ETSI White Paper No.51

ENI Vision:
Understanding the Operator Experience Using Cognitive Management

First edition – January 2023

ISBN No. 979108262072

Author:

Dr. John C. Strassner (Futurewei Technologies)

ETSI
06921 Sophia Antipolis CEDEX, France
Tel +33 4 92 94 42 00
info@etsi.org
www.etsi.org

Editing Team:

Dr. Raymond Forbes	Huawei Technologies (UK) Co. Ltd
Ms. Haining Wang	Intel Corporation (UK) Ltd
Miss. Ziting Zhang	China Telecommunications Corporation
Ms. Chao Wu	NTT Corporation
Mr. Antonio Gamelas	PT Portugal SGPS SA

Contents

Executive Summary

This White Paper describes the design of a novel cognitive network being done in ETSI ISG ENI. Its novelty lies in the use of a model-driven engineering process, coupled with multiple closed control loops, to implement a formal cognition model to drive learning and decision-making. This is used to enhance the overall operator experience.

This White Paper first gives a brief introduction to the purpose of ETSI ISG ENI, along with its benefits and status. It then describes in detail the cognitive management being designed. Cognition is the process of acquiring and understanding data and information in order to produce new data, information, and knowledge. Context awareness keeps track of and catalogues a collection of measured and inferred knowledge that describes the environment in which a managed entity exists or has existed. This, coupled with historical information and newly ingested data, is used by the situation awareness module to gather the relevant data and behaviour that pertain to contextual circumstances and/or conditions of a system or process in order to understand the meaning and significance of these data and behaviours with respect to the set of goals that the system is trying to achieve. The cognition model infers how processes, actions, and new situations are likely to evolve in the near future, and then selects a set of actions to move the system state to the best possible state for achieving its goals. This results in an adaptive management system that seeks to meet its goals as a function of context. This provides more informed information to the operator as well as intelligently managing the behaviour of the system.

The remainder of this White Paper describes some of the relevant technical details of the ENI System as well as five innovative Proof of Concepts.

PART A: Overview

A.1 Introduction

A.1.1 Objectives

The objectives of this White Paper are to provide:

- an exploration of the ENI cognition model,
- an examination of multiple facets of the cognition process, including context awareness, situation awareness, and the formal cognition process (perception, comprehension, and action),
- a description of how elements of human cognition can inspire ENI cognition,
- an explanation of how cognitive management can be used to enhance the operator experience,
- show how cognitive management can enable automation and autonomic behaviour.

A.1.2 Scope of this White Paper

This White Paper describes how the novel system architecture of the Experiential Networked Intelligence Industry Specification Group (ENI ISG) intelligently manages, predicts, adjusts, and optimizes network behaviour using cognitive management.

Cognition is the process of acquiring and understanding data and information in order to produce new, data, information, and knowledge. A cognitive system uses cognitive processes to understand how past behaviour, coupled with currently ingested contextual data and information, affect the goals that the ENI System is trying to achieve. The ENI Cognitive Management system draws from human decision-making processes to better comprehend the relevance and meaning of ingested data. Cognitive management enables the ENI System to experientially learn to improve its operation and performance. It also describes how cognitive management enables automation and provides autonomic behaviour.

A.2 ISG ENI Work Programme Summary

A.2.1 Purpose of ISG ENI

As technology becomes increasingly complex, managing telecommunication infrastructure grows in complexity. Operators need data-driven, context-aware tools to help them make the right decisions at the right time. This enables actionable decisions to facilitate prompt adaptation to various business and operational needs.

The purpose of the 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 to recognize and incorporate new and changed knowledge, and hence, make actionable decisions more quickly.

The main objectives of ISG ENI are listed as:

- To develop standards for a Cognitive Network Management system, incorporating a set of closed control loops. The closed control loops are based on extensions to the “observe-orient-decide-act” model (e.g., the incorporation of situation awareness, learning, and reasoning capabilities).

- To adapt the usage of available network resources and services of the managed system using Cognitive Management. This enables the real-time evolution of user needs, environmental conditions, and business goals to determine which services should be offered during a given context.
- To create specifications of a modular and extensible ENI system that can be implemented in a phased approach to accelerate ICT's adoption for emerging systems/networks and technologies.
- To define how cognitive management can use different types of policies (e.g., imperative, declarative, and intent) can be used to specify adaptive behavioural changes.
- To develop standards that define a model-driven architecture [4], described by functional blocks and appropriate Reference Points and Interfaces, that supports adaptive and evidence-driven service operation using cognitive management to provide the required Operator Experience.
- To quantify the Operator Experience by introducing metrics and an associated evaluation procedure.
- To provide a telemetry processing framework that uses context and situation awareness to learn and reason about which data should be collected using what types of processing mechanisms to support information collection and measurement about network performance, network resources and services.
- To provide external Reference Points, along with APIs and DSLs, to be used for ENI to interact in an operator-controlled manner with existing and emerging ICT systems. This promotes interoperability between ENI and Assisted Systems as well as migration towards AI-assisted ICT services.

A.2.2 Benefits

The ENI System is based on an experiential architecture. This means that it learns through experience (i.e., the operation of the systems that it assists in governing, as well as direction from administrators). This self-learning principle improves operator experience and enables the system to adjust the offering of services in response to contextual changes. The ENI System provides the following important benefits:

- measures and quantifies the operation and performance of an operator's resources and services,
- enables personalized services to be provided to customers,
- automates 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,
- learns from the operation of the network, and decisions made by the operator, to provide more efficient management of the system and increased agility to achieve its goals.

A.2.3 Status

The ETSI ENI ISG was created in February 2017. Members today come from operators, vendors, and research institutes all over the world. The ETSI ISG ENI is open to ETSI members and non-ETSI members alike. Organisations are welcome to join the ISG effort, contribute to the development of these key specifications and demonstrate Proofs of Concepts. To join, please contact: isgsupport@etsi.org.

Release 3 is currently working on 11 Work Items, which are briefly summarised as follows:

GS ENI-001v3.2.1 – Use Cases. Specifies additional use cases and scenarios that are enabled with enhanced experience through the use of network intelligence. It describes how ENI can be applied as solutions of the identified use cases in accordance with the ENI Reference Architecture.

GR ENI-002v321 – Requirements. Specifies the requirements of how intelligence is applied to the network and applications in different scenarios to improve experience of service provision and network operation. Also, how intelligence enables dynamic autonomous behaviour and adaptive policy driven operation in a changing context. In Release 3 the Specification will add:

- Requirements derived from API descriptions
- Requirements derived from System Architecture
- Requirements derived from new use cases.

GR ENI-004v311 – Terminology. Provide terms and definitions used within the scope of the ISG ENI, in order to achieve a "common language" across all the ISG ENI documentation. This work item will be updated as the ENI specifications develop.

GS ENI-005v311 – System Architecture. Continues the development of GS ENI 005 v2.1.1 to add:

- define and specify APIs, Interfaces, and protocols used by ENI based on information and data models
- specify the ENI cognition model in detail
- enhance the description and specification of the ingestion, normalisation, and output generation of data, information, and policies (imperative, declarative, and intent) in greater detail
- enhance the description and specification of the control loops used in ENI
- enhance the description and specification of policy management used in ENI
- enhance the description and specification of architectural principles for interacting with other groups within and outside ETSI.

GR ENI-009v121 – Data Mechanisms. The document revises ETSI GR ENI 009 to describe data operation requirements and mechanisms to better serve ENI system:

- data format among the Functional Block of ENI system and towards the external world (internal Functional Blocks, Knowledge Representation)
- data Conversion and possibility to translate AI data model to be adapted to / from external system (external trained model imported into ENI)
- consistency of data format and interface to accelerate the Autonomous Network evolution process
- ensure that customer privacy is not disclosed in the entire lifecycle of data collection, processing, and utilization (Federated Learning).

GR ENI-010v121 – Evaluation of categories for AI application to Networks. Revises GR ENI 010 to:

- investigate quantitative evaluation criteria of network autonomicity categories
- perform deeper research of more quantitative factors that determine those categories

- define a set of accurate scoring criteria that complies with the evolution of the ENI architecture
- define a data model covering an entire operator's network or just a specific domain.

GR ENI-013v111 – Intent Policy Model Gap Analysis. Produce a gap analysis report on intent information models based on existing SDO work, including the policy management model as specified by ETSI ISG ENI, and, if needed, will also provide a list of recommendations on general guidelines addressing the high-level policy model of each SDO’s intent policy model, and how these guidelines compare with those stated in the ETSI ENI system architecture.

GR ENI-015v111 – Processing and Management of Intent Policy. Describe procedures for processing Intent Policy. It will describe knowledge management for Intent Policy, including how to use a Knowledge Graph to manage intent policies and intent policy knowledge, as well as describe procedures for lifecycle management of intent knowledge. Finally, it will provide typical use cases and requirements that can reduce management complexity.

GS ENI-019v311 – Representing, Inferring, and Proving Knowledge in ENI. Define the ENI information model, along with at least two different examples of how to derive technology-specific data models from the ENI information model. Finally, it will explain how ontologies can be incorporated to augment, enhance, and specify meaning and different relationships between modelled entities. This document is specific to and enhances the current ENI System Architecture.

GS ENI-030v311 – Transformer Architecture for Policy Translation. Enhance the information described in GR ENI 018 and specify how a transformer architecture can be used to translate input policies to ENI Policies for use in cognitive networking and decision making in modern system design. The transformer-based architecture will be used to parse, understand, and translate text.

GR ENI-031v311 – Construction and application of fault maintenance network knowledge graphs. Describe use cases and a construction method of fault maintenance knowledge graphs. This is supported by defining data requirements, schema design, and the knowledge application interface fault maintenance knowledge graphs. This GR will encompass research and investigation activities that will address wireless networks at the first stage.

A.2.4 Membership

The following picture is a snapshot of the current membership of ISG ENI.

Figure A.1: ISG Members and Participants

A.3 ENI Cognitive Management

A.3.1 ENI Cognition Principles

The ENI Cognitive Management Functional Block is based on an innovative cognition model (see ETSI GS ENI-005 [1]). A cognition model defines how cognitive processes, such as comprehension, action, and prediction, are performed and influence decisions. The ENI cognition model draws heavily on how human cognition is performed. More specifically, the perception portion provides the notion of classifying data into pre-defined representations that are understood and relevant to the current situation; memory is used to increase comprehension of the situation; actions are judged by how effectively they perform to support the situation.

ENI Cognitive Management learns from experience to improve its performance. This includes:

- acquiring new knowledge from instruction or experience,
- revising and correcting existing knowledge,
- combining existing data and information to infer and deduce new knowledge.

Cognitive Management enables 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 Management Functional Block then evaluates the meaning of the data and determines if any actions need to be taken to ensure that the goals and objectives of the system are met. This includes improving or optimising performance, reliability, and/or availability. These metrics are exemplary - it is possible to use other metrics to meet application-specific needs.

ENI Cognition Management is based on human cognition. Cognitive psychology defines three interacting layers, called reactive (or subconscious), deliberative, and reflective. Reactive processes shall take immediate responses based upon the reception of an appropriate external stimulus. In humans, these processes correspond to instinctual and learned behaviours. In ENI, such processes may have no understanding of the semantics of the external stimulus; rather, they respond with some combination of pre-defined and learned reactions.

Deliberative processes shall receive data from and can send recommendations and/or commands to the reactive processes. In humans, this part of the brain is responsible for the ability to achieve more complex goals by applying short- and long-term memory in order to create and carry out more elaborate plans. In ENI, these processes accumulate and generalize knowledge from experience and combine that with what is learned from other people and systems.

Reflective processes supervise the interaction between the deliberative and reactive processes. In humans, these processes enable the brain to reformulate and reframe its interpretation of the situation in a way that may lead to more creative and effective strategies. In ENI, these processes consider what predictions turned out wrong, along with what obstacles and constraints were encountered, in order to prevent sub-optimal performance from occurring again.

A.3.2 The ENI Cognitive Control Loop

A.3.2.1 Overview

Cognition Management is based on an enhanced version of the OODA control loop. Four enhancements to OODA were made. First, OODA was designed to apply to a *single* decision-maker. ENI's version is designed to accommodate *collaborative* decision-making. Second, a dedicated *planning* stage was inserted between the orient and decide cycles. This was done to accommodate situation awareness. Third, learning was inserted to monitor all phases of each control loop. Fourth, policy management is used to make each functional block in the ENI cognition loop *configurable* using a *standardized set of commands*. This is shown in Figure A.2.

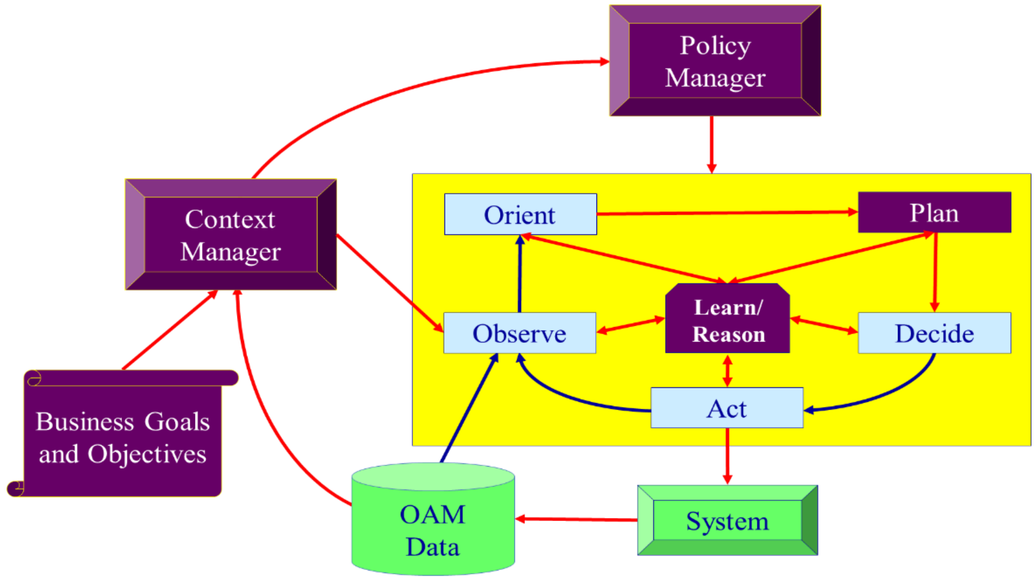

Figure A.2: Simplified ENI Cognition Closed Control Loops

A.3.2.2 Functional Operation

The "Observe" part of the OODA control loop ingests current information from the system being monitored, as well as relevant contextual, situational, and historical information (this is represented in Figure A.1 by the Context Manager-to-Observe connection). There are two changes from OODA. First, the OODA loop assumed that different data and information was normalized and combined. This is performed by the Data Ingestion and Data Normalisation Functional Blocks (see clause B.1.5). Second, the Policy Manager ensures that the correct data and information is gathered and adjusts monitoring points and telemetry settings appropriately.

The "Orient" part of the OODA control loop analyses, evaluates, and prioritizes information. This consists of semantic annotation, knowledge inferencing, and situational processing. In each case, the Knowledge Management, Context-Aware Management, Cognition Management, and Situational Awareness Functional Blocks may perform all or part of these tasks (see clause B1.5). The purpose of this part of the control loop is to ensure that input data and information is adapted to the current context and situation. Semantic Annotation is a set of processes that examine the input data, and annotates it based on the current context and situation to enable it to be better understood by subsequent Functional Blocks. Inferencing discovers more about the nature and meaning of the data. Situational processing analyses the information to determine if anything has just occurred that threatens its system goals. If problems are identified, then situational processing decides what is likely to happen, and how that affects the goals that the system is trying to achieve. This produces a set of possible alternative actions.

The "Plan" portion consists of deliberative and reactive processing. The former compares the input stimuli to short-term events, and their related responses. Planning is then initiated to consider different types of responses and their effect on the current needs of the system being managed. The latter compares the input stimuli to short- and long-term events, and their related responses. Planning is then initiated to consider different types of responses and their effect on the goals (both short- and long-term) of the system being managed. In both cases, the responses are then ranked according to how well each satisfies the goals of the system being managed. The action that ranks the best is then executed. The set of possible responses, along with their stimuli, are recorded for further analysis.

The "Decide" portion consists of two Functional Blocks: Model-Driven Engineering (MDE) and Policy Management (see clause B1.5). The purpose of this part of the cognitive control loop is to decide whether any action should be taken to preserve the goals of the system. The MDE Functional Block translates the set of corrective actions to take to an internal representation that enables ENI Policies to be created by the Policy Management Functional Block to capture those actions. These are then translated to a form usable by the system being managed by the Denormalization and Output Generation Functional Blocks and are sent to the system being managed by the API Broker (see B.1.5).

Learning and reasoning functions will then compare this and other actions to see if the collected set of actions were the optimal responses that could be taken. Conceptually, reflective processing is a type of meta-management, where the current needs of the system being managed are weighed against the goals that are to be achieved.

A.3.3 ENI Cognition Management Functional Block

The Cognition Management Functional Block is a set of functions, organized as lower-level Functional Blocks, that enables the ENI System to understand ingested data and information, as well as the context and situation that defines how those data and information were produced.

The Cognition Management Functional Block mimics some of the processes involved in human decision-making to better comprehend the relevance and meaning of ingested data. Cognitive Management includes reflexive and habitual behaviour that respond to long-term intentions similar to how reflexive actions are performed by the human brain. Cognitive capabilities include functions that reflect processing as done in the human brain, such as perception, reasoning, learning, and planning. Critically, a system that uses cognition shall be able to explain why it acted a certain way in response to stimuli, and more importantly, can learn whether that action was incorrect and, if correct, whether it was optimal.

The Cognition Management Functional Block uses existing knowledge to validate and generate new knowledge. This means that new knowledge may be added, and in some cases, existing knowledge may be changed. Hence, the ENI System uses a dynamically changing set of repositories (as opposed to other management systems, which typically use repositories that use fixed content). A cognition framework uses multiple diverse processes and technologies, including linguistics, computer science, AI, formal logic, neuroscience, psychology, and philosophy, along with others, to analyse existing knowledge and synthesize new knowledge.

A.3.4 ENI Cognition Process

A.3.4.1 Overview

An ENI System is based on how humans think. A cognitive system is a system that can reason about what actions to take, even if a situation that it encounters has not been anticipated or seen before. A cognitive system can learn from its experience to improve its performance. It can also examine its own capabilities and prioritize the use of its services and resources. In addition, it is able to explain what actions it took and accept external commands to perform necessary actions.

Cognition is the process that consists of three processes: acquiring data (perception), understanding information (comprehension), and producing new data, information, and knowledge that supports taking actions. More specifically, the perception portion classifies data into pre-defined representations that are understood and relevant to the current situation. Short- and long-term memory are used to increase comprehension of the situation. Finally, the knowledge obtained produces actions that are judged by how effectively they perform. These concepts are supported in cognitive psychology, modelled using three interacting layers, called reactive (or subconscious), deliberative, and reflective.

Reactive processes take immediate responses based upon the reception of an appropriate external stimulus. Such processes have no sense for what external events "mean"; rather, they simply respond with some combination of instinctual and learned reactions.

Deliberative processes receive data from and can send "commands" to the reactive processes; however, they do not interact directly with the external world. This part of the brain is responsible for our ability to achieve more complex goals by applying short- and long-term memory in order to create and carry out more elaborate plans. This knowledge is accumulated and generalized from personal experience and what we learn from others. Note that memory capacity must be limited. Large messages cannot be easily understood. The same argument shows that consciousness must be serial. Messages arriving in parallel are not likely to be understood.

Reflective processes supervise the interaction between the deliberative and reactive processes. These processes enable the brain to reformulate and reframe its interpretation of the situation in a way that may lead to more creative and effective strategies. It considers what predictions turned out wrong, along with what obstacles and constraints were encountered, in order to prevent sub-optimal performance from occurring again. This also includes self-reflection, which analyses how well the actions that were taken solved the problem at hand.

A.3.4.2 Situational Awareness

The purpose of the Situation Awareness Functional Block is to enable the ENI System to be aware of events and behaviour that are relevant to a set of entities in the system being assisted and/or governed), and how those events and behaviour affect the achievement of the current set of goals that are being worked on by the ENI System. This includes the ability to understand how recommendations and commands given by the ENI System impact the current set of goals, both immediately and in the near future. Situation awareness is especially important in environments where the information flow is high, and poor decisions have the possibility to lead to serious consequences (e.g., violation of SLAs).

A cognitive system is a system capable of independently developing strategies for and solving human tasks. A cognitive system is both context- and situation-aware. A cognitive system may draw on multiple sources of information, including both structured and unstructured digital information, as well as sensory inputs (visual, gestural, auditory, or sensor-provided).

A cognitive system should be able to adapt its governance in accordance with changing information and context. A cognitive system should also be able to adapt its functionality as its goals and requirements evolve. Cognitive systems may use deterministic mechanisms to make decisions; however, cognitive systems should mainly use probabilistic processes for decision-making.

The individual Functional Blocks of a cognitive system, as well as multiple cognitive systems, are to be able to collaborate on a set of tasks. The specific set of Functional Blocks assigned to the collective is driven by their suitability to accomplish the tasks of the current set of goals. The ENI System uses registered characteristics of each Functional Block that describe their capabilities, along with applicable metadata, to make the selection. Once the set of tasks has been completed, the collective may disband.

An individual system will be capable of taking over mission-critical functions on its own in case coordination or communication with other collaborating entities is not functioning. The collective should furthermore optimally and efficiently complete its tasks in such a situation.

A.3.4.3 Architectural Implications

The human nervous system is a distributed parallel system with many different specialized processors that are used in the reactive, deliberative, and reflective processes. This highly collaborative architecture inspires the ENI system architecture. ENI Functional Blocks define modular specialized functions that are guided by the Cognitive Management Functional Block. A semantic bus, which is able to route messages based on their *meaning*, distributes messages and information to the ENI Functional Blocks on a situationally aware basis.

Cognitive systems should generate hypotheses, reasoned arguments, and recommendations. Cognitive systems should be able to generate explanations of their reasoning processes. A cognitive system will be able to reason about what actions to take, even if a situation that it encounters has not been anticipated. It will learn from its experience to improve its performance. It should be able to examine its own capabilities and prioritize the use of its services and resources, and if necessary, explain what it did and accept external commands to perform necessary actions.

Each Functional Block is able to recruit other Functional Blocks to accomplish tasks. Lower priority tasks are typically completed after all higher priority tasks unless dependencies exist between them. This approach enables teams of Functional Blocks to take on goals that individual Functional Blocks cannot achieve themselves.

A.3.5 Enabling Automation and Autonomic Behaviour

A.3.5.1 Terminology

This White Paper differentiates between the terms automatic, autonomous, and autonomic.

Automatic refers to pre-programmed task execution. Typically, this is a reflexive action, in that a set of alternatives as well as effects are not considered before the action is taken. In addition, if there are no existing rules or mechanisms to program an automatic action, then it cannot occur.

Autonomous is the ability to make an informed, independent decision. There are two very different definitions for this term. The first focusses on the notion of self-governance. The second expands this to include self-management. In this white paper, we will use the first definition. This is primarily due to the disadvantages of *cognitive offloading*, which also offloads learning and judgment.

Autonomic also has two very different definitions. As used in biology (e.g., the autonomic nervous system), autonomic means *involuntary*. As used in technologies including autonomic computing and networking, this refers to both self-management and, more importantly, *self-awareness* (see "From Autonomic Computing to Autonomic Networking: An Architectural Perspective" 2008 [2]). This White Paper will use the second definition.

A.3.5.2 Misconception About Autonomics Solved by Cognition

Self-awareness requires cognition. It has been proven that global behavioural management is not achievable as a "whole". However, it can be achieved by breaking a global system into a set of smaller systems that are each autonomic. Each of these autonomic systems can then be broken into a set of modular functional blocks that are each dedicated to fulfilling a specific task. This is performed by the Cognition Management ENI Functional Block.

A control loop cannot determine "what good behaviour is", as adaptive systems change their behaviour in response to changing user needs, business goals, and environmental conditions. Control loops are simply the mechanism to analyse the operation of the system in a particular context according to its goals and implement any required changes. The analysis portion also leads to detecting new information and knowledge. In addition, self-awareness requires the autonomic system to also sense changes in itself. This set of analysis, sensing, learning, and commands is governed by the situational awareness ENI Functional Block.

Cognitive systems should be used to *augment* human decision-making and action processes. Cognitive systems are not meant to replace humans, but rather, enhance them. This is why the ENI System operates in two distinct modes: recommendation and command. The former enables an ENI System to function as an assistant that recommends actions to take. The latter enables an ENI System to function as a "super-orchestrator", governing other management components (e.g., orchestrators, management systems, and controllers). Autonomic Systems are indeed more intelligent, but that is not their primary benefit. For example, consider smart phones. People do not buy them because they are "more intelligent", but rather, because they have features that are important to users. Similarly, autonomic systems use their intelligence to transform a system into an Intelligent Provider of Services. The "intelligence" is the ability to adapt to change using cognition.

Finally, autonomics is not just about self-configuration, self-healing, self-optimization, and self-protection. These are all *benefits* of a cognitive system. None of these benefits would be possible without two key ingredients: *self-awareness* and *self-knowledge*. These two capabilities collectively enable the cognitive system to recognize change in itself and the environment, analyse its performance with respect to its current goals, and comprehend the effects of issuing a command in that situation.

A.3.6 Enhancing the Operator Experience using ENI Cognitive Management

Operators have to manage services provided by the network (e.g., streaming multimedia or call quality) in increasingly complex networking environments. This is exacerbated by new technologies deployed in new applications, including the Internet of Things, 5G, telehealth, Smart Cities, and communications over multiple media. Networks are now a software-driven fusion of virtual and physical networks that may have different Key Performance and Quality Indicators, and even different Service Level Agreements, under different conditions.

In the past, reactive approaches were common. When a problem was reported, support teams mobilized. But alas, service delivery was already adversely impacted. Proactive management used Key Performance Indicators (KPIs) to help predict problems. But this didn't account for changing user needs, business goals, and environmental conditions. Cognitive network management analyse current and historical data using machine learning algorithms to anticipate future issues and automate the optimization of the network. The ENI System uses cognitive networks as an *experiential* learning mechanism, which enables the network to learn from experience of its own as well as operator decisions.

Cognitive operations by themselves are not enough. For example, in a typical scenario, a technician will create a trouble ticket, access multiple systems and knowledge bases to diagnose the problem, run multiple applications to identify the root cause of the problem, and then either apply the right solution or escalate the ticket to the next tier of support. This is an inefficient, tedious, and lengthy process.

Compare this to the operation of a cognitive network management system. Cognitive systems understand the underlying *meaning* of why telemetry-based information occurred as a function of context. Cognitive systems reason by using formal logic or other mechanisms to form and prove hypotheses as to why a problem occurred and how to fix it. Cognitive systems learn experientially and become more efficient and accurate over time as their knowledge increases and is refined.

The ENI System can function as either an advisor or (when sufficient trust is established) or manager. It has the ability to use policies authored in natural language and provides trends and predictions that notify the operator of problems that will occur before they impact a service. Intelligent telemetry and dynamic knowledge bases are used to provide evidence that supports ENI's recommendations and commands.

Cognitive management can be applied to other domains besides network management. For example, a cognitive supply chain is able to predict distribution and inventory bottlenecks and adapt accordingly. Cognitive health management may suggest achievable health protocols based on the subject's habits that are ultimately more effective. Cognitive financial management may produce better results by considering user-specific factors such as age and financial risk tendencies, even if the rate of return is lower. Cognitive social networks may provide better user experience through having a clearer understanding of how people behave and what a user wants in different scenarios.

All of these examples demonstrate a common theme: a cognitive computing system is trained to learn based on interactions and outcomes. Building on this, the ENI Cognitive System assesses the current performance of the network contextually and compares that performance and operation to the goals that it is given. This increases its relevance regardless of change.

A.4 Summary and Future Work on ENI Cognitive Management

A.4.1 Summary

This White Paper has described a Cognitive Network using the ENI System Architecture. The ENI cognition model is defined in terms of a set of hierarchical closed control loops based on extensions to the Observe-Orient-Decide-Act model. These extensions include accommodate *collaborative* decision-making, inserting a dedicated *planning* stage between the orient and decide cycles, adding *learning* to monitor all phases of each control loop, and using *policy management* to standardize commands. The combination of these enhancements enables the ENI Cognitive Architecture to *adapt* its behaviour according to changes in user needs, business goals, and environmental conditions. For example, it could reconfigure a set of 5G slices to meet changing service needs.

Each ENI Cognitive control loop is based on an innovative cognition model (see ETSI GS ENI-005 [1]). A cognition model defines how cognitive processes, such as comprehension, action, and prediction, are performed and influence decisions. The ENI cognition model is based on experiential learning similar to how humans learn. This is reinforced by realizing three types of human cognition, called reactive (or subconscious), deliberative, and reflective. Reactive processes take immediate responses based on receiving an appropriate external stimulus but may have no understanding of why they are taking the action. Deliberative processes accumulate and generalize knowledge from experience and instruct the reactive processes which actions to take. Reflective processes supervise the interaction between the deliberative and reactive processes. These processes consider what predictions turned out wrong, along with what obstacles and constraints were encountered, in order to prevent sub-optimal performance from occurring.

Cognitive Management enables the ENI System to understand ingested data and information, as well as the context that defines how those data were produced. This enables the meaning of the data to be evaluated and prioritized and determines if any actions need to be taken to ensure that the goals and objectives of the system are met. This includes improving or optimising performance, reliability, and/or availability. Each ENI System Functional Block is able to recruit other Functional Blocks to accomplish tasks in a prioritized manner. This approach enables teams of Functional Blocks to take on goals that individual Functional Blocks cannot achieve themselves.

The ENI Cognitive System uses hypotheses and reasoning to make decisions. This approach enables explanations to be generated about decisions taken. However, it is important to note that Cognitive systems should be used to *augment* human decision-making and action processes, as opposed to *replacing* humans. This is why the ENI System operates in two distinct modes: recommendation and command. The former enables an ENI System to function as an assistant that recommends actions to take. The latter enables an ENI System to function as a "super-orchestrator", governing other management components (e.g., orchestrators, management systems, and controllers).

A Cognitive system is an important step beyond automation towards realizing Autonomic Systems. In particular, ENI's novel cognition model defines functionality beyond autonomous systems by actively optimising resources and services to meet changing system goals. The use of policy management enables different constituencies to express their goals and intents, and ENI to respond, in a standardized manner.

A.4.2 Future Work

Future work will include specification of APIs and protocols, policies, and additional PoCs.

The ENI System uses its information model, in conjunction with derived data models, to define APIs. This model-driven approach provides a clear representation of the capabilities of a system with a definition that is structured, well defined and computer friendly. Model-driven architecture (see Model Driven Architecture (MDA) - MDA Guide rev. 2.0 [4]) provides an abstract representation of software engineering methods, using models as the main artifacts. This approach enables UML (see "OMG Meta Object Facility (MOF) Core Specification, Version 2.5.1 [3]) models to be customized using profiles, stereotypes, and tagged values.

The ENI information model (see ETSI GS ENI-005 [1]) provides an abstraction layer that unifies the different concepts and terminology that different actors use under a common and consensual set of definitions. Software patterns (see "Design Patterns: Elements of Reusable Object-Oriented Software", 1994 [5]) allows developers to build modular and reusable libraries of extensible objects. The ENI information model includes Service, Resource, Policy, and Metadata object hierarchies.

ENI defines a novel information model (see ETSI GS ENI-019 [6]) that enables different types of policies to be represented. This is made possible by a set of unique abstractions realized in the policy model. The content of any policy will be made up of one or more policy statements; a policy statement may be made up of one or more policy clauses; the model semantics enable the type of policy to determine the allowed set of policy components that it can contain. Currently imperative (e.g., when an event occurs, if conditions are satisfied, then actions may be executed), declarative (e.g., logic-based policies), and intent (e.g., restricted natural language) policies are defined.

Additional PoCs will continue to be defined to demonstrate key behaviours and applications of ENI.

PART B: ISG ENI Technical Details

B.1 System Architecture

B.1.1 Introduction

The ENI System is an innovative, policy-based, model-driven functional architecture that improves operator experience. The ENI System targets network automation and assists in decision-making of humans as well as machines. It takes context and goals into account to meet the needs of the business more efficiently. For example, the ENI System enables the network to change its behaviour (e.g., determine the set of services offered) in accordance with changes in context. This includes business goals, environmental conditions, and the varying needs of end users.

This is achieved by using policy-driven closed control loops that use emerging technologies, such as big data analysis, analytics, and artificial intelligence mechanisms, to adjust the configuration and monitoring of networks and networked applications. It dynamically updates its acquired knowledge to understand the environment, including the needs of end users and the goals of the operator, by learning from actions taken under its direction as well as those from other machines and humans (i.e., it is an experiential architecture). It also ensures that automated decisions taken by the ENI System are correct and increase the reliability and stability, and lower the maintenance required, of the network and the applications that it supports. It improves and simplifies the management of network services through their visualization and enables the discovery of otherwise hidden trends and interdependencies.

Finally, it helps determine which services are appropriate to be offered for a specific context, and which services are in danger of not meeting their Service-Level Agreement (SLA), as a function of changing context. In order to achieve the latter and to assist in monitoring and improving infrastructure efficiencies, it draws knowledge from telemetry shared with it.

B.1.2 A Personalized Assistant

An ENI System operates in two different modes, called "recommendation mode" and "management mode". The first mode is analogous to that of a Recommender system - it provides recommendations to the Assisted System. In contrast, management mode authorizes the ENI System to provide decisions and commands to be implemented by the Assisted System. These decisions and commands are subject to the approval of the Assisted System.

The ENI System also supports a mixed mode of operation (i.e., the combination of recommendation and management modes) for different sets of decisions. For example, if the Assisted System decides that recommendations belonging to a particular category have been verified, it may change decisions for that category to operate in management mode. Each such decision is independent of other categories.

B.1.3 Design Principles

An ENI System is defined as a set of Functional Blocks. Each Functional Block is described in terms of its inputs, outputs, state, and optionally, transfer function. This specifically means that this architectural framework does not define a specific implementation. The following are generic design principles that are followed in an ENI System Architecture.

1) The internal implementation of a Functional Block is not defined.

2) Functional Blocks may be nested, where the nested Functional Blocks provide greater detail for the containing Functional Block.
3) Management, control, interaction with the Assisted System and orchestration of the ENI System will be defined in terms of Reference Points that may use APIs and/or DSLs.
4) The ENI System may provide management, control, and/or orchestration commands and recommendations to the Assisted System.
5) Architectures from other SDOs will be interfaced by using a subset of dedicated Reference Points for that purpose.
6) An ENI System will use an API Broker to insulate the ENI System from having to know what entity it is specifically communicating with.
7) Any Functional Block may use negotiation to achieve its goals.
8) ENI will use role-based modelling to enable different functions and services to be viewed, accessed, and managed.
9) ENI will use one or more closed control loops as part of its main processing architecture. The primary control loop shall be at least one OODA-like closed control loop.

B.1.4 The Assisted System

The system that the ENI System provides recommendations and/or management commands to is referred to as the "Assisted System". An ENI System ingests and analyses data from the Assisted System, and then provides recommendations and/or commands to manage its behaviour. Changes to the Assisted System are not required in order to facilitate the use and rapid adoption of ENI.

The ENI System uses an API Broker to mediate between the ENI System and the Assisted System. This is because the ENI System and the Assisted System typically have different APIs. The API Broker defines the correct way for one system to request services from the other system without requiring an ENI System to understand the details of every API of every Assisted System that wants to communicate with it. All communication is done using external Reference Points.

B.1.5 Functional Architecture

B.1.5.1 Introduction

A high-level Functional Block diagram that includes the use of an API Broker is shown in Figure B.1. It is important to realize that a linear flow from input to output is not prescribed and does not have to actually occur. The arrows in this figure represent the directionality of data and information using any of the eighteen External Reference Points defined by the ENI System Architecture.

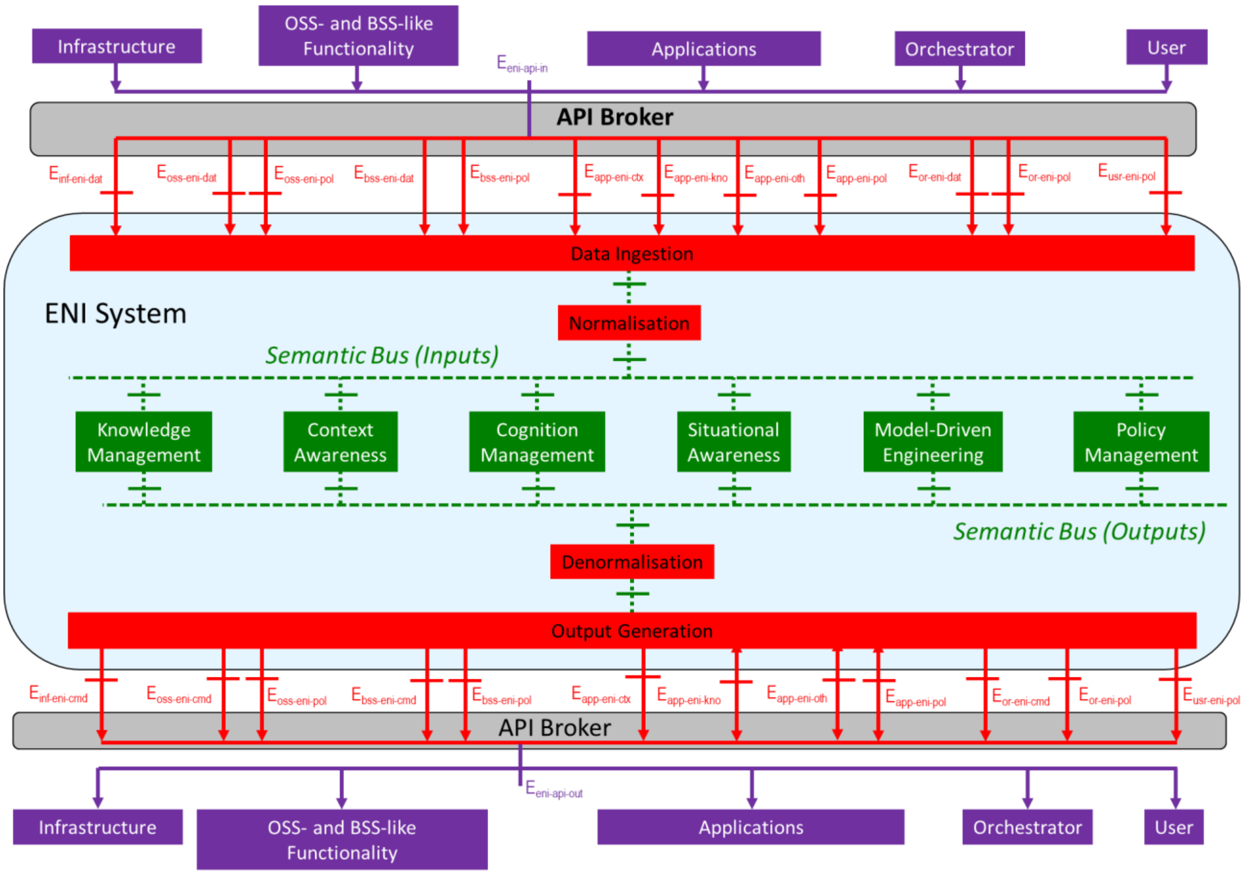

Figure B.1: High-Level Functional Architecture of ENI

The ENI API Broker decouples the ENI System from other external systems that it communicates with. It translates between the ENI APIs and APIs of the Assisted System. More specifically, the ENI API Broker ingests APIs through an appropriate ENI External Reference Point, analyses the API, and then routes the functionality of the ingested API to an appropriate ENI Functional Block (typically the Data Ingestion Functional Block).

Another purpose of the API Broker is to manage APIs. This includes the authentication and authorization of the entities that want to communicate using ENI APIs (e.g., between the ENI System and third-party applications, and vice-versa).

The ENI Functional Blocks are briefly described in the following subclauses.

B.1.5.2 Input and Output Processing

Input processing consists of two Functional Blocks: the Data Ingestion and the Normalization Functional Blocks. It is possible to combine these two Functional Block if desired.

All external data enters the Data Ingestion Functional Block. The Data Ingestion Functional Block is to collect data from multiple input sources and implement common data processing techniques to enable ingested data to be further processed and analysed by other ENI Functional Blocks.

The Normalization Functional Block processes and translates data received from the Data Ingestion Functional Block into a form that other ENI Functional Blocks are able to understand and use. This enables the data used by the Functional Blocks in an ENI System to be interpreted and understood in a unified and consistent manner; this facilitates the reuse of these Functional Blocks, enables their modularisation and generalisation, and supports vendor neutrality.

Similarly, output generation consists of two Functional Blocks: Denormalization and Output Generation. It is possible to combine these two Functional Blocks if desired.

The Denormalization Functional Block processes and translates data received from other Functional Blocks of the ENI System into a form that can be transformed to a form that a set of targeted entities are able to understand. For example, different data models may be used by different ENI Functional Blocks. It is possible to represent the same concept differently in different data models (e.g., customer data in an LDAP or X.500 directory vs. the same customer data in an RDBMS). ENI addresses this by ensuring that each data model is derived from a single information model, which facilitates reconciling these different representations of the same concept into a single object.

The Output Generation Functional Block converts data received by the Denormalization Functional Block into a form that the Assisted System is able to understand. This includes defining an appropriate set of protocols, changing the encoding of the data, and other related functions.

B.1.5.3 Knowledge Management and Processing

Knowledge Management and Processing consists of three Functional Blocks: Knowledge Management, Context-Aware Management, and Cognition Management Functional Blocks.

The Knowledge Management Functional Block represents information about both the ENI System as well as the system being managed. This includes differentiating between facts, axioms, and inferences. It also contains various repositories used by all ENI Functional Blocks.

The Context-Aware Management Functional Block describes the state and environment in which a set of entities in the Assisted System (i.e., system being assisted and/or governed) exists or has existed. Context-aware management is used to continuously update the context in which decisions are made. It thus serves as an important filter for assessing if goals are being met and the relevance of ingested data.

Context consists of measured and inferred knowledge, and typically changes over time. For example, it is possible that a company has a business rule that prevents any user from accessing the code server unless that user is connected using the company intranet. This business rule is context-dependent, and the system is required to detect the type of connection of a user and adjust access privileges of that user dynamically.

The Cognition Management Functional Block enables normalized ingested data and information, as well as the context that defines how those data were produced, to be understood. It can then evaluate the meaning of the data and determines if any actions need to be taken to ensure that the goals and objectives of the system are met. The Cognition Management Functional Block mimics some of the processes involved in human decision-making to better comprehend the relevance and meaning of ingested data.

B.1.5.4 Goal-Based Processing

The Situation Awareness Functional Block enables the ENI System to be aware of events and behaviour that may affect the behaviour of the Assisted System. Situation awareness enables the system to understand what has just happened, what is likely to happen, and how both may affect the goals that the system is trying to achieve. This implies the ability to understand how and why the current situation evolves. The ENI System observes the evolving of different situations, examining them for patterns within each situation and between different situations. Such knowledge is stored in the knowledge base of ENI. As such, identifying changes in both the current situation as well as possible future situations are critical for understanding how the environment is changing, and how those changes affect the goals that ENI is trying to achieve or maintain. For example, security situation awareness could include being aware of the scope and impact of the attack, correlating that with the behaviour of the adversary, so that effective counter measures can be implemented.

B.1.5.5 Generating Recommendations and Commands

The Model Driven Engineering (MDE) Functional Block uses a set of domain models that collectively abstract all important concepts for managing the behaviour of objects in the system(s) governed by the ENI System.

Reusable models define a set of concepts that are shared by all constituencies that use them. It maximizes productivity by defining common definitions and usage of concepts used by different entities, including Functional Blocks and systems.

A model is made reusable through the use of common elements that are arranged in similar patterns, or templates. This practice is called design patterns (see Design Patterns: Elements of Reusable Object-Oriented Software [5]), and occurs at multiple levels of abstraction, including objects, Functional Blocks, and even architectures.

The MDE Functional Block is responsible for enabling software development to be accomplished using models instead of code. The advantage of MDE is that models are, by definition, machine-readable. Hence, they can be used to specify Functional Blocks, programs, and applications. An example of MDE is generating code directly from a model.

MDE is a software development and implementation methodology that uses models in the design, understanding, implementation, deployment, operation, maintenance, and modification of software systems. A set of domain models abstract the concepts and activities to be managed. The MDE approach is meant to increase productivity by maximising compatibility between Functional Blocks and systems through the reuse of standardized models.

A set of models may be defined based on different viewpoints. Formally, a viewpoint is an abstraction of the function and behaviour of a system using a selected set of architectural concepts; this facilitates focusing on a particular aspect or set of responsibilities of the system. Model transformation tools and services are used to align the different models (e.g., deriving a set of data models from an information model), and for generating code.

The Policy Management Functional Block provides decisions to ensure that the system goals and objectives are met. Policies are used to provide scalable and consistent decision-making. Policies are generated from data and information received by the Knowledge Management and Processing set of Functional Blocks.

Policies may be used in several ways in an ENI System:

- Policies are defined by an ENI System for managing, monitoring, controlling, and orchestrating behaviour of Functional Blocks in the Assisted System.
- Policies are defined by an ENI System to request changes in the Assisted System (e.g., for monitoring a new output).
- Policies that are input to an ENI System by an external entity (e.g., end-user or application) are subject to verification by the ENI System (e.g., they need to pass a parsing or compilation stage with no errors or warnings produced).

In each case, policies may represent goals, recommendations, or commands. Typically, any information to be conveyed to the Assisted System or its Designated Entity take the form of a set of policies. Each set of policies may be made up of one or more imperative, declarative, and/or intent policy. The details of policy definition, generation, and processing are defined in clause A.6.3.9.

B.1.6 Reference Points

An ENI Reference Point is the logical point of interaction between specific Functional Blocks. Each Reference Point defines a set of related interfaces that specify how the Functional Blocks communicate and interact with each other.

An ENI External Reference Point is a Reference Point that is used to communicate between an ENI Functional Block and an external Functional Block of an external system.

An ENI Internal Reference Point is a Reference Point that is used to communicate between two or more Functional Blocks that belong to the ENI System. This communication stays within an ENI System and is not seen by systems that are external to an ENI System.

B.2 Use Cases

B.2.1 Introduction

The following describes two new use cases that are both related to Proof of Concepts (see clause B.3).

B.2.2 Use case #3-6: "Intent-based Cloud Management for VDI service"

An increasing number of enterprises are adopting VDI (Virtual Desktop Infrastructure) to support their employees in accessing a standard virtual desktop environment from various locations. In a VDI service, VDI operators need to determine and adjust cloud resource configuration (e.g., the number of VMs to be placed on each host, and the CPU and memory allocated for each VM). However, this decision requires a high level of skill and experience. Improper decision can lead to poor user experience or low resource efficiency. To assist the VDI operators with cloud resource decisions, this use case describes an

IBCM (Intent-based Cloud Management) system based on the ENI reference architecture. The IBCM automatically calculates the optimum cloud resource amount configuration that does not deteriorate the user experience as well as maximizes the cloud resource efficiency. This use case is further illustrated in PoC#14.

B.2.3 Use Case #3-7: "Intelligent Vehicle Diversified Service Fulfilment"

With the rapid development of Internet of things and mobile communication technology, the mobile access network represented by the intelligent vehicles has developed promptly. The Internet of vehicles is confronted with several problems. For example, in high-speed driving scenarios based on IP, it might occur to have communications delay and data dropped, which cannot 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. 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, how to ensure the audio and visual experience for the service remains a challenge. This use case proposed a Polymorphic network (PINet), which supports the coexistence and collaboration of IP, content, identification, geospatial location, and other self-defined identities in the same physical network in order to support services. An ENI system generates intelligent strategies 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.

B.3 Proofs of Concept

B.3.1 PoC #15: PINet — Polymorphic Intelligent Network

PINet (Polymorphic Intelligent Network) is defined as a network composed of multiple network functions running on standardized software and hardware interfaces and intelligent operation and maintenance management systems. It supports various capabilities, including routing, switching, interconnection, and other functions.

This PoC shows how PINet and ENI system cooperate to provide feasible design strategy for operators to automatically deploy network according to different types of business and application scenarios.

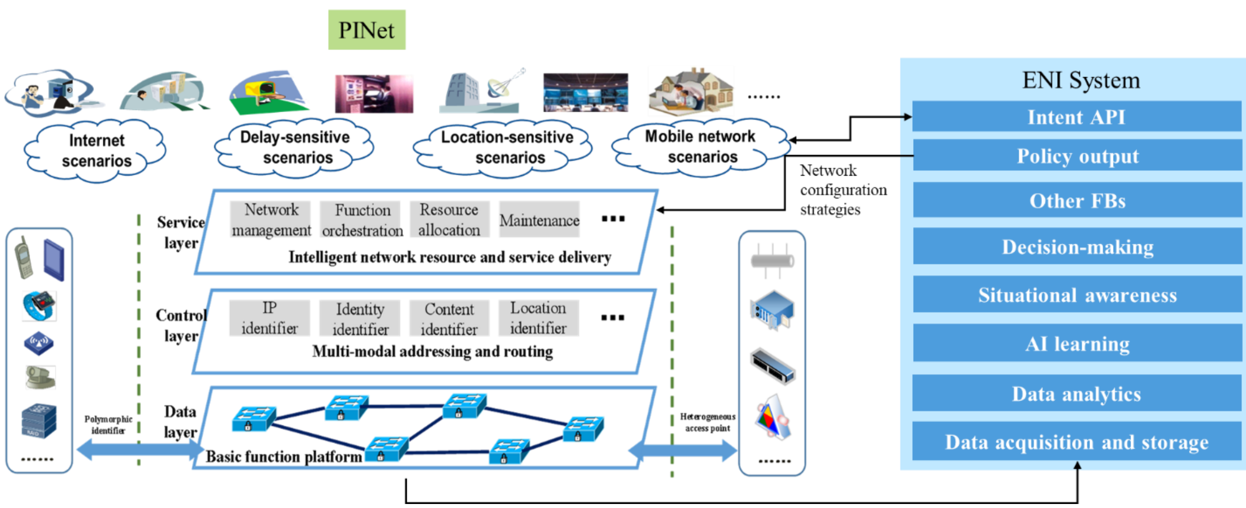

Figure B.2: PoC architecture and interaction with ENI system

The ENI system assists PINet to realize a polymorphic network, which is responsible for adaptive adjustment and configuration between network resources and diversified services in the following two major aspects:

- The ENI system collects user's network business requirements through the intent-based interface, makes detailed planning and performance expectations, and translates between business needs and offered services and sends the resulting service requirements to PINet. The control layer of PINet receives the policies and uses them to establish a polymorphic heterogeneous identification addressing and routing model. Furthermore, the data layer of PINet configures the topology, protocol, software/hardware, interface, etc. of the basic network to support upper layer application.
- The ENI system supports intelligent network management and network behaviour optimization of PINet by using cognitive management. The control operating logic can be formulated as a closed control loop of "perception-comprehension-plan-decision-action-adaptation" to realize efficient adaptive and fitting of network resources and upper services based on real-time perception of business requirement and resource status.

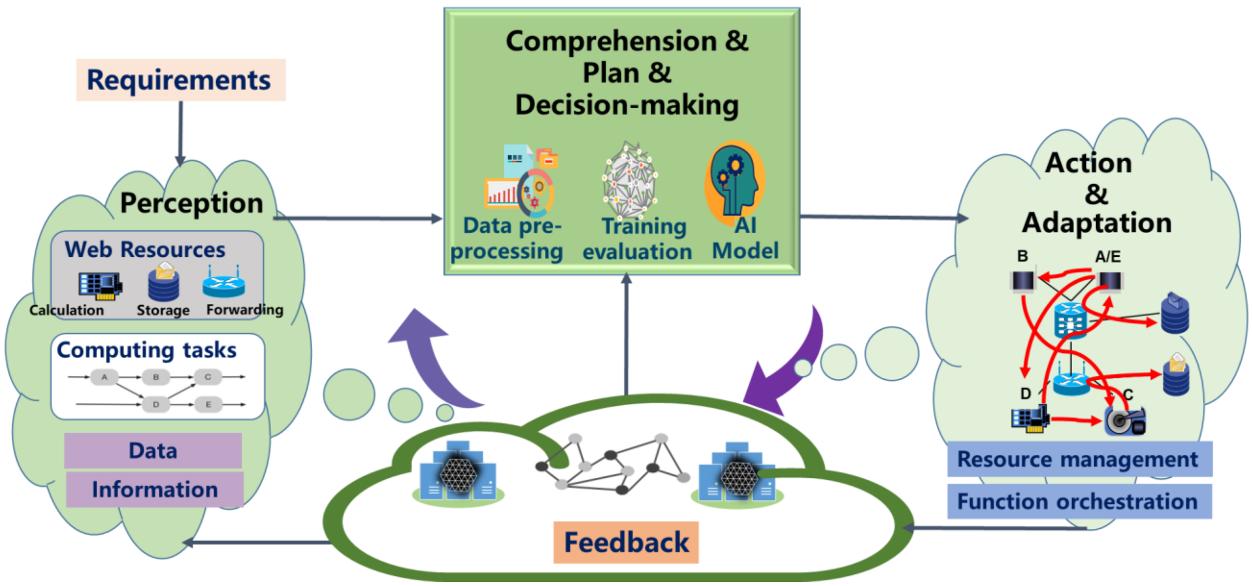

Figure B.3: Simplified Cognition Closed Control Loop Used in PINet

More specifically, "perception" means that the system ingests data in order to construct a resource view and a business view. "Comprehend" analyses data from the perception phase and relates it to service and business requirements through observing current status information, contextual, situational, and historical information, along with the user's intent. It then generates a state machine to manage the behaviour of the system. The "Plan" compares the current state to its desired state and, if different, generates different sets of state transitions (called paths) to move the system state to an acceptable state (i.e., one that achieves the goals of the intent). "Decide" chooses the best path from the paths available in the planning phase. "Act" translates the best path to a set of configuration and associated monitoring commands. "Adaptation" means that the system determines if the context, situation, and/or goals have changed, and manages resources and orchestrates functions to meet these changes. After adaptation the control loop is adjusted in accordance with changed system and business views and continues to operate to achieve its goals.

B.3.2 PoC #14: Intent-based Cloud Management

This PoC explores the application scenarios of intent-based management in cloud resource management for various services. In this PoC, representative scenarios of intent-based cloud management are identified: (1) Intent-based Cloud Management for VDI service, and (2) Intent-based Cloud Management for NFV workloads

The PoC demonstrated that the intent-based cloud management abstracts knowledge from the cloud telemetry data, infers the cloud performance under various workload and resource conditions, thus decide the cloud resource configuration that meet the intent – performance goal in this PoC.

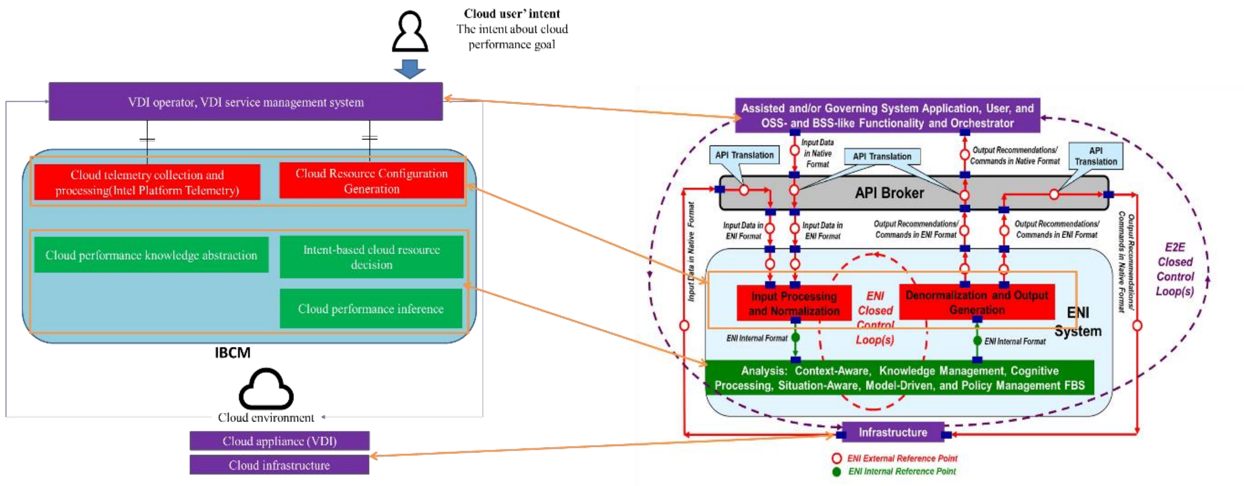

Figure B.4: Mapping to ENI reference architecture

For the two identified scenarios, the PoC has shown that intent-based cloud management is able to maintain the VDI performance intents with the minimal resource amount for the VDI service and manage to keep the latency and packet drop intents always satisfied, by using the smallest amount of resources for 5G User Plane Functions.

B.3.3 PoC #13: Intelligent Coverage Optimization of 5G Massive MIMO BS

To achieve high speed, massive connection, ultra-low latency, and high-quality user experience, 5G Massive MIMO adopted beam forming technology. Traditional networks used to optimize network performance by adjusting the tilt of the antenna. 5G Massive MIMO adopted adaptive beam pattern adjustment to serve different scenarios. However, flexible combination of beam envelope gain, horizontal beamwidth, vertical beamwidth, tilt, and azimuth of the antenna increases management complexity. Traditional manual adjustment method is not applicable for 5G Massive MIMO coverage optimization.

This PoC will demonstrate the use of AI based data analysis to enable policy-based coverage optimization for Massive MIMO Base Station. The high-level architecture of this PoC is shown below:

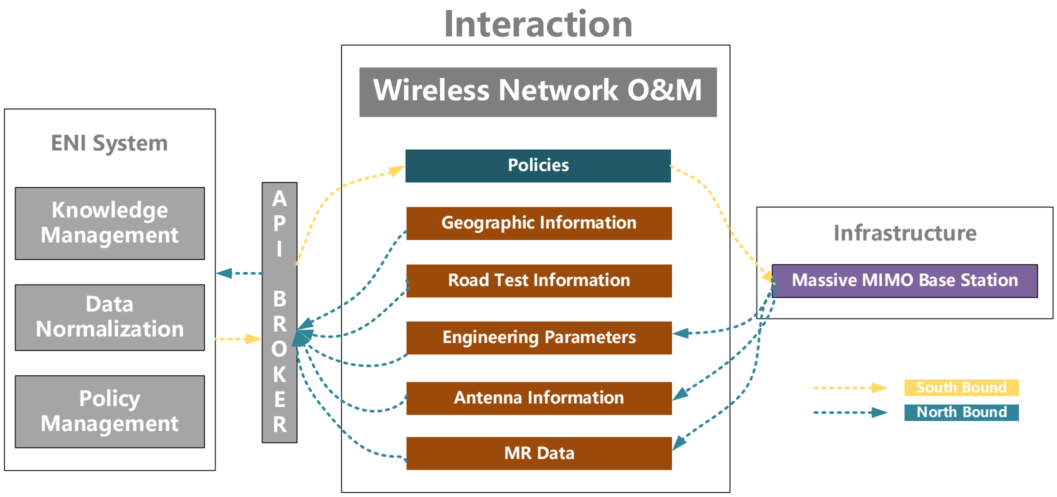

Figure B.5: Interaction between systems for intelligent coverage optimization of 5G Massive MIMO BS

Massive MIMO Base Station data is collected by different subsystems of the wireless network O&M system, which is connected to ENI system through an API broker. MR data, BS information (e.g., Engineering parameters, antenna information, etc.), geographic information (e.g., electronic map), etc. are extracted from external database. This enables the ENI System to analyse the data according to the requirements so that optimal coverage policies can be obtained and applied by operator to optimize BS coverage.

B.3.4 PoC #12: Intelligent Transport Network Optimization

This PoC demonstrates the intelligent optimization for transport network by applying AI/ML algorithms. Based on information including the topology of network and detailed configuration parameters of virtual network functions (VNF), and the pre-determined optimization policy, the ENI system intelligently analyses the overall topology and reach a global optimized network capacity and output the optimization plan automatically.

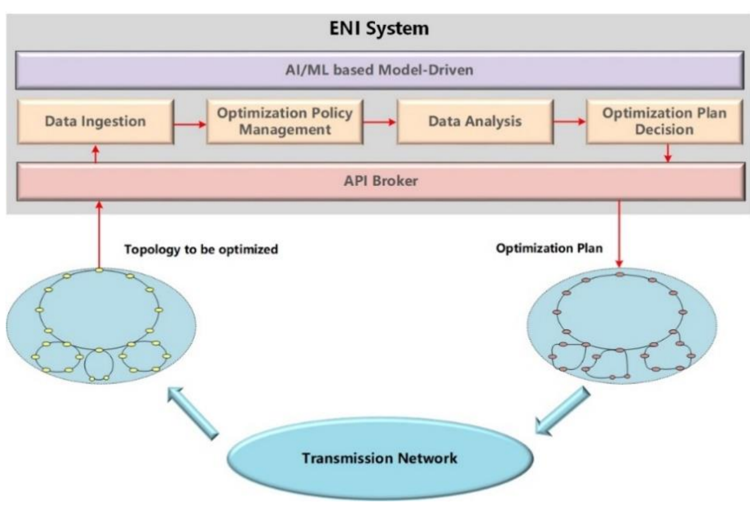

Figure B.6: Intelligent Transport Network Optimization

The architecture of the intelligent Transport Network Optimization (iTNO) system is shown below:

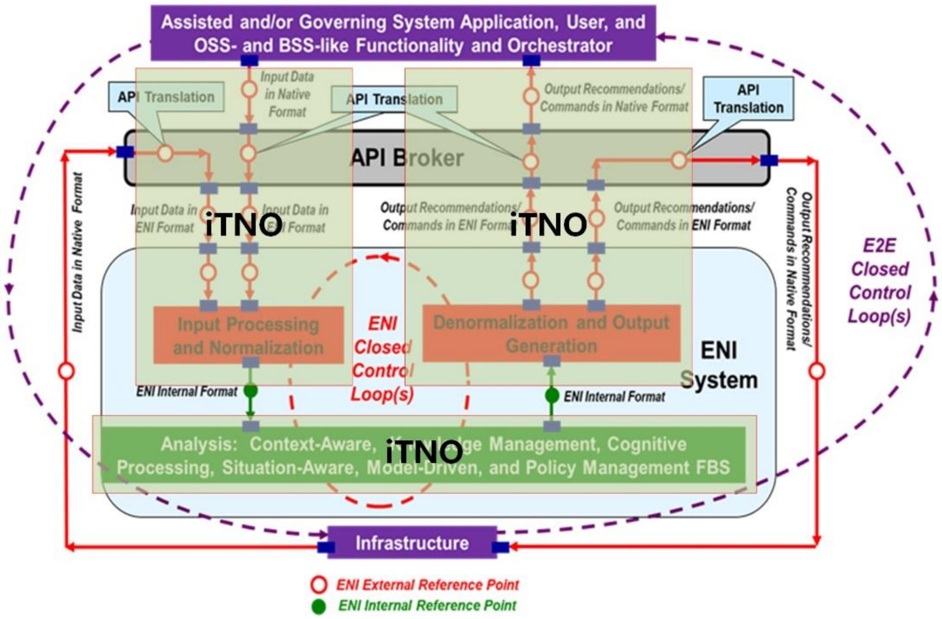

Figure B.7: Architecture of iTNO system

Detailed information from the transport network is collected and parameters used for optimization are also input to iTNO. These data are transformed into a common format based on normalization algorithms and will be further processed by other FBs. Based on the network topology and pre-processed data, iTNO analyses the relation amongst capacity of transmission ring, topology of transmission ring and the capacity of VNF. Based on the global utilization rate of capacity, iTNO executes iterative optimization until meeting termination conditions. According to analysis result, iTNO decides the optimization plan which will be used by the operator to optimize the transport network to reach globally optimum capacity utilization rate.

B.3.5 PoC #11: Intelligent Energy Management of DC

This PoC demonstrates viable solutions and methodologies for the energy management of DCs (Data Centre) using a set of AI algorithms based on DC dynamic environment data to help DCs achieve a better PUE (Power Usage Effectiveness) and reduce OPEX for telecom operators.

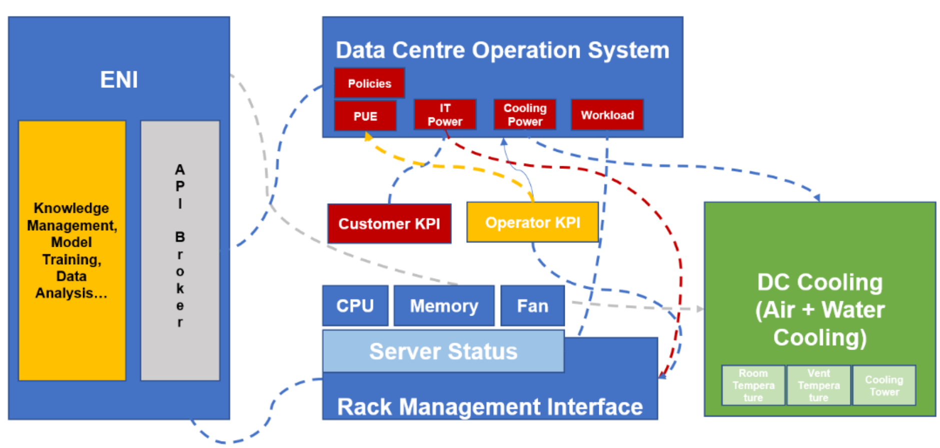

Figure B.8: System architecture of the PoC

Figure B.8 shows an interaction of internal and external reference points. In this scenario, different subsystems of DC energy management can be connected to the ENI system where DC data collected through the data acquisition process are provided to the ENI system. This enables the ENI System to analyse the data according to the requirements so that appropriate energy power saving policies are obtained and applied to the DC system.

The intelligent DC energy management system contains big data platform deployed in centralized cloud, EDA in edge platform and front end in each DC. Data including basic data of DC room, temperature data, real-time data of cabinet, real-time data of air conditioning and equipment parameter data are collected and uploaded to centralized system to analyse based on AI. Energy management policies are released to front end system to achieve automatic control of the air supply, air conditioning or water-cooling unit.

B.4 Collaboration with other Standards Organizations

ISG ENI is dedicated to interacting with other SDOs in all of its activities. The focus of this white paper is the development of a cognitive network. Hence, the main SDOs that ISG ENI will focus with are:

- ETSI TC INT AFI, focusing on interacting with autonomic architectures
- ETSI ISG SAI, focusing on securing artificial intelligence
- ETSI TC CYBER, focusing on developing cybersecurity for cognitive intelligence
- EU H2020 Projects
- ITU-T SG13 ML5G, focusing on machine learning for cognitive intelligence
- MEF, focusing on interacting with information and data models and APIs
- TMF, focusing on interacting with autonomous networks

Annex: Terminology Specific to This White Paper

The following terms apply to this White Paper.

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.

assisted system: system that the ENI system is providing recommendations and/or management commands to.

cognition: process of acquiring and 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.

cognitive offloading: replacing human judgment with technology.

Domain Specific Language (DSL): small human-understandable language that uses a higher level of abstraction to communicate and configure software systems for a particular application domain.

ENI Reference Point: the logical point of interaction between specific Functional Blocks.

ENI External Reference Point: ENI Reference Point that is used to communicate between an ENI Functional Block and an external Functional Block of an external system.

ENI Internal Reference Point: ENI Reference Point that is used to communicate between two or more Functional Blocks that belong to the ENI System.

functional architecture: defines the major functions of each module, and their interaction.

functional block: abstraction that defines a black box structural representation of the capabilities and functionality of a component or module, and its relationships with other functional blocks.

Model-Driven Engineering (MDE): approach in which models are central to all phases of the development and implementation processes.

policy: set of rules that is used to manage and control the changing and/or maintaining of the state of one or more managed objects.

declarative policy: type of policy that uses statements from a formal logic to describe a set of computations that need to be done without defining how to execute those computations.

imperative policy: type of policy that uses statements to explicitly change the state of a set of targeted objects; the canonical form is a triple, consisting of a set of Event, Condition, and Action Boolean clauses.

intent policy: type of policy that uses statements from a restricted natural language (e.g., an external DSL) to express the goals of the policy but does not specify how to accomplish those goals.

self-awareness: the recognition of the environment and performance of an entity in that environment.

self-knowledge: the understanding of the current situation and the effects of issuing a command in that situation.

situation: set of circumstances and conditions at a given time that may influence decision-making.

situation awareness: perception of data and behaviour that pertain to the relevant circumstances and/or conditions of a system or process, the comprehension of the meaning and significance of these data and behaviours, and how processes, actions, and new situations inferred from these data and processes are likely to evolve in the near future.

References

The following references are used in the White Paper:

[1] ETSI ISG ENI, "Experiential Networked Intelligence (ENI); System Architecture", GS ENI 005

[2] D. Raymer. S. van der Meer, J. Strassner, "From Autonomic Computing to Autonomic Networking: An Architectural Perspective", Fifth IEEE Workshop on Engineering of Autonomic and Autonomous Systems, 2008

[3] OMG, "OMG Meta Object Facility (MOF) Core Specification", version 2.5.1, October 2019
Available at: https://www.omg.org/spec/MOF/2.5.1/PDF

[4] OMG, "Model Driven Architecture (MDA) - MDA Guide rev. 2.0", 2014
Available at: https://www.omg.org/cgi-bin/doc?ormsc/14-06-01

[5] E. Gamma, R. Helm, R. Johnson, J. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, 1994, ISBN: 0-201-63361-2

[6] ETSI ISG ENI, "Experiential Networked Intelligence (ENI); Representing, Inferring, and Proving Knowledge in ENI", GS ENI 019