***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.

ETSI GS ENI 033 V4.0.6 (2024-03)

**Group Specification**

Experiential Networked Intelligence (ENI);

Definition, Requirements and Procedure of Intent Policy Multi-Stage Translating

<

Reference

DGS/ENI-0033v411\_MSTranslating

Keywords

policy management, intent management

***ETSI***

650 Route des Lucioles

F-06921 Sophia Antipolis Cedex - FRANCE

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

Siret N° 348 623 562 00017 - APE 7112B

Association à but non lucratif enregistrée à la

Sous-préfecture de Grasse (06) N° w061004871

***Important notice***

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

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

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

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

If you find a security vulnerability in the present document, please report it through our

Coordinated Vulnerability Disclosure Program:

<https://www.etsi.org/standards/coordinated-vulnerability-disclosure>

***Notice of disclaimer & limitation of liability***

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or

other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

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

***Copyright Notification***

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.
The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2023.

All rights reserved.

Contents

Intellectual Property Rights 5

Foreword 5

Modal verbs terminology 5

Executive summary 5

Introduction 5

1 Scope 6

2 References 6

2.1 Normative references 6

2.2 Informative references 6

3 Definition of terms, symbols and abbreviations 6

3.1 Terms 6

3.2 Symbols 7

3.3 Abbreviations 7

4 Overview of Intent Policy Multi-Stage Processing 7

4.1 Introduction 7

4.2 Description of Intent Policy Multi-Stage Processing 8

4.3 Functional Architecture 8

4.4 Procedures of Intent Policy Multi-Stage Processing 9

4.4.1.1 Overview 10

4.4.1.2 The Sequence Generation Method 10

4.4.1.3 Use Cases 10

4.4.3.1 Motivation 11

4.4.3.2 The Process of Generating User-Level Templates 11

4.4.3.3 Benefits 12

5. Intent Policy Multi-Stage Processing Design Requirements 12

5.1 Introduction 12

5.2 Functional Requirements for Intent Policy Multi-Stage Processing 13

5.3 Non-Functional Requirements for Intent Policy Multi-Stage Processing 13

6. Use Cases 13

6.1 Use Case #1: User intent to the network polices 13

6.1.1 Use case context 13

6.1.2 Description of the use case 14

6.1.2.1 Overview 14

6.1.2.2 Motivation 14

6.1.2.3 Actors and Roles 15

6.1.2.4 Initial context configuration 15

6.1.2.5 Trigger conditions 15

6.1.2.6 Operational Flow of the actions 15

6.1.2.7 Post-conditions 15

Annex A (normative or informative): Title of annex 17

A.1 First clause of the annex 17

A.1.1 First subdivided clause of the annex 17

Annex (informative): Bibliography 18

Annex (informative): Change History 19

History 20

# Intellectual Property Rights

Essential patents

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

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

Trademarks

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

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

# Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group 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](https://portal.etsi.org/Services/editHelp%21/Howtostart/ETSIDraftingRules.aspx) (Verbal forms for the expression of provisions).

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

# Executive summary

The present document specifies a high-level functional abstraction of the process of intent policy Multi-Stage translating in ENI system in terms of Functional Modules, Internal Reference Points and working pipelines.

# Introduction

The present document defines a high-level functional abstraction of Intent Policy Multi-Stage Processing. The organization of the present document is as follows. Clause1 defines the scope of the present document. Clauses 2 and 3 provide normative and informative references and definition of terms, respectively. Clause 4 provides an informative overview of Intent Policy Multi-Stage Translating, including its motivation, benefits, important concepts and an overview of its Functional Modules. Clause 5 defines important design principles of the processing, and then specifies the different Functional Modules that make up the processing. Clause 6 provides some use cases of Intent Policy Multi-Stage Processing.

# 1 Scope

This work is augments existing intent policy translating procedure in ENI. The purpose of this work item is to describe intent policy multi-stage translating in ENI system, and to enhance intent policy multi-stage translating.

This GS also defines the output(s), input(s), internal process and interaction of every stage during intent policy multi-stage translating.

Intent policy multi-stage translating is a detailed procedure that can translate an intent policy according to the Policy Continuum. There is an external knowledge graph to be added to provide a set of multi-stage general processing scheme for intent policy.

# 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 at [https://docbox.etsi.org/Reference](https://docbox.etsi.org/Reference/).

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 005 (V3.1.1): "Experiential Networked Intelligence (ENI); System Architecture".

[2] ETSI GS ENI 030 (V3.1.1): "Experiential Networked Intelligence (ENI); Representing, Inferring, and Proving Knowledge in ENI".

[3] ETSI GS ENI 019 (V3.1.1): “Transformer Architecture for Policy Translation”

## 2.2 Informative references

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

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

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

[i.1] <Standard Organization acronym> <document number><version number/date of publication>: "<Title>".

[i.2] etc.

# 3 Definition of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the [following] terms [given in ... and the following] apply:

**AI model**: in this article, it refers to models that are capable of processing the understanding and generation of natural language.

**business intent**: abstract network intent input by the user

**model parser**: processing user intent using large language models or NLP methods

**network entities**: entities representing partial network information, such as time, bandwidth, etc.

**intent-level templates**: network intent with fixed format

**service-level templates**: template incorporating Quality of Service, Access Control List, Service-Level Agreements, and Network Function Virtualization, etc.

**user-level templates**: service-level template that incorporates user preferences, device information, etc.

**network polices**: a domain specific language (DSL) generated from a user-level template.

**conditional random fields**: a probabilistic structural model, which is used for sequence labeling and word segmentation tasks, can be regarded as undirected graph model or Markov random field.

**knowledge graph**: a structured representation of knowledge that uses a graph data structure and formal logic to represent entities and their relationships

**named entity**: a word or phrase that refers to an item or process of interest

**named entity recognition**: a task in information extraction that identifies named entities in data

**network slice**: the technology of cutting a physical network into multiple virtual networks to meet the needs of different services.

**on-demand service**: a service that can be provisioned and used as needed

**part-of-speech tagging**: a natural language processing technique used to determine the grammatical category of each word in a sentence, such as nouns, verbs, adjectives, etc.

## 3.2 Symbols

For the purposes of the present document, the [following] symbols [given in ... and the following] apply:

## 3.3 Abbreviations

For the purposes of the present document, the [following] abbreviations [given in ... and the following] apply:

**AI**: Artificial Intelligence

**BIO**: Beginning-Inside-Outside

**BMES**: Beginning-Middle-End-Singleton

**CRF**: Conditional Random Fields

**ML**: Machine Learning

**LLM**: Large Language Model

**NER**: Named Entity Recognition

**NLP**: Natural Language Processing

**PoS**: Part-of-Speech

**QoE**: Quality of Experience

**QoS**: Quality of Service

# 4 Overview of Intent Policy Multi-Stage Processing

## 4.1 Introduction

This clause provides an informative introduction to Intent Policy Multi-Stage Processing in the ENI Policy Management Functional Block. Clause 4.2 describes the background and motivation of Intent Policy Multi-Stage Processing, and then provides a high-level description of Intent Policy Multi-Stage Processing in the ENI system. Clause 4.3 describes the functional architecture of Intent Policy Multi-Stage Processing in terms of Processing Stages. Clause 4.4 introduces each step of Intent Policy Multi-Stage Processing.

## 4.2 Description of Intent Policy Multi-Stage Processing

4.2.1 Background

With the development of the research of the sixth-generation wireless communication network (6G), the network business scenarios are increasingly diversified. The integration of artificial intelligence and communication technology has become a strong support for new business scenarios, further driving the development of the network.

The key challenge of current 6G research is to achieve on-demand services in all scenarios. Here, "on-demand" is a service that can be provisioned and used as needed, as opposed to traditional services that have to be provisioned in advanced. Traditional services also are typically used for a fixed period of time. On-demand services reflect the specific needs of users; this in turn requires the accurate understanding of the user's intent, and the ability to efficiently extract key information and convert it into accurate network policies in order to achieve fully automated networks. Providing a high-quality customer experience for verticals and consumers is an essential part of the research in 6G technologies.

4.2.2 High-Level Description of Intent Policy Multi-Stage Processing

This framework decomposes the user's natural language intent into multiple levels, and makes the abstract intent concrete step by step using the method of hierarchical decomposition. This ensures the accurate, timely and standardized translation of a natural language intent policy into a multi-level fine-grained network template that can support on-demand services.

## 4.3 Functional Architecture

Intent Policy Multi-Stage Processing replaces the abstract entities in the natural language intent input from the user with the corresponding entities in the network, and expresses the expected network state as quantifiable QoS, QoE and/or descriptive network operations. It consists of four steps: generate intent-level templates, generate service-level templates, generate user-level templates, and generate network policies.

Among the above steps, the application of named entity recognition (NER) and knowledge graph is an essential part. Named entities generally refer to entities with specific meanings or strong references in the text, which usually include names of people, places, names of organizations, dates and times, proper nouns and so on. The purpose of NER is to extract the above entities from unstructured input text, and to identify more types of entities according to business requirements, such as product name, model, price, etc. Knowledge graph is a graphical representation method that connects related data with entities as the center. It contains a large number of entities and relationships. In this way, the complex relationships between entities can be clearly displayed, which can help users better understand the data and discover the patterns and laws hidden in the data.

In the first step of intent-level templates generation, entity extraction is firstly carried out from the AI model, along with their attributes and relationships, and how they can be used in ENI Policies.Then, according to the results of entity extraction in the previous step, different service-level templates are selected according to the types of multiple entities and the predicted user intent. In the step of generating user-level templates, the technology of "AI model external knowledge base" is used, that is, the knowledge graph is input into AI model as knowledge base. After AI model generates an intent-level template and converts it into a service-level template, it will fill the service-level template according to the equipment information and user preferences in the knowledge graph, and then convert it into a user-level template, the purpose is to achieve the optimal QoS through the knowledge graph on the basis of meeting the overall network policy intent. Finally, network policies are generated based on user-level templates. The specific steps are as follows:

* Step 1: Input the natural language intent entered by the user into the AI model to obtain the corresponding entities in the network.
* Step 2: Extract different network attribute demands from the original input and select appropriate service-level templates. For example, the access control list template of the online meeting service helps users connect to the primary server, and the service-level agreement template allocates resources such as bandwidth to users.
* Step 3: Achieved the measurement specified by QoE of the service template augmented and/or supplemented by information from the knowledge graph.
* Step 4: Adjust the user-level template based on the user device, access network status, and user preferences.
* Step 5: Convert user-level templates into network policies using specific algorithms defined in a JSON config file.

Network entities in Step 1 mainly include:

1. Operation (OPT): Specific operations or tasks performed on the target entities, such as adjust, open, etc.
2. Target (TGT): a specific object or entity that should be operated in a specific task, such as gateway, students, etc.
3. Service (SRV): Functions or applications provided to users, such as video conference.

NOTE: The following three service-level templates are used as examples in Step 2: access control list (ACL) templates, service-level agreement (SLA) templates, and network function virtualization (NFV) templates. For details and examples, see the following section.



Figure 4.3.1. Intent Policy Multi-Stage Processing Architecture Diagram

## 4.4 Procedures of Intent Policy Multi-Stage Processing

4.4.1 Generate Intent-Level templates

#### 4.4.1.1 Overview

Because users typically cover complex online needs and some subtle expressions of demand in a single input process, it is crucial to categorize user needs into different intent-level templates based on the results of entity extraction first.

This clause describes how to convert abstract user intents into corresponding network entities through the AI model, and then generate the corresponding intent-level templates. The benefits of this approach and the requirements are also described.

In the process of the Intent Policy Multi-Stage Processing, the main task of generating intent-level templates is to identify the key vocabulary and semantic information based on the user input text information, and then combine the above information into a complete intent-level template.

Specifically, in this stage, the AI model preprocesses the text input by the user (i. e. business intent) and generates a complete intent-level template to better understand the user's intent. Intent-level templates contain a variety of keywords, such as operation, target entity, service, etc.

The purpose of generating intent-level templates is to be able to more accurately understand the user's abstract intention, and to provide the basis for subsequent processing, which directly affects subsequent processing and the user experience. A good intent template should have the following characteristics:

* Be able to clearly express the user's intent;
* Contains enough information to guide subsequent processing;
* Has a certain flexibility, can adapt to different use scenarios and user needs.

#### 4.4.1.2 The Sequence Generation Method

Sequence generation is a natural language processing (NLP) task that aims to generate a sequence or text based on given conditions or contextual information. Sequence generation can be used in many different application areas, including dialogue systems, summary generation, machine translation, text classification, etc.

Sequence-generated tasks can generally be divided into two types:

* conditional sequence generation
Conditional sequence generation is the generation of a sequence or text based on a given condition. These conditions can be input text, images, speech, etc., as well as generated output sequences or text. Conditional sequence generation is often used in dialog systems, question answering systems, text classification and other tasks.
* unconditional sequence generation
Unconditional sequence generation is the generation of sequences or text without regard to any conditions or context information. This type of task is often used in applications such as text summarization.

In sequence generation, different methods and techniques can be used to generate sequences or text, including rule-based methods, probability-based methods, and neural network-based methods.

A rule-based approach uses predefined rules to generate sequences or text. These rules can be formulated according to domain knowledge and linguistic knowledge, such as grammar rules, vocabulary collocation rules, semantic rules, etc. Rule-based approaches are often appropriate for specific tasks and domains, and require specialized knowledge and experience to develop rules.

Probability-based methods use probabilistic models to generate sequences or text. These models typically use Hidden Markov models (HMM), conditional random fields (CRF), or probability graph models to model the probability distribution of a sequence. By calculating probabilities, the best generated sequence or text can be found and optimized and adjusted. Probability-based methods are generally suitable for tasks with short sequences, such as named entity recognition, part-of-speech tagging, etc.

Neural network-based approaches use deep learning techniques to generate sequences or text. These techniques typically use recurrent neural networks (RNNS) or Transformer models to learn representations of input sequences or text and use those representations to generate output sequences or text. Neural network-based methods are generally suitable for tasks with long sequences, such as machine translation, text classification, etc.

#### 4.4.1.3 Use Cases

When the user inputs the natural language intent, AI model will convert it into an intent-level template. The format of intent-level template has certain restrictions, and it is generally < Entity Type >: Target Entity. Entity types include users, groups, endpoints, services, time, bandwidth, traffic and so on.Then, based on the generated intent-level templates, they are used as a reference to generate service-level templates.

* User Intent: The online meeting of all students will be held at 9:00 today.
* Intent templates:<Object>: all students, <Service>: online meeting, <Time>:9:00

4.4.2 Generation of Service-Level Templates

This clause describes how to select the corresponding service-level templates based on the entity type and the predicted user intention, and gives some examples of service-level templates.

In complex network scenarios, routing control, resource allocation, and VNF deployment coordination are often required to achieve network status. Different network architecture levels and adjustment requirements require different policy description standards. The labeled word-encoded sequence output can be used as input for deep learning, predicting the possible intentions based on the requirement and selecting the corresponding template. According to the standard network slice service function, network slice template can be divided into various templates such as QoS (quality of service) template, ACL (access control list) template, SLA (service-level agreement) template and NFV (network function virtualization) template. The following takes QoS template and ACL template as examples to illustrate the process of generating service level template.

Step 1: Determine the service requirements and determine the network slice template type according to the intent-level template generated in the previous step.

For example, the intent level template for intent: The online meeting of all students will be held at 9:00 information is: <Object>: all students, <Service>: online meeting, <Time>: 9:00. The slice template required for intention guess are QoS template and ACL template.

Step 2: Conform to the corresponding service level template according to the intention level template. If necessary, the corresponding entity information can be filled according to the current network environment, business scenarios and performance requirements.

For example, the Qos template generated by converting the intention-level template generated in the previous step is: <Object>: all students, <Service>:online meeting, <Time>:9:00, <QoS>:[20Mbps, 100Mbps], The resulting ACL template is: <Src>: IP of all students, <Dst>:IP of the server, <Time>:9:00, <Middlebox>:(Firewall, close)

4.4.3 Generation of User-Level Templates

#### 4.4.3.1 Motivation

Entity extraction combined with general service standard method can determine the current service-level templates, but because in the same service, different user equipment, user preferences, etc., also can bring the difference of user demand and experience, lead to the method is difficult to handle the differences between different user transmission, then cannot achieve the optimal QoS. By introducing an external knowledge graph, high customization can be achieved to better meet the specific needs of the user groups.

#### 4.4.3.2 The Process of Generating User-Level Templates

There are four main steps to creating user-level templates:

Step 1: Introduce the external knowledge graph to assist in building a multi-user QoE model.

Step 2: decompose the service-level templates at the user layer, and adjust the decision differently according to the refinement differences between different transmission pairs in the previous layer template.

NOTE: Differentiation adjustment mainly includes the underlying network operational objects and the fine-grained quantified indicators or network operations expected by the underlying network corresponding to the previous layer template.

Step 3: For the necessary information in the underlying network transmission layer, and adjust the index according to the user device, access network status and user preference.

Step 4: Generate user-level templates based on external information.

Here is an example of this section:



Figure 4.4.1 Example of generating user-level templates

#### 4.4.3.3 Benefits

The introduction of user-level templates during intent policy multi-level processing has several benefits:

1. It enables the strategy generation module to make differentiated decisions at the user level and ensure highly customized network slices to meet the specific needs of different user groups.
2. It reduces the time and computational cost of network slice configuration, provides high-quality services, and maximizes the efficiency of network resources.
3. Using external knowledge, the framework is portable and scalable to quickly adapt to a variety of personalized network service scenarios.

4.4.4 Generation of Network Polices

In the intent policy multi-level processing stage, the final step is to generate the corresponding Domain Special Language (DSL) according to the user-level templates generated in the previous step, and the details will be expanded in the following chapters.

# 5. Intent Policy Multi-Stage Processing Design Requirements

## 5.1 Introduction

The following clauses define functional and non-functional requirements of Intent Policy Multi-Stage Processing, respectively.

 NOTE: In Release 4, the clauses focus on the knowledge graph, the form of the intent policy and the category of the intent template.

## 5.2 Functional Requirements for Intent Policy Multi-Stage Processing

1. The ability to deal with complex abstract intentions. Intent policy multi-stage processing should be able to identify and extract relevant information from different types of user intents.
2. The ability to extract abstract intents into network entities and establish intent templates. The system should be able to use a reasonable AI model to mine the intent entities and output them into a series of key network entities labeled with the corresponding entity category to establish the corresponding intent templates.
3. The ability to predict and select the corresponding service-level template according to the demands. The system should be able to predict different policy description criteria based on the output entity category, entity content, and different network architectures and adjustment requirements, and then choose a more accurate network slicing template.
4. The ability to learn and update network slicing templates according to the knowledge graph. The system should be able to continuously learn the user needs and differences in different scenarios through the knowledge graph, and establish a multi-user QoE model according to the current different user devices and user preferences, and make differentiated adjustments to the template of the previous layer.
5. The ability to generate corresponding network policies according to the network slicing templates. The system should be able to identify the corresponding network state based on the output user-level template and translate it into the corresponding network configuration and policies.

## 5.3 Non-Functional Requirements for Intent Policy Multi-Stage Processing

  **Reliability**: The system should ensure high availability and stability to prevent user data loss caused by faults.

  **Scalability**: The system should be able to support large-scale users and data, and can be flexibly expanded with the development of the business.

  **Security**: The system should ensure the security and privacy of data to avoid the disclosure of sensitive information and attacks.

  **Maintainability**: The system should be easy to maintain and upgrade, including code readability, reusability, testability and other aspects.

  **Ease of use**: The system should provide a friendly user interface and interaction methods, so that users can easily use the system.

  **Performance**: The system should have high performance and efficiency, and can quickly process a large number of requests and data.

# 6. Use Cases

## 6.1 Use Case #1: User intent to the network polices

### 6.1.1 Use case context

This use case covers the entire process from user intent to network policies. From the initial abstract intent input by the user, to the intermediate state templates, which are divided into service-level and user-level templates, to the final network policies, each step refines and visualizes the previous step, while also laying the foundation for the generation of the next step and providing clearer guidance.

### 6.1.2 Description of the use case

#### 6.1.2.1 Overview

The user inputs natural language intent into the system so that the system can generate network policies that can be issued based on the user's needs, preferences, and current network environment. The process includes:

* Inputting user intent. Compared to the specific configuration and parameter information entered by the network administrator in traditional cases, user intent only contains an abstract requirement expressed in natural language. For example, "I want to watch live football matches at 7 p.m."
* Grouping of intent-level templates. The parameters of intent-level templates have similarities with the specific parameters and configurations mentioned in the previous step as input by the network administrator, but they cannot be applied directly. They can be considered as a representation of abstract user requirements, forming intermediate template states with certain formats and expressive capabilities to adapt to different types and formats of user input.
* Generation of service-level templates. This template is further transformed based on the intent-level template determined in the previous step, considering service requirements and network slicing templates. It has more information about network environments and business scenarios and is mainly divided into QoS templates and ACL templates.
* Generation of user-level templates. This template is based on the service-level template and takes into account user diversity, device diversity, and network environment diversity, formulating templates that best meet the needs of each user.
* Generation of network policies. This step extracts and combines information from the service-level templates through algorithms to form a down-gradable Domain Special Language (DSL), which is the network policy.

#### 6.1.2.2 Motivation

Under the traditional network management model, enterprises need to hire professional network administrators to configure and manage the network. These administrators need to have deep network knowledge and experience, and be able to skillfully input various parameter information to ensure the normal operation of the network. However, this approach not only puts high demands on the human resources of the enterprise, but also often makes the management process cumbersome and complex, prone to errors.

However, with the support of AI models, network management becomes more intelligent and efficient, and the requirements for personnel are downgraded from professional network administrators to ordinary users. This means that even users without network management experience can manage the network through simple operations. The key to the implementation of this intelligent network management method lies in the way AI models handle input information. In traditional network management, administrators need to input specific parameter information for configuration and management. However, driven by AI models, they only need to input abstract intents, such as "I want to upload large files to a cloud server", and AI models can automatically translate them into corresponding network policies for network management. This approach greatly simplifies the complexity of network management, lowers the management threshold, improves management efficiency, and at the same time, AI models can continuously learn and optimize their own algorithms, constantly improving the accuracy and efficiency of network management.

The emergence of intent-level templates makes network management more intelligent and efficient. Compared with traditional network management, user input language often should follow certain formats and norms. With the help of intent-level templates, users can input their intent more freely, without worrying about whether their language cannot be recognized by the network or if they have missed some important information. It automatically translates user input intent into key-value relationship pairs through entity mining at both word level and semantic level. This technology has high accuracy and adaptability, and can adapt to different types and formats of user input.

Service-level templates and user-level templates are intermediate state templates with the same format. They are both network management technologies based on AI models, with higher flexibility and adaptability. Service-level templates determine template types according to user needs, such as QoS templates, ACL templates, etc. These templates allow users to control network behavior at a finer granularity, thus better meeting user needs. For example, the QoS template can be used to set the priority of network traffic, thereby ensuring the smooth transmission of important business traffic. User-level templates are templates that correspond to user input. It is not only translated from the intent of user input, but also combined with factors such as user's preference habits, current network environment, etc., to give the best templates. This technology allows AI models to directly learn human preferences and habits, thus better meeting user needs.

#### 6.1.2.3 Actors and Roles

* Regular user.
* Network administrator.
* ENI System.

#### 6.1.2.4 Initial context configuration

* Regular users input abstract intentions;
* AI models are launched and ready to go.

#### 6.1.2.5 Trigger conditions

* The change in user intent will cause the AI model to re-analyze and mine intent entities.
* The change in the network environment will cause the user-level templates to be reconstructed.

#### 6.1.2.6 Operational Flow of the actions

* Users input abstract natural language intent to express their needs for network configuration. For example: "Hello, please allow access to the SSH of the Linux servers.";
* The AI model performs entity mining on user intent to form key-value relationship pairs. At the same time, the AI model deeply understands the intent statement to form other entity information that is not explicitly required but necessary in the intent statement. For example:<Group>:Linux Servers, <Middlebox>:Firewall, <Protocol>:SSH;
* The AI model selects the corresponding service-level templates based on the entity type and predicted user intent, mainly including QoS (quality of service) template, ACL (access control list) template, For example, the QoS template is: <Group>:Linux Servers, <Middlebox>:Firewall, <Protocol>:SSH, <QoS>:[2Mbps, 10Mbps].The ACL template is: <Src>: IP of all students, <Dst>:IP of the server, <Group>:Linux Servers,<Middlebox>:(Firewall, close), <Protocol>:SSH;
* The system introduces an external knowledge graph to help establish a multi-user QoE/QoS model;
* Different strategies are adjusted for the detailed differences in different key-value pairs in the service-level template, while user-level templates are established based on user preferences and the current network environment. For example: Slicing template 1：<Src>:Ip if user 1, <Group>:Linux Servers, <Middlebox>:(Firewall, close), <Protocol>:SSH, <QoS>:[2Mbps, 10Mbps]. Slicing template 2：<Src>:Ip if user 2, <Group>:Linux Servers, <Middlebox>:(Firewall, close), <Protocol>:SSH, <QoS>:[2Mbps, 10Mbps]. Slicing template 3:...;
* The system translates the user-level templates of different users by calling algorithms to generate network policies for each user. For example:[ x : (ipDst =192.168.0.2/24, SSH=\*) -> .\* dpi .\*.bandwidth(2,10,mbps) ]

#### 6.1.2.7 Post-conditions

* The user's intent has been accurately understood and parsed, and has been translated into executable network policies.
* The network policies have been optimized based on the user's preferences and the current network environment.
* The network policies have been successfully issued to the corresponding network devices, such as firewalls, routers, etc.
* In a multi-user environment, each user's QoE/QoS is guaranteed and not affected by competition for network resources.
* If the network environment changes, the system can automatically adjust the network policy to maintain the best quality of service.

Annex A (normative or informative):
Title of annex

# A.1 First clause of the annex

## A.1.1 First subdivided clause of the annex

Annex (informative):
Bibliography

Annex (informative):
Change History

| Date | Version | Information about changes |
| --- | --- | --- |
| 04/2023 | V0.0.1 | Initial Draft |
|  |  |  |
| 12/2023 | V0.0.4 | Use Case #1 Added. |
|  |  |  |

# History

|  |
| --- |
| **Document history** |
| <Version> | <Date> | <Milestone> |
| v0.0.1 | 2023-04-03 | Start of work |
|  |  |  |
|  |  |  |
|  |  |  |

*Latest changes made on 2022-03-14*