

# TM Forum Introductory Guide

## Autonomous Networks

## Technical Architecture

IG1230

Team Approval Date: 28-May-2021

Release Status: Production	Approval Status: TM Forum Approved
Version 1.0.1	IPR Mode: RAND

## Notice

Copyright © TM Forum 2021. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to TM FORUM, except as needed for the purpose of developing any document or deliverable produced by a TM FORUM Collaboration Project Team (in which case the rules applicable to copyrights, as set forth in the [TM FORUM IPR Policy](#), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

This document and the information contained herein is provided on an “AS IS” basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

TM FORUM invites any TM FORUM Member or any other party that believes it has patent claims that would necessarily be infringed by implementations of this TM Forum Standards Final Deliverable, to notify the TM FORUM Team Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this deliverable.

The TM FORUM invites any party to contact the TM FORUM Team Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this TM FORUM Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this TM FORUM Standards Final Deliverable. TM FORUM may include such claims on its website but disclaims any obligation to do so.

TM FORUM takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this TM FORUM Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on TM FORUM's procedures with respect to rights in any document or deliverable produced by a TM FORUM Collaboration Project Team can be found on the TM FORUM website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this TM FORUM Standards Final Deliverable, can be obtained from the TM FORUM Team Administrator. TM FORUM makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

Direct inquiries to the TM Forum office:

181 New Road, Suite 304  
Parsippany, NJ 07054 USA  
Tel No. +1 862 227 1648  
TM Forum Web Page: [www.tmforum.org](http://www.tmforum.org)

# Table of Contents

<b>Notice</b> .....	2
Table of Contents .....	4
List of Figures.....	6
List of Tables.....	7
Overview.....	8
Readers Guide .....	9
Document Structure.....	9
Audience .....	10
Section Overview .....	10
Terminology .....	11
1       Overview of Autonomous Networks.....	13
1.1   Introduction .....	13
1.2   Scope .....	13
1.3   Objectives of Autonomous Networks .....	14
1.4   Evolution to Autonomous Operations .....	15
1.5   Technical Requirements.....	16
1.6   Key Concepts.....	17
2       Autonomy Concepts and Principles .....	19
2.1   Automatic, Autonomic, Autonomous .....	19
2.2   Principles of Human-Machine Interaction .....	19
2.3   Architectural Principles .....	23
3       Autonomous Network Technical Architecture.....	28
3.1   AN Framework Overview .....	28
3.2   Core Concepts .....	29
3.3   Autonomous Domain .....	31
3.4   AN Technical Architecture Diagram .....	33
3.5   Patterns for AN to AN system Integration .....	34
3.6   Patterns for AN to Non-AN System Integration.....	35
3.7   AN Architecture Model .....	36
3.8   AN Technical Architecture using other Reference Architectures .....	38
4       AN Levels .....	40
4.1   Levels of Autonomous Networks .....	40
4.2   Benefits of Levels .....	40
4.3   AN Level Classification.....	41

4.4	AN Level Evaluation.....	42
4.5	A Practical AN Level Evaluation Methodology .....	43
5	Autonomous Domains.....	47
5.1	Definition.....	47
5.2	Example of an Autonomous Domain .....	48
5.3	Multi-domain Collaboration and Autonomy.....	50
6	Intent-driven Interactions .....	51
6.1	Intent and Intent Evolution .....	51
6.2	Implications of Intent .....	52
6.3	General Properties of intent .....	53
6.4	Classification of Intent.....	54
6.5	Intent within the AN Framework .....	55
6.6	Intent-driven Management.....	60
7	Control Loop Mechanisms .....	62
7.1	Control Loop Concepts.....	62
7.2	Evolution of Control Loop Mechanisms .....	65
8	Knowledge and Intelligence .....	70
8.1	AI in AN Technical Architecture .....	70
8.2	Developing and using AI Models.....	71
8.3	Relevant AI Technologies for AN scenarios.....	74
8.4	AI in common telecoms use cases .....	76
8.5	AIOps Service Management.....	77
9	Intelligent Network Infrastructure.....	80
9.1	Intelligence in Equipment Design.....	80
9.2	Digital Awareness and Intelligence Enabling .....	81
9.3	Infrastructure Requirements.....	82
10	Conclusions .....	83
11.	Administrative .....	84
11.1	Abbreviations .....	84
11.2	References.....	85
11.3	Document History .....	86
11.4	Acknowledgements.....	86
12	Appendix I: AN Technical Architecture Metamodel.....	89
13	Appendix II: AN Technical Architecture Guides .....	90

# List of Figures

Figure 0-1 Readers Guide to AN Technical Architecture .....	9
Figure 1-1 Framework to Technical Architecture .....	14
Figure 1-2 Autonomous Networks Strategy Map.....	15
Figure 1-3 From Automated to Autonomous Operations.....	16
Figure 1-4 Key Autonomous Network Concepts .....	17
Figure 2-1 Slider of Autonomy.....	20
Figure 2-2 Slider Control sets the Autonomy Level (0-5) .....	20
Figure 2-3 Humans and Machines are Agents.....	21
Figure 2-4 Intelligent Agents .....	21
Figure 2-5 Human delegates Responsibility to Machine but not Accountability.....	22
Figure 2-6 Machine is a Party, Decision Maker is a PartyRole .....	22
Figure 2-7 AI Safety Quadrants.....	24
Figure 2-8 Security infrastructure for autonomous network.....	26
Figure 3-1 Building Blocks of Framework .....	28
Figure 3-2 TM Forum Autonomous Network Framework.....	29
Figure 3-3 AN can sense and adapt to its environment .....	30
Figure 3-4 Autonomous Domain Business Service Categories.....	32
Figure 3-5 AN Technical Architecture.....	33
Figure 3-6 Intent interactions between 2 AN systems (domains or platforms).....	34
Figure 3-7 Adapter to Intent Integration Pattern.....	35
Figure 3-8 Models of Managed Entities .....	37
Figure 3-9 Key Entities in model.....	38
Figure 3-10 ETSI ZSM scope within AN Technical Architecture.....	39
Figure 4-1 A practical AN Level evaluation methodology .....	43
Figure 5-1 Autonomous Domains are composite.....	49
Figure 5-2 Autonomous Domain Capabilities.....	49
Figure 5-3 Platform versus Domain .....	50
Figure 5-4 Multi-domain Collaboration and Autonomy .....	50
Figure 6-1 Intent Handling function .....	56
Figure 6-2 Autonomous Network operation based on Intent handling .....	57
Figure 7-1 Control Loop within a single Autonomous Domain .....	62
Figure 7-2 Control Loop Mechanism of Autonomous Network .....	63
Figure 7-3 Responsibility can be shared across the control loop .....	64
Figure 7-4 Evolution of Control Loop Modelling .....	66
Figure 7-5 Control Loop Operations .....	67
Figure 7-6 FOCALE – the Autonomic Computing Element and Closed Loops .....	67
Figure 7-7 Control Loop high-level conceptual model .....	68
Figure 7-8 Control Loop ABE.....	69
Figure 8-1 Layers of AI and Network Intelligence in AN Reference Architecture .....	71
Figure 8-2 Closed Loop process for Model Development and Inference.....	72
Figure 8-3 Automated Machine Learning ( AutoML).....	74
Figure 8-4 Federated Learning .....	75
Figure 8-5 Knowledge is connecting the dots (graphic by David Somerville) .....	75
Figure 8-6 AIOps in a general software lifecycle process .....	78
Figure 8-7 Processes in scope of the AIOps Framework .....	78
Figure 11-1 Concepts and Relationships .....	89
Figure 11-2 Open Digital Architecture Package Diagram .....	89

# List of Tables

Table 1 Readers Guide.....	10
Table 2 Terminology .....	12
Table 3 The 4 A's.....	19
Table 4 TM Forum AN Levels Framework.....	41
Table 5 Decomposition of Control Loop Operation Processes.....	44
Table 6 Processes defined within AIOps Service Management Framework.....	79
Table 7 Infrastructure Requirements for SDO Cooperation.....	82
Table 8 Related Guides.....	90

## Overview

One of the main drivers for automation in communications service providers' (CSPs') networks is the urgent need to reduce complexity so that they can lower operating costs. This is especially important as networks become software-defined and 5G is deployed. Even more importantly, CSPs must automate their networks and operations to deliver rapid service adaptation and deployment, with the dual objectives of improving customer satisfaction and increasing revenue. A new TM Forum project is addressing the complex business issues surrounding both, along with solving technical challenges.

Using artificial intelligence (AI) and machine learning, CSPs are aiming to fully automate the lifecycle of the services they deliver to end customers and of internal network services, which are chains of technical components. Increasingly, both kinds of services are being delivered across a diverse ecosystem of partners that can include not only multiple CSPs but also many network component suppliers and third-party application providers. The idea is to *abstract* the network as a set of software services and then use intent, automated closed control loops, and machine learning to make networks and operations self-configuring, self-optimizing, self-organizing, self-healing, and self-evolving.

*The purpose of abstraction is not to be vague, but to create a new semantic level in which one can be absolutely precise. (E.W. Dijkstra)<sup>1</sup>*

These abstractions are catalogued and explained in this document as part of an overall AN blueprint, to guide operators in the building and measuring of the autonomous capabilities in their platforms and networks.

Implementing automation is both a business and technical innovation challenge that will need a small number of straightforward implementation and integration patterns to achieve self-organizing autonomous networks. These must be agreed upon and adopted by all the players.

To do this work, TM Forum established the TM Forum Autonomous Networks Project, which began with the publication of a white paper called [Autonomous Networks: Empowering Digital Transformation for the Telecoms Industry](#). Members from BT, China Mobile, Ericsson, Huawei, Orange and Telstra collaborated on the paper and presented it at Digital Transformation World 2019 and the team has grown to include members from over 30 different organizations around the world.

---

<sup>1</sup>The Humble Programmer by Edger W. Dijkstra  
<https://www.cs.utexas.edu/~EWD/transcriptions/EWD03xx/EWD340.html>

# Readers Guide

## Document Structure

The AN Technical Architecture guide is structured in this release into three separate guides (see Figure 0-1) aimed at a specific set of audience needs:

- **IG1230: Autonomous Network Technical Architecture**  
Audiences: Product Managers, Architects and Developers.
- **IG1230A: Autonomous Network Scenario Realizations**  
Audiences: Architects and Developers.
- **IG1230B: Autonomous Network – Industry Standards**  
Audiences: Product Managers, Architects and Developers.



**Figure 0-1 Readers Guide to AN Technical Architecture**

Please see Table 8 Related Guides in the appendixes for more details on 1230A and 1230B.

## Audience

This Autonomous Network Technical Architecture is intended for architects and implementers wishing to understand the key principles for designing and realizing agile Autonomous Networks.

### Audience Topics

#### For Product Managers

- IG1230 (this guide) : Gives Product Managers an overview of the main concepts and principles in implementing AN. Also, it provides examples taken from catalysts of how the AN Framework is related to practical implementations.

#### For Architects and Developers

- IG1230 (this guide) : Explains the overall rationale for AN and describes the key concepts and principles. Provides the technical overview of the AN Reference Architecture
- IG1230A: Provides examples and results of industry proofs-of-concepts and from TM Forum catalyst projects.
- IG1230B: Provides an industry standards landscape on AN and stresses the need for better collaboration.

## Section Overview

The previous sections gave an overview of the key concepts, but other concepts are important which are developed further in this document.

Section	Topic	What you will learn
2	Autonomy Concepts & Principles	Discussion of additional concept important for machine and human interaction for both Tenant interaction and especially Supervision and control Business service
3	AN Technical Architecture	The building blocks of AN
4	AN Levels	How to measure the level of autonomy of a task within autonomous networks
5	Autonomous Domains	Autonomous domains that encapsulate operational self-management capabilities within a specific boundary and expose service capabilities
6	Intent-driven Interactions	Describes how layers of operational management can leverage an intent or declarative approach to specifying needs and requirements
7	Closed Loop Mechanisms	Details the mechanisms behind how management systems respond to changes, events, and intents and adapt to new environmental conditions
8	Knowledge and Intelligence	Human and Machine Intelligence (AI) in AN
9	Intelligent Network Infrastructure	Network elements and network architecture leveraging intelligence.
10	Conclusions	Concluding thoughts on the future of autonomous networks and the next steps for the AN Project

Table 1 Readers Guide

## Terminology

Term	Definition / Source
Autonomy	The capability to make decisions free from human control.
Automatic	Able to operate independently of human control
Autonomous	Having autonomy.
Autonomic <sup>2</sup>	Acting or occurring involuntarily ( <a href="#">Merriam-Webster Dictionary</a> ). Resulting from internal stimuli; spontaneous ( <a href="#">The Free Dictionary</a> )
Autonomous Domain	An autonomous domain is a system (or set of systems or agents) that is capable of autonomous behavior (e.g., resolve tasks, adhere to objectives) without manual human intervention. The autonomous domain does this by realizing self-management capabilities using a closed control loop mechanism, using four key phases: awareness, analysis, decision-making, and execution. It is a domain with an administrative governance boundary that defines the scope of encapsulated autonomous behaviors.
Autonomous Network	An Autonomous Network is a system of networks and software platforms that are capable of sensing its environment and adapting its behavior accordingly with little or no human input.
Autonomous Network Framework	<i>The collection of concepts described in this document that enable automation, agility and intelligence in innovation by CSPs and vendors and simplified network and IT Platform architecture for integration &amp; operations.</i> <i>The ultimate goal is NOT automation, but rather simplification for ALL users.</i> <i>Based on words in the scope section 1.2. but does that not capture the Business and Service platform concept.</i>
Accountability	Accountability requires the AI and people behind the AI to explain, justify, and take responsibility for any decision and action made by the AI. Mechanisms, such as governance and tools, are necessary to achieve accountability. <sup>3</sup>
Business Service	A Business Service is a service that is delivered to business customers by business units. For example, delivery of financial services to customers of a bank, or goods to the customers of a retail store. Successful delivery of business services often depends on one or more IT services. A business service may consist almost entirely of an IT service – for example, an online banking service or an external website where product orders can be placed by business customers.
Responsibility	The opportunity or ability to act independently and take decisions without authorization. Oxford Dictionary
Intent	A set of operational goals that a network should meet and outcomes that a network is supposed to deliver, defined in a declarative manner without specifying how to achieve or implement them
Control Loop	Control loops are used to enable autonomous systems to adapt their behavior to respond to changes in user needs, business goals, or environmental conditions.
Closed Loop	Autonomous system accomplishes a task without human taking an active role during task execution. Also referred to as a Closed Control Loop.

<sup>2</sup> First reference to term in networking was to ‘Autonomic Computing’ by Kephart and Chess (IBM, 2003) a was defined as ‘computing systems that can manage themselves given high-level objectives from administrators.

<sup>3</sup> LF-AI Trusted AI [principles](#)

Term	Definition / Source
Open Loop	<p>Autonomy: A human operator starts, stops, or changes a goal or constraint related to a task during task execution.</p> <p>Control theory: A control system that does not have a feedback loop and thus is not self-correcting.</p>
Platform	A Platform is a collection of systems that collectively provide a well-defined block of business functionality exposed via open APIs.
Domain	<p>A domain is a subset of a specific management area.</p> <p>Depending on the context, the term "domain" has different meanings. It is a good practice to qualify a domain, e.g., security domain, ODA Domain.</p> <p>Originally from TMF263F "Relationships between the concepts of Domain and Platform"</p>
Interpretability	Interpretability is about the extent to which a cause and effect can be observed within a system.
Explainability	Explainability is the extent to which the internal mechanics of a machine or deep learning system can be explained in human terms. (Sometimes confused with <i>Interpretability</i> ) Interpretability is about being able to discern the mechanics without necessarily knowing why. Explainability is being able to quite literally explain what is happening.
Agent	A Human or a Machine in the context of AN. An agent is an intelligent entity that runs independently, acts by itself, is affected by external environments, continuously detect from the environment to improve its capabilities, and combines inference and knowledge representation, it has the characteristics of autonomy, reactivity, adaptability, communication, and self-learning
Machine Learning (ML)	Getting computers to learn from data in the form of observations and real-world interactions in order to create a model of the real-world.
ML Pipeline	A set of logical entities (each with specific functionalities) that can be combined to form an analytics function (ITU)
Management Domain	Management Domain class represents a special grouping of ManagedEntities that has two important properties. First, it is used to partition managed objects into a meaningful logical grouping. Second, it defines a common administrative domain that is used to administer the managed objects that it contains. (GB922 "SID Root Entities" and TR275 "Core Networking Resources Business Entities")

**Table 2 Terminology**

# 1 Overview of Autonomous Networks

## 1.1 Introduction

Level 5 autonomy in an autonomous system means the system is self-sustaining and runs “hands-free”, that is with no human control involved. In the case of complex communication networks, Level 5 seems to be quite some distance away from happening and as of now, no CSPs have achieved this level. In some scenarios, Level 5 may not even be desirable given regulations, but no one can argue that the journey towards full autonomy is not a challenge! And as is typically the case, it is exactly these difficult challenges where the real value lies for the entire telecom ecosystem. We are undertaking this journey towards self-driving networks so that we can gain capabilities that we don’t possess currently. Some of the self-management capabilities have been described in the AN business requirements and framework guide [IG1218] and we recap on these briefly in the next section and describe the strategy behind autonomous networks.

TM Forum members are collaborating in the [Autonomous Networks Project](#) to develop a common understanding of what defines autonomous networks and build a consensus about how to implement them.

## 1.2 Scope

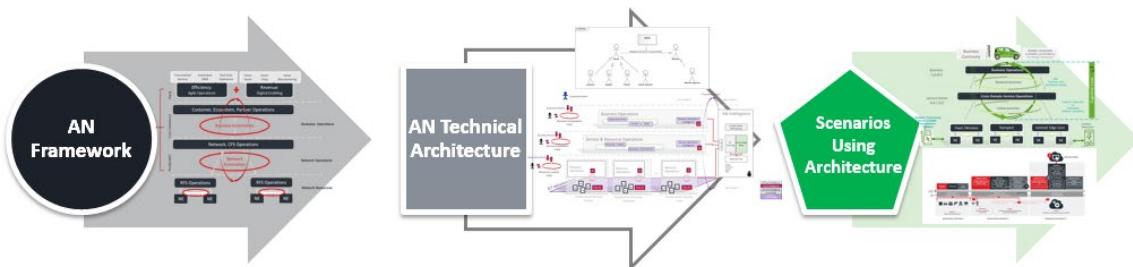
In May 2019, TM Forum published a whitepaper on Autonomous Networks on the autonomous network concept and a level of autonomy classification for the automation of intelligent operations. The whitepaper sets out the business drivers for CSPs to go beyond simple automation to autonomy and describes both the challenges and opportunities – principally automation, simplified network architecture enabling efficient intelligent operations, and rapid and agile integration of intelligent network innovation by CSPs and vendors. The ultimate goal is not automation, but rather *simplification* for all users. The whitepaper defines a high-level conceptual **Autonomous Network Framework** together with a 6 Level classification for autonomous networks and their application to scenarios.

Some key characteristics of the Autonomous Network are as follows:

- *AN is about services, operations, simplified network architecture.*
- *AN is service-agnostic and technology agnostic.*
- *The Framework does not focus on specific “technologies” such as AI.*

## 1.3 Objectives of Autonomous Networks

The AN Framework serves both the business and technology aspects of Autonomous Networks. The objective of this guide is to develop a concrete technical architecture of the Framework so that we go deeper into how the architecture can be realized in real projects and implementations (see Figure 1-1).



**Figure 1-1 Framework to Technical Architecture**

This document defines a Technical Architecture that when implemented:

- *Realizes the AN Business Architecture requirements including layering, closed control loop within and across Autonomous Domains, and exhibits 'Self X' capabilities.*
- *Are 'Agile by Design' using configuration - rather than coding to define an AN system solution.*
- *Uses a simple AN Domain model that can be realized and combined in many ways (akin to building with Lego™ blocks which have a simple form that can be combined to create complex implementations).*
- *Allows varied combinations of AN Domains and associated control loops such that they can be simply combined to realize complex AN systems.*
- *Allows heterogeneous technology implementations to be combined and simply integrated to enable innovation and evolution.*

In the short to medium-term organizations face challenges related to efficiency. They need to reach new levels of *operational efficiency* – i.e., to do more with the same level of budget.

The strategy map (Figure 1-2) explores the business drivers of the high-level AN Framework. The *financial* perspective of the AN strategy is referred to here as the Business Perspective and the key sub-goals are those of Increased *Efficiency* and Increased *Revenue*. The value proposition of the improved customer experience is represented in the Customer Perspective, and this is what influences how well we can achieve our overall objectives.

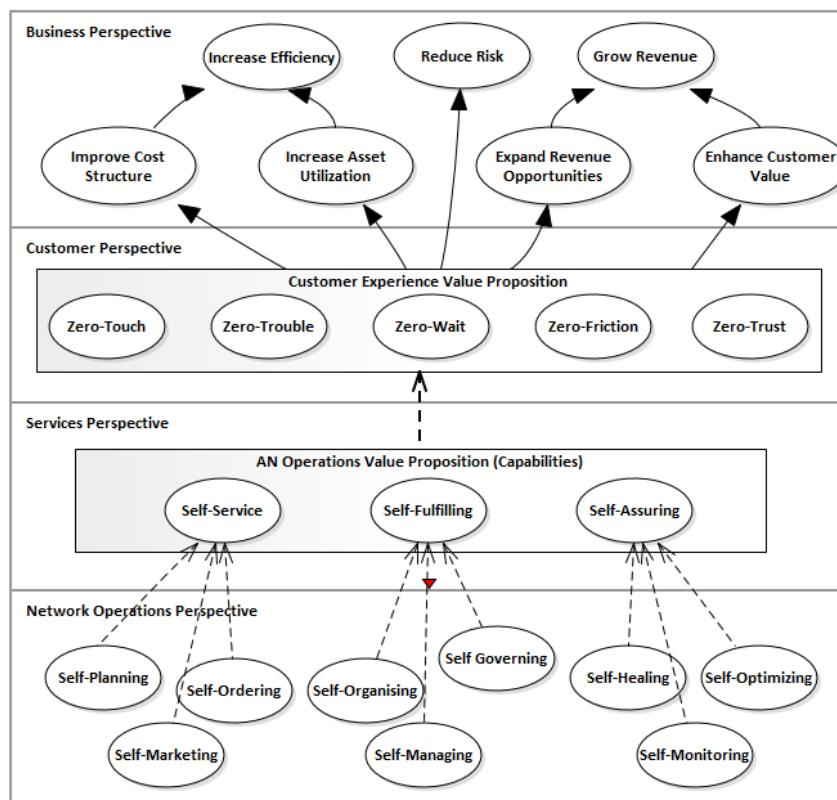


Figure 1-2 Autonomous Networks Strategy Map

The “Zero-X’s” refer to **zero-touch**, **zero-trouble**, **zero-wait**, **zero-friction**, and **zero-trust** and describe the characteristics of the way we want our end-customers to consume the new revenue-generating services. The “Self-X’s” capabilities are shown as supporting the Zero-X’s in the customer perspectives and classified in a hierarchy where those pertain to Services (e.g., self-service) are supported by those that are more focused on the network resources (e.g., self-monitoring).

## 1.4 Evolution to Autonomous Operations

The self-management capabilities are realized within Autonomous Domains, that enable the strategic use of AI technology within Autonomous Domains for Observation, Orientation, Decision, and Action (OODA) process steps within closed control loops. AI solutions without these AN technical concepts will lead to the tactical use of AI solutions with coupling between the CSP operations and the decision-making in Domains which will greatly inhibit lasting operation agility and flexibility.

Some of the challenges of Traditional Network Management include:

- Complex and time consuming to specify the resource models
- Tight coupling with tenants  
requires a lot of tenant shared knowledge
- Difficult to implement and hard to change.

In terms of evolution, we can progress from manual operations to automated operations and ultimately autonomous operations.

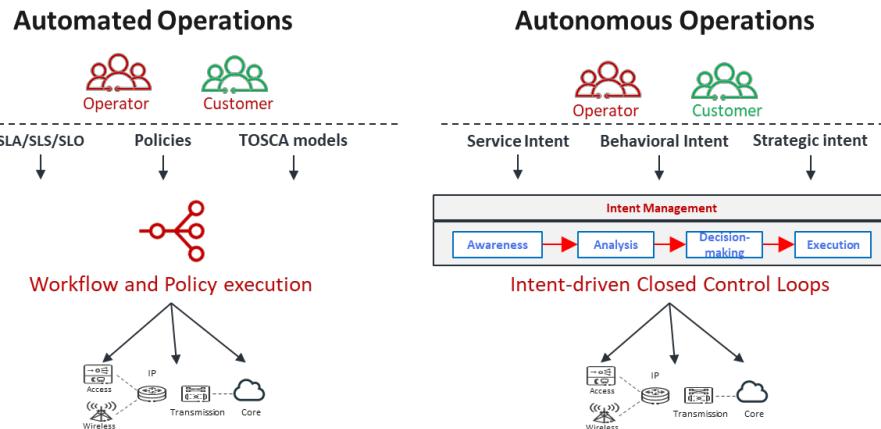


Figure 1-3 From Automated to Autonomous Operations

**Manual operations:**

Input: Documents and work orders

- A team of specialists operates the network manually by configuring, provisioning, assuring, optimizing ...
- All decisions are made, and all actions are initiated by humans.

**Automated operations: where we automate those manual tasks**

Input: SLA/SLS/SLO, policies, TOSCA models, etc.

- Workflows and policies invoking actions through dedicated API
- Tight coupling between services and policies
- ‘Zero-touch’ experience only as long as policies match situations
- Intelligence and decision-making by humans at design-time.

**Autonomous operations:**

Input: Intent for setting goals and targets

- A paradigm shift from the explicit invocation of actions to goals-based autonomy
- AI can explore and find new solutions
- Zero-touch because the machine can assess utility, consequences, risk, etc.

**The benefits of the Autonomous Networks approach include:**

- Agility through intent and the consistency imposed by closed control loop mechanisms
- Support Self-X management capabilities
- The basis for strategic and comprehensive use of AI.

## 1.5 Technical Requirements

The evolution of network operations needs to support the following four technical requirements:

- *Autonomous Domain*  
Autonomous Networks move away from the rigid network integration models of the past where internal details of a network realization are exposed to the tenant /consumer. Those models prevent and inhibit Self X business capabilities by creating a high level of coupling between the tenant/consumer view and the implementation. Autonomous domains are the key concept for enabling agility through the separation of concerns- especially the ‘what’ from the ‘how’ - in establishing and evolving Autonomous Network systems and for allowing their supervision by CSP Operations.
- *Intent-driven Interfaces*  
*In contrast to the very-detailed traditional resource model APIs, Intent-driven APIs focus on*

*describing the ‘what’ – the ‘what’ being the expectations , the desired outcome from an Autonomous Domain – rather than exposing details of ‘how’ to configure the resources within an AN Domain. Intent-driven service interfaces for the tenant consumer are an essential prerequisite for enabling Autonomous Domains supporting self X capabilities.*

- *Self-X Capabilities and Closed control Loop Optimization*

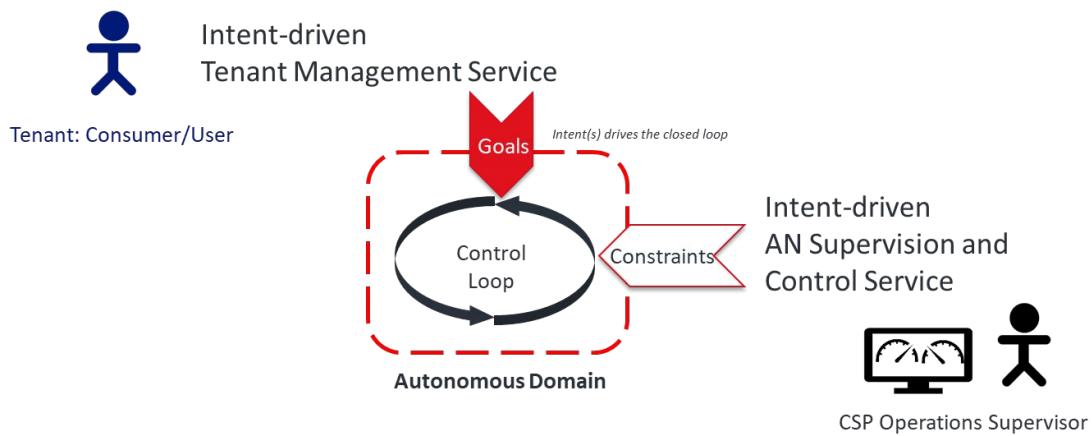
*Closed control loops are an essential part of an Autonomous domain optimization of the tenant service. Multiple control loops may be used to optimize over different timescales (e.g., NGMN requirements of fast and slow control loop). However, the goals for these closed controls are not just set by the Tenant Consumer Intent-driven Service but also by the CSP. The CSP sets goals for aggregate Autonomous domain tenant service operational behavior—for example, resource usage optimization and optimization of the trade-off between individual tenant service types and instances.*

- *Decoupled Operational layers supporting Operational Flexibility*

*Current APIs provide very specific information on the type of service required. Due to this, the Service Provider has to either render a service meeting exact requirements ( even if it means inefficient leverage of resource pool ) or reject the service (even though the service requestor might have found slightly impaired service acceptable ) or provide the best-effort service ( the service requestor may not find the service acceptable ). For communication services to be leveraged for mission/business-critical services, the services must be rendered cost-effectively. The requestor can express tolerance boundaries in different dimensions and the Service Provider can unambiguously understand and run the service within those boundaries.*

## 1.6 Key Concepts

Figure 1-4 Key Autonomous Network Concepts shows that the Autonomous Domain is the central concept for Autonomous Networks. The Autonomous Domain concepts support the technical evolution from traditional Network management approaches to AI-enabled Autonomous Networks.



**Figure 1-4 Key Autonomous Network Concepts**

Two key actors are closely involved in the interaction with Autonomous Networks:

- *The Tenant Customer: sets the goal for the autonomous network. The goal describes the tenant customer requirements including their SLA requirements.*
- *The CSP Operation Supervisor: Operations is responsible for the control and supervision of the aggregate behavior of the AD Business Services across all tenants, set operational objectives such as resource usage optimization and tenant service requirements via intent-driven service. The CSP Operations Supervisor sets the level of autonomy.*

The intentions of these two actors are not the same due to different knowledge and responsibility and conflicts may arise that become more severe as the actual service rendering services become constrained. The Control Loop should have the capability to detect these conflicts and the ability to resolve conflicts. In cases where the conflict cannot be resolved, it is necessary to notify the relevant actors for further human intervention (i.e., escalation).

Autonomous Domains may be composed of one or more closed loops, also bind the closed control loops used to realize the Intent-driven service to the supporting resources according to defined supervision and control objectives. The control loop can be used across multiple domains, a bigger Autonomous Domains can be constructed by aggregating multiple smaller autonomous domains together. These and other concepts are described later in this document. A metamodel (using UML notation) for all the concepts provides a formal unpinning of how Autonomous Domains are defined, linked, and automated (see appendix).

## 2 Autonomy Concepts and Principles

### 2.1 Automatic, Autonomic, Autonomous

The following section outlines the concepts and principles of what mean by the term *Autonomy* and having autonomy (i.e., *autonomous*). We explore the dynamic and dialogue between the Human (arguably still the most important part of the Autonomous Network) and the Machine (the AN). We examine the interaction between Machines and the Humans who trust them, delegate to them, and in the case of semi-autonomous networks, the humans who operate them. In the Readers Guide, we provided a short definition of some of the key terminology. To recap, we revisit these definitions in the table below.

**Table 3 The 4 A's**

Autonomy	The capability to make decisions free from human control.
Automatic	Able to operate independently of human control
Autonomous	Having autonomy.
Autonomic	Acting or occurring involuntarily. To respond without thinking.

*Automatic* means working by itself with little or no direct human control. An automatic task is not manually performed as no humans are involved.

*Autonomic* means acting involuntarily or unconsciously; like a humans' autonomic nervous system. Therefore, an autonomic function is one performed "without thinking", no cognition, or no policy involved.

*Autonomous* means having the freedom to act independently. Having the freedom to govern itself or control its own affairs.

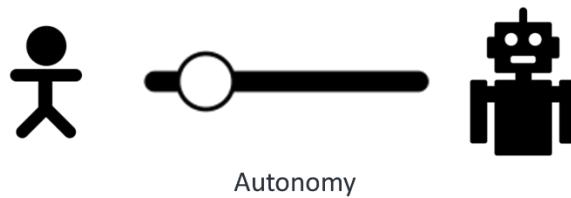
A simple way to distinguish between "autonomous" and "automated" is to examine the amount of adaptation, learning, and decision-making that is integrated into the system. Automated systems typically run within a well-defined set of parameters and are very restricted in what tasks they can perform. The decisions made or actions taken by an automated system are based on predefined heuristics.

Networks will evolve from simple automation to autonomic control and ultimately to autonomous behavior. The sophistication of self-management capability will also increase as these autonomy level increases, such as a progression from self-healing to self-optimization and then to self-governing abilities. Overall, the capabilities will evolve from "doing", to "responding" and "deciding". It is only in the decision-making cycle that the learning processes can be applied because it analyzes decisions taken and measure their effectiveness over time.

### 2.2 Principles of Human-Machine Interaction

If Automation in networks is about *doing*, then Autonomous Networks are about *learning-by-doing*. But who is doing the *teaching*? In autonomous systems up to Level 4, the teaching is performed by expert human operators. The systems are also learning by themselves, or self-learning based on experiential data. For example, machine learning on data will augment the current automation systems to enable capabilities such as self-healing, etc., but it is the human intelligence and human knowledge that effectively 'teaches' the AN. This balance of self-learning versus directed learning at hints the new duality of humans and machines in networks of the future. The following section explores Human-Machine interactions, the roles, and the delicate balance they must strike to enable trustworthy yet independent and autonomous systems.

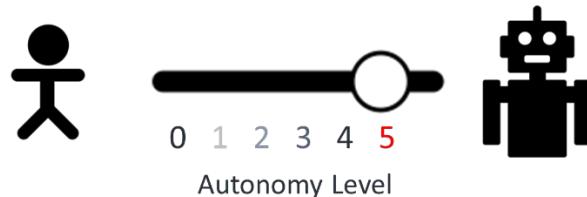
### Autonomy is a Continuum



**Figure 2-1 Slider of Autonomy**

The concept of “sliding autonomy” or adjustable autonomy attempts to dynamically merge the good qualities of both humans and machines (robots) into a single system (AN) [Brookshire]. Figure 2-1 illustrates the so-called “slider” of autonomy where control and decision-making within the autonomous system are decided by adjusting an overall level of autonomy.

Individual components of the autonomous network may operate at discrete levels of autonomy (Figure 2-2) and these *levels* form the basis for determining how mature are the autonomic capabilities of the autonomous network (more on levels in section AN Levels )



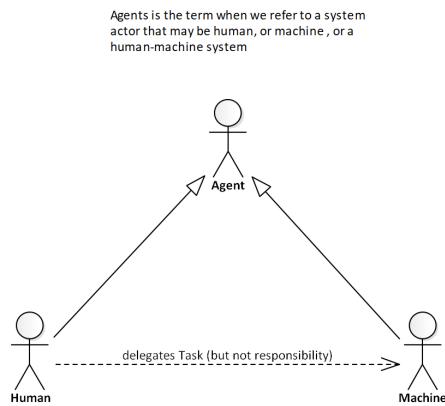
**Figure 2-2 Slider Control sets the Autonomy Level (0-5)**

### Humans are part of AN

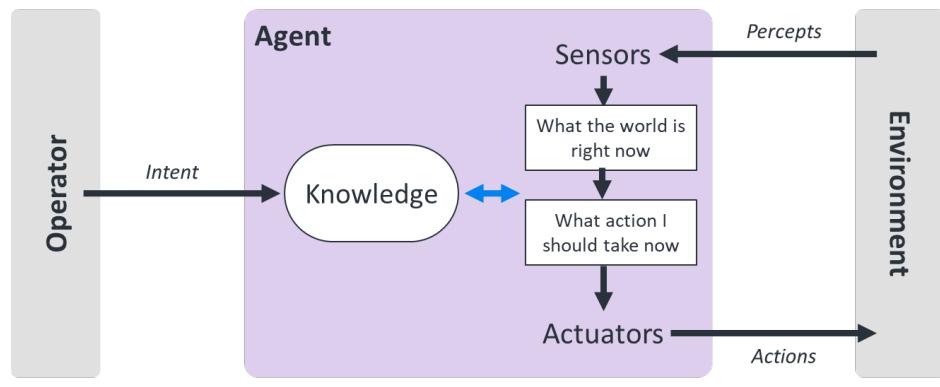
If we consider that achieving Level 5 Autonomy is quite some time away and that for regulatory reasons alone it may well be expedient to keep at least some humans involved. Network operations people need not worry about automating themselves out of the picture. Their roles will change significantly, and the skillsets needed will transform in a short space of time. Mundane automation work will be handed off to autonomous systems, but the business side will always need people to oversee the network and design and monitor the new generation of autonomous systems. AN will still require people who understand how the network works to *teach* or train the artificial intelligence itself while providing oversight and algorithmic tweaking. In reality, humans need to stay ‘in the loop’ as automated systems take on more intelligent tasks. Machines may not be trained to handle all unexpected situations; humans need to remain in the control loop to take care of such situations. The machine will learn and improve its knowledge over time, and one day may become self-sufficient in some areas.

### Humans and Machines are Agents

A human agent has eyes, ears, and other organs that act as *sensors* with hands, legs, mouth, and other body parts acting as *actuators*.

**Figure 2-3 Humans and Machines are Agents**

Likewise, a Machine agent such as those used in Autonomous networks use virtual and physical sensors and actuators, and this equivalence of humans and machines sensing and effecting their environment is important for us to understand how both can work together to achieve outcomes.

**Figure 2-4 Intelligent Agents**

The operator defines business intent as the only input to the intelligent system that implements the intelligent agent (see Figure 2-4).

- Business intent can change very frequently: for example, a dynamic SLA.

The environment is delivering data about its state, and it is changed by actions.

- The environment is dynamically changing by a huge number of factors.

The Zero-touch intelligent control system, or *Agent*:

- *Perceives the environment*
- *Has knowledge about the intent*
- *Has knowledge about the world and the relationships and processes within*
- *Is capable of thinking: Reason, Plan, Evaluate, Predict, ...*
- *Dispatches actions towards the environment*

### Human delegates Responsibility to Machine but not Accountability

The use case diagram (Figure 2-5) illustrates the relationship between Humans and Machines in terms of sharing the overall task burden if not the entire responsibility for the role. The human delegates the task

execution to the machine. The machine may use AI/ML to assist in its execution; it may even have cognition of the task at hand, but ultimately the responsibility lies with the human. The *principal-agent*<sup>4</sup> problem recognizes that responsibility, when given to a machine, lies with the *delegator*, in this case, the human even though the "Task" is performed by the machine. Hence, humans delegate tasks to machines but not responsibility. Accountability is a different but related concept and is now closely tied to the concept of governance. Put more simply, accountability is about the ability to report on or explain your actions. As such, accountability happens *after* the task is executed.

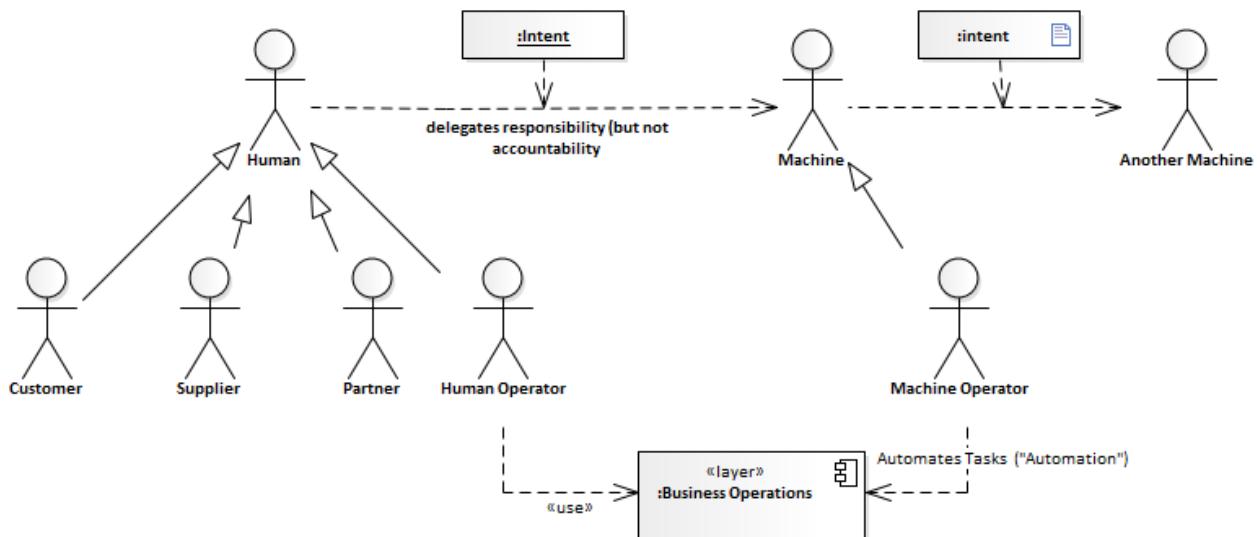


Figure 2-5 Human delegates Responsibility to Machine but not Accountability

### Machine is a Party, Decision Maker is a PartyRole

The class diagram illustrates how we can use the existing TM Forum Information Framework (SID) concepts and patterns of *Party* and *PartyRole* to model Humans and Machines [SID]. It serves also to disambiguate the roles that a human and machine may perform. What is important to note is that the roles performed remain the same regardless of whether a human or machine performs them.

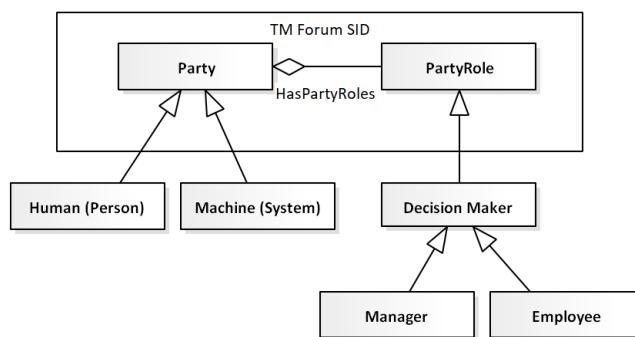


Figure 2-6 Machine is a Party, Decision Maker is a PartyRole

<sup>4</sup> [https://en.wikipedia.org/wiki/Principal-agent\\_problem](https://en.wikipedia.org/wiki/Principal-agent_problem)

## 2.3 Architectural Principles

### Future Mode of Network Operations is Human-Machine Collaboration

As AI continues to mature and play an increasingly important role in operations, network operations management will be shifting from the traditional "person + process" model to the new "Human + Machine" Collaboration Model.

In the "Human + Machine" Collaboration Model, machines assist operations by applying AI and machine learning (ML) algorithms to large data sets collected by operations support tools and business systems and simulating human thought processes and behavior, such as risk perception, pattern recognition, data analysis and interpretation, complex decision-making, and execution of elaborated tasks. The "Human + Machine" Collaboration Model provides operations management with AI and ML capabilities that free operational personnel from repetitive and routine tasks that machines can do even better than humans.

Under the new operating model, new technologies will greatly empower the traditional operations teams. The human's unique experience and judgment skills are being transformed into data and then injected into the knowledge base. The machine learning ability will form the closed loop of the networks' self-decision and self-execution capabilities. As a result, the convergence of human insight and machine learning will offer unlimited possibilities for network operations. Below, the figure shows how the role of person and machine is evolving through different development stages of the intelligent operations system transformation.

### Autonomous Networks are composed of platforms with different Autonomy Levels

The earliest developed Autonomous Networks will be assembled from platforms with varying degrees or levels of autonomy support. Many Autonomous Domains may be Level 3 for example, while others may achieve Level 4 and only require some basic human supervision. Some operational platforms will control service lifecycles at Level 3 while other services may be needed to achieve Level 3 due to the need for "human-in-the-loop" processes for that particular service.

As such, the dissonance or "friction" generated by the disparity of autonomy levels is a significant challenge to overcome for the viability of Autonomous Networks. Does a drop-off in autonomy level in one platform cause the overall autonomy level of an Autonomous Domain to degrade to the lowest common denominator? These autonomy level and their evaluations are discussed in more detail in Section 4.

### Trust and Explainability

Trust is established between a Human and a Machine. Trust, therefore, exists between the Trustor and the Trustee. Trust can be achieved by providing consistency of results over a long period (See [IG1229] –for more on Trust principles). If Autonomous Networks depend on trust, then explainable AI applications in telecom networks become very desirable and even necessary. Currently, AI is mainly to assist decision-making, while configurations still need manual operation. For example, in an energy-saving scenario in the data center, AI reasoning output to shut down a hundred servers, the operator needs to know: are there any critical applications running on the hundred servers? What factors lead to the result of this reasoning? What are the consequences of performing this operation? The reasoning process of the existing AI algorithm is black-box, which cannot gain sufficient *trust* in the result.

Secondly, in the future, systems can take action without manual confirmation. But we will still need to know the reason for the action. For example, why did the system shut down particular devices? To do a review or especially an audit, we also need to know the evidence for the decision. There are three main approaches in the industry to solve this problem:

- 1) Optimize the existing algorithms for explainability.
- 2) Introduce new and explainable methods of AI learning and reasoning, such as Symbolic Learning.
- 3) Adding the explainable modules in the original model to simulate the behavior.

One example of why telecom needs explainable AI is in the area of Network configuration where configuration changes need manual execution. The operators need to know the reason before they execute the configuration change. In the future, an autonomous system can take action without manual confirmation: e.g., if one access node does not work, the system can detect the failure and share the node service to other nodes.

Some of the main challenges around explainability include:

1. **The need for a valid interpretation study method:** for example, how to effectively modify the original classifier and how to quickly obtain the interpretation data set.
2. **The need to create a transparent and understandable model instead of the original unexplainable model:** By transparent, we mean an increased understanding of how AI was created and deployed. The scope of solutions is for further study but may include mask constraints, limit the complexity of the classifier, modify the classifier structure to have both the depth of learning and the intelligibility of the decision tree.
3. **Behavior Simulation:** Adding the explainable modules in the original model to simulate the behavior, to achieve explainability.

## AI Safety

For AI to be broadly adopted into an autonomous network it must be safe. “Safe AI” or “AI safety” can be defined as the way in which AI is deployed and used so as to not harm humanity, be fair and equitable to all, without bias, and generally be a positive influence on consumers and businesses alike. While businesses and algorithm designers may set out with those principles in mind, the optimizing nature of AI algorithms can have unintended consequences. Figure 2-7 outlines the four quadrants of safety.

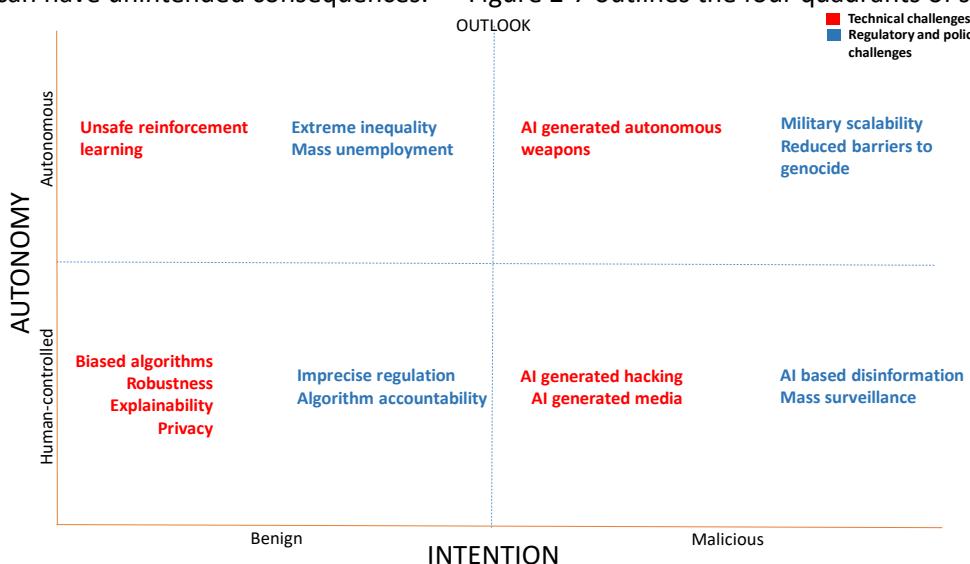


Figure 2-7 AI Safety Quadrants

The regulatory and policy aspects of AI are not the scope of this work, and thus the focus is on what technical strategies can help with AI safety for these quadrants:

- Autonomous learning, benign intent

By definition, autonomous learning agents learn “on the fly”. Thus, it is impossible to precisely know how they will behave when deployed, which presents numerous safety concerns:

- 1 *Unsafe exploration: the agent performs potentially dangerous actions in trial-and-error searches for optimal behavior.*
- 2 *Reward hacking: the agent exploits unintended optima of a naively specified reward function.*
- 3 *Negative side effects: to achieve a specified goal optimally, the agent causes other undesirable outcomes.*
- 4 *Unsafe interrupt: the agent learns to avoid human interruptions that interfere with the maximization of specified rewards.*
- 5 *Absent supervisor: the agent learns to alter behavior according to the presence or absence of a supervisor that controls rewards. This issue has become quite prevalent in the chatbot space and is the reason many companies currently disable their chatbots outside core business hours.*

- *Human-controlled, benign intent*

Examples in this category include supervised algorithms, such as classifiers, that are trained at a point in time, and then deployed to generate business value (decision making, action-taking, etc.). Despite the best intentions of the engineers involved in creating the algorithm AI safety concerns may still arise. It may perform very well on datasets used in the production process, but still may not be fit for purpose for the following reasons:

- 1 *Lack of robustness: despite performing well on test data, algorithms may perform very differently on other data.*
- 2 *Bias: well-trained robust algorithms can still reflect biases that exist in the training data or biases that arise in feature selection.*
- 3 *Privacy handling: if the deployment of an algorithm exposes information about members of the training set, sensitive information can be gathered directly from the model even without the release of training data.*
- 4 *Explainability: easily interpretable AI algorithms are key, lack of explainability means algorithms can't be rationalized or audited.*

- *Human-controlled, malicious intent*

Similar to the cybersecurity space, there will, of course, be malicious usage of AI that needs assessment and counter-strategies. Defensive solutions to these problems are likely to be both technical and policy-based.

## **Security by design**

The cybersecurity industry follows a predictable pattern: attackers find a way to penetrate a security system that was once heralded as impenetrable. The security breach is eventually discovered, the vulnerability is addressed, and attackers subsequently look to find a new way in. This pattern is unlikely to change with the only possible change being that the cycle times will continue to be reduced. With this in mind, from an autonomous network security standpoint, the goal is not to make it impossible for criminals to access critical information, as this is a task that cannot be accomplished with any degree of certainty. Rather, a more pragmatic and achievable target is:

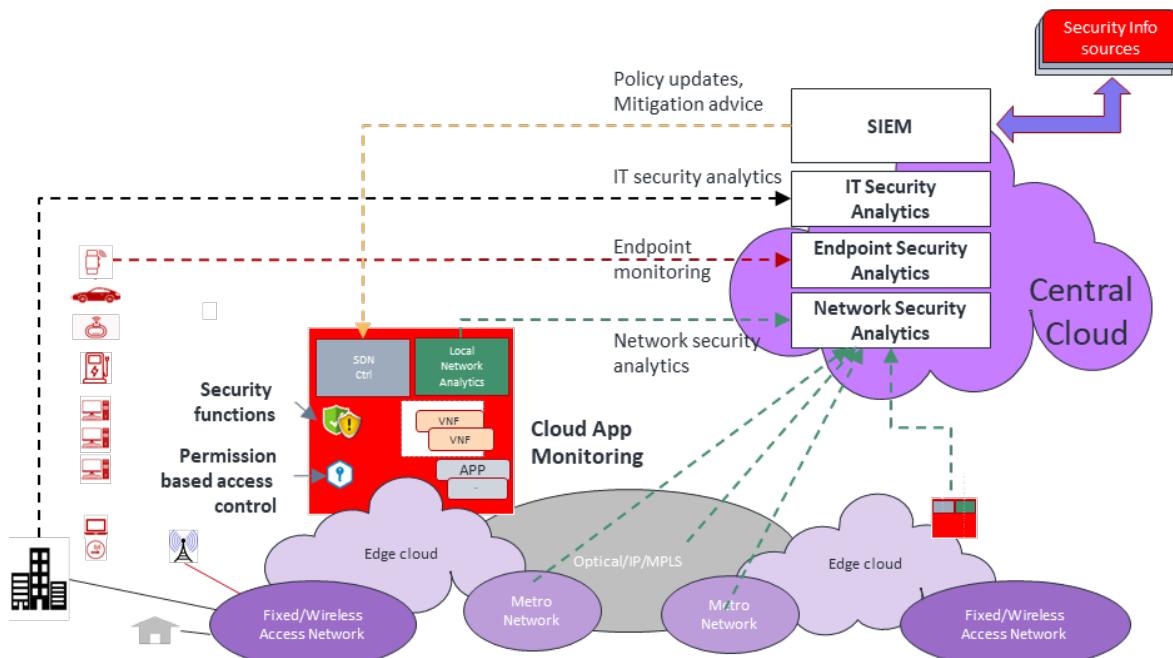
*The autonomous network should make the cost of illegally accessing sensitive information much higher than the value of the information to the attackers.*

Security assessment and enforcement has, over time, shifted from a predominantly manual activity into dynamic risk management activity, where continuous situational security awareness is complemented with proactive and reactive capabilities. This requires new models, systems and methods capable of capturing and measuring:

- *dynamic models of the state of IT and networking systems*

- threat intelligence information that is continuously gathered from both internal and external sources
- business objectives, or intent (which could be driven by customers, partners, industry regulators, competitive forces, or other)

The combined use of endpoint, perimeter, and network-based security is required to protect the data and applications of a network. An end-to-end solution is required, and a variety of functions (and supporting technology) are needed, a high-level depiction for which is provided in Figure 2-8. From an autonomy perspective, end-to-end security should be an autonomous domain (described in section 3.3).



**Figure 2-8 Security infrastructure for autonomous network**

The following functions represent the key functional blocks for end-to-end security within an autonomous network architecture:

**Endpoint monitoring** – this will be deployed on all devices with sufficient embedded intelligence, from cars, electric vehicle charging units, parking meters to door locks, vacuum cleaners, and washing machines.

**Cloud application monitoring** – all applications running in the cloud will be monitored for security purposes.

**Permission-based access control** – over time biometric authentication will be used to improve the authentication of users.

**Security proxies with payload traffic analytic capabilities** – The increased use of end-to-end encryption is an attempt to protect the privacy of user data in transit between two points. While effective this makes it very difficult for security tools to effectively monitor networks for evidence of infiltration or, for example, compromised devices. This creates an ironic contradiction: the mechanisms consumers and enterprises are using to protect data and privacy are putting information at greater risk. Thus, security proxies must support protocols offering flexible levels of encryption to enable visibility into some payload traffic, for QoS purposes, as well as security and law enforcement.

**Security analytics** — security analytics must be deployed on a scale capable of capturing and storing virtually all network activity among all devices, including IoT devices. This will enable both real-time threat detection and mitigation, as well as forensic analysis. Such analytics will typically be deployed as virtualized functions in enterprise data centers, as well as in central and local edge clouds.

**Security information and event management systems (SIEMs)** — These platforms support both security information management and event management. They provide real-time analysis of security alerts generated by applications and network functions. Many of the products<sup>5</sup> on the marketplace today have evolved into security autonomic platforms that rapidly activate or modify security policies automatically based on the perceived threat.

**SDN (Software-defined networking)** — SDN is a key network technology that provides dynamic flexibility to allow the network to rapidly and automatically adapt to security threats. The combination of SDN and SIEMs together with security analytics will result in the overall orchestration of cybersecurity defense.

**Ecosystem support for threat intelligence and security expertise** — This will involve network and cloud service providers and enterprises working closely with government and international agencies to ensure the protection of data and the infrastructure that depends on it. An autonomous network should support all such engagements.

TM Forum's technical report TR263D addresses security and policy management in Hybrid Infrastructure Platforms (HIP) [TR263D].

---

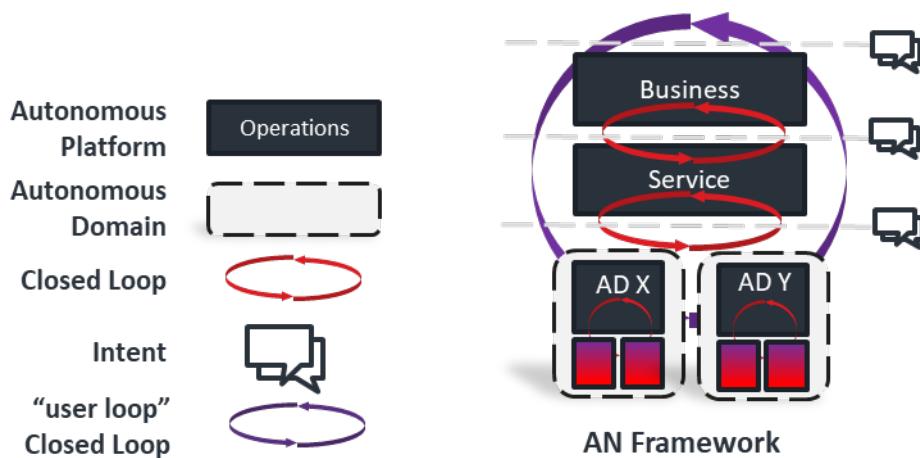
<sup>5</sup> Contemporary examples: Splunk Phantom, FireEye Security Orchestrator, IBM SOAR, Palo Alto Networks Cortex XDR

© TM Forum 2021. All Rights Reserved.

## 3 Autonomous Network Technical Architecture

### 3.1 AN Framework Overview

The key technologies enabling the AN transformation are intent-driven interfaces/closed loop automation/Intelligence-driven decision making (AI/ML/Knowledge). The holistic method to reduce complexity is decomposing the network into multiple self-governing domains that only interact with other layers/domains/users via intent-driven interfaces (API or Human interaction). The self-governance or autonomy of the domains is achieved via carefully coordinated (orchestrated) closed loops and intelligence takes an important role to enable parts or even entire processes of the closed loop via intent.



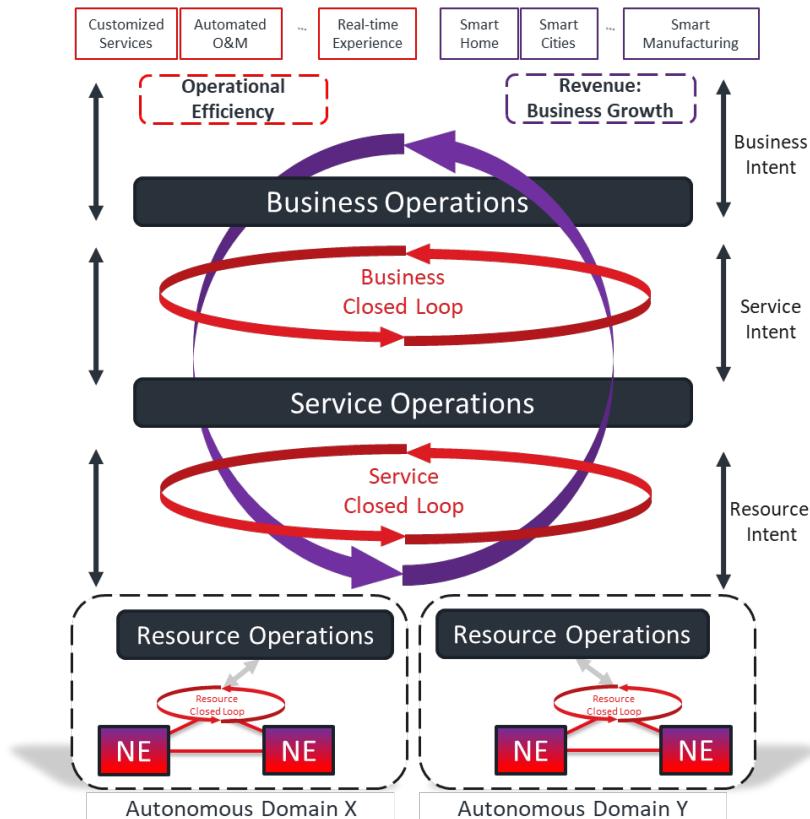
**Figure 3-1 Building Blocks of Framework**

The goal of the AN Technical Architecture is to define a minimal set of technical concepts and principles to realize the requirements described in the Autonomous Networks Framework summarized below.

The idea is to define and realize Autonomous Domains that are self-contained, self-managing, self-optimizing, and self-healing, operating with autonomous capabilities and capable of implementation using different internal technologies. One key requirement is to have Autonomous Domains and control loops operating across different scopes and different business functions. Figure 3-2 TM Forum Autonomous Network Framework shows a framework of composable domains that span ecosystems, customers, products, services, and resources –comprising networks computing and software.

Enablement of these solutions permits more advanced arrangements of interacting Autonomous Domains to be constructed from simple Autonomous Domains each following a standardized integration pattern described in 5.2.1. This requires the AN Technical Architecture model to be simple, self-similar at multiple levels of granularity and scope, and capable of composition into more complex arrangements. A precise definition of these concepts is needed in a formal metamodel to capture configurations of these concepts in a machine-readable and “processable” format suitable for automated design, configuration, and deployment.

At the heart of AN is the idea of intelligent software that continuously make decisions about what to do next while operating in a dynamic, heterogeneous network environment. Such decisions are taken autonomously, or with some degree (or formally ‘Level’) of autonomy, that depends on the influence /feedback (also termed ‘experience’) from human operators or other software systems (both traditional and autonomous).



**Figure 3-2 TM Forum Autonomous Network Framework**

The AN framework positions the intelligent systems across all 3 operational layers and encapsulates decision-making using closed control loops within Autonomous Domains (Principles of Autonomy, Modularity) and also collaborative (or federative) autonomy using cross-layer control loops using intent-driven interfaces and intelligence and knowledge based interfaces. This chapter of the technical architecture will describe the operational layers and the concept of autonomous domain and explore some integration patterns. Subsequent chapters will explore the concepts of Levels, Domains, Intent and Control loops, etc., in greater detail.

### 3.2 Core Concepts

The four fundamental concepts introduced in this AN Technical Architecture to support the AN objectives are:

- **Autonomous Domains.**
- **Objects** (or Managed Entities) and their taxonomy.
- **Business Services** that are exposed by Autonomous Domains are based on the notion of **Intent-driven Service** definitions. Business Services exposed by one domain may also satisfy dependencies from other Autonomous Domains. Autonomous domains may expose different Business Services to different actors including Tenant/ consumer and Operations Supervisory roles. Each Business Service may have a different lifecycle for the information (Knowledge Objects) that they expose.
- Support of (closed) **Control Loops** within Autonomous Domains and across Autonomous Domains.

Together these deceptively simple formally defined concepts when implemented allow sets of interconnected Autonomous Domains to exhibit controlled, dynamic and responsive behavior at a scale and velocity not achievable using traditional systems' management concepts. These agile characteristics are essential for the operational management of the coming generation of applications, services, and

management based on virtualization and cloud-based Communication Services and Networks. This is the communications industry equivalent to the aerospace 'fly-by-wire' challenge. The adaptive behavior of Autonomous Domains can be supervised and controlled through formally defined delegated responsibilities operating to policies authorized by other Autonomous Domains.

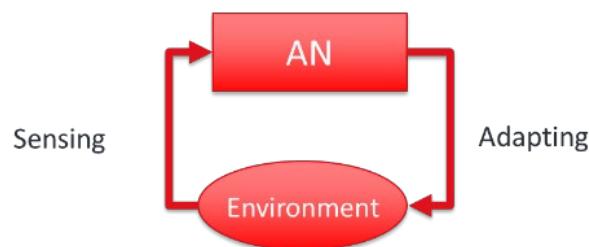
### Autonomous Network

The AN Business Requirements guide [IG1218] describes what we expect an Autonomous Network should provide:

- *Provide a Zero-X experience to end-users (zero-wait, zero-touch, zero-trouble, zero-trust, zero-friction)*
- *Supports Self-X management capabilities that enable the Zero-X experience*
- *And comprise of simplified network architecture, virtualized components, intelligent agents and decision engines to automate business/network operations.*

This leads us to a more functional definition as follows. An Autonomous Network is

*A set of network and software platforms that are capable of sensing its environment and adapting its behavior accordingly with little or no human input*



**Figure 3-3 AN can sense and adapt to its environment**

Figure 3-3, above illustrates the closed loop nature of this definition. After the AN senses the environment it can anticipate its next action and it can learn from its previous taken decisions. This is how the AN becomes adaptive rather than autonomic or reactive. Autonomous networks are self-governing systems that can configure, heal, and optimize themselves and are capable of making decisions with little or no human direction or supervision. They can adapt to their environment, optimizing and personalizing the end-user experience of the services they provide. Autonomous Networks result in networks that cost less to operate, with overall improved agility, security, and resiliency. It is a system that can measure, analyze, and control itself continuously and efficiently, driven by business objectives. These business objectives are specified using SLA and Intent and underpinned with trust. Put even more simply, an autonomous network is one that can *measure, analyze, and control itself* continuously<sup>6</sup>.

### An Autonomous Network

- has the freedom to act independently, govern itself, and control its behavior with no external humans or machine acting upon it.
- consists of a simplified network architecture, virtualized components, automating agents, intelligent decision engines.
- presents composable capabilities to create intelligent business and network operations.
- can self-configure, monitor, correct (self-repair), defend and analyze, resulting in a network that costs less to operate, while increasing agility, security, and resiliency.

---

<sup>6</sup> This paraphrases work from Kelleher et al in “Empowering Self-Driving Networks” [KELL]

3GPP defines an Autonomous Network as a “telecommunication system (including management system and network) with autonomy capabilities which is able to be governed by itself with minimal to no human intervention”. Given that definition, how do we build such a network and how do we deploy them? The following sections will describe some of the main building blocks required.

### 3.3 Autonomous Domain

This concept defines the scope of autonomous behavior delegated by an organization and is the basic building block of AN Technical Architecture. An Autonomous Domain is defined as:

*An administrative governance boundary (Management Domain) that defines the scope of delegated autonomous behaviors.*

It exposes a set of Business Services based on the capabilities of a set of objects over which the AD exercises governance ( policy, rules, patterns including closed control loops), Is self-managing self-scaling and manages its own lifecycle and publishes its Service Capabilities and Business Services in a Catalog / Repository.

Objects can include Intent Services, management interfaces, (sub) Domain managers, Control loops, controllers, orchestrators, decision-makers, AI analytics, monitors, policies, etc.

#### **Operational Layers are decoupled (Business, Service, Resource)**

The AN framework uses a layered architecture pattern, where the architecture is stratified into three layers - a business operations layer, a service operations layer, and a resource operations layer. Each layer runs in self-operating mode and hides the details of domain implementation, operations, and the functions within the domain to the consumer.

A prerequisite of an autonomous architecture is the efficient separation of the “operating regions” for the control loops. This can be achieved by decoupling control systems and autonomous platforms into these Autonomous Domains thereby ensuring that they control “different independent outputs based on independent inputs”, as defined in ETSI GANA.

#### **Business Services**

GB986 -CP Frameworks Metamodel has the formal definition of Business Service as :

*A Business Service exposes the functionality of Business Roles or Collaborations to their environment.*

*This functionality is accessed through one or more Business Interfaces. It may access Business Objects.*

*A Business Service is defined as the externally visible (“logical”) functionality, which is meaningful to the environment and is realized by business behavior (Business Process, Business Function, or Business Interaction).*

*A Business Service exposes the functionality of Business Roles or Collaborations to their environment.*

*This functionality is accessed through one or more business interfaces. A Business Service is realized by one or more Business Processes, Business Functions, or Business Interactions that are performed by the Business Roles or Business Collaborations, respectively.*

*The name of a Business Service should preferably be a verb ending with “-ing”; e.g., “transaction processing”. Also, a name containing the word “service” may be used.*

This definition is preferred as it is supported by a precise definition of relationships with other Integration Framework concepts in particular those associated with multiple prior interface definitions: MTNM MTOSI MTOP OSS/J and can be used as Integration metamodel for open APIs and 3GPP IRP and other SDO management interfaces.

A **Business Service** is a service that is delivered to business customers by business units. For example, delivery of financial services to customers of a bank, or goods to the customers of a retail store. Successful delivery of business services often depends on one or more IT services. A business service may consist almost entirely of an IT service – for example, an online banking service or an external website where product orders can be placed by business customers.

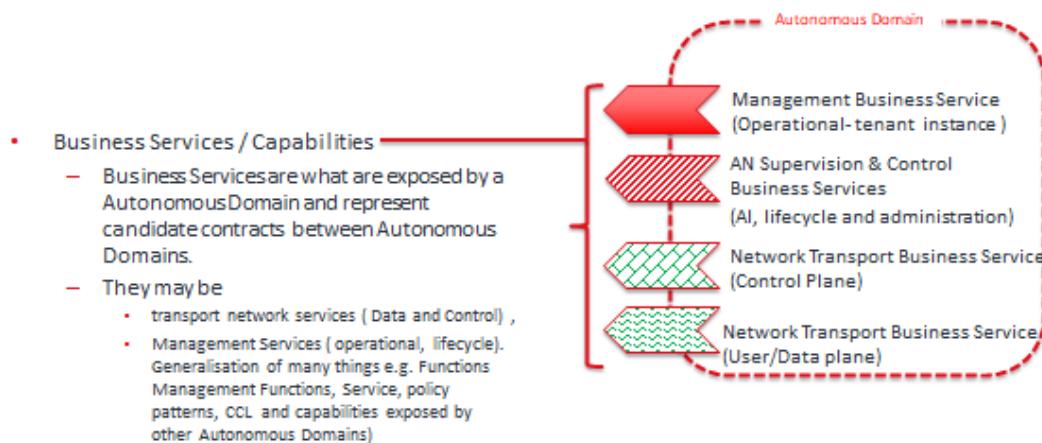
**TOGAF Business Service:** Represents a service provided by the business, which may then be realized by one or more IS services.

### Categories of Business Services

An Autonomous Domain may expose services of different types in part depending on which parts of the AN Business Architecture are being defined and realized.

## Autonomous Network Technical Architecture Concepts

### Business Service (Capability) Categories



© 2018 TM Forum | 6

**Figure 3-4 Autonomous Domain Business Service Categories**

The categories that have been identified are based on different roles consuming those service and include:

- **Management Business Service** (Operational processes focused on tenant instances) used by systems providing tenant instances of services ultimately to end consumers (retail and enterprise).
- **AN Supervision & Control Business Services** (lifecycle, AI, and administration) used by SP planners and developer roles.

For Autonomous Domains representing communication network resources additional business services are needed:

- Network Transport Business Service (Control Plane) to control plane application user roles.
- Network Transport Business Service (User/Data plane) to end-user application roles.

### AI-Enabled

AI is evolving from **perceptual intelligence to cognitive intelligence**. In the next 10 years, technologies such as neural networks, knowledge graphs, and domain migration will make it possible to achieve system autonomy in telecom networks. Combining AI with other technologies can significantly improve O&M efficiency. It can **replace manual operations** that were once required to solve large numbers of repeated and complex computing tasks in the telecom field. It can also improve telecom **network prevention and forecast capabilities** based on big data volumes.

While the training platform usually requires tremendous computing power, it is suitable to be **deployed centralized in the cloud or local cloud** infrastructure, then open the data channel with each AI-enabled management system.

While the inference works in a nearly online manner, it utilized the AI model rendered from the training platform, dealing with the inputted real-time data, giving analysis results in a short time, the result usually supporting decisions or provide a guide for further processes. Considering this online characteristic of the **inference**, they are normally **deployed in application layers**.

Nowadays, all operation/business **supporting systems** are all evolving to intelligence by using knowledge and inference in their own domains, even in the **NE layer**, in order to offload the huge data exchange efforts also enabling to have necessary analysis abilities, on the one hand, NE consuming local data to get conclusions, on the other hand, expecting offering less size but more valuable data to upper managing/controlling layer.

Then in the future networks, the AI capabilities will be deployed in layers:

- the cloud layer: training platform and knowledge center, providing model training service, and digitalized knowledge.
- the operation and management layer: model and inference application in specific domains.
- NE layers leverage built-in AI capabilities provided by devices.

### 3.4 AN Technical Architecture Diagram

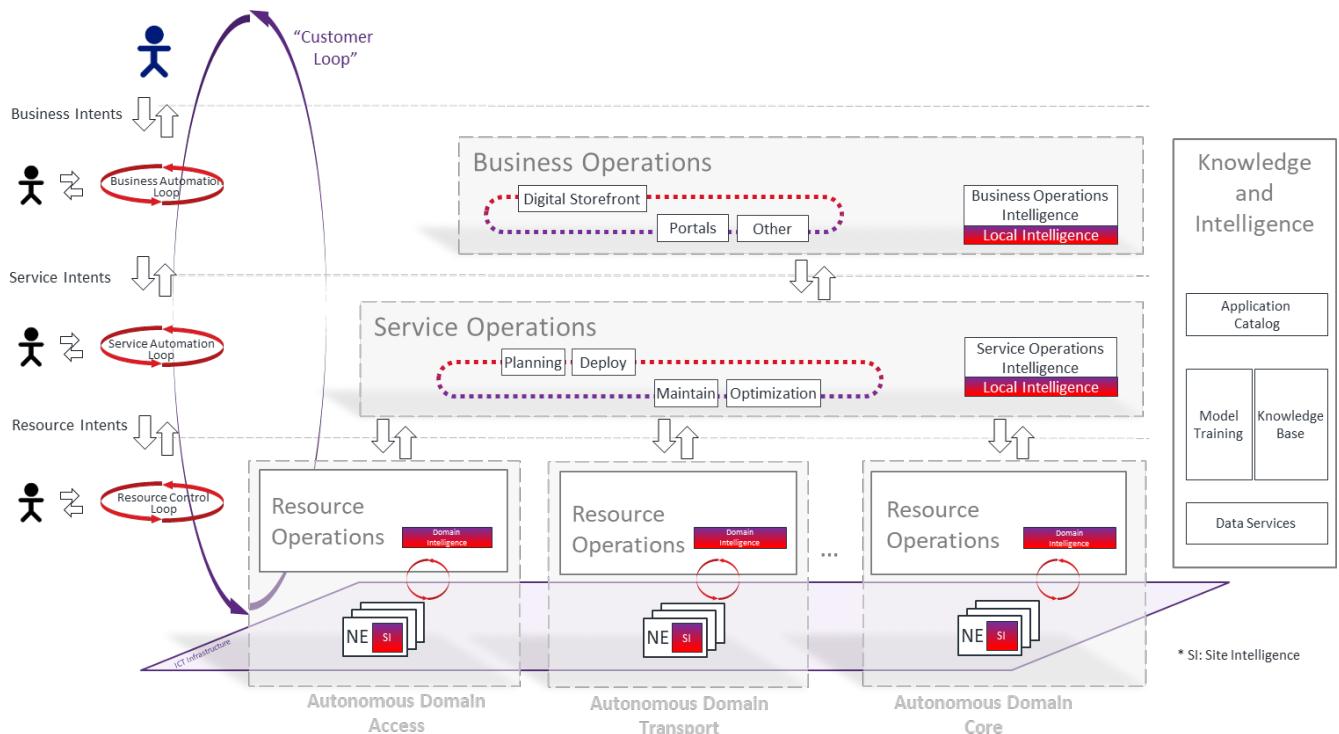


Figure 3-5 AN Technical Architecture

- The key technologies enabling the AN transformation are: intent-driven interfaces/closed loop automation/Intelligence-driven decision making (AI/ML/Knowledge).

- The holistic method to reduce complexity is decomposing the network into multiple self-governing domains that only interact with other layers/domains/users via intent-driven interfaces (API or Human interaction).
- The self-governance or autonomy of the domains is achieved via carefully coordinated (orchestrated) closed loops and Intelligence plays an important role to enable parts or even whole flows of the closed loop.

### AI-enabled

AI technology is used in many different platforms and systems, and different deployment scenarios. AI services are provided to support these systems including AI training, data lakes, network and service knowledge base, and an AI application market.

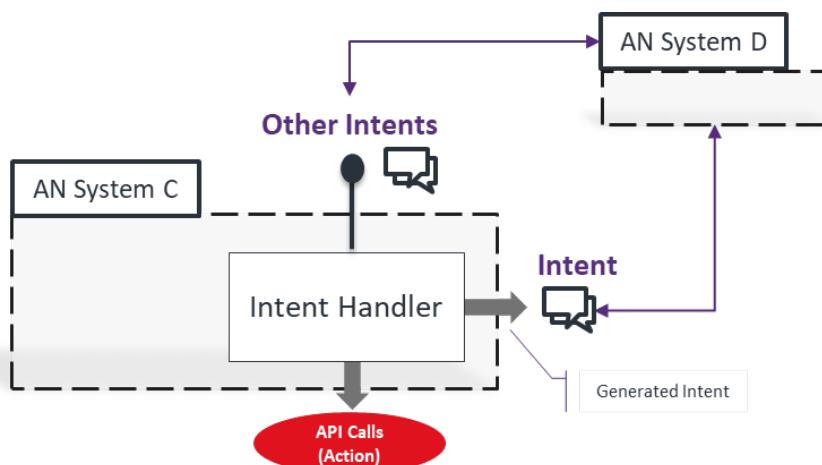
### Service Operations

Based on the E2E process of planning, construction, maintenance, and optimization, and three major capabilities of service collaboration, assurance, and analysis, achieve cross-vendor and cross-domain service-layer autonomy and control loop management.

### Resource Operations

Based on network management, control, and analysis capabilities, automatically translates upper-layer services and application intents into network behaviors, continuously ensures the SLA commitment of network connections or functions and implements autonomous control loop management of single-domain networks.

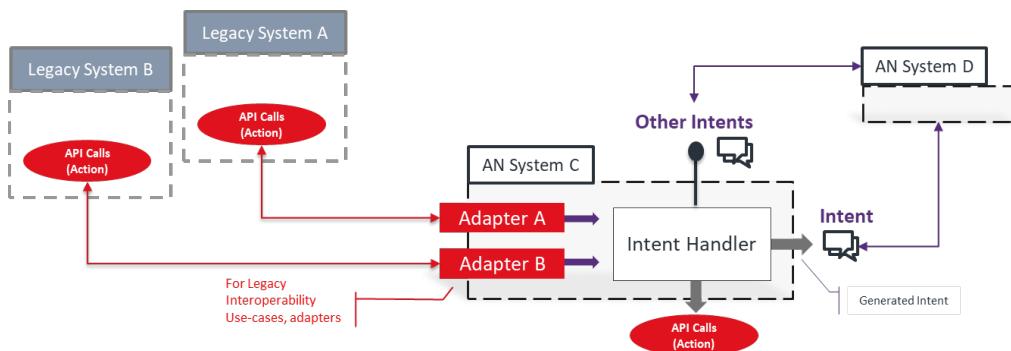
## 3.5 Patterns for AN to AN system Integration



**Figure 3-6 Intent interactions between 2 AN systems (domains or platforms)**

### 3.6 Patterns for AN to Non-AN System Integration

When discussing Intent-driven Networking (IBN) and Autonomous Network (AN) Systems we need to keep in mind the fact that not all systems within a Communication Service Provider's (CSP) stack will be swapped out and replaced with IBN supporting AN systems at the same time. There are likely to be transition periods where certain network, OSS or BSS systems don't support integration with ANs using Intent-driven interfaces.



**Figure 3-7 Adapter to Intent Integration Pattern**

To support this transition period, mechanisms are required that allow AN systems to interoperate with non-AN systems. Since it may be difficult or undesirable to modify non-AN systems (especially legacy systems) the AN system will need to be modified to support integration. This is in line with approaches used in other areas where interoperability between systems having different capabilities is a major challenge (e.g., ETSI ENI's "assisted" system approach [ENI005]).

To address this challenge, integration with non-AN systems requires special *adapters* to be provided in the AN system that can handle API requests from legacy systems. This includes both synchronous and asynchronous request patterns.

This approach is shown in Figure 3-7 with the following major components:

- *Legacy (non-AN) systems (Systems A and B) with potentially different integration APIs*
- *Integration Adapters within AN System (proposed one per non-AN system API)*
- *Intent Handler: A component that handles incoming intents and triggers actions (API calls) and intent calls to other AN systems (System C and D).*

#### Integration Adapters

The integration adapter is a critical component in the integration. We would need to examine each non-AN API integration point to derive specific requirements for the integration adapter. This is why we propose a one-to-one mapping between the non-AN system and the integration adapter. There may be common aspects of standardized APIs that could allow adapter reuse, but it may also be simpler to start with a disaggregated approach.

Mapping of requests to intents requires:

- *Ability to maintain request state which is synced with intent-driven state*
- *Ability to map requests to pre-created intents*
- *Ability to generate intents based on requests – a more difficult task but may be required for certain cases*
- *Generating a suitable response to non-AN requests from intent-driven internal state*

A point requiring further investigation is whether the adapters can call out to non-AN systems (i.e., request originates in the AN system but uses an adapter instead of an Action generated by the Intent Handler). To avoid overburdening the adapter function and to streamline flows all outbound requests from the AN system should pass through the Intent Handler unless we find use cases where that cannot be supported.

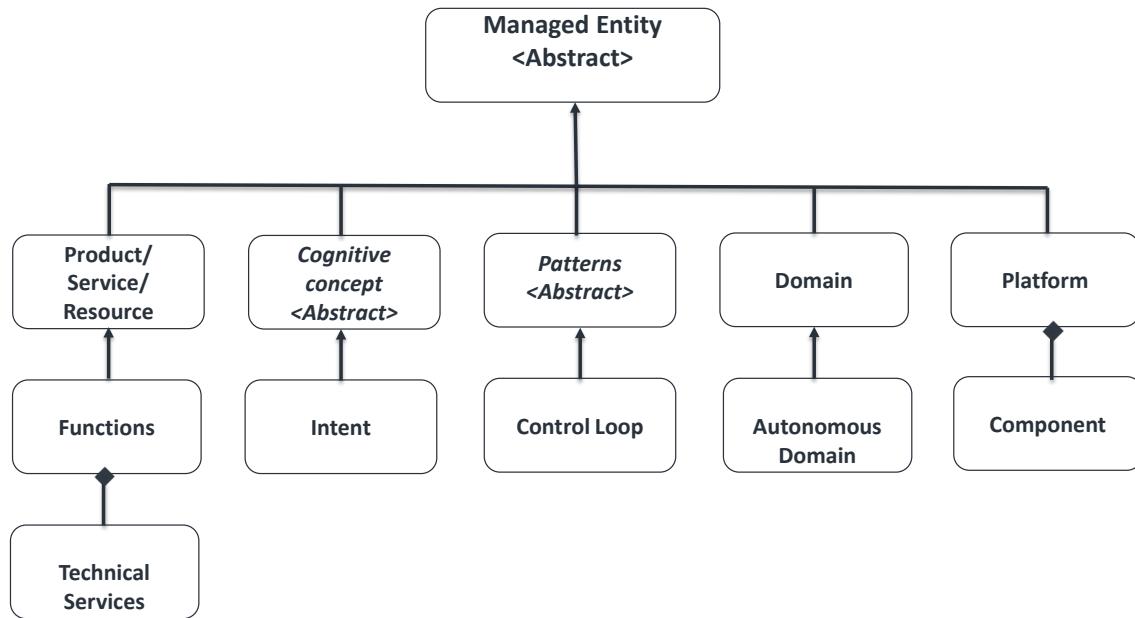
### Integration Adapters and the Slider of Automation

The “slider” of automation that can be used to manage the AN system will also impact the operation of the adapters. For example, during a planned outage (maintenance/upgrade scenario) we may wish to isolate non-AN systems that are not impacted to ensure their normal operation.

A topic for further investigation is how the adapters and Intent Handlers could impact the AN system as a whole. This requires some form of monitoring around the adapters (incoming relative to AN system). Similarly, monitoring is required around the Actions issued by the Intent Handler (outgoing relative to AN system). This is required to ensure the AN system autonomously disconnects from non-AN systems when it cannot reliably participate in the integration. There could be issues arising from degradation in the non-AN system (e.g., incoming requests are incorrect or outgoing Actions are correct but fail at a high rate) or in the AN system (e.g., incoming requests are valid but not processed correctly or outgoing Actions badly formed).

## 3.7 AN Architecture Model

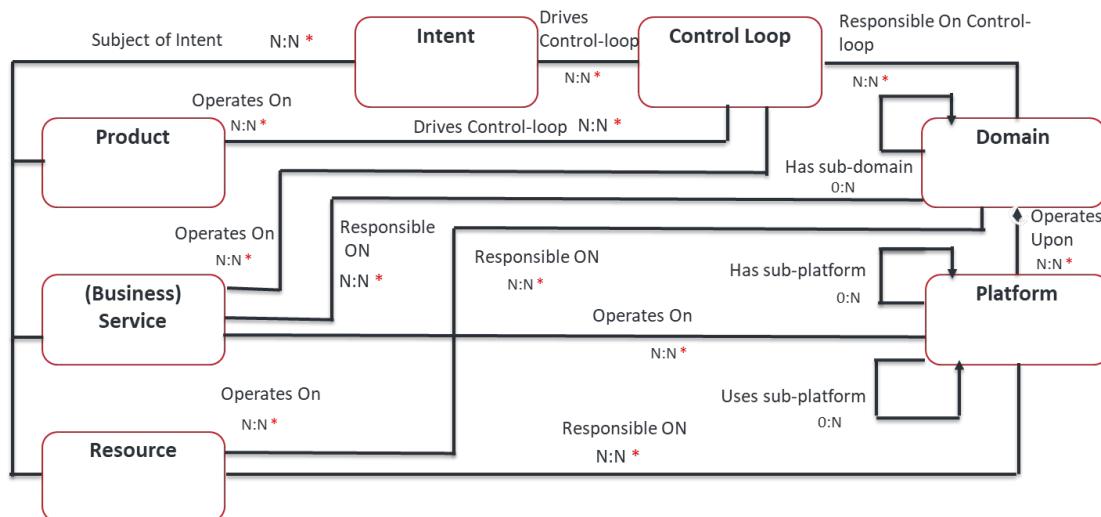
One of the key challenges in autonomic management is the alignment of concepts and models across the industry. We need the right *domain models* to support these new concepts. A unified model for the entities is significant for integration efforts and APIs that are going to be generated in this area. The SID part of TM Forum’s Framework is used as the foundation information model and is enhanced to support the Autonomic Management needs. As the concepts and business entities used within autonomy stretch beyond the remit of SID, the concept of a “metamodel” is being used. The metamodel is composed of a set of individual domain models that are used for specific areas and a top-level set of relationships among them can be looked at. Figure 3-8 shows the top-level entities that we believe are necessary for autonomous self-management. Some of them are well-established data models that the telecommunication industry is using for many years while others include new and evolving data entities that should take their place in the metamodel.

**Figure 3-8 Models of Managed Entities**

The models for products, services, and resources are well known and established. In recent years, several SDOs have looked at the modeling of Domains and Platforms. In other contexts, such as knowledge and cognitive automation, new models and patterns are needed. Each of the entities in the figure point to a specifically required information model. The abstract classes of a “Cognitive Concept” that includes Intent management and the “Pattern” that include the Control loop, mean that some more entities may be developed under these areas. The metamodel provides an overview different managed entities to form a complete holistic model needed for autonomous management. In general, the key data entities represent different areas of responsibility:

- The different types of business-related managed entities: Products, Services, and Resources are the subjects of management.
- The *Intent* concept is categorized under more abstract concepts such as cognition.
- The *Patterns* provide the new mechanisms that support the AN management, such as the control loop.
- *Domains* provide boundaries and areas of responsibilities of the different management roles, either human-based or machine-based
- *Platforms* are the systems that fulfill the different management roles.

The main relationships among these data models are described in the diagram below.

**Figure 3-9 Key Entities in model**

Please note that the abstract classes and the lower level entities were taken out of the diagram for readability. Additionally, some of the relationships that are described here represent multiple lower-level relationships w elaborated. The notation “N: N \*” is used to annotate that such an elaboration at a lower level exists. At this level, we can observe the key relationships:

- *The Business data entities: Product, Service, Resource may be “Subject of Intent”. At the lower level data model, this means that each one of them may own a set of Intents, while a higher level Intent may be applied on a set of such of these managed entities.*
- *Intents drive the control loops. Again, this depends on the level of the Intent. A high-level Intent may be translated to multiple control loops or a Control loop may be set with a specific Intent that it controls.*
- *The Control loop operates on the main Business data entities: Product, Service, and Resource.*
- *Now, going to boundaries and Domain Management, the Domain gets the responsibility on Business entities, such as: Products, Services and Resources.*
- *The Domain may be responsible for some of the Control loops, while multi-domain control loops are also possible, yet there is always a single Domain responsible for running a control loop, even if this is eventually delegated to multiple domains.*
- *The Platforms are the actual systems that execute the management activities, they operate upon the instructions and restrictions as provided by the Management Domains.*

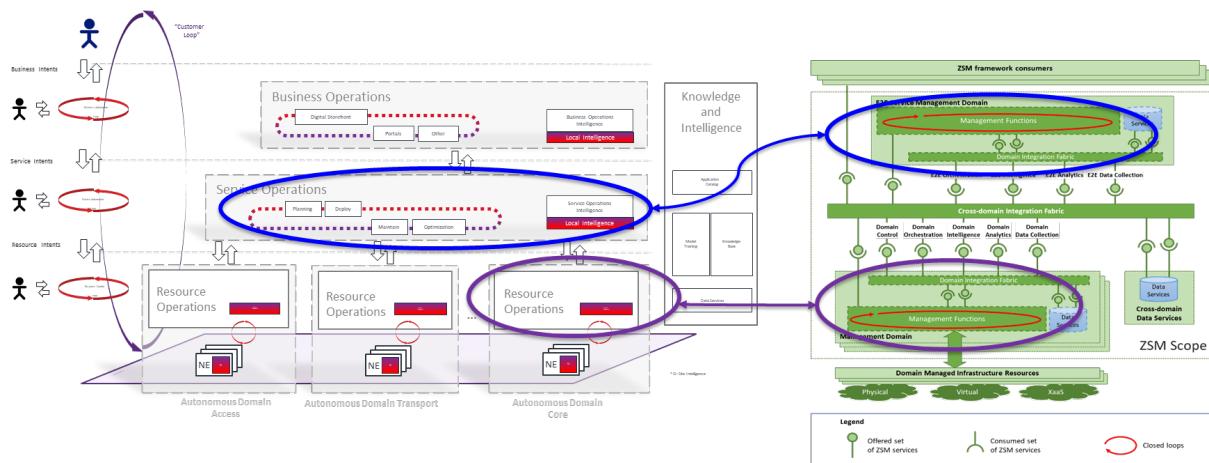
It is worth realizing that as described in the hierarchy view of the managed entities, most of these relationships are translated to associations at the level of functions and Platform components that are the parts of the Domains and Platforms respectively. The second next level of the model is beyond the scope of this current document and will be published in subsequent technical deliverables.

### 3.8 AN Technical Architecture using other Reference Architectures

TM Forum's existing assets and best practices covering business operation, service operation, and resource operation will be updated as we progress our understanding of AN.

SDOs with specific domain expertise are also studying on relevant areas of service/resource operations, especially network lifecycle operations. It is better to find a way to harmonize these study efforts, and take the advantages of each SDO, make sure the outputs could complement each other.

As an example of this harmonization, we introduce in Figure 3-10 the mapping between AN Technical Architecture and the ETSI ZSM Framework. Other mappings with relevant SDOs including open source communities will be elaborated upon in upcoming AN studies and SDO coordination activities.



**Figure 3-10 ETSI ZSM scope within AN Technical Architecture**

ETSI ZSM's E2E Service Management Domain is a management domain specialized to manage E2E services and corresponds to the AN Service Operations layer. The set or collection of Management Domain(s) correspond to AN's Resource Operations layers.

## 4 AN Levels

### 4.1 Levels of Autonomous Networks

The AN whitepaper proposed a harmonized classification system intending to provide a common understanding of network autonomy for autonomous networks. As such, this classification of levels builds a maturity model<sup>7</sup> to assess the level of automation achieved by an autonomous network:

- *Define the concept of autonomous networks*
- *Identify six levels of network automation from “no automation” to “full automation”.*
- *Base definitions and levels on functional aspects of technology.*
- *Describe categorical distinctions for a stepwise progression through the levels.*
- *Educate a wider community by clarifying for each level what role (if any) operators have in performing the dynamic network operations task while a network automation system is engaged.*

### 4.2 Benefits of Levels

In the previous section, we described the levels and explained the categories used to determine levels. In this section, we describe in greater detail how to leverage the 6 levels to perform assessments of autonomy support.

Similar to what the levels of driving automation means to the automotive industry [SAE J3016], a common understanding of the levels of the autonomous network would be helpful to the telecommunications industry. The operator's operational and management efforts and dedicated relevant resources will vary according to different autonomous network levels. It would be beneficial for operators to have a clear view on the expectation of the level of their network (s) to indicate the maturity of the network autonomy so that they could first focus on the important features which should be prioritized to achieve a certain level of autonomy in their networks. This approach will also help operators to smoothly migrate to a higher autonomy level and attain better operational efficiency.

Autonomous Network Levels have the following benefits:

- Providing an **evaluation basis for measuring the level** of an autonomous network along with its components and workflows.
- Providing reference for **gaps and priorities analysis** for standardization works on network autonomy.
- Providing guidance to operators, vendors and other participants of the telecommunications industry for **roadmap planning**.

---

<sup>7</sup> Please refer to **IG1252 Autonomous Network Levels Evaluation Methodology** [IG1252] for a more comprehensive description of the concepts, including Autonomous Networks Level methodology, operational processes, their underlying sub-processes and tasks and the evaluation approach for assessing an Autonomous Domain or Autonomous Network.

## 4.3 AN Level Classification

### AN Level

TM Forum defines AN Levels as follows:

*AN Levels describe the level of autonomic capability in a given operational workflow or for an autonomous domain (L0 to L5).*

*AN Levels identify contextual autonomous capability.*

AN Levels describes how well autonomy capabilities have been applied in the network management workflow. The participation of the human and systems in the network management workflow are important factors to evaluate the network autonomy level. For each AN level, those tasks that can be performed by the telecom system, those tasks to be performed by a human, and those tasks that are be performed by cooperation of human and telecom system needs to be clarified. As such, in the highest autonomy level, all tasks are performed by the system.

### AN Level Framework

The TM Forum AN Level Framework is used for evaluating the autonomous capability of the autonomous system or systems.

**Table 4 TM Forum AN Levels Framework**

Autonomous Levels	L0: Manual Operation & Maintenance	L1: Assisted Operation & Maintenance	L2: Partial Autonomous Networks	L3: Conditional Autonomous Networks	L4: High Autonomous Networks	L5: Full Autonomous Networks
<b>Execution</b>	P	P/S	S	S	S	S
<b>Awareness</b>	P	P/S	P/S	S	S	S
<b>Analysis</b>	P	P	P/S	P/S	S	S
<b>Decision</b>	P	P	P	P/S	S	S
<b>Intent/Experience†</b>	P	P	P	P	P/S	S
<b>Applicability</b>	N/A			Selected Scenarios		All Scenarios

P People (manual)      S Systems\* (autonomous)

\* **Note 1:** Systems including management system, O&M tools and network.

† **Note 2:** Intent/Experience is the group of tasks which translate intent from operator or customer into detailed operations which may affect one or more of the following groups of tasks (i.e., awareness, analysis, decision, execution), also evaluate and feedback intent fulfilment information (e.g., the intent is satisfied or not).

† **Note 3:** Intent/Experience Level rating does not mean that intent-driven approaches are only applicable at Level 4 and Level 5. It means that the expectation of a L4 system is that it can delegate and respond to tasks with a higher degree of abstraction. Intent driven interactions achieve this abstraction.

<b>Level 0 - Manual management</b>	The system delivers assisted monitoring capabilities, which means all dynamic tasks have to be executed manually.
	In Level 0 Zero-X business capabilities are not applicable.
<b>Level 1 - assisted management</b>	The system executes a certain repetitive sub-task based on pre-configured to increase execution efficiency.
	In Level 1 Zero-X business capabilities are only available to an individual element.

<b>Level 2 - Partial autonomous network</b>	The system enables closed loop O&M for certain units based on the AI model under certain external environments. In Level 2 Zero-X business capabilities are available to individual AN case but not be able to be linked up to a service.
<b>Level 3 - Conditional autonomous network</b>	Building on L2 capabilities, the system with awareness can sense real-time environmental changes, and in certain network domains, optimize and adjust itself to the external environment to enable intent-driven closed loop management.
	In Level 3 Zero-X business capabilities are available to select use cases that are able to be linked up to a service.
<b>Level 4 - High autonomous network</b>	Building on L3 capabilities, the system enables, in a more complicated cross-domain environment, analyze and make decisions based on predictive or active closed loop management of service and customer experience-driven networks.
	In Level 4 Zero X business capabilities are available to E2E full lifecycle operations of select services.
<b>Level 5 - Full autonomous network</b>	This level is the end-goal for telecom network evolution. The system possesses closed loop automation capabilities across multiple services, multiple domains, and the entire lifecycle, achieving autonomous networks.
	In Level 5 Zero X business capabilities are available to E2E full lifecycle operations of any services.

The AN Level Framework illustrates the combination of both human operators (P) and autonomous systems (S) involved in achieving autonomy and proposes the following *lifecycle management categories* in the evaluation of autonomous networks levels:

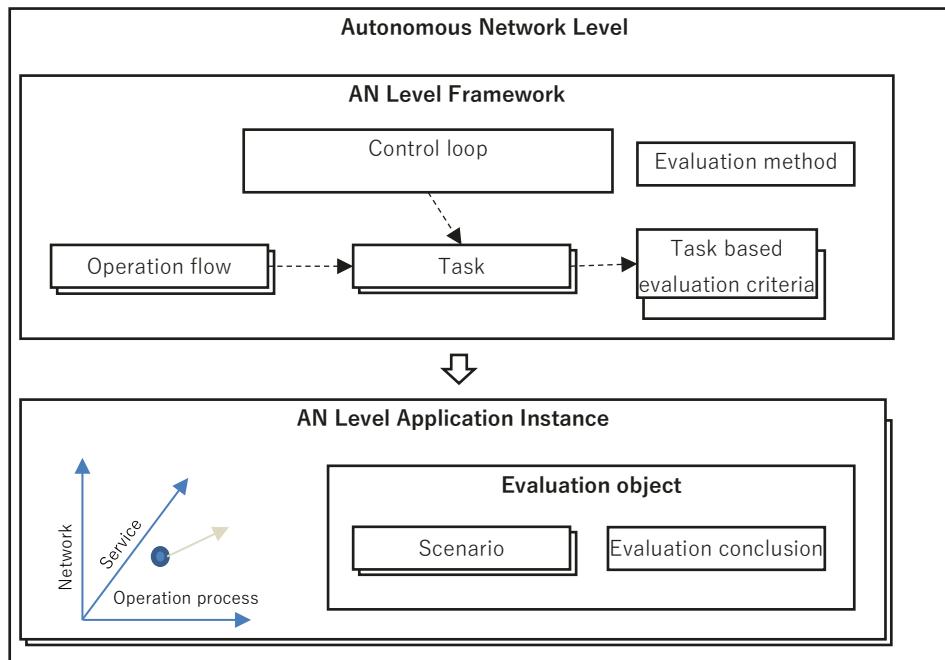
- *execution,*
- *awareness,*
- *analysis,*
- *decision,*
- *intent/experience.*

The lower levels (L0, L1, L2) can be achieved now and deliver immediate cost and agility benefits in certain scenarios. An operator can then evolve to higher levels, gaining additional benefits and addressing a wider range of scenarios. Network automation is a long-term objective with step-by-step processes, from providing an alternative to repetitive execution actions, to observing and monitoring the network environment and network device status, making decisions based on multiple factors and policies, and providing effective perception of end-user experience. The system capability also starts from some service scenarios and covers all service scenarios. This transformation will take several years to fully develop, and so the autonomy levels provide a graduated process of introducing automation with autonomy capabilities into different domains to bring immediate value. A proposal for a methodology to determine the level of autonomy for particular tasks is now described in section 4.4.

#### 4.4 AN Level Evaluation

Network operations consist of physical/logical objects and their state map for each object, all the operations are fulfilled by the *intents* that shift one entity to a desired specific status. The process of AN Level evaluation could be taken as auditing the implementing phase of intent. Evaluating the autonomy and quality of the implementation of given intents in a specified system/solution. Considering the fulfillment of intents are cascading/iterating, the process and tasks fulfilling the intents are also grouped recursively, and the iteration could be in great depth, in this case, the collecting of operation performance and automation ratio cannot be carried out by a human, but better to be kept by the system itself, requirements and regulations for the statistic and reporting need to be addressed on this purpose. There is a cognitive process of human actions, including awareness, analysis, decision, execution, each intent implementation are a complete cognitive process instance, intents may be iterated, but if we only evaluate the first iteration, or say level 0 or 1 in cascading, it will simplify the evaluation process. When compared to the overall system automation evaluation, it is suitable for humans to carry out the *evaluating* task.

## 4.5 A Practical AN Level Evaluation Methodology



**Figure 4-1 A practical AN Level evaluation methodology**

### 1. Generic Process of AN Level Evaluation

Evaluation applies to “evaluation objects”, the first step of evaluation is choosing the “evaluation object” to be evaluated. After the “evaluation object” chosen, we break down the operation process of the evaluation objects into operation tasks, aggregate operation tasks into AI recognition processes and activities, evaluate and score operation tasks based on the task criteria( person, person-system, system), then calculate the weighted average of operation tasks, obtain the scores of AI cognitive processes and activities. Then, calculate the weighted average of the scores of AI cognitive processes and activities, and finally obtain the AN level score of the evaluation objects.

### 2. Control Loop Mechanisms

Network autonomy can be implemented in different closed control loops across the operational layers, but the complexity of network autonomy depends on the layer. It will be a greater challenge for the AN to achieve the network autonomy in a cross-domain layer rather than within a single domain layer because more autonomy mechanisms need to be introduced for the coordination across different domains.

Following are potential scopes of Autonomous Network Level:

- *Business control loop layer; business is the managed object in this layer.*
- *Service control loop layer. Service is the managed object in this layer.*
- *such as mobile 2C services , home broadband services, etc. Resource control loop layer. Resource/network is the managed object in this layer, such as home network, wireless network, fixed access network, IP metro network, etc.*

The Autonomous Network Level in this version is applicable to service closed loop layer and resource closed loop layer.

### 3. Operation -flow

Based on the best practices, each operational layer of the autonomous network can be decomposed using the following operation lifecycle -flow:

- *Full-lifecycle -flow in Business layer: Including customer lifecycle management processes and product/offering lifecycle management processes.*
- *Full-lifecycle flow in services layer: Including the flow of resource facing services planning, build, fulfillment, assurance. Also includes the flow of customer-facing services planning, build, fulfillment, assurance.*
- *Full-lifecycle flow in Resource layer: Including the flow of resource/network planning, build, maintenance, and optimization.*

Further decomposition of the operation flow of services closed loop and resource closed loop allows us to obtain the following sub-flow:

**Table 5 Decomposition of Control Loop Operation Processes**

Operation - flow	Sub-flow	Description
Planning	Network planning	Based on the customer's business intention, service development objective, network construction plan, and network capacity analysis and prediction, output the network planning solution. Based on the planning solution, the network survey, equipment procurement, and technical requirements of the solution, output the network low-level design.
Deployment	Design and deployment	Based on the networks low-level design, complete hardware and software installation and optimization, output acceptance reports, and complete equipment and network configuration (pass the acceptance criteria and ready for operation & maintenance).
Maintenance	provisioning and configuration	Configure services and networks based on service provisioning requirements.
	Fault management	Set monitoring rules based on the customer's O&M policies, monitor the services and network status in real time, detect faults or potential risks in a timely manner, demarcate and locate the faults, analyze the root causes, and rectify the faults or potential risks.
	Service change and Network Change	Based on network change requests generated from monitoring and troubleshooting, parameter optimization, and planning and design, Analyze the impact on user services ,output change constraints (time window and service interruption time), formulate network change solutions, and implement the changes to eliminate network faults or potential risks and improve user experience.
Optimization	Optimization parameter adjustment	Based on network performance tests, customer complaints/feedbacks, and resource utilization, Formulate and implement network optimization solutions, to meet customer service experience and resource utilization requirements.

#### 4. Proposed Standardized Task

Based on best practices, decompose the operation process and sub-process of services closed loop and resource closed loop into operation tasks.

The process and sub-process breakdown principles are as follows:

- *Completeness: All tasks in the workflow in the operation process must be included.*
- *Balance: The division granularity of each task is relatively balanced and decoupled.*
- *Uniqueness: The operation task does not overlap and does not have functional intersections.*

#### 5. Task Level Evaluation Criteria

A key criterion for determining the *intelligence* level of each task in L0 to L5 is to consider what agents are performing the task:

- *Person: The task is completed by person, and no system assistance capability is available.*
- *System assist person: The task is jointly completed by person and system. The main operations are completed by person and a few tasks are automatically completed by the system.*
- *Person assist system: The task is jointly completed by person and system. The main operations are automatically completed by system and a few tasks are completed by person.*
- *System: The task is automatically completed by system.*

#### 6. Control Loop

Control loop is used to describe the necessary steps to fulfill certain management purposes. A workflow is composed of one or more management tasks. The autonomy capabilities of the tasks in the workflow may impact the network autonomy level.

Following are the potential categorization of the tasks in a workflow:

- ***Intent translation:*** *The group of tasks which translate intent from operator or customer into detailed operations which may affect one or more of the following groups of tasks (i.e., awareness, analysis, decision, execution), also evaluate and feedback intent fulfilment information (e.g., the intent is satisfied or not).*
- ***Awareness:*** *The group of tasks that monitor and understand what happening or exists in quality/state of network/services, such as collecting network/services information, preprocess information and selective notification, etc.*
- ***Analysis:*** *The group of tasks which analyze data generated in the awareness phase, use technologies such as model inference and analysis to further predict the future change trend of the above network status, and make a recommendation for decision.*
- ***Decision:*** *The group of tasks that decide the necessary management operation for execution, e.g., network configuration or adjustment.*
- ***Execution:*** *The group of tasks which execute the management operations and feedback the result to the intent requestor.*

For example, the relationship between tasks and AI-based cognitive workflow in monitoring and troubleshooting operation process is as follows:

- *A task group of monitoring visualization, fault identification, and risk prediction constitutes awareness.*
- *Tasks group of problem demarcation and locating constitutes analysis.*
- *Tasks group of solution generation, evaluation, and decision-making constitute decision.*
- *Tasks group of solution implementation and verification constitute execution.*

## 7. AN Level Evaluation Object

There are several dimensions to specify the evaluation objects, including: a. the service(HBB/leased line/...), b. the network(e.g., Wireless/fixed access/ipran/ip metro/etc.), c. the operation process (see Figure 4-1)

## 8. Scenario

The evaluation object may have multiple scenarios based on internal and external conditions. For example:

- *Multiple fault types of networks in domain network assurance processes*
- *Multiple Service-related device types in the multi-domain network fulfillment process*

The network autonomy can be implemented for different scenarios, the complexity of network autonomy depends on the detailed scenarios it applied. Also, it will be more challenge for the telecom system to achieve network autonomy for full scenarios than for certain scenarios. For example, the autonomy applicability of network deployment will be more challenging for a combined outdoor-indoor scenario than for an outdoor-only scenario.

## 9. Evaluation Method:

Defines how to perform level evaluation based on the task and task criteria and specifies the evaluation process and calculation method. The procedure of the assessment method is as follows:

- Step1: Determine the evaluation object.
- Step2: Describe the operation process.
- Step3: Task Map to operation Task
- Step4: Evaluate the task level.
- Step5: Calculate the evaluation object AN level.
- Step6: Analyze and output the AN level evaluation conclusion.

## 10. Evaluation Conclusion:

The AN level result of an evaluation object, including the evaluation score, AN capability analysis of strength and weakness, and improvement suggestions.

The evaluation result of an evaluation object is provided, including the score, analysis of strength and weakness, and improvement suggestions.

## 5 Autonomous Domains

### 5.1 Definition

This concept of an Autonomous Domain defines the scope of autonomous behavior delegated by an organization and is the basic building block of the AN Technical Architecture.

As previously described in section 4.1 , an Autonomous Domain is defined as:

- *An administrative Governance Boundary (Management Domain) that defines the scope of delegated autonomous behaviors. It exposes a set of Business Services based on the capabilities of a set of objects over which the AD exercises governance ( policy, rules, patterns including Closed control loops),*
- *Is self-managing self-scaling and manages its own lifecycle and publishes its Service Capabilities and Business Services in a Catalog / Repository.*
- *Objects can include Intent-based APIs, management interfaces, (sub) Domain managers, Control loops, controllers, orchestrators, decision-makers, AI analytics, monitors, policies.*
- *An Autonomous Domain is the lowest unit of composition of an Autonomous Network.*

Autonomous Network involve multiple layers and closed loops. Autonomous domains serve as the basics unit that can fulfill the closed-loop automation of the lifecycle of specific network operations of Autonomous Networks based on the business disposition of network functions and operations. This reduces technical complexity and conceals the variations of different vendor implementations, thus supporting E2E business requirements of AN services.

The boundary of autonomous domains is based upon the network operation requirements and business decision of each CSP. The instantiation of autonomous domain can be defined by CSP based on the factors such as service types, network technologies, deployment locations, and maintenance organization relationship. The examples of autonomous domain instances can be the closed loops of access, metro backbone, core, edge, customer network from infrastructure perspective, or SD-WAN, VoLTE, CDN etc. from service perspective.

The basic principles of the operations of autonomous domains are:

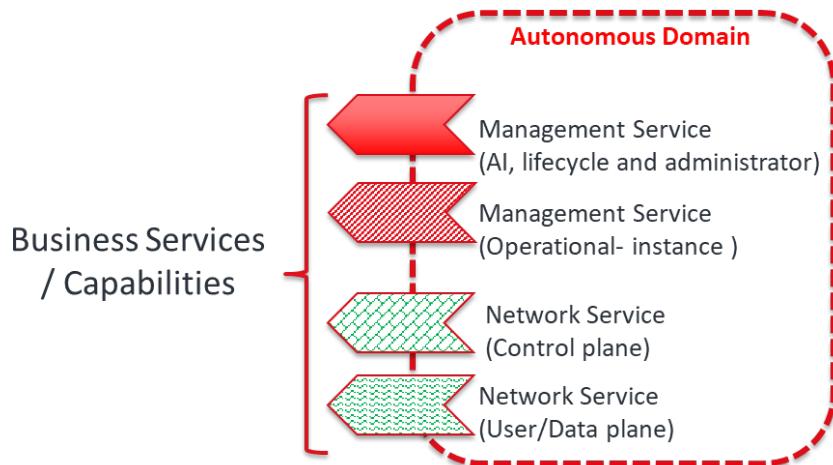
- *Autonomy of individual autonomous domain: each autonomous domain runs in self-operating mode per business objective and hides the details of domain implementation, operations and the functions of the domain elements to the users of autonomous domains.*

Collaboration of cross autonomous domain: multiple instances of autonomous domains can be collaborated by upper layers service operations using the intent-based interaction to fulfill the lifecycle of network/ICT services.

The key characteristics of autonomous domains:

- *Can model the exposure of network and service capabilities as a platform and/or services to enable higher-level business services to utilize network capabilities at the domain level instead of element level.*
- *Can specify a set of rules at the business level (e.g., Service level availability, service level guarantee based on response times, repair times etc.) that can be automatically monitored and effected across all domains of the architecture in support of closed loops.*
- *Can be instantiated per business dispositions that represent network operations of the future e.g., access, edge, core, network services and so on.*

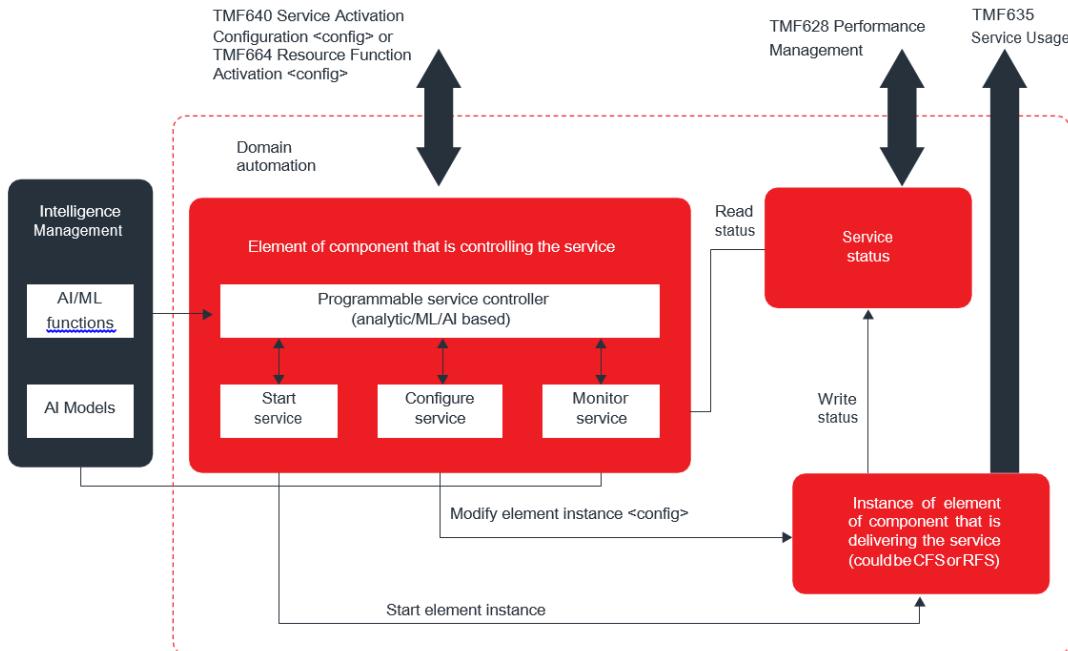
- Each instantiation can be decoupled from another and expose a set of domain-based services via common intent driven interaction/ Open APIs to upper layer or other domains.



**Fig 5.3 Autonomous Domains expose Business Services**

The internal business capabilities are represented by the building block. Each Autonomous Domain exposed exposes services – Business Services - each exposing a part of the capabilities or Functionality of the Autonomous Domain. These Business Services are the basis of the Integration Pattern for Autonomous Domains. Business Services can be realized by collections of TM Forum Open APIs.

## 5.2 Example of an Autonomous Domain



**Fig 5.6 Implementation functions for implementing Autonomous Domains**

The flow for how these features implement self-healing is described below:

- Start the service running using TMF640 service configuration and activation or TMF664 resource function activation – includes the “intent” parameters.
- Service controller will start a service element **START SERVICE**
- Service element will start producing outputs (TMF635) and performance statistics (TMF628) **PRODUCE SERVICE STATUS**
- The internal service monitor will read the status and compare it against the service intent **MONITOR SERVICE**
- If the intent is not being achieved, then based on analytics, ML or AI the service controller will attempt to achieve the intent by either reconfiguring the service or starting a new service element **REPAIR SERVICE**
- Closed loop control is delivered within the domain. **CLOSED LOOP CONTROL**

### Autonomous Domain is a composite structure

The Autonomous Domain must expose a precise *boundary* that only exposes control points for that domain (Figure 5-2). Several composition options for Autonomous Domains can be supported as shown below in Figure 5-1.

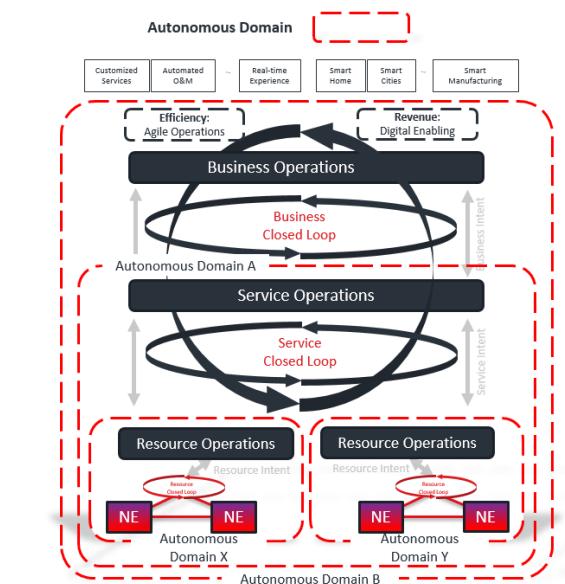


Figure 5-1 Autonomous Domains are composite

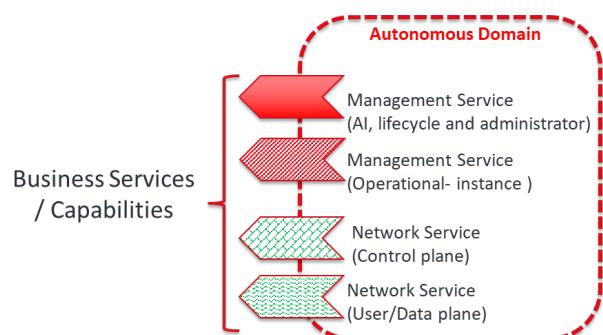


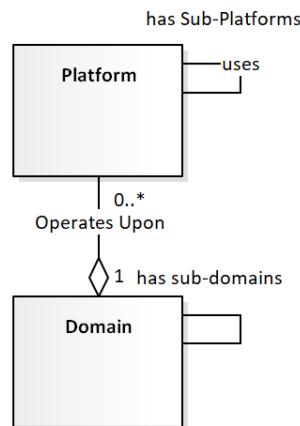
Figure 5-2 Autonomous Domain Capabilities

Autonomous Domains can vary in scope depending on the delegation of responsibility to each Autonomous Domain by a governance authority. The governance authority chain of command is a separate consideration not addressed in this section.

An Autonomous Domain enclosed within another domain means that part of the scope of responsibility of the outer AD has been delegated to another Autonomous Domain but that the governance rules applied to the out domain also apply to the inner domain. However, the inner Autonomous Domain may have additional rules and constraints applied.

## Autonomous Platform

A Platform is typically made up of many systems, associated processes, Information, and organization managed as a coherent unit to deliver a well-defined set of business/platform capabilities through simple to use Open APIs. A platform has a governance boundary that exposes one or more coherent blocks of business functionality - 'the what' - that is realized using a combination of systems, people, processes, and information - 'the how' [TR262].



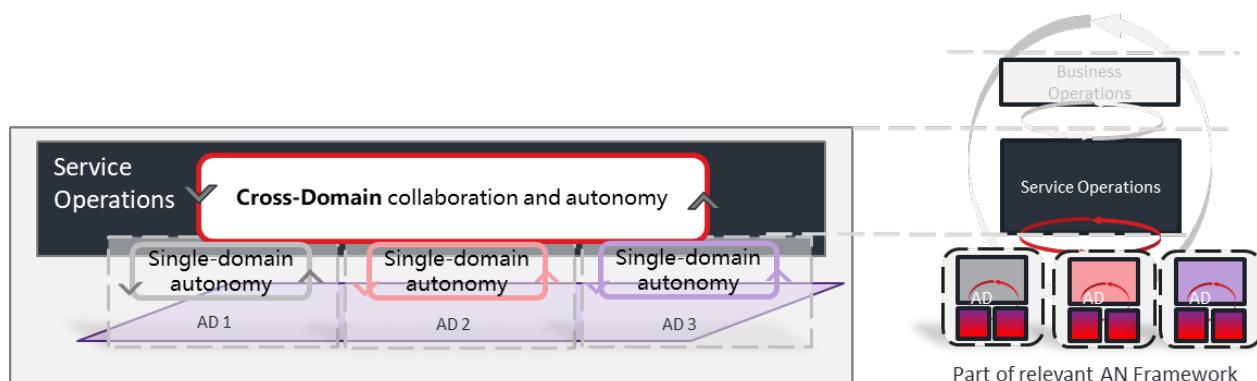
**Figure 5-3 Platform versus Domain**

A subdomain is a domain that is a member of another domain.

## 5.3 Multi-domain Collaboration and Autonomy

Autonomous Domains can be both autonomous yet collaborative – these are the two key principles which define them:

- *Autonomy within Autonomous Domain:* Each autonomous domain runs in self-operating mode per business objective and hides the details of domain implementation, operations, and the functions of the domain elements to the users of the autonomous domains by using an abstraction layer of service APIs.
- *Collaboration across multiple Autonomous Domains:* Multiple autonomous domains are collaborating and forming a new aggregated/cascaded domain, the cascading domains interact via an intent interface. Taking an example of the umbrella network manager (categorized as part of service operation for now), which usually held by carriers while not a single vendor.



**Figure 5-4 Multi-domain Collaboration and Autonomy**

## 6 Intent-driven Interactions

Autonomy in networks means that business, service, and resource operations are able to adapt decisions and actions dynamically to changing goals and requirements and cover a broad range of situations without human interventions. In this new environment, the business objectives of the operator as well as expectations of customers and users need to be communicated to the software that constitutes an autonomous network. This is the role of intent. Intent describes the business objective of operators as well as the expectations of customers and users. In this respect, intent establishes machine-readable knowledge about goals, targets, requirements, and constraints. Intent defines **what** an autonomous network is expected to achieve, but it leaves the details of **how** a network is designed and operated to the internal operations of the network platform. This means that the smart software in the platform can constantly optimize how the service is delivered and we can incrementally add new technologies like Analytics and Machine Learning to constantly improve the implementation. For autonomous networks, the Intent-interaction model is a must, not optional. The role of Intent is beyond relieving the burden of the user knowing implementation details. More importantly, it sets the autonomous system's internal goal. And the system then takes proactive actions to achieve the stated goal based on its observation of the environment. So intent is the fundamental mechanism to utilize the service of the autonomous networks.

### 6.1 Intent and Intent Evolution

The term intent appeared in telecommunication and IT already in 2014 as “intent-driven networking” in the context of Software-defined Networks (SDN). Consequently, many early implementations of intent appeared in SDN controllers in 2015 and 2016. The early use of intent in SDN was very in a technical network context and was directly involved in selecting alternative policies. The intent was either initiating processes or it was used in branching within decision trees while provisioning and managing SDN assets. The definition of intent at the time reflected this use of intent. For example, in 2015 IETF states that intent is **“An abstract, high-level policy used to operate the network.”**

In the meantime, the understanding of intent as a concept and its role in automation has evolved from the policy and rule’s centric view towards being a universal entity for declaring and communicating goals, requirements, and constraints. The direct link to implementation assets such as policies and processes becomes less eminent with the tendency to be explicitly excluded. Furthermore, the setting of intent is often directly linked to human expectations and choices. In this respect, IETF has presented a new definition of intent in 2020, now intent is defined to be: **“A set of operational goals that a network should meet and outcomes that a network is supposed to deliver, defined in a declarative manner without specifying how to achieve or implement them.”**

This example shift in definition can be observed throughout literature and standardization. The speed of this transition might vary, but a common direction is very evident.

In 2019 a technology assessment by Ericsson research did a broad overview of usages of intent throughout the industry and academic publications. It summarized the findings in a new definition: **“[...] an intent is a declaration of business goals, operational goals or states that a system should meet, without specifying how to achieve them (i.e., not prescribing events, conditions or actions).”**

The examples of newer definitions introduce the idea that intent is a declaration of goals independent of their implementation. Furthermore, it expands the scope of what intent describes and where it is used. Business and higher operational levels are in scope rather than just the network.

This newer definition of intent is also in line with the description of intent presented by ETSI ZSM:

**“Today intent-driven systems refer to intelligent software able to understand the user goals and translate automatically into a concrete prescription of service or network configuration.”**

**“Intent is understood to be the information of knowledge objects describing these goals.”**

From the perspective of the user, these goals express the expectation the user has directly with respect to the actions and behavior of the system. They might also formulate derived instrumental goals needed to fulfill human expectations.

The AN project definition follows the trend visible in the evolution of the *intent* concept and defines intent as follows:

*“Intent is the formal specification of all expectations including requirements, goals, and constraints given to a technical system”*

This definition of intent focuses on the *user* perspective. The user has expectations, which are directly or indirectly expressed as goals, requirements, and constraints. This also means that the intent formalism must be capable of expressing all semantics involved when users express their expectations that are relevant for correct operation for an automated zero-touch domain.

These formally modeled and common semantics allow a technical system to understand what it is expected to do. It also implies that intent is encapsulated in self-contained objects, which can be exchanged over APIs to communicate the user expectations and operational expectations.

The definition also points at completeness. All expectation is formulated by intent. From the system perspective, this means that all it needs to operate and optimize against is the set of intent given to it and nothing else.

## 6.2 Implications of Intent

Intent is “specification of all expectations including requirements, goals, and constraints” that applied to “a technical system”. We further deduce that intent will apply to certain objects/entities, that will be offered as the specification of the intent interface of the “technical system”. 3GPP defines intent as “a desire to reach a certain state/position for a specific entity”. That means:

- *intents apply to entities.*
- *Intent is not “doing” something, but to make something in a certain state, with a clear objective and goal.*
- *The intent occurs on a specific working object, and the intent expresses the expectation of the state of a particular object, while the states usually consist of lifecycle status and attribute status, such as the instantiation (appearance) of a service and the maintenance of its SLA attributes. Another example would be a base station, its status shifts from unmounted in a warehouse to be deployed in a site, then to completion of configuration and getting ready for service.*
- *a specific “intent set” to be offered by a system.*
- *Before interacting with a system, the supported “intent set” must be clear, describes the supported objects and the states shifting graph, it could be catalog-based, or only described in documents.*

*With this “intent catalog”, the user or third-party system will know what you can do and what intents make sense in your system.*

### Separation of Intent from Intent Handling

If we define the intents as expectations on entities states, then “intent” is not a brand new concept, it has always existed, even in the days of imperative UI/API based operation.

One difference is that, in the past, intent generation and implementation were all performed by human beings and not a separate process. In autonomous networks, it is necessary to separate the “intent” and its realization. Setting up an agent/handler, separating the interaction of intent and the agent/handler which handling it, and the agent will fulfill the intent intelligently and automatically. It necessarily requires a clear definition of intents, a grammar for describing intents, abstracting the intents in different layers, and clarifying the mechanisms to dealing with the intent and lifecycle of intents. All these concerns and implications will be covered in the following sections.

## 6.3 General Properties of intent

<b>Intent is declarative</b>	It leaves room for the autonomous network to explore options for finding the optimal solution. Intent declares the wanted results rather than prescribing a specific solution. Ideally, intent expresses a utility level goal that describes the properties of a satisfactory outcome rather than requiring a specific outcome.
<b>Intent is infrastructure agnostic and portable</b>	The expectation expressed by intent originates from contracts and business strategy. It does not change if the underlying system is replaced or modified. While implementation and capability differences between system vendors will continue to exist, intent can be ported between system generations and implementations.
<b>Intent is complete</b>	Intent defines all goals and expected behavior. If it is not specified as intent, it is not a goal the system needs to consider. This also means that concerns that were common sense in human-operated systems would need to become explicitly expressed as intent.
<b>Intent is composable</b>	Multiple intents are given to the autonomous network, and it is expected to consider them all together. Unlike traditional software systems, where requirements are analyzed offline to detect and resolve conflicts before implementation, intents are added during run-time. Therefore, an essential capability of an autonomous system would be to detect and resolve conflicts.
<b>Intent is persistent</b>	Intent is valid as long as the goals and requirements it expresses are relevant. For example, an intent that specifies a service needs to be delivered would not become invalid once the service is initially provisioned. Intent is rather the reason for keeping the service operational and assure its performance. Therefore, intent has a life cycle that is actively managed by the user or function that has generated it.
<b>Intent is measurable</b>	It uses measurable and ideally standardized metrics to define the target state. This allows the automated evaluation of success as well as identification of issues and optimization opportunities.
<b>Intent is comprehensible</b>	It must be understandable by humans while being formally and unambiguously specified allowing it to be processed by machines. It must be comprehensive in what it specifies for matching the semantics within the autonomously operated domain, and the scope of autonomously operating tasks.

## 6.4 Classification of Intent

### **Service-specific intent**

Service-specific intents are related to the delivery of a service instance. For example, the intent on business level could be to deliver a contracted service according to agreed functional specification and non-functional requirements expressed by target KPI. On the business operations layer, this is specified by Service Level Agreements (SLA) in combination with Service Level Objectives (SLO). In order to fulfill SLA and deliver the service, the service operations layer would be involved by specifying the technical details of the service. This specification corresponds to the technical requirements of an SLA defined as Service Level Specification (SLS). The fulfillment of the service might involve orchestration and that is steered by TOSCA models. TOSCA models allow the declarative specification of all service elements, needed resources, and their dependencies and topology.

This means that there are at least three different types of service-specific intent. Intent containing SLA would be connected to service contracts on the business operations layer. Intent containing SLS and the technical specifications of SLA would be used to order a service from the service operations layer with all functional and non-functional requirements. Intent containing TOSCA models would be the input to the orchestration subdomain of service operation.

### **Service-agnostic Intent**

Non-service specific intents or service-agnostic intents are globally applicable rather than valid for a single service instance. They express global expectations. This can for example be minimum standards to be applied to the entire portfolio. While these intents are called non-service specific, they would therefore contribute further details of how services are operated.

Furthermore, non-service specific intents can address the expected behavior of the autonomous system including for example prioritization, risk-taking, or communication with humans.

### **Legal, regulatory and standards compliance**

It is important to follow legislation and regulation within the markets involved in service deliveries. Intent can explicitly ask for the legislation or standards to follow and it might differentiate by market region. The level is also a choice if it is understood as meeting defined standards or integrating certain functions. For example, the use of a minimum level of encryption as default within all network slice links can be a functional requirement of a service. Instead of defining this in every service individually, an operator might use a global intent to specify this as a minimum default security level applicable to all services.

### **Financial gain and margin**

Automated contract negotiation might be used for ad-hoc contracting of service without human involvement. A system capable of doing this needs the direction of what it is expected to accept or not. Intent can be used to define respective targets.

### **“Common sense” Intent**

Human common sense is what makes human operational decisions still superior to automation. It allows us to weigh in considerations and situational conclusions, which rely on experience and contextual knowledge, which might not be accessible to the technical system. For example, a human technician knows without the need to be told explicitly that saving resources is an important concern. The technician also knows that some customers are more important than others. This helps to correctly prioritize and plan beyond what is possible from observed data or specifications of services alone. When replacing human decisions with a technical system expected to deliver high-quality operation, common-sense considerations are critical to

the model. However, based on the current state of the art it is not possible to be complete and reach the human level common sense with currently available techniques and tools. Nevertheless, it is still possible to cover many key concerns. Ideally, they are formulated as a utility, because that would immediately create review capabilities of actions and states.

### **Save resources**

The need to save resources is a typical example of a commonsense concern. It can be expressed for example by defining resource value metrics and priorities per resource types. A good metric to base the intent on can be resource utilization, because only productively used resources generate income.

### **Avoid unwanted bias**

A bias such as *priority* for high-value customers is a welcome feature. But other unwanted biases might undermine the reputation of the operator or even break laws. Examples are social/racial/sexual preferences exhibited by the autonomous operation. The root cause can be that bias in the training data did lead to bias in the resulting ML models. Another source can be the bias that is introduced into policies from the prejudices of the developer. The operator choice would then be to use intent for setting goals that steer the sensitivity of bias detection mechanisms with respect to certain types of bias.

### **Conflict resolution hints**

Intents will conflict with each other, and the zero-touch operations system would need to resolve these conflicts with optimized and balanced actions. For doing this it needs any kind of information indicating a good prioritization strategy. For example, intent-driven metrics for customer or resource value can be used to convey valuable information for prioritization.

### **Behavioral and Interaction Intent**

The operator should have a choice of how the zero-touch system behaves and how it interacts with the human workforce.

### **Reporting frequency and scope**

Intent can be used to express what the system is supposed to report to the operator and what to include in logging and historical data recording. It can furthermore specify how to report with respect to formats and channels to be used.

### **Risk-taking**

If an automated system is allowed to act completely autonomously there is a good chance that unforeseen situations will lead to actions that degrade the operation. This can happen if models are exposed to new situations outside their intended scope. The operator might want to limit the risk the autonomous network is allowed to take and formulate the constraints using intent.

### **Escalation to human decision**

The operator can also define when the system is expected to not operate fully autonomously but escalate decisions to a human operator. This can be the action wanted from the autonomous network if its operational actions were found to be too risky. The setting of respective thresholds and targets is also subject to intent.

## **6.5 Intent within the AN Framework**

Traditionally only the network professionals can understand network implementation technology, which hinders the service innovation and negatively impacts the customer experience. Intent driven interaction is an important bridge between business requirements and network implementation. Transforming business intent into services and related network resource requirements is crucial to achieving business goals. Intent represents the concerns and objectives of the users of an autonomous network. It, therefore, varies with the diversity of user types and roles:

**Business intent** represents the objectives of a business user. This includes for example the delivery of a custom application defined by SLA. Operators expect their autonomous network to operate service contracts while meeting revenue targets. Their customers expect a good user experience.

**Service intent** represents the objectives of a service user. A service is expected to deliver functional as well as non-functional attributes. This includes targets for example on connectivity, bandwidth, latency or availability.

**Resource intent** represents the objectives of resource users. Resources are expected to be allocated so that the performance and quality of service targets are met.

This indicates that intents target a great variety of concerns across the Autonomous Network Architecture. Therefore, the handling of intent is distributed throughout the autonomous network layers and autonomous domains.

The interactions between different layers are through intent APIs. For example, business intent would be handled in business operation and resource intent would be handled within the autonomous domain that matches the concerns addressed by the intent.

An intent handling function is the basic architectural building block to assemble intent-driven operation. Figure 6-1 shows the intent handling function with its intent API. It receives intent, takes decisions if and how to act, and then dispatches operational actions. The actions can be conventional API calls towards other functions. If the target functions have implemented intent handling, the operational activities can be defined their goals by using subsequent and typically more detailed intent.

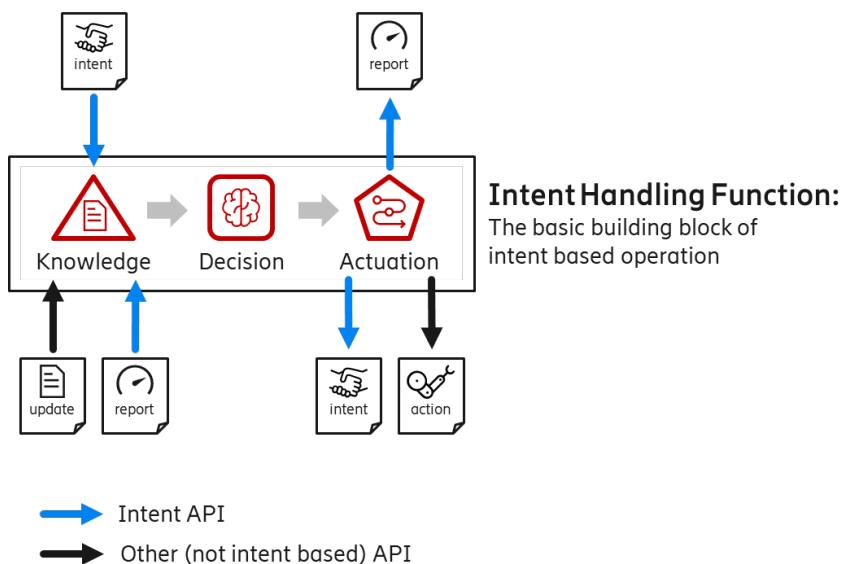
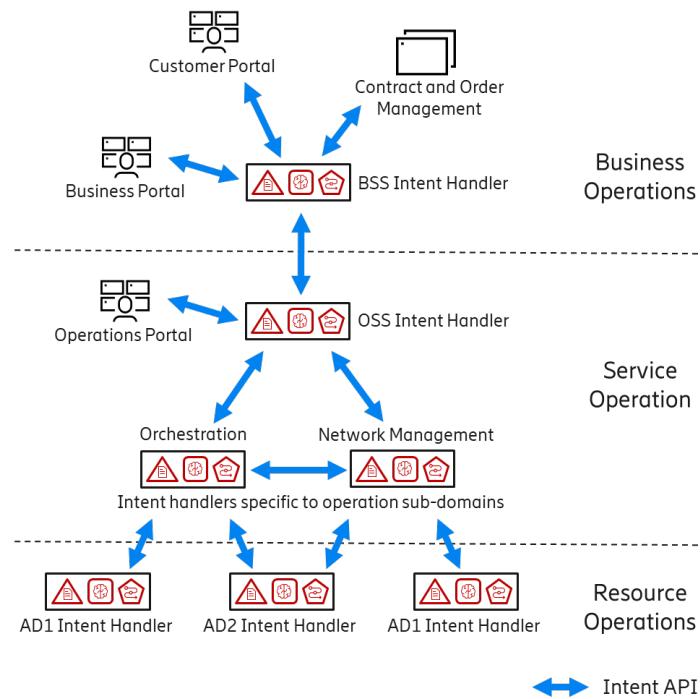


Figure 6-1 Intent Handling function

Figure 6-2 illustrates by example how various instances of intent handlers are allocated across the layers and functional domains of the Open Digital Architecture. Intent can originate directly from user input through front-end portals. Additional intent would be derived automatically, for example from contracts and service orders. An intent handling function would operate its domain by analyzing the discrepancy between the observed state of the network and the wanted state expressed by intent. The main task of intent handling would be to close this gap as much as possible. It determines the optimal operational state

according to all given intent and then acts to transition the network into this state. This process includes the resolution of inevitable conflicts between intent through prioritization and optimization.



**Figure 6-2 Autonomous Network operation based on Intent handling**

The intent handling function can act by defining the goals of neighboring or subordinate domains through further intent. Furthermore, for all intent that is given to an intent handler, it is expected to report progress and status back to the source of the intent. This closes a loop. The intent mechanism is therefore instrumental for creating control loops throughout the Autonomous Network Architecture.

### Intent Requests

Intent-driven interactions within an autonomous network are interactions between various domain-specific intent handlers using the intent API. Intent originates from users through portals, or it is generated based on other API calls. For example, an incoming service order would lead to the generation of a respective intent by contract and order management in the business operations layer.

The Intent API is concerned with the life-cycle management of intent objects. It is solely a knowledge manipulation API. All domain-specific information is encapsulated within the Intent objects rather than directly mapped into API methods. This means the Intent API is domain-independent and applicable to all instances of intent handlers irrespective of their allocation within the layers and domains of the ODA.

## Specifying Intent

Examples for the initial setting of intent are shown in Figure 6.4.

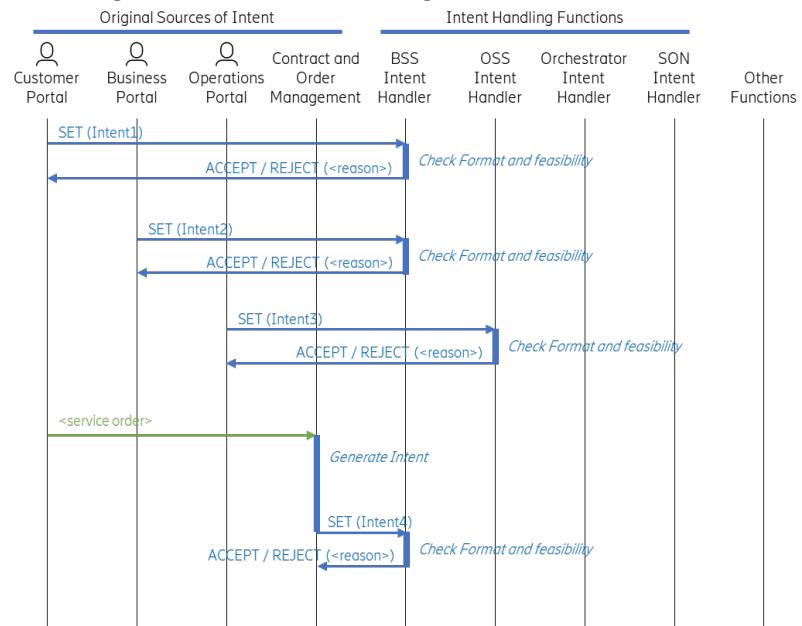


Figure 6-4 Intent API: Sources of intent

Intent handlers typically receive multiple intents indicating the desired state. They then decide on actions to optimally fulfill the intent and transition the system state as close to the desired state as possible. This involves setting intent in subsequent layers or neighboring domains within the architecture. Figure 6-5 demonstrates how subsequent intent is set within various intent handlers.

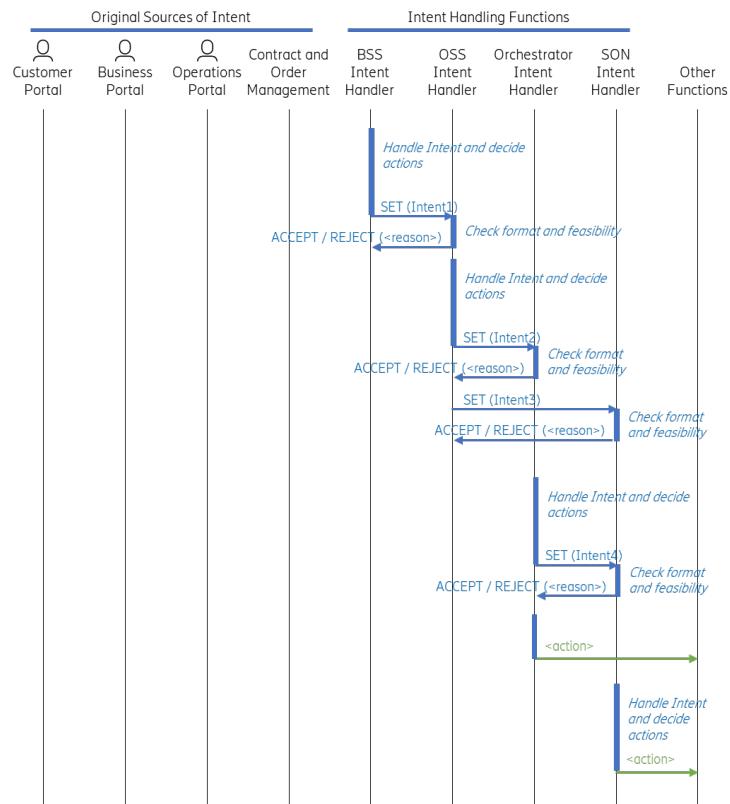


Figure 6-5 Intent API: Subsequent intent and actions

When an intent is set, the receiving intent handler decides if it can accept the intent. The intent needs to be formatted in a way the intent handler can parse and process it. Also, the intent needs to contain semantics the intent handler understands. This is the case if the intent contains expectations and supplementary information that matches the scope of responsibility of this intent handler. However, if the intent is understood, in the scope of the intent handler and matching its capabilities it might still not be able to accept it if it concludes it cannot fulfill the intent with the currently available means. If either of these criteria are not fulfilled, the receiving intent handler would reply with a rejection stating the reason.

#### **Modification and Removal of intent:**

Intent is generally persistent and stays valid until actively removed or modified. The entity that has originally set the intent is responsible to manage its life cycle, including removal. It can be possible to model intent with a validity time or other pre-defined removal criteria. Whenever the criteria are met, the removal process will be started automatically, and the intent origin would be informed about the automated removal.

An intent handler can receive the request to modify and remove the intent at any time. It is important to note, that the modified or removed intent is usually just one out of many intents valid for the intent handler. Its modification changes the overall wanted state. This means that the intent handler would need to transition its operational state into a new target state optimal for the remaining set of intents. The intent handler would dispatch actions accordingly.

The actions needed in reaction to intent modification or removal might involve the removal or modification of subsequent intent. If for example a service-specific intent is removed after a service contract got canceled, the chain of subsequent intents that govern the service delivery would also be removed. Ultimately, resources occupied by the service are released.

#### **Reporting**

When an intent is set the intent handler is expected to report its fulfillment state. The report can be as simple as sending a list of all metrics requested in the intent with the currently measured value. For requested services, the current provisioning status (e.g., deployed, not deployed, degraded, ...) would be a good summary report in addition to the metrics.

The frequency and criteria for when a report is expected as well as additional content would be subject to dedicated intent or expectations within an intent. Example possibilities are:

- *Regular reports by time interval.*
- *Event-based reports, for example, if major changes, such as incidents and degradation or the resolution of them occur.*
- *No proactive reports, but the possibility to request reports is needed.*

All these reporting options can be combined as needed.

#### **Probing**

Probing refers to sending an intent to an intent handler for testing its impact on the service and resource operation. Instead of including the intent into the live operation, the intent handler would deliver a report stating the expected fulfillment success of the intent if it would be set. This mechanism allows testing if an intent would be accepted and if not accepted, it can provide the reason why. Furthermore, it allows testing the operational performance if the intent would be set. A more detailed reply might be available that contains ranges or thresholds in KPI that are possible to fulfill. Probing can therefore be a key mechanism in intent negotiations.

### **Collaborative judgment of alternative actions**

An intent handler might be in the situation that it has found multiple alternatives with different properties, but it is not able to decide on the best overall option. It can seek a judgment from the originator of its intents. This includes reports of potential detailed intent fulfillment for each of the options. The originator of the intent can then reply with a verdict about its preferences. This mechanism can also be used for escalations that involve human decisions.

## **6.6 Intent-driven Management**

There is a large gap between business requirements and network implementation. In the traditional carrier software system, the EMS/NMS of a vendor provides northbound interfaces, and the OSS system performs the system integration. In the future, the traditional integration mode will bring the following problems:

- *Multi-vendor differences still exist: Device details vary in different fields. Interface protocols, parameters, and capabilities vary from vendor to vendor. Interface standardization is slow, which directly affects the provisioning of new services.*
- *Fine-grained API interface granularity: Traditional northbound interfaces are provided based on network resource objects (such as VRF and interface). Although they can be flexibly used, service provisioning on the live network is usually based on multiple technologies (such as composite VPN). Multiple interfaces need to be assembled and used together, and upper-layer services need to be aware of specific service scenarios. In addition, different network technologies are selected for invoking, which increases network implementation complexity.*
- *More complex networks: In the 5G era, network devices are more complex, functions are more diversified, and device management parameters are more diversified. If the current integration mode is used, the system integration time may be increased from 6 to 12 months to over 18 months.*

In the AN architecture, the entire network is divided into multiple Autonomous Domains. Each AD has self-monitoring, self-configuration, and self-optimization capabilities. The upper-layer system can ignore the implementation details in the AD, which simplifies system integration. When the upper-layer system needs to meet a customer requirement, the customer requirement can be broken down into one or more networks, such as "end A to end Z", "bandwidth", and "latency", without considering the technologies, paths, and parameters to be configured on the network.

Intent refers to the requester's *expectations* of the network, service, or resource to meet and maintain a certain desired state, but the intent does not describe the methods and steps used to achieve this state. Generally, the intent has the following characteristics:

- *Customized language description: The network is regarded as a black box and the input information is customized parameters understood by external systems.*
- *Concise/abstract: independent of vendors, resources, protocols, and implementation.*
- *Declarative: Emphasize what instead of how.*
- *Measurable/Verifiable: The intention should have clear criteria for evaluating whether the objective has been achieved. In this way, when the indicator changes are detected, the system can automatically adjust the objective.*

The translation from user intents to network languages needs to be implemented hierarchically. Each lower layer appears as a “black box” for the upper layer. The upper layer does not need to understand the technical implementation of the layer. The upper layer converts the objects understood by the upper layer into resource objects understood and managed by the layer. The conversion process is based on the context of the current layer.

# 7 Control Loop Mechanisms

## 7.1 Control Loop Concepts

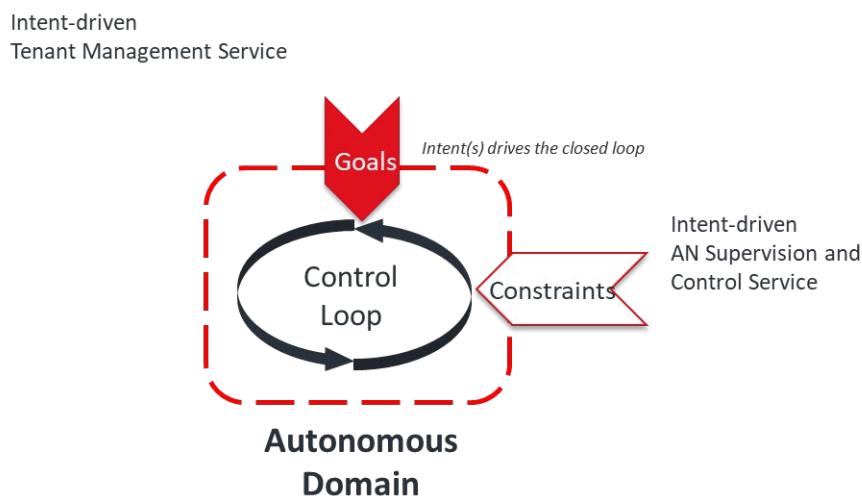
### Control Loop Definitions

Control loops are used to enable autonomous systems to adapt their behavior to respond to changes in user needs, business goals, or environmental conditions. A closed control loop is a mechanism in an autonomous system that accomplishes a task without human taking an active role during task execution.

An open control loop is where human operator starts, stops, or changes a goal or constraint related to a task during task execution. From control theory, an open loop control system does not have a feedback loop and thus is not self-correcting.

### Control Loop within Autonomous Domain

As a purpose of an Autonomous Domain is to define the scope of autonomous behavior then one approach to implementation is for control loops to operate within an Autonomous Domain. The goals for a Control Loop within a Domain are set in part by the intent-driven tenant Management Business Service to support the provision and operation of tenant services, and in part by the policies, constraints, and scope set by the governance authority through the *AN Supervision and Control Business Service* (see next section). This simple starting point is illustrated below.



**Figure 7-1 Control Loop within a single Autonomous Domain**

The operation of control loops across multiple Autonomous Domains is described later and is critically dependent on the definition of these two types of business services.

### Closed Control Loop Mechanism

Control loops tie together a rich set of concepts. Figure 7-2 shows how control loop mechanisms can be driven by an intent expression with a stated expectation or goal.

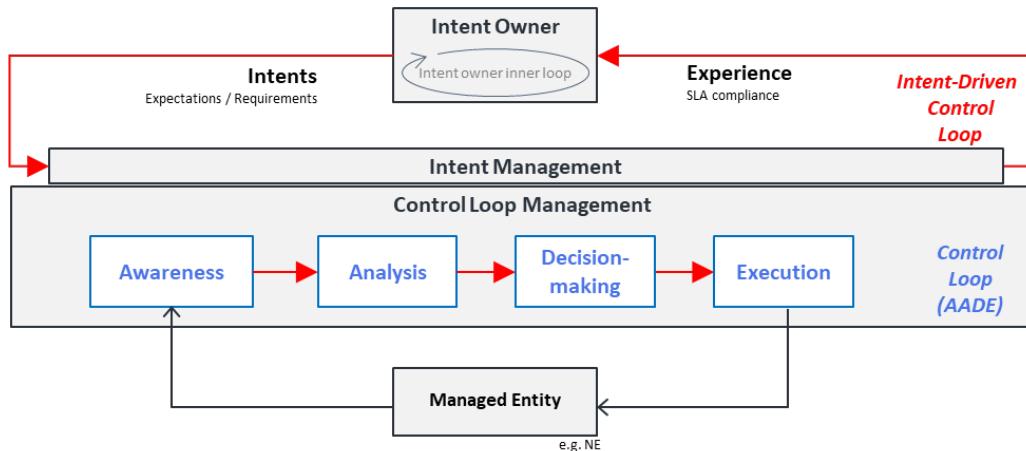


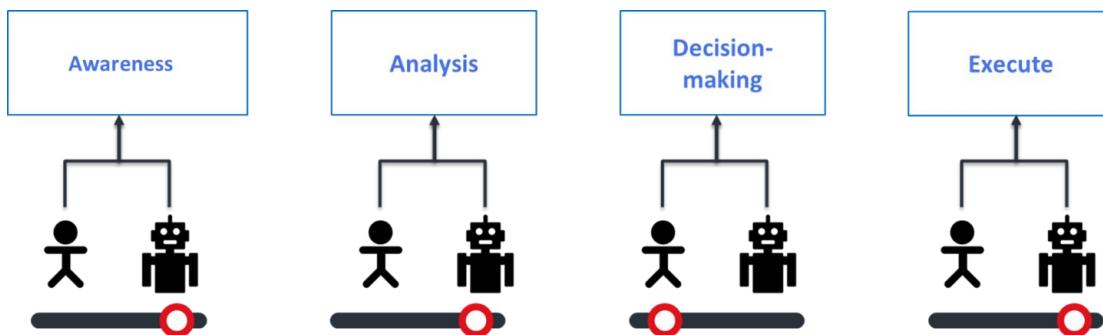
Figure 7-2 Control Loop Mechanism of Autonomous Network

The Intent *drives* the control loop. Intent handling refers to a group of tasks that translates intent expression from intent owner into detailed operations and/or policies that may affect one or more of the following groups of tasks (i.e., awareness, analysis, decision, and execution).

- **Awareness:** *The group of tasks which include network and service data (e.g., configuration data, performance data, alarm data, etc.) collection and necessary data pre-processing (e.g., data cleaning, filtering, statistics, etc.) with the purpose of monitoring network and service information (including network and service performance, network and service anomaly, network and service event, etc.).*
- **Analysis:** *The group of tasks that analyzes the obtained information or existing historical information to further predict the future change trend of the business, service or resource entities, and make a recommendation for decision.*
- **Decision (-making):** *The group of tasks that evaluate and decide the necessary operation for execution, e.g., network configuration or adjustment.*
- **Execution:** *The group of tasks which execute the operations*

### Cooperative Closed Loops

Figure 7-3 below illustrates how each phase or stage of the AADE control loop can be influenced by both humans and machines. While the loop is always closed, i.e., there is no need for intervention by the human, the possibility exists for either the human or machine to build an adaption into the control loop. For example, an intent from an operator's person could be a new additional constraint used in the analysis phase of the loop (see *constraints* in Figure 7-1). In Level 1, the machine has no decision-making authority and does everything the human says – as illustrated by the position indicator (red circle) in Decision-making is set to 'human'). Going up through these levels, the machine can either provide options to the human, choose for the human, or even be completely independent of the human in level 5.



**Figure 7-3 Responsibility can be shared across the control loop**

The sharing of responsibility will vary per situation, per operator, per human, and per instance. This responsibility is again best viewed as a continuum controlled by a slider that adjusts the level between the human operators and the autonomous systems.

### Intervention Capability

Following on from this close cooperation between humans and machine (or heteronomy) we also must explicitly handle a take-over request from the human. The capability of the supervisory layer to intervene is critical to allowing a layer of abstraction autonomously. This allows the supervisory layer or higher layer of automation to confidently allow autonomous behavior, as it can always intervene in case of exceptional circumstances and during the phase of confidence-building. In the Autonomous Driving realm, this is sometimes referred to as a request-to-intervene (RTI) and is very important for the Level 3 level of automated driving.

### Cognitive Closed Loops

Cognition is the process of understanding data and information and producing new data, information, and knowledge. A cognitive control loop selects data and behaviors 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 [ENI]. As such, a cognitive control loop resembles the human brains cognitive skills in inferring, reasoning and understanding its environment. The awareness and execution part of the control loop relate to sensing and effecting the environment, but the analysis and decision processes are elevated by collecting operational knowledge and autonomously learning from experiences in past decision loop cycles. For example, in the analysis stage of an AADE control loop, another inner control loop may be used to process additional contexts and infers from a knowledge base or from AI/ML trained model and then ultimately adapts its analysis before passing this to the decision stage of loop.

### Multiple Control Loops across AN Architecture

The AN Business Architecture provides for multiple overlapping Control Loops operating across multiple Business layers. The notion is that these Cross-Layer Control Loops provide the coordination, and optimization of the individual Autonomous Domains so that they collectively optimize across domains as well as with domains.

This can be affected by creating a model for how control loops operating across domains by coordinate and influence the control loops within an inner domain. This can be achieved by outer Control Loops and Outer Domains controlling the inner control Loops of inner domains through the AN Supervision & Coordination Business Service of an inner control loop. Critical to this model is the self-similarity of the Domain /Control Loop model at all levels of granularity and scope. Such an approach has been documented in [IG1177 ODA Intelligence Management Implementation Guide R18.5.1](#), which links the needs of Autonomous Domains and ODA and is an evolution of ETSI GANA work [GANA]. It defines for ODA how such a hierarchy of control loops can work by the implementation of ODA Intelligence Management functionality, and the use of the Decision-making Elements (DE) to execute a design template which includes governance, policy, goals, and ownership principles, exposing the desired customer-facing services (CFSs) to ODA Core Commerce Management. The outcome is the definition and description of:

- Governance and Policy requirements
- Generic information relating to Peer Domains within the Federation of Domains, and the embedding of Autonomic/Cognitive capabilities within ODA Intelligence Management
- The main KPIs to be exchanged at Federation Reference Points ODA Intelligence Management functionality, and the use of the Decision-making Entities (DE) to execute a design template which includes Governance, Policy, Goals and Ownership principles, exposing the desired customer-facing services to ODA Core Commerce Management.

The outcome is the definition and description of:

- Governance and Policy requirements
- Generic information relating to Peer Domains within the Federation of Domains, and the embedding of Autonomic/Cognitive capabilities within ODA Intelligence Management
- The main KPIs to be exchanged at Federation Reference Points

Federation References Points in ODA Intelligence Management correspond to the AN supervision and control business services.

## 7.2 Evolution of Control Loop Mechanisms

Control loops as managed entities are implemented by management systems, at both BSS & OSS levels for several years already. However, at the current time, it seems that technical foundations for cognitive-based loops are not fully stable. The implementation of AN based control loops depends on the evolution of multiple areas:

- *Standard agreed levels of network automation – The definition of these will support bring the scale of automation into a single architecture, which will let the telecommunication industry to create an evolution path to automation.*
- *Managed Entities Behavioral Models – Automation Networks require the development of higher-level abstraction infrastructures for behavior at areas that are not yet sufficiently evolved, mainly for Intent Management. These include data models and APIs for concepts like: Intent, Requirements, Capabilities, Behavior, Decisions. Such concepts should have standard data models.*
- *Standard definition of Business Services – such definitions are already available from several standard organizations yet may require multi-SDO alignment.*
- *Standard definition of autonomous domains – such definitions are required to set the governance limits among systems that are now acting autonomously.*
- *Control loops Operational Models – The decomposition of closed loop to standard types of activities. Such models already exist. Please see section 7.2.2.*
- *Standard Control loops Data models and APIs*

- *Control Loops Lifecycle Operation Definitions* – The management of control loops as first-class managed entities require standard lifecycle operations (see section 7.2.3).
- *Technical Infrastructures* – some infrastructures such as policy frameworks and cloud management tools are already identified as a baseline for autonomous management systems.

The diagram below shows a map of the different levels of concepts and infrastructures that are reflected in the technical evolution of control loops.

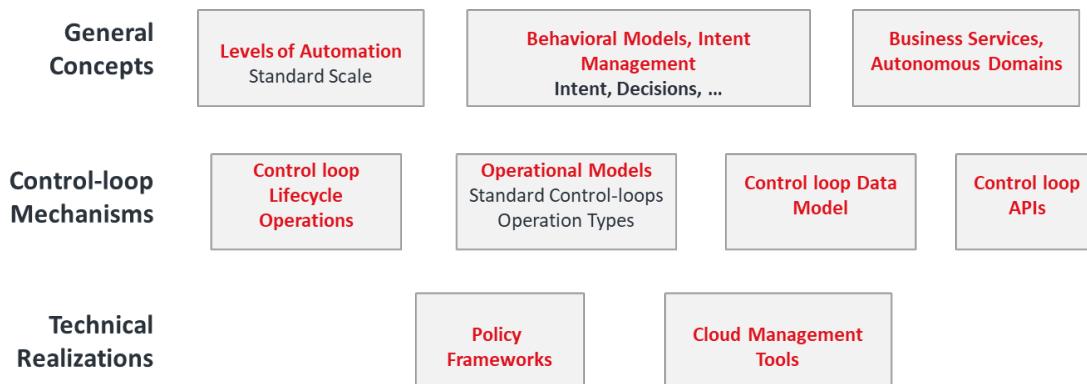


Figure 7-4 Evolution of Control Loop Modelling

### AN Control Loop Lifecycle Management

Control loops become first-class managed entities as part of AN framework. This means that the AN framework should support their lifecycle operations. There are two levels of lifecycle management:

- *Control loop Specification* – The specification level of the control loops where the behavior of the control loop is determined. The specification is often implemented by a template.
- *Control loop Instance* – The specific instance-level control loop. Control loop instances are generated by various level management and orchestration components, using the guidelines and restrictions as specified in the control loop specifications.

A Control loop specification may go through the following operations:

- *Create*
- *Modify*
- *Activate/Deactivate/State change*
- *Delete*

A specification can be used for instantiation only when it is in an active state.

Often, the specification of the control loop is used by multiple software components, and it needs to be distributed among them. The distribution of the initial specification or changes to the specification is considered to be embedded in these operations, provided by underlying framework infrastructures.

A Control loop instance may go through the following operations:

- *Create (instantiated)*
- *Modify*
- *Activate/Deactivate/State change*
- *Delete*

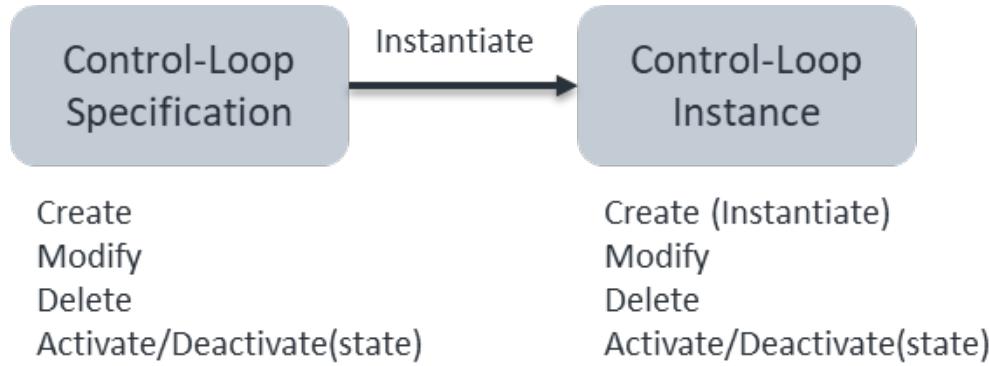


Figure 7-5 Control Loop Operations

### AN Control Loop Operational Models

Extensive research work has been done by research organizations and SDOs on generic operational models for autonomic networks. Key examples of that are:

- IBM MAPE (An architectural blueprint for autonomic computing)
- The CASCADAS project (Component-ware for Autonomic Situation-aware Communications, and Dynamically Adaptable Services)
- FOCALE (Foundation - Observe - Compare - Act - Learn – rEason. FOCALE – A Novel Autonomic Networking Architecture)

These include proposals for models describing how the various functional components of autonomous networks should communicate for achieving the goals of the autonomous network. Each of these models has a specification of an Autonomous Element (AE) and second level describing the inter-relationships among the AEs. ETSI GANA (Generic Autonomic Networking Architecture) conducted related research providing recommendations how this should be implemented by CSPs. It is assumed that these will continue to evolve during the coming years, so a good approach is to support a wide set of options for connectivity of the key autonomic management functions, possibly by a message bus, as suggested in some of the models (FOCALE for example).

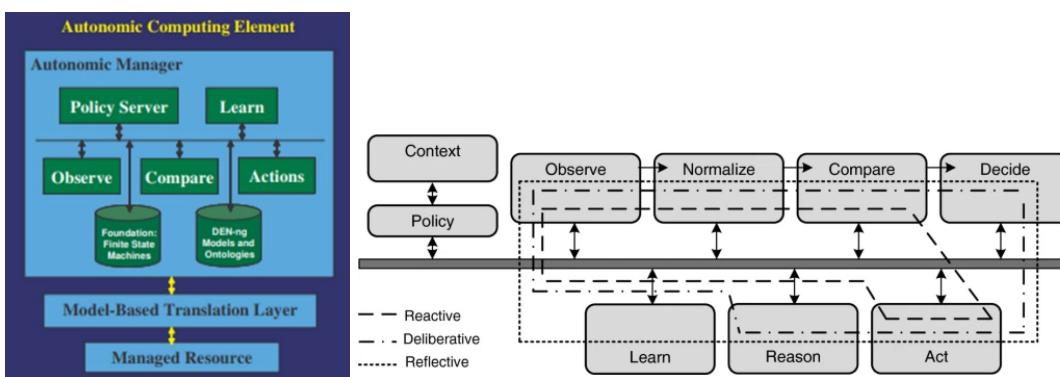


Figure 7-6 FOCALE – the Autonomic Computing Element and Closed Loops

This guide does not assume a specific computing model. The examples are closer to the IBM MAPE language. Yet, nothing in the described framework is binding it to a specific model.

### AN Control Data Model Concepts

As the control loop is a managed entity, it requires a data model to support its operations. Looking at the control loop top-level structure, we can observe its data components:

- *Control loop Scope – This is a set of managed entities that the control loop intent is applicable to. These can be products, services or resources, based on the level that the control loop is operating at. The scope of the control loop has to be included in the boundaries of the autonomous domain.*
- *Control loop Intent (Goal) – Any Control loop needs to have an intent expressing a goal. The pattern that is used to describe the goal depends on how the behavior of the managed entity (the scope) is described. The most common pattern for an intent is to target KPIs specifying target values to reach or to maintain. In specific cases the terms SLO (service level objective) or OLA (Operational Level Objective) are used. A requested state is another commonly used pattern when a state machine is used for managing the operational behavior of a managed entity.*
- *Control loop Decision (optional) – A control loop may follow a decision, a line of action that guides its behavior. Having this guiding decision is not mandatory for any control loop, the behavior can be determined at a tactical level without following a decision, yet it may be applicable for some cases. Examples for such conceptual decisions at different levels may be: “Select X as a partner”, “refund the customer”, “increase capacity even at higher costs.”*
- *Control loop Implementation – To reach the required intent, a set of activities needs to be set and activated. These activities correspond to one of the operational models. For example, if the FOCAL model is used, the types of activities may be derived from the model (e.g. Observe, Compute, Act...). The activities follow a flow that may be static or dynamic. In many of the current flows, there is a linear, unconditioned flow of activities, such as: monitor->analyze->plan->execute. However, flows can be dynamic, and the next activity can be decided by the result of the previous one. For example, if a monitoring activity discovers a potential severe SLA violation, a customer account system may be involved in deciding on the course of action.*

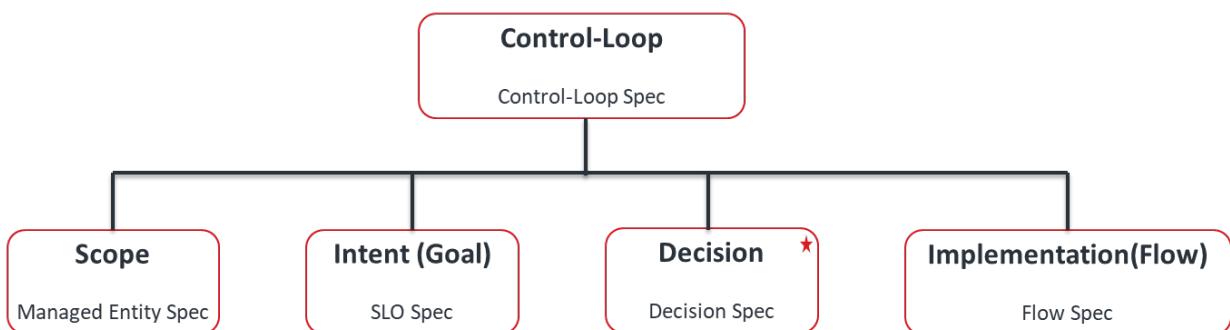
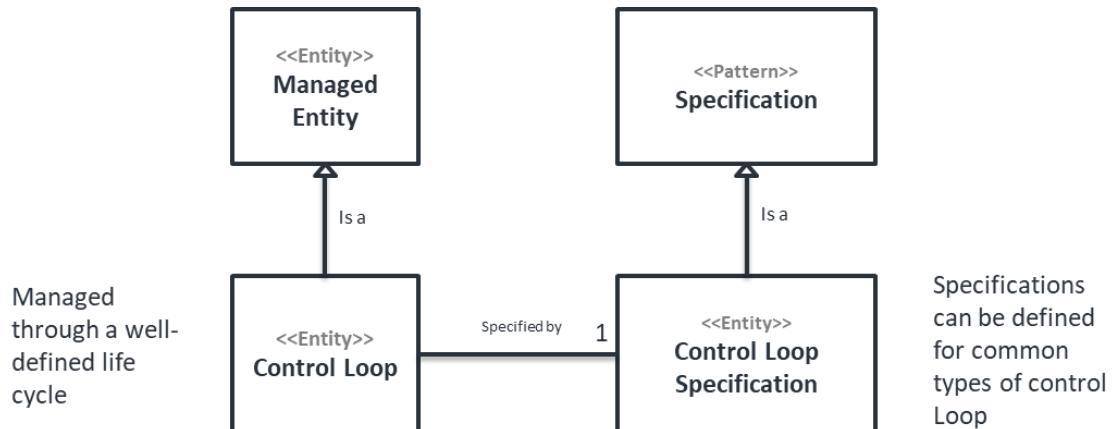


Figure 7-7 Control Loop high-level conceptual model

## Control Loop is a Managed Entity

Control Loops have the following qualities:

- *are managed entities*
- *are life cycle managed*
- and use the *specification pattern*



**Figure 7-8 Control Loop ABE**

# 8 Knowledge and Intelligence

With the improvement of computing power, algorithm innovation, and the accumulation of massive data, Artificial Intelligence technologies are playing an increasingly important role in many fields.

Telecommunications is one such field and has a massive scope and potential for applying AI technologies. How to use these powerful analysis, judgment, and prediction capabilities, provided by AI algorithms is now the focus of this section of the AN technical architecture guide. It aims to describe how AI, and more specifically machine learning approaches, can empower network elements, networks, and management systems to make better decisions and also looks at how the AN Framework can integrate this “network intelligence” with the planning, construction, maintenance, operation, and optimization of telecom networks operations management systems.

## 8.1 AI in AN Technical Architecture

In future automated networks, AI/ML will become ubiquitous as a core component of network operations and maintenance intelligence. Considering the hierarchical characteristics of networks and management systems and the differences in IT resource requirements at each stage of the AI application process (knowledge acquisition, training, and inferring), it is, therefore, necessary to build AI capabilities in each layer of the network to support an automatic AI-driven closed loop at each layer, and thereby achieve intelligence for different scenarios, and ultimately to realize autonomous networks.

### Two modes of AI usage:

- *Development Mode (Sandbox): utilizing cloud computing resources and network data assets for AI-related data governance, model training, and providing AI models for the application layer.*
- *Runtime Mode (Production): The AI model provided by the development mode is used for real-time inferring and intelligent decision-making at each level of the architecture to achieve an autonomous closed loop at the network layer, and AI capabilities can be provided to internal and external systems to provide intelligent capabilities to the upper-level operational applications.*

### Layered Intelligence

- *“Cloud + AI” – an open Knowledge and Intelligence platform, accelerate network AI innovation and development.*
- *“Management layer + AI” – involving AI Inference framework, promote intelligent analysis and decision capability in the management layer.*
- *“NE + AI” – Make the network real-time perceptible with built-in AI chips and sensors, enabling data collection acceleration and rapid inference at the edge.*

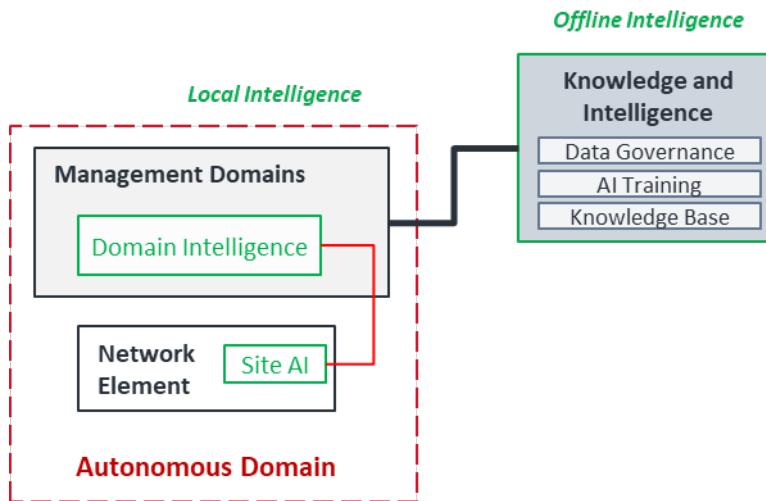


Figure 8-1 Layers of AI and Network Intelligence in AN Reference Architecture

The Knowledge and Intelligence platform is a centralized design and development platform and is the source for the digitization of operations, network and human expert knowledge. Since AI model training typically requires significant computing power, it is generally recommended that knowledge extraction, data governance, and model training be performed in a cloud environment. At the same time, cloud AI platforms require the ability to publish and share models, which can reduce redundant development, facilitate model/knowledge sharing, replication and promotion in multi-location and build AI ecosystem, and be the "knowledge center" and "library" of the whole network. Cloud AI also needs to support cross-layer, cross-domain, and cross-vendor AI capability construction to reduce duplication of investment.

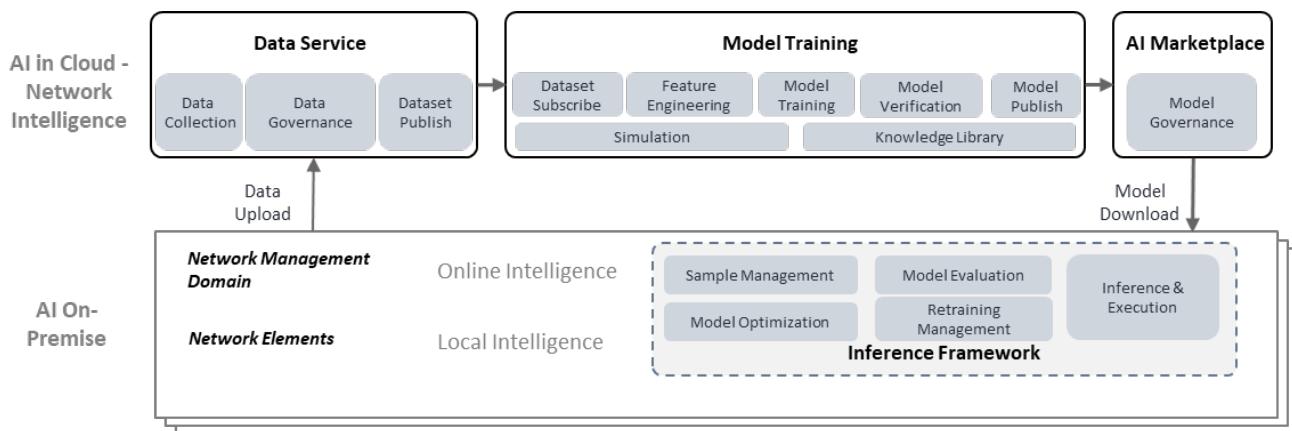
The localized Intelligence includes management *domain Intelligence* and network element intelligence ("Site AI"). Management *domain Intelligence* provides an online AI inference service for the hierarchical autonomous network, focusing on real-time collection and filtering of network data and it is central to the overall network intelligence implementation. Due to the limitations of AI computing power and storage space, it mainly focuses on *local* real-time perception analysis and decision processing.

## 8.2 Developing and using AI Models

The development and deployment of AI models in telecoms face significant challenges:

- *High data sensitivity*
- *Difficulty in obtaining sample data*
- *High requirements for model accuracy*
- *Model generalization is difficult*
- *Demanding inference environment*

An end-to-end closed loop process is therefore required for AI development and application to achieve better model precision and outcome.



**Figure 8-2 Closed Loop process for Model Development and Inference**

### Data Service

Data Service provides security guarantee for the whole life cycle of data collection, transmission, management and release, supports data traceability and data trajectory visualization through log management, supports data life cycle process traceability and auditable operation, and prevents malicious denial. In the data collection phase, it desensitizes personal private data and provides differentiated control policies for different information security levels and personal private data.

Data Service provides a one-stop toolchain for data collection, storage, analysis, modeling, governance and publishing, as well as intelligent data labeling capabilities to help users quickly complete high-quality data preparation for model development and training in the telecom field.

### Model Training

The technical learning curve for AI model development in telecoms is high, requiring developers to understand both AI and network. Experts in telecoms have fewer AI skills and experience to draw from. With the large number of available AI algorithms and the wide range of ML toolsets, it leads to high “trial and error” costs and low overall efficiency. Therefore, it is required to establish a one-stop training platform and development process to ensure efficient model development and improved efficiency through a “wizard”-style development interface.

The basic model development process includes:

- *Dataset Management*

Imports data used by the model to be trained; provides the maximum value, minimum value, average value, standard deviation, and visualized data distribution tool to evaluate and analyze data quality; and provides the intelligent error correction tool to correct data errors.

- *Feature Engineering*

Provides tools such as chi-square, information entropy, and normalization to analyze and extract features from sample data and provides visualized feature extraction and data analysis results.

- *Model Training*

Web style IDEs are required, online and offline collaborative model development should be supported, and mainstream AI algorithm frameworks such as TensorFlow, PyTorch, MXNet, and vendor-developed AI

framework MindSpore<sup>8</sup> also considered to be supported. Tasks can be concurrently submitted for multiple model “trainings”.

- *Model Verification*

Provides a model verification environment to verify the model accuracy.

- *Model Publish*

Provide a one-click publish function to publish the model to Marketplace for download.

## **Marketplace**

- *Provide model repository function, provide a publishing place for the developed models, manage and monitor the models running on the ground and re-training management.*
- *Support the browsing and ordering of application models, automatic update and installation deployment, and grayscale publishing. Support convergence and scheduling of multiple models.*

## **Inference Framework**

- *Inference execution: model compatibility should be considered in the inference stage, e.g., supporting models developed from deep learning frameworks like Tensorflow, machine learning ones such as MLLib, and even support for trained models that could be running on Python/Java-based AI framework. Support multi-model inference execution, regular scheduling, and provide external API interface.*
- *Model retraining: Create retraining services based on model packages, create retraining tasks by changing input parameters, generate new model packages, and support retraining scenarios with only data changes and cloud-based collaborative retraining scenarios.*
- *Inference evaluation: Support manual or automatic evaluation of inference results by users, support visualization of inference, query history of inference and output statistical reports.*
- *Local Optimization: It supports automatic re-training based on local labeled sample set and incremental learning with online data to ensure model accuracy. Users can flexibly define re-training policies according to AI scenarios (e.g., re-training starts if the model accuracy is lower than 75%), evaluate the model based on local samples, and select the optimal version for the model upgrade.*

## **Standardization is needed for AI Models**

The development of AI models is not a “one-time-only” process. As the network changes so too must the model. It is necessary to unify the management and continuous optimization of the model through the cloud AI platform to ensure that the accuracy of the model meets production requirements. The development of AI-related specifications is key to ensuring efficient collaboration between AI cloud and local clouds, to achieve the efficiency of cloud-local connectivity, and to support the exchange and flow of models and data. AI specifications that are required include AI Data Specifications and AI model specifications. The data specification is used to shield the differences between different vendors' device data, and the model specification is used to unify the operating environment of different vendors' device models to ensure that AI models and knowledge can be flowed and shared.

TM Forum’s AI Project team are advancing proposal in the area of AI Model specification and have published Open API specifications in the area of Data Governance.

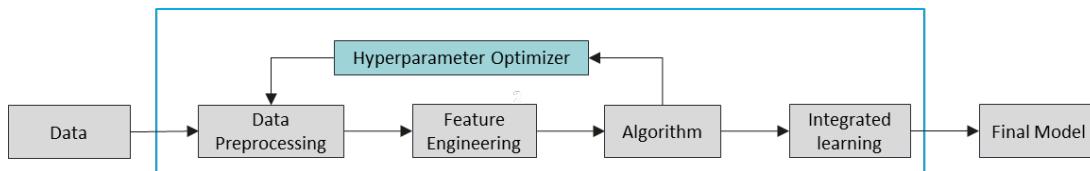
---

<sup>8</sup> <https://github.com/mindspore-ai/mindspore>

## 8.3 Relevant AI Technologies for AN scenarios

### Automated Machine Learning

Given the high technical learning curve, long development cycles, and high cost associated with “traditional” machine learning, it is necessary to introduce Auto ML capability to improve the efficiency of the development and training process as soon as possible.



**Figure 8-3 Automated Machine Learning ( AutoML )**

Such an approach could reduce the data modeling cycle from months to days, accelerating the incubation of AI models in the production platforms.

### Core Platform Features

- *Support for pipeline, incremental, and distributed hyper-referential optimization of search, enabling rapid improvement of model performance to expert levels, reducing time spent from days to hours.*
- *Automatic feature engineering, automatic feature search algorithms based on reinforcement learning, temporal key feature extraction algorithms, lowering the development entrance threshold, and allowing a beginner to develop high-quality models.*
- *The aforementioned AutoML model is made more stable through multi-algorithm model integration and optimization.*

### Transfer learning

Transfer learning addresses the following issues.

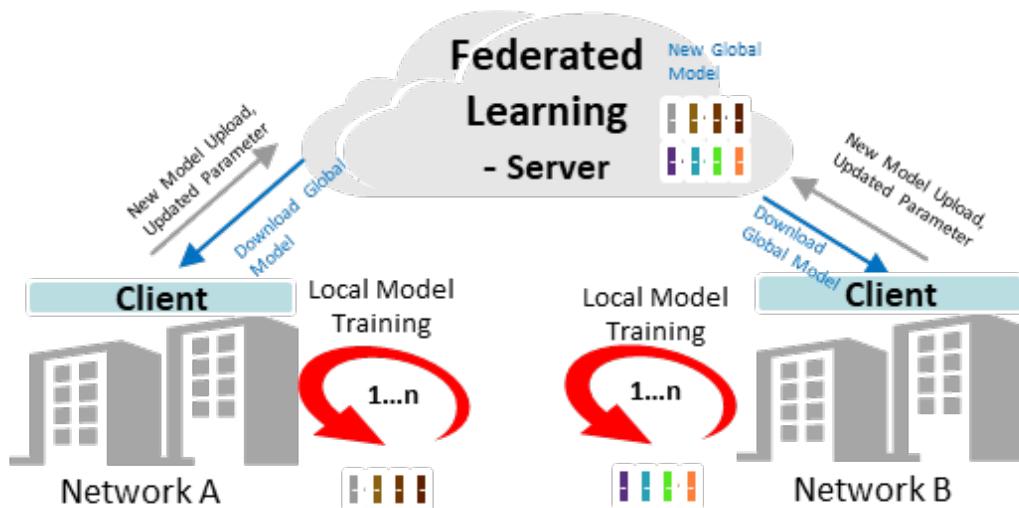
- *The distribution of data for new application scenarios is different from that of generic scenarios, and generic models cannot be directly reused.*
- *Less training data for new application scenarios, poor accuracy of training models from scratch*
- *Low computational resources make it difficult to train models from scratch, e.g., edge computing scenarios.*

### Core Technical Points:

- *Provide a migration assessment methodology for datasets to support the selection of datasets and data for migration.*
- *There are more than a dozen transfer learning methods that industry can learn from, such as deep learning methods like DAN, MMD, DDC, and machine learning methods like TCA, KMM, and PCA, which can be used to support the migration of AI models like DC PUE, Crystal Speech, and others.*

### Federated Learning

The application of federal learning is designed to solve the problem of data silos within carrier networks while protecting data privacy from leaving the local area, thereby improving AI model accuracy.

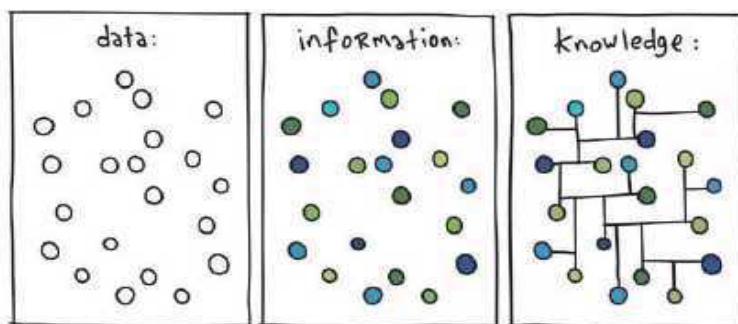


**Figure 8-4 Federated Learning**

- *Training only with local data, not exchanging the data itself, exchanging updated model parameters with encryption.*
- *Use the data from different environments to train the model, and then send it to the device after collecting and updating the model to improve the generalization ability.*
- *Leveraging the idle computing power of edge devices for training to improve model personalization and adaptation training efficiency without affecting core business experience.*

### Knowledge Graph

There are two layers of Artificial intelligence: the perceptual layer and the cognitive layer. The perceptual layer, which is the computer's visual, auditory, tactile, and other perceptual capabilities. Humans have made important breakthroughs in the field of voice recognition, image recognition, and other perceptual fields, and machines have become closer and closer to humans in terms of perceptual intelligence. The second layer is the cognitive layer, which refers to the machine's ability to understand the world and have the ability to think. The cognitive world is realized through the accumulation of a large amount of knowledge. In order for the machine to gain cognitive ability, it is necessary to establish a rich and accurate knowledge base. The knowledge graph is a key measure of the machine's cognitive ability. A *Knowledge Graph* describes concepts, entities and their relationships in the objective world in a structured form, expresses information in the network in a form that is closer to the human cognitive world, and provides an ability to better organize, manage and understand the vast amount of information in telecommunication networks. The knowledge graph aids semantic search, but also shows great power in intelligent question and answer and operation and maintenance assistance. Together with big data and deep learning, the knowledge graph is one of the core drivers of the Internet and artificial intelligence.



**Figure 8-5 Knowledge is connecting the dots (graphic by David Somerville)**

A knowledge graph is a data model that can assemble a "knowledge" network containing large numbers of entities, attributes, and relationships providing experts with a fast and convenient way to retrieve and infer knowledge. The core is to observe and perceive the world through data and realize intelligent services such as classification and prediction and automation. As an important repository of expert knowledge, the knowledge graph promotes information searching, intelligent question and answer, and many other intelligent applications.

## 8.4 AI in common telecoms use cases

In this section we identify four key telecom AI use case categories :

1. *Perception and Prediction*
2. *Detection and identification*
3. *Control optimization*
4. *Process optimization*

### **Perceptual and Prediction**

Taking BTS<sup>9</sup> energy saving as a typical example, where energy saving is achieved by controlling the switching time of BTS carrier amplifiers, the key is to make a judgment based on cell traffic trends. In this case, it is unrealistic to rely on the manual or human prediction of massive BTS traffic. The LSTM<sup>10</sup> neural network algorithm can predict the future PRB traffic time series through the historical time series of PRB traffic, and then configure the energy-saving switching strategy of each base station.

### **Detection and Identification**

Typical applications such as fault identification and root cause location, which is a long-standing problem of network operations and maintenance, a base station power failure will lead to transmission, base station decommissioning, power environment and other 11 work orders, how to quickly locate the root cause, reduce invalid work order distribution to save maintenance costs need to be addressed. Frequent item mining, clustering, knowledge graph are all AI technologies that can achieve root cause location.

### **Control Optimization**

Control optimization is a common field of AI technology in the field of telecommunications, such as 5G antennas, horizontal beam, vertical beam, antenna down-angle, antenna azimuth constitute tens of thousands of combinations, artificially need weeks to find a better combination of parameters, but the introduction of AI can rely on reinforcement learning and deep learning to quickly find the optimal combination from a large number of parameter combinations, to maximize resource utilization.

### **Process Optimization**

Robotic Process Automation (RPA) is an AI application where people configure robots to mimic and integrate the interaction between humans and data systems and thus automate business processes. By using a user interface (UI) to capture data and restore human operating processes, machine learning and deep learning are used to solidify a large number of repetitive operating processes in the telecom network operations and maintenance process, replacing humans with machines to achieve 24/7 automated operations. Meanwhile, in certain complex scenarios in the telecom field, it is necessary to combine the above multiple capabilities to solve specific problems. This combination, termed "AIOps", has gradually

<sup>9</sup> Wireless - Base Transceiver Station (BTS)

<sup>10</sup> Long Short-term Memory (LSTM)

© TM Forum 2021. All Rights Reserved.

been accepted in IT and now the CT area. AIOps is a key service in the evolution of telecom network intelligence and a key capability that enables automated driving networks. The service capabilities used include KPI anomaly detection, log anomaly detection, fault identification and root cause localization, all of which can be flexibly arranged and combined to suit different business usage scenarios.

## 8.5 AIOps Service Management

### The need to operationalize AI

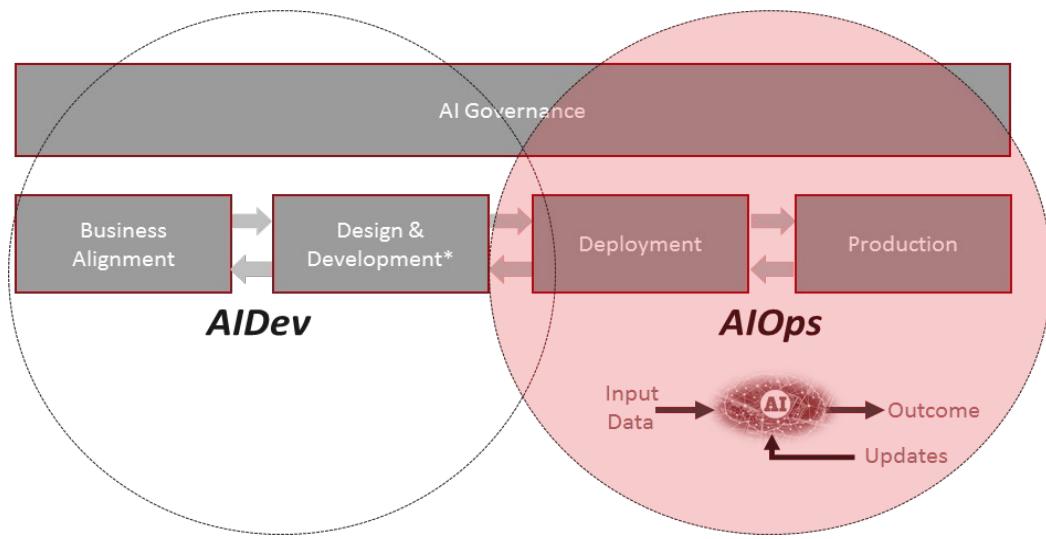
Large-scale deployments of AI create new opportunities but also additional challenges such as how to operationalize AI, i.e., how to deploy, operate, control, maintain and govern hundreds or thousands of AI components which will eventually form part of core IT and network systems architecture. Unlike traditional software, AI-based components may reason, make decisions, predict, learn, and evolve autonomously when exposed to new input data. AI components are more “black box”, i.e., opaque, and non-deterministic in behavior. While this enables a more agile set of operating capabilities within the CSP it also significantly increases risk. To that end, AI operations need to be compliant with emerging regulatory frameworks for AI aiming to enable a trustworthy, responsible, reliable and secure development and adoption of AI applications and services. In the context of autonomous networks, this applies to individual AI software components, entire AI-driven autonomous domains, or even higher level cross-domain or business level processes. Sometimes it may be necessary to control, or throttle, the level of autonomy that a given component has at a point in time, and such decisions are informed by the operational processes used.

To address these challenges, TM Forum is leading an initiative to formulate an industry-agreed framework called “AIOps Service Management Framework”, which focuses on reengineering the software and service lifecycle processes required to operationalize AI software at scale. This framework enables operations teams, process owners and business users to exploit AI safely and properly, thereby maximizing its benefits, mitigating its inherent risks, and ensuring the appropriate level of quality, reliability, transparency, and regulatory compliance. The AIOps Service Management Framework complements the Autonomous Networks Framework in that the AI operational processes defined in the former are a subset of the overall process required for autonomous operations defined in the latter. Both frameworks address ‘operations’ but AIOps Service Management Framework is squarely concerned with *AI Applications Operations* whereas the AN Framework develops the business and technology architecture more generally across business, service, and resource operations.

### AIOps in the Software Lifecycle

The AIOps Service Management Framework is agnostic to any specific software management methodology and is general enough to be inclusive of most software engineering lifecycles, including both traditional (waterfall or V-Model) and agile methodologies. AIOps SMF scope is on re-engineering operational processes in the following areas:

- **Deployment**, representing the bridge or gate between the Development and Production stages.
- **Production**, the subset of processes aimed to operate and maintain the services in Production live environments, including the **Maintenance** process.
- **Decommission**, addressing the removal of a system release from Production.
- **Operations Governance**, which includes the subset of governance processes that are needed to manage and govern specifically the Deployment and Production stages of the lifecycle.



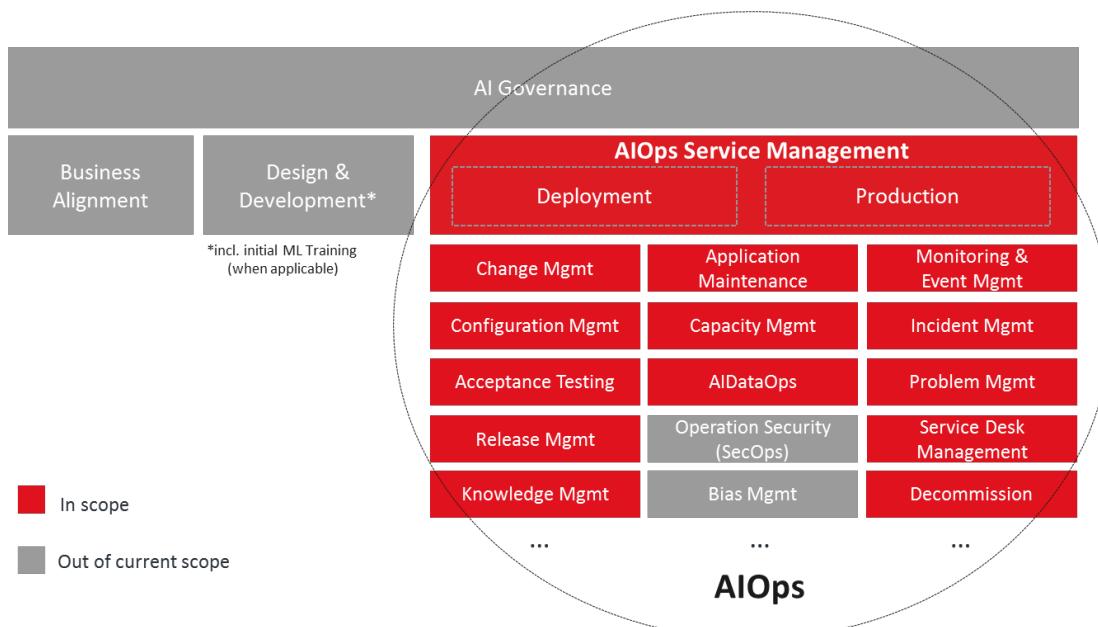
\* incl. the initial ML Training (when applicable)

**Figure 8-6 AIOps in a general software lifecycle process**

Each AIOps process within the lifecycle will describe the “as-is” process, provide a gap analysis, along with AIOps process reengineering guidelines and use cases. Figure 8-6 above shows the stages of the lifecycle covered by the AIOps Service Management framework mapped on a general software development lifecycle.

### AIOps Service Management Framework

TM Forum’s AIOps Service Management Framework redesigns the service management processes needed to safely and effectively deploy, operate and govern AI components that are integrated within network and IT systems and architectures. These management processes need to cater to a ‘blend’ of AI and traditional applications that run side-by-side in CSPs’ operations. The figure below shows the AIOps Service Management framework highlighting in red the processes that are currently in scope (IG1190).



**Figure 8-7 Processes in scope of the AIOps Framework**

The dynamic nature of AI software and its capability to learn and evolve autonomously, which creates a continuum between the Deployment and Production stages. AI components move from the ‘deployable’ state to the ‘live’ state and vice versa, blurring the lines between Deployment and Production. For this reason, we consider the Deployment and Production processes to be part of AIOps and we do not assign them to any specific stage of the lifecycle. This will give companies greater agility in their operations, as they can organize their processes and assign responsibilities to the teams according to their strategy, ambition level regarding network autonomy, organizational choices, and operational context. The table below lists the processes that are currently within the scope of the AIOps Service Management Framework.

**Table 6 Processes defined within AIOps Service Management Framework**

Process	Description
<b>Configuration Management</b>	Configuration Management ensures that all components (also called Configuration Items, CIs) of systems and services are uniquely identified, baselined and maintained and that changes to them are controlled across the whole service lifecycle.
<b>Change Management</b>	Change Management ensures a smooth transition of all changes to Production live environments. It minimizes the risks, preserves the quality of service, avoiding incidents and outage, and prepares the final users/consumers to adopt the new capabilities/features.
<b>Acceptance Testing</b>	Acceptance Testing ensures the overall quality against the expected targets of all new and existing updated software and services, before declaring and certifying them as “operations ready” and “deployable” to the Production live environment.
<b>Release Management</b>	Release Management plans and manages the deployment of any software release (i.e., package of changes) from Development to Production environments (and in general, to any relevant environment).
<b>Knowledge Management</b>	Knowledge Management ensures that reliable and complete information and knowledge is available to the right parties (people, consumers, 3rd parties...) at the right time throughout the service lifecycle.
<b>Monitoring &amp; Event Management</b>	Monitoring & Event Management, monitors, detects, filters, and correlates all relevant events occurring throughout the Production environments and initiates the corresponding activities to respond to and address those events.
<b>Incident Management</b>	Incident Management addresses all events that have or could have relevant impacts (outages, incidents, defects/bugs, operational issues, quality degradation ...) on the services and, in line with their priority, manages the recovery of the service operations according to the agreed SLAs, minimizing the impacts on users and business processes and managing the necessary communication.
<b>Problem Management</b>	Problem Management is responsible for the diagnosis of the root cause of incidents, issues, vulnerabilities, and weaknesses related to systems and services, and, in line with their priority, ensures the permanent resolution of those problems, when appropriate.
<b>Service Desk Management</b>	Service Desk Management is the process responsible for managing the lifecycle of all different types of demands (Service or User Requests) submitted by the end-users to the Service Desk.
<b>Application Maintenance (incl. ML Training)</b>	Application Maintenance ensures the proper modifications of software and its components after delivery to Production in order to improve their performance, quality, efficiency, or other relevant attributes. In AIOps, this process includes also the ML Training and re-training process (when applicable), predictive and self-healing maintenance activities.
<b>Capacity Management</b>	Capacity Management ensures that cost-justifiable capacity of services and related components (software, hardware, network etc.) is able to deliver, the agreed service level targets, presently and in the future, in a timely and effective manner in the specific context of the CSP.
<b>AIDataOps</b>	Being data, the fuel driving the operations and evolution of AI software, AI Data Operations management (AIDataOps) become vital to ensure that the right data in Production are properly collected, processed, transformed, stored, safeguarded, available and accessible at the right time, according to the business needs, laws, regulations, security policies and efficiency drivers.

# 9 Intelligent Network Infrastructure

Some challenges exist in the infrastructure layer in both physical and functional aspects. Autonomous Network as a term implies a more effective infrastructure, and this infrastructure may take several forms:

- *Intelligence in equipment design*
- *Digital awareness and intelligence enabling*

## 9.1 Intelligence in Equipment Design

### Lightweight Equipment in Wireless Sites

To simplify site construction and reduce site construction costs, high-density and integrated blade devices are required. Higher bands are used for 5G scenarios for the development of network services. Higher bands and more services require more sites, so more pole sites will be constructed in the 5G era.

To support fast and efficient deployment of massive sites, the following requirements should be met:

- *No reconstruction of site infrastructure.* For example, do not add poles to or reinforce antenna installation platforms and do not expand power supply units (PSUs)
- *Improve energy efficiency and reduce energy consumption.* For example, keep improving the low-power design for radio frequency (RF) and baseband unit (BBU) modules; power on and off devices based on the service volume.
- *Design devices in all standards and allocate spectrums on demand.* For example, enable 5G in RF and baseband modules; support spectrum sharing between LTE and New Radio (NR) in all scenarios.
- With faster bit-rate growth of networks, base stations need to manage radio resources more accurately, improving the *real-time data analysis capability* of wireless networks and maximizing resource utilization.

### Fiber Network Simplification

Usually, the consideration of network simplification is holistic and end-to-end. For example, in the envisaged autonomous optical network, the following design criteria should be followed:

- One fiber to meet all services, one optical network covers home users, enterprise users, and 5G transport scenarios.
- One device to meet all services, especially for the convergence site.
- Adopting the E2E dynamic slicing technology to provide comprehensive network protection and differentiated SLAs for various services.
- Employ new architectures and technologies to sharply improve network resource utilization and performance, e.g., all-optical backplane.

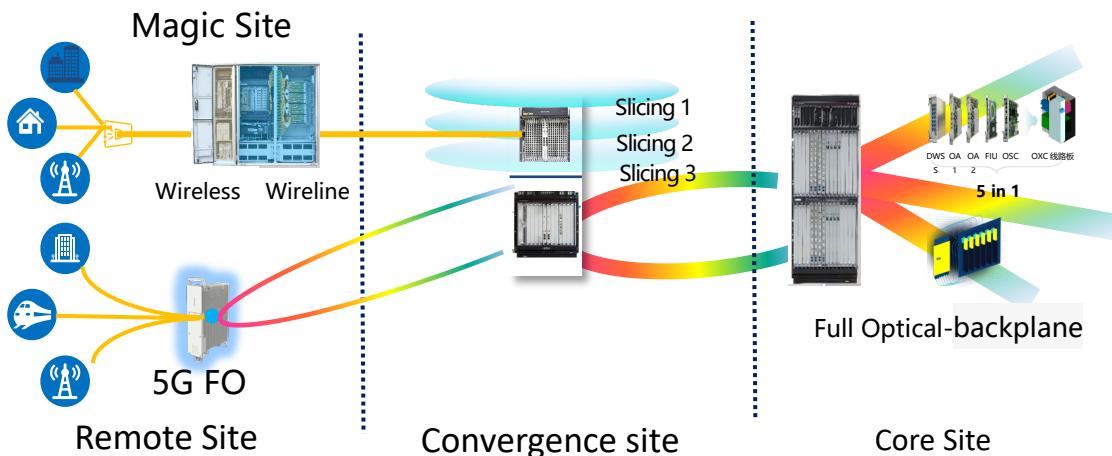


Figure 9-1 Simplification of Optical Network

## 9.2 Digital Awareness and Intelligence Enabling

### Network Awareness

Network awareness is the basic prerequisite for closed loop network services and automatic driving. Therefore, devices must have sensing components, and the capability of sensing resources, services, and surrounding environments are increasingly stronger. Multi-dimensional real-time sensing must be provided, covering service flows, resources, topology status, O&M events, and power consumption.

### Site Intelligence Inference Framework

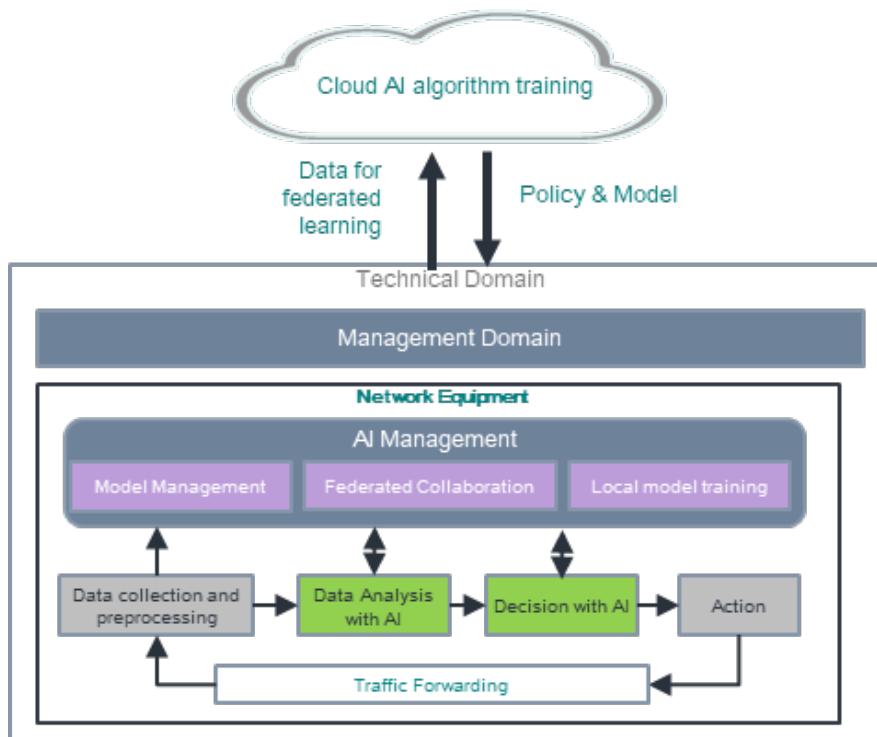


Figure 9-3 Site Intelligence at NE

As described in the previous section (Knowledge and Intelligence), the generic AI architecture of N will be layered, including the centralized AI knowledge base and training service, and the local inference components. Site Intelligence (SI) follows the same framework and especially benefits from the federated learning technology, which resolves model generalization issues under customer data privacy protection and improves the business identification rate. Federated learning uses partial data of multiple nodes to learn global models, fully exploiting the data value of different nodes and improving model training efficiency.

### 9.3 Infrastructure Requirements

The study and regulation of infrastructure evolution are mostly handled by specific standard organizations in a particular domain, e.g., 3GPP for wireless access/core, ETSI F5G for fixed networks, ETSI NFV for virtualized functions. The TM Forum cooperates with these organizations on the cross-domains aspects of the network transformation towards AN. The TM Forum presents the need for simplification of infrastructure as requirement input for AN evolution (see Table 7). The requirements related to the field need to be highlighted by the corresponding domain standard organizations.

**Table 7 Infrastructure Requirements for SDO Cooperation**

	3GPP	ITU-T	ETSI	TM Forum
<b>Wireless Site</b>	Lightweight equipment Real-time NE awareness AI inference	AI inference		AI inference
<b>Core Network</b>	Normalized protocol Real-time NE awareness AI inference		Simplified architecture	AI inference
<b>IP network</b>		AI inference		AI inference
<b>Fiber Network</b>		Lightweight equipment Simplified architecture AI inference	Simplified architecture	AI inference

## 10 Conclusions

An architecture is useful if it gives some flexibility in addressing unanticipated changes in the future. If an architecture has some malleable, morph-able quality that allows sufficient freedom in adaptation then it allows the details to be defined by the implementers and to discover particular implementations that meet their needs. Network complexity is not simplified by removing complexity. It is simplified when it is “sidestepped”, compartmentalized into autonomous domains, and then handled in a robust, consistent fashion (intent, closed loop automation) where the closed loops deal with known failures and end states rather than attempt to perfectly predict all eventual end-states. The complexity still exists of course but only within the networks bounded context. The models we employ do not simply employ an abstracted view of the real complexity, but rather attempt to model the aspects that matter. When we say AN can “deal with complexity” in end-to-end networks we mean that the AN has developed a certain *resilience* to the scale of the endless permutations of failure and the even greater number of optimal states for networks. In this sense, we say that the AN is context-aware. Knowing what you need to know when you need to know it. This awareness brings clarity and purpose to the orchestrated flows (closed loops) that drive operations. This AN technical architecture advocates the understanding of the platform's capabilities via closed loop flows and the decoupling and “weak-linking” of platforms using intent. The separation of domains into modular, redundant and diverse (multi-vendor) components allows AN to handle failure, risk in the complex networking world.

The AN Project Deliverables Guide (IG1260) summarizes the project's published deliverables and also gives an outlook for how the TM Forum's AN Project progresses into next year. We expect that future deliverables will go into more detailed modeling aspects of Intent and Control Loops but also make contributions on API standards in these areas.

The future of work and the future of network automation may evolve in a similar path towards autonomy, bringing more freedom for human workers. Autonomy gives some direction (intent), requires a little supervision (human operations), but expects that the tasks are then completed by the machine with no human intervention. In traditional control loop approaches, we have relied on people to define policies. For Autonomous Networks, we expect that control loops will eventually become self-tuning and fully adaptive. However, a degree of human oversight is needed. At some stage, our ability to remove humans from the loop will not be a scientific decision but a business one [TATAN]. However, full autonomy does seem a high bar to reach right now, but with steady progress, level by level, we can reach the self-managing and self-governing future of Autonomous Networks.

# 11. Administrative

## 11.1 Abbreviations

Abbreviation	Descriptive
ABE	Aggregate Business Entity
AI	Artificial Intelligence
AN	Autonomous Networks
API	Application Programming Interface
AutoML	Automated Machine Learning
CSP	Communication Service Provider
Framework	<p>TM Forum Frameworx is a suite of standards-based tools and best practices which provide the blueprint for effective business operations, enabling companies to assess and improve performance by using a proven, service-oriented approach to implementation.</p> <p>Frameworx (with an 'x') describes the entire suite which includes all four core frameworks (with a 'k').</p> <p>Frameworx consists of:</p> <ul style="list-style-type: none"> <li>• <b>TM Forum Business Process Framework (eTOM)</b> – a comprehensive, industry-agreed, multi-layered view of the key business processes required to run an efficient, effective and agile digital enterprise</li> <li>• <b>TM Forum Information Framework (SID)</b> – provides standard definitions for all the information that flows through the enterprise and between service providers and their business partners</li> <li>• <b>TM Forum Application Framework (TAM)</b> – provides a common language and means of identification for buyers and suppliers across all software application areas</li> <li>• <b>TM Forum Open APIs</b> – a complete library of standardized application program interfaces to integrate existing legacy end-to-end management applications as well as future platform services</li> </ul>
HIP	Hybrid Infrastructure Platform
IPR	Intellectual Property Rights
ML	Machine Learning
NE	Network Element
OODA	Observe, Orient, Decide, Act
RAND	Reasonable and Non-Discriminatory (licensing terms)
SDN	Software Defined Networking
SDO	Standards Developing Organization
SID	Shared Information and Data (TM Forum Information Framework)
SLA	Service Level Agreement
TA	Technical Architecture

## 11.2 References

#	Title	Organization
ANWP1	<a href="#">Autonomous Networks: Empowering Digital Transformation for The Telecoms Industry.</a>	TM Forum
Brookshire	<a href="#">Jonathan D. Brookshire 2014 Enhancing Multi-Robot Coordinated Teams with Sliding Autonomy</a>	Paper
CSA	Management services for communication service assurance; Requirements (3GPP TS 28.535)	3GPP
ENI005	<a href="#">System Architecture Experiential Networked Intelligence (ENI) - System Architecture</a>	ETSI
FOCALE	<a href="#">FOCALE – A Novel Autonomic Networking Architecture - Strassner, Agoulmine, Lehtihet 2006</a>	Paper
GB1002	<a href="#">GB1002 Artificial Intelligence User Stories &amp; Use Cases R19.0.0</a>	TM Forum
GB999	<a href="#">GB999 ODA Production Implementation Guidelines R19.0.0</a>	TM Forum
GB999	GB999 <a href="https://www.tmforum.org/resources/how-to-guide/gb999-oda-production-implementation-guidelines-v4-0/">https://www.tmforum.org/resources/how-to-guide/gb999-oda-production-implementation-guidelines-v4-0/</a>	
GSMAWP	GSMA White Paper presented @MWC 2019, June 26-28 Shanghai (AI and Automation – Future Networks)	GSMA
IDMS	Intent- driven management service for mobile networks (TR 28.812/TS 28.312)	3GPP
IDMS	Management Data Analytics Service (see TR 28.809)	3GPP
IDMS	Autonomous Network Levels (see TR 28.810/TS 28.100)	3GPP
IETF	<a href="#">Intent-Based Networking - Concepts and Definitions</a>	IETF
IG 1167	ODA Functional Architecture v6.0.0ODA-IM -Tayeb	TM Forum
IG1139	<a href="#">IG1139 Business Rationale and Technical Overview for Orchestration and Autonomic Control Loops R16.0.1</a>	TM Forum
IG1190	AIOps Service Management Framework <a href="https://www.tmforum.org/resources/toolkit/ai-operations-toolkit">https://www.tmforum.org/resources/toolkit/ai-operations-toolkit</a>	TM Forum
IG1193	<a href="#">IG1193 Cross-Industry Autonomous Networks – Vision and Roadmap.</a>	TM Forum
IG1218	<a href="#">IG1218 Autonomous Networks – Business requirements &amp; architecture.</a>	TM Forum
IG1229	TMF IG1229 Autonomous Networks Guiding Principles	TM Forum
Kelleher2018	Kelleher et al in <a href="#">"Empowering Self-Driving Networks"</a>	Paper
Li	<a href="https://tools.ietf.org/pdf/draft-li-nmrg-intent-classification-03.pdf">https://tools.ietf.org/pdf/draft-li-nmrg-intent-classification-03.pdf</a>	IETF
NGMN2020	<a href="#">5G White Paper 2</a> by NGMN Alliance July 2020	NGMN
RFC7575	Autonomic Networking: Definitions and Design Goals <a href="https://tools.ietf.org/html/rfc7575">https://tools.ietf.org/html/rfc7575</a>	IETF
Sloman	<a href="#">Domains: A framework for structuring management policy</a> Morris Sloman Imperial college	Paper
Strassner	<a href="#">The Use of Control Loops in Autonomic Networking</a> (2016)	IETF
TATAN	Towards A Truly Autonomous Network – Rakuten Positioning paper P. Imai, P. Harvey, T. Amin.	ITU
TR262	<a href="#">TR262 Management Platform Blueprint and Application to Hybrid Infrastructure R17.5.1</a>	TM Forum
TR262	Management Platform Blueprint and application to Hybrid Infrastructure (April 2018)	TM Forum

#	Title	Organization
TR270	Trust Challenges and Opportunities <a href="#">TR270</a> <a href="https://www.tmforum.org/resources/technical-report/tr270-trust-challenges-and-opportunities-r17-5-0/">https://www.tmforum.org/resources/technical-report/tr270-trust-challenges-and-opportunities-r17-5-0/</a>	TM Forum
Y3172	Y.3172 : Architectural framework for machine learning in future networks including IMT-2020	ITU

## 11.3 Document History

### 11.3.1 Version History

Version Number	Date Modified	Modified by:	Description of changes
0.1	20-Aug-2020	Kevin McDonnell	Initial Draft. Sections 1,3,6,7,8,11,12,13,14
0.36	04-Oct-2020	Kevin McDonnell	Final draft for team review (Full history omitted)
1.0,0	23-Nov-2020	Alan Pope	Final edits before publication
1.0.0	18-Jan-2021	Adrienne Walcott	Updated to reflect TM Forum Approved Status
1.0.1	28-Apr-2021	Kevin McDonnell	Split v1.0 into 3 Guides: IG1230, IG1230A, IG1230B. Use cases are moved to 1230A. Industry Efforts moved to 1230B. Old Chapter 1 and 2 merged. Revisions, Errata all sections.
1.0.2	04-May-2021	Luca F Varvello	Added sub-section on AIOps
1.0.5	26-May-2021	Kevin McDonnell	Incorporate Team Review comments
1.1.0	28-May-2021	Alan Pope	Final edits prior to publication

### 11.3.2 Release History

Release Status	Date Modified	Modified by:	Description of changes
Pre-production	23-Nov-2020	Alan Pope	Final edits prior to publication
Production	18-Jan-2021	Adrienne Walcott	Updated to reflect TM Forum Approved Status
Pre-production	28-May-2021	Alan Pope	Final edits prior to publication
Production	26-Jul-2021	Adrienne Walcott	Updated to reflect TM Forum Approved Status

## 11.4 Acknowledgements

This document was prepared by members of the TM Forum Autonomous Networks project.

### 11.4.1 Guide Lead & Author

Member	Title	Company
Kevin McDonnell	Senior Director, Intelligent Automation	Huawei

#### 11.4.2 Main Contributors (Alphabetically)

Member	Title	Company
Azahar Machwe	OSS Automation	BT Group plc
Dave Milham	Chief Architect	TM Forum
James O'Sullivan	Product Director, Intelligent Automation	Huawei
Jörg Niemöller	Expert of Analytics and Customer Experience	Ericsson
Luca Franco Varvello	ICT Senior Consultant & Advisor	Huawei
Vinay Devadatta	Practice Head (Innovation & Industry Relations)	Wipro Technologies
Wang Lei	Systems Expert	Huawei
Wang Xu	Systems Expert	Huawei
Xie Yuan	Systems Expert	Huawei
Yuval Stein	AVP Technologies	TEOCO corporation

#### 11.4.3 Additional Inputs (Alphabetically)

Member	Title	Company
Abdul Majid Hussain	Solutions Architect	Telstra
Abinash Vishwakarma	Lead Business Analyst	Netcracker
Andy Corston-Petrie	Senior Research Manager	BT Group plc
Brad Peters	Architect	NBNCo Ltd
Christian Maître	VP Smart City	Orange
Dong Sun	Chief Business Strategist, Digital Transformation	Futurewei Technologies
Emmanuel A. Otchere	Chief Technical Expert VP, Standards & Industry Development	Huawei
Joe Isaac	Principal Architect	Wipro Technologies
Johanne Mayer	Consultant	Ciena
Liu Hongbo	Deputy General Manager, Intelligent Network Center	China Unicom
Luigi Licciardi	Consultant, Executive Advisor	Huawei
Manoj Nair	Senior Solutions Architect, CTO Office	Netcracker
Min He	Chief Architect	Futurewei Technologies
Qiao Zizhi	Senior Engineer, Intelligent Network Center	China Unicom
Steve Iatropoulos	Client & Industry CTO	Microsoft
Tayeb Ben Meriem	Senior Standardization Manager	Orange
Thierry Reynard	OSS Consulting Manager	ETIYA
Vance Shipley	CEO	Sigscale
Zheng Guangying	Autonomous Driving Network Standard Expert	Huawei
Zou Lan	OSS Standard Prime	Huawei

#### 11.4.4 TM Forum Staff

Name	Title	Company
Aaron Boasman-Patel	VP, AI & Customer Experience	TM Forum
Alan Pope	Collaboration Manager	TM Forum
David Milham	Chief Architect	TM Forum
Ian Turkington	VP, Architecture & APIs	TM Forum
W. George Glass	CTO	TM Forum

## 12 Appendix I: AN Technical Architecture Metamodel

Some of the key views of the AN Technical Architecture are described more formally as part of an architectural blueprint with metamodels and UML diagrams. A future TMF deliverable will publish the UML models and diagrams that were developed during the development of this architecture guide.

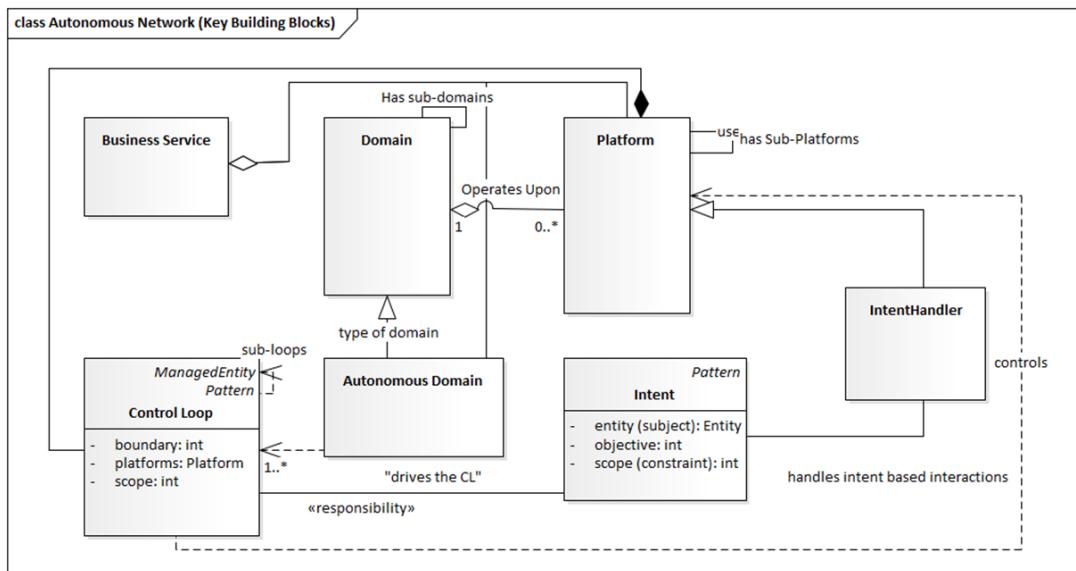


Figure 12-1 Concepts and Relationships

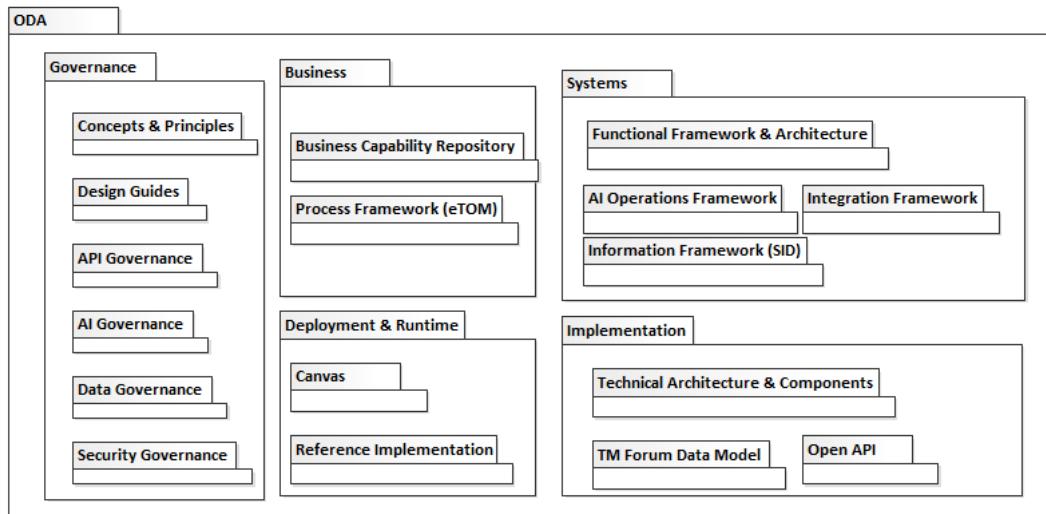


Figure 12-2 Open Digital Architecture Package Diagram

## 13 Appendix II: AN Technical Architecture Guides

**Table 8 Related Guides**

<b>Related Guides</b>		
<b>IG1230A</b>	AN Industry Standards	<p>A brief look at why SDO collaboration is so important to standardization efforts.</p> <p>Autonomous Network practices can improve both the operational efficiency and end-user experience of today's operator use case. This guide describes scenario realizations of operator use cases using an autonomy mindset and, in particular, the mechanisms described in the AN technical architecture (IG1230).</p>
<b>IG1230B</b>	AN Scenario Realizations	<p>Real-world examples of where AN concepts were applied to deliver self-management capabilities.</p> <p>Autonomous Network initiatives are being progressed in multiple standards developing organizations (SDOs), such as ETSI, 3GPP, and GSMA to name but a few. An alignment on Autonomous Network concepts, a shared vision and framework would help the industry to progress in a coordinated way on this important initiative.</p>