ETSI GR ENI 015 V4.1.1 (2024-05)

GROUP REPORT

Experiential Networked Intelligence (ENI); Processing and Management of Intent Policy

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
DGR/ENI-0025_Int_Policy_proces

Keywords
artificial intelligence, network

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 Web server (https://ipr.etsi.org/).

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

Trademarks

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

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

Foreword

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

Modal verbs terminology

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

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

1 Scope

The present document describes the following topics:

- Enhanced procedures for processing Intent Policy, e.g.:
    - detail the Procedures of intent policy processing;
    - conflict detection and resolution between different Intent Policies.
- Knowledge management for Intent Policy, including:
    - how to use a Knowledge Graph to manage Intent policies;
    - how to use a Knowledge Graph for managing Intent policy knowledge;
    - procedures for lifecycle management of intent knowledge, e.g. import, update, delete, and query of the intent knowledge.
- Typical use cases and requirements which can reduce the management complexity for Intent Users, e.g.:
    - use cases for Business Users/Operational Users/Technical Users that are all users of Intent.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

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

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

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

[i.1] ETSI GS ENI 005 (V3.1.1): "Experiential Networked Intelligence (ENI); System Architecture".

[i.2] ETSI GR ENI 008 (V2.1.1): "Experiential Networked Intelligence (ENI); InTent Aware Network Autonomicity (ITANA)".

[i.3] ETSI GR ENI 016 (V2.1.1): "Experiential Networked Intelligence (ENI); Functional Concepts for Modular System Operation".

[i.4] ETSI GS ENI 030 (V4.1.1): "Experiential Networked Intelligence (ENI); Transformer Architecture for Policy Translation".

[i.5] IEEE Access 8 (2020): "Knowledge graph completion: A review" 192435-192456, Chen Zhe et al.

[i.6] TM Forum IG1253 (V1.2.0): "Intent in Autonomous Networks".

[i.7] MEF 95: "MEF Policy Driven Orchestration", Strassner, J, editor, July 2021.

[i.8] ETSI GS ENI 019 (V3.1.1): "Experiential Networked Intelligence (ENI); Representing, Inferring, and Proving Knowledge in ENI".

[i.9] Stanford definition of "Propositional Logic".

[i.10] ETSI GS ENI 033 (V4.1.1): "Experiential Networked Intelligence (ENI); Definition, Requirements and Procedure of Intent Policy Multi-Stage Translating".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

bag-of-words model: model which represents text data as a collection of words, regardless of the order of words

NOTE: The bag-of-words model can determine the important words in the text by counting the number of occurrences of each word in the text data.

formal logic: study of propositions, statements, or assertively used sentences and of deductive arguments. Formal logic is the basis of deductive reasoning, which is used to derive a conclusion from a set of premises with certainty

NOTE: Formal logic can also be used to prove the correctness of algorithms by reasoning formally or mathematically about the algorithm.

graph database: database that uses graph structures to store and navigate relationships (i.e. edges) between entities (i.e. nodes) in a graph

information extraction: process of identifying and extracting words and phrases with potential semantic information from text data

knowledge fusion: integration of knowledge from different sources, forms, and structures

knowledge graph: directed cyclic graph that embeds semantics in the nodes and edges using a formal logic (RDF as a minimum; types of Description Logics, such as or an OWL 2 Profile, are preferred)

named entity recognition: identification of entities with specific meanings in data such as text or images, such as person names, organizational structures, place names, etc.

part of speech tagging: process of determining the part of speech of words to better understand the text

quality improvement: process of improving the accuracy, timeliness, completeness, reliability, and robustness of knowledge graph through a series of technical means and algorithms

relationship extraction: process of extracting relationships between entities from raw data

NOTE: Relationship extraction includes character pattern based extraction methods, grammar pattern based extraction methods, and semantic based extraction methods.

state machine: abstract mathematical model used to describe the patterns of transitions and behaviours of an object between various states

topic model: model which is used to identify topics or keywords in text data, such as Latent Semantic Analysis (LSA) based on probabilistic topic model

word embedding: technique used in natural language processing and machine learning to represent words as low dimensional vectors that can be used for tasks such as text classification and sentiment analysis

NOTE: Word embedding can be obtained through neural network learning or manually compiled.

3.2 Symbols

Void.

3.3 Abbreviations

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

ACLs	Access Control Lists
API	Application Programming Interface
BSS	Business Support Systems
CPL	Cloud Private Line
CSRUD	Create-Store-Read-Update-Delete
DL	Description Logic
DSL	Domain Specific Language
GDB	Graph Database
IDS	Intrusion Detection Systems
IPFB	Intent Parser Functional Block
KG	Knowledge Graph
LSA	Latent Semantic Analysis
NAS	Network Attached Storage
NER	Named Entity Recognition
NLP	Natural Language Processing
O&M	Operation and Maintenance
OPEX	Operational Expenditure
OSS	Operations Support System
OTN	Optical Transport Network
OWL	Web Ontology Language
PMFB	Policy Management Functional Block
QoS	Quality of Service
RAN	Radio Access Network
RDF	Resource Description Framework
SPO	Subject-Predication-Object
UHD	Ultra High Definition
URI	Uniform Resource Identifier
VR	Virtual Reality
W3C	World Wide Web Consortium
XR	Extended Reality

4 Background and Overview

4.1 ENI Purpose

Operational Expenditure (OPEX) for network management is one of the operators' biggest concerns. The use of intent in ENI can help operators effectively reduce OPEX. Intent enables different constituencies to express intent policies using a language that is natural to themselves. Hence, the associated business benefit is to understand business needs and use them to determine offered services and resources. This means that offered services can dynamically adapt to changing user needs, business goals, and environment conditions, which will greatly improve the efficiency of network management and reduces the cost of network Operation and Maintenance (O&M).

4.2 Intent Work in ENI

Intent Policy expresses the goals to be accomplished by using a restricted natural language (e.g. an external DSL), without focusing on how to achieve those goals. The definition and introduction about intent policy can be found in clause 6.3.9.3 of ETSI GS ENI 005 [i.1], and a previous study in ETSI GR ENI 008 [i.2] further discussed the architecture of intent policy (in clause 5.1.2), life cycle management (in clause 5.3) and the translation process of intent policy (in clause 5.2.3), as well as related use cases (in clause 6).

4.3 Introduction to Knowledge

Knowledge is defined in clause 6.3.4 of ETSI GS ENI 005 [i.1] and further detailed in clause 6.3.4 of ETSI GS ENI 005 [i.1]. Briefly, knowledge analyses data and information, understands its meaning, and can predict what has happened, is happening, or what is possible to happen in the future. Knowledge management is an essential part in intent aware network, which is responsible for managing the life cycle of the knowledge in the Data and Knowledge Repositories, including storage, assessment, use, sharing, and refinement of knowledge assets. ENI develops knowledge and wisdom from observed, measured, and inferred data and information, as described in ETSI GR ENI 016 [i.3].

5 Procedures of Intent Policy Processing

5.1 Introduction

In ETSI GR ENI 008 [i.2] and ETSI GS ENI 005 [i.1], the procedures of intent policy translation are shown in different detail. In general, the intent policy is created by the intent creator and sent to the ENI System. The ENI System translates and processes the received intent policy and notifies the intent creator of any errors. If there are no errors, then the intent policy is executed on the selected component(s) of the Assisted System.

Considering the complexity of user's goals, the intent policy can be separated into multiple intent policies to simplify the general procedures of intent policy processing, such as translation and named entity recognition. The processed intent policy has three main lifecycle phases (see clause 6.4 of the present document): Translation, Deployment, and Execution.

The benefits of this process are as follows:

1) Ensure the realization of the intent creator's goals by performing two functions automatically:

   a) Translating the intent creator's policy from natural language to a form that can be verified by the ENI System. Subsequent translations could be performed to make the translated policy implementable by the Assisted System. If the translation fails, then errors found will be sent to the intent creator, and processing stops until the intent creator resubmits an edited intent policy.

   b) Testing any new intent policies to ensure that they can be executed correctly. If the testing results are not able to meet the goals defined by the intent creator, then information describing this will be sent to the intent creator, and processing stops until the intent creator resubmits an edited intent policy.

2) Automate the entire process (all processes are triggered automatically, especially automatic testing), while still providing flexibility to the intent creator. For example, the intent creator can have control on each state change of the intent policy lifecycle (see clause 6.4).

3) Support the flexible use of intent policies that are managed by a knowledge process, such as the Knowledge Management Functional Block of ETSI GS ENI 005 [i.1] (see clause 6.3.4 of ETSI GS ENI 005 [i.1]).

NOTE: A group of intent policies could also be managed as a unit (e.g. a high level intent with an associated set of detailed policies on each resource).

5.2 Intent Policy Processing Operation

The original Intent Policy, which is written in a restricted natural language, is sent to the ENI System. It is ingested by the API Broker and sent to the Policy Management Functional Block (PMFB). The PMFB checks to see if the ingested Intent Policy has been seen before (e.g. by matching the text of the newly ingested Intent Policy with the text of a previously stored Intent Policy). If the original Intent Policy matches a previous Intent Policy that has been successfully parsed, verified, and stored, then the ENI System checks to see if anything has changed in the ingested Intent Policy (e.g. parameter values). If nothing has changed, then the PMFB retrieves the previously translated representation of the matched Intent Policy and sends it to the API Broker. The API Broker transforms the translated Intent Policy to a form (i.e. a set of objects and/or code) that the Assisted System can understand. The translated Intent Policy is then sent to the Assisted System. If a change in the ingested Intent Policy is detected, then the PMFB will first verify the proper operation of the Intent Policy by testing it before sending it to the Assisted System. If this test fails, then all errors and warnings are collected and sent back to the Intent Creator. If it passes, then the PMFB sends the verified Intent Policy to the API Broker as described above.

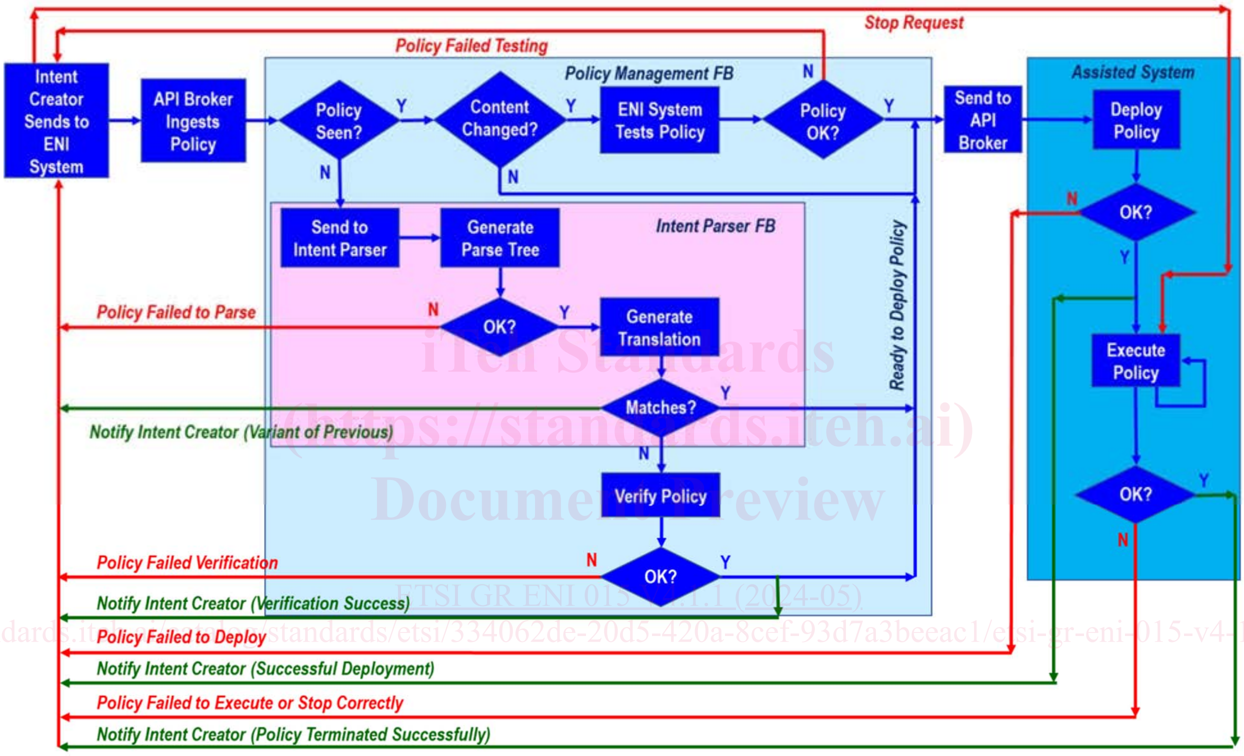

Figure 5.1: Detailed Procedures of Intent Policy Processing

Alternatively, if the original Intent Policy is not found, then it needs to be parsed, optionally compiled and translated, and subsequently verified. This is shown in the rest of Figure 5.1, starting with the "Send to Intent Parser" block. The steps shown in Figure 5.1 are summarized as follows:

1) The input Intent Policy is parsed by the Intent Parser, which is part of the Intent Parser Functional Block (IPFB). The parsing process is described in detail in ETSI GS ENI 005 [i.1], translator architecture in ETSI GS ENI 030 [i.4] and ETSI GR ENI 008 [i.2]:

   a) If the Intent Parser fails to generate a parse tree, then all errors and warnings are collected and sent to the Intent Creator for correction.

   b) Otherwise, the parse tree is generated and sent to the Intent Translation function, which is another Functional Block that resides in the IPFB.