ETSI GR ENI 004 V3.1.1 (2023-07)

Experiential Networked Intelligence (ENI); Terminology

Disclaimer

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

Reference
RGR/ENI-004v311_Terminology

Keywords
artificial intelligence, network, terminology

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 provides terms and definitions used within the scope of the ETSI ISG ENI. The purpose is to define a common lexicon for use across all deliverables of ENI.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

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

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

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

[i.1] ETSI GR NFV 003 (V1.7.1): "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.2] MEF 95: "MEF Policy Driven Orchestration", July 2021.

[i.3] MEF 55.0.3: "Amendment to MEF 55: Service Orchestration Functionality", January 2018.

[i.4] MEF 55: "Lifecycle Service Orchestration (LSO): Reference Architecture and Framework", March 2016.

[i.5] MEF 78.1: MEF Core Model (MCM)", July 2020.

[i.6] Gamma E., Helm R. Johnson R. and Glissades J.: "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, November 1994. ISBN 978-0201633610.

[i.7] ISO/IEC 2382-28: "Information technology -- Vocabulary".

[i.8] ISO/IEC/IEEE 42010: "Systems and software engineering -- Architecture description".

[i.9] The Atlan Data wiki definition of structured data.

[i.10] ETSI GR ENI 007 (V1.1.1): "Experiential Networked Intelligence (ENI); ENI Definition of Categories for AI Application to Networks".

[i.11] IETF RFC 8321: "Alternate-Marking Method for Passive and Hybrid Performance Monitoring".

[i.12] Mitchell, Tom M.: "Machine Learning", McGraw-Hill, 978-0070428072.

[i.13] ETSI Directives.

[i.14] Gruber, Thomas R.: "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", International Journal of Human Computer Studies, Vol 43, pp 907-928, 1993.

[i.15] Strassner, J., Agoulmine, N., Lehtihet, E.: "FOCALE - A Novel Autonomic Networking Architecture", ITSSA Journal 3(1), 64-79, 2007.

[i.16] ETSI TR 103 240 (V1.1.1): "Powerline Telecommunications (PLT); Powerline communication recommendations for smart metering and home automation".

[i.17] Strassner. J.: "Knowledge Representation, Processing, and Governance in the FOCALE Autonomic Architecture", chapter 11 of Autonomic Network Management Principles, 2011, Elsevier.

[i.18] Void.

[i.19] ETSI TR 102 748 (V1.1.1) "Electromagnetic compatibility and Radio spectrum Matters (ERM); Impact of the trend towards flexibility in spectrum usage on the principles for drafting Harmonized Standards and the ETSI work programme for Harmonized Standards".

[i.20] ETSI GR ENI 016 (V2.1.1) "Experiential Networked Intelligence (ENI); Functional Concepts for Modular System Operation".

[i.21] ETSI GS ENI 019 (V3.1.1): "Experiential Networked Intelligence (ENI); Representing, Inferring, and Proving Knowledge in ENI".

[i.22] ETSI GS ENI 005 (V3.1.1): "Experiential Networked Intelligence (ENI); System Architecture".

[i.23] ETSI GR ENI 018 (V2.1.1): "Experiential Networked Interlligence (ENI); Introduction to Artificial Intelligence Mechanisms for Modular Systems".

[i.24] ETSI GR ENI 003 (V1.1.1): "Experiential Networked Intelligence (ENI); Context-Aware Policy Management Gap Analysis".

[i.25] ETSI GS ENI 006 (V2.1.1): "Experiential Networked Intelligence (ENI); Proof of Concepts Framework".

3 Definition of terms, symbols and abbreviations

3.1 Terms

0 to 9

Void.

A

abstraction: hiding of unnecessary details to focus on data and information that is relevant for defining a particular concept or process

NOTE: ETSI GR ENI 003 [i.24] defined abstraction as the "process of focusing on the important characteristics and behaviour of a concept, and realizing this as a set of one or more elements in an information or data model". The above definition is introduced to emphasize the importance of hiding (not deleting) unnecessary details, and more importantly, removing the constraint of use for a model. Abstraction is fundamentally a mental process that may take the form of model elements, but does not have to.

action: set of operations that may be performed on a set of managed entities, it represents a transformation or processing in the system being modelled

NOTE: An Action either maintains the state, or transitions to a new state, of the targeted managed entities. The execution of an Action may be influenced by applicable attributes and metadata. As defined in MEF PDO CfC [i.2].

actor: role, played by an external entity (human or machine), which interacts with the subject of a use case

NOTE: An actor is always a type of stakeholder (but not vice versa). See stakeholder.

agent: computational process that implements the autonomous, communicating functionality of an application:

- **software agent:** software that acts on behalf of a user or another program

- **software autonomous agent:** software agent that acts on behalf of the entity that owns it without any communication from the owning entity
- **software intelligent agent:** software agent that reasons about its environment and take the best set of actions to satisfy a set of goals

NOTE: This has the connotation of containing AI mechanisms to provide the reasoning and decision-making capabilities.

- **software multi-agent:** set of software agents that are physically separate that work together to satisfy a set of goals

anomaly: measurable consequences of an unexpected change in state of a datum, or set of data, which is outside of its local or global norm

API: set of communication protocols, code and tools that enable one set of software components to interact with either a human or a different set of software components

NOTE: This is also known as an Application Programming Interface.

API Broker: software entity that mediates between two systems with different APIs, enabling the two different systems to communicate transparently with each other

architecture: set of rules and methods that describe the functionality, organization, and implementation of a system:

- **cognitive architecture:** system that learns, reasons, and makes decisions in a manner resembling that of a human mind

NOTE 1: Specifically, the learning, reasoning, and decision-making is performed using software that makes hypotheses and proves or disproves them using non-imperative mechanisms that typically involve constructing new knowledge dynamically during the decision-making process.

- **deliberative architecture:** symbolic world model that enables problem-solving components to be built using a sense-plan-act paradigm
- **functional architecture:** model of the architecture that defines the major functions of each module and how each module interacts with each other
- **hybrid architecture:** system made up of reactive and deliberative components that are combined into a hierarchy of interacting layers, where each layer reasons at a different level of abstraction
- **reactive architecture:** system that is aware of changes that affect its computations and adjusts accordingly

NOTE 2: The adjustment is made by reacting to an event in real-time without centralized control. The availability of new information drives program logic execution.

- **software architecture:** high-level structure and organization of a software-based system. This includes the objects, their properties and methods and relationships between objects.

Artificial Intelligence (AI): computerized system that uses cognition to understand information and solve problems

NOTE 1: ISO/IEC 2382-28 [i.7] defines AI as "an interdisciplinary field, usually regarded as a branch of computer science, dealing with models and systems for the performance of functions generally associated with human intelligence, such as reasoning and learning".

NOTE 2: In computer science AI research is defined as the study of "intelligent agents": any device that perceives its environment and takes actions to achieve its goals.

NOTE 3: This includes pattern recognition and the application of machine learning and related techniques.

NOTE 4: Artificial Intelligence is the whole idea and concepts of machines being able to carry out tasks in a way that mimics the human intelligence and would be considered "smart".

assisted system: system that the ENI system is providing recommendations and/or management commands to is referred to as the "assisted system"

attention: part of a neural architecture that dynamically computes a weighted distribution on input text, assigning higher values to more relevant elements

NOTE: Attention mimics cognitive attention as performed in the human brain. It selectively enhances some parts of the input data while diminishing other parts. Instead of encoding the input sequence into a single fixed context vector, the attention model develops a context vector that is filtered specifically for each output time step. The model then predicts next word based on context vectors associated with these source positions and all the previous generated target words.

autonomous network: set of self-governing programmable and explainable systems that seamlessly deliver secure, context-aware, business-driven services that are created and maintained using model-driven engineering and administered by using policies

Autonomous Network Responsibility Index (ANRI): level of responsibility delegated to the AN in all the Operational Procedures bind to the lifecycle management of each Autonomous Domain and E2E Service

axiom: statement that is assumed to be true, in order to serve as a starting point for further reasoning

B

behaviour: way in which a set of objects function

NOTE: This includes how the object reacts in a particular situation given one or more events.

bias: systematic difference in treatment of certain objects, ideas, or people in comparison to others:

- **algorithmic bias:** algorithm that possesses systematic and repeatable errors that create unfair outcomes
- **emergent bias:** reliance on an algorithm that has not been adjusted to evaluate new forms of data
- **inductive bias:** set of assumptions that are used in a machine learning algorithm used to predict outputs for inputs that it has not encountered

bidirectional encoder representations from transformers: unsupervised deep learning strategy that utilizes bidirectional models that considers all words of the input sentence simultaneously and then uses an attention mechanism to develop a contextual meaning of the words

blackboard: architecture that uses a shared workspace that a set of independent agents contribute to, which contains input data along with partial, alternative and completed solutions

BSS-like functionality: functionality supporting customer-facing activities for the operator

NOTE: Examples include customer service, rating, order management, billing and settlement.

C

camelCase: a naming convention in which the first letter of each word in a compound word is capitalized, except for the first word

NOTE: This is also called lowerCamelCase.

capability: type of metadata that represents a set of features that are available to be used from a managed entity

NOTE: These features may, but do not have to, be used. These features may represent all or a subset of the functionality provided by a managed entity. Since a Functional Block is a type of managed entity, Capabilities can be defined for Functional Blocks as well. A Capability provides information about the functionality of a managed entity that enables management entities to decide whether that managed entity is useful for a given task.

case-based reasoning: use of existing experiences and knowledge to understand and solve new problems

catastrophic forgetting: tendency of an artificial neural network to forget previously learned information when learning new information

choreography: set of processes that define how entities interact from a global point-of-view

NOTE: That is without a single point of control. Compare this definition to Orchestration.

class: template for defining a specific type of object that exhibits a common set of characteristics and behaviour:

- **abstract class:** class that cannot be directly instantiated
- **concrete class:** class that can be directly instantiated

classifier: procedure that predicts which elements of a set belong to which (pre-defined) classes:

NOTE: The classification is done using training data whose category membership is known, and can be thought of as a function that assigns a new observation a class label.

- **binary classifier:** classifier that decides whether or not an input belongs to one of two groups (i.e. classes) based on a classification function
- **discriminative classifier:** classifier that differentiates an object using class labels

NOTE: This directly estimates the conditional probability of P(Y|X). An example is logistic regression.

- **generative classifier:** classifier that learns a model of the joint probability of an input x and the label y, uses Bayes rules to calculate p(Y|X), and then assigns the most likely label

NOTE: This estimates P(Y|X) by estimating P(Y) and P(X|Y). An example is Naïve Bayes classifier.

- **hierarchical classifier:** classifier that maps input data into a tree-like set of output categories by first, classifying at a low level, and then iterating each lower-level classification into a higher-level classification
- **linear classifier:** classifier that assigns a label based on a linear combination of its features
- **probabilistic classifier:** classifier that assigns a label to an object based on a (conditional) probability distribution

closed loop control: self-regulating mechanism in which outputs of a system are provided to a system that compares the current state to a desired state (or set of states); the comparison is then used to adjust the behaviour of the system

NOTE 1: Positive feedback increases the correction value, while negative feedback reduces the correction value.

NOTE 2: Positive and negative feedback can be combined to achieve the needs of a system. In addition, more complex forms of closed loop control exist, such as Proportional-Integral-Derivative (PID) control. See control theory.

clustering: grouping of a set of objects such that objects in the same group are more similar to each other, by one or more measures, than to other objects in other groups

cognition: process of understanding data and information and producing new data, information and knowledge:

- **cognition model:** computer model of how cognitive processes, such as comprehension, action and prediction are performed and influence decisions

collaborating: two or more managed entities cooperate to accomplish a given task

column-oriented database: database that organizes data by field

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

compiler: computer program that translates the content of a source programming language into a different, or target, programming language

component: part of a System that has operational and/or management significance

NOTE: A Software Component is an encapsulation of a set of related functions and/or data that perform a set of specific purposes and have a set of associated semantics and behaviour.

compute node: object that performs a set of calculations according to a set of algorithms

concept drift: not taking changing data and its meanings into account when training an ML model [i.12]

condition: set of attributes, features and/or values that are to be compared with a set of known attributes, features, and/or values in order to determine what decision to make

container: object that stores collections of other objects in an organized manner

context: collection of measured and inferred knowledge that describe the environment in which an entity exists or has existed

control loop: mechanism that senses the performance of an object or process being controlled to achieve desired behaviour:

- **adaptive closed control loop:** closed control loop whose controlling function adapts to the object or process being controlled using parameter that are either unknown and/or vary over time
- **closed control loop:** control loop whose controlling action is dependent on feedback from the object or process being controlled

NOTE 1: This type of control loop measures the difference between the actual and desired values of a set of variables to adjust a set of parameters to change the behaviour of the system to bring the actual value closer to that of the desired value.

- **cognitive closed control loop:** closed control loop that selects data and behaviours to monitor that can help assess the status of achieving a set of goals, and produce new data, information, and knowledge to facilitate the attainment of those goals
- **distributed closed control loop:** closed control loop whose components are physically distributed among different locations
- **federated closed control loop:** set of semi-autonomous closed control loops that use formal agreements to govern their interaction and behaviour
- **hierarchical closed control loop:** closed control loop that is organized in the form of a tree
- **open control loop:** control loop whose controlling action is independent of the output of the object or process being controlled

NOTE 2: This type of control loop does not link the control action to the object or process being controlled (it simply continues to apply the control action).

- **peer closed control loop:** two or more closed control loops that may interact, but are independent of each other

control plane: communication between entities that enables forwarding and routing of traffic to work

NOTE: Control plane packets are destined to or locally originated, by entities themselves (e.g. they go to a network entity and direct how traffic flows). Compare to Data Plane.

control theory: application of mechanisms to regulate the behaviour of a target system

NOTE: Control theory includes linear and nonlinear control mechanisms.

coupling: amount of interdependence between two components, modules or systems

D

data: facts and statistics collected together for reference or analysis:

- **database:** structured set of data held in a computer, especially one that is accessible in various ways
- **data lake:** centralized storage repository that stores raw data that are in the form of structured, semi-structured and unstructured format
- **data mart:** subset of a data warehouse focused on a particular line of business, department or subject area
- **data mining:** procedure that discovers patterns in, and extracts knowledge from, data sets

NOTE 1: For the purposes of ENI, these patterns are of two principal types:

1) patterns that cause the generation of data; and

2) patterns that relate data (typically in a semantic manner).

- **data model:** representation of concepts of interest to an environment that is dependent on data repository, data definition language, query language, implementation language and/or protocol (typically, but not necessarily, all five)

NOTE 2: As defined in MEF PDO CfC (MEF 95 [i.2]).

- **data plane:** path that the end-user traffic takes through a network

NOTE 3: It is made up of traffic that goes through network entities, not to a network entity. Compare to Control Plane.

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

NOTE 4: A data warehouse is designed for query and analysis as opposed to transaction processing. It analyses and reports on data from operational systems as used in decision-support systems.

decidable: procedure that determines, by a mathematical formal means in a finite amount of time, whether a formula is valid

decision making: set of processes that result in the selection of a set of actions to take from among several alternative possible actions

declarative policy: type of policy that uses statements to express the goals of the policy, but not how to accomplish those goals

NOTE 1: State is not explicitly manipulated, and the order of statements that make up the policy is irrelevant.

NOTE 2: In the present document, Declarative Policy will refer to policies that execute as theories of a formal logic.

NOTE 3: As defined in MEF PDO CfC (MEF 95 [i.2]).

denormalisation: process of changing information from a canonical form to one specialized for a particular actor and/or domain

design pattern: general, reusable solution in a given context to a commonly occurring software problem:

NOTE: This type of design pattern is not an architecture and not even a finished design; rather, it describes how to build the elements of a solution that commonly occurs. It may be thought of as a reusable template.

- **design pattern, architecture:** general, reusable solution in a given context to a commonly occurring problem in the design of the software architecture of a system
- **design pattern, software:** general, reusable solution in a given context to a commonly occurring problem in the design of a software system

designated entity: operator, nms, ems, controller or orchestrator acting on behalf of the assisted system

NOTE: The Designated Entity is a trusted entity.

digital twin: virtual representation of a physical object or system across its lifecycle, using real-time data to enable understanding, learning and reasoning

domain: collection of entities that share a common purpose:

NOTE 1: Each constituent Entity in a Domain is both uniquely addressable and uniquely identifiable within that Domain. This is based on the definition of an MCMDomain in MEF 78.1 [i.5].

- **administrative domain:** domain that employs a set of common administrative processes to manage the behaviour of its constituent Entities
- **management domain:** domain that uses a set of common Policies to govern its constituent Entities

NOTE 2: A Management Domain refines the notion of a Domain by adding three important behavioural features:

1) it defines a set of administrators that govern the set of Entities that it contains;
2) it defines a set of applications that are responsible for different governance operations, such as monitoring, configuration, and so forth;
3) it defines a common set of management mechanisms, such as policy rules, that are used to govern the behaviour of MCMManagedEntities contained in the MCMManagementDomain.

This is based on the definition of an MCMDomain in MEF 78.1 [i.5].

domain technical expert: technical expert that has authority within a domain

E

ENI application programming interface: set of communication mechanisms applied between two or more software components

NOTE: It consists of tools, object methods, and other elements of a model and/or code. APIs simplify producing programs, since they abstract the underlying implementation and only expose objects and flow of information, and the characteristics and behaviour of those objects. This prevents the unnecessary exposure of objects.

ENI external reference point: reference point that is used to communicate between an ENI Functional Block and an external functional block (e.g. a functional block of the OSS, BSS or assisted system)

NOTE: Where an ENI External Reference Point crosses between two organizational entities is not specified in this release.

ENI framework: set of abstractions that provide reusable and extensible mechanisms to provide generic functionality

NOTE 1: The ISO/IEC/IEEE 42010 [i.8] defines the term **architecture framework** as: "An architecture framework establishes a common practice for creating, interpreting, analysing and using architecture descriptions within a particular domain of application or stakeholder community".

NOTE 2: The ENI Framework also uses its abstractions to enable the ENI System to dynamically adapt to changing business goals, user needs, and environmental conditions. The ENI Framework hence provides a standard way to build and deploy applications and application components.

ENI hardware interface: point across which electrical, mechanical, and/or optical signals are conveyed from a sender to one or more receivers using one or more protocols

NOTE: A hardware interface decouples the hardware from other functional blocks in a system.

ENI interface: point across which two or more components exchange information:

- **ENI API interface:** ENI interface set of communication mechanisms through which a developer constructs a computer program