thus re-orchestrate the network accordingly) or reject (and keep resources free). This approach fits very well with the role of the Cognition Framework Functional Block, which shall act to maximize (resp. minimize) the provider's revenues (in respective to network's health). That is, in a scenario in which network slices bring different revenues to the infrastructure provider, the system may learn to reject requests that are economically not efficient. On the other hand, if the decision has to be taken on a purely infrastructure occupation basis, the effort needed to re-orchestrate the network shall be taken into account. Thus, slices that may require a substantial re-orchestration of the network will be rejected.

- **Elastic re-orchestration:** once slices are admitted in the system, they have to be served according to the required set of SLAs between the tenants and the operator. As one of the fundamental tasks of an ENI System is to be very efficient, the elastic re-orchestration function of the Cognition Framework Functional Block shall predict the future load of each network slice to efficiently compose the VNFs in order to minimize the resource utilization while keeping the KPIs stable. This is a forecasting problem that builds on the feature extraction of the traffic of each network slice. By building on this load forecast (it can be either short- or long-term) the elastic re-orchestration may trigger:

    i) the scaling of a specific VNF (up or down);

    ii) its relocation; or

    iii) the amendment of the Service Function Chain that compose a Network Slice (i.e. adding or removing one network function).

- **Elastic VNFs:** while the aforementioned items relate mostly with the management and orchestration of the system, there is a third dimension of elasticity which directly tackles the design of a VNFs. Analogously to the elastic re-orchestration, a VNF may re-shape its behaviour depending on i) the predicted load and ii) the amount of assigned (computational) resources. This can be applied to any kind of VNF in the system, but advantages will be higher with the most resource-consuming ones, like e.g. RAN- or routing-related VNFs. This functionality is highly intertwined with the elastic orchestration described above as:

    i) elastic orchestration decision shall be taken considering the "elasticity" of the VNFs; and

    ii) elastic operations shall be used only when a wrong resource assignment has been performed, so to efficiently avoid an abrupt interruption of the functionality and rather provide a graceful degradation of the performance until new resources are assigned.

#### 5.3.5.3.2 Interfaces

According to ETSI GS ENI 005 [3], the elastic management and orchestration use case shall employ the following Reference Points and related interfaces:

- The Reference Point towards BSS (Ebss-eni-reg) to lead, perform, or optimize the slice admission control in the system.
- The Reference Point towards OSS (Eoss-eni-dat and Eoss-eni-cmd) to gather information from the OSS about the current status of the system and possibly enforce re-orchestration commands.
- The Reference Point towards the Orchestrator (Eor-eni-cmd and Eor-eni-cfg) to enforce lower-level re-orchestration procedures in the system.
- The Reference Point towards the Infrastructure (Einf-eni-dat and Einf-eni-cmd) to gather information about the current load of the system and perform lower-level configuration on the elastic VNFs.

#### 5.3.5.3.3 Flow of information

The ENI System continuously gathers data and information through the abovementioned interfaces from the different layers of the network. These data are made available to the Context-Aware Management, the Situation-Awareness, and the Cognition Framework Functional Blocks by the Data Ingestion and Normalization and the Knowledge Representation Functional Blocks. The latter collect the information and make it "readable" to the other Functional Blocks of the ENI System. Then, the Cognition Framework Functional Block runs the AI algorithms that enable elasticity management and orchestration, taking as input the data coming from the network, the context analysis provided by the Context-Aware Management Functional Block. The decisions elaborated by the Cognition Framework Functional Block need to be compliant with the inputs or constraints that may come from the Situation Awareness and the Policy Management Functional Blocks. Once validated, those decisions are then translated by the Denormalization and Output Generation Functional Block into instructions for the Network Management and Orchestration System of the assisted network. This data and information exchange is illustrated as an example in Figure 5-16.

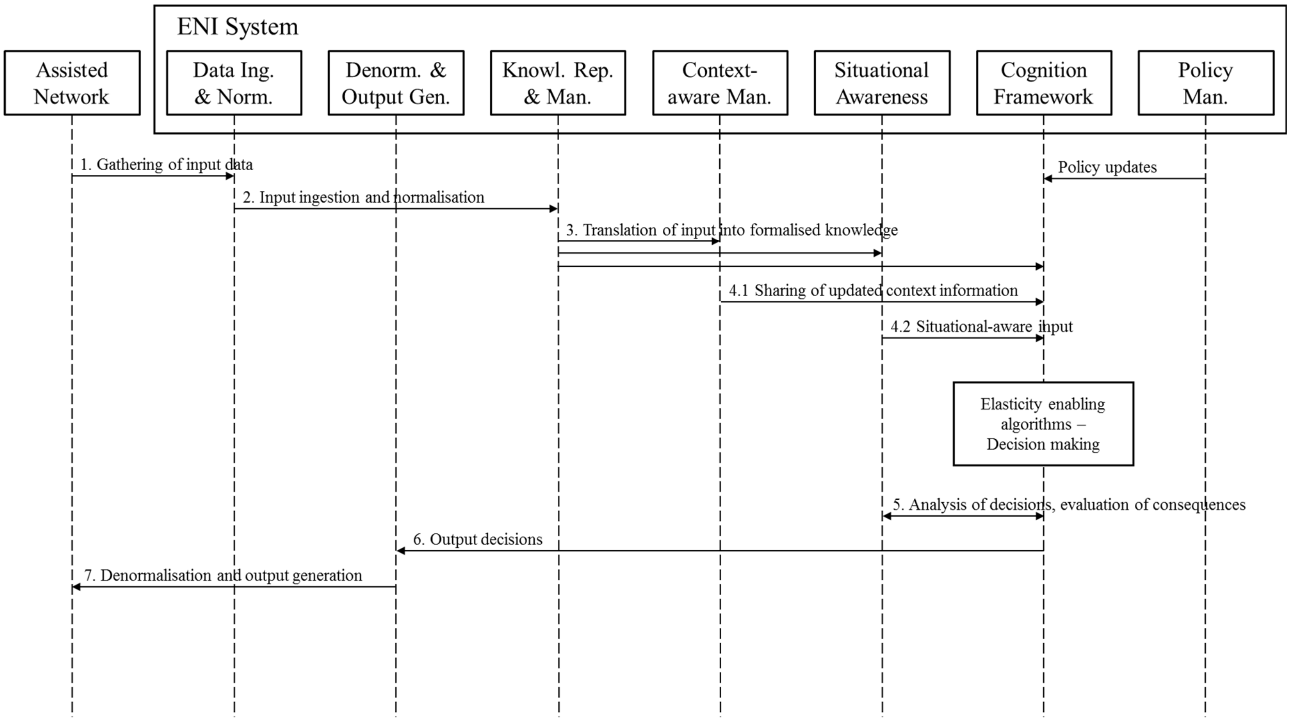


**Figure 5-16: Example Flow of information among the ENI System's functional blocks and the Assisted System**

The flow of information and the exchange of messages between the assisted network and the ENI System shall be as frequent as needed to guarantee the timeliness and effectiveness of the decisions recommended or enforced, but their frequency can vary according to the network load and the speed or significance of the changes in measured or inferred context. The delivery of instructions and recommendations from the ENI System to the assisted Network Management and Orchestration System can happen at different "time granularities" for the different elasticity mechanisms. This can depend on the different complexities of the involved machine learning algorithms that run in the ENI System or on the urgency and "costs" (in term of complexity, time, impacted resources, etc.) related to the implementation of the decisions.

## 5.3.6 Use Case #2-6: Application Characteristic based Network Operation

### 5.3.6.1 Use case context

While Self-Organizing Network (SON) capabilities have advanced over the past decade, the present document centres mainly on self-configuration, self-healing, and self-optimization of network performance. However, a network with good network performance KPI is not always a reliable indicator of the user's perception of the Quality of Experience (QoE) gleaned from using services on that network. For example, non-network related performance measurements can also impact user QoE including service provisioning time, repair times, customer complaint resolution time, brand perception, and more. Moreover, there could be subtle combinations of real-time KPI but that when averaged over a period of time may appear sufficient, but the interim variations can have a strong negative impact on user experience. The real impact has to be measured at an application performance level as this is what the user experiences first hand.

The Use Case proposes a study of how user perception of service quality QoE relates to expectations for application performance, and network performance.

The latter can be challenging to detect since there can be thousands of KPI per second in a real E2E network service flow for a single user, hence the motivation for AI/ML learning techniques that can process massive amounts of performance and management data and detect subtle patterns that can have a large impact on user experience.

### 5.3.6.2 Description of the use case

#### 5.3.6.2.1 Overview

3GPP specifies use cases for SON, but:

- Has not yet any targeted solution to improve wireless performance and further user QoE.
- Has not proposed any feasible proposal for self-planning.
- Has not yet defined any practical algorithms to implement SON.

The Use Case proposes a study of how User perception of service quality QoE relates to expectations for application performance, and network performance.

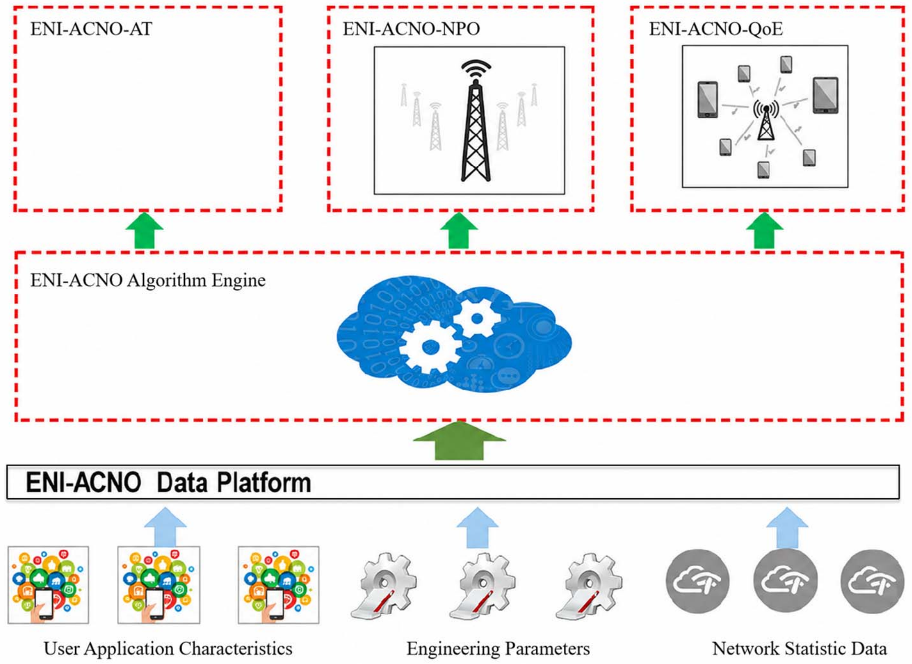


**Figure 5-17: An illustration of the ENI-ACNO framework**

#### 5.3.6.2.2 Motivation

SON is driven by performance optimization rather than ultimately improving user QoE. The impact of application characteristics on network performance and further on QoE are not also considered in 3GPP SON. The present document intends to fill that critical gap by developing and implementing an ENI-ACNO system (illustrated in Figure 5-17), that will optimize network performance and QoE.

#### 5.3.6.2.3 Actors and Roles

- Mobile Network Operator Clients for the ENI-ACNO system.
- Network Infrastructure:
    - 3G/4G/5G Wireless communication networks.
- ENI-ACNO System Framework:
    - ENI-ACNO Data Platform collects network performance statistics and DPI statistics.
    - ENI-ACNO Algorithm Engine profiles cell traffic and application characteristics through clustering and labelling analytics.

  - ENI-ACNO-AT optimizes capacity and coverage through automatically tuning antenna azimuth and down tilt.
  - ENI-ACNO-NPO optimizes the targeted cell KPIs through automatically tuning cell engineering parameters.
  - ENI-ACNO-QoE improves user QoE through automatically tuning cell engineering parameters.
- Operations System:
  - ENI-ACNO is the system to implement and execute the policies of AT, NPO, and QoE through automatically tuning the corresponding network parameters.
  - ENI-ACNO engine prioritizes the targeted performance indicators to be optimized based upon cell profile.

##### 5.3.6.2.4 Initial context configuration

- The ENI-ACNO system analyses cell traffic patterns and corresponding application QoE characteristics by utilizing clustering algorithms.
- The ENI-ACNO system determines the targeted and prioritized KPIs for optimization based upon the individual cell configuration (profile and label).
- The ENI-ACNO-AT learns how to optimize capacity and coverage through tuning cell engineering parameters, such as, but not limited to cell azimuth and down tilt.
- The ENI-ACNO-NPO learns how to optimize the targeted KPIs through tuning cell engineering parameters, such as received power, etc.
- The ENI-ACNO-QoE learns how to optimize user QoE through tuning cell engineering parameters which impact certain KPIs and also impact user perception of Quality.

##### 5.3.6.2.5 Triggering conditions

- Application performance changes.
- Existing capacity and coverage cannot meet the capacity and coverage threshold.
- Existing network performance cannot meet the performance thresholds.
- Existing user QoE cannot meet the QoE threshold.

##### 5.3.6.2.6 Operational flow of actions

- **Cell Labelling and Clustering:**
  - Customers label each cell cluster according to their application characteristics, and determines the relevance/priority of targeted KPIs for each cell.
- **Cell Application Characteristic Profiling:**
  - ENI-ACNO system collects and analyses the application characteristics information indicating the application usage pattern of each cell in the network.
- **Targeted KPIs Identification and Prioritization:**
  - ENI-ACNO system labels each cell according to its application characteristics and prioritizes the targeted KPI to be optimized.
- **Cause Effect Deriving:**
  - ENI-ACNO-AT system derives the relationship between the engineering parameters and the cell capacity and coverage performance.

- ENI-ACNO-OPN system derives the relationship between the engineering parameters and network performance indicators.
- ENI-ACNO-QoE system derives the relationship between the engineering parameters and user QoE.

- **Engineering Parameter Tuning:**
    - ENI-ACNO-AT adjusts the engineering parameters (such as cell down tilt, azimuth) to optimize the capacity and coverage.
    - ENI-ACNO-OPN adjusts the engineering parameters (such as received power) to optimize the network performance.
    - ENI-ACNO-OPN adjusts the engineering parameters (such as received power) to optimize the user QoE.

#### 5.3.6.2.7 Post-conditions

- The engineering parameters are dynamically tuned according to the behaviour change of application characteristics.
- After the engineering parameters are tuned:
    - Capacity and Coverage will be optimized.
    - Targeted KPIs will be optimized.
    - User QoE will be improved.

### 5.3.6.3 Mapping to ENI reference architecture

#### 5.3.6.3.1 Functional blocks

The functional blocks for application characteristic based network operation using AI is shown in Figure 5-18.

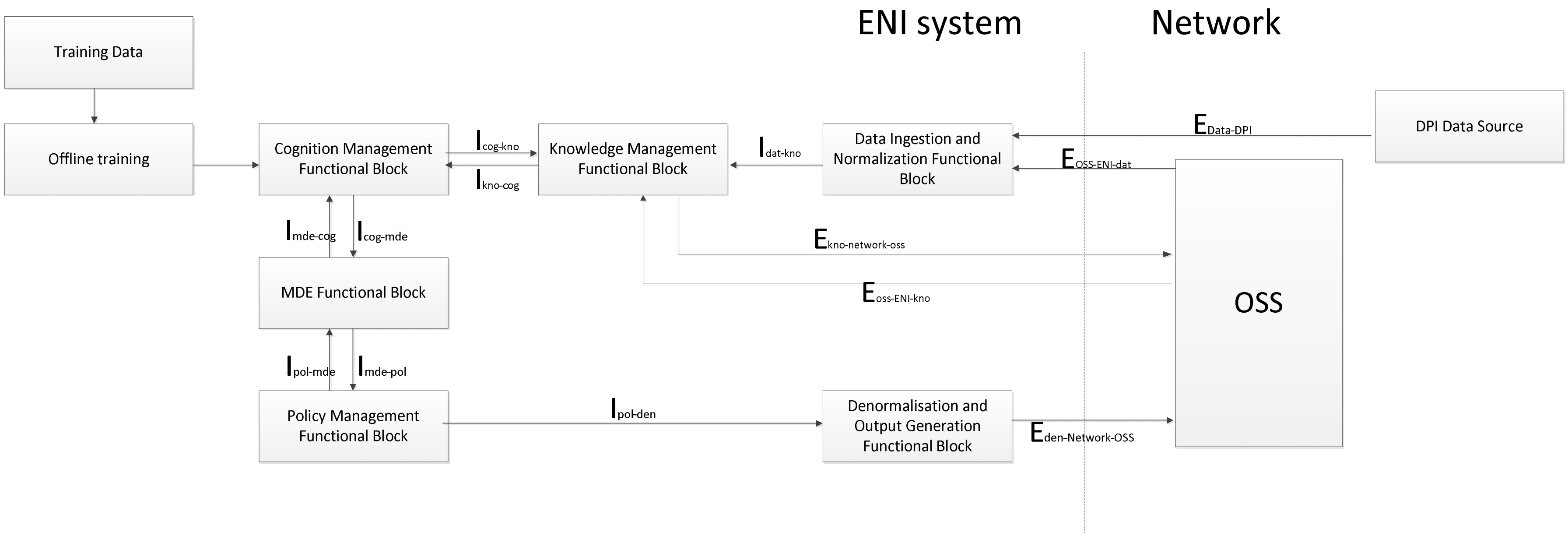


**Figure 5-18: Mapping to ENI reference architecture for Application Characteristic based Network Operation**

For training data, it can be pre-prepared one or the one from OSS and DPI data Source.

The "Data ingestion and Normalization Functional block" collects the network parameters like network performance from OSS and DPI data from "DPI Data Source". These data are filtered and normalized accordingly.

The "Knowledge Management Functional Block" generates inferences and passes that as well as data to "Cognition Management Functional Block". This functional block would also evaluate performance of model in "Cognition Management Functional Block". When model performance deteriorates, an updated feature set (including feature name and parameter) without invalid features is generated based on a feature set corresponding to the current data pattern (e.g. distribution and statistic characteristics, etc.) and a score table of current features of model (including feature name, validity score and validity flag), in which the validity score of a feature is negatively related to correlation with other features. And the feature list and this score table of features will be exchanged with OSS for review, modification and confirmation.

The "Cognition Management Functional Block" utilizes data and inferences to generate predictions based on modelling provided by offline training. For the modelling in "Cognition Management Functional block", it will be iterated periodically based on the data from DPI Data Source and OSS as well as updated features and parameters generated by "Knowledge Management Functional Block".

The "MDE Functional Block" translates the predictions into the form that is understandable by "Policy Management Functional Block". More usages of "MDE functional block" will be further investigated.

The "Policy Management Functional Block" makes the polices and inputs the decisions to "Denormalization and Output Generation Functional Block", which generates execution command to OSS.

#### 5.3.6.3.2 Reference Point

$E_{OSS\text{-}ENI\text{-}dat}$ defines data exchange between OSS and ENI System.

$E_{Data\text{-}DPI}$ defines data exchange between DPI data source (e.g. certain entity in core network or data centre) and ENI System.

Reference Point $I_{dat\text{-}kno}$ defines internal Reference Point between "Data ingestion and Normalization Functional block" and "Knowledge Management Functional Block".

Reference Point $I_{kon\text{-}cog}$ and $I_{cog\text{-}kon}$ define internal Reference Point between "Knowledge Management Functional Block" and "Cognition Management Functional Block".

Reference Point $I_{mde\text{-}cog}$ and $I_{cog\text{-}mde}$ define internal Reference Point between "Cognition Management Functional Block" and "MDE Functional Block". $I_{mde\text{-}pol}$ and $I_{pol\text{-}mde}$ define internal Reference Point between "MDE Functional Block" and "Policy Management Functional Block".

$I_{pol\text{-}den}$ defines data exchange between "Policy Management Functional Block" and "Denormalization Functional and Output Generation Block".

$E_{den\text{-}Network\text{-}OSS}$ defines data exchange between "Denormalization Functional and Output Generation Block" and OSS.

$E_{kno\text{-}network\text{-}oss}$ and $E_{oss\text{-}ENI\text{-}kno}$ define data exchange between "Knowledge Management Functional Block" and OSS.

#### 5.3.6.3.3 Flow of information

The flow of information is shown in following Figure 5-19.

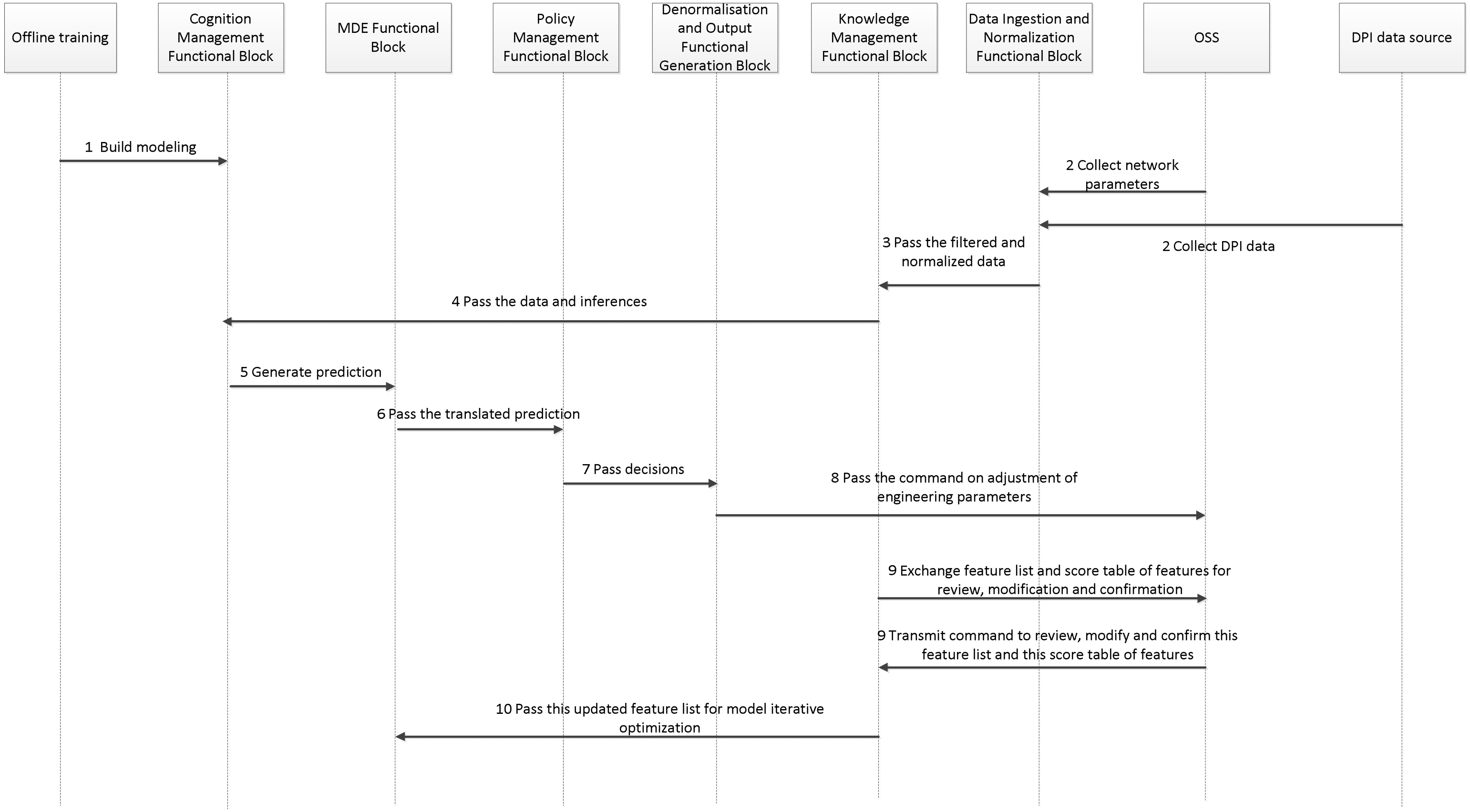


**Figure 5-19: Flow of information for Application Characteristic based Network Operation**

Step 1: Is about building modelling via offline training.

Step 2: Is that "Data ingestion and Normalization Functional block" collects data like network parameters from OSS and DPI data from DPI data source like certain entity in core network or data centre.

Step 3: Is that the filtered and normalized data is passed to "Knowledge Management Functional Block", which will pass the data and inferences in step 4.

Step 4: Pass the data and inferences to Knowledge Management Functional Block.

Step 5: "Cognition Management Functional Block" generates predictions and pass it to "MDE functional block", which is a model-driven functional block based on latest architecture.

Step 6: The translated predictions that are understandable by "Policy Management Functional Block" are passed.

Step 7: The decisions are passed to "Denormalization Functional and Output Generation Block".

Step 8: Command on adjustment of engineering parameters is passed to OSS.

Step 9: "Knowledge Management Functional Block" evaluates model performance and sends this feature list and this score table of features to OSS for review, modification and confirmation. Meanwhile, the command for review, modification and confirmation is transmitted from OSS to "Knowledge Management Functional Block".

Step 10: The updated feature set is delivered to "Cognition Management functional block" for model iterative optimization.

## 5.3.7 Use Case #2-7: AI enabled network traffic classification

### 5.3.7.1 Use case context

Network traffic classification plays an important role in network operation and management. Based on traffic classification techniques (e.g. port-based technique, payload-based technique, statistics-based technique), unknown traffic is categorized into a number of classes at the level of protocol (e.g. HTTP, SIP), public application (e.g. video, voice, download, instant messaging, online games, and virtual reality), and enterprise private application (e.g. desktop cloud, voice conference and video conference).

Network traffic classification is regarded as a fundamental work in 5G techniques (e.g. network slicing, network orchestration, and user plane policy rules). It is very essential for network operators to classify network traffic. On one hand, by processing different traffic classes respectively, network traffic classification supports numerous network closed-loop control activities in terms of network security, traffic engineering, Quality of Service (QoS). On the other hand, by providing traffic change or distribution information at network or service level, to support policy-making processes. Besides, classification results sever as the guideline for Operations Support System (OSS) and traffic forecast.

In addition, nowadays more and more traffic in the network is encapsulated in encrypted ways. HTTP over TLS/SSL (well known as HTTPs) and many other private encrypted protocols used by 3rd parties hide key features of Application Level in flows, posing great challenges to network traffic classification's effect. So a more intelligent and innovative mechanism needs to be introduced into future network to face such situation.

### 5.3.7.2 Description of the use case

#### 5.3.7.2.1 Overview

In this use case, one or more machine learning classifiers are trained and applied in the ENI System to classify real-time network traffic. The model training approach can be categorized as offline machine learning and online machine learning. For the offline training, designated network traffic is captured from network interfaces according to predefined rules, which is labelled to a corresponding class by manual or tool-assisted methods. The raw data stream features (e.g. destination IP address, destination port number and protocol type) or extracted packet features (e.g. packet length, inter-packet arrival time, and session time) with specific classification labels compose to the training data set. Then target models (e.g. Port-based classification model and Statistics-based classification model) are trained and generated based on machine learning algorithms (e.g. Random Forests, Convolutional Neural Network, and Recurrent Neural Network). In terms of the online training, the training data set is composed of data stream features or extracted packet features labelled automatically based on history classification results. The model parameters are dynamically adjusted according to the accuracy of classification results. Besides, based on the incremental learning, the model is further trained to adapt to application updating and avoid accuracy deterioration. It does not retrain the model. In the inference phase, the well-trained modules are implemented in ENI System and play the role of network traffic classification.

#### 5.3.7.2.2 Motivation

Network traffic classification can be achieved by various methods, such as port-based technique and payload-based technique. However, with the growth in the diversity of applications, traffic volume and the proportion of encapsulated traffic and enterprise private applications, traditional methods mentioned above are inefficient and even fail to classify network traffic. Hence, statistics-based technique is gradually being noticed and widely applied:

- Port-based technique: determining application layer protocol by an IP address and a firstly registered port in Internet Assign Number Authority (IANA). This method is efficient with high accuracy and low resource-consuming, but not reliable as many applications use dynamic IP address and port numbers (e.g. P2P).
- Payload-based technique: Deep Packet Inspection (DPI) is widely used in telecom networks, which inspects characteristic signatures in the packet payload to identify an application traffic. The method has high accuracy, but has high complexity and labour cost for matching the signature strings in packet Characteristic signatures shall be kept up to date, as the applications change very frequently. In addition, it cannot be used for encapsulated traffic.
- Statistics-based technique: the recent methods are using Machine Learning (ML) to classify data flows based on statistical packet features (e.g. packet length, packet transmit rate, inter-packet arrival time, etc.). As this technique is not based on IP address, port numbers, protocols and characteristic signatures, it does not encounter the problems of pervious methods. However, as collecting flow statistical characteristics, this method is not capable of a real-time traffic classification, which can affect time-sensitive policy making, e.g. game traffic steering, etc.

In this use case, AI enabled Network Traffic Classifier (AI-NTC) is proposed in the ENI System, which deals with the situation of non-available payloads or dynamic ports.

AI has achieved a better-than-human recognition rate in the field of image classification. Therefore, the AI-NTC transforms data stream features or extracted packet features into images and models the network traffic classification as the 'traffic image' classification. That is, by collecting data at the transport level (instead of application level) and applying machine learning algorithms, the ENI System is able to achieve accurate classification result with a relatively low overhead.

Such a scenario is illustrated in Figure 5-20.

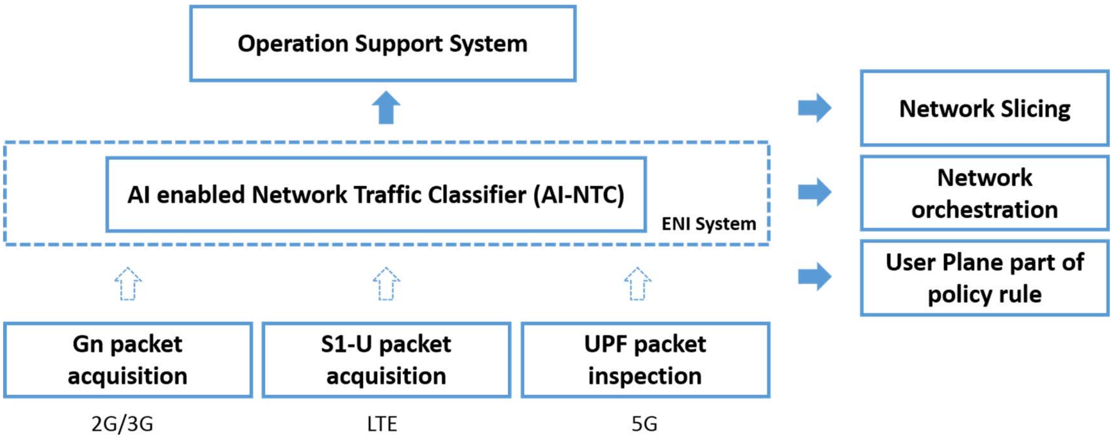


**Figure 5-20: Scenario of AI enabled network traffic classification**

#### 5.3.7.2.3 Actors and Roles

- Network Administrations: entity/person who defines target traffic classes and usage of classification results.
- Network Infrastructure: network resources for collecting target traffic data to ENI System, and management systems whose policies are based on traffic classification results.
- ENI System: system solution used to receive target traffic data from network infrastructure; to classify traffic data into target classes; to provide results to outer network systems in predefined formats (e.g. IP flow protocol defined IETF RFC 6645 [i.7]).

#### 5.3.7.2.4 Initial context configuration

- Mapping target classifications to predefined labels.
- Defining target feature types extracted from traffic datasets or raw data streams.
- Defining classification results formats.
- AI-NTC in ENI System is enabled with AI capability, which is initialized to learn target traffic classification patterns through target features or data streams in training data set automatically.
- Related management systems connect to ENI System for acquiring classification results.

#### 5.3.7.2.5 Triggering conditions

AI-NTC provides the classification results to outer related management systems, which check the change of the traffic classification distribution pattern. When the pattern changes drastically (for instance) and the current network resource usage cannot meet the requirement of this new pattern. Some network re-organization actions would happen step by step, e.g. allocating more UPFs, assigning different uplink-classifier policies, configuring the QoS priority and selecting routing, for steering some parts of the traffic.

#### 5.3.7.2.6 Operational flow of actions

ENI System with AI-NTC is intended to enhance the capability of traffic classification and optimize the network management based on the real-time classification results of traffic. This kind of mechanism is realized by introducing the new AI capability (e.g. normally based on the deep-learning algorithm and architecture), with the following flow of activities:

1) ENI System initiates a process to collect training traffic (can be extracted from the running network or off-line uploading) with specific classification labels, and use the labelled data to train the classification module. This process can be on-line all the time or for some time periods predefined.

2) ENI System initiates another process to infer the target traffic's classes by loading the well-trained module mentioned in 1). This process can be realized in real time.

3) ENI System arranges the classification results and provides them to outer management systems.

4) The related system receives the classification results and adjusts the policies according to the predefined rules.

#### 5.3.7.2.7 Post-conditions

- The network resources such as network slices, QoS priority and routing are re-allocated and re-adjusted according to the real-time traffic classification results. The changed network is well optimized based on that current traffic pattern and network KPIs.
- The module parameters are dynamically adjusted according to the classification result's evaluation from the network by means of, for example, reinforcement learning.
- The model can adapt to application updates and model performance deterioration by means of incremental learning.

### 5.3.7.3 Mapping to ENI reference architecture

#### 5.3.7.3.1 Functional blocks

**1) Option one: offline training**

Figure 5-21 illustrates the mapping to ENI reference architecture for the Use Case of 'AI enabled network traffic classification', in which a statistics-based model is established through offline training.

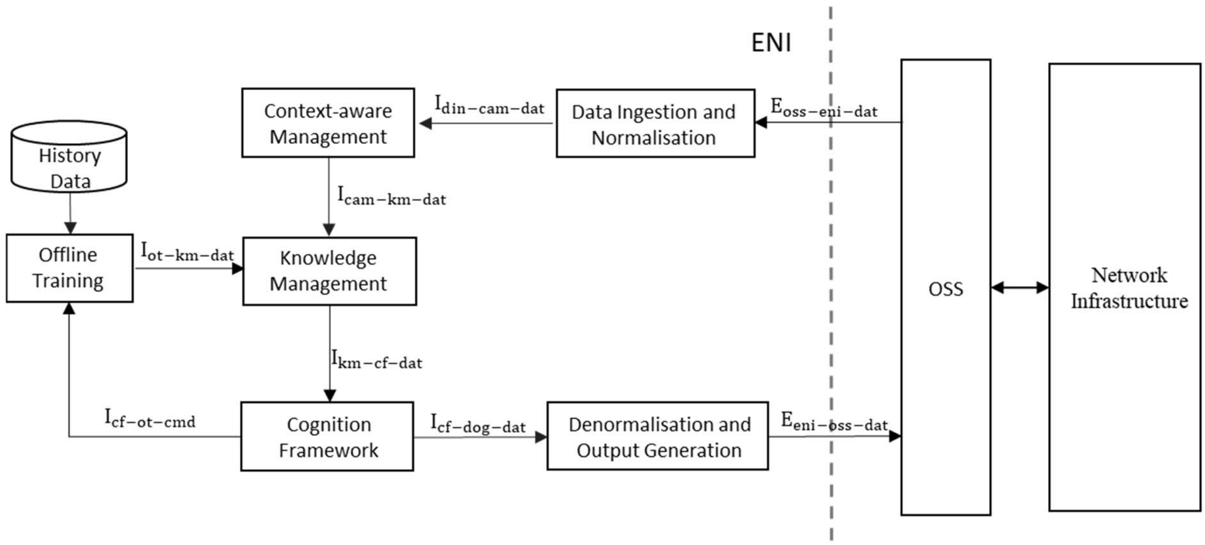


**Figure 5-21: Mapping to ENI reference architecture (option one)**

Data Ingestion and Normalization Functional Block collects the raw data stream and extracts current data stream features (e.g. destination IP address, destination port number and protocol type) and packet features (e.g. packet length, inter-packet arrival time, and session time) in accordance with the classification tasks and then normalize them to a common format.

Context-aware Management Functional Block interacts with the Data Ingestion and Normalization Functional Blocks to receive environment data (e.g. application updating). This functional block defines target classifications (e.g. video and webpage) corresponding to personalized and customized requirements.

Knowledge Management Functional Block stores a statistics-based model, which are generated through offline training based on extracted packet features (e.g. packet length and packet transmit rate) with classification labels, and defines a formal and consensual representation of knowledge so that the computer system could implement the machine learning algorithms and perform reasoning using the knowledge representation. When the statistics-based model is updated, it is stored in Knowledge Repository.

Cognition Framework Functional Block performs inferences by using the well-trained model and extracted features to get classification results. Depending on the classification accuracy, this block determines whether the classification model should be adjusted and informs Offline Training Functional Block. Then an updated statistics-based model is reloaded in Cognition Management Functional Block.

Denormalization and Output Generation Functional Block translates a normalized classification label to a form that the OSS can understand and then transfers it to OSS. According to the real-time traffic classification results, OSS can take some actions to optimize network performance.

**2) Option two: combination of offline training and online training**

Figure 5-22 illustrates the mapping to ENI reference architecture for the Use Case of 'AI enabled network traffic classification', in which a statistics-based model is generated through offline training and a data stream-based model is generated through online training are deployed.

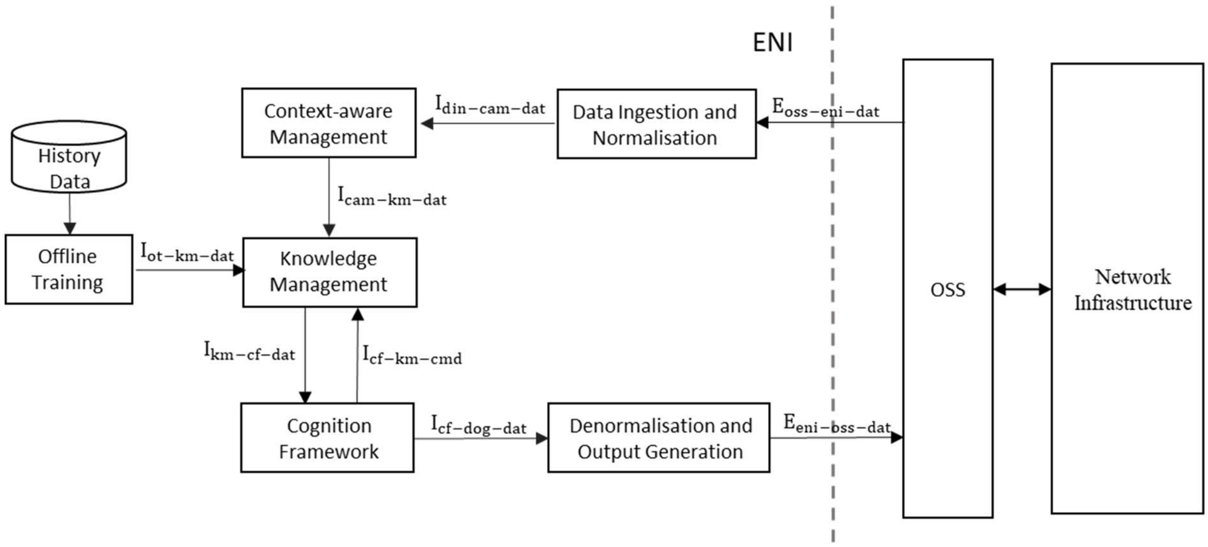


**Figure 5-22: Mapping to ENI reference architecture (option two)**

Data Ingestion and Normalization Functional Block collects the raw data stream and extracts current data stream features (e.g. destination IP address, destination port number and protocol type) and packet features (e.g. packet length, inter-packet arrival time, and session time) in accordance with the classification tasks and then normalize them to a common format.

Context-aware Management Functional Block interacts with the Data Ingestion and Normalization Functional Blocks to receive environment data (e.g. application updating). This functional block defines target classifications (e.g. video and webpage) corresponding to personalized and customized requirements.

Knowledge Management Functional Block stores a statistics-based model, which are generated through offline training based on extracted packet features (e.g. packet length and packet transmit rate) with classification labels, and defines which features should be extracted according to target traffic classes. Besides, this functional block also either edits the existing knowledge or adds a new knowledge about current data stream features and the corresponding classification label in Knowledge Repository if any of the required knowledge changes (i.e. classification label of the same data stream features changes) or non-available. Depending on the updated knowledge, an online training process of a model based on data stream features is triggered. Then an updated model based on data stream features is stored in Knowledge Repository.

Cognition Framework Functional Block implements both statistics-based model and data stream-based model, which are respectively generated through offline training based on extracted packet features (e.g. packet length and packet transmit rate) with classification labels and online training based on data stream features (including destination IP address and protocol type) with classification labels derived from the statistics-based model. Then, classification results and corresponding model confidences are inferred from above two models. In this functional block, a synthesis model confidence is defined as a function of two model confidences above. When the synthesis model confidence is higher than a classification threshold, a classification label of current data stream is determined. Otherwise, if the synthesis model confidence is lower than an updating threshold (which is larger than the classification threshold), knowledge about current data stream features and the classification label are transmitted to Knowledge Management Functional Block and stored in Knowledge Repository. When a data stream-based model is updated, Cognition Framework Functional Block reloads it from Knowledge Management Functional Block.

Denormalization and Output Generation Functional Block translates a normalized classification result of current data stream to a form that the OSS can understand and then transfers it to OSS. According to the real-time traffic classification results, OSS can determine a traffic control policy for current data stream, such as traffic engineering, QoS management.

### 5.3.7.3.2 Interfaces

**1) Option one: method of offline training**

$E_{oss\text{-}eni\text{-}dat}$ defines data exchange between the ENI System and the OSS (Assisted System). The OSS collects traffic data from network infrastructure that belongs to different domains (e.g. RAN/Fixed Access, Transport, and Core) and sends it to ENI System.

$E_{eni\text{-}oss\text{-}dat}$ defines data exchange between the ENI System and the OSS. The ENI System provides real-time traffic classification results so that the OSS could take actions accordingly, e.g. checking the change of the traffic classification distribution pattern.

$I_{din\text{-}cam\text{-}dat}$ defined data exchange between Data Ingestion and Normalization Functional Block and Context-aware Management Functional Block. Data Ingestion and Normalization Functional Block passes the normalized data, information of application updating and context information to Context-aware Management Functional Block.

$I_{cam\text{-}km\text{-}dat}$ defined data exchange between Context-aware Management Functional Block and Knowledge Management Functional Block. Knowledge Management Functional Block passes target classifications and feature extraction rules to Knowledge Management Functional Block.

$I_{km\text{-}cf\text{-}dat}$ defined data exchange between Knowledge Management Functional Block and Cognition Framework Functional Block. Knowledge Management Functional Block passes a formal and consensual representation of knowledge to Cognition Framework Functional Block to perform inferences.

$I_{cf\text{-}dog\text{-}dat}$ defined data exchange between Cognition Framework Functional Block and Denormalization and Output Generation Functional Block. The data includes the traffic classification labels inferred by ENI System.

$I_{cf\text{-}ot\text{-}cmd}$ defined recommendations and/or commands exchange between Cognition Framework Functional Block and Offline Training Functional Block. Cognition Framework Functional Block informs Offline Training Functional Block to adjust module parameters or retrain the traffic classification model.

$I_{ot\text{-}km\text{-}dat}$ defined data exchange between Offline Training Functional Block and Knowledge Management Functional Block. Offline Training Functional Block loads an offline training model and its updated model into Knowledge Management Functional Block.

**2) Option two: method of combining offline training with online training**

$E_{oss\text{-}eni\text{-}dat}$ defines data exchange between the ENI System and the OSS (Assisted System). The OSS collects traffic data from network infrastructure that belongs to different domains (e.g. RAN/Fixed Access, Transport, and Core) and sends it to ENI System.

$E_{eni\text{-}oss\text{-}dat}$ defines data exchange between the ENI System and the OSS. The ENI System provides real-time traffic classification results so that the OSS could take actions accordingly, e.g. checking the change of the traffic classification distribution pattern.

$I_{din\text{-}cam\text{-}dat}$ defined data exchange between Data Ingestion and Normalization Functional Block and Context-aware Management Functional Block. Data Ingestion and Normalization Functional Block passes the normalized data, information of application updating and context information to Context-aware Management Functional Block.

$I_{cam\text{-}km\text{-}dat}$ defined data exchange between Context-aware Management Functional Block and Knowledge Management Functional Block. Knowledge Management Functional Block passes target classifications and feature extraction rules to Knowledge Management Functional Block.

$I_{km\text{-}cf\text{-}dat}$ defined data exchange between Knowledge Management Functional Block and Cognition Framework Functional Block. Knowledge Management Functional Block passes a formal and consensual representation of knowledge to Cognition Framework Functional Block to perform inferences. And Knowledge Management Functional Block reloads an updated model into Cognition Framework Functional Block.

$I_{cf\text{-}km\text{-}dat}$ defined data exchange between Cognition Framework Functional Block and Knowledge Management Functional Block. Cognition Framework Functional Block transmits knowledge about current data stream features and the corresponding classification label to Knowledge Management Functional Block.

$I_{cf\text{-}dog\text{-}dat}$ defined data exchange between Cognition Framework Functional Block and Denormalization and Output Generation Functional Block. The data includes the traffic classification labels inferred by ENI System.

$I_{ot\text{-}km\text{-}dat}$ defined data exchange between Offline Training Functional Block and Knowledge Management Functional Block. Offline Training Functional Block loads an offline training model and its updated model into Knowledge Management Functional Block.

### 5.3.7.3.3 Flow of information

**1) Option one: method of offline training**

The flow of information is given in Figure 5-23.

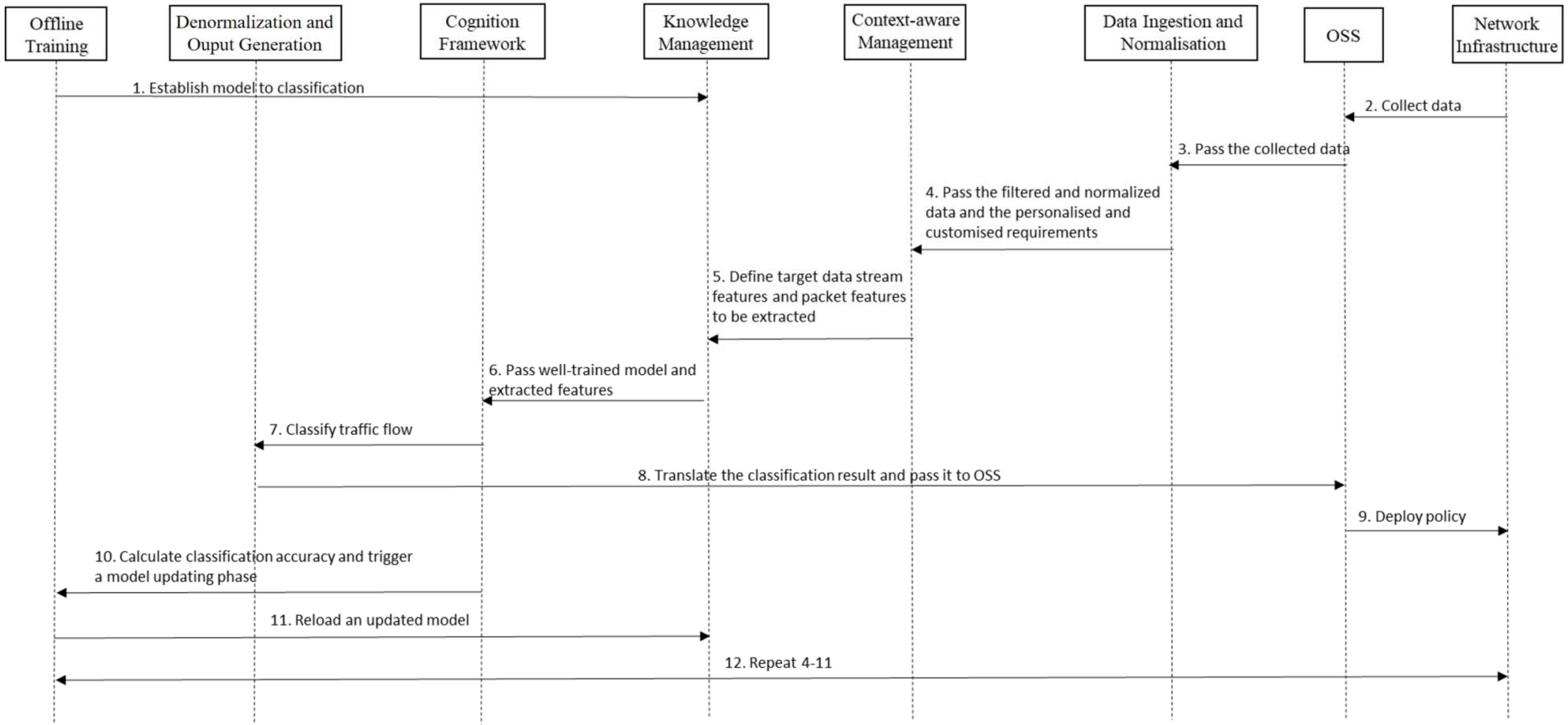


**Figure 5-23: Scenario of AI enabled network traffic classification (option one)**

Step 1: Training the traffic classification model with historical data.

Step 2: The OSS collects unknown traffic data from network infrastructure.

Step 3: The collected data is sent to the ENI System.

Step 4: Data Ingestion and Normalization Functional Block processes the raw data and then transfers the processed the personalized and customized requirements to Context-aware Management Functional Block.

Step 5: Context-aware Management Functional Block defines target data stream features and packet features which is to be extracted and informs Knowledge Management Functional Block.

Step 6: Knowledge Management Functional Block passes the filtered and normalized data, as well as extracted data stream features and packet features to Cognition Framework Functional Block.

Step 7: Cognition Framework Functional Block performs inferences and sends classification results to Denormalization and Output Generation Functional Block.

Step 8: Denormalization and Output Generation Functional Block gets the traffic classification results and converts them to a form that OSS can understand, and then transfers it to OSS.

Step 9: OSS adjusts the configuration of network infrastructure.

Step 10: Cognition Framework Functional Block depends on classification accuracy and informs Offline Training Functional Block to retrain and update model.

Step 11: Offline Training Functional Block provides Cognition Framework Functional Block with the updated model and then Knowledge Management Functional Block replaces the saved model.

Step 12: Repeat step 4 to 11.

**2) Option two: method of combining offline training with online training**

The flow of information is given in Figure 5-24.

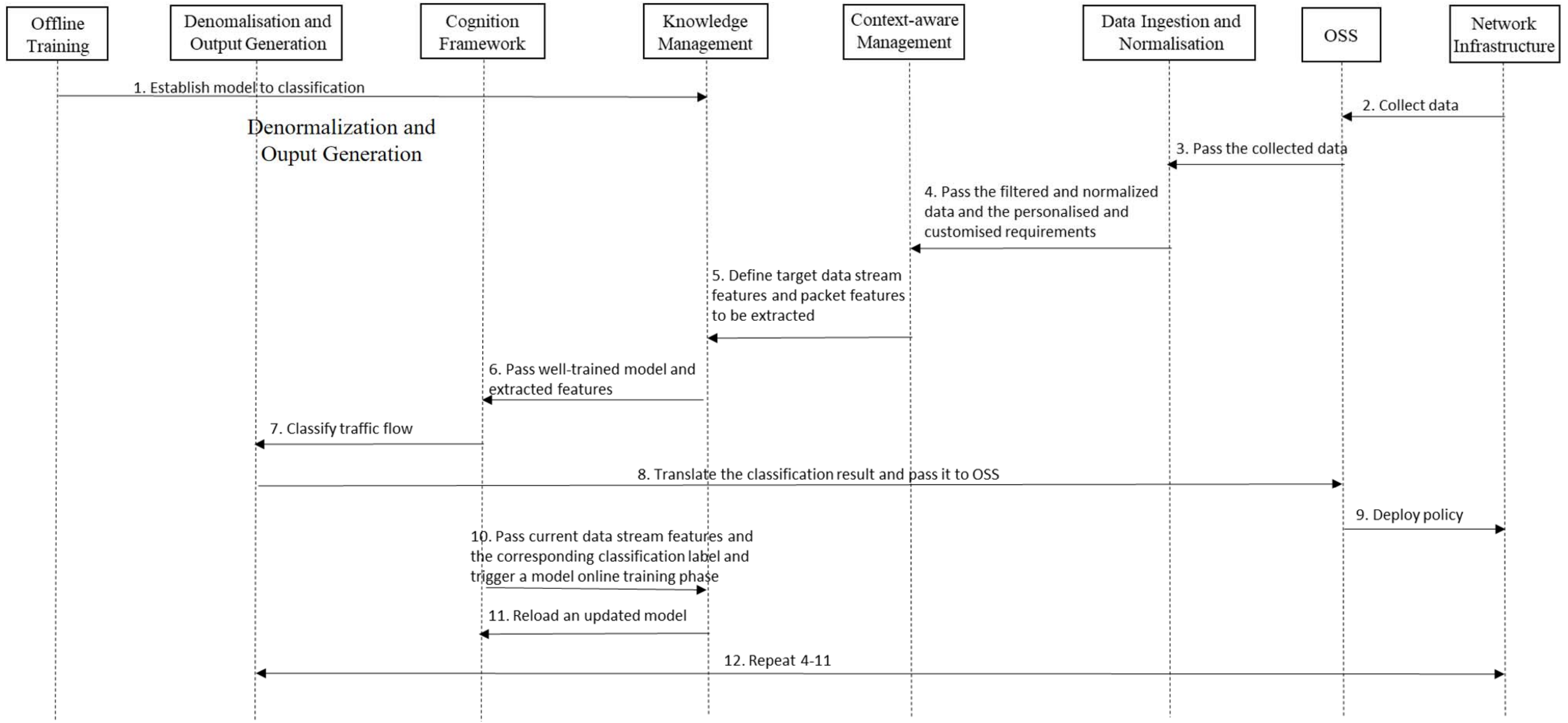


**Figure 5-24: Scenario of AI enabled network traffic classification (option two)**

Step 1: Training the traffic classification model with historical data.

Step 2: The OSS collects unknown traffic data from network infrastructure.

Step 3: The collected data is sent to the ENI System.

Step 4: Data Ingestion and Normalization Functional Block processes the raw data and then transfers the processed the personalized and customized requirements to Context-aware Management Functional Block.

Step 5: Context-aware Management Functional Block defines target data stream features and packet features which is to be extracted and informs Knowledge Management Functional Block.

Step 6: Knowledge Management Functional Block passes the filtered and normalized data, as well as extracted data stream features and packet features to Cognition Framework Functional Block.

Step 7: Cognition Framework Functional Block performs inferences and sends classification results to Denormalization and Output Generation Functional Block.

Step 8: Denormalization and Output Generation Functional Block gets the traffic classification results and converts them to a form that OSS can understand, and then transfers it to OSS.

Step 9: OSS adjusts the configuration of network infrastructure.

Step 10: Cognition Framework Functional Block transmits knowledge about current data stream features and the classification label to Knowledge Management Functional Block and depends on synthesis model confidence to trigger an online training in Knowledge Management Functional Block.

Step 11: Knowledge Management Functional Block provides Cognition Framework Functional Block with the updated model and then Cognition Framework Functional Block replaces the saved model.

Step 12: Repeat steps 4 to 11.

### 5.3.8 Use Case #2-8: Automatic service and resource design framework for cloud service

#### 5.3.8.1 Use case context

Cloud service based on Virtualisation technology enables prompt on-demand realization of various functions on the Virtualised platform. An increasing number of Service Providers (SP) such as ecommerce companies, science institutes, are implementing various kinds of services and functions, e.g. web service, machine learning in the cloud environment provided by the Cloud Provider (CP). As shown in Figure 5-25 the SP is concerned about the service requirements, such as the functionality of the service, the levels of security and reliability, and the ability to handle workloads. In contrast, CP needs to know the composition of resources and amount of resources to be allocated when fulfilling the service orders. The resource composition describes the types of resources and the connectivity between them. For instance, resource composition for a basic web service is web server instance connected to a database instance. Resource amount is the amount of vCPU and memory, disk to allocate to each instance aforementioned. Therefore, in accordance with the service requirements, cloud resource composition and amount need to be designed in various phases of cloud service delivery. Currently the design heavily relies on the human decision-making process either on the SP side or the CP side. Practices to design the cloud resource include:

- Self-service approach. The SP is provided with a management interface to manage cloud resources and needs to decide their resource requirements. The approach requires the SP to have IT expertise and may put up barriers to SPs wishing to enter the market.

- Cloud-consultant approach. The SP is assisted by cloud consultants from the CSP who collect the service requirements from SP and decide resource details accordingly. The approach demands the CSP spend high Operating Expenditure (OPEX) and leads to them needing a relatively long time to deliver services.

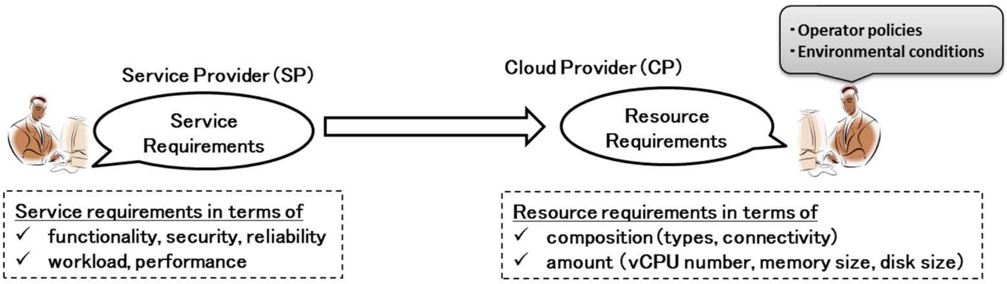


**Figure 5-25: Design cloud resources in accordance with service requirements**

### 5.3.8.2 Description of the use case

#### 5.3.8.2.1 Overview

As addressed above, the current approaches to design the cloud resources lead to high human cost on the SP or CP side. This use case aims to reduce the cost by automating the design process leveraging the previous human design/operation knowledge. The automatic cloud resource design framework is expected to be able to design cloud resource in accordance with service requirements in the following 3 aspects as shown in Figure 5-26:

- Service requirement analysis: SPs describe the Service Requirements through various channels, e.g. natural language and GUI. The system needs a requirements analysis entity that is responsible for analysing Service Requirements captured in various channels e.g. natural language or APIs, and categorizing the requirements into atom requirements, i.e. requirements regarding functionality, security, reliability, performance, etc. In the case of service requirements in natural language, the analysis function is realized based on the natural language processing model, which is generated from machine learning of human analysis results data set.

- Resource composition design: This entity takes the result of the requirements analysis, and is able to customize the resource composition automatically in accordance with the atom service requirements using customize rules. These customize rules define the actions to be taken, i.e. how to derive resource composition given different functionality, security and reliability requirements. The rules to customize the service composition may be statically generated by the operator or dynamically based on the machine learning of previous composition design examples.

- Resource amount design: This entity decides the resource amount needed to satisfy performance requirements. Besides the workload and the performance requirements, environmental conditions and operator policies need to be considered to decide the amount of computation resources allocated to e.g. Virtual Machines (VMs). In this context, environmental conditions may include the static conditions, e.g. Central Processing Unit (CPU) clocks and memory architecture of the host, and dynamic conditions, e.g. resource utilization ratio of the physical host to which the VM is allocated. In addition, operator policies are statically set by the SP or CP and restrict the operation state within a desired range, e.g. an operator policy restricts the computation resource usage of a VM to 50 % - 90 % to prevent resource underuse or overuse. The model used to calculate the resource amount is generated from the log data including the resource amount configuration data, performance data, environmental data and operational state data. These data can be obtained by utilizing the ENI System external RPs including or-eni-cfg, app-eni-ctx, inf-eni-dat.

The resource design result includes the resource composition and the resource amount. This information is fed back to the SP engineer and to the CP operator for evaluating the design result. On the other hand, the resource management and orchestration system are enforced with the design result to implement the service requested at the upper level. Finally, the Infrastructure system, which is a NFVI-like System composed by APPs, VMs and overall hardware, is also enforced through the inf-eni-dat RPs with the design result to implement the service requested at the lower level. On its turn, this NFVI-like System feeds back the Resource Design System, though the inf-eni-dat RP, with raw data for evaluation as part of the monitoring process.

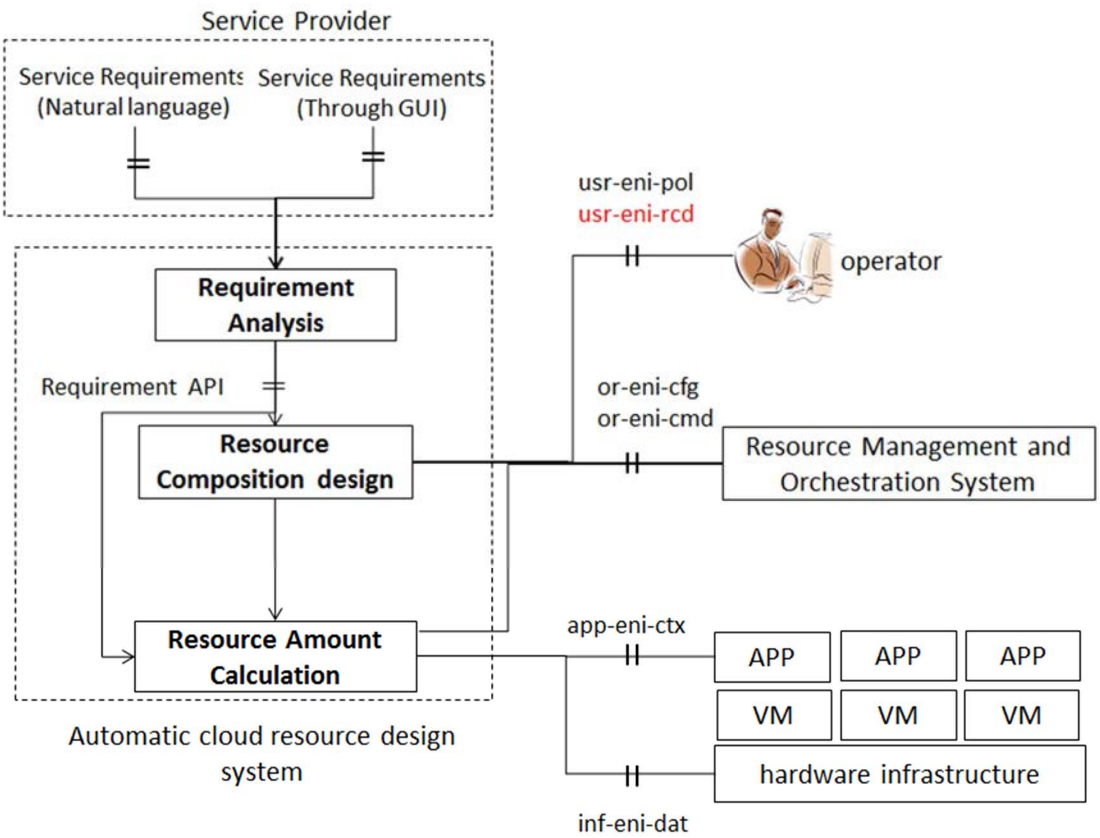


**Figure 5-26: Automatic Cloud Resource Design System**

#### 5.3.8.2.2 Motivation

Automatic design of cloud resource benefits in the consultation, design, and operation phases of cloud service delivery. For example, in the consultation phase, for an SP who plans to migrate services implemented in an on-premises environment into a cloud environment, it can be used to show the performance that can be achieved after the migration and the needed cloud resources and cost. In the design phase, it enables the automatic design of resources, thus service design time and human labour reduction can be expected. In the operation phase, it is able to adjust the resource composition and amount in accordance with the changes, thus ensuring the continuous satisfaction of service requirements, which contributes to higher customer satisfaction.

#### 5.3.8.2.3 Actors and Roles

**SP:** it requests a cloud service to the design system, and use the cloud service to implement specific functions and services. Therefore, it is more concerned on the service requirements (functionality, security, performance, etc.).

**Engineer of SP:** it is the Assisted System (party) of Automatic Cloud Resource Design System. The design result produced by the Automatic Cloud Resource Design System is fed back to it, and it is able to accept, decline or edit the design result and instruct the Resource management and orchestration Assisted System to orchestrate the resource based on the design results. It is able to add/edit/delete the operator policies used in the system directly through user-eni-pol RP.

**CP:** it provides cloud service to the SP. To implement the cloud service, it needs the information about the composition of cloud resources, and the amount of computation resources (vCPU, memory, and disk).

**Operator of CP:** it is the Assisted System (party) of Automatic Cloud Resource Design System. The design result produced by the Automatic Cloud Resource Design System is fed back to it, and it is able to accept, decline or edit the design result and instruct the Resource management and orchestration Assisted System to orchestrate the resource based on the design results. It is able to add/edit/delete the operator policies used in the system directly through user-eni-pol RP.

**Automatic Cloud Resource Design System:** it consists of three main entities i.e. service requirements analysis, resource composition design, and resource amount design, as defined in clause 5.3.8.2.1 on overview of the Use Case.

**Resource management and orchestration system:** it is the Assisted System of Automatic Cloud Resource Design System. It is responsible for the orchestration, configuration, and activation and monitoring of the cloud resources.

**NFVI-like infrastructure:** it is the Assisted System composed by APPs, VMs, and other hardware usually associated with a Virtualised network infrastructure.

##### 5.3.8.2.4 Initial context configuration

The automatic cloud resource design system exposes interfaces for the SP and the CSP operator from the three function blocks. The SP or CSP can use the system to do the requirement analysis, resource composition design and resource amount design.

##### 5.3.8.2.5 Triggering conditions

- Consultation: The SP engineer uses the Automatic Cloud Resource Design System to estimate the needed resource before requesting the cloud service.
- Design: The SP places a service request and, to meet the service requirements, the Automatic Cloud Resource Design System is utilized to decide the resource needed.
- Operation: After the service delivery, changes in workload, environment etc., may be detected that may lead to service requirements violation. Based on the monitored data that is fed back from the NFVI-like infrastructure System, the Automatic Cloud Resource Design System is able to redesign resource composition and the resource amount to ensure the service requirements are satisfied again.

##### 5.3.8.2.6 Operational flow of actions

1) The models used respectively in the three function blocks in the automatic cloud resource design system are trained based on previous design experiences and operation log data as mentioned before.
2) The service requirement analysis function parses the service requirements into atom standard requirements.
3) The resource composition design function customizes the resource composition according to the atom requirements.
4) The resource amount calculation function decides the resource amount in accordance with the workload, performance requirement based on current environmental conditions and operation policies.
5) The operator optimizes the design result if the service requirements are not satisfied.
6) The system models are retained based on the new human design results and monitoring data.

At the beginning, action 1 is taken to prepare the automatic cloud resource design system.

In the consultancy phase, the automatic cloud resource design system works in recommendation mode. The SP inputs the service requirements in any available formats. Actions 2 to 4 are taken and the output of the automatic cloud resource design system is feedback to the SP. The SP may also input partial requirements, in this case, the system takes one or multiple actions in action 2 to 4 and feedback the results to the SP.

In the design phase, the automatic cloud resource design system works in recommendation or management mode. Actions 2 to 4 are taken. The results are passed to the CSP operator and resource management and orchestration system. If the requirements are not met, action 5 is taken.

In the operation phase, the automatic cloud resource design system works in recommendation or management mode. The system collects the monitoring data of service, and if there are changes in workload, environmental conditions, and operation policies, the automatic cloud resource design system is used to adjust the resource.

The system takes action 6 to update the system model automatically or under the instruction.

#### 5.3.8.2.7 Post-conditions

In various phases of cloud service delivery, precise and seamless resource design in accordance with service requirements in a much shorter time compared to human design. The human cost to design the resource is reduced to a large extent. Meanwhile, the accuracy of resource decision is improved by the continuous data collection and training of models.

## 5.3.9 Use Case #2-9: Intelligent time synchronization of network

### 5.3.9.1 Use case context

With wide application of information technology and Internet, many industries and fields are rapidly putting the business on the internet. For example, automation and networking have been realized in aviation, finance, railway transportation, medical and other systems, so there shall be a coordinated and unified time to ensure these systems can work together. At the same time, data traceability and analysis, data interaction of information system such as Internet of things, cloud computing and big data are all based on accurate time. Therefore, it is more and more important to get a unified standard time.

High-precision time synchronization is one of the key requirements of 5G network. The application scenarios of time synchronization is very extensive. In some scenarios, time synchronization is needed on a large scale while the accuracy of time synchronization is not highly requested. However, the time synchronization is strictly required which should reach nanosecond in some application scenarios, such as military command system, financial transaction system. Therefore, it is significant to improve the accuracy of time synchronization.

For these reasons, an intelligent time synchronization of network scheme is proposed, which can effectively improve accuracy of time synchronization. The prediction model can accurately predict time offset and time skew rate by using AI technology. According to predict results, the system can adjust its clock time to reduce the deviation from the standard time.

### 5.3.9.2 Description of the use case

#### 5.3.9.2.1 Overview

In this use case, intelligent time synchronization of network is based on AI-based prediction module which can effectively improve accuracy of time synchronization. This intelligent time synchronization can be used in many scenarios such as fault location, transport intelligent dispatching, financial transaction system and public security management. The intelligent system is divided to three modules:

- Data acquisition module that used to obtain observation data (e.g. clock time, time offset, frequency offset, time skew rate, and channel environment) and accuracy requirement according to predefined rules.
- Control module that used to decide which machine learning algorithm to invoke according to accuracy requirement.
- AI-based prediction module that used to predict the time offset and time skew rate.

In the model training phase, the prediction models are trained on the basis of appropriate machine learning and observation data. Then the trained prediction model estimates the time offset and time skew rate. In the inference phase, AI-based prediction module implemented in the ENI System. The ENI System provides the predicted results to the external system. The external system dynamically adjusts the local clock time to reduce the deviation from the standard time, and achieves accurate time synchronization. In addition, the parameters of model can be dynamically adjusted according to the prediction results.

#### 5.3.9.2.2 Motivation

Time synchronization requires that the time offset between system time and standard time be limited to a small range. There is a deviation when standard time is distributed to the subordinate node by using wired or wireless network, namely time delay which including signal processing delay, signal transmission delay, medium access delay, noise interference delay, etc. Therefore, the system time is not synchronized with standard time, or time synchronization accuracy is not ideal.

Time synchronization can be achieved by various methods, such as time synchronization based on the Precision Time Protocol (PTP). However, with the increasing demand on time synchronization, the traditional methods cannot meet the accuracy requirements of time synchronization in some scenarios (e.g. traffic management system, fault location, and financial transaction system).

Time synchronization based on PTP: master clock and slave clock add time stamps at the network link layer in order to accurately record the time when the message is received or sent. The master clock and slave clock obtain the timestamps by exchanging message through four times, thereby achieving calculation of the time offset. However, the time offset is inaccurate for PTP ignores the asymmetry of channel .In addition, PTP requires hardware support which costs a lot.

In this use case, an AI-based time synchronization is proposed in the ENI System. Appropriate algorithms in machine learning are introduced to estimate the time offset and time skew rate according to different accuracy demands on time synchronization, thus accuracy of time synchronization is improved.

The intelligent time synchronization of network is illustrated in Figure 5-27.

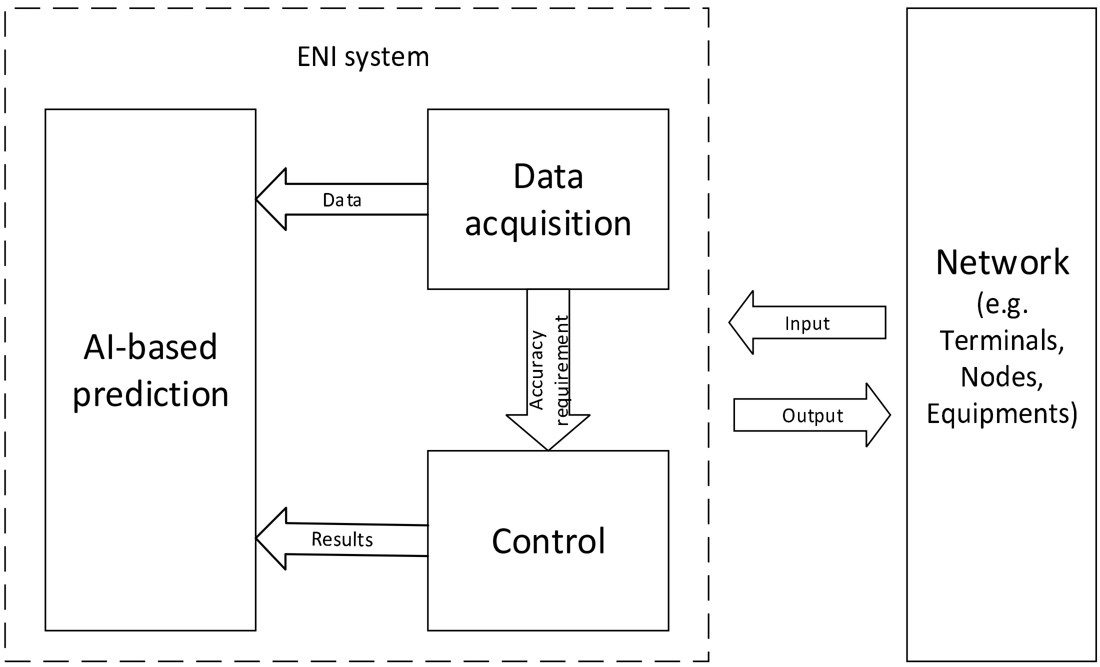


**Figure 5-27: Intelligent time synchronization of network**

#### 5.3.9.2.3 Actors and Roles

- ENI System: system solution used to receive observation data from network; to predict the time offset and time skew rate according to the observation data; to provide results to outer network systems. (Input: observation data (e.g. clock time, time offset, frequency offset, time skew rate, and channel environment) and accuracy requirement. Output: the prediction results).
- Network: entity that provide the observation data and accuracy requirement.

#### 5.3.9.2.4 Initial context configuration

- Define the accuracy level of time synchronization.
- Observation data obtains on the basis of predefined rules, and then match the accuracy requirement with the accuracy level.
- Time synchronization in ENI System is enabled with AI capability, which is initialled to learn target prediction through observation data.

#### 5.3.9.2.5 Triggering conditions

In this use case, when the predicted rules cannot hit the accuracy of time synchronization or with the approaching predicted period, the process of predicting the time offset and time skew rate will be initiated. In the meantime, the parameters of the prediction model will be dynamically adjusted according to the prediction results.

#### 5.3.9.2.6 Operational flow of actions

In the ENI System, AI-based time synchronization is intended to improve the accuracy of time synchronization by introducing the new AI capability. The time offset and time skew rate can be predicted by using machine learning method, with the following flow of activities:

1) The ENI System initiates a process to obtain observation data (e.g. clock time, time offset, frequency offset, time skew rate, and channel environment) by using predefined rules.

2) Determine which machine learning algorithm to call according to the accuracy level.

3) The ENI System initiates a process to train AI prediction model on basis of observation data.

4) The trained AI prediction model estimates the time offset and the time skew rate.

5) The ENI System provides the predicted results to the external system.

6) In order to reduce the deviation between local time and standard time, the external system adjusts its clock time based on the received results and predefined rules.

#### 5.3.9.2.7 Post-conditions

- The external system adjusts its clock time on the basis of the prediction results provided by the ENI System. In this way, the deviation between the system time and the standard time is reduced, and the accuracy of time synchronization is improved.

- The prediction model dynamically adjusts the parameters of the model based on the prediction results.

- At the next prediction period, the prediction model will predict the time offset and time skew rate for the next state.

- The ENI System dynamically adjusts the period of time synchronization based on the feedback from the external system.

### 5.3.10 Use Case #2-10: Intelligent Content-Aware Real-Time Gaming Network

#### 5.3.10.1 Use Case context

There is an increasing trend that people access to their games with mobile devices anytime anywhere. By the end of 2018, at USD 70,3 billion, mobile gaming accounts for more than 50 % of global gaming revenue and 76 % of mobile application revenue on device.

Network latency plays a crucial role in user experience for a number of real-time online games, as this can directly affect the performance of a player in these games. For example, for real-time multiplayer competitive games, an End-to-End (E2E) latency above 20 ms will lead to that players lose games and user experience severely degrades. On the contrary, for certain online games, the impact of larger E2E latency is less significant. Higher data rates are required by players that pursue top gaming and visual qualities. A high-resolution 60 Frames Per Second (FPS) game will generate heavy loads in DownLink (DL) as the bit rate increases. Meanwhile, action games will generate heavy loads in UpLink (UL) due to frequent operations by the player.

It can be anticipated that emerging new online games will have diverse and demanding requirements on E2E latency. Legacy or 5G networks currently assign Quality of service Class Identifiers (QCIs) to E2E links in order to optimize latency and data rates. However, these QCIs can only provide coarse indications for real-time online games and hence are not able to provide distinguishing optimizations for different types of games.

### 5.3.10.2 Description of the use case

#### 5.3.10.2.1 Overview

In the current 5G standard, there are coarse indicators, i.e. QCI, flagging the QoS (e.g. delay budget) requirements for different bearers. One of the QCI is known as real-time gaming, which defines one set of Key Performance Indicators (KPIs) such as delay budget and traffic priority for all types of real-time games. However, current 5G standard does not have the signalling or protocol in place for end-to-end link optimization to the granularity of contents - different 'genre' of games, and it is unlikely to have this capability in standards in near future. For data privacy purpose, it is impossible for the 5G network to know what type of game is being played, and how to intelligently schedule network resources accordingly. A feasible way to provide optimized end-to-end links is to design a content-ware 5G network, which uses AI to understand network contexts, e.g. the 'genre' of the game, by analysing traffic pattern. The content-ware 5G network can then intelligently schedule network resources to meet traffic requirements.

The new Use Case is applying AI technologies, and the resulting interfaces and network components to the optimization of real-time mobile gaming links. The key goal of this use case is to provide top quality of gaming experience by intelligent management and scheduling of network resources.

#### 5.3.10.2.2 Motivation

Management of radio resources for real-time gaming links are non-trivial, because different types of real-time games have different throughput and delay-budget requirements. Also, each gaming traffic is different as the interaction between a player and the game is unique. The current QCI categorize all real-time gaming into one group and this does not provide optimized gaming experience or resource management. As a result, AI is a promising way to tackle this quickly emerging use case.

The application of AI under such context will bring generate tailor-made QoS requirements for each mobile gaming wireless link. The AI-enabled 5G network can smartly allocate radio resources to provide optimization on these links.

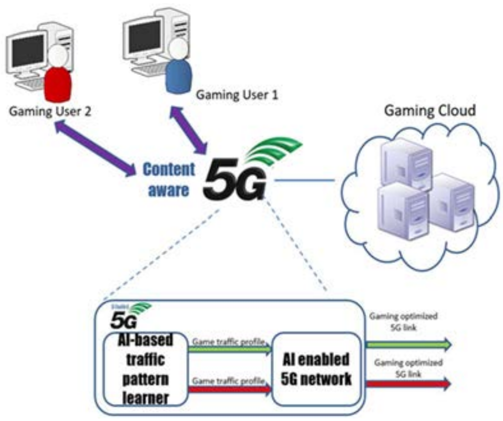


**Figure 5-28: Content-Aware gaming network**

#### 5.3.10.2.3 Actors and Roles

- Operator: Provides load estimation, uplink and downlink traffic profiles to the ENI System.
- AI-based traffic pattern learner: Collects uplink and downlink traffic pattern and generates game traffic profiles to the 5G network.
- 5G network: Collects game traffic profiles and provides optimized 5G links for gaming.

#### 5.3.10.2.4 Initial context configuration

The initial network will be a regular 4G/5G network loaded with a trained AIML-based traffic pattern learner.

#### 5.3.10.2.5 Triggering condition

When the data traffic is matched with a gaming profile.

#### 5.3.10.2.6 Operational flow of actions

The content-aware network comprises a traffic analyser, a gaming profile generator and a content-aware network manager. The operational flow includes the optimizing a network, in connection with a session of a particular user of a gaming service in the network, with the following steps:

- analysing data flow between the particular user and the network;
- generating a profile as a result of the analysis and passing the profile to the network where the profile is indicative of a particular type of game; and
- configuring the network in accordance with the profile.

Flow of actions include passing the UL and DL gaming traffic to the AIML-based traffic learner. The AIML-based traffic learner will analyse and generate game profiles for each user (without decoding packets). After receiving the decision about game profiles from the traffic pattern analyser, the mobile gaming network can prioritize gaming traffic based on the contents (game profiles) and schedule optimized radio resources to the gaming client.

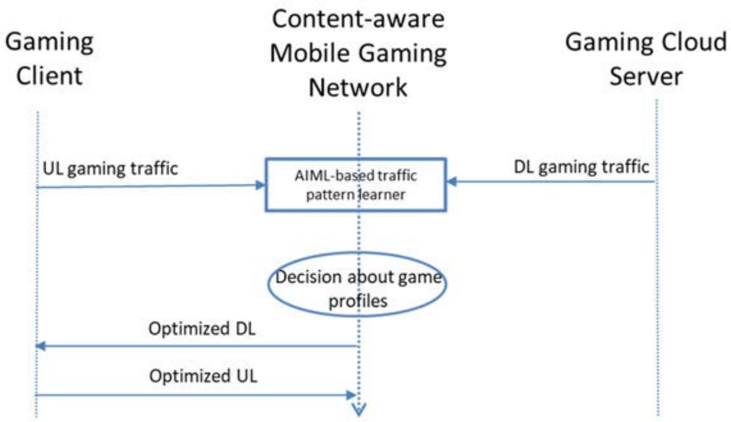


**Figure 5-29: Operational flow of actions for the Use Case**

#### 5.3.10.2.7 Post-condition

Every E2E links of diverse types of real-time gaming have met their requirements on E2E latencies and data rates.

## 5.3.11 Use Case #2-11: Intent-driven operating for user-centric cloud-network convergence services

### 5.3.11.1 Use case context

Cloud services are constantly reshaping social production and lifestyles. Home broadband applications (such as cloud games), large enterprises (such as governmental and finance enterprises), and small- and medium-sized enterprises (such as hospitals and e-commerce enterprises) are greatly facilitated by cloud services. Enterprises are using cloud services to reduce O&M costs, and applications are gradually being migrated to the cloud. The multi-cloud strategy and multi-cloud services have become a trend in the industry. One-hop cloud access via Optical Transport Network (OTN) has been widely used in the industry and has become the mainstream choice for enterprises and cloud leased line services due to its various advantages, including high bandwidth, low latency, hard isolation, high reliability, and one-hop cloud access. However, it has always been difficult to manage services on OTN due to network configuration complexity and multi-vendor collaboration challenges. To overcome the difficulty, the system needs to be more intelligent so that a user without relative knowledge can express his/her demands.

### 5.3.11.2 Description of the Use Case

#### 5.3.11.2.1 Overview

Cloud Private Line (CPL) services connect cloud service users to edge or cloud data centres, and edge or cloud data centres to each other, with deterministic connection performance. They may represent point-to-point, point-to-multipoint, multipoint-to-point, or multipoint-to-multipoint connectivity service topologies, and may be implemented using connected or connection-oriented paradigm-supporting technologies. Data Centres may be operated by the CPL service customer, by the CPL services provider, by some other service provider(s), or by any combination of these. CPL service traffic consists in machine-to-machine data flows with a range of characteristics. Dynamically mass-customized CPL services are driven operationally by CPL service consumers, either semi-manually (e.g. through a user-facing provisioning portal) or more usefully, directly by consumer scheduling software systems. Intent is obviously a useful service-driving paradigm to support such capability.

#### 5.3.11.2.2 Motivation

In the cloud-network convergence scenario, users need enough bandwidth as well as computing resources. To be more specific, the user can express an intent of creating a cloud-network convergence service. This intent is then automatically fulfilled by provisioning the corresponding services and allocating the required resources. The already fulfilled intent can be modified by the user. The new intent can be automatically fulfilled by provisioning the corresponding services and allocating the required resources. The Intent-based system monitors the parameters of the cloud-network convergence service (e.g. bandwidth usage), and automatically triggers the closed-loop actions (e.g. increase max bandwidth) in order to guarantee the intent.

The user's intent, expressed in a natural language, can be translated by the ENI System. The use of the ENI System allows users to achieve their intents only through ordinary descriptions without understanding specific technical details. In this use case, the intent translation process in ENI System is formulated as a Question Answering (QA) problem, in which the key information can be accurately extracted from the sentences. The ENI System transforms an intent from a simple declarative form to a very complex and technology specific form in an automated way.

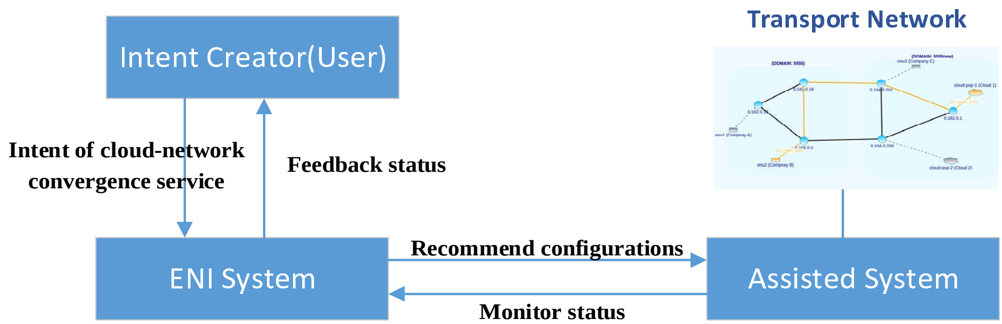


**Figure 5-30: A General Architecture of Cloud-network Convergence Services**

Figure 5-31 shows the framework of the use case mapping to the ENI reference architecture.

- SO: Provides end-to-end service orchestration.
- SDN-C: SDN controller for transport networks.
- DCAE: Performs data analytics of the telemetry data.
- Policy: Executes policies.
- AAI: Data store for network and service configurations, network resources, inventory, etc.

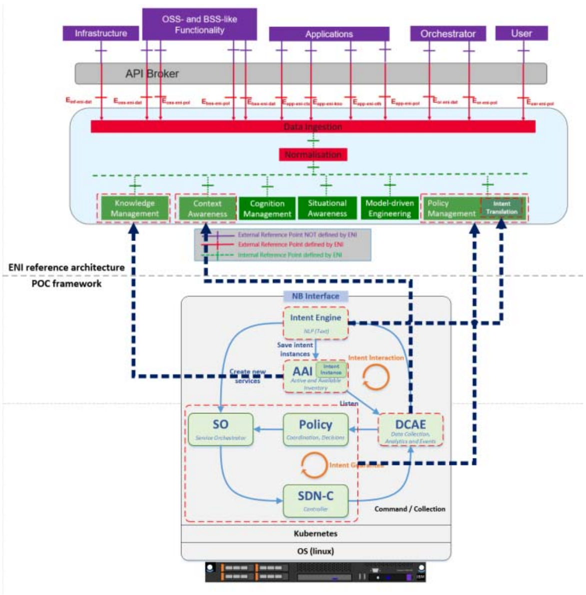


**Figure 5-31: Use case architecture mapped to ENI reference architecture**

### 5.3.11.2.3 Actors and Roles

User: the cloud-network convergence service consumer.

ENI System: receives and translates user's intent and make closed-loop decisions.

Network Operator: manages the network topology and computing resources.

### 5.3.11.2.4 Initial context configuration

Both topology of transmission network and information of computing resources have been input into ENI System.

Optimization policy has been input into ENI System.

The Natural Language Processing (NLP) algorithm is well trained and successfully runs in ENI System.

### 5.3.11.2.5 Triggering conditions

The ENI System receives user's intent using NLP techniques. For example, a simple user input describing customer's need in a natural language may be "I need a connection from company A to Cloud Two, with a bandwidth of 2 Gbps".

### 5.3.11.2.6 Operational flow of actions

1) ENI System receives the user's intent. The user expresses, in a natural language, an intent to connect one (or more) enterprises to one (or more) clouds, as well as his/her expectation for the quality of the service.

2) ENI System translates the user intent into more specific service intent parameters and sends the recommend configurations to the assisted system.

3) The assisted system receives the recommend configurations to fulfil the user's intent.

4) The user validates whether the demand is met.

5) ENI System continuously monitors the conditions of the network against the intent specification to ensure compliance with the intent.

6) If the network state cannot meet the user's intent, ENI System sends new recommend configurations to the assisted system.

#### 5.3.11.2.7 Post-conditions

The intent translation and intent life-cycle management have been achieved.

The intent assurance is achieved by the closed-loop automation.

## 5.3.12 Use Case #2-12: Energy Efficiency Evaluation for Artificial Intelligent Data Centres

### 5.3.12.1 Use case context

This use case involves evaluating the energy efficiency of Artificial Intelligence Data Centres (AIDCs). AIDCs are an infrastructure specifically designed to provide computing power, algorithms, and data services for Artificial Intelligence (AI), supporting the rapidly expanding AI application business. Unlike traditional Internet Data Centres (IDCs), AIDC exhibits distinct characteristics in hardware configurations, network architectures, heat dissipation technologies, and energy consumption costs. These key differences make existing evaluation methodologies for IDC inadequate for assessing AIDCs, such as inaccurate and non-comprehensive evaluations, conflict between evaluation opinions, etc. Consequently, specialized efficiency evaluation tailored for AIDCs are imperative.

### 5.3.12.2 Description of the Use Case

#### 5.3.12.2.1 Overview

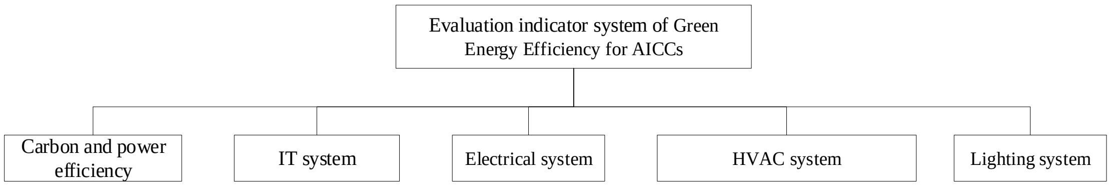


**Figure 5-32: Green evaluation index system of AICCs**

The green energy-saving of intelligent computing centres has emerged as a critical concern for operators. The current practices for AIDC green evaluation include:

- Power Usage Effectiveness (PUE) remains the primary evaluation metric. Compared to IDCs, AIDCs have a significantly increase on the energy consumption of IT equipment. Additionally, with the widespread adoption of liquid-cooling techniques, AIDCs generally have lower PUEs. As a result, PUE no longer provides an objective and comprehensive evaluation of the energy efficiency utilization level in AIDCs.
- The evaluation opinions of all parties are simply summarized and aggregated. Practical evaluation opinions may include conflicts, which prevent parties from reaching a consensus on the evaluation results.
- The energy efficiency of AIDCs often changes dynamically. Traditional static evaluation may cause evaluation results to lag behind actual situations.

#### 5.3.12.2.2 Motivation

The ENI system can be used for evaluating the green energy-saving level of AIDCs The following specific functions and objectives are proposed in the ENI system:

- Comprehensive assessment: ENI system can analyse and evaluate the energy efficiency and environmental sustainability of various components within an AIDC, including carbon and power efficiency, IT systems, electrical systems, Heating, Ventilation, and Air Conditioning (HVAC) systems, and lighting systems.
- Energy evaluation without conflict: ENI system can use the consensus decision evaluation model to evaluate the green and energy-saving performance of AIDC, which dynamically coordinates stakeholder feedback to ensure consensus in decision-making outcomes.

- Good interaction: ENI system achieves evaluation interaction by employing natural language processing models or Large Language Models (LLMs), which are fine-tuned using evaluation result samples to improve accuracy.

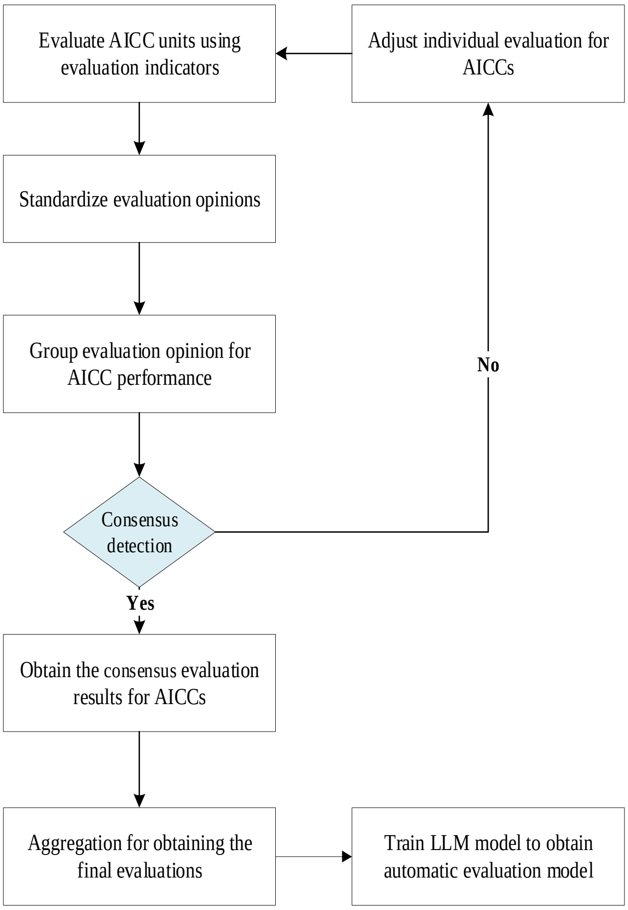


**Figure 5-33: Green Evaluation Process of AICCs**

#### 5.3.12.2.3 Actors and Roles

- ENI System: Evaluates the environmental sustainability and energy-saving status of AIDCs by integrating evaluator opinions and coordinating feedback from all stakeholders to reach consensus-based evaluation decisions.
- Operator: Manage the AIDCs and confirm the green and energy-saving levels of AIDCs.
- AIDC Infrastructure Management: Provides data on IT performance, electrical and systems, lighting efficiency, and other parameters for evaluation.
- Evaluation expert: Evaluate some qualitative indicators in the evaluation framework of AIDCs.

#### 5.3.12.2.4 Initial context configuration

- An evaluation indicator system covering carbon and power efficiency, IT system, electrical system, HVAC system, and lighting system.
- Consensus thresholds for reaching consensus on evaluation opinions.
- Assessment scale for qualitative index.
- AIDCs under construction or operation.

#### 5.3.12.2.5 Triggering conditions

The following trigger types associated with the ENI System may be identified:

- Periodic execution of green energy-saving and energy efficiency evaluation of AIDCs.
- The ENI System predicts that the AIDC energy consumption will exceed a certain threshold in a specific period.
- The ENI System determines to optimize the configuration of the AIDC and its subsystem settings.

#### 5.3.12.2.6 Operational flow of actions

1) The ENI System acquires all energy consumption data of AIDC and its subsystems (including IT, electrical, HVAC, and lighting systems) according to the evaluation index system.
2) The ENI System calculates quantitative evaluation indicators to assess performance.
3) The ENI System employs the evaluation of expert opinions to evaluate qualitative indicators.
4) The ENI System integrates individual evaluators' opinions into a group consensus using a consensus decision-making evaluation model.
5) The ENI System calculates the consensus level of the group opinion; if consensus is not achieved, individual evaluation opinions are adjusted iteratively until alignment is reached.
6) The ENI System aggregates the consensus opinions to generate a final evaluation decision.
7) The ENI System uses evaluation results as training samples to fine-tune natural language models (LLMs) to develop an automated energy-saving evaluation model for AIDC.
8) Based on the ENI System's evaluation outcomes, operators determine the AIDC's green certification level and formulate targeted optimization and maintenance strategies.

#### 5.3.12.2.7 Post-conditions

The operators receive detailed and actionable insights into AICC energy efficiency given by the ENI System, thus enabling targeted improvements to reduce energy consumption, lower carbon emissions, and enhance operational sustainability.

## 5.3.13 Use Case #2-13: Intelligent Satellite-Terrestrial Network Optimization

### 5.3.13.1 Use case context

This use case focuses on optimizing coverage and capacity in satellite-terrestrial integrated networks through AI/ML-driven dynamic resource allocation. As global communication demands surge - especially in aerial, maritime, and mountainous regions - traditional networks face coverage gaps and inefficient resource utilization. Satellite-terrestrial integration addresses these challenges but requires intelligent orchestration to achieve seamless connectivity and adaptive traffic steering.

### 5.3.13.2 Description of the Use Case

#### 5.3.13.2.1 Overview

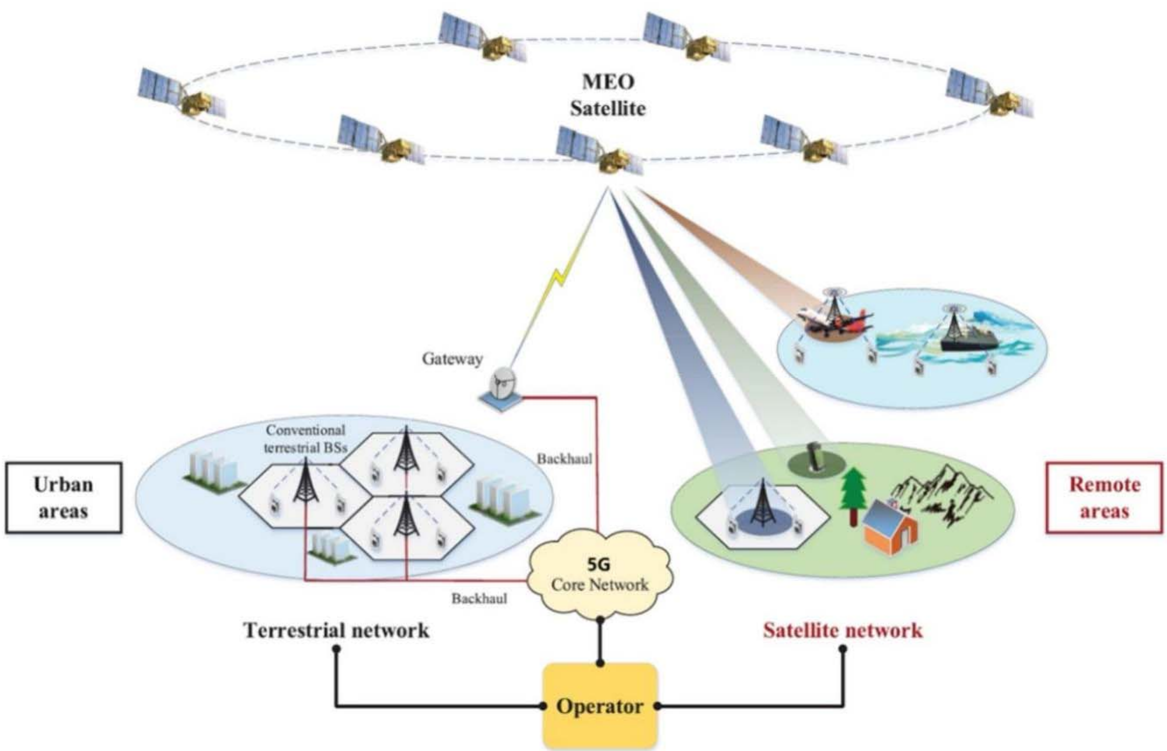


**Figure 5-34: Satellite-terrestrial integrated network architecture**

Current limitations include:

- **Coverage gaps:** Terrestrial networks fail in remote/emergency scenarios (e.g. earthquakes).
- **Resource inefficiency:** Low-orbit mega-constellations underutilize spectrum and hardware.
- **Static operations:** Fixed beam patterns cannot adapt to dynamic traffic demands.

This use case leverages AI to enable:

1) **Hand-and-Arm Architecture:** Unified management of satellite (MEO) and terrestrial (5G/B5G) access.
2) **Intelligent On-Demand Coverage:** Dynamic beam hopping and bandwidth allocation using multi-agent DRL.
3) **Adaptive Transport Protocols:** SMAC-layer encapsulation for end-to-end QoS across hybrid networks.

#### 5.3.13.2.2 Motivation

The ENI System provides knowledge on how to make full use of the limited satellite resources to match the traffic demands efficiently. The Multi-Beam Satellites (MBS) play an important role to solve this problem, which can generate a large number of beams and flexibly allocate these beam resources to improve communication coverage and transmission capacity. Traffic steering allows to optimize the dynamic beam hopping and resource utilization, users' service perception and the system throughput are jointly optimized by directing the traffic to the beam that provides the best performance. However, this process needs to be tied closely together with mobility management, which assures optimized mobility performance, for example a reasonable complexity of the algorithm and high efficiency in terms of time-varying traffic requests. This use case describes a dynamic beam pattern and bandwidth allocation scheme based on Deep Reinforcement Learning (DRL), which can fully exploit three degrees of freedom (i.e. time, space and frequency) of the beam, overall benefits including:

- **Real-time traffic steering:** Redirects users to optimal beams (satellite/terrestrial) based on latency/capacity needs.
- **Emergency resilience:** Automatically shifts traffic to satellite networks during terrestrial failures.
- **Resource optimization:** Increases spectrum efficiency by 40 %+ through DRL-based beam hopping.

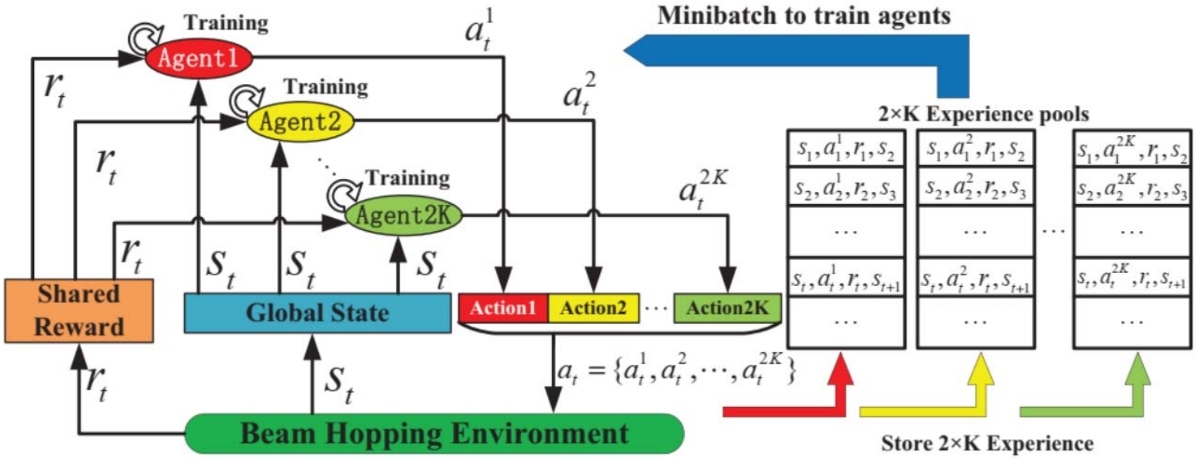


**Figure 5-35: Deep Reinforcement Learning (DRL) optimization workflow**

### 5.3.13.2.3 Actors and Roles

1) **ENI System:**
   - Collects network state data (traffic load, link quality).
   - Runs DRL models for beam/resource allocation.
   - Orchestrates satellite-terrestrial handovers.
2) **Network Operation Control Centre (NOCC):**
   - Deploys AI policies to satellite/ground stations.
   - Monitors global network performance.
3) **User Equipment (UE):**
   - Reports location/service demands.
   - Executes seamless access switching.
4) **Satellite/Gateway Nodes:**
   - Implement adaptive coding/modulation (LDPC).
   - Forward data at SMAC/network layers.

### 5.3.13.2.4 Initial context configuration

- **Network topology:** Integrated MEO satellites + 5G/B5G base stations.
- **DRL model:** Pre-trained for beam hopping/bandwidth allocation.
- **Consensus thresholds:** For handover decisions (e.g. latency > 50 ms triggers satellite fallback).
- **Protocol stack:** SMAC encapsulation for hybrid IP/CCSDS compatibility.

#### 5.3.13.2.5 Triggering conditions

- Periodic network optimization cycles (e.g. every 15 mins).
- UE density spikes > 200 % in a region (e.g. disaster zones).
- Terrestrial link failure predicted/occurred.
- Latency exceeds SLA thresholds (e.g. >100 ms for emergency services).

#### 5.3.13.2.6 Operational flow of actions

1) ENI System collects real-time data: UE locations, traffic demands, link quality.
2) DRL agents compute optimal beam patterns/bandwidth allocation:
    - **Beam Agent:** Adjusts illumination direction per coverage gap.
    - **Bandwidth Agent:** Allocates spectrum per traffic priority.
3) NOCC deploys policies to satellites/terrestrial nodes.
4) SMAC-layer encapsulation ensures QoS during satellite-terrestrial handovers.
5) Adaptive LDPC coding dynamically adjusts to channel conditions.
6) UE seamlessly switches access points (satellite ↔ terrestrial).
7) ENI System evaluates KPIs (throughput, latency) and retrains DRL models.

#### 5.3.13.2.7 Post-conditions

- Operators receive optimization reports:
- UEs experience < 20 ms handover latency during access switching.
- Network maintains > 99,999 % availability in disaster scenarios.

## 5.3.14 Use Case #2-14: Space-Ground Cooperative Network Slicing

### 5.3.14.1 Use Case Context

Space-ground cooperative networks integrate terrestrial mobile communication networks (e.g. 5G) and satellite networks to deliver seamless global connectivity. However, fundamental differences exist in **network slicing protocols** between these heterogeneous networks:

- **Divergent slicing methods:** Classification, quantity, and construction of slices differ.
- **Incompatible control planes:** Slices cannot interconnect directly due to protocol mismatches.
- **Dynamic service requirements:** Real-time voice, data, control signalling, and short messages demand varying latency, bandwidth, and security levels.

Existing network slicing frameworks lack adaptation mechanisms for cross-domain interoperability, limiting end-to-end service guarantees in integrated space-ground environments.

### 5.3.14.2 Description of the Use Case

#### 5.3.14.2.1 Overview

The space-ground cooperative network slicing architecture is shown in Figure 5-36. This use case includes the programmable slicing gateway and the space-ground cooperative slicing control system between the terrestrial mobile communication network and the satellite network. The programmable slicing gateway is the transit channel for the slicing service data flows of space-ground cooperative network. With definable Message parsing, processing and forwarding capabilities, the gateway accurately identifies and controls slicing services, and achieves the data mapping between slices according to the configuration policy provided by the control system. The service consistency and continuity of service data can be ensured in space-ground cooperative network slicing and realize the adaptation of heterogeneous network slices. The space-ground cooperative slicing control system interacts with the space-ground network slicing control planes respectively, to open the slicing session channel between the space and ground network cooperatively. The control system can optimize the matching mode of service traffic and network resources, and intelligently generate the configuration policy of the programmable slicing gateway, thus improving the end-to-end quality of slicing service in space-ground cooperative network

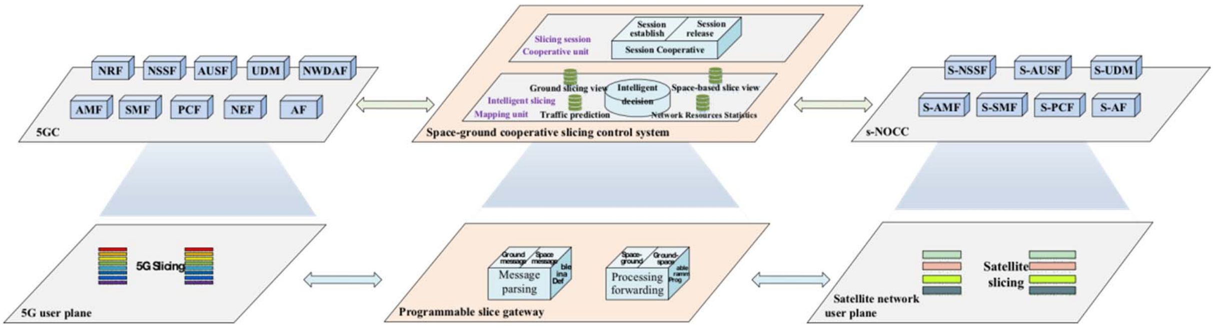


**Figure 5-36: Space-Ground Cooperative Network Slicing Architecture**

#### 5.3.14.2.2 Motivation

The ENI System enables intelligent interconnection of terrestrial and satellite network slices by:

1) **Data-plane adaptation:** Mapping VLAN/IP identifiers across domains for unified service control.

2) **Control-plane coordination:** Exchanging slicing policies between 5GC and SNOCC to optimize end-to-end QoS.

3) **AI-driven resource prediction:** Using Graph Convolutional Networks (GCN) and Gated Recurrent Units (GRU) to forecast traffic loads and allocate resources proactively.

In the space-ground cooperative network, there are many types of service requirements and wide coverage. The performance requirements of services such as real-time voice, data transmission, control signalling, and short message have different performance requirements, and the service delay, bandwidth, and security requirements all change in real time. To meet the differentiated application requirements of wide-area information networks, the space-ground cooperative network needs to dynamically construct differentiated network slices involving different service characteristics, accurately match the resource requirements of different service data, and realize multi-service converged application. An intelligent slice mapping mechanism based on spatial-temporal correlation. Through traffic prediction to establish the prediction model of resource demand of network services, it can respond to the service characteristics and the transformation of access node in real time. Thus, the slices of network resources can be matched as needed with the wildly fluctuating traffic in the space-ground cooperative network.

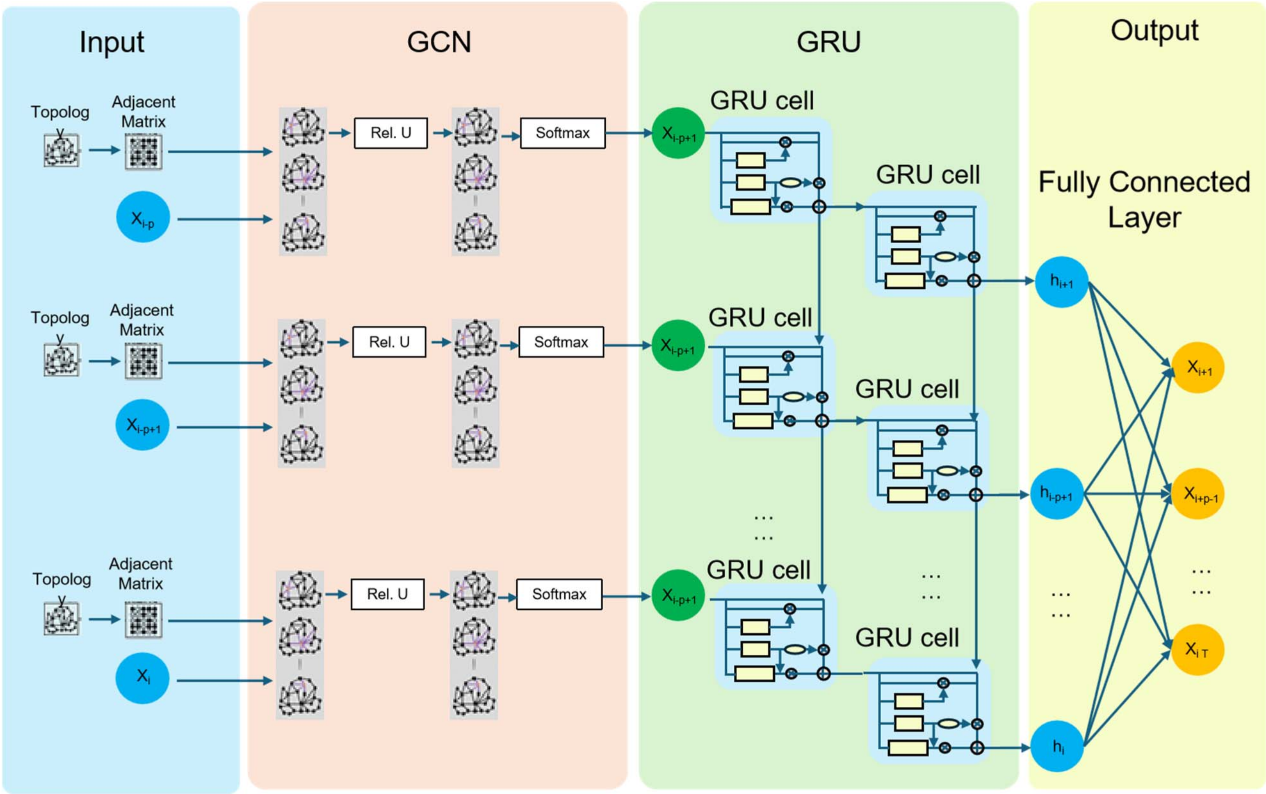


**Figure 5-37: Intelligent slice mapping based on spatial-temporal correlation**

Figure 5-37 shows the intelligent slice mapping diagram based on spatial-temporal correlation. Graph Convolutional Network (GCN) and Gated Recurrent Unit (GRU) are used to extract the temporal-spatial characteristics of the historical traffic load of each node in the space-ground cooperative network slicing, which is to provide a decision basis for slice mapping. Firstly, the network topological features are captured by GCN to obtain the spatial dependence. Secondly, the dynamic changes of node attributes are captured by GRU to obtain the local time trend of traffic load. Finally, the multi-output fully connected layer of artificial neural network is used to realize the transformation from traffic load to resource demand, and output the predicted result. The system monitors the network resource status in real time; slices are allocated network resources based on the predicted results of slicing service requirements to complete slicing adaptation decisions and ensure the service requirements of the business.

### 5.3.14.2.3 Actors and Roles

1) **ENI System:**
   - Orchestrates slice adaptation policies.
   - Predicts resource demands using GCN+GRU models.
   - Mediates control-plane signalling between 5GC and SNOCC.
2) **Programmable Slicing Gateway:**
   - Translates data-plane identifiers (e.g. VLAN/IP) across networks.
   - Enforces traffic policies based on ENI directives.
3) **Operator:**
   - Manages end-to-end slice performance and SLA compliance.
4) **UE (User Equipment):**
   - Initiates PDU sessions requiring cross-domain slicing.

#### 5.3.14.2.4 Initial Context Configuration

- **Network Infrastructure:**
  - Terrestrial 5G network with network slicing capability.
  - Satellite network with slicing support.
- **ENI Components:**
  - Slicing mapping management module.
  - Session collaboration processing engine.
  - GCN+GRU prediction model for traffic forecasting.

#### 5.3.14.2.5 Triggering Conditions

The following trigger conditions associated with the ENI System may be identified:

- New PDU session request from UE requiring space-ground connectivity.
- Resource utilization exceeds thresholds in terrestrial/satellite segments.
- Service QoS degradation detected by ENI monitoring.

#### 5.3.14.2.6 Operational Flow of Actions

1) **Session Initiation:**
   - UE requests PDU session via terrestrial 5G network.
   - 5GC selects SMF/UPF and notifies ENI session collaboration unit.
2) **Slice Mapping:**
   - ENI maps terrestrial session to satellite slice using:
     - **Adaptation rules** (e.g. service type, QoS class).
     - **Resource prediction** from GCN+GRU model.
3) **Satellite Session Setup:**
   - ENI directs SNOCC to establish/modify space-based session.
   - Satellite SMF/UPF configured dynamically.
4) **Data Path Activation:**
   - Programmable gateway bridges data planes (VLAN/IP translation).
   - End-to-end slice channel opened (UE → 5G → Satellite → DN).
5) **Dynamic Optimization:**
   - ENI monitors performance, adjusts slices via control-plane coordination.

#### 5.3.14.2.7 Post-Conditions

- UE achieves seamless access to Data Network (DN) via integrated slice.
- Operators gain unified visibility into cross-domain slice performance.
- Resource utilization optimized through predictive scaling.

# 5.4 Service Orchestration and Management

## 5.4.1 Use Case #3-1: Context-aware VoLTE Service Experience Optimization

### 5.4.1.1 Use case context

As the mobile network evolves to 4G and 5G, an all-IP network will provide high definition voice transmission. VoLTE, namely Voice over LTE, it is an IP data transmission technology, which does not need a 2G or 3G network. In VoLTE, all business bears on 4G network, and can realize the unification of data and voice services using the same network. As a result, the 4G and 5G networks not only provide high-speed data services, but also provide high-quality audio and video services, the latter achieved via the use of the VoLTE technology.

### 5.4.1.2 Description of the use case

#### 5.4.1.2.1 Overview

Conventionally, operators rely on field or drive tests to determine the Reference Signal Received Power (RSRP) for smooth VoLTE service experience. However, such tests are not adequate and efficient enough to support increased quality and capacity demands, because it is difficult for a human expert to do thorough tests everywhere and every day, and such tests are also error prone. Moreover, VoLTE RSRP is configured in a RAN statically, which consequently results in VoLTE call drop or handover to 2/3G unnecessarily.

VoLTE operation requires the RSRP to be adaptively configured to meet the changing context.

#### 5.4.1.2.2 Motivation

It has been observed that the RSRP configuration is relevant to many factors, such as mobile terminal type, user location, voice codec, traffic load, time of day, etc. These factors may change frequently, which makes it very difficult to find a deterministic function to model this dynamism. Therefore, an intelligent entity (i.e. the ENI System) can be used to collect the relevant data, use one or more AI mechanisms to analyse the data, and then predict the proper RSRP. When an ENI System sends the predicted RSRP to operations system, the VoLTE RSRP will be adjusted according to the different VoLTE service information. Furthermore, RAN monitors the fulfilment of the QoS requirements. If the QoS requirements are no longer fulfilled, a notification is sent to the ENI System (and/or OSS), which will take actions to either adjust to lower QoS requirements or to terminate the service. With this control loop enabled by ENI System, the VoLTE service experience can be optimized adaptively and responsively in contrast to time-consuming and inefficient manual field tests.

An illustration of the Interoperability between VoLTE and 2G and 3G is given in Figure 5-38.

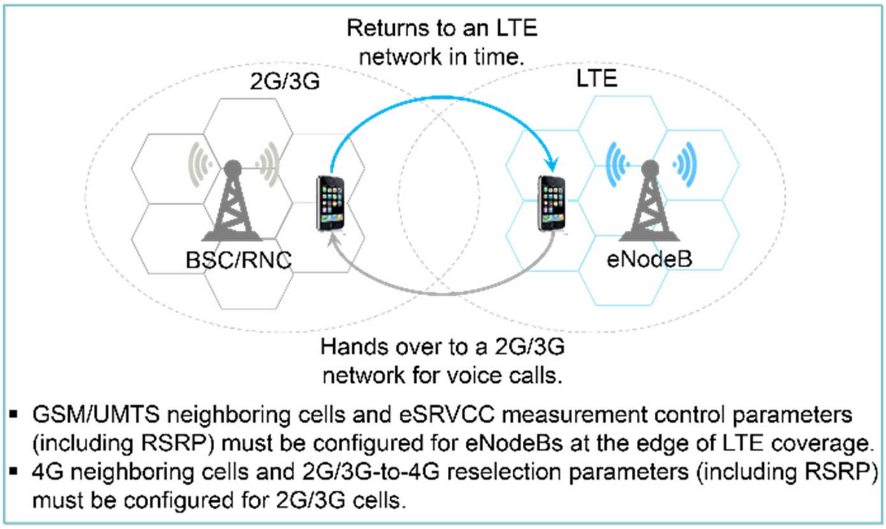


**Figure 5-38: Interoperability between VoLTE and 2G and 3G**

### 5.4.1.2.3 Actors and Roles

- Radio Access Network: monitors whether the QoS requirements are met and notifies the ENI System.
- ENI Engine: collects and analyses VoLTE service information and contextual data, and dynamically determines if the RSRP should be reconfigured.
- Operations System: as defined by ETSI TS 132 101 [i.3], adjusts VoLTE RSRP according to the policies generated by the ENI System.
- Network Administrator: responsible for configuring the network.

### 5.4.1.2.4 Initial context configuration

- The VoLTE RSRP was configured in RAN statically.
- The ENI System has learned how to configure the RSRP in order to ensure the VoLTE service experience.

### 5.4.1.2.5 Triggering conditions

The current VoLTE RSRP does not meet the VoLTE continuity coverage requirement.

### 5.4.1.2.6 Operational flow of actions

1) ENI System collects and analyses VoLTE service information (and any other necessary information, such as contextual data).
2) ENI System determines what the RSRP should be according to the current VoLTE service information.
3) Operations system reconfigures the RSRP according to the output of the ENI System.
4) RAN monitors whether the QoS requirements are met; if not, it notifies the ENI System.
5) ENI System recommends appropriate changes (e.g. rollback the RSRP, or make other configuration changes) to meet the QoS requirements.
6) Operations system implements recommended changes.

### 5.4.1.2.7 Post-conditions

- The VoLTE RSRP dynamically adjusts according to the changing network environment.
- The VoLTE service experience was optimized adaptively.

### 5.4.1.3 Mapping to ENI reference architecture

#### 5.4.1.3.1 Functional blocks

The functional blocks for context-aware VoLTE Service experience using AI are shown in Figure 5-39.

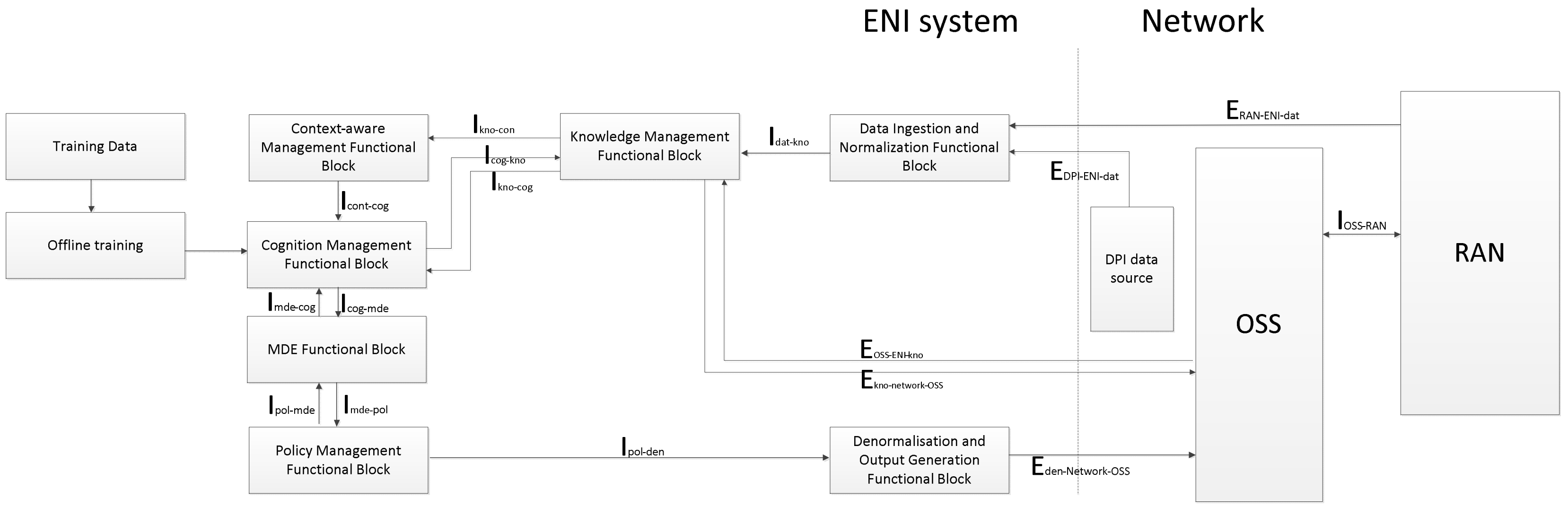


**Figure 5-39: Mapping to ENI reference architecture for Context-aware VoLTE Service Experience**

For training data, it can be pre-prepared one or the one from RAN and DPI data Source.

The "Data ingestion and Normalization Functional block" collects DPI data from DPI data source (e.g. core network or data centre) and radio parameters from RAN. These data are filtered and normalized accordingly.

The "Knowledge Management Functional Block" is used to store knowledges and generate inferences. There may be some interactions between "Knowledge Management Functional Block" and OSS on feature set update which will be used for modelling iteration in "Cognition Management Functional Block".

The "Context-aware Management Functional Block" is used to generate context information of users.

The "Cognition Management Functional Block" utilizes data as well as inferences to generate predictions based on modelling provided by offline training. For the modelling in "Cognition Management Functional block", it will be iterated periodically based on the data from DPI Data Source and RAN.

The "Policy Management Functional Block" makes the polices and inputs decisions to "Denormalization and Output Generation Functional Block", which generates execution command to operating system like OSS. Next, OSS would send the command to RAN for adjusting radio parameters like transmission power and so on.

#### 5.4.1.3.2 Reference Points

$E_{RAN\text{-}ENI\text{-}dat}$ defines data exchange between RAN and ENI System.

$E_{Data\text{-}DPI}$ defines data exchange between DPI data source (e.g. certain entity in core network, data centre) and ENI System.

$I_{dat\text{-}kno}$ defines internal Reference Point between "Data ingestion and Normalization Functional block" and "Knowledge Management Functional Block".

$I_{kno\text{-}con}$ defines internal Reference Point between "Knowledge Management Functional Block" and "Context-aware Management Functional Block".

$I_{cont\text{-}cog}$ defines internal Reference Point between "Context-aware Management Functional Block" and "Cognition Management Functional Block".

$I_{cog\text{-}kno}$ and $I_{kno\text{-}cog}$ define internal Reference Point between "Cognition Management Functional Block" and "Knowledge Management Functional Block".

$I_{cog\text{-}mde}$ and $I_{mde\text{-}cog}$ define internal Reference Point between "MDE Functional Block" and "Cognition Management Functional Block".

$I_{pol\text{-}mde}$ and $I_{mde\text{-}pol}$ define internal Reference Point between "MDE Functional Block" and "Policy Management Functional Block".

$I_{pol\text{-}den}$ defines internal Reference Point between "Policy Management Functional Block" and "Denormalization and Output Generation Functional Block".

$E_{den\text{-}network\text{-}OSS}$ defines data exchange between "Denormalization and Output Functional Generation Block" and "OSS".

$E_{kno\text{-}network\text{-}oss}$ and $E_{oss\text{-}ENI\text{-}kno}$ define data exchange between "Knowledge Management Functional Block" and OSS.

### 5.4.1.3.3 Flow of information

The flow of information is shown in Figure 5-40.

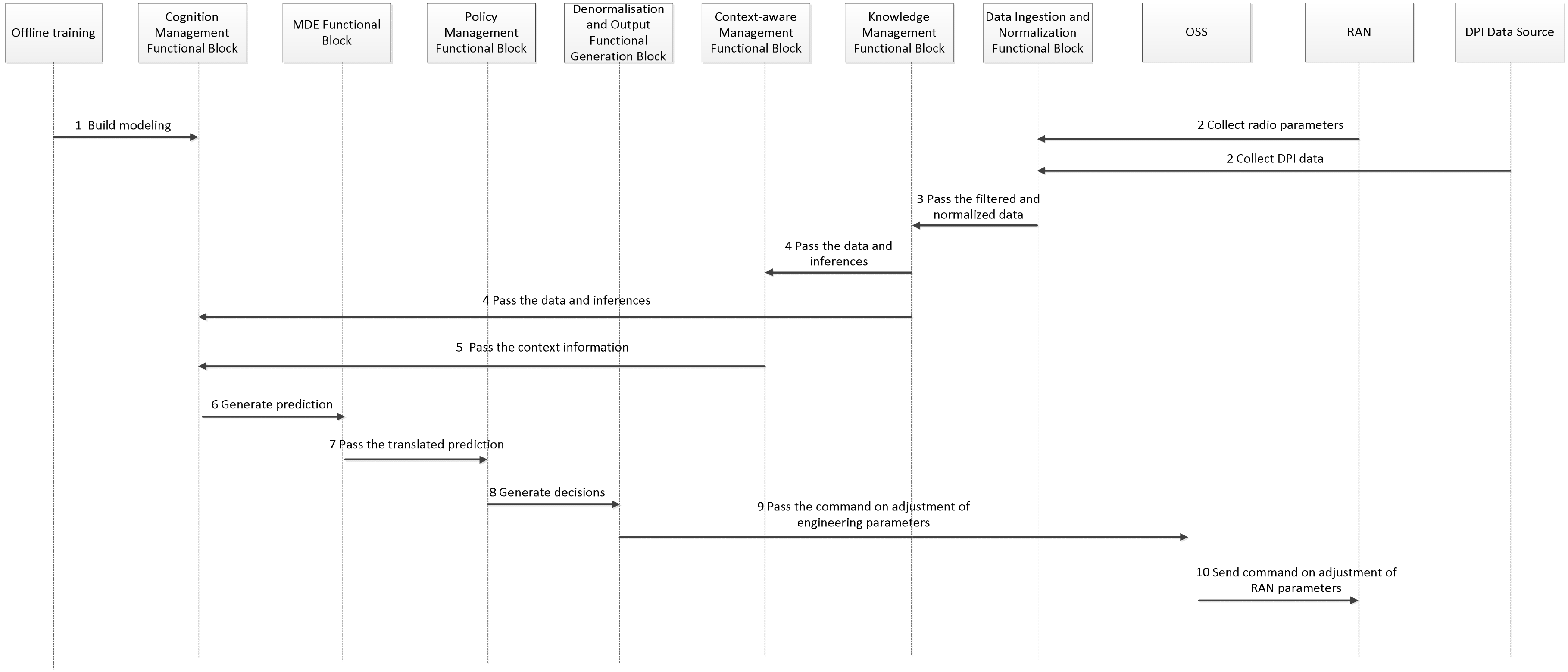


**Figure 5-40: Flow of information for Context-aware VoLTE Service Experience**

Step 1: Is about building modelling via offline training.

Step 2: Is that "Data ingestion and Normalization Functional block" collects DPI data from DPI data source and radio parameters from RAN.

Step 3: Is that the filtered and normalized data is passed to "Knowledge Management Functional Block".

Step 4: The data and inferences are passed to "Context-aware Management Functional Block" and "Cognition Management Functional Block".

Step 5: The context information generated from "Context-aware Management Functional Block" is passed to "Cognition Management Functional Block".

Step 6: The prediction is generated in "Cognition Management Functional Block" and passed to "MDE Functional Block".

Step 7: The translated predictions are generated and passed to "Policy Management Functional Block".

Step 8: "Policy Management Functional Block" generates decisions to "Denormalization Functional" and "Output Generation Block".

Step 9: Command on adjustment of engineering parameters is passed to operating system like OSS.

Step 10: Command on adjustment of RAN parameters like transmission power is passed to RAN.

There would be some steps additionally on model iteration between "Knowledge Management Functional Block" and "Cognition Management Functional Block".

## 5.4.2 Use Case #3-2: Intelligent network slicing management

### 5.4.2.1 Use case context

The concept of network slicing has been introduced by the NGMN 5G whitepaper [i.1], which enables multiple logical self-contained networks to use a common physical infrastructure platform, enabling a flexible stakeholder ecosystem that allows technical and business innovation integrating network and cloud resources into a programmable, software-oriented network environment. From the perspective of 3GPP TR 23.799 [i.2], network slicing enables operators to create networks customized to provide optimized solutions for different market scenarios which demands diverse requirements, e.g. in the areas of functionality, performance and isolation.

In particular, network slicing can be used to support very diverse requirements imposed by different type of services, e.g. IoT services as well as 5G services as eMBB/mMTC/uRLLC in a single physical network (universal and heterogeneous) by using flexibility and scalability.

### 5.4.2.2 Description of the use case

#### 5.4.2.2.1 Overview

The realization of the network slice concept is accomplished using Network Slice Instances (NSIs), see Figure 5-40. An NSI is an instance of a logical representation of Network Function(s) and corresponding resource requirements necessary to provide the required End-to-End (E2E) telecommunication services and network capabilities. An NSI typically covers multiple technical domains, which includes terminal, access network, transport network and core network, as well as DC domain that can host third-party applications from vertical industries.

In the early stage of network slicing deployment, there could be only a few NSIs. The deployment may occur in a semi-automatic mode. As the number of NSIs increases and scenarios, such as, dynamic instantiation of NSIs or runtime adaptation of the deployed NSI emerge, more advanced technologies will be desired to support network slicing and its further evolution, see e.g. 3GPP TR 23.799 [i.2]. Specifically, management functions could become real-time, implying that the difference between management and control will gradually disappear. Some management functions will be tightly integrated with the NSIs as well as the network infrastructure.

This use case is applying the ENI System to enhance and optimize the network slice management and control operations.

Other possible scenarios are network slicing where an operator can dynamically change a given slice resource reservation, considering that each slice may be assigned for a specific type of service or service class. Moreover, hybrid scenarios, where network slicing and resource sharing are applied at the same time are also envisaged. These scenarios are not addressed in this release.

#### 5.4.2.2.2 Motivation

In current networks, technical domains are normally coordinated via centralized network management system. In 5G, performing real-time cross-domain coordination through distributed lower layer such as control plane would be possible, with potentially unified control logic of different domains.

Advanced automation and AI algorithms can be applied in a unified, "holistic" network manner, which could be scalable and flexible, and which might then achieve runtime deployment and adaptation of NSIs.

In the context of ENI, the ENI System can be used to enhance and optimize the network slice management and control operations.

Figure 5-41 shows an example of two network slice instances that are being created using the 3GPP 5G infrastructure. Note that Figure 5-41 shows functions used in Network Slicing for the interaction purposes of ENI. In a production environment other capabilities are available.

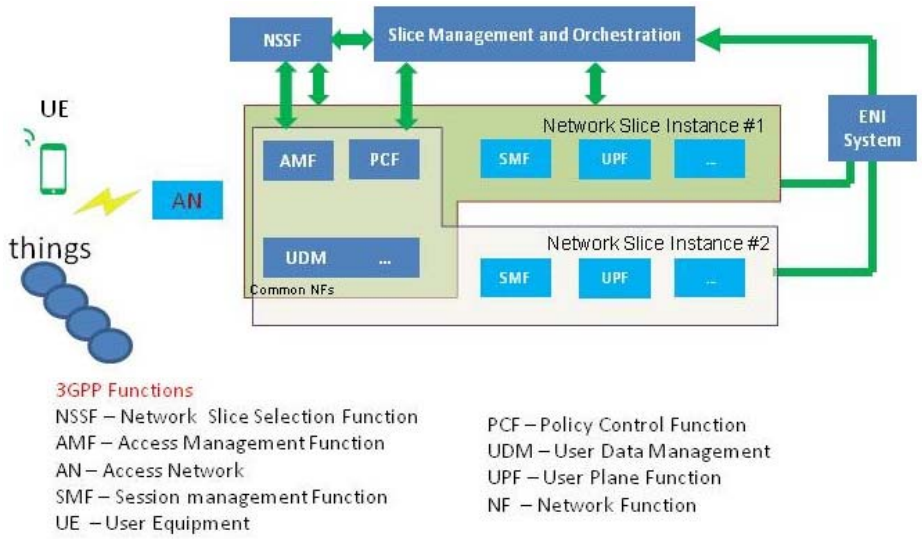


**Figure 5-41: Example of a network slicing management and orchestration scenario**

#### 5.4.2.2.3 Actors and Roles

- Slice Management and Orchestration: entity that manages and orchestrates the life cycle of slices. Note that this entity is administrated by a network operator.
- Network Infrastructure: infrastructure used to create and maintain the slice.
- ENI System: system solution used to assist and optimize the operation of the Slice Management and Orchestration entity.
- IoT devices: devices that can be represented as Things and can take part in the operation of the end to end slice.
- UE: any device, e.g. Smart phone that is using the 3GPP cellular technology and can take part in the operation of the end to end slice.

#### 5.4.2.2.4 Initial context configuration

The slices are created and configured; The ENI System through AI and machine learning capabilities is learning the configuration of the applied slices and as well the traffic patterns used by each of these slices; moreover, the ENI System measures the utilization of the network and other relevant parameters that define the satisfactorily operation of each slice and that needs to conform to the Service and Network KPIs requested by the operators.

#### 5.4.2.2.5 Triggering conditions

Triggering conditions that may affect resources in transport network plane include but not limited to, the following situations, e.g. input traffic adjustment, network health deterioration, routing path switching, bandwidth overutilization, packet loss rate increase, latency increate.

When associated with assisted networks that use network slicing, the ENI System should actively and passively monitor related triggering conditions for anomaly detection (fault, error and unusual behaviour), prevention and fast recovery. Therefore, the ENI System should be distributed in the whole life-cycle management of transport network slicing, including probes deployment, status collection and analysis, decision making, verification and delivery.

When the ENI System concludes that the measured parameters associated with the operation of each slice do not conform to the Service and Network KPIs requested by the operators, then the ENI System notifies the Slice Management and Orchestration entity about this event.

#### 5.4.2.2.6 Operational flow of actions

ENI System applying analysis and machine learning technologies can be used to enhance and optimize the network slice management and control operations and to assist the Slice Management and Orchestration entity to resolve any abnormal operation of each slice; some of the ENI System activities are listed below:

1) ENI System analyses the collected data associated to e.g. network topology, network traffic load, service characteristic, user location and movement, VNF type and placement constraints, infrastructure capability and resource usage, etc.

2) While performing the monitor process, the ENI System is aware of the existence of triggering conditions for e.g. network health, bandwidth utilization, loss rate, latency, and performs anomaly (fault, error and unusual behaviour) demarcation.

3) ENI System produces a proper context aware policy to indicate to the network slice management entity when, where and how to place or adjust the network slice instance (e.g. reconfiguration, scale-in, scale-out, change the template of the network slice instance), including the network slice functions and their configurations, in order to achieve an optimized resource utilization according to the possible change of service requirements and/or the network environment.

4) ENI System provides possible root causes analysis report.

#### 5.4.2.2.7 Post-conditions

The abnormal operation of the slice is resolved and the slice performs and conforms according to the Service and Network KPIs requested by Operators.

## 5.4.3 Use Case #3-3: Intelligent carrier-managed SD-WAN

### 5.4.3.1 Use case context

Software-Defined Wide Area Network (SD-WAN) is an approach of designing and deploying an enterprise WAN that uses SDN to determine the most effective way to route traffic to remote locations. SD-WAN allows enterprises to reduce the cost of expensive leased Multi-Protocol Label Switching (MPLS) circuits by sending lower priority, less-sensitive data over cheaper public Internet connections, as well as by reserving private links for mission-critical or latency-sensitive traffic like VoIP.

With the carrier managed SD-WAN service, enterprises can free up from network management and monitoring, and focus more on the business itself.

### 5.4.3.2 Description of the use case

#### 5.4.3.2.1 Overview

The enterprise has a hybrid Wireless Access Network (WAN), which includes the high quality private MPLS circuit, as well as economic public Internet access and wireless access for last resort. The devices at the edge of the enterprise network are managed by a network supervisory controller. Furthermore:

- Each enterprise can have customized SD-WAN services through the web portal via an Application Programming Interface (API). Enterprises customize their own networked experience based on their business needs, such as cost priority, quality priority, or cost-effective priority. Similarly, enterprises may customize their communication service levels assurance requirements based on the needs demanded by their enterprise communications applications. Therefore, the network supervisory controller may adapt intelligent policies according to the preceding enterprise needs.

- Different WAN traffic will be handled by the intelligent WAN policies, in order to make the usage of bandwidth resource more efficient. These policies steer traffic depending on WAN resource capabilities and according to application performance demands. This dynamic process saves time and optimizes resources.

- As conditions on the network change, the network supervisory controller may detect them through monitoring mechanisms, and adapt traffic routing through intelligent WAN policies, which automatically prioritize critical applications while dynamically suppressing non-critical ones.

As the logic of switching across different circuits becomes more complex, network intelligence can help Network Administrators to better manage the SD-WAN service.

The current Use Case is further described by the following set of components and features.

#### 5.4.3.2.2 Motivation

The main advantage of using the ENI System is that, by using AI and context-awareness, it can monitor the network and help enterprises to optimize their services and resources, hence allowing enterprises to focus more on their businesses.

A further advantage of applying the ENI System to SD-WAN services is that it can expose an Intent based interface that allows enterprises to customize their service using natural language with a terminology that is familiar to them.

EXAMPLES: Such Intent policies in SD-WAN could be:

- "All personal devices will access Internet using the Overlay connection".
- "All traffic belonging to Users in Administrative group will go through the firewall".
- "All policies applied to personal devices will also apply to guest devices".
- "John is part of the Administrative group".
- "Personal devices cannot access social media platforms more than one hour per day".

Additionally, the ENI System may also use AI methods in order to optimize the service and suggest policies adaptations to Network Administrators, e.g.:

- Scenario 1: Guest devices have been using a large part of the bandwidth with video streaming. The ENI System may trigger an alarm identifying the need to adapt an existing or create a new one to lower the priority for guest devices.

- Scenario 2: Every last day of the month, company A backups all data to a server in the central office. The ENI System could suggest a periodic rule so that all traffic coming from registered devices and destined to the backup server bypasses the firewall (preventing unnecessary use of resources and speeding up the backup process).

Figure 5-42 provides a pictorial representation of the Use Case described.

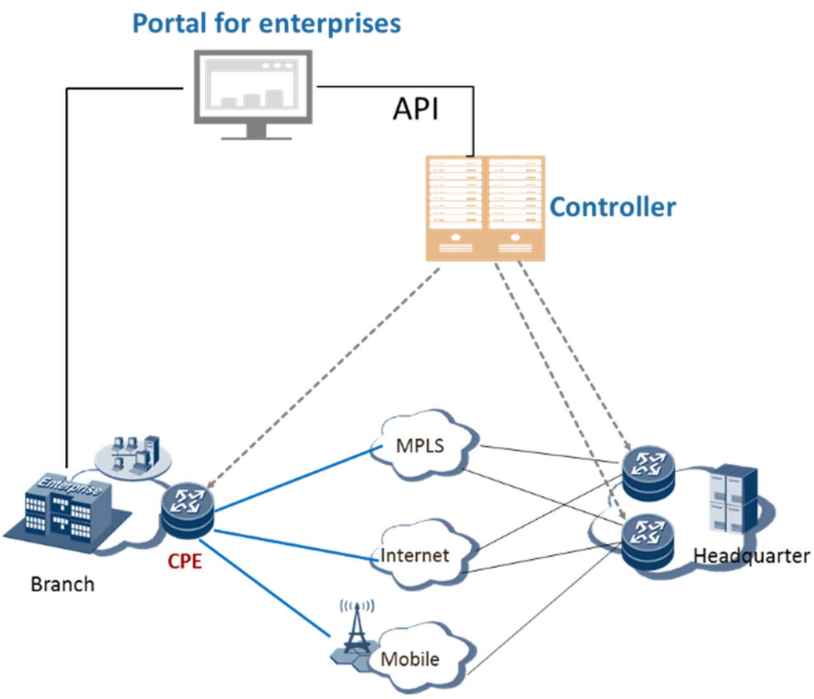


**Figure 5-42: Intelligent carrier managed SD-WAN service**

### 5.4.3.2.3 Actors and Roles

The presence of the following actors/entities as well as their associated roles are envisaged in the current Use Case:

- Carrier: Entity that provides MPLS/Internet connection to enterprises, as well as access to a web portal via one API in order to allow the enterprise to customize the services/connections.
- Network Administrator: Entity/person that configures the network topology and builds the WAN policies.
- ENI System: Entity that receives the intent policies from Network Administrators, monitors the services/connections and translates the intent policies into instructions/configurations to be enforced and executed by network devices.

Additionally, the ENI System may also optimize the services/connections and suggest policies, e.g. under the scenarios above identified.

### 5.4.3.2.4 Initial context configuration

The Network Administrator, according to enterprise's needs, configures the services/connections policies, e.g. Intent policies, for each application, which requires a large amount of backup work.

### 5.4.3.2.5 Triggering conditions

A new traffic pattern is detected by the ENI System as a new application and due to the collected information it is recognized as a potential new social network. The new traffic pattern is causing some optimization problems with Internet access.

### 5.4.3.2.6 Operational flow of actions

The following sequence of actions may be identified after the occurrence of the trigger:

1) The ENI System requests a confirmation from the Network Administrator to treat the new application as a social network.

2) The Network Administrator confirms that the new application is indeed a social network.

3) The ENI System identifies related policies and analyses past history looking for:

   a) Policy violations before the identification of the new application.

   b) Impact of the new application on network optimization.

4) The ENI System suggests the Network Administrator some changes to existing policies:

   a) alter existing policies e.g. "Personal devices cannot access social media platforms more than one hour per day" to "Personal devices cannot access social networks more than one hour per day", in order to include both Facebook and the new application; or

   b) add new policy e.g. "All Social Network traffic have the lowest priority when accessing the Internet" to mitigate the access problems cause in the network by the new application when accessing the Internet.

5) The Network Administrator acknowledges the new policies and confirms the changes to the ENI System.

6) After confirmation, the ENI System provokes the enforcement of the new changes in policies by configuring appropriate network components.

##### 5.4.3.2.7 Post-conditions

The new application is categorized and customized configuration is registered in the portal. The network traffic generated by different applications is routed to different paths according to the services/connections policies and application configuration.

After the changes to the network policies and devices configuration, the SD-WAN service is running under optimal conditions.

### 5.4.4 Use Case #3-4: Intelligent caching based on prediction of content popularity

#### 5.4.4.1 Use case context

With the development of communication technologies, more and more user terminals will access the mobile cellular network, which will lead to explosive growth of network data traffic and content diversity. How to ensure that user terminals can be fast, effective and secure getting what users want is a big challenge for mobile networks. In today's cellular networks, a large amount of mobile traffic is generated by social and mobile applications. A significant portion of such traffic consists of popular contents that are repeatedly transmitted to mobile users and unnecessarily consume extra backhaul bandwidth and resource of radio access networks. Therefore, Mobile Caching (MC) has received significant attentions as an efficient approach to improve spectral efficiency and reduce backhaul load of mobile networks by bringing contents near mobile users. It is very important to improve efficiency of caching content management in MC due to limited cache storage resources within mobile cellular networks. The cache decision by content popularity prediction can improve the cache hit ratio of cached content, reduce the backhaul bandwidth cost of cellular networks, and reduce the content access delay of mobile users.

#### 5.4.4.2 Description of the use case

##### 5.4.4.2.1 Overview

In this use case, the AI-based MC is implemented in a mobile edge cache equipment, which includes but is not limited to a core network gateway, a base station, a user equipment. The AI-based MC (AI-MC) is divided into four modules: AI prediction module cache decision module; control module; and information collection module.

The prediction module, which is aligned with the Knowledge Management function block in ETSI GS ENI 005 [3], is used to predict the popular content. It is divided into two sub-modules: a global AI prediction module and a local AI prediction module. The results of the local prediction module represent the content that the local user prefers, while the prediction results of global prediction module represent the content that the whole network user likes. In general, the prediction results of the two sub-modules are different.

The cache decision module, which is aligned with the knowledge & model function block in ETSI GS ENI 005 [3], will integrate the results from the two sub-modules, which form the AI prediction module, to make the final decision of which contents should be cached. The result of the cache decision module is defined as a content list sorted by the popularity of the content. The content recorded in the list will suggest to be cached.

The control module, which is used to initialize other module at the beginning of the processing, is responsible for the controlling of the process of AI-MC, such as determining the cycle of the overall process. And it is aligned with the knowledge & model function block in ETSI GS ENI 005 [3].

The information collection module, which is aligned with the data ingestion function block in ETSI GS ENI 005 [3], is used to collect key information which includes but is not limited to the hit ratio, the delay, which defined as the time since the user initiated the request until the response was received; the classification of content; the user behaviour, and to pre-process the information that has been collected to make the dataset.

The AI-MC provides the ability to cache popular content in the cache equipment in advance, which can significantly reduce network traffic and improve the user experience, like reducing the content acquisition latency.

##### 5.4.4.2.2 Motivation

In order to improve user experience and reduce access latency in a congested network environment, caching has emerged. The current caching technology, which only follows the simplest replacement principle and does not perform caching operations on popular content, is not effective. This caching technology only improves network performance in a limited way. In view of the existing problems of existing caching technology, an AI-MC is proposed which includes four modules.

In this use case, The AI-MC, which are characterized by the artificial intelligence prediction modules and other supporting modules, are used in the ENI System. Artificial intelligence technology can dramatically improve the accuracy of popular content prediction. The AI-MC predicts popular content by loading trained model and collected information then caches popular content in advance, which plays a key role in reducing latency and improving network stability.

The system can be described in Figure 5-43.

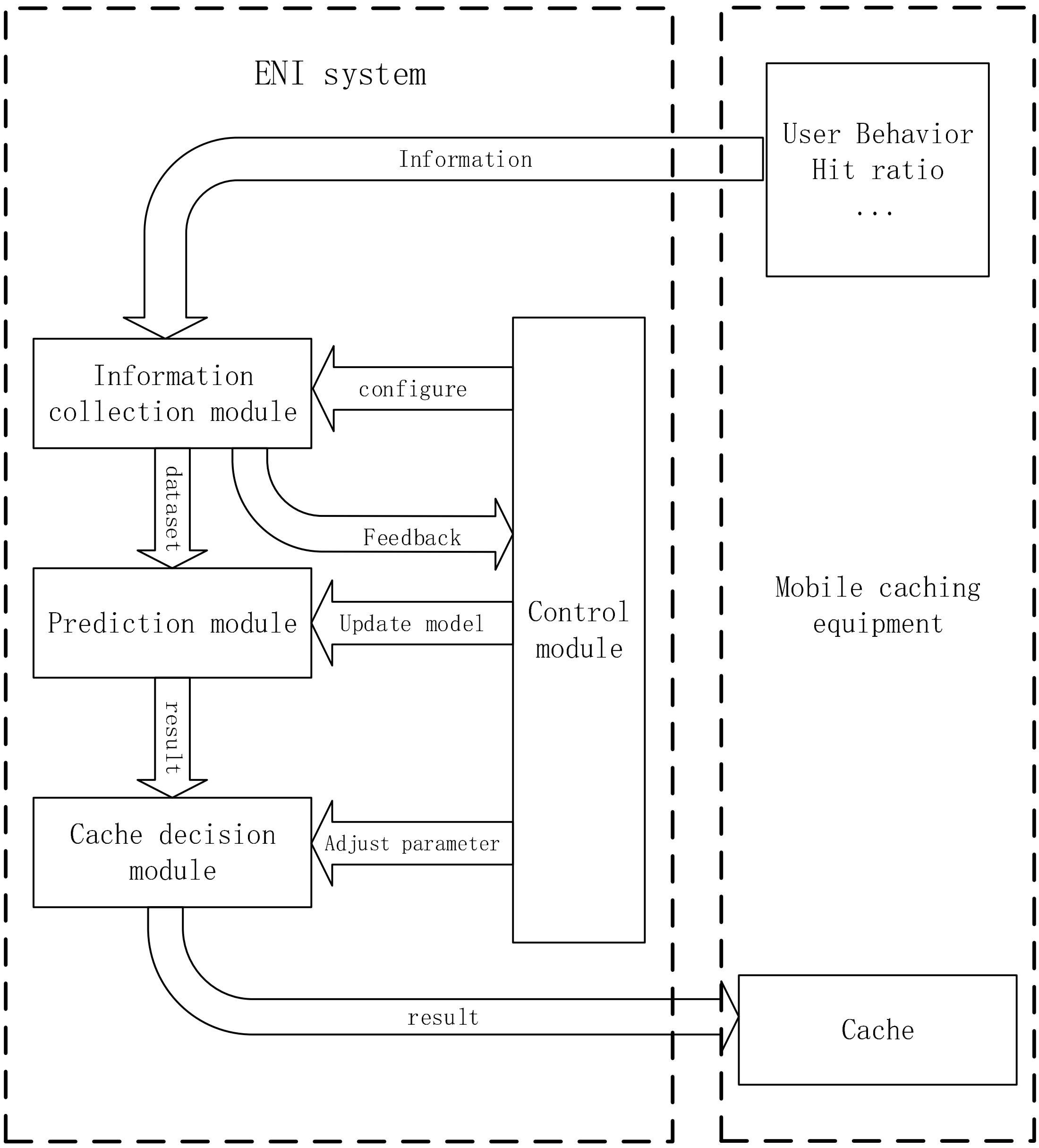


**Figure 5-43: AI-based mobile caching**

#### 5.4.4.2.3 Actors and Roles

- ENI System: System solution used to receive target information such as the user behaviour from MC equipment; to provide results to external system in predefined formats.
- AI-MC: entity/person who derives results of popular content which is suggested to be cached.
- MC equipment: A device used to provide the necessary information such as users behaviour, hit ratio; to provide the storage resources.

#### 5.4.4.2.4 Initial context configuration

- Mapping target content to predefined labels.
- Defining the prediction results formats.
- ENI System initials AI-MC.

- AI-MC in ENI System is enabled with AI capability, which is initialled to predict popular content through well-processed dataset.
- Related management systems connect to ENI System for acquiring results.

#### 5.4.4.2.5 Triggering conditions

In this use case, the cache hit ratio will be monitored by mobile edge cache equipment, when the cache hit ratio is lower than the predetermined value, or the prediction period is temporary, a new work process will be started under the control of the control module. In addition, the parameters of the AI prediction module will be adjusted by the control module based on the indicators.

#### 5.4.4.2.6 Operational flow of actions

ENI System with AI-MC is intended to enhance the capability of mobile edge caching and optimize the content acquisition latency based on the prediction results of the popular content. This kind of mechanism is realized by introducing the new AI capability (e.g. normally based on the deep-learning algorithm and architecture), with the following flow of activities:

1) The ENI System initials a process to receive training data from database and pre-processes the data to make dataset.
2) The ENI System initials a process to train the AI prediction module according to the dataset which have been well-processed in 1).
3) The ENI System presents the prediction results by loading the AI prediction module.
4) The ENI System combines the result of the two prediction modules to derive the popular content.
5) The external system commanded to initial a caching process to cache the popular content which has not been cached and replace the cached content that is not marked as popular content in the final results.

#### 5.4.4.2.7 Post-conditions

- In a work cycle, the ENI System needs to get metrics such as cache hit ratio, also needs to receive the user behaviour information.
- The control module updates the model parameters of the AI prediction module based on the information collected by the information collection module.
- The control module determines the caching period based on the configuration presented by the ENI System.
- The external system updates their cached content according to the result presented by ENI System.

### 5.4.5 Use Case #3-5: Service experience optimization of E2E slicing involving both OSS and BSS

#### 5.4.5.1 Use case context

Slicing constitutes a key feature in 5G systems and works as an enabler to guarantee diverse 5G services like eMBB/mMTC/uRLLC. Service experience optimization of E2E slicing is critical for network operation and customer management especially when considering elastic and dynamic change of slicing resources.

### 5.4.5.2 Description of the use case

#### 5.4.5.2.1 Overview

The focus of this use case is about how to optimize service experience by using E2E Quality of Experience (QoE) prediction and interaction with the client/tenant. In addition, the predicted QoE is also used for (re)configuration recommendation for slicing management in the OSS domain and marketing/sales in the BSS domain. The detailed architecture applying for this proposed use case is shown in Figure 5-44. Briefly speaking, ENI can calculate QoE for each segment domain, including Access Network (AN), Transport Network (TN) and Core Network (CN), based on collected KPIs and therefore calculate E2E QoE. To make the prediction more accurate, ENI System needs to interact with client/tenant about Mean Opinion Score (MOS) which stands for client's experience on the quality of slicing and update the model accordingly. For example if MOS has big difference with calculated QoE, it means AI model used by ENI System is not accurate and needs to be updated or iterated. Otherwise it means ENI has good performance on prediction of slicing quality. Based on the predicted QoE, the ENI System provides (re)configuration recommendations to the OSS and BSS, respectively.

NOTE: The OSS and BSS Functional Blocks (FBs) are not in the scope of the ENI architecture, only the info data that is exchanged via the interfaces that ENI maintains with those systems. However, just to turn the reading clearer, a very brief description is provided to each of them.

In the OSS domain, Communication Service Management Function (CSMF), Network Service Management Function (NSMF) and Network Sub-Slicing Management Function (NSSMF) are typical FBs defined and standardized by 3GPP. In the BSS domain, the Slicing application store and the Slicing service middle office should be regarded as particular instances of FBs applicable to the current Use Case scenario where Slicing application store manages different slicing product and Slicing service middle office deals with custom relation management, business intelligence, charging, market and so on. In addition, reference to some steps belonging to the approach described in the Operational Flow of Actions, clause 5.4.5.2.6, is depicted in Figure 5-44.

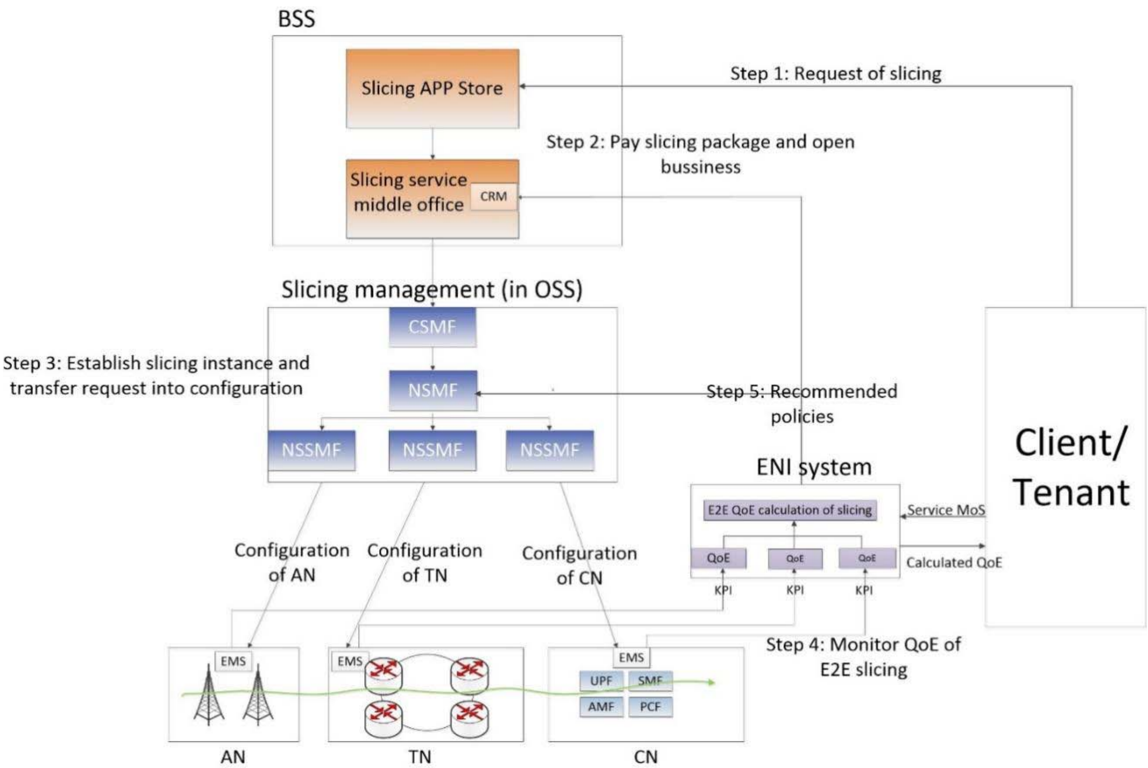


**Figure 5-44: Possible deployment scenario of Service experience optimization of E2E slicing involving both BSS and OSS**

Regarding "Calculated QoE" and "Service MOS", they constitute the input and the output messages that make part of a wider process that is further explained in clause 5.4.5.2.4.

### 5.4.5.2.2 Motivation

The motivation of this use case is to optimize service experience of E2E slicing by interacting with client/tenant and provides more accurate policy recommendations to both OSS and BSS.

### 5.4.5.2.3 Actors and Roles

The key actors are listed below:

- Client/Tenant: the actor which requests/pays for the slicing service.
- BSS: system that provides slicing store and service middle office.
- Service middle office: example of FB that is used in the current Use Case and provides customer relation management, business intelligence, charging and market/sales management in the BSS domain.
- Slicing management: example of functionality in the current Use Case that is accomplished in the OSS domain by using the CSMF, NSMF and NSSMF FBs.
- AN: Access Network.
- TN: Transport Network.
- CN: Core Network.
- ENI System: system that calculates QoE for each domain and E2E QoE.
- EMS: Element Management System located in the Infrastructure systems (AN, TN and CN), which collects network KPIs, delivers them to the ENI System, and performs (re)configurations enforced by the OSS.

### 5.4.5.2.4 Initial context configuration

Slicing instance is created and activated based on client/tenant's payment. EMS is interacting with slicing management in OSS and provides network KPIs to ENI System.

ENI System is able to learn QoE for an E2E slicing service by using AI-based technology, e.g. via AI algorithm based on these KPIs, and by interacting with client/tenant regarding MOS and prediction of QoE for the slicing service being offered.

The MOS stands for slicing experience score sensed from client/tenant, for example among 1-10, "1" means worst experience while "10" means best experience of slicing. Such quantitative mechanism should be defined. For QoE, the same way should be defined as well. One usage is ENI System may need to send such value to BSS/OSS via external interface. Another usage is ENI System is able to interact with Client/Tenant. Certain MOS model, for example quantified by 1-10, should be used here so that client/tenant has the ability to return the experience of slicing quality. This is only applicable when option 1 deployment mode of operation, explained below, is used. Otherwise, no predicted QoE value is returned in option 2.

### 5.4.5.2.5 Triggering conditions

In both of the implementation options described below, the ENI System should report a policy recommendation to both BSS and OSS. However, in option 1, it calculates the predicted QoE of the E2E slicing service and includes it in the message sent to the assisted OSS and BSS systems. In option 2, no value is passed, so it should be calculated by one of the FBs of the 3GPP system.

#### 5.4.5.2.6 Operational flow of actions

The following sequence of high level actions may be identified after the occurrence of the trigger:

- The ENI System obtains KPI from EMS of AN/TN/CN.
- The ENI System predicts QoE of E2E slicing using AI algorithm based on these KPIs.
- The ENI System generates recommended policy to OSS and BSS. There are two options:
    - option 1: ENI calculates the (re)configuration parameters and passes them to the OSS in dedicated interface. Those configurations will be standardized by ENI and will belong to its info model; or
    - option 2: Recommended policy coming from ENI is just a trigger for the 3GPP FBs calculating the (re)configurations and enforcing them into the network segments.

  In option 1, there is no need for the ENI System to wait for the client/tenant response to the "Calculated QoE" message before sending the "Recommended policies" to the OSS system. The intention is just to exchange calculated QoE and MOS to update AI model if the values have large difference.

- After obtaining a request from ENI System, 3GPP functional blocks in OSS domain or functions in BSS domain will take relevant actions, for example NSMF receives response from ENI System that E2E QoE is too low and it is recommended to reconfigure certain network parameters. Then NSMF will pass the recommended configuration of network to EMS.

#### 5.4.5.2.7 Post-conditions

Service experience is optimized by ENI System and business/network management is benefited based on accurate policy recommendations.

### 5.4.5.3 Mapping to ENI reference architecture

#### 5.4.5.3.1 Functional blocks

The detailed functional blocks are shown in Figure 5-45 and Table 5-2.

**Table 5-2: Mapping of ENI FBs to Use Case functionalities description**

| ENI Functional Block | Use Case functionalities description |
|---|---|
| Cognition Management Functional Block | This is the key function block which calculates E2E QoE of slicing via trained model. |
| Context-aware Management Functional Block | This block is used to predict context of data. |
| Knowledge Management Functional Block | This block is used to do knowledge management. |
| Data Ingestion and Normalization Functional Block | This block is used to filter and normalize the data. |
| MDE Functional Block | This block is used for translating the outcome of Cognition Management Functional Block to the value that can be understood by Policy Management Functional Block. |
| Policy Management Functional Block | This block is used for policy generation based on input from MDE functional block. |
| Denormalization and Output Generation Functional Block | This block is used to denormalize the data that could be used for OSS and BSS. |
| NOTE: The applicability of each FB in terms of what is its role in the overall implementation of the Use Case can only be seen as an example based on ETSI GS ENI 005 [3]. | |

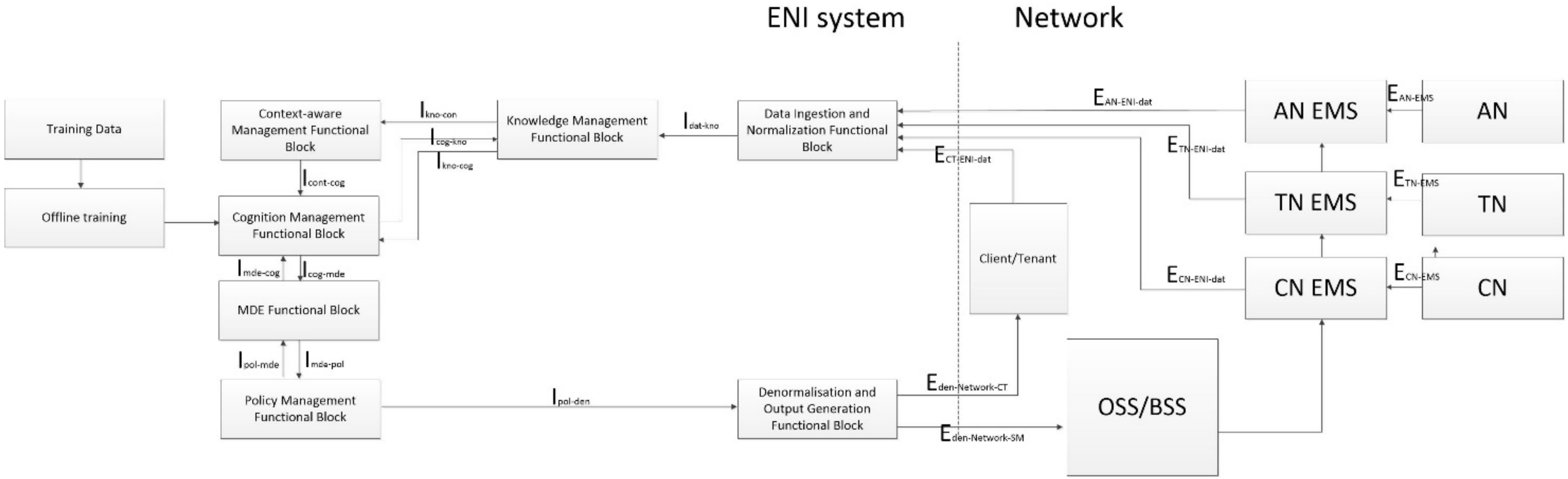


**Figure 5-45: Mapping to ENI reference architecture for service experience optimization of E2E slicing considering both BSS and OSS**

### 5.4.5.3.2 Interfaces

Table 5-3 contains an example of a set of Reference Points (RPs) belonging to ETSI GS ENI 005 [3], which may be used on the implementation of a solution for the scenario described by this use case.

NOTE: The applicability of each RP in terms of what is its role in the overall implementation of the Use Case can only be seen as an example, based on ETSI GS ENI 005 [3].

**Table 5-3: Mapping of ENI RPs to Use Case functionalities description**

| ENI External Ref. Points | Use Case functionalities description |
|---|---|
| $E_{AN-EMS}$ | Interface between AN and EMS. Traffic data from AN is transmitted via such interface and EMS uses such data to calculate KPI. |
| $E_{TN-EMS}$ | Interface between TN and EMS. Traffic data from TN is transmitted via such interface and EMS uses such data to calculate KPI. |
| $E_{CN-EMS}$ | Interface between CN and EMS. Traffic data from CN is transmitted via such interface and EMS uses such data to calculate KPI. |
| $E_{AN-ENI-dat}$ | Interface between AN EMS and "data ingestion and normalization functional block". KPI data from AN is transmitted from such interface and ENI uses such data to calculate QoE of this domain. |
| $E_{TN-ENI-dat}$ | Interface between TN EMS and "data ingestion and normalization functional block". KPI data from TN is transmitted from such interface and ENI uses such data to calculate QoE of this domain. |
| $E_{CN-ENI-dat}$ | Interface between CN EMS and "data ingestion and normalization functional block". KPI data from CN is transmitted from such interface and ENI uses such data to calculate QoE of this domain. |
| $E_{CT-ENI-dat}$ | Interface between Client/Tenant and "data ingestion and normalization functional block". Such interface is used for message (e.g. MOS) exchange from Client/Tenant to ENI System. |
| $I_{dat-kno}$ | Interface between "data ingestion and normalization functional block" and "Knowledge Management Functional Block". Such interface is used for transmitting data for filtering/normalization. |
| $I_{kno-con}$ | Interface between "Context-aware Management Functional Block" and "Knowledge Management Functional Block". Such interface is used for transmitting data of "Knowledge Management Functional Block" for context processing. |
| $I_{cog-kno}$ | Interface between "Cognition Management Functional Block" and "Knowledge Management Functional Block".<br>Such interface is used to transmit for example control signalling for "Cognition Management Functional Block" what data needs to be processed. |
| $I_{kno-cog}$ | Interface between "Cognition Management Functional Block" and "Knowledge Management Functional Block". Such interface is used to transmit for example processed data, characteristic value generated by "Knowledge Management Functional Block", for prediction by "Cognition Management Functional Block". |
| $I_{cont-cog}$ | Interface between "Cognition Management Functional Block" and "Context-aware Management Functional Block". Such interface is used for transmitting context related data from first functional block to second functional block. |
| $I_{cog-mde}$ | Interface between "Cognition Management Functional Block" and "MDE Functional Block". Such interface is used for transmitting QoE related data. |
| $I_{mde-pol}$ | Interface between "Policy Management Functional Block" and "MDE Functional Block". Such interface is used for transmitting QoE related data. |
| $I_{pol-den}$ | Interface between "Policy Management Functional Block" and "Denormalization and Output Generation Functional Block". Such interface is used for transmitting policy related data. |
| $E_{den-Network-CT}$ | Interface between "Client/Tenant" and "Denormalization and Output Generation Functional Block". Such interface is used for transmitting QoE from ENI to Client/Tenant. |
| $E_{den-Network-SM}$ | Interface between "OSS/BSS" and "Denormalization and Output Generation Functional Block". Such interface is used for transmitting QoE and recommended policy from ENI to OSS/BSS. |

### 5.4.5.3.3 Flow of information

The flow of information is shown in Figure 5-46.

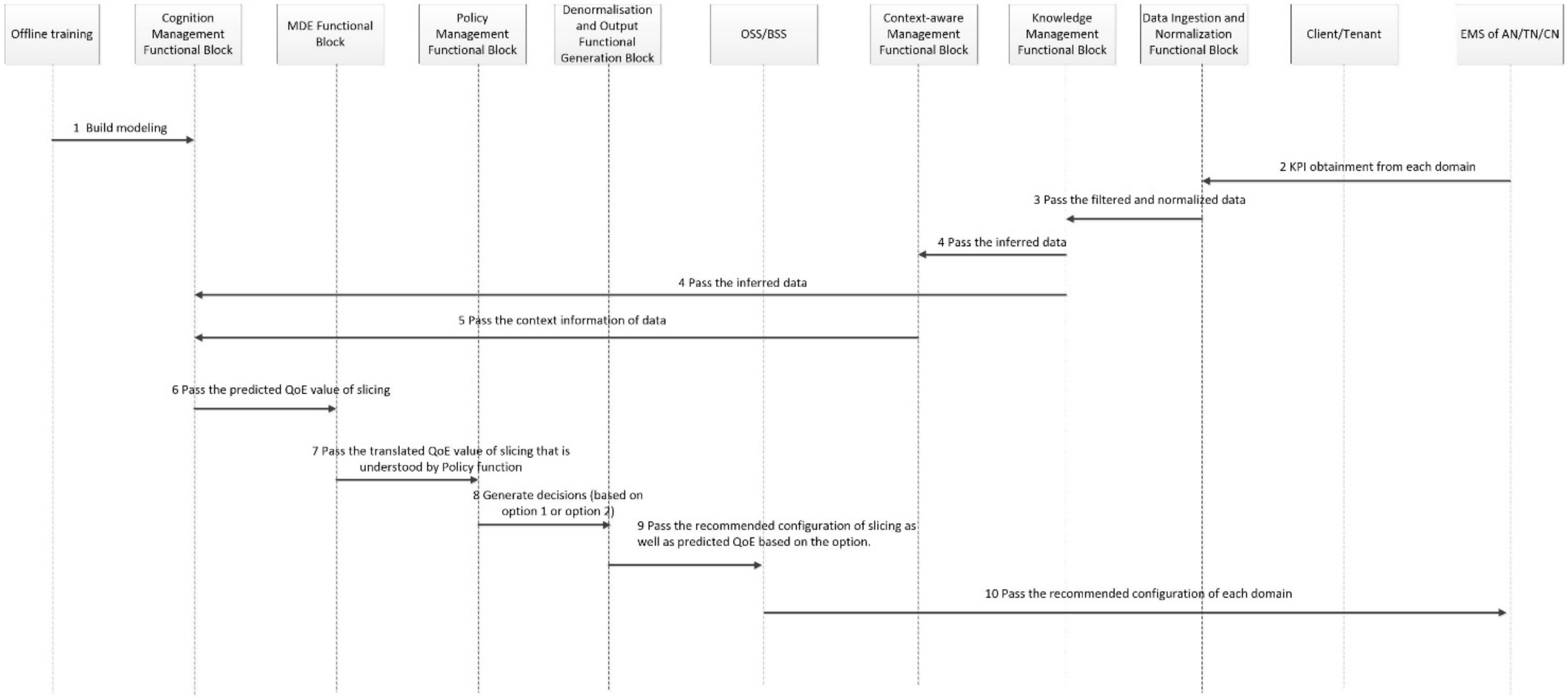


**Figure 5-46: Flow of information for "service experience optimization of E2E slicing involving both BSS and OSS"**

In step 1, the model is offline trained and built to "Cognition Management Functional Block". In step 2, KPI is obtained from AN/TN/CN EMS and passed to "Data Ingestion and Normalization Functional Block", which does the filtering and normalization and passed the processed data to "Knowledge Management Functional Block" in step 3. The inferred data is passed to both "Context-aware Management Functional Block" and "Cognition Management Functional Block" in step 4. In step 5, "Context-aware Management Functional Block" passes context information of data to "Cognition Management Functional Block".

The "Cognition Management Functional Block" predicts QoE of the E2E slicing service being offered and passes the predicted QoE to "MDE Functional Block" in step 6. It always calculates and predicts this parameter value, however it is passed to the OSS/BSS system only if the deployment scenario operates in option 1. Then, this QoE value that is understood by policy function is passed to "Policy Management Functional Block" in step 7 and the internal model is updated. In step 8, "Policy Management Functional Block" generates policies and passes them to "Denormalization and Output Functional Block". In step 9, the recommended policy is passed to slicing management in OSS domain and BSS domain with or without the predicted QoE service value and associated (re)configurations of network elements, i.e. option 1 or option 2 described above. In step 10, the recommended (re)configurations are enforced into EMS.

## 5.4.6 Use Case #3-6: Intent-based Cloud Management for VDI service

### 5.4.6.1 Use case context

An increasing number of enterprises are adopting DaaS to support their employees in accessing the standard virtual desktop environment from various locations. According to Gartner (Table 5-4), DaaS is expected to have the most significant growth in the worldwide public cloud service market in 2020, increasing 95,4 % to USD 1,2 billion. The expansion of the DaaS market has been accelerated due to the demand to work from home during the COVID-19 pandemic. It is also forecast that the DaaS market will continue to grow rapidly in 2021 and 2022, and the revenue from DaaS services will double in 2022 compared with that in 2020.

**Table 5-4: Worldwide public cloud service revenue forecast (Millions of U.S. Dollars)**

| | 2019 | 2020 | 2021 | 2022 |
|---|---|---|---|---|
| Desktop as a Service (DaaS) | 616 | 1 203 | 1 951 | 2 535 |
| Cloud Service Total Market | 242 697 | 257 867 | 306 948 | 364 062 |

An overview of a DaaS platform is shown in Figure 5-47. DaaS is usually implemented on a private cloud or public cloud platform empowered by virtualization technology such as Kernel-based Virtual Machine (KVM) and XEN. A DaaS VM instance is assigned for each DaaS user, and the DaaS user is able to access his/her DaaS instance and conduct daily operations on the guest OS of the instance.

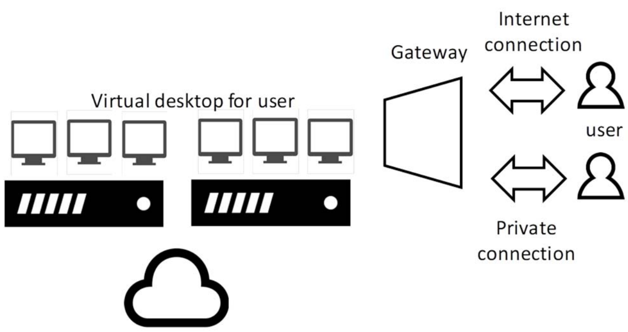


**Figure 5-47: DaaS system view**

### 5.4.6.2 Description of the use case

#### 5.4.6.2.1 Overview

In a VDI service, in order to maintain users' QoE, VDI administrators need to determine and adjust cloud resource configuration, including the number of VMs to be placed on each host, the CPU and memory allocated for each VM appropriately. However, this decision requires a high level of skill and experience. Improper decision can lead to poor user experience or low resource efficiency.

To solve the challenge, IBCM is designed to capture the knowledge about the VDI resource design, IBCM trains a model based on the performance, resource and workload data those are collected from the VDI service and the cloud environment. By using the abstracted knowledge, IBCM is able to determine the optimal resource configuration for various user QoE requirements (intent). Consequently, reduction of OPEX including the human resource cost, time cost and cloud resource cost can be expected by autonomizing the cloud resource decision process by the IBCM.

#### 5.4.6.2.2 Motivation

To ensure employee work efficiency, the enterprise client of a DaaS service shall keep a high level of user QoE. Since QoE is directly impacted by the cloud resource amount allocated for the DaaS services, the DaaS operator has to carefully design the cloud resource amount to meet the QoE requirement. According to our interviews with DaaS operators, deciding and adjusting the cloud resource amount mainly relies on experienced operators, and the design approach is also called human-based resource design approach in the present document. The decision relies on DaaS operator's knowledge and experience about the dependency among cloud-resource-amount configurations, DaaS workload patterns, and DaaS performance. The task is highly complex due to the numerous resource-amount configurations as well as the high variation in DaaS workload.

Along with the expansion of DaaS, human-based resource design for DaaS services is increasing, while the number of experienced DaaS operators is highly limited, and hiring and training of DaaS operators usually incurs huge cost. Furthermore, the manual resource-design process requires hours to days and may lead to long-term QoE degradation due to the latency of resource design.

Thus IBCM is designed in order to automatically calculate the optimum number of VMs that does not deteriorate the user experience. Thus realizes reduction of human cost and resource cost.

#### 5.4.6.2.3 Actors and Roles

VDI operator: manages the VDI service, specify the VDI QoE requirement for the users, i.e. the intent for VDI service, check whether the VDI users is having sufficient QoE.

IBCM system: learns the casual relationships between the VDI workload, resource configurations and the corresponding QoE; determines optimal resource configuration for the intent.

VDI service system and the cloud infrastructure: provide the required information to the IBCM system, execute the resource configuration determined by IBCM system.

#### 5.4.6.2.4 Initial context configuration

In a VDI service, virtual desktop environments are implemented as VM instances on public/private cloud hosts. VDI users conduct their daily work in the virtual desktop instances. The VDI service system and the cloud infrastructure collect the VDI performance log data, IBCM trains the performance inference model based on the log data.

#### 5.4.6.2.5 Triggering conditions

The VDI operator specifies the VDI intent through the IBCM GUI.

#### 5.4.6.2.6 Operational flow of actions

Step 1: The VDI operator specifies the intent through the GUI.

Step 2: IBCM checks if the current resource configuration meets the intent.

Step 3: If not, IBCM calculates the number of instances to be allocated to the host that meet the intent as well as the expected performance. The result is fed back to the operator for confirmation.

The decision is transformed into machine-readable resource orchestration template and handed to VDI resource management system for implementation.

#### 5.4.6.2.7 Post-conditions

The optimal VDI resource configuration for the intent is applied. The VDI service system continuously monitor whether the intent is satisfied and report the result to the operator. The log data are also collected are future IBCM model training.

### 5.4.6.3 Mapping to ENI reference architecture

#### 5.4.6.3.1 Functional blocks

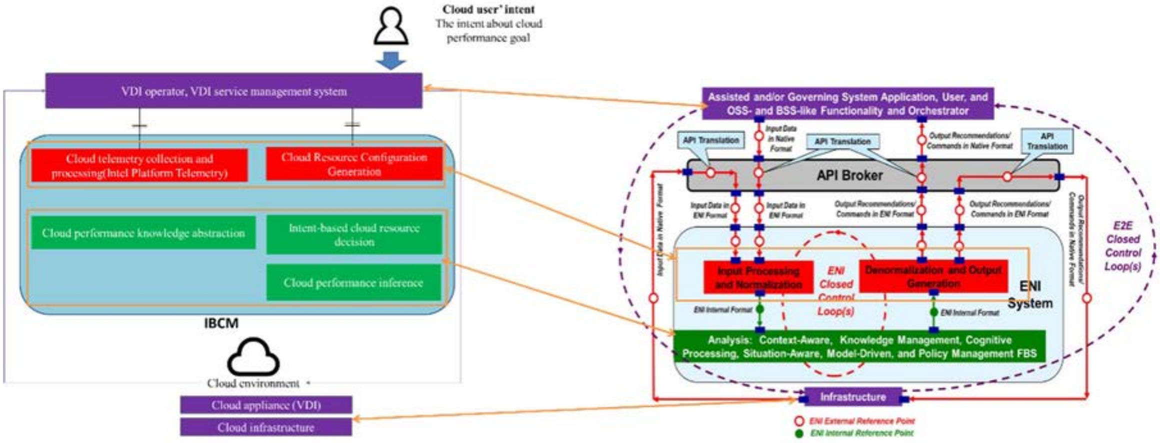


**Figure 5-48: Mapping to ENI reference architecture**

The architecture of IBCM is shown in Figure 5-48.

The IBCM system is a closed-loop system. Firstly, telemetry logs about the performance, etc., are collected from the cloud environment, and knowledge about the resource and performance causal relationship is abstracted from the logs. When cloud users input their cloud performance goal as intent, the knowledge is then used to make decision about the resource amount configuration for the intent. Finally, the resource configuration is implemented in the cloud environment and provided to the cloud users, and logs about the performance, etc. are collected.

IBCM has five main Functional Blocks (FBs) as follows:

- Cloud telemetry collection and processing:
  - This FB collaborates with telemetry collection agents in the cloud environment to collect the cloud telemetry log data, such as cloud resource amount configurations, cloud application workload amount, cloud and application performance log data from various level of the cloud environment, and conducts combinations, normalizations, etc. of the log data.
- Cloud performance knowledge abstraction:
  - In the FB, models are trained on the basis of the collected log data. It trains models which infers the performance for the given the workload amount and the resource configuration information. The trained models are passed to the cloud performance inference FB.
- Cloud performance inference:
  - In this FB, first, the workload requirement, and configurable patterns of cloud resource amount are inputted to the generated models to infer the cloud performance. The pairs of inferred performance and the corresponding resource amount are passed to the intent-based cloud resource decision FB.

- Intent-based cloud resource decision:
  - This FB decides the resource amount that meets the intent on the basis of the performance inference results. For each pair of inferred performance and the corresponding resource amount, the FB examines whether the inferred performance satisfies the intent, if yes, the corresponding pattern of resource amounts is outputted as resource solutions. Cloud resource configuration generation for the intent.

In this FB, the selected resource amount is embedded in a resource orchestration template/command e.g. yaml and sent to resource orchestrators.

The recommend resource configuration is also sent back to the cloud provider for confirmation or manual adjustment if necessary. The provider is able to confirm/revise the resource decision and instruct launching of the VMs accordingly. In accordance with the resource decision, the resource orchestrators allocate the resources and activate the VMs, and the service is provided to the cloud user. The performance and system logs are collected and potentially used to update the models.

The IBCM framework can be mapped to the ENI reference architecture (Figure 5-48). The cloud user/provider, cloud monitoring and management system are treated as Assisted and/or Governing users and systems in ENI System respectively. The cloud telemetry collection and processing of IBCM and the cloud resource configuration generation of IBCM play the roles of Input Processing and Normalization, Denormalization and Output generation of ENI System respectively. The cloud performance knowledge abstraction, cloud performance inference and intent-based cloud resource decision largely involve analysis process that is similar with ENI analysis parts, including the context awareness, knowledge management, situation awareness, policy management, etc.

## 5.4.6.3.2 Flow of information

The flow of information is given in Figure 5-49.

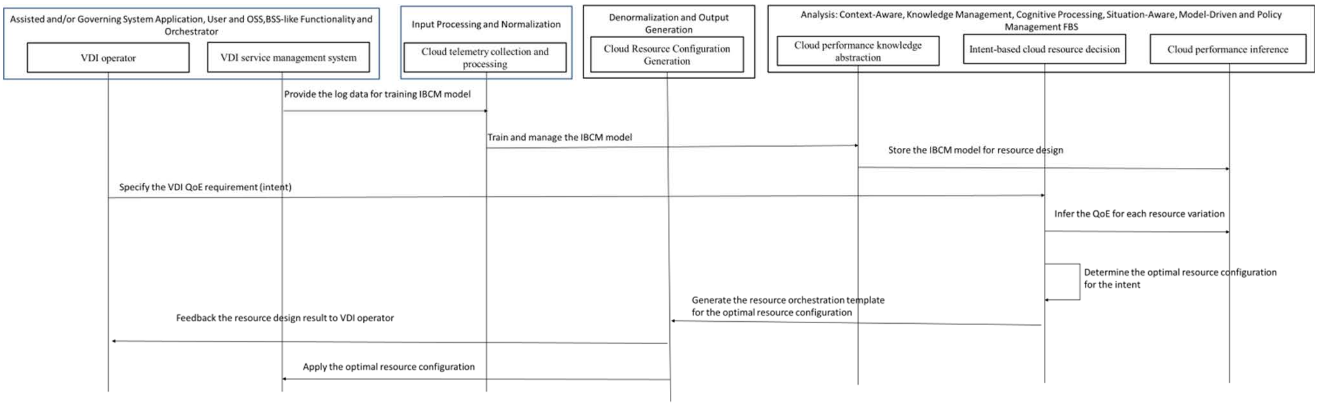


**Figure 5-49: Procedure of IBCM to decide the optimal resource for user intent**

The following step description follows the diagram:

Step 1: The VDI service management system collects and provided the log data about VDI resource, workload and performance data to the cloud performance knowledge abstraction FB.

Step 2: The cloud performance knowledge abstraction FB trains the IBCM models and stores the model in the cloud performance inference FB.

Step 3: The VDI operator specifies the VDI QoE requirements as intent, the intent-based cloud resource decision FB handles the intent, instructing the cloud performance inference FB to infer the QoE for possible resource configuration, and selecting the optimal resource configuration that meet the QoE requirements.

Step 4: The cloud resource configuration generation FB generates the resource orchestration template in accordance with the optimal resource configuration.

Step 5: The generated cloud resource orchestration template is feedback to the VDI operator for confirmation; then it is applied to the cloud environment via the VDI service management system.

## 5.4.7 Use Case #3-7: Intelligent vehicle diversified service fulfilment based on polymorphic network

### 5.4.7.1 Use case context

With the rapid development of Internet of things and mobile communication technology, the mobile access network represented by the intelligent vehicles has developed promptly. As a complete end-to-end network system, the Internet of vehicles can accomplish the comprehensive network links, such as Vehicle to Network (V2N), Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), Vehicle to People (V2P), in order to realize the integration of network from on-board mobile internet, the inter-vehicle network, and the in-vehicle network, as shown in Figure 5-50. The Internet of vehicles makes use of sensor technology to sense the state information of vehicles and environment data, and uses wireless communication network and intelligent information processing technology to realize the management of traffic, intelligent decision-making of traffic information services and control of vehicles, which is aimed to provide the safe, comfortable, intelligent and efficient driving experience to users.

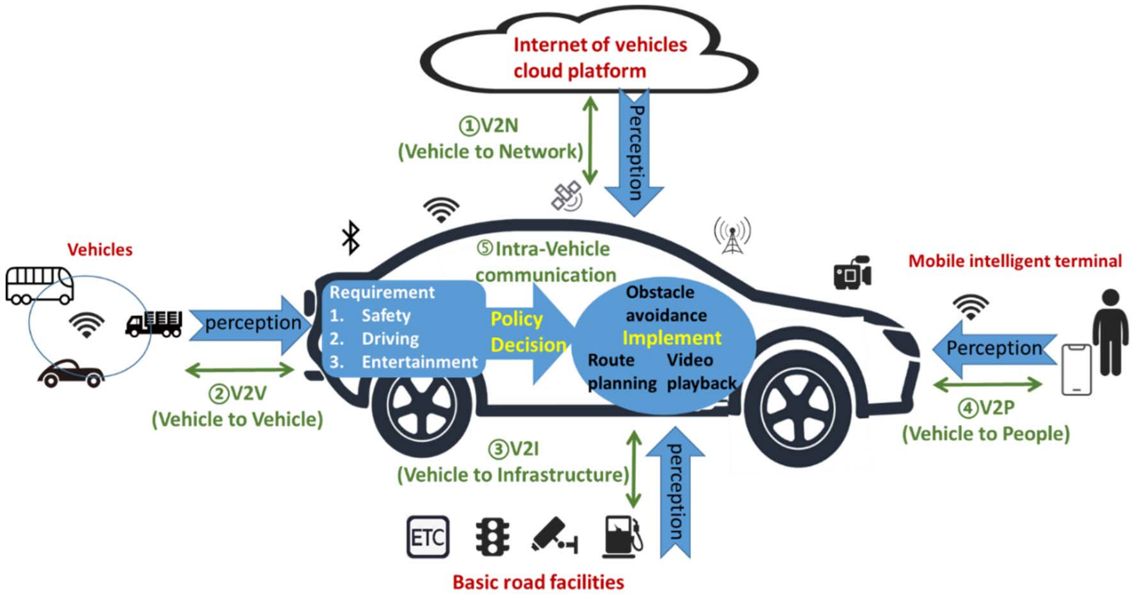


**Figure 5-50: Typical internet of vehicles communication scenarios**

However, the current Internet of vehicles based on IP in a single model cannot support the requirements of a variety of business scenarios. The Internet of vehicles is confronted with several problems.

EXAMPLE: In high-speed driving scenarios based on IP, it might occur to have communications delay and data dropped, which can not guarantee the QoS requirements for data integrity and timeliness. Secondly, the traffic-accident reports based on IP are not as timely or reliable as geographic location. If the moving vehicle notices a traffic accident ahead, it should report the emergency to the road management centre. Then the centre is supposed to take into account the travel direction and warn the vehicle based on the accurate geographical location currently, which can assure driving safety. Thirdly, the Internet of Vehicles scenarios should offer drivers interactive entertainment options, High-Definition video transmission, and other varied services based on video resources, entertainment information, and other interactive data. Since the ability and workload of network will affect the audio and visual experience, the IP based single model can not support the edge computing network to achieve high reliability.

Polymorphic Network (PINet) is a network composed of multiple or more network models running on standardized software and hardware interfaces and intelligent operation and maintenance management systems. It supports the full-dimensional definitions of routing mode, switching mode, interconnection mode, network element form, transmission protocol, service attributes, etc.

PINet contains the high coverage and widely used infrastructure, which is compatible with various access technologies in the basic device layer and supports different scenarios service types in the application layer. By using it in the application scenario of the Internet of vehicle, the above problems in Internet of vehicles can be perfectly solved. Since the polymorphic network allows for the free switching of multiple models (such as identifier based, location based, content based and customized models) according to end-user requirement and business scenarios. Therefore, the requirement of low delay, bounded jitter and high robustness can be achieved in an efficient manner, improving the intelligent levels of vehicle services [i.8].

### 5.4.7.2 Description of the use case

#### 5.4.7.2.1 Overview

PINet supports full-dimensional definitions of functions such as data forwarding, heterogeneous interconnection, addressing routing, resource scheduling and function arrangement in the whole architecture. IP, content, identification, geospatial location and other user-defined identities can coexist and collaborate in the same physical network, which breaks through the bottleneck of traditional single IP carrier network and meet the diversified and professional demand for efficient services.

The core demand of communication in the Internet of vehicles lies in the guarantee of each telecommunication subject in the moving process, especially the strict demand for deterministic time delay, so as to support the real-time performance of vehicle positioning and control signals. In order to ensure the driving safety of vehicles, it requires the overall network to ensure that when the IP address changes and switches during the movement, the upper layer scenario is not interrupted and there is almost no delay effect. The Internet of vehicle cloud platform can interact with the vehicle based on video resources, real-time road conditions, weather forecast and other information, which provides drivers and passengers with entertainment services, road condition prediction and route navigation.

The traditional network configuration and operation based on a static strategy model are difficult to adapt to the complexity of current Internet of vehicle services, resulting in network inefficiency and poor user experience. These flaws in traditional network can be corrected by PINet. PINet provides identity-based addressing and routing (globally unique), the delay issue brought on by frequent IP address switching in dynamic environments can be resolved. Figure 5-51 illustrates the main difference between traditional IP based Internet and PINet. In order to overcome the limitations of the conventional single IP carrier network and satisfy the varied and professional demand for effective services, PINet supports the coexistence and collaboration of IP, content, identification, geospatial location, and other self-defined identities in the same physical network.

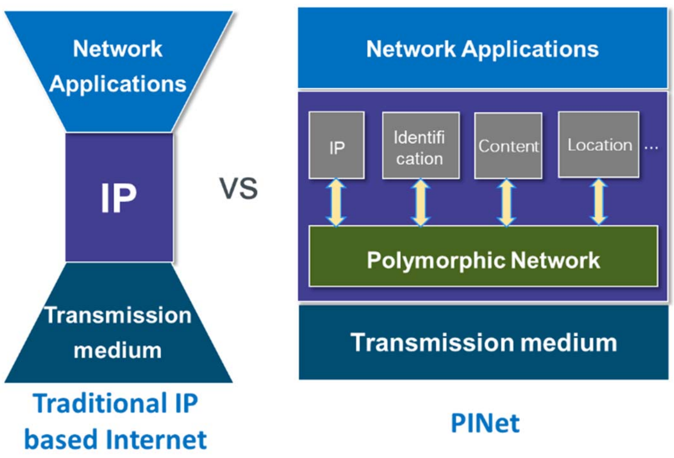


**Figure 5-51: Difference between traditional IP based Internet and PINet**

ENI System provides a "perception-decision-adaptation" closed loop with its intelligent management and control capability to realize the diversified service requirement of Internet of vehicle. ENI System collects resources and environment data in real time from cloud platform, road facilities, vehicles, sensors, analyses and matches the resource and environment information with specific safety, driving and entertainment requirement from end-users by using artificial intelligence technology, output intelligent strategies to targeted entities. In this way, PINet automatically switches different models according to strategies of ENI System and configures underlying routes, resources and protocols. And the vehicles automatically perform obstacle avoidance, route planning, video playback and other operations, which can provide the optimized driving route and comfortable entertainment service on the premise of ensuring the safety of vehicles and passengers.

### 5.4.7.2.2 Motivation

The ENI System collects data (such as weather information, road and driving condition, etc.) both inside and outside of vehicles, monitoring end-user's requirement and intent. It then uses intelligent technologies to analyse and match to corresponding business or service scenario. Real-time business and resource status should be continuously monitored, so as to realize efficient adaptive adaptation and fitting of network resources and upper services.

By using ENI System, intelligent strategies can be generated and output to PINet and vehicles to configure network and perform corresponding operations according to different requirement scenarios. For example, in the path planning scenario, ENI System can make use of geographic location based network to collect information of paths, vehicles, pedestrians, traffic signals and so on, and utilize artificial intelligent algorithms to carry out congestion prediction and path routing. When application requirement changes to entertainment, ENI System can leverage content based network model to download cache video resources to provide drivers with interactive entertainment services.

### 5.4.7.2.3 Actors and Roles

- Internet of vehicle: collects data from the on-board mobile internet, the inter-vehicle network and the in-vehicle network, and then input the collected data into the ENI System; execute the policies delivered by ENI System, for example, obstacle avoidance, route planning and video playback, etc.
- End-users: input intent or service requirements to the ENI System, including drivers, passengers, vehicles, etc.
- ENI System: collects data from Internet of vehicle and intent from end-users; analyse and learn end-users business requirements pattern, generate detailed strategies for the user-initiated business according to the basic business parameters and performance expectations, and relies on the fitting relationship between business and service to realize the mapping from business requirements to network configuration strategies and operation instructions, and then distribute them to PINet and vehicles respectively.
- Polymorphic network: performs model switching strategies and configure the corresponding fundamental network topology, software/hardware, interface and routes.

### 5.4.7.2.4 Initial context configuration

Initially, the Internet of vehicle should collect enough data including road condition and weather condition in the early range, in order to train original AI models. ENI System should learn the communication requirements corresponding to different business scenarios in the Internet of vehicles, correlates with the optimized network model, and builds the initial policy database based on well-trained models.

### 5.4.7.2.5 Triggering conditions

ENI System detects or predicts that the driving safety level exceeds the specified threshold, or received the feedback of obvious changes in the external environment (such as crowded roads and extreme weather), or the change requirement from users (such as request entertainment resources), which will indicate switching to the desired network model.

### 5.4.7.2.6 Operational flow of actions

1) ENI System collects and store data from Internet of vehicle cloud platform, road facilities, vehicles, sensors and End-users, including weather information, road and driving condition and service request, etc.

2) ENI System uses AI algorithm to build the relation between service requirement to corresponding network model and its required resources.

3) ENI System learns the service pattern, detects and predicts the required resource, road and driving conditions changes in a certain period in the future.

When triggering conditions occurs:

4) ENI System generates network model switching strategies, detailed network configuration and resource scheduling strategies to PINet, and associated operation strategies to targeted vehicles. For example:

   a) In high-speed driving scenarios, communications delay and data dropped exceed the threshold, which implies that current network condition can not guarantee the QoS requirements for data integrity and timeliness. The output Policies to PINet may include switching to identity based network model, as well as specific network configuration and resource scheduling strategies. Identity based network model can be chosen since it allows for seamless transition at any location. The globally unique feature enables it to provide mobile support for the Internet of vehicles.

   b) In safe-driving scenarios, if the moving vehicle finds a traffic accident ahead, it should notify the road management centre about the emergency. ENI System perceives this circumstance, inform PINet to configure geographic location base network model, because that has quicker and more precise perception of position. Additionally, ENI System employs AI algorithm to conduct out path planning and guide moving vehicles in a safe manner based on the road information, emergency information, and vehicle driving information that have been collected. At the same time, ENI System sends the accident alert to the vehicles within 5 kilometres behind the accident, reminding them to give way urgently.

##### 5.4.7.2.7 Post-conditions

The scenarios of the Internet of vehicles can be constantly enriched by continuously learning of historical collected data, which improves the application scope of intelligent decision-making models. The Internet of vehicle continuously monitors whether the requirement is satisfied, and report the evaluation result for further optimization.

### 5.4.8 Use Case #3-8: AI based family broadband network user experience optimization.

#### 5.4.8.1 Use Case context

With the development of Gigabit broadband network, broadband network can satisfy requirement of all Internet entertainment access of family customers. However, during the services operation, it is often faced that the case of network performance of broadband network is good, but customers' experience is not as good as the network performance. It is hard to fully use the massive data from network by traditional methods for evaluating the user experience and resolving the related issues. An intelligent solution is needed to realize the dynamic optimization of quality of experience for the service providers.

#### 5.4.8.2 Description of the use case

##### 5.4.8.2.1 Overview

In this case, ENI System will get massive history data from operator, including OSS data, BSS data and customer's subjective evaluation and complaint data e.g. net promoter score data. These history data will be used to train customer experience model， and the qualified customer experience model will be stored in the ENI System. ENI System will get periodic operation data form operator, analyse and obtain poor quality users and the root causes of poor quality, and determine the poor quality optimization policy.

##### 5.4.8.2.2 Motivation

As mentioned above, customer experience problems mainly expose by customer complaints. However, customers are not network professionals and it is difficult for them to describe the specific reasons. Operators have difficulties to handle accurately customer complaints problem. The AI based user experience optimization solution can discover, predict, optimize, repair problems in advance and care for customers to maximize perception of customers' experiences.

### 5.4.8.2.3 Actors and Roles

The presence of the following actors/entities as well as their associated roles are envisaged in the current Use Case:

- **Customers:** are family broadband network users.
- **Operator:** have the following capabilities:
  a) Customer experience management.
  b) Supply customer experience data to ENI System.
  c) Receive the policy sent by ENI System, and push policy to execution entity.
- **ENI System:** train customer experience model with history data and use customer experience model to collect and analyse periodic customer experience data. ENI System provides optimization policy for operator to realize optimization poor quality customer:

NOTE: Customer experience data include OSS, BSS, customer's subjective evaluation and complaint data.

  a) OSS data: optical network unit optical power, TCP average retransmission rate, etc.
  b) BSS data: User's bill, charge, Pricing, etc.
  c) customer's subjective evaluation and complaint data: survey data, customer complaint data, etc.

- **Operations Support System (OSS):** In this case, OSS system represents network management and optimization system, and manage network Infrastructure, network services, Customer service and network application, etc.
- **Business Support System (BSS):** In this case, BSS manage settlement, billing and customer service, etc.
- **Network Infrastructure:** Network infrastructure includes optical line terminal, optical network unit, broadband remote access server, etc.

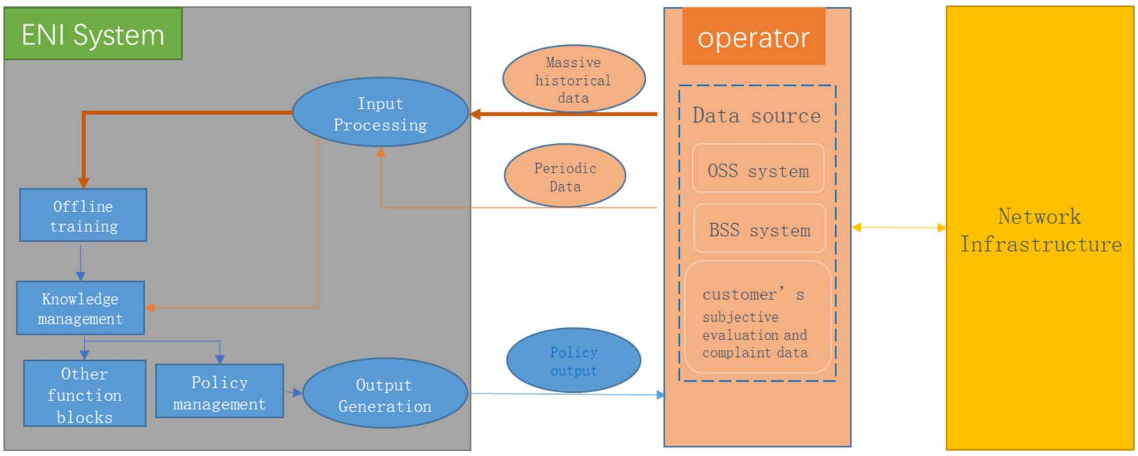


**Figure 5-52: Relationship with ENI System**

### 5.4.8.2.4 Initial context configuration

- The customer experience data used to model train should be typical, comprehensive and accurate.

NOTE: Customer experience data include OSS data, BSS data, customer's subjective evaluation data and complaint data.

- ENI System can get customer experience data by operator.

#### 5.4.8.2.5 Triggering conditions

- ENI System collects customer experience data to analyse customer experience situation.
- Operator sends service optimization requests to ENI System.

#### 5.4.8.2.6 Operational flow of actions

The following sequence of actions may be identified:

1) **Training of Intelligent model:** ENI System periodically gets history customer experience data to train Intelligent customer experience optimization model.

NOTE 1: Massive history customer experience data include OSS data, BSS data, customer's subjective evaluation data and complaint data.

2) **Triggering optimization:** There are two triggering optimization situations. The first is ENI System obtaining periodic customer experience data from operator to analyse customer experience situation; the second is operator sending service optimization requests to ENI System.

3) **Policy generation:** ENI System analyses customer experience data output optimization policy and sends optimization policy to operator.

NOTE 2: Policy may be used for network optimization, customer care, etc.

4) **Policy execution:** operator receives optimization policy and pushes the optimization policy to the execution entity.

NOTE 3: According to the specific scenarios, the execution entity may be the electric operation maintenance system of OSS or the business hall of BSS, etc.

5) **Policy enhancements:** operator collects feature data from significantly improved in customer experience and sends feature data to ENI System for iterative optimization of intelligent customer experience optimization model.

NOTE 4: Feature data may be complaints, flow consumption, etc.

6) The actual service scenario conditionally triggers full or part of above actions.

#### 5.4.8.2.7 Post-conditions

- Significantly improved customer experience.

### 5.4.8.3 Mapping to ENI reference architecture

#### 5.4.8.3.1 Mapping to ENI reference architecture

The mapping to ENI architecture for AI based family broadband network user experience optimization is shown in Figure 5-52a.

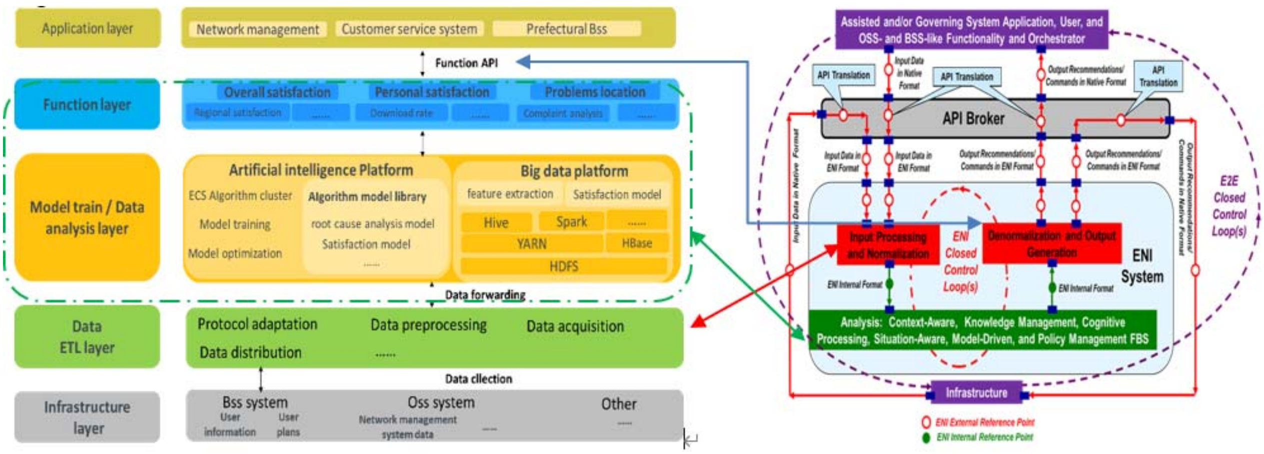


**Figure 5-52a: Mapping to ENI reference architecture**

AI based family broadband network user experience optimization system architecture include six parts.

**Infrastructure layer:** produce user experience data, including OSS data, BSS data, etc.

**Data ETL (Extract-Transform-Load):** dispose of data, including data acquisition, data pre-processing, data distribution, etc. It can be mapped to ENI data ingestion and normalization functional block for data collection and normalization.

**Model train and data analysis layer:** be used to model training and data analysis, etc. including functions such as data storage, feature extraction, model library, model optimization, etc. It can be mapped to ENI analysis functional block for data exploration and learning, model training and optimization.

**Function layer:** provide service satisfaction analysis function, including overall satisfaction, personal satisfaction, problems location (e.g. connection failure location and identification), etc. It can be mapped to ENI analysis functional block for functional encapsulation.

**Function API:** it can be mapped to ENI denormalization and output generation functional block for function of user experience optimization opening.

**Application layer:** receive user experience optimization policy, and dispose user experience problems.

### 5.4.8.3.2 Flow of information

The flow of information is given in Figure 5-52b.

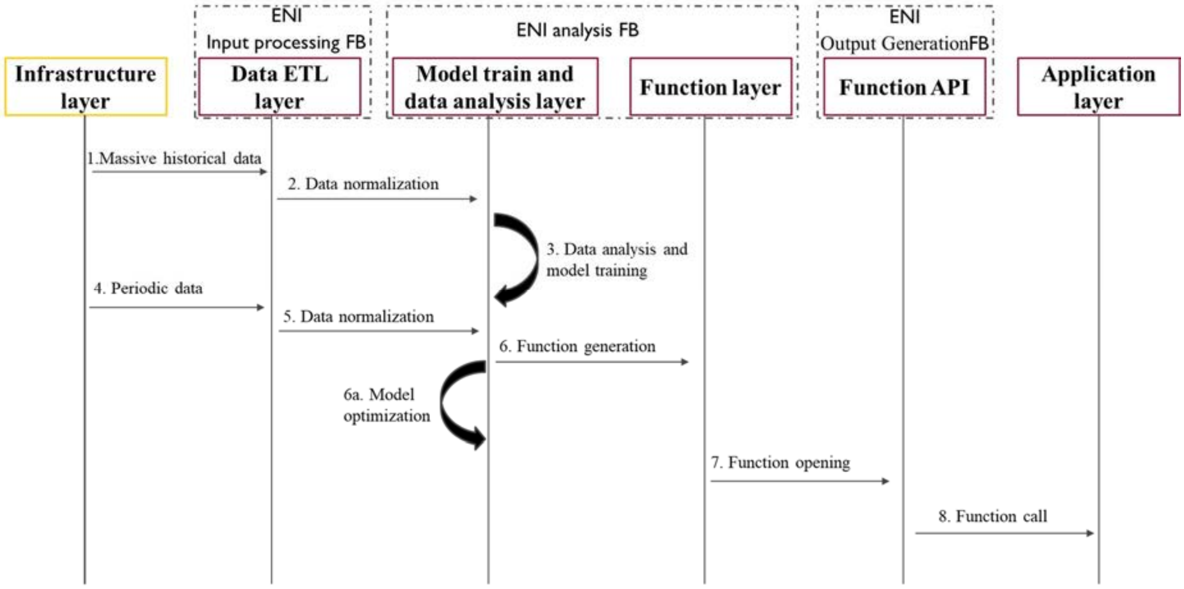


**Figure 5-52b: Flow of information**

Step 1: Infrastructure layer sent massive historical data to data ETL layer.

Step 2: The data ETL layer normalizes the data and sends the data to the model train and data analysis layer.

Step 3: The model train and data analysis layer processes normalization data and a generate user experience model.

Step 4: Infrastructure layer sent periodic data to data ETL layer.

Step 5: The data ETL layer normalizes the data and sends the data to the model train and data analysis layer.

Step 6: After learning, model train and data analysis layer has generated atomic capabilities for customer experience optimization and encapsulated these atomic capabilities in the functional layer.

Step 6a: When the model train and data analysis layer calls the existing knowledge cannot understand the normalized periodic data, it will analyse and learn periodic data to optimize the customer experience optimization model.

Step 7: Functional layer 's atomic capabilities are opened to the public through functional layer API.

Step 8: Application layer implements capability call through functional API.

## 5.4.9 Use Case #3-9: Intent-Driven Home Intranet Management

### 5.4.9.1 Use Case context

Home router hardware has been upgraded for generations. Currently, the hardware performance of a home router is capable of more complex jobs, other than routing. Meanwhile, the latest router system normally is compatible with complex jobs, such as workload balancing, resource orchestration, etc. However, to accomplish these jobs mentioned above, the user currently should have some computer and network management skills. This prerequisite has become a barrier to making the home Intranet develop further. To overcome this difficulty, the system needs to be more intelligent so that a user without relative knowledge can express his/her demands and the router responds correctly.

With the vigorous development of the Internet of Things and the emergence of various intelligent devices, the concept of 'Smart Home' has gradually become one of the foci of the industry. One of the core ideas of Smart Home is to manage the family Intranet according to user intent policies.

### 5.4.9.2 Description of the Use Case

#### 5.4.9.2.1 Overview

The access of many smart devices to the home network brings many new services such as Extended Reality (XR), Network Attached Storage (NAS), Ultra High Definition (UHD) video streaming; these new services gradually complicate the structure of the home network and requires more dedicated network management.

Normally, network management is a very complex process for a home user, which takes a considerable amount of time for learning the relative knowledge to accomplish family Intranet management, a user is required to:

- Have basic network knowledge. The user needs to get familiar with common network protocols and packet transmission mechanisms. The user needs to be able to come up with a reasonable management plan.
- Be familiar with Network Management. The user needs to have experience in managing networks so that the user can understand what different network telemetry mean, which kind of data is important to monitor, etc.

#### 5.4.9.2.2 Motivation

The current home network management method is simple. The user should input network policy directly to the network. Thus, there are certain requirements for users' professional skills. However, home users often do not have much experience on network management. Moreover, different from other network management scenarios, since home network users usually do not have much relevant knowledge. Therefore, the user's intention input is usually more obscure than for other use cases. For example, these users describe requirements for the service layer only. To achieve the request, the network needs to orchestrate the resources.

ENI System can translate the intent policy input into a restricted natural language. The introduction of the ENI System in the home network environment can effectively reduce the knowledge requirement threshold for users to manage the network. With the help of an ENI System, users can not only use traditional computer language to operate the network, but also express their expectations for the network or a specific business running on the network by directly using natural language. The ENI System can accurately extract the keywords in the sentences, determine the real needs of users, translate those needs into appropriate network commands, and automatically dispatch them to the corresponding home network devices.

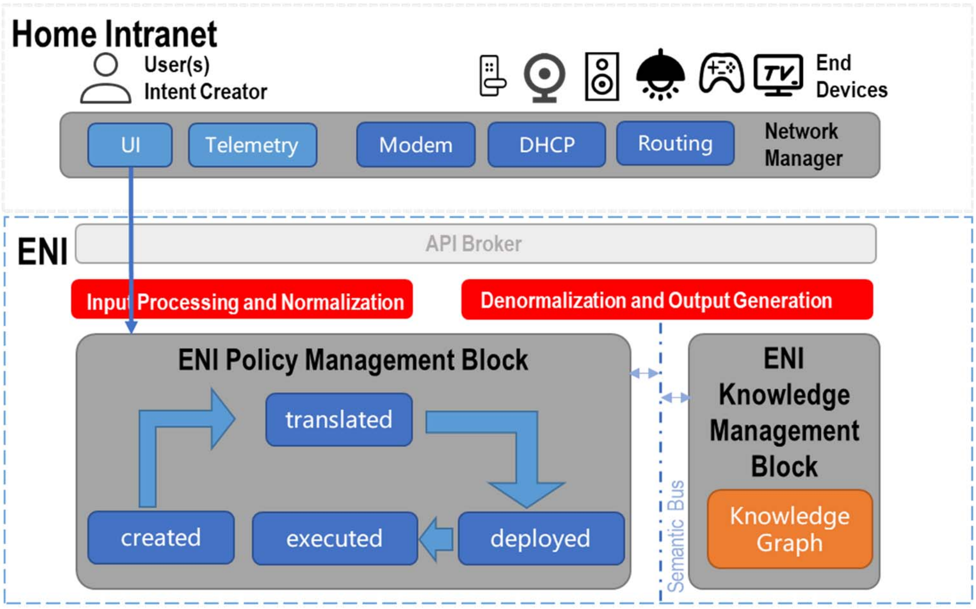


**Figure 5-53: A General Architecture of Home Intranet Management**

#### 5.4.9.2.3 Actors and Roles

**User:** the family Intranet user. Assume in this case, the user has not learned any relevant skills.

**End Devices:** Intelligent devices connected to the family Intranet. Normally, they are laptops, intelligent TVs, cell phones, VR helmets, etc.

**ENI System:** The system that receives intent polices from the users, and translates the intent policies into polices.

**Network Manager:** The system that monitors services provided and changes their configurations according to user intent.

**Running Applications:** Applications that are currently running on any end device.

**Terminated Applications:** Applications that were running on end devices but terminated before the ENI System starts processing a new intent policy.

### 5.4.9.2.4 Initial context configuration

End Devices are connected to the home Intranet and they are all functional.

The home Intranet runs in a stable state.

ENI constructs a knowledge graph with the following attributes and relations and stores it in the knowledge management Functional Block.

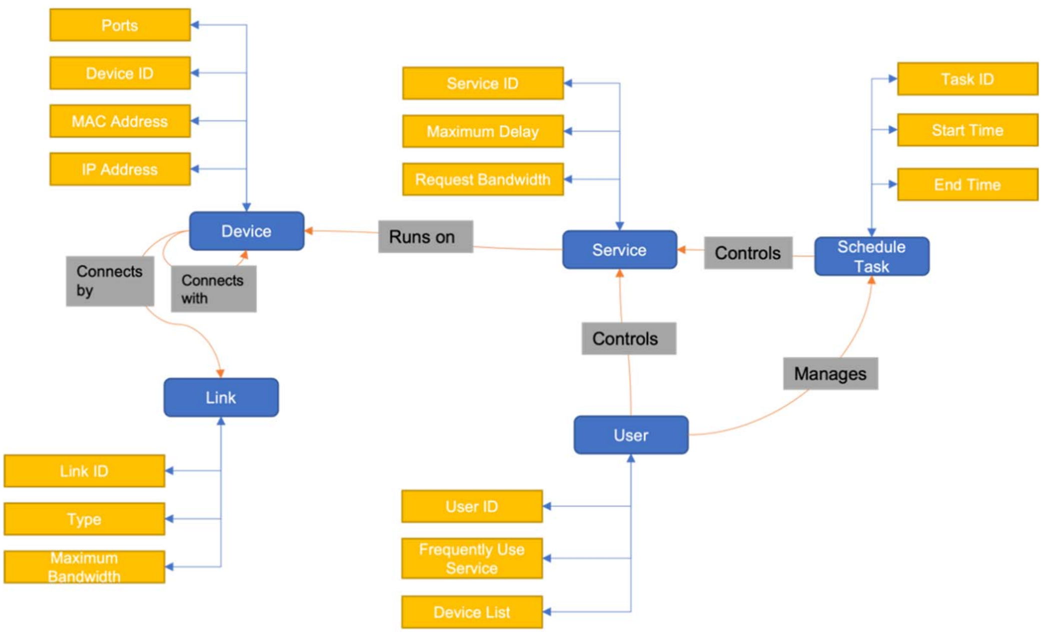


**Figure 5-54: Structure of the Home Intranet Management Knowledge Graph**

### 5.4.9.2.5 Triggering conditions

The user inputs an intent policy which implies a state the user expects the network or running business to achieve.

### 5.4.9.2.6 Operational flow of actions

1) The ENI System receives the new intent policy from users or running applications. Then the ENI System translates the intent to a form that it understands.

2) The ENI System determines if there are any conflicts among new intent policy and processed intent polices. If there are any, jump to step 2a). Otherwise, continue to step 2b).

2a) If the processed Intent Policies were from terminated applications, continue to 2b). Otherwise, jump to 5).

2b) The ENI System translates the new intent policy into suitable recommendations or commands with the assistance of the knowledge graph.

3) The ENI System sends the translated polices to the home router.

4) The home router receives the polices and modifies its resources accordingly.

5) End devices modify their resource consuming strategy to accommodate the change of network.

6) The ENI System writes the information down into a log and reports it to the Network Manager along with recommendations and commands.

7) The user validates whether the application's or user's demand is met.

8) If not, the user may input new intent policies in the form of natural language or a computer programming language to correct the previous intent policy.

Step 7) is optional. If the network state meets the user demand, the step should be skipped.

###### 5.4.9.2.7 Post-conditions

The state of the home Intranet or a business has been changed accordingly.

The user's intent policy has been achieved.

## 5.5 Assurance

### 5.5.1 Use Case #4-1: Network fault identification and prediction

#### 5.5.1.1 Use case context

For a network device or a network service, performance and other problems generally exist before the equipment/service fails. It is important to proactively identify and forecast status of a device/service that is not performing as expected in order for network operation and maintenance management to be able to repair the service before customer requirements are violated. Such identification and predication will need network information to be collected.

This use case takes wavelength division service as an example, which collects information such as FEC_bef, input optical power, laser bias current, and other key factors that can be selected. The information collected can be used to keep track of wavelength division service over time and calculate the device statistics data in a specific time period such as average device downtime in the specified time window. These statistics data can be further used to detect wavelength division service anomaly or improve the accuracy rate for wavelength division KPI anomaly detection.

#### 5.5.1.2 Description of the use case

##### 5.5.1.2.1 Overview

The development of artificial intelligence and big data technologies has brought new chance to the network operation and maintenance management. Big data technology can be applied to analyse huge data generated from network operation and maintenance management, and deep learning method can be used to construct the Intelligent Network Failure Prevention (INFP) system, which can be a sub-system of intelligent analysing and prediction in the ENI System. INFP system can help operators to reduce the OPEX in the network and promote service quality.

##### 5.5.1.2.2 Motivation

Network failure can result in service disruption. The passive strategy is inefficient, and easily lead to long service interruption. By actively learning the health status of history data and intelligent partitioning the current service performance online, AI can be utilized to identify the potential sub-health services and rank these services according to health level. Taking again the example of wavelength division service, one minute rapid optical layer failure location can be achieved. Such a scenario is depicted in Figure 5-55.

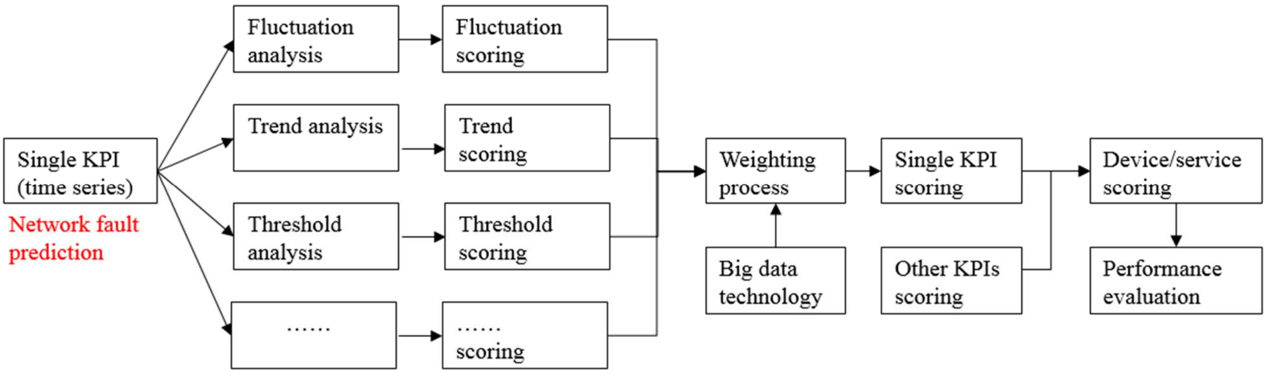


**Figure 5-55: Network fault predication**

#### 5.5.1.2.3 Actors and Roles

- Network Administrators: define threshold for fault prediction.
- Network Performance Analysts: analyses fault prediction report.
- ENI System: collects and analyses data then predicts possible fault and produces detailed report.

#### 5.5.1.2.4 Initial context configuration

Network time series data analysis comprises methods for analysing time series data in order to extract meaningful statistics and other characteristics of the data. Network performance changes over time. ENI System gathers data about the network and monitors the health status of the network.

For network equipment performance evaluation, multiple features are usually extracted from KPI data, such as fluctuation, trend, threshold, etc., and used as the key factors for anomaly analysis.

#### 5.5.1.2.5 Triggering conditions

ENI System detects that network performance degrades below a threshold or the trends of metrics and statistics indicate that a fault may happen.

#### 5.5.1.2.6 Operational flow of actions

1) ENI System calculates the network health indicator and predicts a possible fault, which may happen in future.
2) ENI System outputs detailed information about the fault (e.g. fault probability and fault coverage).

#### 5.5.1.2.7 Post-conditions

Possible fault is identified and reported.

Service provided to customer is verified to be operating correctly.

## 5.5.2 Use Case #4-2: Assurance of Service Requirements

### 5.5.2.1 Use Case context

Nowadays, specific industries such as banking, energy or railroads use dedicated network infrastructures because it is the only way they can guarantee their specific requirements are met. These network infrastructures have huge costs in terms of planning and management, and take several months to be deployed. However, these industries would prefer to have Network Operators deploying and managing these private networks because that is not part of their core business. Moreover, during their lifetime operation, any change to the network infrastructure, no matter how small it is, is a cumbersome task due to the inherent complexity.

To overcome this scenario, Network Operators can replace these dedicated networks by other Virtualised solutions where pinning of virtual resources may be made to physical ones. In that Virtualised context, one of those solutions is the use of the network slicing feature, if slices are capable of meeting the requested strict requirements. In addition, the use of proper resource allocation techniques would also help to solve the situation. However, this approach is very difficult to comply with, because current resource allocation techniques are not able to provide the required performance and assurance with context-awareness capabilities.

### 5.5.2.2 Description of the use case

#### 5.5.2.2.1 Overview

When combining network slices regarding a solution to the situation by making use of the slice/service prioritization and resource allocation concepts, it has to be considered that the dissemination of network slicing across network infrastructures will impact the virtual resource reservation and sharing since, unlike logical resources, physical resources cannot scale or migrate on demand.

NOTE: A network slice may encompass one or more than one services (mapping 1:1 or 1:n). When the mapping is 1:1, prioritization may either be designated by slice prioritization or by service prioritization.

In this use case, only a 1:1 mapping between slices and services is considered, as such slice prioritization will be used.

Considering that each slice may be assigned for a specific type of service or service class, Network Operators will need to enhance their operational systems with the necessary carrier grade assurance capabilities to guarantee the continuous delivery of services characterized by strict requirements. These capabilities are needed to resolve resource allocation conflicts between competing network slices deployed on top of a shared infrastructure in an efficient and dynamic manner. In a shared infrastructure, being aware of a constantly changing context is vital for triggering a set of actions in a timely manner, e.g. scaling resources in order to meet network slice requirements or increase the priority for specific network slices.

A network domain may run several network slices where one of them provides an infrastructure to a specific industry, e.g. an Energy Provider company. This Energy Provider uses the network slices for vital applications that enable them to operate their core business. Because these applications have very strict network requirements, the Network Operator and the Energy Provider establish a customized SLA.

The current Use Case is further described by the following set of components and features.

#### 5.5.2.2.2 Motivation

By using AI and appropriate policies Network Operators will be able to predict potential hazardous situations, where two or more slices are competing for the same resources, and employ preventive measures, e.g. by using resource reservation. In other cases, more specifically when it is not possible to predict a certain scenario in advance, actions still need to be made at runtime in an autonomous way, e.g. by increasing the priority of a given network slice over neighbouring slices. Figure 5-56 provides a pictorial representation of the Use Case just described where the sequence of flow actions representing the resource behaviour in a specific section of an operator network infrastructure is depicted. It is meant to illustrate how an AI-based system is continuously monitoring and is able to predict fault starvation scenarios to trigger the most appropriate and optimal responses for mitigation e.g. slice prioritization enforcement.

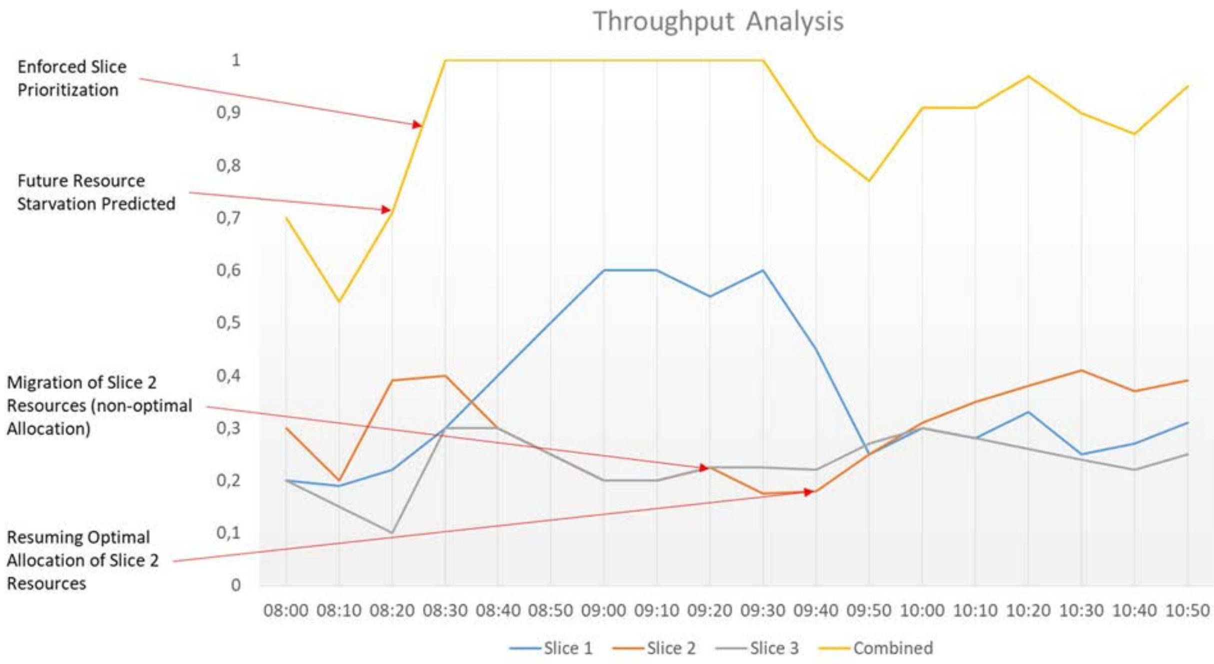


**Figure 5-56: High priority Network Slice Assurance**

#### 5.5.2.2.3 Actors and Roles

The presence of the following actors/entities as well as their associated roles are envisaged in the current Use Case:

- **Customers/Clients:** Specific vertical industries such as banking, energy or railroads that use dedicated network infrastructures.

NOTE: These entities, Customers/Clients, could also be considered as Assisted Systems.

- **Operations Support System/Business Support System (OSS/BSS):** Operational and Business Assisted Systems that belong to the management system of network operators. In this case, they enforce the strict operational and business requirements, which will be translated into slice policies that shall be accomplished in order to fulfil the agreed SLA between the Network Operator and the verticals, i.e. the Energy Provider.
- **Slice Management and Orchestration:** Assisted System administrated by a network operator that manages and orchestrates the life cycle of network slices.
- **Network Operator:** Owner of the Network Infrastructure that is used to provide services to customers/clients.
- **(Shared) Network Infrastructure:** (Shared) Infrastructure Assisted System used to create network slices and maintain their requirements, which delivers raw data to the ENI System during the training phase.
- **Network Slice Instance:** Representation of a network virtual component, which encompasses network functions, capabilities and resources, dynamically provisioned and assigned to a particular type of service or service class that are used by specific industries.
- **ENI System:** System solution used to predict or detect requirements change also involving possible competition for the same shared resources as well as to enforce slice prioritization.

#### 5.5.2.2.4 Initial context configuration

The requirements agreed with customers, formally contracted in SLAs, are communicated by the OSS/BSS Assisted Systems to the Slice Management and Orchestration Assisted System as well as to the ENI System. Furthermore, the ENI System is also notified about the associated policies. At the same time, the Network Slices instances associated to each dedicated network are created and configured accordingly by the OSS/BSS Assisted Systems. All services are running with optimal resource allocation.

The ENI System is operating under normal conditions where raw data is being gathered and processed. Dedicated tools for knowledge extraction, filtering, and fusion are applied to different data sources in order to be combined, filtered, correlated, or otherwise processed to produce new knowledge. On the other hand, context-aware update is also taking place by monitoring the information about the characteristics and behaviour of the environment that ENI interacts with. This context-aware update also enables the ENI System to adapt its behaviour according to changes in context.

#### 5.5.2.2.5 Triggering conditions

At a certain point in time, one slice reveals a considerable deviation from normal resource consumption patterns. Since this slice is deployed over a shared infrastructure where other slices and services are also provisioned, the abnormal behaviour may impact these other slices, including the one established for the Energy Provider, and eventually provoke the violation of the agreed SLA.

#### 5.5.2.2.6 Operational flow of actions

The following sequence of high level actions may be identified after the occurrence of the trigger:

1) The ENI System makes a simulation for the consumption of resources according to the detected spike in a specific zone.

2) The ENI System projection predicts that the spike will impact on the service delivery on what strict requirements is concerned, hence violating past agreed SLAs.

3) In order to preserve the contracted SLAs, the ENI System enforces slice prioritization to guarantee that network slices with strict requirements continue to satisfy those SLAs, since the option of allocating more resources to those slices is not feasible due to a temporary lack of available resources on the above mentioned specific zone.

4) With the change on prioritization for the slices, one of lower priority starts suffering from resource starvation. Once detected by the ENI, in order to mitigate the impact, it makes use of dynamic resource allocation techniques and performs the migration of some resources to a temporary non-optimal location where, however, it is able to better accommodate the network slice.

5) After the spike in resource consumption disappears, if found as appropriate, the ENI System triggers the optimal allocation of resources to the network slice blueprint that was standing before the resource migration; afterwards all services are running with optimal resource allocation.

A more detailed flow diagram, showing an example of a possible solution for the deployment of this use case involving FBs of the ENI Reference Architecture, is depicted in clause 5.5.2.3.3 in order to complement the operational sequence of high level steps just described.

#### 5.5.2.2.7 Post-conditions

The dedicated slices resume normal operation according to contracted SLAs. New knowledge as well as new context begin to be generated once again as normal operation activities are resumed.

### 5.5.2.3 Mapping to ENI reference architecture

#### 5.5.2.3.1 Functional blocks

Table 5-5 contains an example of a set of Functional Blocks (FBs) belonging to ETSI GS ENI 005 [3], which may be used on the implementation of a solution for the scenario described by this use case. For each FB, a brief description of the functionalities that may be accomplished by each one to address a particular feature of the Use Case is made.

NOTE: The applicability of each FB in terms of what is its role in the overall implementation of the Use Case can only be seen as an example based on ETSI GS ENI 005 [3].

**Table 5-5: Mapping of ENI FBs to Use Case functionalities description**

| ENI Functional Blocks | Use Case functionalities description |
|---|---|
| Data Ingestion and Normalization | The purpose of the Data Ingestion FB is to collect data from multiple input sources and implement common data processing techniques so that ingested data can be further processed and analysed by other ENI Functional Blocks.<br>The purpose of the Normalization FB is to process and translate data received from the Data Ingestion Functional Block into a form that other ENI Functional Blocks can understand and use.<br>For implementation purposes, both FBs can be combined, if desired.<br>Upon receiving the uniform data internal format information, the other FBs are entitled to predict/detect abnormal events, i.e. traffic spike or potential resource starvation in the case of this use case. |
| Knowledge Management | The purpose of the Knowledge Management FB is to represent information about both the ENI System as well as the system being managed. It also enables machine learning and reasoning (e.g. by performing inference, correcting errors, and deriving new knowledge).<br>The Knowledge Management framework may make use of dedicated tools for knowledge extraction, filtering, and fusion that may be required in environments where knowledge from different data sources needs to be combined, filtered, correlated, or otherwise processed in order to produce data, information, and knowledge.<br>Upon prediction/detection, or becoming aware, of the occurrence of an event, i.e. a traffic spike or a potential starvation in the case of this use case, this FB may contribute to the delivery of a set of corrective action plans together with other FBs. |
| Context-Aware Management | The purpose of the Context-Aware Management FB is used to describe the state and environment in which a set of entities in the Assisted System exists or has existed. Context consists of measured and inferred knowledge, may change over time, and is continuously updated, hence help to adopt decisions to overcome hazardous situations.<br>Its applicability to the present Use Case relies on the observation and update of the (adapted) behaviour of the network slices associated with the offer of services or type of services according to SLAs.<br>Upon prediction/detection, or becoming aware, of the occurrence of an event, i.e. a traffic spike or a potential resource starvation in the case of this UC, this FB may contribute to the delivery of a set of corrective action plans together with other FBs. |
| Cognition Management | The purpose of the Cognition Management FB is to enable the ENI System to understand normalized ingested data and information, as well as the context that defines how those data were produced. Once that understanding is achieved, the Cognition Framework FB then evaluates the meaning of the data, and determines if any actions should be taken to ensure that the goals and objectives of the system will be met.<br>Upon prediction/detection, or becoming aware, of the occurrence of an event, i.e. a traffic spike or a potential resource starvation in the case of this UC, this FB contributes to the delivery of a set of corrective action plans together with other FBs. |

| ENI Functional Blocks | Use Case functionalities description |
|---|---|
| Situational Awareness | The purpose of the Situational-Awareness FB is to enable the ENI System to be aware of events and behaviour that are relevant to a set of entities in the environment of the Assisted System. This includes the ability to understand how information, events, and recommended commands provided by the ENI System will impact the management and operational goals and behaviour, both immediately and in the near future.<br>The Situation Awareness FB is especially important in environments where the information flow is high, and poor decisions may lead to serious consequences (e.g. violation of SLAs).<br>Upon prediction/detection, or becoming aware, of the occurrence of an event, i.e. a traffic spike or a potential resource starvation in the case of this UC, it works together with other FBs to produce a set of corrective action plans based on information, knowledge, and wisdom.<br>In addition, assisted by the Model-Driven Engineering FB, it also performs the coordination of the analysis and decision making of verdicts, and chooses the optimal plan of action before delivering it to the Policy Management FB. |
| Model-Driven Engineering | The purpose of the Model Driven Engineering FB is to use a set of domain models that collectively abstract all important concepts for managing the behaviour of objects in the system(s) assisted by the ENI System.<br>The MDE approach is meant to increase productivity by maximizing compatibility between Functional Blocks and systems through the reuse of standardized models.<br>In this use case, it assists the Situation Awareness FB in the overall coordination of the analysis and decision making process. |
| Policy Management | The purpose of the Policy Management Functional Block is to provide decisions to ensure that the system goals and objectives are met. Policies are used to provide scalable and consistent decision-making and are generated from data and information received by the Knowledge Management and Processing set of Functional Blocks.<br>Policies may represent recommendations or commands and are conveyed to the Assisted System or its Designated Entity through the Denormalization and Output Generation FBs, i.e. its applicability to the present Use Case relies on the execution of the action plan selected by the Situation Awareness. |
| Denormalization and Output Generation | The purpose of the Denormalization Functional Block is to process and translate data received from other Functional Blocks of the ENI System into a form that facilitates subsequent translation to a form that a set of targeted entities can understand.<br>The purpose of the Output Generation Functional Blocks is to convert data received by the Denormalization Functional Block into a form that the Assisted System or its Designated Entity can understand. This may include defining an appropriate set of protocols, changing the encoding of the data, and other related functions.<br>This FB can be combined with the Denormalization FB, if desired. |
| Lifecycle Management | N/A. |
| Ancillary | N/A. |

### 5.5.2.3.2 Interfaces

Table 5-6 contains an example of a set of Reference Points (RPs) belonging to ETSI GS ENI 005 [3], which may be used on the implementation of a solution for the scenario described by this use case. For each RP, a brief description of the functionalities that may be accomplished by each one to address a particular task of the Use Case is made.

NOTE: The applicability of each RP in terms of what is its role in the overall implementation of the Use Case can only be seen as an example, based on ETSI GS ENI 005 [3].

**Table 5-6: Mapping of ENI RPs to Use Case functionalities description**

| ENI External Ref. Points | Use Case functionalities description (see note) |
|---|---|
| $E_{oss\text{-}eni\text{-}dat}$ | Defines data and acknowledgements exchanged between ENI and the OSS-like Assisted System (or its OSS-like functionality), i.e. data sent to ENI and acknowledged by the Assisted System or by its Designated Entity. |
| $E_{oss\text{-}eni\text{-}cmd}$ | Defines recommendations and/or commands exchanged between ENI and the OSS-like Assisted System (or its OSS-like functionality), i.e. recommendations and commands sent from ENI and acknowledged by the Assisted System or by its Designated Entity. |
| $E_{app\text{-}eni\text{-}ctx}$ | Defines situation- and/or context-aware data and information and acknowledgements exchanged between ENI and the Slice Management FB of the Slice Management and Orchestrator Assisted System (considered as an application). |
| $E_{app\text{-}eni\text{-}kmo}$ | Defines model and/or knowledge information and acknowledgements exchanged between ENI and the Slice Management FB of the Slice Management and Orchestrator Assisted System or by its Designated Entity. |
| $E_{app\text{-}eni\text{-}oth}$ | Defines generic application data and acknowledgements exchanged between applications and ENI, that is neither situation - and/or context-aware data and also is not model or knowledge information. |
| $E_{bss\text{-}eni\text{-}dat}$ | Defines data and acknowledgements exchanged between the BSS-like functionality and ENI, i.e. data sent to ENI and acknowledged by the Assisted System or by its Designated Entity. |
| $E_{bss\text{-}eni\text{-}cmd}$ | Defines data and acknowledgements exchanged between the BSS-like functionality and ENI, i.e. recommendations and commands sent from ENI and acknowledged by the Assisted System. |
| $E_{or\text{-}eni\text{-}dat}$ | Defines data and acknowledgements exchanged between ENI and the Slice Management FB of the Slice Management and Orchestrator, i.e. data sent to ENI and acknowledged by the Assisted System or by its Designated Entity. |
| $E_{or\text{-}eni\text{-}cmd}$ | Defines commands and acknowledgements exchanged between ENI and the Slice Management FB of the Slice Management and Orchestrator, i.e. recommendations and commands sent from ENI and acknowledged by the Assisted System or by its Designated Entity. |
| $E_{inf\text{-}eni\text{-}dat}$ | Defines data and acknowledgements exchanged between the infrastructure and ENI, i.e. data sent to ENI and acknowledged by the Assisted System or by its Designated Entity. |
| $E_{inf\text{-}eni\text{-}cmd}$ | Defines recommendations and/or commands, and acknowledgements, exchanged between the infrastructure and ENI, i.e. recommendations and commands sent from ENI and acknowledged by the Assisted System or by its Designated Entity. |
| NOTE: Functionalities description is an adaptation of the text provided in Table 6-4 of ETSI GS ENI 005 [3] taking into account the specific related functionalities for the present Use Case. | |

### 5.5.2.3.3 Flow of information

Figure 5-56a depicted below illustrates a possible solution for the deployment of the Use Case that has been described in this clause, by making use of FBs that belong to ETSI GS ENI 005 [3]. The functional operational procedures identified in clauses 5.5.2.2.4 to 5.5.2.2.7 are accomplished by the functionalities associated to each of the selected FBs. In addition, only the most relevant actions are shown as well as the exchange of messages between them.

In Figure 5-56a, the flow diagram is split into two parts because the deployment of the Use Case actually implies the detection and subsequent handling of two events: spike in resource consumption and resource starvation. The notes referred in the diagrams may be found in the step-by-step description of Figure 5.56a.

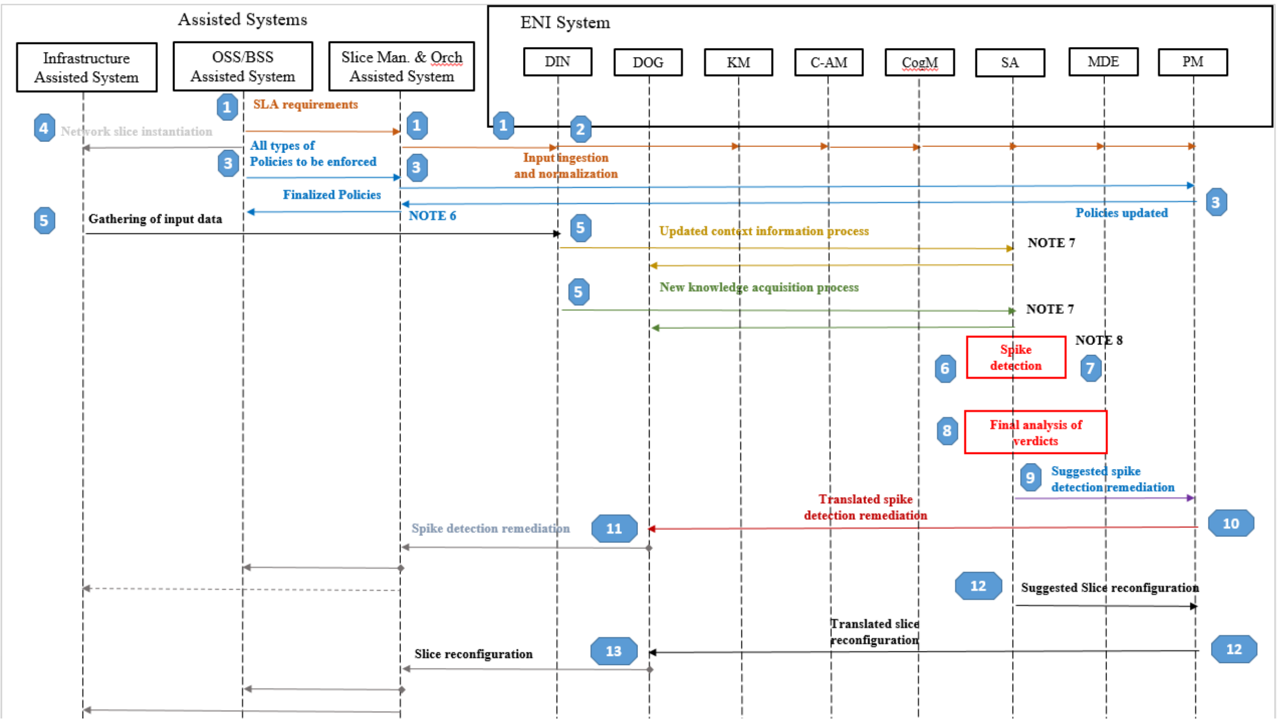


**Figure 5-56a: Part 1: Spike detection and slice prioritization flow diagram**

By using a step-by-step approach, the actions are described as follows:

Step 1: The scenario for this use case starts with the formal settle down of SLAs with the customers, which implies the fulfilment of agreed strict requirements. These have to be communicated by the OSS/BSS Assisted Systems to the Slice Management and Orchestration Assisted System as well as to the ENI System.

Step 2: The SLA requirements are received by the Data Ingestion and Normalization FBs, which perform its conversion into a normalized form in such way that can then be analysed and understood by the other FBs of the ENI System for further processing.

Step 3: Furthermore, the ENI System shall also become aware of the possible associated policies, which may be enforced in all these FBs subsequently by the OSS/BSS Assisted System.

NOTE 1: The OSS/BSS Assisted System is entitled to enforce not only these associated policies but also any other type of slice policies at any time, see step 3 in Figure 5-56a.

Step 4: At the same time, the Network Slices instances associated with each dedicated network are created and configured accordingly by the OSS/BSS Assisted System in the Infrastructure Assisted System. After that, all services are running with optimal resource allocation.

Step 5: The ENI System enters then in its normal operation where raw data starts being gathered and processed after the conversion made by the Data Ingestion and Normalization FBs. The overall behaviour of the FBs, in this phase, is indicated in Table 5-4.

Step 6: At a certain point in time, one slice reveals a deviation from normal resource consumption patterns and, since it is deployed over a shared Infrastructure Assisted System, where other slices and services are also provisioned, the abnormal behaviour may impact those other slices. Any events (e.g. alarms) or data that indicate this abnormal behaviour are received by the DIN FB. It may use various processes, including simulating the consumption of resources corresponding to this abnormal behaviour. After prediction/detection of the occurrence, other FBs, e.g. KM, C-AM, CogM, SA and MDE, become aware of the situation by extracting the information from the semantic bus. The SA FB stores all the parameters and associated configurations that make part of the network slice blueprint that was standing before the detection of the spike.

Step 7: Upon taking care of the abnormal occurrence, each one of the involved FBs, including the Situational Awareness and the Model-Driven Engineering FBs, work together and may use a variety of algorithms to process data, and generate information, knowledge, and wisdom in order to prevent network slices from violating the agreed SLAs. In this way, they reach verdicts represented by action plans to be taken upon. Since for the case of the present UC, the option of allocating more resources to those slices is not feasible due to the lack of available resources on the affected specific zone, slice prioritization shall be used instead.

NOTE 2: A process involving back and forth messages, between all the concerned FBs and the MDE FB, may take place at this point. However, in order to simplify the drawing, it is not depicted.

Step 8: Final analysis of verdicts conflicts, and evaluation of consequences, is coordinated by the Situational Awareness FB, which is in charge of the harmonization of the output decisions. In the execution of this task, it is assisted by the Model-Driven Engineering FB, and chooses the optimal plan of action avoiding the possibility of different plans having conflicting actions.

Step 9: Once performed the evaluation of verdicts, and the selection of the ultimate action plan, the Situational Awareness FB sends the output data to the Policy Management FB.

Step 10: The responsibility of the PM is to take the verdict (i.e. the final single action plan selected by the SA FB) and translate it into a set of Policies. After that, the PM FB forwards the (internally normalized) data to the Denormalization and Output Generation FBs.

Step 11: In its turn, before sending the result to the external Assisted Systems, the Denormalization and Output Generation FBs process the output data received from the Policy Management FB, and translate it to an external format understandable by them.

Step 12: Finally, the Situation Awareness FB, which had stored the parameters and associated configurations that make part of the network slice blueprint, initiates the reset to the conditions that applied when the spike was detected by sending an appropriate message to the Policy Management FB. In its turn, the PM FB translates it into a set of policies and sends the result to the Denormalization and Output Generation FB.

NOTE 3: The task described in step above is not usually performed since, typically, in network management, the action plan changes configurations in order to correct the problem, which implies returning an optimized state. After applying the plan of actions, the original state makes no sense anymore, however, the reset is performed in this use case just to show in the flow diagram the way it is performed.

Step 13: Once received the message, the Denormalization and Output Generation FB performs the translation of its contents to a format that can be understand by the external Assisted Systems.

As a result of the projection, a resource starvation is also predicted on a specific zone of the infrastructure that is supporting several network slices.

The handling of this situation is shown in the second part of the flow diagram as depicted in Figure 5-56b.

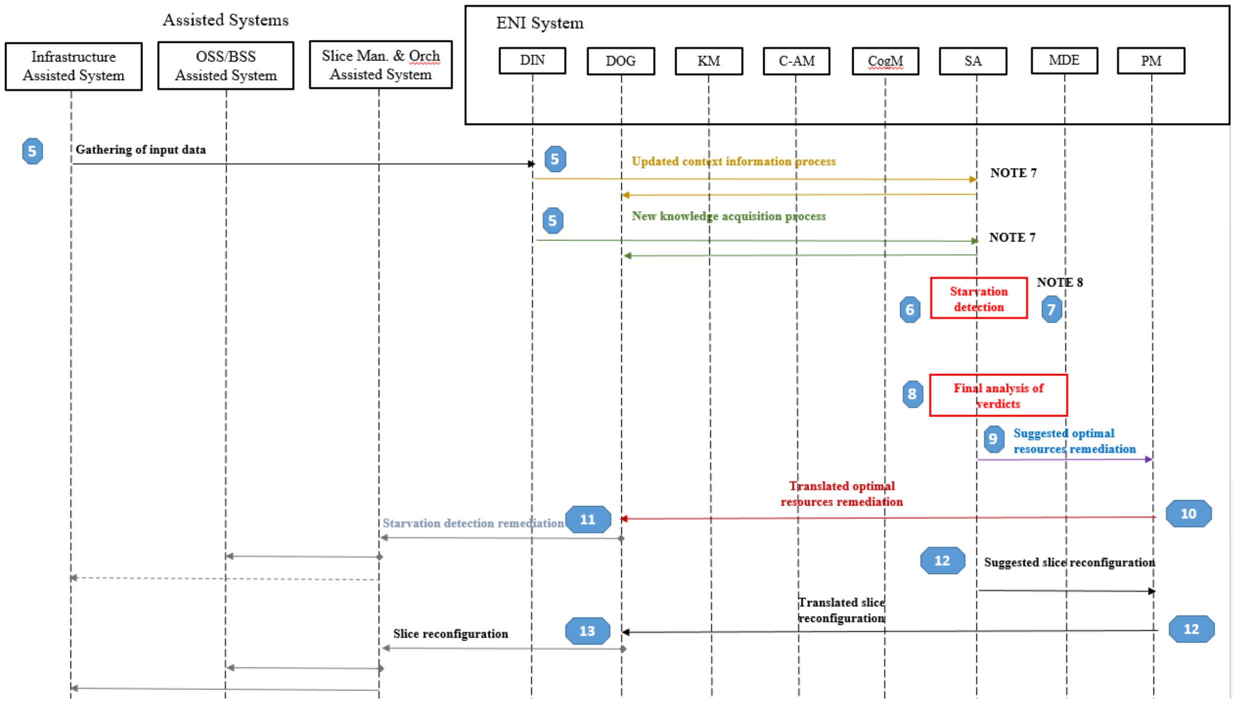


**Figure 5-56b: Part 2: Starvation detection and optimal resources reallocation**

Again, the Situation Awareness FB, to mitigate the impact, may work together with other relevant FBs by making use of dynamic resource allocation techniques and reach a set of verdicts that are oriented to perform the migration of some resources to a temporary non-optimal location where, however, it is able to better accommodate the network slice. As for the spike occurrence, final analysis of verdict conflicts, and evaluation of consequences, is coordinated by the Situational Awareness FB with the assistance of the Model-Driven Engineering FB.

After the spike affecting resource consumption disappears, the Situational Awareness FB triggers the optimal allocation of resources to the network slice blueprint that was standing before the resource migration.

In the end, all services are according to the agreed SLA.

## 5.5.3 Use Case #4-3: Network fault root-cause analysis and intelligent recovery

### 5.5.3.1 Use case context

Traditional fault maintenance requires manual processing. The cost is high, and the fault locating efficiency is low and the period is long. It is hoped that applying machine learning algorithms in network fault root-cause analysis and intelligent recovery to form a more efficient solution, shorten the time of fault recovery and improve the efficiency of network maintenance.

When faults occur, AI algorithm (e.g. Decision Tree Algorithms) is used to the calculates the fault self-recovery policy with alarms data, network topology data, network service data collected from the Monitoring System (MS). Then the fault self-recovery operation is delivered to network through the Multi-vendor Command Platform (MCP). Self-recoverable faults can be quickly recovered and users are unaware of the faults. If a fault cannot be rectified, accurate diagnosis can be performed to locate the root cause (e.g. Big Data Mining Algorithms, Deep Learning Algorithms) and help engineers quickly rectify the fault.

### 5.5.3.2 Description of the use case

#### 5.5.3.2.1 Overview

There are many difficulties and challenges in network operation and maintenance work, including:

- With the increasing scale and complexity of telecommunication network, operators not only have to monitor a large amount of real-time information generated by various highly integrated devices, but also need to deal with massive alarm data. In order not to reduce user experience, the faults should be quickly recovered.
- For the fault alarms, the current method is to complete fault diagnosis and recovery by manual execution of instructions, or need maintenance personnel to check on site, and then deal with the fault. This method has long recovery time, low operation and maintenance efficiency.
- Fault recovery work is complex. The experience cannot be shared, because the current fault recovery methods is different in different regional and most maintenance personnel change even every day.

It is imperative to achieve root cause analysis and self-recovery in maintenance work, and reduce the workload of maintenance personnel by using AI technology and ENI System.

#### 5.5.3.2.2 Motivation

For solving the above problems and challenges, based on AI techniques the Use Case analyses and processes network alarms intelligently. The analysis results are feed to the external operation system (e.g. Multi-vendor Command Platform and monitoring system) to improve the level of network maintenance intellectualization.

Such a scenario is illustrated in Figure 5-57. The following specific functions and objectives is proposed in the ENI System in this use case:

- Scenario analysis: This module is aligned with the knowledge & models function block in ETSI GS ENI 005 [3]. Building the base of self-recovery fault scenario based on the historical fault recovery data. When faults occur, the fault scenario analysis function of ENI System is used to judge the self-recovery faults. This part of the fault does not dispatch tickets to deal with, thus reducing the number of tickets, reducing the cost of maintenance.
- Decision-making: This module is aligned with the knowledge & models function block in ETSI GS ENI 005 [3]. Calculates the fault self-recovery policy by using the decision-making model in ENI System, thus the time of artificial judgment is saved, and the fast recovery of fault is realized.
- Big Data Mining: This module is aligned with the knowledge & models function block in ETSI GS ENI 005 [3]. The data mining model in ENI System is used to achieve automatic generation of alarm correlation rules.
- RCA and SIA: This module is aligned with the knowledge & models function block in ETSI GS ENI 005 [3]. The function of Root Cause Analysis (RCA) model is to identify root alarm and derivative alarm based on the correlation rules. The Service Impact Analysis (SIA) model is used to find the correlation of root alarms and fault. Combining RCA and SIA functions to achieve the accurate location and delimitation of root cause fault, and then achieve accurate dispatch of tickets avoiding the errors and invalid tickets, improve manual fault recovery efficiency.

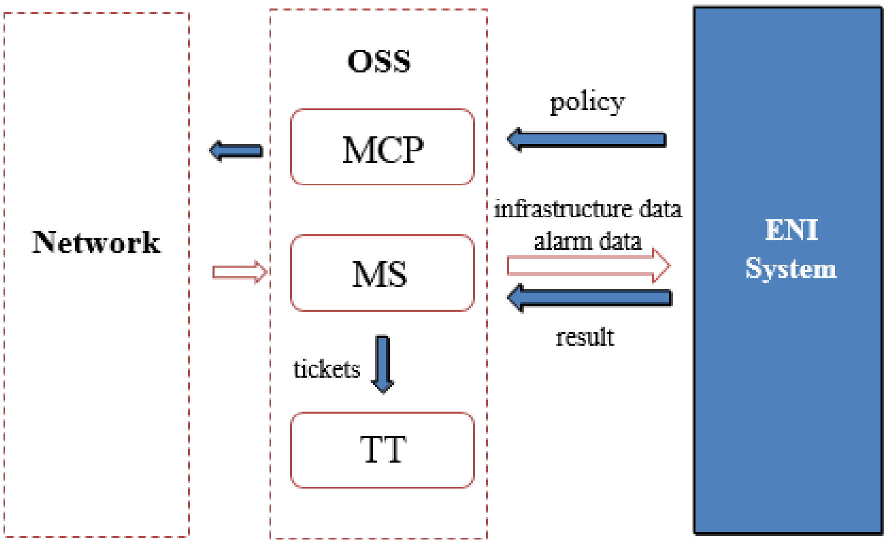


**Figure 5-57: Scenario of AI and ENI System enabled network fault root-cause analysis and intelligent recovery**

### 5.5.3.2.3 Actors and Roles

- MS: Monitoring System, monitor 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 ENI System. Generating tickets based on the result of ENI Feedback.
- TT: Trouble Ticket, issue the ticket and handing by maintenance personnel.
- MCP: Multi-vendor Command Platform, responsible for command message passing from ENI System to the network.
- ENI System: system solution used to receive alarm data and network infrastructure data from MS; to analysis the fault scenario, calculates the fault self-recovery policy; to locate the root cause of network fault.

### 5.5.3.2.4 Initial context configuration

- Defining the analysis results formats of ENI System.
- Training intelligent decision model with the historical fault recovery data.
- Collecting alarm data and network infrastructure data in real-time.
- Building the base of alarm correlation rules and defining the parameters of data mining model, including time window, support and confidence.
- Related external systems connects to ENI System for acquiring decision-making and root-cause analysis results.

### 5.5.3.2.5 Triggering conditions

In this use case, as long as the ENI System receives the fault alarm dispatched by monitoring system, the function of fault scenario analysis and intelligent decision-making will be triggered. When the fault alarm cannot be self-recovery, RCA and SIA functions will be triggered, and the analysis results will be feedback to the monitoring system.

##### 5.5.3.2.6 Operational flow of actions

ENI System with the AI capability, including scenario analysis, big data mining, decision- making, RCA&SIA, is intended to achieve root cause analysis and self-recovery in maintenance work, and reduce the workload of maintenance personnel. This kind of mechanism is realized with the following flow of activities:

- ENI System receives the fault alarm uploaded from the monitoring system, then it starts the fault scenario analysis function, and determines whether the fault is self-recovery.
- For the self-recovery fault, ENI System uses the intelligent decision-making model to calculate the fault self-recovery policy, and output the policy to MCP. The monitoring system monitors whether the fault has been recovered and feeds back to the ENI System, which can iteratively optimize the intelligent decision-making model through feedback information.
- Otherwise, ENI System triggers the functions of RCA and SIA with the correlation rules and network infrastructure data to locate the root cause of the fault.
- ENI System output the analysis results to the monitoring system to generate trouble ticket. Then trouble ticket system will issue the ticket and handing by maintenance personnel.
- If the fault is recovered, the result is fed back to ENI System to form a closed loop.

##### 5.5.3.2.7 Post-conditions

- The base of fault scenarios is constantly enriched, and the application scope of intelligent decision-making model is improved, through the continuous learning of historical data.
- External monitoring system realizes intelligent monitoring by utilizing the alarm data analysis ability of ENI System.
- The base of alarm correlation rules is improving, through the mining of historical alarm data by ENI System. Then the accuracy of root cause analysis is increasing.
- The parameters of data mining model can be adapted according to the results of root cause analysis.

### 5.5.4 Use Case #4-4: IP Network Congestion Prediction and Prevention

#### 5.5.4.1 Use case context

With the evolvement of Internet based on TCP/IP, the scale, users and traffics of it have experienced an explosive growth since 1990's. The network congestion has become more serious and complex due to the ever-increasing network application types and dynamic network parameters such as active sessions and round trip time. Congestion often results in decline of Quality of Service (QoS) in terms of transmission delay and throughput, while the network resource utilization like bandwidth and buffers are also affected seriously. The congestion control is always a hot spot in the field of network research.

Most current network solutions focus on post event congestion optimization, such as relying on TCP end-to-end congestion control algorithms. However, this approach will not be able to meet the increasing demands of various applications in complex networks in the future. Therefore, it is necessary to explore more intelligent predictive and proactive solutions to reduce congestion events and reduce the pressure of complex network operation and maintenance.

### 5.5.4.2 Description of the use case

#### 5.5.4.2.1 Overview

The main goal of network congestion control is to control the data traffic entering the network, ensure that the communication subnet is not overwhelmed by the data flow sent by users, and make reasonable use of bottleneck resources. The basic mode of network congestion control nowadays is that the end system is responsible for responding to congestion, while the router is responsible for monitoring network congestion. Congestion signals are perceived by the source system, and then the load injected into the network is adjusted. However, this mode cannot meet the customer experience needs of agile business opening and "zero-fault" perception, and achieve the vision of ENI's "Better customer experience&Improved QoE of service".

In order to break through the existing network congestion operation and maintenance optimization mode and process, and reconstruct a more intelligent and automated operation mode, this case proposes a dual closed loop technology scheme of task closed-loop and intent closed-loop:

- Task closed-loop: In the absence of new business requirements from users, based on the network performance data and network resource data collected by Operation and Maintenance Centre (MOC) and Resource Management Centre (RMC) in the Operation Support Systems (OSS), uninterrupted traffic prediction and current network resource evaluation are completed. Once abnormal trends or congestion faults are detected, optimization strategies are automatically generated through congestion warning and control algorithms, and optimization instructions are issued by the Multi-vendor CMD Platform (MCP), Thus achieving task closed-loop at the network layer.
- Intent closed-loop: In the case of new business requirements raised by users, the collection of user business requirements/intentions is completed through the unified portal provided by Business Support Systems (BSS). Technologies such as deep learning and semantic recognition are used to analyse and predict traffic on topology links, and analyse user intentions. Complete congestion warning based on traffic prediction results and intent analysis results. If a warning occurs, the congestion control algorithm will automatically generate an optimization strategy, and the simulation preview of the optimization strategy will be completed through digital simulation technology. Based on the simulation results, user intention feedback and adjustment of the optimization strategy will be completed, thereby achieving intent closed-loop at the business layer.

#### 5.5.4.2.2 Motivation

This case refers the options for communication between a System ENI and the Assisted System in ETSI GS ENI 005 [3]. ENI provides the necessary AI capabilities to BSS and OSS, achieving a full closed-loop process from user demand intention analysis, traffic prediction, congestion control to user feedback, hoping to significantly improve the digital capabilities and operational efficiency of operators.

Such a scenario is illustrated in Figure 5-58. The following specific functions and objectives is proposed in the ENI System in this use case:

- Data Ingestion& Normalization: This module is aligned with the Data Ingestion Functional Block and Data Normalization Functional Block in ENI architecture ETSI GS ENI 005 [3]. This module is aligned with the knowledge & models function block in ENI architecture ETSI GS ENI 005 [3]. Collect network business data, network performance data, and network resource data through OSS and BSS, and complete data cleaning, normalization, and fusion.
- Intent analysis: This module is aligned with the Cognition Management Functional Block in ENI architecture ETSI GS ENI 005 [3]. Using semantic recognition technology to convert the user business requirement/intention that collected through the user portal into network language.
- Traffic prediction: This module is aligned with the Context-Aware Management Functional Block in ENI architecture ETSI GS ENI 005 [3]. By mining specific patterns and data characteristics in historical network data, predict network traffic for a period of time in the future, and anticipate the occurrence of network congestion.

- Resource assessment: This module is aligned with the Model Driven Engineering Functional Block in ENI architecture ETSI GS ENI 005 [3]. ENI will input the network resource data collected through RMC (e.g. network topology, link and service latency, jitters, packet loss rate, etc.), combined with user intent analysis results, into the evaluation model to evaluate whether the network resources meet the business scenario.
- Congestion Warning and Control: This module is aligned with the Model Driven Engineering Functional Block and Policy Management Functional Block in ENI architecture ETSI GS ENI 005 [3]. Determine whether to trigger congestion warning mechanism by combining traffic prediction and resource assessment results. If congestion warning is triggered, the automatic generation of congestion optimization strategy is completed. This is the core function of implementing intent closed-loop.

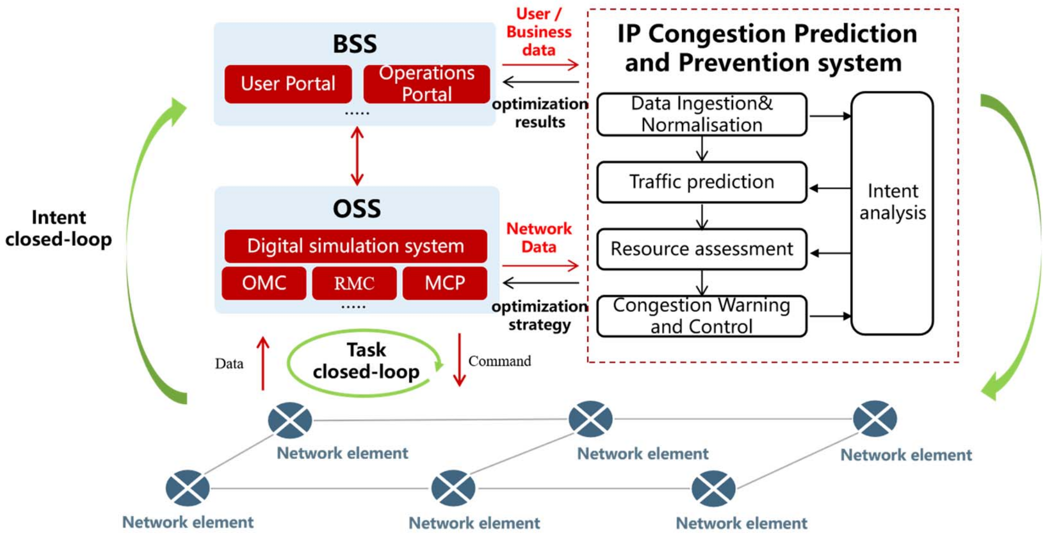


**Figure 5-58: Scenario of ENI System enabled IP network congestion prediction and prevention**

#### 5.5.4.2.3 Actors and Roles

The key actors are listed below:

- ENI System: System that could predict network congestion and formulating optimization strategies; provide results to external system in predefined formats.
- BSS: system that provides user portal and operations portal.
- OSS： system that provides Digital simulation system, OMC, RMS and MCP.
- Digital simulation system：Build the basic and functional models in the digital twin network by multi-dimensional modelling of existing network data. Implement simulation verification of congestion optimization strategies.
- OMC: responsible for collecting network performance data such as IP network traffic and alarms, etc.
- RMC: responsible for collecting network resource data such as network topology, device configuration, bandwidth, link and service latency, etc.
- MCP: responsible for command message passing from ENI System to the IP network.
- IP Network：implementing data transmission and sharing through IP protocol.

#### 5.5.4.2.4 Initial context configuration

- Defining the analysis results formats of ENI System.

- Training traffic prediction model, resource assessment model and congestion control model with the historical multi-source data.
- The corresponding business template has been formed by abstracting user intent requirements, covering the user's business activation process.
- Collecting network performance data and network resource data in real-time.
- Related external systems connects to ENI System for acquiring the optimization strategies and the results of traffic prediction, resource assessment and digital simulation.

##### 5.5.4.2.5 Triggering conditions

- ENI perceives that users have new business needs and intentions
- The traffic prediction result exceeds the set threshold.
- After resource assessment, it is determined that the current network resources do not meet the business scenarios and user intentions.
- A new congestion fault has occurred.

##### 5.5.4.2.6 Operational flow of actions

ENI System with network congestion prediction and prevention function is intended to improve the efficiency of IP network operation and enhance the customer experience of IP services. This kind of mechanism is realized by introducing the new AI capability (e.g. normally based on the deep-learning algorithm and architecture), with the following flow of activities.

1) The ENI System initials a process to receive training data from database and processes the data to make dataset.
2) The ENI System initials a process to train the AI prediction module according to the dataset which have been well-processed in 1).
3) The ENI System utilizes traffic prediction function to analyse and predict the traffic of each link in the IP network.
4) The ENI System analyses user intentions and combines traffic prediction results for congestion warning.
5) The ENI System formulates optimization strategies for early warning links based on intelligent congestion control algorithms, and outputs the optimization strategies to the digital twin simulation network.
6) The optimization results are simulated and rehearsed in the digital simulation system before the strategy is issued.
7) Monitor the optimization results through the OSS system and provide feedback to ENI and users to verify whether the task and intention closed-loop have been achieved

##### 5.5.4.2.7 Post-conditions

- Significantly improved customer experience.
- Realize end-to-end and hop by hop measurement of network traffic at the business level, and quickly restore real-time business paths through digital simulation.
- Realize second level intelligent routing adjustment, and implement "zero- fault" IP service&business activation

## 5.5.5 Use Case #4-5: Fault detection and diagnosis for IDC infrastructure

### 5.5.5.1 Use Case Context

The IDC infrastructure ensures the reliable operation of IDC equipment. The failure and abnormality of IDC infrastructure, such as network and storage facilities, UPS, or cooling systems, can cause the collapse of computing and network services, which leads to significant economic losses. Traditional fault detection and diagnosis use reactive and manual strategies, resulting in delayed responses and high maintenance costs. Artificial Intelligence, especially unsupervised deep learning, can automatically detect faults and identify equipment failures. These technologies offer promising solutions to these challenges.

### 5.5.5.2 Description of the Use Case

#### 5.5.5.2.1 Overview

The current practices for fault detection and diagnosis for IDC infrastructure include:

- Conduct health maintenance of IDC infrastructure by manual monitoring and periodic checks, which cannot identify and locate faults in advance.
- Existing AI-driven fault diagnosis for IDC infrastructure usually requires a large amount of labelled data. These data indicate the fault or non-fault of infrastructure and are used to train the diagnostic model. However, obtaining fault data is often challenging in practice.
- Existing AI-driven fault diagnosis of IDC fails to consider the cumulative impact of the time series to a specific fault, which cannot detect faults in advance based on early anomalies.

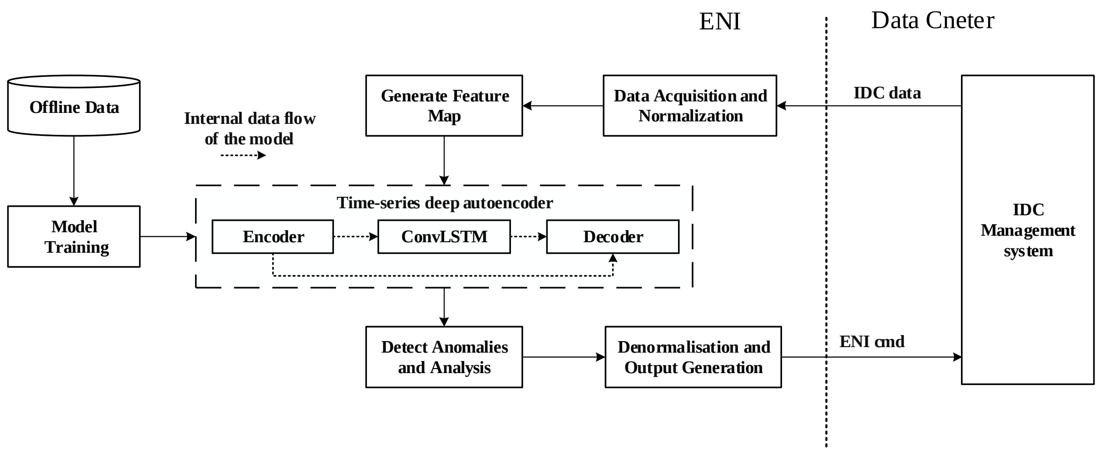


**Figure 5-59: Scenario of AI and ENI system enabled IDC infrastructure fault detection and diagnosis**

#### 5.5.5.2.2 Motivation

The ENI system can be used for fault detection and diagnosis for the IDC infrastructure. As shown in Figure 5-59, the health status of the device can be automatically monitored and identified by using unsupervised deep learning. The following specific functions and objectives are proposed in the ENI system:

- **Condition monitoring:** The ENI system can automatically collect and analyse a large amount of normal data from the IDC infrastructure to establish its condition monitoring.
- **Fault Detection:** The ENI system uses the normal data and the unsupervised time-series deep auto-encoder to identify patterns and anomalies, which indicate potential faults when the operating status of the system exceeds the normal range to a certain extent.
- **Fault Identification**: The ENI system analyses abnormal data and identifies related devices or features causing data anomalies, which locate the specific location of the fault.

#### 5.5.5.2.3 Actors and Roles

- **Operator:** Manages the IDC and confirms health management policies, including the threshold determining the operating status of the equipment as a fault and maintenance schedules.
- **ENI System:** Analyses IDC infrastructure data, detects abnormal equipment operation, identifies malfunctioning components and causes, and provides monitoring and diagnostic results.
- **IDC monitoring system:** Supplies infrastructure operation data, including network and storage facilities, UPS, and cooling systems, to the ENI system and executes environmental changes.
- **Maintenance Team:** Responds to ENI system fault alerts and performs required maintenance actions.

#### 5.5.5.2.4 Initial Context Configuration

- Continuous operation of the IDC infrastructure system.
- Abnormal threshold that is used to determine system anomalies.
- The ENI system performs initial actions related to the collection of infrastructure system information, the use of AI algorithms, and service pattern learning.
- Related external systems connect to the ENI system to provide real-time infrastructure system data and acquire fault detection and identification results.

#### 5.5.5.2.5 Triggering Conditions

- The ENI system predicts that an IDC component is likely to fail within a specific time frame based on historical data and current health status.
- The ENI system receives a manual trigger from an operator to perform a health check on specific IDC components.
- The ENI system detects a change in the operational environment that may affect the health of IDC components.

#### 5.5.5.2.6 Operational Flow of Actions

1) The ENI system acquires operational data from the IDC infrastructure. The obtained data is a type of time series and describes multiple operating status information at multiple time points during the regular operation of the device.
2) The ENI system calculates the correlation between different time-series features pairwise, which form a time-series feature correlation map.
3) The ENI system detects faults using an unsupervised time-series deep auto-encoder, which consists of an encoder with multiple convolution layers, multiple LSTM modules, and a decoder with multiple deconvolution layers.
4) The ENI system uses the encoder to extract key features layer by layer from the feature correlation map. Extracted features from each convolution layer are then fed into each LSTM module to capture time series relationships. The multidimensional time series information by each LSTM is sent to the decoder's deconvolution layers to restore and reconstruct a feature map of the same size as the input data.
5) The ENI system trains a diagnostic model, and the difference between the input feature map and the reconstructed feature map is within a certain range when the system is running normally.
6) When the similarity between the input feature map and the reconstructed related feature map falls within the specified threshold range, the ENI system determines that the target device to be detected is not a fault.
7) If the similarity falls outside the threshold range, the ENI system determines that the target device is a fault.

8) When the ENI system determines that the IDC infrastructure is faulty, it analyses the feature map to seek related devices and identify the fault location. Next, the ENI system issues the fault ticket, which is then handled by maintenance personnel.

##### 5.5.5.2.7 Post-conditions

- The fault scenarios data is constantly enriched, and the application scope of the fault detection model is improved through the continuous learning of historical data.
- The threshold parameters of the fault detection model can be adapted according to the differences in fault scenario data.

## 5.6 Network Security

### 5.6.1 Use Case #5-1: Policy-based network slicing for IoT security

#### 5.6.1.1 Use Case context

In the near future, it is expected that smart cities will be built by using a very large number of IoT devices, where a significant number of them will be connected through 5G. These devices will play a vital role in the deployment of various services (e.g. civil protection or other services provided by the municipality, where each service will have its own target use and different device requirements).

To support this massive deployment of devices, the use of network slices will enable their aggregation either by functionality (e.g. security or city operations management support) or by other types of lower level requirements, such as low latency and high bandwidth.

In this context, the handling of Distributed Denial of Service (DDoS) attacks plays a crucial role as those devices are usually meant to be part of the support to applications/services related to social interest.

NOTE: As an example of the severity of damage that these types of devices can achieve in such environments, consider the October 2016 IoT incident, where several different devices where infected with a Botnet malware designed to perform a DDoS attack. The initial reports point to an attack that roughly doubles previous massive attacks, all thanks to the nature of IoT, where a huge number of devices deployed in a distributed way can be used to target specific network infrastructures.

One of the key benefits of the network slicing concept from the IoT perspective, is that it adds value by offering network and cloud resources that can be used in an isolated, disjunctive or shared manner. In this context, network slicing can be used to support very diverse requirements imposed by IoT services as well as flexibility and scalability to support massive connections of different natures.

Different slices may be used and re-tasked to accommodate changes in context. This requires coordination and management of each slice. It is recommended that one or more AI algorithms are used to pre- and/or post-process the information gathered prior to executing a set of policy rules to manage a set of slices. In addition, the use of different AI algorithms to monitor the execution of the policy rules is recommended to ensure that the new behaviour of the set of slices is correct. The use of AI at these different places in the control loop is necessary to support the integration of millions of devices in complex topologies and distinct communication patterns, while still guaranteeing infrastructure security and optimal resource usage.

The use of AI in concrete scenarios addressing specific situations that involve DDoS attacks enables the ENI System to provide automatic and dynamic responses in different contexts. For example, when a set of IoT devices become infected, they may cause a service degradation or disruption; hence, they need to be isolated in order to be repaired or replaced. This also prevents the spreading of malware.

### 5.6.1.2 Description of the use case

#### 5.6.1.2.1 Motivation

One use of machine learning in the ENI System is to detect specific traffic patterns indicating DDoS or other type of attacks. This is because the increasing sophistication of such attacks makes it harder to use simpler algorithms (e.g. pattern recognition) that focus on a set of predefined information. The symptoms of a DDoS attack include unusually slow network performance and/or the inability to access a particular set of web sites. When this happens, the ENI System will be able to detect and learn from the occurrence by using AI methods. If the new traffic pattern is identified as an attack based on past history, the ENI System will be able to trigger appropriate responses from the related management components. In addition, AI enables different types of attacks to be correlated. For example, different attacks could use different protocols, but all be directed at the same target. This type of conclusion is extremely hard to make without using inferencing.

By using those techniques, the ENI System will be able to identify these and other types of attacks with a shorter timeframe and better precision when compared to today's systems.

Figure 5-60 provides a pictorial representation of the Use Case just described, where the first one shows the isolation of a network device once suspicious traffic behaviour is detected by the ENI System.

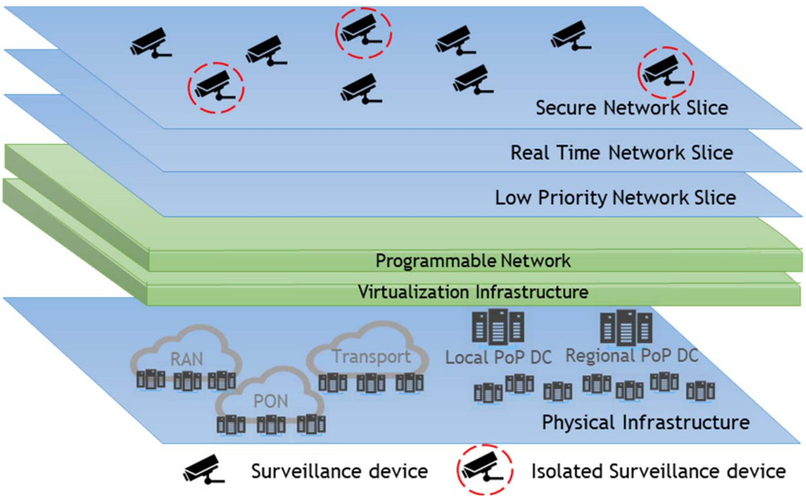


**Figure 5-60: Device isolation within a Network Slice**

#### 5.6.1.2.2 Actors and Roles

The presence of the following actors/entities as well as their associated roles are envisaged in the current Use Case:

- Customers/clients: the operators themselves.
- Network Infrastructure: infrastructure that includes resources and devices that are meant to provide applications/services related to social interest.
- IoT devices: normal devices and infected devices, e.g. those that are victims of a Botnet malware attack.
- Network Administrator: entity/person responsible for the policy design that encompasses the isolation of devices that were victims of DDoS attacks.
- OSS/BSS: components that provide monitoring data slicing management functionalities for ENI to detect and mitigate attacks. In addition, they also provide interfaces to network administrators and customers.
- ENI System: System solution that makes use of AI methods to identify and trigger responses to attacks.

#### 5.6.1.2.3 Initial context configuration

The network is operating correctly.

##### 5.6.1.2.4 Triggering conditions

A first trigger is when the ENI System detects changes in services provided (e.g. a web site).

A second trigger is when the ENI System identifies abnormal traffic patterns from a set of devices.

##### 5.6.1.2.5 Operational flow of actions

The following sequence of actions may be identified:

1) The ENI System monitors services (e.g. a web site) and devices that support the service itself (e.g. the network and a supporting server farm) as well as provide access to the service, looking for anomalous behaviour.

2) The ENI System detects that an abnormal event has occurred (e.g. web site is no longer accessible, or the traffic patterns of a device do not correspond to its expected behaviour).

3) The ENI System analyses the changes indicated by the abnormal event, and determines whether this is an attack or not. If it is an attack, then it notifies the Network Administrator, requesting the necessary operations to mitigate the attack.

4) These OSS/BSS entities enforce related policies and isolate the infected IoT devices from the rest of the network.

5) These OSS/BSS entities also notify the Network Administrator and related customers, if applicable, about the occurrence of the attack and restores normal service to the customer.

##### 5.6.1.2.6 Post-conditions

The infected devices have been identified and isolated from the network in a swift and efficient manner, and all other devices were able to maintain their normal operations.

If the customer's service was interrupted by the attack, then the customer's service is restored.

### 5.6.2 Use Case #5-2: Limiting profit in cyber-attacks

#### 5.6.2.1 Use Case context

There is a great activity over the globe from hackers searching how to obtain direct profits with the digital resources from victims that become their target. Among the several methods that a hacker makes profits with an attack, this use case puts its focus on ransomware and cryptocurrency mining techniques, which are mostly being used by cyber-criminals and give them direct benefits from the victims.

Both attacks search for the same objective but with different effects. With ransomware the hacker uses extortion of the victim under the threat of permanent damage in the victim's systems by encrypting their data. This type of attack due to the extortion is discover by the victim in a short time (regardless of the damage). On the contrary, with a cryptomining or cryptojacking technique, the attacker tries to remain unnoticed and it is difficult to detect, only by illegal resource consumption (CPU, power & memory). Profit is directly related to the time use before being discovered and the number of infections achieved. Attack using cryptomining does not damage or destroy victim data in a first instance but use victim resources and at the end, there is a malware in the system that can be used in the future to cause a greater damage.

This use case provides a solution based in ENI System to limit the damage and therefore decrease the profit of cybercriminals attacks through a network operator.

### 5.6.2.2 Description of the use case

#### 5.6.2.2.1 Motivation

Adjusting to the definition of the ENI System, this use case suggests leveraging Network Functions Virtualisation (NFV) in process of adoption by service provider to offer detection and prevention functionalities as services rather than as products. Moving towards a Security-as-a-Service paradigm allows providing different types of security functionality as detection and mitigation of ransomware and cryptomining attacks This approach allows putting ISPs, Telecommunications operators or Data Centres in a position to offer security services to customers at lowered cost and with reduced CAPEX.

A data centre infrastructure is extended by a certain number of distributed computing clusters to accommodate VNFs at various locations in the network (PoP, clients' premises, etc.). The VNFs perform a wide variety of operations, including security-related ones. Depending on the type of cyber-threats they are meant to detect or mitigate, VNFs may implement e.g. virtual firewalls and Intrusion Detection Systems (IDS).

This network architecture is completed with an ENI entity with specific machine learning algorithms trained to detect ransomware and cryptojacking attacks and an intent based policy language that automatically proposes new security policies to the OSS. The latter is in charge to enforce the policies through the NFVO into the VNFs, following policy-driven control loop.

Figure 5-61 shows the proposal architecture for this use case.

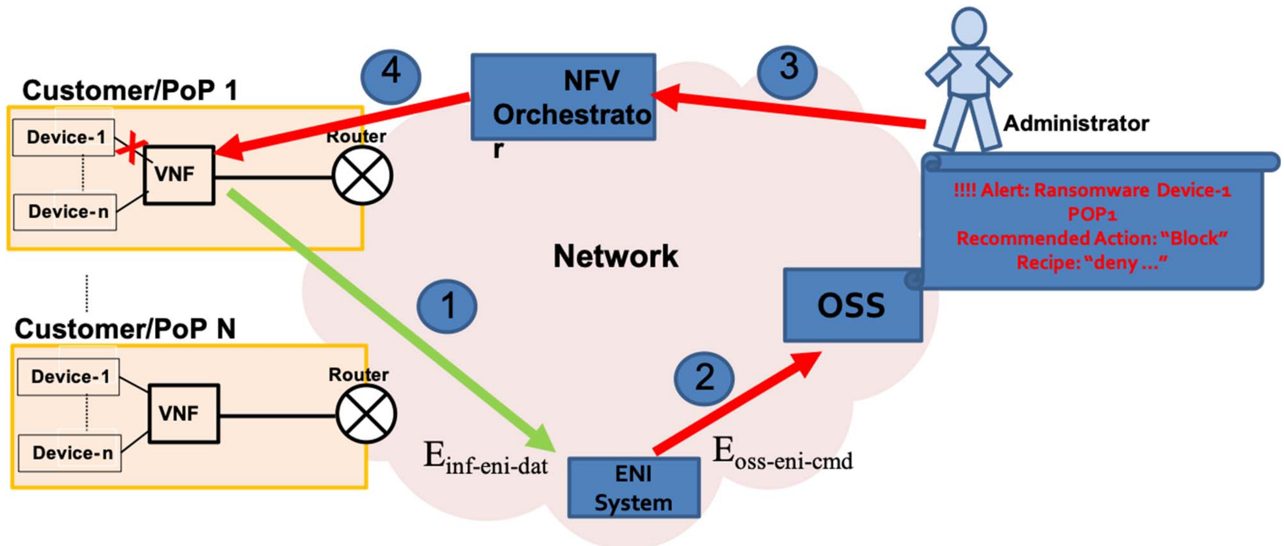


**Figure 5-61: Use Case to limit profit in cyber-attacks using NFV and ENI System**

#### 5.6.2.2.2 Actors and Roles

- Customer devices.
- ENI System: System solution that makes use of AI methods to identify and trigger responses to attacks.
- Network Infrastructure with NFV capacities (NFVI).
- NFV MANO including an Orchestrator (NFVO), that manage interconnectivity for the services.
- Network Administrator taking decisions to apply the proposed policy by ENI System.
- OSS providing interfaces to network administrators and customers.

#### 5.6.2.2.3 Initial context configuration

The network is operating correctly.

#### 5.6.2.2.4 Triggering conditions

The triggering is activated when the machine learning algorithm detects an anomaly based on the collected data from the network traffic (e.g. network flows) and is classified as a ransomware or a cryptojacking attack.

#### 5.6.2.2.5 Operational flow of actions

- Network traffic is monitored by the ENI System monitoring service (e.g. data collected from a virtual DPI or NetFlow probe) (Step 1 in Figure 5-61).
- The machine learning algorithms, in the ENI System, analyses traffic data collected, by means of aggregation and normalization, to produce relevant insights about the attacks.
- The ENI System identifies an attack in the early stage of propagation or mining, and notifies the Network Administrator and propose a recipe (a security policy) to block the infected systems to avoid its propagation or its operation in the case of cryptojacking (Step 1 in Figure 5-61).
- The OSS accepts and applies (Steps 3 and 4 in Figure 5-61) the policy based on ENI recommendation:
  - Ransomware: Isolating affected devices and related protocols from the rest of the network.
  - Cryptojacking:
    a) blocking the cryptomining functionality (block DNS queries, connectivity to mining pools or proxies); or
    b) isolating it from the rest of the network if it is considered a high risk for the network.

#### 5.6.2.2.6 Post-conditions

The infected devices have been identified and isolated, stopping the spread of the attack.

The customer can restore the situation in the infected devices (antimalware cleaning, reinstalling, etc.).

### 5.6.2.3 Mapping to ENI reference architecture

#### 5.6.2.3.1 Functional blocks

The global system (network infrastructure including NFVI, NFV Orchestrator) should be considered a Class 1 assisted system (An Assisted System that has no AI-based capabilities) from the point of view of ENI architecture. This is the current situation of the ETSI MANO functionality, NFVO has not embedded AI engine, but an orchestration capacity based on External Designated Entity of the assisted system (i.e. the operator, OSS, BSS).

In relation with Operation mode, the behavioural is "recommendation mode" [MOP.1]. This mode is recommended in order to allow the operator to acknowledge the mitigation actions, and avoid automatic responses especially if legal aspect is considered of alter client traffic [MOP.5]. Security problem detected by ENI System, is reported to the OSS Dashboard including a recommendation action to solve the problem [MOP.7].

Table 5-7 shows the mapping between ENI System module and ENI architecture functional block.

**Table 5-7: Mapping of ENI RPs to Use Case functionalities description**

| ENI Functional Blocks | Use case description |
|---|---|
| Knowledge Representation and Management | ENI will have a knowledge representation of the network being monitored (e.g. NFV network service descriptor, ISP network topology, address pools allocation clients, network gateway, etc.). |
| ENI Ingestion and Normalization | Different VNFs distributed over the network collect and aggregate traffic information and sent it to the ENI using standard streaming protocols to this functional block, who support several data types (network flows, event logs or alerts from the network) and normalize them in a common format, interpreted easily by ML algorithms (such an array of dataset). |
| Context-Aware Management | Directions of the network flows (client to internet or vice versa) and normal traffic profiles (bandwidth, common protocols) are store in this block. |

| ENI Functional Blocks | Use case description |
|---|---|
| Situational Awareness | N/A. |
| Policy Management | Previous blocks jobs allow to select the best recipe expressed in an Intend based language: such as "block traffic from client IP X" or "deny access" to the domain "miningpool.example.com". It will be reported to the OSS (recommendation mode). |
| Denormalization and Output Generation | Policies are translated to a format understandable by the technology provided by the VNFs or security device (e.g. firewall rule command to block an IP source address to any destination, or configuration change in DNS server database to point to answer NXDOMAIN for "miningpool.example.com"). |
| Cognition Framework | Uses specific machine learning algorithms tailored to detect specific ransomware and cryptomining. |
| Lifecycle Management | N/A. |
| Ancillary | N/A. |

#### 5.6.2.3.2 Interfaces

Table 5-8 describes the Reference point used in this use case where there are only interactions with external system shown in Figure 5-61.

**Table 5-8: Mapping of ENI RPs to Use Case interface description**

| ENI External Reference Points | Description |
|---|---|
| $E_{oss\text{-}eni\text{-}cmd}$ | OSS receive security policies recommendations to address security threats from ENI System. Recommendation mode uses a feasible language compatible with the OSS, such as XML, YAML, etc. Flow (2) in Figure 5-61. |
| $E_{inf\text{-}eni\text{-}dat}$ | ENI System collect data for security monitoring, including networks flows activity, or VNFs system logs. Flow (1) in Figure 5-61. |

#### 5.6.2.3.3 Flow of information

The flow of information is given in Figure 5-62.

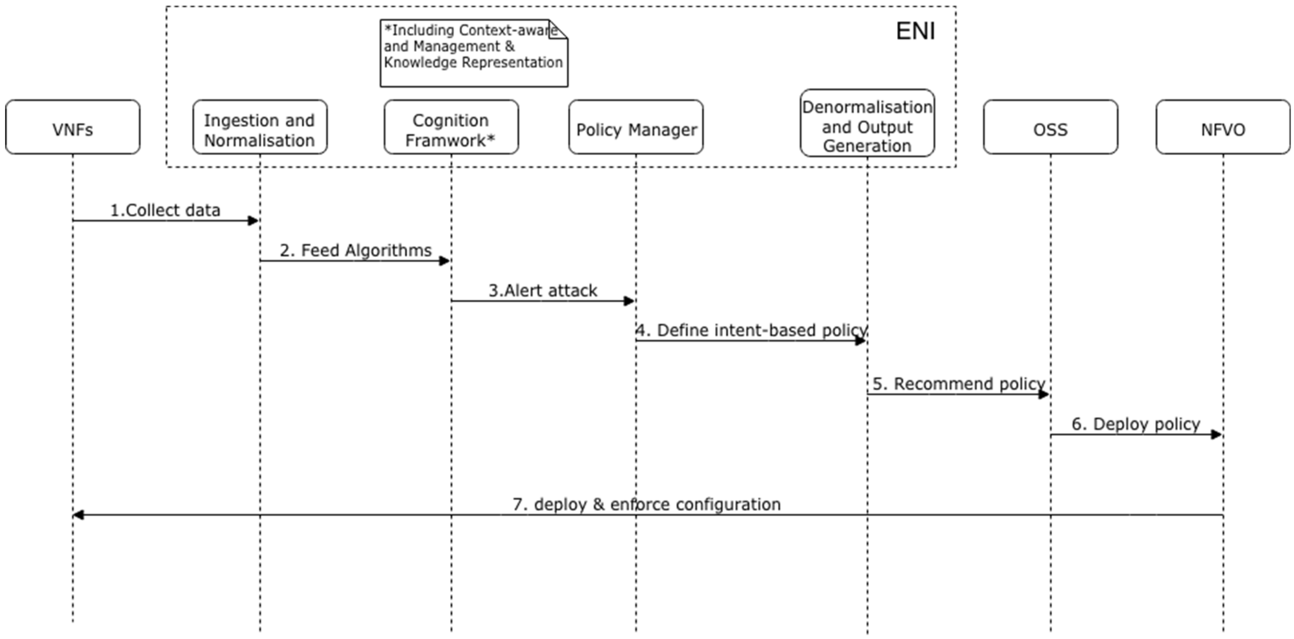


**Figure 5-62: Flow of information among the ENI System's functional blocks and the Assisted System**

Step 1: The monitoring VNFs sends traffic information to the ENI System through the interface $E_{inf\text{-}eni\text{-}dat}$.

Step 2: The Ingestion and Normalization Functional Block translate data from multiple input formats into a normalized form and feed to the machine learning models.

Step 3: An alert is produced by the models in the Cognition Framework functional block with the aim of additional functional blocks.

Step 4: The policy manager defines a recipe (intent based security policy) to mitigate the ransomware or cryptojacking attacks.

Step 5: The mitigation policy recommendation is sent in valid format that the OSS can understand through the interface $E_{oss\text{-}eni\text{-}cmd}$.

Step 6: OSS through and operator decide to apply the recommended policy.

Step 7: NFV Orchestrator enforce the mitigation policy using ETSI NFV reference points. This enforcement can include deploy new VNFs or configuration changes over the existing ones.

# 5.7 AI Agents Use Cases Consumer Use Cases

## 5.7.1 Use Case #6-1: AI Agents to Enable Smart Life

### 5.7.1.1 Description

The growth in the use of AI agents enriches human daily life and supports the intelligent solutions in the industry. For embodied AI agents, there are different types in the market with different capabilities and levels of intelligence, e.g. an intelligent humanoid robot can provide many different types of support across a variety of circumstances, a robot-dog is more lightweight but can provide less service than a humanoid robot, a drone is of low intelligence considering the battery constraint, limiting it to tasks like delivering foods for human beings. For virtual AI entities, the chatbot has entered the area of customer service. According to the latest report from Goldman Sachs [i.9], the global market for humanoid robots could reach USD 38 billion by 2035 and the humanoid robot shipment is expected to hit 1 million units by 2035. Precedence Statistics also show that the global intelligent virtual assistant market size was United States Dollar (USD) 16,17 billion in 2023, accounted for USD 20,42 billion in 2024, and is expected to reach around USD 166,97 billion by 2033, expanding at a Compound Annual Growth Rate (CAGR) of 26,3 % from 2024 to 2033 [i.10].

In future production scenarios, such as industrial grounds, smart cities and hospitals, AI Agents will be used to complement and even replace human labour.

EXAMPLE: These AI agents could be given the order of replacing all instances of part A with part B in the products being manufactured, checking the production line for possible problems or inefficiencies, or reconfiguring the operation of other machines.

As with humans, autonomous AI agents are not only the recipients of direct commands, but they also have the initiative to inform about the status of other components in the factory or even spontaneously raise alarms. The mobile network can provide connectivity to these AI agents as well as provide global perception and assisted AI services to AI Agents, rather than limiting them to local perception and intelligence. The reason is that, on the one hand, the accuracy of local sensors and models sometimes are not high enough to ensure safe and efficient operation. On the other hand, the use of local small-scale models for individual AI agents often results in sub-optimal decision-making. In summary, the mobile network can empower AI agents with network services and resources including sensing, AI, communication, and computing.

Assume that a user owns several types of AI agents, including a smart car, drone, robot-servant and robot-dog, which are produced by different manufacturers and of different capabilities. Further assume that all of these agents have registered with the AI-Core and have access to the mobile network, and each of them is allocated an identity to uniquely identify them. Currently, agents retrieve translated intent from other sources, as this is a complex multi-stage pipeline for all but the simplest of intents. In the future, it is possible to define a set of agents to perform the multi-stage intent translation directly, which will simplify the life of the user.

Considering an example where a user wants to go camping on the weekend, the AI agents are required to work together to make a good camping plan with the assistance of the network. The service flows of this example are as follows:

1) The robot-servant sends the intent of "make a camping plan" to the 6G network on behalf of the user. How the robot-servant gets the user's authorization is out of scope, e.g. it can take verbal commands from the user.

2) The AI agents in the mobile network parse the intent and break it down into sub-tasks, which are then assigned to different AI agents based on their capabilities, e.g. the network AI agent asks local life assistant (a digital AI agent provided by 3rd party) to recommend campsites, instructs the smart car to pick-up family members by designing the optimal route, asks the robot-servant to book food in advance according to the user's taste.

3) The network AI agent builds connections for the involved AI agents since they need to exchange information. For example, the local life assistant needs to send the campsite address to the smart car for designing the routes. Robot-servant selects the restaurant based on the user's taste and sends the restaurant information to the robot-dog while instructing it to pick up the order.

4) The AI agents execute the allocated sub-tasks. When they cannot perform the sub-task well, they request for the network services. For example, due to the limitation of local perception, the smart car requests the sensing service of the network when it determines the optimal route from the user's home to the campsite. The robot-servant requests the AI service of the mobile network to help perform inference of motion control.

5) The network monitors the AI agents that perform sub-tasks, when some exception occurs, it finds an alternative AI agent to complete the sub-task.

Figure 5-63 depicts the described scenario.

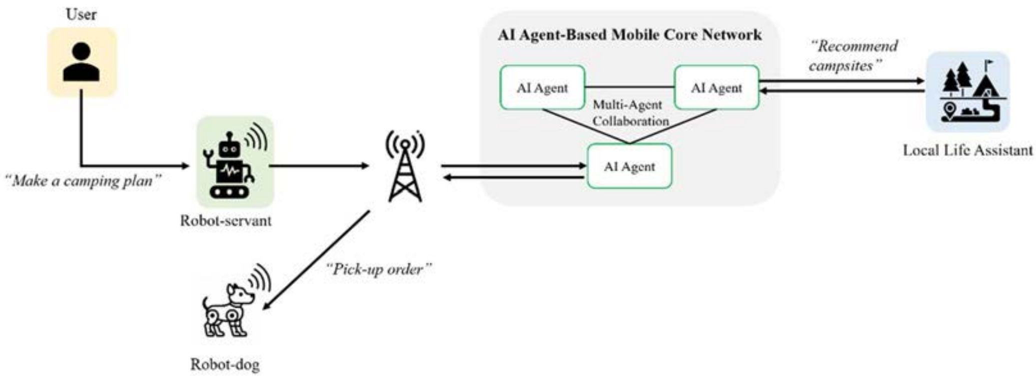


**Figure 5-63: Smart life use case key entities and high-level service flows**

## 5.7.2 Use Case #6-2 on Network-Assisted Collaborative Robots

### 5.7.2.1 Description

Multiple robots can collaborate to accomplish complex tasks that are beyond the capability of any single robot. Examples include carrying heavy objects, monitoring wide areas, or conducting search and rescue operations in disaster zones. To enhance adaptability and generalization across diverse robotic tasks and various robot designs, large models play a crucial role. Equipped with such, robots can perceive their dynamic environments, make informed decisions through advanced reasoning, and execute actions to interact effectively with the physical world. Agentic AI technology offers great potential to implement such intelligent systems, thereby, revolutionizing the next generation of robotic systems. However, due to the inherent limitations in onboard resources and computational power, individual robot agents often rely on the assistance of AI agents equipped with more advanced models to assist them in performing complex tasks efficiently.

For instance, multiple robots can cooperatively perceive their surrounding environments by utilizing onboard sensors that capture diverse, multi-modal data such as speech, images, videos, haptic feedback, and RF signals. These sensing processes across different robots and modalities sometimes are performed asynchronously to accommodate varying operational conditions. The collected sensing data are transmitted to a central network, where large AI models analyse and infer the environmental states.

AI models with relatively smaller sizes can be deployed on robots to perform data pre-processing such that the latency of communication and computation can be balanced. To optimize communication latency and computational efficiency, smaller-scale AI models can be deployed directly on the robots for initial data pre-processing. This distributed approach balances the workload between the robot- and the network agents. Additionally, the network is expected to determine which robots and sensing modalities are essential for the cooperative task, minimizing redundant operations and maximizing overall system efficiency. The service flows of the described use case above are listed as follows:

1) Multiple robots register with a management and orchestration entity in the AI-Core to participate in environmental sensing tasks, such as gesture recognition.

2) The mobile network collects capability information from each robot, including sensor capabilities (i.e. modality, resolution, accuracy), communication capabilities, computational capabilities, and energy consumption profiles.

NOTE 1: This step is a continuation of the registration process. The capability information described is the payload of the registration request from Step 1. This structured data is used to create or update the robot's entity within a suitable entity, such as a Knowledge Graph (KG). The KG serves as the network's "digital twin," storing not just the identity of each robot but a rich profile of its specific attributes, which can be queried. This allows the orchestrator to perform sophisticated queries later, such as, "Find all available robots within Area A with camera resolution > 4K and battery > 50 %."

3) One or more robots send service requests specifying their service request (i.e. the intent to collect information about their surroundings, potential requirements and/or preferences on information accuracy, data delivery latency, and energy consumption) to an Intent Management Entity within the AI-Core.

NOTE 2: Here, a robot switches roles from a potential service provider to a service consumer. It submits its high-level intent not to the general "network" but to a specific Intent Management Entity (IME) within the AI-Core. This interaction is handled by an Intent Translation Pipeline. This pipeline uses NLP and LLM-based techniques to parse the intent and its constraints (accuracy, latency) into a formal, machine-readable specification that the orchestration system can act upon.

4) Based on these requests, an AI Orchestrator (or Supervisor) agent within the AI-Core takes the validated intent from Step 3 and uses a formal planning methodology, such as Hierarchical Task Networks, to decompose the abstract goal ("collect information") into a concrete workflow.

5) The Orchestrator queries the Knowledge Graph (or other entity that is used) to find the optimal set of robots (from the pool of robots registered in Step 1) whose capabilities match the requirements of the decomposed plan. This is where it would "select appropriate data modalities" by choosing robots with the right sensors.

6) The Orchestrator schedules the tasks and delegates specific instructions to the chosen robots.

7) Simultaneously, the Orchestrator delegates a sub-task to the relevant Network Management Domain (e.g. the RAN or Core domain manager) to "configure relevant network equipment." This typically involves creating a dedicated, QoS-guaranteed network slice to ensure the sensing data can be transmitted with the required latency and reliability.

8) Robots, now acting as worker agents, execute the instructions delegated by the Orchestrator in Step 6. The robots transmit their data to a dedicated Data Analytics Service as specified by the Orchestrator's instructions.

9) The raw data sent by the sensing robots is received by a dedicated Data Fusion and Analytics Agent within the AI-Core. This specialized agent is responsible for the post-processing tasks: aggregating data from multiple robotic sources, filtering out noise, and performing higher-level inference (e.g. fusing multiple camera angles to perform gesture recognition). Once this value-added processing is complete, this agent delivers the final, refined information back to the original robot that made the service request in Step 3, thus closing the loop. This entire workflow is monitored by the Orchestrator to ensure the original intent is fulfilled.

NOTE 3: Steps 8 and 9 operate as a pair. Step 8 is the addressable endpoint that the robots transmit their data to. Step 9 is the logical actor within the AI-Core that owns and operates that service. It is the intelligent component that actually receives the data from the service endpoint, fuses data from multiple robots while filtering noise, and generates the final, value-added result.

### 5.7.3 Use Case #6-3 on AI Phone

#### 5.7.3.1 Description

Mobile terminals are envisioned to be transformed into AI phones in the near future, supporting a wide range of sophisticated AI applications. Beyond traditional connectivity services, next-generation mobile communication systems will provide new capabilities to AI phones. For example, the phone becomes a dynamic platform that receives AI models provisioned by the network. The brain of an AI agent can be deployed across various platforms, including cloud servers, edge servers, or end devices. In this model, the network operator can push specialized, task-specific AI models or lightweight agents directly to the "AI Phone" as needed. Significantly, the phone's abilities are no longer static and tied to the apps that the user(s) have installed. Rather, its intelligence can be augmented on-the-fly by the network. In addition, instead of a single, massive, all-purpose AI model residing permanently on the device, smaller, more efficient models tailored to the immediate task can be loaded and unloaded, optimizing resource usage.

This capability enables a compelling future vision. Instead of the user having to identify a specific task, select an appropriate application, and launch the application and its (possibly unfamiliar) unique interface, the user can simply express one or more intents in natural language to the phone via text or voice. The phone's agent(s) then interpret the intent and orchestrates all the necessary sub-tasks in the background. For example, instead of having to learn the particulars of each airline's interface, the phone can check the user's calendar, find flights, book a hotel, arrange ground transport, and anything else required, all by invoking the required services or specialized sub-agents. Hence, the user is freed from the cognitive load of managing a constellation of individual apps. The complexity is abstracted away, and the interaction feels like a continuous conversation with a single, capable assistant. The "apps" still exist as underlying services or functions, but the user no longer interacts with them directly.

While a simple chatbot does not require immense intelligence, the vision described in the present document does. However, it is important to realize that this vision does not have to be implemented all at once. To function as a truly personalized AI agent that can seamlessly fulfil complex, multi-modal requests, the system possesses capabilities that are indeed at the forefront of AI research: advanced reasoning and planning, social intelligence (i.e. the ability of the agent to infer the user's unspoken needs, desires, and emotional state from nuanced cues), and multi-modal fusion (a highly complex task that requires sophisticated AI models).

For example, a user might interact with the next generation mobile core network through their AI phone to place an order for a customized product. The core network would interpret multi-modal inputs from the user, such as images, voice, and text, to understand their intent, generate a unique design, and coordinate with the product supplier to fulfil the order. This requires the core network to process diverse data types to understand the user's intent through agentic AI technology and autonomously deliver customized on-demand services to satisfy the user's request. To enable this, seamless and intelligent communication between the AI phone and the core network's AI agent is essential.

The service flows describing the interaction between the AI phone and the core network AI agents are as follows:

1) The user launches the AI phone client and submits a product customization request using multi-modal inputs, including images and audio.

NOTE: Here, the AI phone is the initiating agent. In order for this to work, the Intent Management Entity needs to expose a set of standardized, secure interfaces over the Integration Fabric that is accessible to authenticated user agents on mobile devices.

2) The phone's AI agent performs initial validation, data compression, and possibly restructures the raw inputs before sending them to the Intent Management Entity within the AI-Core, via the system's secure Integration Fabric.

3) The Orchestrator agent within the AI-Core receives the intent and validates it against the network's KG to check for feasibility, resource availability, and potential policy conflicts. It uses a formal planning technique like Hierarchical Task Networks (HTNs) to decompose the complex intent ("create and deliver a custom product") into a workflow of sub-tasks. The Orchestrator then delegates these sub-tasks to the appropriate specialized agents (e.g. a Multi-Modal Fusion Agent, a Product Design Agent, a Supplier Logistics Agent (a third-party tool), and a Network Slice Manager Agent (for handling the QoS requirements mentioned in the next step).

The AI Orchestrator, as part of its overall plan, issues a derived intent to the relevant Network Management Domain (e.g. the 5G Core domain manager). This intent would be something like, "Establish a network slice for user X with latency < 50 ms and bandwidth > 100 Mbps for the duration of this session". The domain manager is then responsible for configuring the low-level network resources to meet this QoS requirement.

# 5.8 AI Agents Use Cases Business Use Cases

## 5.8.1 Use Case #7-1 on AI Agent-based Customized Network for Smart City Traffic Monitoring

### 5.8.1.1 Description

The new use cases and services supported by the next-generation mobile network are outlined in Recommendation ITU-R M.2160-0 [i.12] and NGMN Alliance [i.1]. These emerging services require the mobile networks to provide various capabilities, including AI-driven and sensing-related capabilities, as well as multi-dimensional resources such as computing power, communication, and data storage. Given these new services and requirements, next-generation mobile networks are required to be able to instantiate isolated, logical networks on-demand, each with guaranteed Quality of Service (QoS) parameters (e.g. bandwidth, latency) tailored to a specific application.

Traditional network management processes that are based on expert knowledge and historical experience are no longer sufficient for providing adequately customized networks for new services that require the flexible assembly of multiple functionalities and multi-dimensional resources. AI agents with their strong capabilities in intent understanding, tool usage, planning, decision-making, task execution and self-evolution, offer a promising approach to customizing networks for new services.

For instance, in a smart city scenario, the mobile network is required to provide a customized network, i.e. a logic network that assembles various network capabilities and resources, to the smart city operator for monitoring passenger and vehicle flow. During peak tourism periods, smart city operators need real-time information on passenger and vehicle flow in scenic spots to implement intelligent traffic dispatch, alleviate congestion, and improve citizens' travel convenience. In this context, the mobile network needs to offer a customized network to monitor these flows during peak hours. Outside the peak commute hours, the monitoring service is automatically terminated by the operators, and the dedicated network for monitoring service is subsequently deleted. Thus, the creation and reclamation of this customized network for services occur on demand. The service flows of this use case are illustrated in Figure 5-64 and are given as follows:

1) The smart city operator submits a high-level, multi-modal intent to a specific service endpoint provided by the network's AI-Core. This request is ingested by an Intent Translation Pipeline, which uses LLMs to parse the natural language and temporal constraints (e.g. "scenic spot A," "from 8:00 to 17:00") into a structured, machine-readable format.

2) The AI Orchestrator receives the structured intent. It then:

    a) Plans the Workflow using Hierarchical Task Networks by decomposing the intent into a sequence of sub-tasks (e.g. "provision slice", "configure analytics", "deploy monitoring agents", "deliver results").

    b) It queries the network's Knowledge Graph to identify available resources, such as cameras in "scenic spot A" and available edge computing nodes for data processing.

    c) It then issues specific sub-intents to the relevant domain managers. For example, it instructs the 5G Core domain manager to instantiate a dedicated network slice with the required QoS, and it instructs an edge orchestration manager to deploy the necessary analytics functions (e.g. object detection) on selected compute nodes. This process of connecting functions is known as service function chaining.

3) Closed-loop assurance is realized by the AI Orchestrator (in the AI-Core) continuously monitoring telemetry data from the active network slice and the analytics functions. If it detects a deviation from the intent (e.g. latency increases, video quality drops), it automatically initiates a self-optimization routine, such as allocating more bandwidth or migrating a processing task to a different edge node, to maintain the agreed-upon service level.

4) The specialized Data Analytics agent, having processed the raw data from the sensors, delivers the structured output (e.g. a JSON object with vehicle and passenger counts) to the smart city operator's designated endpoint at the specified one-minute interval.

5) At 17:00, or upon receiving a termination command, the AI Orchestrator executes the final stage of its plan. It issues delete commands to the relevant domain managers to tear down the network slice, terminate the analytics functions, and release all reserved compute and network resources, ensuring efficient resource utilization.

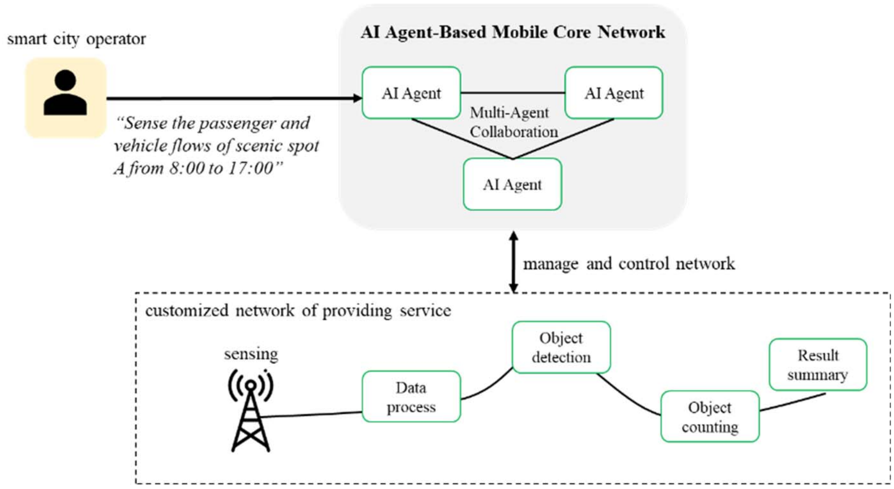


**Figure 5-64: Customized network generated by AI agent-based core network**

## 5.8.2 Use Case #7-2 on AI Agents-Based Customized Network for Smart Construction Sites

### 5.8.2.1 Description

Traditional network management will no longer be sufficient for highly flexible and demanding use cases such as smart construction sites, depicted in Figure 5-65.

At a construction site, multiple shareholder companies share the same space and need to cooperate, requiring high performance for data connectivity. This communication will be handled with nomadic networks, which are mobile and can be deployed at any location. Current construction sites only connect workers with information about the building project or interconnect worker timekeeping systems. In future construction sites, more vehicles and tools, especially connected and automated vehicles like AGVs, trucks, and machinery such as cranes, will be connected to the mobile network:

1) The manager submits a high-level intent (e.g. "Provide basic data connectivity for 50 workers at Site B, accessing project plans on server X"). This request is ingested by the Intent Translation Pipeline, which parses it into a structured format for the AI Orchestrator. The Orchestrator then plans and deploys a simple network slice with standard QoS parameters.

2) The construction site manager orders a network configuration for transmitting LiDAR data and communication for AGV routing, shortly before the AGV's arrival at the construction site. This is a Modification Intent, which is more complex than the initial request. The manager is updating the service requirements. The Orchestrator receives this new intent (e.g. "Add a high-bandwidth, low-latency service for LiDAR data and a high-reliability, low-latency service for AGV control to the existing network at Site B"). This demonstrates the system's crucial ability to manage the entire service lifecycle, not just initial deployment.

3) After receiving the request, the AI Orchestrator takes the new, modified intent and updates its plan for providing communication services to smart construction site operators. This consists of:

    a) Re-planning using Hierarchical Task Networks to determine the necessary changes to the existing service, such as creating new, dedicated network slices for the LiDAR and AGV traffic or modifying the QoS of the existing slice.

    b) It queries the KG for available network resources and deploys the necessary application functions (e.g. data processing for LiDAR, routing algorithms for AGVs) onto edge compute nodes.

    c) It issues updated configuration commands to the relevant Network Domain Managers to implement the changes.

4) During the runtime of the generated generative network, the AI Orchestrator performs a closed-loop service assurance function by continuously monitors the performance of both the basic connectivity and the new high-performance slices. If an AGV's connection latency spikes, the orchestrator's assurance loop detects the SLA deviation and can autonomously trigger an optimization, such as rerouting traffic or allocating more radio resources to maintain the intent. This is a "dynamically generated network".

5) A Monitoring Agent provides periodic status reports (e.g. network uptime, AGV connectivity status, data throughput) to the construction site manager's dashboard, confirming that the service is operating as intended.

6) After the termination of the service, the AI Orchestrator executes the teardown plan, instructing all domain managers to release the network slices and all associated compute and storage resources.


**Figure 5-65: Example representative figure for smart construction sites**

## 5.8.3 Use Case #7-3 on AI Agent Ensuring Game Acceleration Experience

### 5.8.3.1 Description

Acceleration services in mobile networks are designed to enhance the speed, responsiveness, and overall quality of user experiences by optimizing how data are transmitted and managed across the network. These services are critical as mobile data traffic continues to grow rapidly, driven by applications like video streaming and gaming.

Currently, acceleration services offer various packages tailored to specific applications or users. However, the future landscape will involve users interacting with multiple non-network AI agents - such as game agents provided by third parties - that manage phone operations and screen projection. This necessitates a standardized, secure way for agents to communicate, discover each other's capabilities, and exchange intent information, which is the role of an Integration Fabric and a Service Registry.

The existing acceleration service framework is inadequate to support this intelligence-driven ecosystem or to provide truly personalized acceleration services. Currently, the mobile network operators deploy proprietary nodes to directly connect to game servers and uses dynamic routing technology to provide acceleration services for video games based on their latency or bandwidth requirement. However, the customization of the acceleration service according to the intent of users or applications is not supported. Moreover, the network adjusts the acceleration service in a reactive way with the changing user requirements, which in turn affects the user experience due to the delayed reaction time.

Network AI agents can adjust network resources from the core network to the base station in an end-to-end manner based on the received intent to provide a personalized service experience. This intent typically originates directly from the user, from an external AI agent (e.g. a game agent), or from a game company's server that considers the terminal conditions of all game participants. In addition to passively responding to these intents, the network AI agents can proactively predict fluctuations in user experience by analysing network status and historical data, enabling it to adjust policies in advance to maintain service quality.

Moreover, future charging models can leverage the capabilities of network AI agents. For example, subscription fees for AI agents that understand individual users' intents or other agents' intents could be higher. Pricing could also factor in the computing resources consumed by these agents. This approach enables MNOs to offer more perceptible, personalized acceleration experiences that users are willing to pay a premium for, thereby, enhancing revenue growth by delivering personalized and flexible service experiences to mobile subscribers.

The service flows of network AI agent providing game acceleration service are as follows:

1) The consumer subscribes to the game acceleration solution empowered by AI agents. The user plans to play the subscribed game while commuting from City A to City B. A proactive system enables the user's "Personal AI Agent" to share this high-level context (e.g. "User is starting a commute and typically plays Game X") with the network's AI-Core. This allows the network to anticipate the need for the acceleration service.

2) While traveling by taxi to the rail station, the consumer opens the game app. The game agent sends a structured intent (e.g. "Requesting low-latency, high-reliability service for Game X") to the network's Intent Management Entity (IME).

3) The AI Orchestrator receives the intent. It queries the KG and network monitoring services to gather context (user location, speed, current network load).

4) Based on the intent and context, the Orchestrator generates and deploys the initial network policies, likely by provisioning a dedicated network slice with specific QoS parameters.

5) At the railway station, where many users require a guaranteed game experience, their AI agents all connect to the AI-Core. The AI Orchestrator receives multiple, potentially competing, intents. It is required to use a conflict resolution or resource optimization function to balance the needs of all users against the available network resources (e.g. base station capacity). This is a highly advanced capability that moves from single-user service fulfilment to multi-user resource management.

6) When traveling, the user sometimes passes through areas with poor signal coverage (e.g. tunnels). To minimize the degradation in the quality of service, the AI agent can proactively predict the service degradation using historical data and notify the consumer in advance, which possibly affects the user's decision to save or end the game. This is predictive, closed-loop assurance, and consists of three phases:

    a) An Analytics Agent within the AI-Core continuously analyses the user's trajectory and compares it against a KG containing historical network performance data for that geographic route.

    b) The agent's predictive model forecasts an imminent drop in signal quality as the user approaches a tunnel.

    c) This prediction triggers a proactive notification, sent via the Integration Fabric, to the user's game agent, which can then alert the user. This demonstrates the system's ability to prevent a poor user experience rather than just reacting to it.

## 5.8.4 Use Case #7-4 on AI Agent-Assisted Collaborative Energy Distribution in Power Enterprises

### 5.8.4.1 Description

In today's heterogeneous and distributed energy systems, which comprise solar, wind, hydro, and fossil fuel power, energy companies face the challenge of real-time coordination across multiple power stations. These challenges arise from heterogeneity in site capabilities (computing, communication), dynamic fluctuations in energy supply and demand, and the complexity of maintaining synchronization across collaborative operations. By embedding AI Agents within the mobile network architecture, energy enterprises can significantly enhance the intelligence, efficiency, and responsiveness of their operations.

NOTE 1: This describes a complex cyber-physical system where AI acts as the intelligent controller for a distributed power grid, using the mobile network as its nervous system.

AI Agents act as intelligent controller between power infrastructure and the mobile network:

- AI Agents receive user-intent inputs and manage historical data to support adaptive energy distribution strategies. By learning from past consumption patterns and real-time operational status, they can dynamically optimize energy allocation across multiple sites.
- AI Agents analyse multi-dimensional data, such as grid load and environmental conditions (solar, wind, hydro), to dynamically adjust the operation and output of power generation equipment.
- AI Agents analyse real-time energy consumption and distribution loss metrics to make intelligent decisions on whether to prioritize local consumption or dispatch energy across regions, enabling dynamic, large-scale optimization of power distribution.
- AI Agents continuously monitor real-time data streams to detect abnormal energy consumption patterns or equipment anomalies. Based on predefined operational logic or learned behaviours, they can autonomously trigger protective actions within the energy control system.

The integration of AI Agents with mobile networks brings key technical benefits:

- Ultra-Reliable Low-Latency Communication (URLLC) enabled by network slicing and optimized traffic routing, ensures timely delivery of control commands. Critical control commands are prioritized through network slicing, while edge computing enables rapid local decisions. Together, these capabilities form a closed-loop "sense-decide-act" system that enhances renewable energy utilization and maintains grid stability.
- AI Agents leverage lightweight models at the edge to process large volumes of sensor data close to the source, supporting fast and localized decision-making. Meanwhile, core network-based AI Agents leverage large models to perform global analysis and coordination. Through collaboration among multiple models across the edge and core network, this distributed architecture enhances system responsiveness and strengthens overall grid stability.

NOTE 2: This is a two-tiered intelligence model: "lightweight models at the edge" for rapid, local data processing and decision-making, and "core network-based AI Agents" with "large models to perform global analysis and coordination". The edge agents handle real-time control and data filtering, while a central, more powerful agent performs strategic, system-wide optimization.

- AI Agents configure the mobile network to dynamically establish communication links between different power generation sites as well as between the sites and central control centres. This ensures low-latency, reliable, and secure data exchange, enabling real-time coordination and intelligent energy dispatch across distributed facilities.

NOTE 3: The mobile network is transformed from a data transport pipe to a programmable resource that is actively configured by the AI system.

- AI Agents can collaborate with core network functions (e.g. AMF, SMF, PCF as an example for 5G core) to dynamically adjust communication policies, ensuring that energy control traffic is prioritized and resilient to disruptions.

NOTE 4: In conclusion, the system is initiated by a high-level goal, such as "maximize green energy use while maintaining power stability". This declarative intent is the starting point for a complex chain of analysis, planning, and execution.

The service flows of AI Agent-assisted collaborative energy distribution are as follows:

1) The Service Registration process begins with Agents representing each energy site connecting to the AI-Core's Service Registry and publishing their capabilities (power output, storage levels, etc.) to the network's KG. The KG now has a real-time digital twin of all available energy assets, which is essential for the planning phase.

2) The operator submits the high-level business goal to the Intent Management Entity (IME) within the AI-Core. This intent is declarative; it specifies what to achieve, not how to achieve it. An example is "maximize green energy use while maintaining power stability".

3) The central AI Orchestrator performs the following sub-tasks:

   a) The Orchestrator queries the KG and invokes specialized Analytics Agents to forecast energy generation and load based on historical and real-time data.

b) Using the forecast and the intent's constraints, the Orchestrator uses a planning engine (like Hierarchical Task Networks) to create an optimal energy distribution plan. It queries the KG to select the best combination of energy sites to fulfil the plan.

c) The Orchestrator issues derived intents to the 5G Core's Network Domain Manager to configure the necessary network slices with URLLC QoS for the control commands.

4) The Orchestrator dispatches specific, executable instructions (e.g. "discharge battery at 50 MW", "curtail solar output by 10 %") to the lightweight agents at the selected energy sites and edge nodes. The closed loop assurance process consists of the following steps:

a) The local agents at each site execute their assigned tasks.

b) The AI Orchestrator continuously ingests telemetry data from the sites and the network. It compares the real-world outcomes against the predicted model. If there's a deviation (e.g. an unexpected drop in solar output), it automatically re-runs its planning and optimization loop to generate and dispatch an updated strategy, ensuring the high-level intent is always being met.

# 5.9 AI Agents Use Cases Telecom Operator Use Cases

## 5.9.1 Use Case #8-1 on AI Agent-Based Autonomous Network Management

### 5.9.1.1 Description

As future mobile systems are expected to support an increasing array of new technologies and services, mobile network operators are actively seeking opportunities to reduce their operational expenses while meeting the evolving customization needs of these applications. The requirements of these applications are dynamic, necessitating solutions that can adapt rapidly to changing demands.

Predefined standardized processes are generally fixed and do not perform well in situations when the application requirements are changing frequently. This is particularly important for those applications that demand multiple new capabilities, such as AI, sensing, data collection and processing capability. For example, a vehicle related application requires the sensing and data process capability of the network to percept the road condition and process the perception data. Thus, the mobile system is required to orchestrate these new capabilities and multi-dimensional resources (e.g. computing, communication and data resources) to meet the requirements of various applications in an efficient way. That means that the mobile network is required to autonomously generate personalized solutions according to customized requirements of applications.

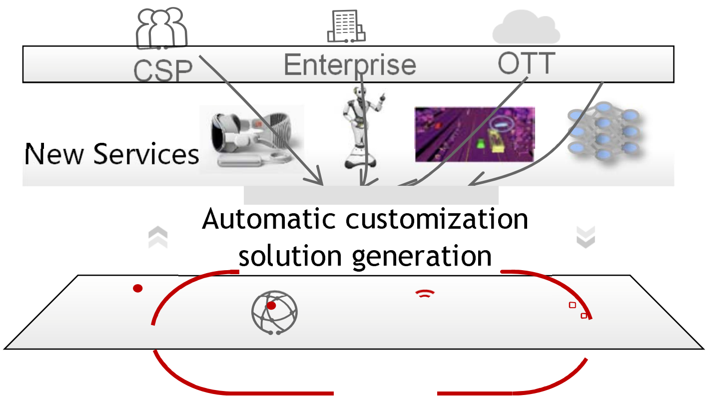


**Figure 5-66: Agent-Based Autonomous Network Management**

Take autonomous network fault analysis and recovery as an example. When a user finds that the network speed is very slow when surfing the internet, the user complains to the network through natural language. The system's first and most critical task is to understand this natural language input and translate it into a formal, machine-readable goal that can trigger the entire automated workflow. The service flows are as follows:

1) The user's natural language complaint is the high-level intent. It is submitted to the NMS, which serves as the user-facing portal to the AI-Core. The request is immediately ingested by an Intent Translation Pipeline. This pipeline uses NLP and LLMs to parse the unstructured complaint, identifying the core intent ("diagnose and resolve performance issue") and extracting key entities like the user's identity, location, and the nature of the problem ("slow speed").

2) Before a plan can be formulated, the system requires context. The "AI Agent for OPS" is best understood as the central AI Orchestrator. Its first action is to query the network's Knowledge Graph (KG) - the "knowledge database" - to gather all relevant information about the user and the affected network segment. This includes topology, active services, current traffic loads, and recent alarms in that area.

3) The AI Orchestrator manages the task decomposition:

    a) Using a formal planning technique like Hierarchical Task Networks, the Orchestrator decomposes the high-level goal into a logical workflow of sub-tasks.

    b) The Orchestrator delegates these sub-tasks to specialized worker agents via the Integration Fabric:

        i) Subtask 1 (Exception Detection): This is assigned to a Monitoring Agent, which invokes data collection tools to pull real-time performance metrics (latency, packet loss, throughput) from the relevant network elements.

        ii) Subtask 2 (Root Cause Analysis): The abnormal metrics are passed to a Diagnostics Agent. This specialized agent uses its analytical models to correlate the data and pinpoint the root cause (e.g. a congested backhaul link, a misconfigured router, or interference in the radio access network).

        iii) Subtask 3 (Fault Rectification): The root cause analysis is passed to a Remediation Agent. This agent generates a solution (e.g. "reroute traffic via backup path") and, critically, can first test this solution in a Simulation Environment or "digital twin" to verify its effect before applying it to the live network. This validation step is essential for ensuring that automated actions do not inadvertently cause new problems.

4) Reporting and Closing the Loop. The AI Orchestrator, having successfully managed the workflow, synthesizes the final report (e.g. "Root cause was a congested link. Traffic has been rerouted. Verify your connection speed.") and delivers it back to the user via the NMS interface, thus closing the loop on the original complaint.

### 5.9.2 Use Case #8-2 on AI Agent-Based Disaster Handling Network Management

#### 5.9.2.1 Description

The development of sustainable networks for disaster handling is critical for ensuring the resilience of communication infrastructure and the efficient delivery of essential services in the event of a disaster. The challenge in network disaster handling is not only to provide robustness and resilience to communication infrastructure in disaster-prone areas but also to provide sustainable solutions that can be rapidly deployed and require minimal maintenance, reducing the overall cost and environmental impact of network operations. As approaches to resiliency often imply, incorporating a certain amount of extra service capacity and allocating additional network resources, sustainability of these approaches is crucial to create incentives to adoption, and it is an important aspect of overall network sustainability. Figure 5-67 shows a representative system architecture for network disaster handling.

The sustainable networks for network disaster handling focus on solutions for the use of an integrated, resource shared edge cloud continuum network in the case of disasters or unintended events such as earthquakes, flooding, major power shutdown, or system failures. The aim is to ensure the efficient operation of communication systems for disaster response operations, coordination of rescue and relief operations, and always-on network services such as emergency, medical, and rescue services. This use case considers also the process of restoring the network and its services to their pre-disaster state or to a new sustainable state. It is critical to ensure that the network can seamlessly transition from the emergency state to the normal operating state with minimal downtime and without compromising the quality of service.

The common network solution is controlled by a centralized location based on predefined rules. However, in this case, the AI Agents need to be trained to work in distributed mode with partial knowledge of the network and without a centralized location.

NOTE: While an AI Orchestrator in the AI-Core might manage the network during normal operations, individual agents need to be capable of autonomous, peer-to-peer collaboration to restore services when that central authority is unreachable. This decentralized structure provides robustness, as the failure of some agents does not cripple the entire system.

The set of AI Agents have to cooperate to enable a communication network based on the current physical network conditions. This use case is split into two main stages: before a disaster event and during one. In the first stage, the AI Agents monitor different types of data, including network telemetry collected by AI Monitoring Agents, external environmental feeds ingested via APIs by special Environmental Agents, physical infrastructure sensors collected by special Physical Sensing Agents, and other contextual data, as part of its normal operations. The AI-Core uses a Knowledge Graph for all relevant data, acting as the "digital twin" of the operational environment. A specialized Analytics Agent continuously analyses the aggregated data in the KG and performs cross-domain correlation. It identifies patterns that indicate a potential disaster, such as: "The National Weather Service API is reporting a hurricane with a projected path directly over our primary fibre optic route in Florida."

By correlating external environmental data with internal network and infrastructure data, the system can move from being reactive to being predictive. It can anticipate the impact of a disaster before it happens and begin taking preparatory actions, such as rerouting traffic away from the projected storm path or spinning up resources in a safe location.

In the second stage, the AI Agents try to re-establish connection between themselves and reconfigure the network to bring back communication and network services. The corresponding service flows are:

1) During normal operation: The AI Orchestrator directs Monitoring Agents to collect telemetry. This data is fed into a specialized Analytics Agent that runs predictive models to identify potential threats (e.g. tracking a storm's path) and forecast their likely impact on specific network infrastructure. This analysis continuously updates the network's Knowledge Graph (KG), which can serve as a distributed source of truth for all agents.

2) During disaster operation:

   a) One or more AI Agents in each isolated network segment use their local sensors and cached knowledge of the network's last known good state to identify which connections have failed. This is a reactive, autonomous behaviour based on local perception.

   b) Each surviving AI Agent needs to collaborate with other surviving AI Agents in its local segment. Using robust Peer-to-Peer (P2P) communication protocols, they negotiate a strategy to re-establish connectivity. This could involve activating backup links (e.g. satellite) or dynamically creating a mesh network between surviving nodes. This requires AI Agents to have the authority to reconfigure local network elements.

   c) To restore service in a completely isolated segment, the collaborating local agents could decide to instantiate a lightweight, containerized version of essential core network functions (e.g. a local AMF or UPF) on an available edge compute node. This creates a self-contained, functional "network-in-a-box" until connectivity to the main core network can be restored.

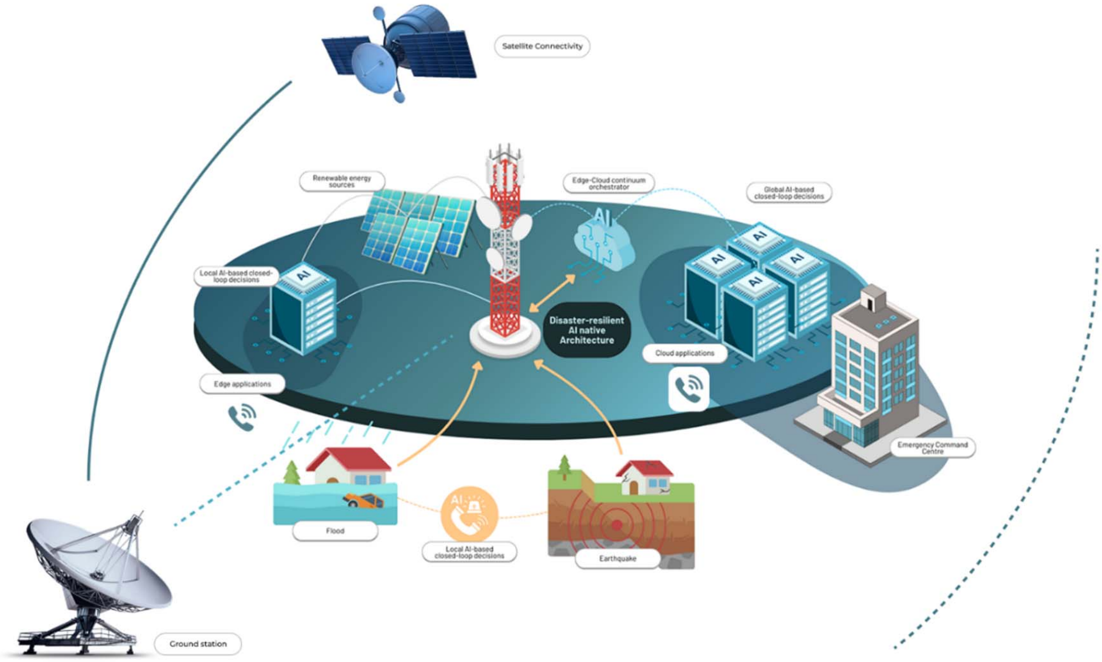


**Figure 5-67: Representative figure for network disaster handling**

## 5.9.3 Use Case #8-3 on AI Agent-Based Time-Sensitive Network Management

### 5.9.3.1 Description

Communication forms a bedrock for industrial automation in the realm of Industry 4.0. Communication networks for Industry 4.0 applications need to support deterministic, low latency and high reliability communication. Currently many industrial applications utilize wired Time Sensitive Networking (TSN), however, to improve flexibility and support mobility, wireless TSN extensions are needed. This use case focuses on time sensitive applications in a Non-Public Network (NPN) deployment that supports multiple Radio Access Technologies (RATs), including IEEE 802.11 [i.11] in addition to cellular network. Time-sensitive networking can be utilized in different parts of the NPN (i.e. in the Radio Access Network (RAN)), in the transport network, to inter-connect different open RAN (O-RAN) components, or to interconnect different core network components. When using inside RAN, the time sensitive features can be enabled in multiple RATs (multi-RAT). This use case considers the utilization of multi-RAT for enabling end-to-end time-sensitive networking for Industry 4.0 applications such as factory automation, control-to-control communication, process control, logistics (automated guided vehicles), operator assistance, collaborative robots, and remote control of robots.

This use cases covers the establishment of a unified method to manage and control multi-RAT within access network enabling seamless and end-to-end network optimization. A unified multi-RAT data plane with time-sensitive networking capabilities will be utilized to improve reliability by performing replication in different RATs and channel bands, to support seamless and deterministic handover and to improve end-to-end cross-domain latency. On top of the unified network, a unified AI-agent based management and orchestration plane will be responsible to manage and orchestrate/coordinate TSN features across different network domains (multi-RAT RAN, transport, core). The management of TSN features will be done on different time scales based on the time criticality of network control loops. The unified AI-agent based network management and orchestration plane will support real-time control loops for TSN features on the RAN and transport, near real-time control loops for TSN features inside the O-RAN components and core and non-real-time control loops for other network policies. The service flows of this use case are as follows:

1) Real-time monitoring on different granularity scale (on per-hope, per-node and per-flow scale) information are shared with the unified AI Agent-based management and orchestration planes. Such information includes low-level information (e.g. Received Signal Strength Indicator (RSSI), Signal to Noise Ratio (SNR), interference levels, current resource utilization) from multi-RAT RAN as well as other monitoring information from transport and core.

2) The network operators will share the information about applications requirements for applications that will run are admitted being started in the network.

3) The AI-agent based management, and orchestration plane will split the task into other sub-task to train AI-agents for different sub-tasks based on monitoring and application requirements information.

4) Different AI-agents will perform different tasks, e.g. one AI-agent will determine the TSN schedules in multi-RAT RAN links, another will determine TSN schedules in transport, another can determine the coordination mechanism between TSN features in different network domain etc.

5) Once the AI-agents' outputs are combined, the decision is communicated from unified AI-agent based management and orchestration plane towards the network nodes that need to be (re)-configured.

## 5.9.4 Use Case #8-4 on AI Agent-Driven Core Network Signalling Optimization

### 5.9.4.1 Description

The Core Network handles a wide variety of control plane processes to support registration, session establishment, mobility management, policy enforcement, and user authentication. As the number of connected devices grows and network services become more diverse, the volume, complexity, and frequency of control plane interactions increase dramatically.

This surge poses challenges in:

1) Control plane overload during busy hours or traffic bursts.

2) Inefficient signalling flows in dynamic or unpredictable network conditions.

3) Redundant or low-value signalling messages that waste processing power and delay user service setup.

To address these challenges, AI Agents embedded within the Core Network that are co-located with the AMF, SMF, PCF, and NWDAF are leveraged to analyse real-time and historical control plane data and dynamically optimize behaviours across the network.

These AI Agents are trained to:

1) Identify high-cost or redundant signalling paths.

2) Detect patterns of signalling storms or inefficient UE behaviours.

3) Predict signalling traffic loads and proactively offload signalling to secondary AMF instances or local UPF branches.

4) Recommend changes to signalling timers, retry policies, registration areas, or PDU session procedures.

5) Trigger intent-based signalling policies (e.g. "optimize for minimal registration latency" or "reduce signalling load during congestion").

6) In some cases, the AI Agent can initiate signalling flow compression, reduce message exchanges for known UE behaviour profiles, or even adapt the network's state machine behaviour for specific use cases (e.g. stationary IoT UEs).

The service flows of AI Agent-driven core network signalling optimization are as follows:

1) The AI Agent continuously monitors appropriate metrics from relevant Network Functions (e.g. the AMF, SMF, and PCF). Examples include registration request frequency, PDU session setups, mobility-triggered signalling, and retransmissions.

2) Based on policies and historical learning, the AI Agent detects excessive signalling in a specific region (e.g. frequent re-registration by stationary IoT devices).

3) It evaluates the situation and:

   - Recommends suppressing redundant signalling or modifying UE policy rules via PCF.

- Instructs AMF to aggregate signalling procedures (e.g. combined registration and session management).
- Informs NWDAF and OAM to adjust mobility area planning.

4) The signalling load is reduced, setup latency improves, and energy consumption is minimized.

The AI Agent continues learning from the effects of these optimizations for future scenarios.

## 5.9.5 Use Case #8-5 on AI Agent-Based Core Networks to Enhance User Experience

### 5.9.5.1 Description

With the rapid development of mobile Internet, users' expectations for network quality and experience have grown significantly. The mobile core network is required to ensure basic communication functions as well as cater to users' diverse and personalized demands. However, traditional core networks face significant challenges in adapting to complex and dynamic user behaviour, network conditions, and heterogeneous service requirements, often falling short of delivering an optimal user experience. Some typical examples of such are listed as follows:

1) During network congestion, high-value users cannot receive timely handling and responses when their services experience abnormalities, thereby affecting the user experience.

2) Insufficient analysis of user equipment performance and network adaptation leads to inefficient resource utilization and compromised user experience.

To address these challenges, AI Agents are envisioned to become deeply integrated sub-components of the mobile core network. These can perceive customer attributes, network status, and service experience in real time, which typically involves multi-agent collaboration cross technical domains (e.g. CN, OSS/BSS, IT domains). By leveraging advanced analysis and decision-making capabilities, they enable significant enhancements to the user experience throughout network usage.

These AI Agents are trained to:

1) Accurately identify customer attributes and distinguish the needs of different user groups based on multi-dimensional data such as user identity, consumption level, and usage habits.

2) Monitor and understand various network-specific data such as network load, link quality, node status.

3) Evaluate service experience by analysing indicators such as service response time, throughput, and packet loss rate from various monitoring systems, and locate key factors affecting user experience.

4) Adjust resource allocation strategies dynamically based on comprehensive analysis of customer attributes, network status, and service experience, giving priority to high-priority traffic and critical services.

5) Predict user behaviour and business demand trends, schedule resources and adjust strategies in advance to achieve proactive service assurance.

6) Learn from historical allocation results and services feedback to continuously optimize the core network resource scheduling model and adapt to new scenarios.

In such a setting, the service flows of AI agent-based core networks to enhance user experience are as follows:

1) The system performs continuous, multi-source data ingestion into a central Knowledge Graph (KG).

   a) Monitoring Agents within the network domain stream real-time telemetry (traffic, link status, service indicators) into the KG.

   b) A specialized BSS/IT Integration Agent securely queries the operator's CRM and BSS databases and populates the KG with relevant, anonymized customer attributes (e.g. user_ID: 123 has tier: "platinum" and active_SLA: "business_video").

2) A specialized Analytics Agent (akin to a 3GPP NWDAF) continuously queries the now-rich KG. It executes complex queries that correlate data across domains, such as: "Find all users with tier: "platinum" whose active video service flows are currently experiencing packet loss > 5 %." This allows the agent to identify not just network problems, but network problems that are impacting high-value customers and violating their SLAs.

3) According to the analysis results, the Analytics Agent detects an SLA violation for a VIP user and raises an alert. This alert triggers the central AI Orchestrator, which plans and delegates a remediation strategy:

   a) VIP Video Conference: The Orchestrator issues a command to the Policy Control Function (PCF) in the 5G Core, instructing it to dynamically increase the QoS Class Identifier (QCI) for the VIP user's specific data flow, ensuring it receives priority treatment.

   b) Gaming Congestion: The Orchestrator, recognizing a pattern of many users accessing the same gaming service, instructs the SMF to steer the UPF selection for these users towards an edge UPF with a more direct, low-latency path to the game servers.

   c) Device Performance: The Orchestrator, having ingested the user's device capabilities into the KG, can instruct the PCF to apply a policy that signals an application server (e.g. a video platform) to serve a lower-bitrate stream to that specific device, ensuring a stable experience.

4) The Monitoring Agents continue to stream telemetry, allowing the Analytics Agent to verify that the remediation actions had the desired positive impact on the user's QoE. This feedback is used to continuously refine the AI models, improving the effectiveness of future interventions.

# 6 Recommendations to other ENI Work Items

The requirements extracted from the Use Cases captured in the present document are specified in the Requirement document ETSI GS ENI 002 [2]. These requirements are divided into service requirements, functional requirements, and non-functional requirements. The service requirements are further sub-divided into:

- general requirements;
- service orchestration and management;
- network planning and deployment;
- network optimization;
- resilience and reliability;
- security and privacy.

The functional requirements are further sub-divided into: data collection and analysis, policy management and data learning. The non-functional requirements are further sub-divided into: performance requirements, operational requirements and regulatory requirements.

The ENI architecture ETSI GS ENI 005 [3] will support all the requirements generated from the Use Cases.

# Annex A (informative):<br>Bibliography


- ETSI GR ENI 003 (V1.1.1): "Experiential Networked Intelligence (ENI); Context-Aware Policy Management Gap Analysis".

# History

| Version | Date | Status |
|---|---|---|
| V1.1.1 | April 2018 | Publication |
| V2.1.1 | September 2019 | Publication |
| V3.1.1 | December 2020 | Publication |
| V3.2.1 | May 2023 | Publication |
| V4.1.1 | May 2026 | Publication |