Technical Report

TR-436
Access & Home Network O&M Automation/Intelligence

Issue: 1
Issue Date: February 2021

Notice

The Broadband Forum is a non-profit corporation organized to create guidelines for broadband network system development and deployment. This Technical Report is a draft, is subject to change, and has not been approved by members of the Forum. This Technical Report is owned and copyrighted by the Broadband Forum, and portions of this Technical Report may be owned and/or copyrighted by Broadband Forum members. This Technical Report is only available to Broadband Forum Members and Observers.

Intellectual Property

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of this Technical Report if it were to be adopted as a Technical Report, and to provide supporting documentation.

Terms of Use

Recipients of this document may use it only for internal review and study purposes, and to provide to the Broadband Forum the comments and notification requested in the preceding paragraph. Any other use of this Technical Report is expressly prohibited without the prior written consent of the Broadband Forum. THIS Technical Report IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT AND ANY IMPLIED WARRANTIES ARE EXPRESSLY DISCLAIMED. ANY USE OF THIS Technical Report SHALL BE MADE ENTIRELY AT THE USER'S OR IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY USER, IMPLEMENTER, OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS Technical Report, INCLUDING BUT NOT LIMITED TO, ANY CONSEQUENTIAL, SPECIAL, PUNITIVE, INCIDENTAL, AND INDIRECT DAMAGES.

All copies of this Technical Report (or any portion hereof) must include the notices, legends, and other provisions set forth on this page.

Issue History

Issue Number	Approval Date	Release Date	Issue Editor	Changes
1	23 February 2021	23 February 2021	Ken Kerpez, ASSIA	Original

Comments or questions about this Broadband Forum Technical Report should be directed to info@broadband-forum.org.

Editor: Ken Kerpez, ASSIA

Work Area Director(s): Bruno Cornaglia, Vodafone
George Dobrowski

Project Stream Leader(s): Ken Kerpez, ASSIA

Table of Contents

Table of Figures

Table of Tables

Executive Summary

This Technical Report facilitates intelligent and automated O&M for Access & Home networks, which includes:

1) Automatically identify typical anomalies and faults in Access & Home network based on Machine Learning (ML), Artificial Intelligence (AI) algorithms and expert experience, realize rapid troubleshooting, and thus reduce invalid on-site and sporadic issues.

2) Perform predictive analysis based on ML and AI to achieve preventive O&M and self-healing in case of failure.

This Technical Report is an output of the Automated Intelligent Management (AIM) project within the Broadband Forum. Use cases were developed, studied, and are internally posted in the Broadband Forum as part of the development of this Technical Report.

1 Purpose and Scope

1.1 Purpose

TR-436 is the first document in the Automated Intelligent Management (AIM) project stream, to construct access and home networks with automation and intelligence, to achieve higher O&M efficiency and lower OPEX.

1.2 Scope

The scope of this document defines:

1) Logical framework that shows functional blocks and information flows between management entities and the subtended home and access networks.

2) Requirements that support the logical framework.

2 References and Terminology

2.1 Conventions

In this Technical Report, several words are used to signify the requirements of the specification. These words are always capitalized. More information can be found be in RFC 2119 [16].

MUST	This word, or the term “REQUIRED”, means that the definition is an absolute requirement of the specification.
MUST NOT	This phrase means that the definition is an absolute prohibition of the specification.
SHOULD	This word, or the term “RECOMMENDED”, means that there could exist valid reasons in particular circumstances to ignore this item, but the full implications need to be understood and carefully weighed before choosing a different course.
SHOULD NOT	This phrase, or the phrase "NOT RECOMMENDED" means that there could exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications need to be understood and the case carefully weighed before implementing any behavior described with this label.
MAY	This word, or the term “OPTIONAL”, means that this item is one of an allowed set of alternatives. An implementation that does not include this option MUST be prepared to inter-operate with another implementation that does include the option.

2.2 References

The following references are of relevance to this Technical Report. At the time of publication, the editions indicated were valid. All references are subject to revision; users of this Technical Report are therefore encouraged to investigate the possibility of applying the most recent edition of the references listed below.

A list of currently valid Broadband Forum Technical Reports is published at www.broadband-forum.org.

Document	Title	Source	Year
[1] TR-384	Cloud Central Office Reference Architectural Framework	BBF	2018
[2] TR-413	SDN Management and Control Interfaces for CloudCO Network Functions	BBF	2018
[3] TR 103 473 v1.1	Autonomicity and Self-Management in the Broadband Forum (BBF) Architectures	ETSI	2018
[4] TS 103 195-2 (v1.1.1)	Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture; Part 2: An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management	ETSI	2013
[5] White Paper No 16	GANA - Generic Autonomic Networking Architecture – Reference Model for Autonomic Networking, Cognitive Networking and Self-Management of Networks and Services	ETSI	2016

[6]	GS ENI 005 V1.1.1	Experiential Networked Intelligence (ENI);System Architecture	ETSI	2019
[7]	GS NFV-MAN 001	Network Functions Virtualisation (NFV); Management and Orchestration	ETSI	2014
[8]	GS NFV-INF 005	Network Functions Virtualisation (NFV); Infrastructure; Network Domain	ETSI	2014
[9]	GS NFV-IFA 005 V2.1.1	Network Functions Virtualisation (NFV); Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification	ETSI	2016
[10]	GS NFV-IFA 006 V2.1.1	Network Functions Virtualisation (NFV); Management and Orchestration; Vi-Vnfm reference point - Interface and Information Model Specification	ETSI	2018
[11]	GS NFV-IFA 007 V2.1.1	Network Functions Virtualisation (NFV); Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification	ETSI	2016
[12]	GS NFV-IFA 008 V2.1.1	Network Functions Virtualisation (NFV); Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification	ETSI	2016
[13]	GS NFV-IFA 013 V2.1.1	Network Functions Virtualisation (NFV); Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification	ETSI	2016
[14]	TS 123 501	5G; System Architecture for the 5G System	ETSI	2018
[15]	TS 129 500	5G; 5G System; Technical Realization of Service Based Architecture; Stage 3	ETSI	2019
[16]	RFC 2119	Key words for use in RFCs to Indicate Requirement Levels	IETF	1997
[17]	Y.3172	Architectural framework for machine learning (ML) in future networks including IMT-2020.	ITU-T	2019
[18]	GB922	Information Framework (SID)	TM Forum	2016
[19]	TMF633	Service Catalog API REST Specification	TM Forum	2020
[20]	TMF638	Service Inventory API User Guide	TM Forum	2020
[21]	TMF641	Service Ordering API User Guide	TM Forum	2020
[22]	TMF645	Service Qualification API User Guide	TM Forum	2020

2.3 Draft References

The reference documents listed in this section are applicable to this Technical Report but are currently under development and are expected to be released in the future. Users of this Technical Report are advised to consult the source body for current status of the referenced documents or their successors.

Document	Title	Source	Year
[23] WT-411	Definition of interfaces between CloudCO Functional Modules	BBF	TBD

2.4 Abbreviations

This Technical Report uses the following abbreviations:

AAA	Authentication, Authorization, Accounting
AI	Artificial Intelligence
AIM	Automated and Intelligent Management
AIMO	AIM Orchestrator
AL	Automated Loop
AN	Access Node
BBF	Broadband Forum
CF	Collection Function
CFS	Customer Facing Service
CLA	Closed Loop Automation
CPE	Customer Premises Equipment
DCP	Data Collection and Processing
DE	Decision Element
DF	Distributor Function
DL	Data Lake
E2E	End-to-End
EMS	Element Management System
ETSI	European Telecommunications Standards Institute
FB	Functional Block
IAM	Identity Access Management
IdP	Identity Provider
I/O	Input/Output
KB	Knowledge Base
LA	Loop Automation
LCM	Life Cycle Management
MANO	Management and Orchestration
ME	Managed Entity
MF	Model Function
ML	Machine Learning
NBI	Northbound Interface
NE	Network Element
NFVI	Network Function Virtualization Infrastructure
NS	Network Service
OLA	Open Loop Automation

ONF	Open Networking Foundation
PF	Policy Function
PPF	Pre-processing Function
RFS	Resource Facing Service
RL	Reinforcement Learning
SBI	Southbound Interface
SDN	Software Defined Networking
SRC	Source
TR	Technical Report
VIM	Virtualized Infrastructure Manager
VNF	Virtualized Network Function
VNFC	Virtualized Network Function Component
VNFD	VNF Descriptor
VNFM	VNF Manager
WA	Work Area

3 Technical Report Impact

3.1 Energy Efficiency

TR-436 has no impact on energy efficiency.

3.2 Security

TR-436 has no impact on security.

3.3 Privacy

TR-436 has no impact on privacy.

4 AIM Framework Required Capabilities and Characteristics

This section describes the required capabilities and characteristics of the AIM framework.

> **Note:** Throughout this document, the wording “AIM framework” appears in contexts related to different degrees of detail in the solution description and the requirements it has to comply with. A requirement expressed for the whole AIM framework spreads on all the involved AIM elements (functional subsystems and blocks, pipeline components, storage elements, interfaces, data-exchange common buses, etc.) as duly applicable to the requirement text itself.

Automated and Intelligent Management (AIM), in an extreme schematic way, is about empowering management and control functionalities and processes with automation capabilities based on intelligent decision making.

The AIM framework encompasses various degrees of intelligence including AI/ML, to help Network Operators and Service Providers operating and managing their network and services by enabling and improving automation, optimization and promptness of processes.

The AIM framework leverages on the following Loop Automation (input-recommendation-action) paradigms:

- "Open Loop Automation" (OLA) – recommended actions are applied upon human supervision.
- "Closed Loop Automation" (CLA), – recommended actions are applied without a human supervision.

> **Note:** The adoption of OLA or CLA for a specific automation task/process is not necessarily static; instead, it allows adaptability and flexibility in the way the Loop Automation is implemented.

The required capabilities and characteristics to be provided by the AIM framework are grouped per macro-categories in the sections below.

4.1 Architectural Characteristics

This section, as well as Section 5, contains high level requirements referred generically to the “AIM framework”. These requirements are supported by the definitions in this Technical Report, and imply support by the appropriate components and functionalities as applicable.

In Section 6, most requirements are more tightly scoped around specific components and functionalities.

R-1 The AIM framework MUST support standard interfaces/points of reference.

R-2 The AIM framework MUST support standard data models (e.g., YANG) for:

- Telemetry data modelling
- Nested OLA/CLA result communication

R-3 The AIM framework MUST be applicable to hybrid physical and virtualized infrastructure.

R-4 The AIM framework SHOULD support a microservice and container-based environment.

R-5 The AIM framework MUST support the KPIs and telemetry data related (but not limited) to:

- Network Creation
- Service Delivery
- Service Assurance
- Customer Experience

R-6 The AIM framework MUST support network domains’ federation (e.g., Access, Transport, Core)

R-7 The AIM framework MUST support consuming telemetry data and KPIs fetched from other (not network related) external domains (e.g., video content consumption, location-based information history and forecasts).

R-8 The AIM framework MUST NOT negatively impact Lawful Intercept (LI).

Example: ongoing lawful intercept activities are not affected by service or network optimizations and adjustments performed by the framework

4.2 Data treatment

R-9 The AIM framework MUST support a Collection Function (CF) for various types of data (including telemetry, events, alarms, logs and notifications) with the following collection task customization features:

- Creation/activation/modification/deletion
  Examples of collection tasks:
  - troubleshooting - usually short term
  - recurring/continuous monitoring (e.g., SLA monitoring)
  - analytics (e.g., trend analysis) - long term

  Examples of collection task configurable characteristics:
  - Target Managed Entities
  - Collected primary parameters and filters (e.g., alarm severity), derived parameters (e.g., correlated alarms, threshold crossing events, status change events)
  - **Note:** Certain derived parameters may be calculated by the CF itself
  - Collection frequency (applies only to telemetry)
  - Duration
- Support and select transfer mode of collected data:
  - Streaming/Bulk
  - Pull/Push
  - Publish/Subscribe

R-10 The AIM framework MUST support a Pre-processing Function (PPF) with the following customizable features:

- formatting/normalization and pre-processing/filtering of collected data
- attaching metadata to collected data (e.g., to single records or to groups of them)
- aggregation of the collected data
  Examples: exchange of cross-domain data may not need to be highly detailed and/or fine grained hence data may be aggregated or passed with a lower sampling while in-domain data may be collected and consumed with a higher detail and sampling

R-11 The AIM framework MUST support a Distributor Function (DF) for various types of activities:

- feedback to lower Automated Loop
  Examples of outputs:
  - actions (for CLA and, upon human decision, for OLA)
  - recommendations (for human decision in OLA)
- feedforward to higher Automated Loop
  Outputs and their transfer mode depend on the requests from the CF of the DE governing the higher Automated Loop

4.3 Intelligence

R-12 The AIM framework MUST support an Intelligence functionality configurable and programmable for defining the following:

- Input parameters, desirably as part of a standardized interface

- Knowledge, decision logics and algorithms
- Outputs: recommendations, next best action, etc.
- Dispatching schemes for outputs: to human, to other system, etc.

Note: The configuration/programming interface shall be standardized. E.g., this could be implemented via APIs via which the Intelligence functionality can adapt inputs/outputs to various blocks of the hierarchical architecture.

R-13 The AIM framework MUST support self-healing functions and autonomic behaviors (e.g., service profile optimization, network configuration optimization, QoS optimization, traffic steering)

R-14 The AIM framework MUST support both OLA and CLA paradigms

R-15 The AIM framework MUST support hierarchical OLA/CLA

- Local/inner loops operate first, and if needed, expose an abstracted view to the outer loops
- Loops have different time-scale reaction

R-16 The AIM framework SHOULD support integration with 3rd party AI/ML platforms and tools

R-17 The AIM framework MUST support orchestration and chaining of Intelligence functionalities and closed loops

R-18 The AIM framework SHOULD support mixed deployment of centralized and distributed Intelligence functionalities

R-19 The AIM framework MUST support configurable Analytics services

Dimensions of analytics include: time-based (past, recent, real time, near future, future), locational, functional, quantitative (statistics), trends, polices, external influences, forecasting

R-20 The AIM framework SHOULD support AI model offline training, testing and evaluation to identify an “operation ready” and “deployable” model.

The same capabilities, with the exception of offline training, apply to any DE’s model.

R-21 The AIM framework MUST support governance capabilities on the models used by the Intelligence functionality, like:

- activation/deactivation of deployable AI Models
- monitoring of decision-making performances (e.g., output quality, suggested/applied actions) against expected objectives

R-22 The AIM framework MUST support capabilities to audit/supervise AI/ML decisions against security and privacy criteria

4.4 Data Store

R-23 The AIM framework SHOULD support Data Lakes, Data Warehouses and Data Hubs

R-24 The AIM framework MUST support data governance like access authorization and control

R-25 The AIM framework MUST support Security-by-design, Integrity-by-design and Privacy-by-design of data at rest, in transit and in use

5 AIM Framework Principles

This section describes the principles for a generic AIM framework, taking into consideration the required capabilities and characteristics described in Section 4.

Its sections shall not be construed as straight implementation guidelines. Instead, they describe architectural principles, logical building blocks and interrelationships and, ultimately methodological considerations that are all propaedeutic to the AIM architecture specification in Section 6.

Furthermore, in the reminder of this section some concepts and nomenclature are borrowed from frameworks defined by other bodies, e.g., the ETSI GANA framework [3][4][5].

5.1 Loop Automation

In its simplest form, Loop Automation (LA) operates on a logic involving one or more Managed Entities (MEs) and Decision Elements (DEs).

Definitions

A **Managed Entity** is a managed resource or a set of managed resources.
Fundamental MEs are atomic resources at the bottom of the management hierarchy, such as individual protocol or other types of managed mechanisms hosted in a network element (NE) or in the network in general.
Composite MEs, such as whole NEs themselves, are composed of multiple Fundamental MEs.
Typically, MEs do not implement any AI/ML capabilities but simpler degrees of intelligence.

A **Decision Element** is a component that implements decision-making capabilities via some degree of intelligence and reasoning over inputs mainly received from the associated Managed Entities (MEs).
The DE operates as part of an Automated Loop aimed at self-* features of a functionality or system (self-configuration, self-optimization, etc.) performed via dynamic and adaptive management and control of associated Managed Entities (MEs) and their configurable and controllable parameters. DEs can make use of information sources (data lakes, inventories) to enhance their decisions.

A DE generates, as main output, recommendations, that are either automatically converted into actions on the MEs (Closed Loop Automation) or supervised by human for decision (Open Loop Automation).

The simplest example of an automated loop with one ME and one DE is:

ME input $\Rightarrow$ DE recommendation $\Rightarrow$ action on ME

In real-world networks, automation is based on much more complex interactions and loops.

Figure 1 shows the possible nesting of control loops when the communication involves five different Managed Entities (ME) within a single domain and the concept can be easily extended to more MEs and to multiple domains.

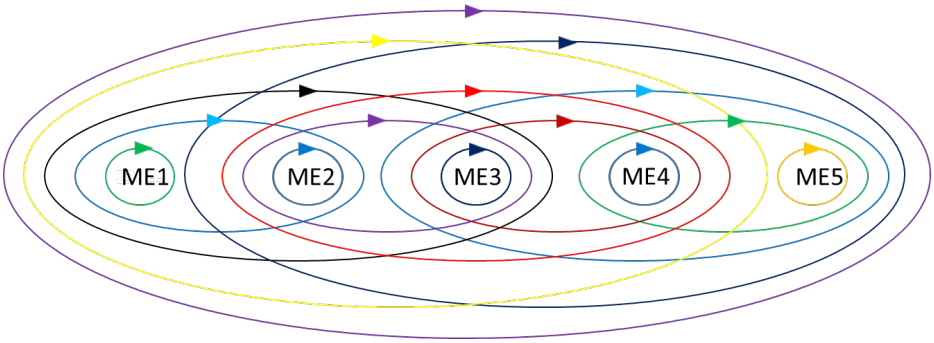

Figure 1: Breakdown of the Number of Control Loops

Efficient Loop Automation architectures are based on engineering of the Automation Loops, e.g., in terms of:

R-26 The AIM framework MUST support the efficient use of resources.
The Managed Entities that take part in an automated loop are logical entities and can be contained in one single Network Element (NE) or can span multiple NEs. Nested loops are meaningful only if extraneous duplication of logic or data does not occur.

R-27 The AIM framework MUST support hierarchical ALs.
The CLA/OLA network operations paradigms can train with known data, then operate by trial, error, and factoring that error back into the next try. Furthermore, it is essential, that the same process used for the first implementation is also used for the corrected/healed/scaled iteration. This requires a single closed-loop process that, through hierarchical nested loops, implements fulfilment, scaling, healing and optimization so that assurance and fulfilment converge into one continuous process.

R-28 The AIM framework MUST support domains federation.
The requirement of implementing federated/hierarchical control loops realizes the Keep It Simple Stupid (KISS) principle of beginning with the simplest, most deterministic action. There will be automated loops operating across multiple domains like Access, Transport, Metro, Core etc. and managing different Managed Entities (ME [1]) within each domain.
A federated approach makes it easy to span multiple domains or multiple technologies within a domain and allows problems to be corrected at the most local, easily understood, fastest level first. Moreover, domain expertise is extremely valuable in performing the initial root-cause analysis and correction.

Best practices for Loop Automation, as well as the ETSI GANA recommendations, suggest that the lower-level inner loops operate first, and if needed, expose abstracted views as inputs to a higher-level outer loop where service orchestration and end-to-end service model / assurance can take over.

This implies a loose coupling architecture and API exposure of lower-loop state, and an explicitly federated approach to enable E2E automated loops establishment to serve more sophisticated delivery and assurance processes. These best practices reduce complexity, reduce risk and minimize disruption as well scaling of domains and capabilities over time.

5.2 Hierarchical Automated Loops

The simple example of an Automated Loop in Section 5.1 alludes to its pipelined operation, in coherence with the logical functionalities described in Section 4.

ITU-T Y.3172 [17], as well as ETSI ENI [6], fully develop the concept of a Machine Learning based pipeline and its functional components.

Section Appendix I, reports the definitions of ML components provided by ITU-T Y.3172 [17] and ETSI ENI [6], which are functionally quite similar.

For unification purposes, this Technical Report, has adopted the ITU-T Y.3172 [17] nomenclature:

- Collection Function (CF): responsible for data collection
- Pre-processing Function (PPF): responsible for data processing
- Model Function (MF): responsible for knowledge handling
- Policy Function (PF): responsible for the application of policies to the MF outputs
- Distributor Function (DF): responsible for distributing the outputs of the MF

ITU-T Y.3172 [17] definitions have also inspired the functional specification of the AIM architecture in this Technical Report, but this Technical Report is not constrained by the ITU-T Y.3172 [17]and ETSI ENI [6] frameworks.

Instead, the AIM framework in this Technical Report, goes beyond them to best fit into Broadband Forum's portfolio of specification for network transformation towards Cloud, SDN, NFV based architectures.

Figure 2 reprises the simplest Automated Loop touched upon in Section 5.1 and offers a represents Loop Automation both as a pipeline of logical components and as a feedback circuit.

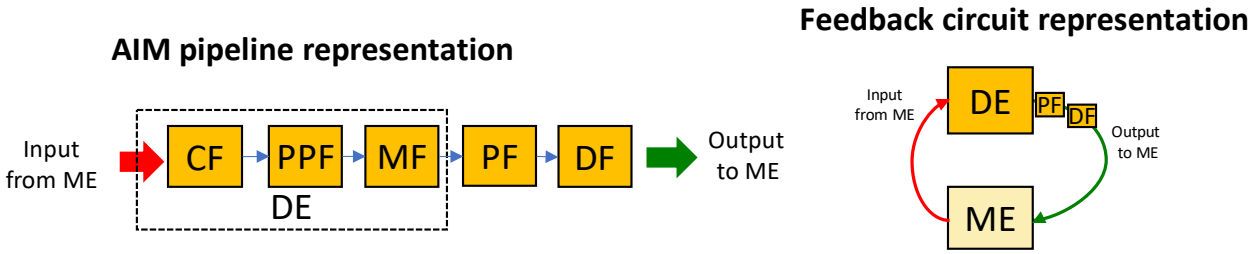

Figure 2: Loop Automation - pipeline and loop representations

This figure aims at highlighting the key principles of Loop Automation:

- The DE is the central function in an Automated Loop; it is the intelligent engine that makes it "turn"
- The DE is functionally modular and can be composed flexibly at the time of instantiation to best serve the objectives of the Automated Loop that the DE governs and fuels
- The DE always embeds (at least) one Model Function (MF) which is responsible of the decision-making process based on the interpretation of the environment inputs and of generating (not necessarily executing) feedback actions that "close the loop".
- The MF implements a certain degree of intelligence (see Section 5.4 for more details) which inherently is tailored to the degree of complexity of the environment to be interpreted and the degree of sophistication of the decision the MF is required to take.
- The DE operates in an AIM pipeline which takes advantage of external functions that allow to operate on the MF's outputs by flexibly tuning the policies applied to them (PF) and flexibly dispatching them (DF).

Examples of instantiation of simpler or more complex DEs and their chaining in an AIM pipeline are provided in Section 6.5.

Real world Automated Loop architectures can be more complex and dynamic than the sketch in Figure 2 and, for efficient field operation and other objectives, they should better be based on a hierarchy of Automated Loops per the modern practices of software-defined network design.

Figure 3 schematically depicts a hierarchy of Automated Loops by chaining DEs operating at different levels. With respect to Figure 2 the PF and DF blocks are omitted for simplicity of the diagrams and to highlight yet again that the DE is the essential component.

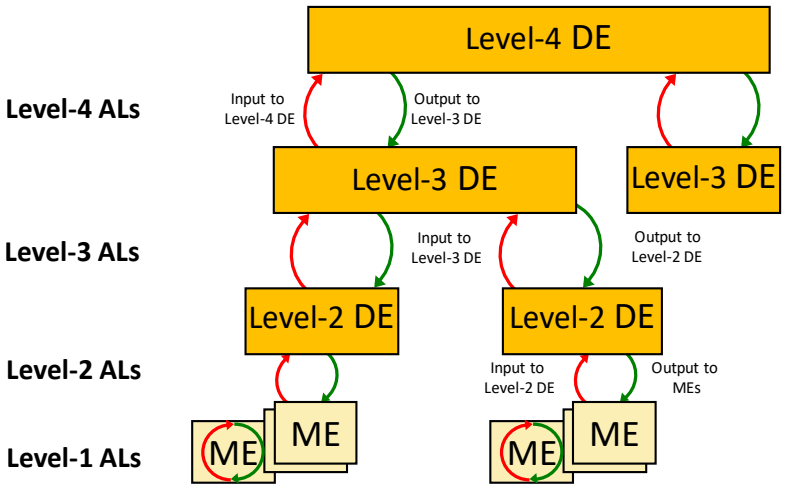

Figure 3: Automated Loops Hierarchy

R-29 The AIM framework MUST support hierarchical Automated Loops coherently with the principles of layered abstraction of Software Defined Networks. This translated into the ability to assign Level-n DEs outputs as potential inputs to a Level-(n+1) DE of higher Automated Loop.

The GANA framework [4] defines four Automated Loops levels assigned to network functional entities of increasing complexity. Details are reported in I.1 including the typical degree of intelligence that the GANA framework recommends for each level of DE.

R-30 The CLA/OLA hierarchy SHOULD be implemented via the vertical chaining of ALs based on the Level-n DEs exposure of an abstracted set of data (red arrows in Figure 3) and of control and management functionalities (green arrows in Figure 3) to the Level-(n+1) DE, i.e., no Automated Loop spans across multiple levels.
Note: This implementation choice, though not prescriptive, is recommended because it enables separation of concerns and avoids inter-level dependencies.

A vertically-chained hierarchy guarantees that a unique higher-level DE governs the assigned managed resources (i.e., lower-level DEs or, at the bottommost level, the MEs) for collection and action purposes. This principle is key in an environment where multiple clients need to consume collected data from and to issue commands to the same network assets.

Automated Loops spanning across the hierarchical levels are possible in the form of “logical ALs” via the chain of underlying ALs which expose a degree of resources abstractions which increases with the AL level.
This architectural choice is, once again, aligned with the recursive nature of software-defined network design and with modular design and dynamic lifecycle practices of NFV environments.
The advantages are abundantly described in literature:

- Loosen dependencies across DEs levels
- Limit technology upgrades or changes as well as modules reusability.
- Defer the actions on network resources to the SDN element in charge, to allow:
  - commands/data requests reconciliation and conflicts avoidance
  - load healing on the management interface of network resources
  - commoditized standard data handling and command actuation reference point for fully flexible and dynamic AL lifecycle

This Technical Report does not forbid implementing ALs that span directly (rather than logically) across the CLA/OLA hierarchy levels. However this would imply multiple DEs to directly access the same resources heavily hampering the fruition of the above advantages unless they share a common Collection Function.

5.3 Domain Federation

The concept of domain federation consists in the implementation of cross-domain interfaces between domain-specific DEs (e.g., Dom1-DE and Dom2-DE) whereby the decision process of Dom1-DE is then based not only on the inputs from Dom1-MEs but also on inputs received from Dom2-DE and vice-versa.

There are two way of implementing domain federation:

1) Via a cross-domain interface implementing operations and primitives that need to be exchanged between the federated domain-specific DEs.
2) Via an orchestration element hierarchically higher than Dom1-DE and Dom2-DE.

The latter approach is highly preferable because 1) requires added complexity to the federated domain-specific DEs. Furthermore, the introduction of one interoperability interface per each pair of federated domain-specific DEs.

On the other hand, 2) requires a minimal additional complexity on the higher tier orchestrator for the cross-dispatching of selected information to the federated domains at the advantage of keeping the domain-specific DEs totally decoupled and less complex.

R-31 The AIM framework MUST implement domain federation via a cross-domain orchestration element.

In the CloudCO architecture, which encompasses the Access and Edge domains and, as described in the excerpt from TR-384 [1], may also be implemented through multiple CloudCO domain instantiations, two cases for domain federation, but not limited to, are:

- Federation among the Customer Premises, Access, Edge and DC SDN M&C elements either via cross-domain interfaces or via the CloudCO Domain Orchestrator coordination
- Federation between multiple CloudCO Domain Orchestrators either via cross-domain interfaces or via the coordination of the E2E Service Orchestrator

Other possible examples of domain federation may encompass the SDN M&C elements and the VNFM.

ETSI GANA defines a federation architecture based on the concept of a Knowledge Plane (KP) constituted by the Network Level DEs, a distributed system of federated information servers for Information eXchange (ONIX) and a Model-Based-Translation Service (MBTS) for translating information and commands/responses.

5.4 AIM Degrees of Intelligence

With the advent of Artificial Intelligence (AI) and Machine Learning (ML) techniques, combined with easier access to considerable computational power, a strong focus and expectations are put over applications with intelligent capabilities and fast decision-making that simulate human-like reasoning.

This section elaborates a two degree, levels of intelligence in decision-making:

ML AI - Trained (supervised or unsupervised) rather than explicitly programmed rules and models. Rules and models are derived from the statistical structure of the training set. In case of Deep Learning (DL) models are derived from a multi-layer representation of the training set

Rule-based AI - Hardcoded rules crafted by programmers that do not involve any learning. Expert knowledge intelligence based on generally reusable solutions and actions in response to specific conditions or known and

identified problems. Include predefined logics (e.g., equational, semantic, syntactic) or constraints (physics, SAs, etc.). Sufficiently large set of explicit rules for manipulating knowledge.

R-32 The AIM framework MUST support DEs that implement ML AI or Rule-based AI, or both.

As discussed in sections 5.1and 5.2, at the core of AIM, the Decision Element (DE) applies “reasoning” to its environment, i.e., a protocol, a function, a whole node, a network of nodes and systems. Then the DE makes decisions that most effectively satisfy a set of defined goals and known constraints. The decision-making consists in selecting the best actions among several possible alternatives.
Decision-making is a knowledge-based process often improved by learning techniques for the dynamic construction of new knowledge.

The ability to learn from the environment and from the outcomes of the DE’s decisions is used to identify two degrees of intelligence. This ability is based on the collection of environment data and information but also built via experience.

Under this standpoint intelligence may be used to understand what is happening, but also to explain what has happened and to reliably predict what may happen in the environment.

The levels of intelligence, that one DE may implement, are listed below according to the levels of complexity in building their knowledge and applying it.

5.4.1 Machine Learning Artificial Intelligence

Machine Learning (ML) Artificial Intelligence (AI) techniques are very sophisticated decision-making algorithms supported by powerful learning capabilities and modeling techniques.

This degree of intelligences allow to improve the understanding and to continuously gain knowledge about the network and service environment, and hence achievement of the operator goals such as improved management and automation (thanks also to proactive decisions), improved network utilization and agility, being able to detect and respond to real-time changes, while maintaining QoS levels or SLAs.

Supervised and Unsupervised ML, described in literature, enable systems to gain knowledge from data without necessarily being explicitly programmed, influenced and driven by already available knowledge. Supervised learning aims to learn mapping function from given input (labeled training data set) to the output. Unsupervised Learning aims to learn a function that describes a hidden structure, characteristics from unlabeled training data set. Reinforcement Learning (RL) is a goal-oriented learning process, based on learning by interactions with the environment, possibly simulated for some initial steps. The RL agent aims to optimize an objective by interacting with the environment based on a direct trial-and error process in live network. This is the so-called “learning by doing” or “self-learning”.

Concrete examples of Supervised Learning techniques are Decision Trees, Artificial Neural Networks (ANN). ANNs have already been used with great success in image recognition.

Concrete examples of unsupervised learning techniques are Clustering and Principal Component Analysis (PCA).

The main output of ML AI process and techniques is a model for the environment, which is created and trained via an appropriate data set. The trained model for the environment is tested on different data sets and evaluated against expected model performances indicators, (reliability, convergence, false negatives and false positives thresholds, etc.) before deployment in the live network.

5.4.2 Rule-Based Artificial Intelligence

Rule-based AI is another type of intelligence, in certain forms already adopted in existing networks. Expert knowledge intelligence based on generally reusable solutions and actions in response to specific conditions or

known and identified problems. Some tasks require the work of experts who apply various rules and their empiric knowledge.

The Expert Systems and Rules Engines have been one of the first attempts to emulate the human intelligence and decision-making process, by inferring from existing knowledge and dealing, better than humans, with vast amounts of complex rules.

Modern Expert Systems can more easily incorporate new knowledge and thus update themselves. The use of these knowledge-based systems is of huge interest and application for network management and service provision, for the automation of complex decision-making processes (e.g., network configuration systems and tools), for the application of complex interpretation rules (for example inferring environmental conditions and activities from sensor data), for prediction (inferring likely consequences of a given situation) or for diagnosis (inferring causes of malfunctions and correlated repair actions).

Rule-based AI is mainly based on known and successfully experimented observe-and-react schemes. However, the learning dimension is essential: those schemes need to be continuously verified and possibly improved via the discovery, experimenting and performance evaluation of new behavioral rules.

Expressing the observe-and-react schemes and behavioral rules as coded applications enables automation, reactivity, adaptability and reusability of these knowledge-based intelligences. The predefined logics (e.g., equational, semantic, syntactic) or constraints (physics, SLAs, etc.) rather than a true decision-making process may also be considered part of Rule-based AI.

Examples of this type of intelligence are traffic engineering equations, self-healing protection configurations, real-time adaptive bitrate media streaming.

In this case the observe-and-react logic is coded via equations. These equations and, more frequently, systems of equations have a quite static structure albeit shaped by coefficients, boundaries and constraints that guide their reaction thresholds, outcomes, convergence, etc.

This type of intelligence requires low implementation complexity, while its hardwired nature brings along extreme reactivity.

These two characteristics together make it very suitable for Automation Loops requiring quasi real-time responses and operating on network assets with limited processing and storage resources.

Rule-based AI generally has a low level of learning abilities involved but still the equations knobs as well as the input variables can be adjusted based on experience or retuning of thresholds or goals.

The ability to implement efficiently the equation intelligence via software is key for automation, reactivity, adaptability and reusability.

5.5 Information Model

In order to create synergies and to make data available to be consumed within the same organization but also to be shared with business partners, the creation of a data catalogue and common vocabulary for all the data hosted in various data stores becomes paramount.

The TM Forum has developed the Information Framework (SID) GB922 [18]. The SID is a representation of business concepts, their characteristics and relationships, described in an implementation independent manner. The Information Model:

- Acts as a language for different stakeholders to communicate effectively and precisely
- Provides a single common definition of important entities across the Organization
- Is the standard used for developing integration services (interfaces) between applications (IT systems)
- Helps understand the information that is spread across the applications

R-33 The AIM framework SHOULD support an information model based on shared data catalogues and a common vocabulary.

The use of a common and shared Information Model introduces a reference standard to support heterogeneous IT landscapes and business definitions, in a logical way, that is based on inherent properties of the information itself rather than on any existing physical data storage structures.

R-34 The AIM framework MUST support TM Forum SID GB922 [18] as the foundation for its own information model for the unambiguous characterization of the data referenced in the data catalogue.

R-35 The AIM framework MAY implement its own catalogue of metadata to help discover, describe, assemble and govern data sets, or integrate with a 3rd party catalogues and data governance systems via standard APIs.
Considering the data catalogue will span multiple domains across the organization the latter should be the preferred choice.

For example, the metadata schema may be based on the Dublin Core mode, which maintains an authoritative specification of all metadata terms. The Dublin Core terms are intended to be used in combination with metadata terms from other compatible vocabularies. As such, the AIM framework should be flexible and programmable with respect to what vocabularies to implement.

There are several different types of network data like (but not limited to) flows, counters, logs, and various KPIs with no obvious or consistent way to combine and correlate them. There is no standard way to integrate network data with other sources such as compute resources of a Virtual Machine (VM) or call center data, etc.

6 AIM Architecture

Network operators are seeking effective ways to incorporate AIM functionalities into networks incorporating SDN and virtualization. While many use cases (e.g., troubleshooting, traffic prediction, traffic optimization) can benefit from such integration, there are some important challenges:

- Cost-effective integration between AIM functionalities and SDN network;
- Benefits but also risks of potential discontinuities on operation and management practices that AIM functionalities, together with management mechanisms for network functions, may generate.

An architectural framework for the integration of AIM functionalities is provided to address these challenges, providing a standard specification and efficient methodologies for integrating AIM functionalities into SDN networks.

Building on the high-level characteristics and capabilities listed in Section 4, and the principles described in Section 5; this section provides a set of common and reusable building blocks, the AIM pipeline components and their relationships with SDN architecture elements, identifying the reference points, which enable loosely coupled integration.

Following operators needs to exploit the benefits of using intelligence above and beyond pure network operations, to fulfill also service business logics, the AIM framework specified in this section spans up to the E2E Service pipeline orchestration and makes use of an NFVI and in particular the use of its general-purpose network switches, compute nodes and storage, as well as Management and Orchestration functionality (MANO). The AIM framework is also widely applicable regardless of the specific IaaS/PaaS virtualization environment in place in a given Operator deployment.

The NFVI hosts VNFs, which implement in software most of the functions orchestrated within the AIM pipeline while exploiting the standard data handling and command actuation reference point offered by the SDN infrastructure to interact with the live network.

AIM Data Input/Output (I/O) operations are implemented by VNFs that are chained and orchestrated in tandem with the NFVI to deliver a control loop function.

The AIM architecture has the following benefits:

1. It enables fine-grained design, control and implementation of control loops, in part due to the decoupling of the key functions of the pipeline.
2. It supports the flexibility and scalability offered by the virtualization of the AIM pipeline functions.
3. It supports automated, rapid deployment and on-boarding of AIM pipelines.
4. It provides for actions on the live network resources to the SDN element in charge, to allow:
    - Commands/data requests reconciliation and conflicts avoidance
    - Load healing on the management interface of network resources
    - Commoditize standard data handling and command actuation reference point for fully flexible and dynamic AL lifecycle

This section also describes how the logical functional blocks of the AIM pipeline are implemented and how they interact via standard interfaces and data models. The AIM framework allows for dynamic placement and chaining of the various pipeline functional blocks as well as allowing external consumption and data sharing via domain federation.

It describes how the AIM interacts with the Customer Management layer (Analytics, CRM & Order management, Assurance, billing etc.) in order to build end-to-end control loops for existing or newly created services.

It describes the AIM SBI and AIM NBI, which allow the deployment of 3rd party pipeline functionality (implemented as VNF) and/or the consumption of AIM functionality by 3rd party without having to expose the internals of the AIM architecture. The 3rd party consumable API can also be used to on-board and create new pipelines besides to federate multiple domains.

Some requirements in this section are followed by explanatory text, after a carriage return with no spacing. This explanatory text further defines the requirement.

6.1 Architecture Basic Components

Recalling the concepts briefly introduced in section 5.2, Table 1 describes the basic components, i.e., software modules, that can be part of an AIM pipeline and included in an AIM application.

R-36 The AIM framework MUST support the deployment of Automated Loops based on pipelines flexibly composed of the AIM components specified in Table 1.

Table 1: AIM Basic Components, AIM pipeline, AIM application

Component Name	Component Description
SouRCe (SRC)	It is the source of data that can be used as input to the AIM pipeline.
Collection Function (CF)	It is responsible for collecting data from one or more SRC logical nodes. A Collector Function may have the capability to configure SRC logical nodes. For example, may be used to control the nature of data, its granularity and periodicity while it is generated from the SRC.
Pre-processing Function (PPF)	It is responsible for cleaning data, aggregating data or performing any other preprocessing needed for the data to be in a suitable form so that the AIM Model Function can consume it.
Model Function (MF)	It is a model, in a form which is usable in an AIM Live Pipeline.
Policy Function (PF)	It enables the application of policies to the output of the MF node. This node can be used to monitor model performances. Moreover, this node may govern the impact of the output to a live operational environment or to other systems thanks to specific rules that can be put in place by network operator.
Distributor Function (DF)	It is responsible for identifying the SINK(s) and distributing the output of the MF node to the corresponding SINK nodes. It may have the capability to configure SINK nodes.
SINK	This is the target of the output on which the AIM application takes action.
AIM pipeline	Set of the basic AIM components integrated by the chaining mechanism that enables the flow of information and control.
AIM application	A complete software application may include a set of pipelines with all kind of AIM basic components (SRC, CF, PPF, MF, PF, DF,SINK) or just few of them (e.g., SRC, MF, DF).

6.2 Architecture Logical subsystems

The AIM framework is composed of the following four logical subsystems whose support is defined by the requirements below:

R-37 An AIM Live Pipeline logical subsystem MUST be supported (highlighted in yellow in Figure 4)

R-38 An AIM Sandbox SHOULD be supported

R-39 An AIM Management logical subsystem MUST be supported (highlighted in yellow in Figure 4)

R-40 A Knowledge Base logical subsystem MUST be supported

These logical subsystems are illustrated in Figure 4.

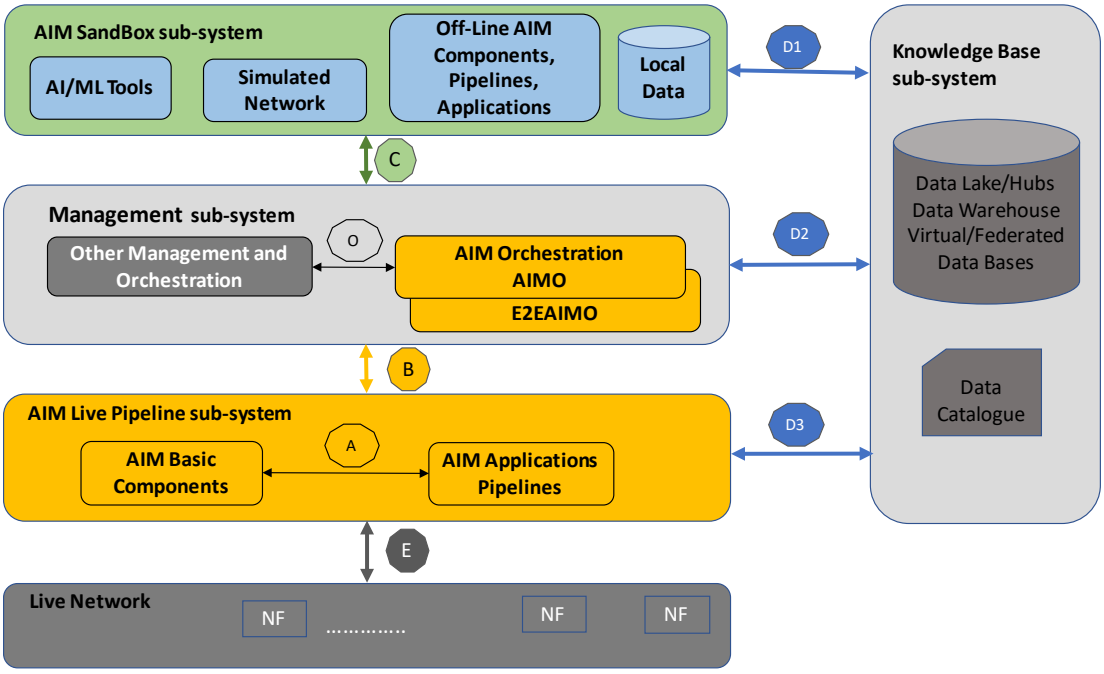

Figure 4: AIM Architecture Logical Subsystems

6.2.1 AIM Live Pipeline subsystem

The AIM Live Pipeline logical subsystem is deployed and running on Operators environment placed in the operational infrastructures, thanks to the AIMO and/(or) E2EAIMO that oversee(s) the instantiation of the AIM components, set-up of the entire AIM pipelines and run-time monitoring of the AIM applications.

R-41 The AIM Live Pipeline logical subsystem MUST support the ability to deploy, instantiate and run software modules, i.e., AIM basic components, each with specific functionalities, combined to form AIM applications.

6.2.2 AIM Sandbox subsystem

Requirements in this section apply only if the optional AIM Sandbox is supported.

R-42 The AIM Sandbox MUST support a dedicated off-network environment that allows the hosting of AIM components and AIM pipelines to train, test and evaluate them before deploying them as part of the running AIM Live Pipeline logical subsystem in the operational environment.

Being that the Model Function (MF) is the core component of AIM pipelines, there are some specific model lifecycle phases that should be implemented and managed in an AIM Sandbox.

R-43 An AIM Sandbox logical subsystem, for ML AI type of AIM applications, MUST support Model Selection, Design and Training. For Rule-based AI type of AIM applications the Model Selection, Design and Training SHOULD be supported.

Model selection and design is the first step and requires access to various data sources and data exploration mechanisms to help the model designer. Before deploying the model in the operational environment, the model needs to be trained, tested and evaluated off-network, using available data and AI/ML tools, network simulators or other simulators of the environment to be observed by the model, particularly useful for Reinforcement Learning models to realistically assess the model behavior, prior to deployment.

R-44 An AIM Sandbox logical subsystem SHOULD support Model Update
Based on reported model performances, the operator decides to retrain the model, or in some cases, to select a new model. Once the model has been updated it can be reinjected into operational environment.

R-45 An AIM Sandbox logical subsystem SHOULD support selection of the best model, training, testing and evaluation of the effects of model outputs before deploying a model on live operational environment.

Model training, testing and evaluation, are done in AIM Sandbox logical subsystem independently of the live network, i.e., off-line. For training and/or testing purposes, the AIM Sandbox can use data generated from a network simulator and/or from the live network. Available historical data previously generated from real network may also be used for training and/or testing purposes.

The AIM Sandbox logical subsystem allows AIM pipelines to adapt to dynamic (network, service) situations where a variety of conditions may change (e.g., network capabilities and resources).

R-46 An AIM Sandbox logical subsystem SHOULD include AI/ML tools to train, test and evaluate ML models, Off-line AIM components/pipelines/applications, simulated networks, data from live networks, access to historical data from real network, and eventually local data.

R-47 An AIM Sandbox SHOULD be controlled and managed by the AIMO and/or E2EAIMO according to the specifications in the AIM Intent.

AIMO and/(or) E2EAIMO provides feedbacks to AIM Sandbox regarding AIM Live Pipelines, so that the AIM Sandbox may provide continuous adaptation to the dynamically changing network/service conditions and environments. This allows the operator to re-select, re-train, re-test and re-evaluate the model(s) for a given AIM application.

6.2.3 AIM Management logical subsystem

The AIM Orchestrator (AIMO) orchestrates AIM logical subsystems and the End-to-End AIM Orchestrator (E2EAIMO) provides orchestration across AIM domains.

R-48 The AIM Management logical subsystem MUST support the AIM Orchestrator (AIMO) and the End-to-End AIM Orchestrator (E2EAIMO), i.e., the AIM specific management and orchestration functional blocks. These AIM Orchestration blocks collaborate with other management and orchestration functions (Domain Orchestrator, E2EOrchestrator) of the SDN-NFV infrastructure thus enabling the extension of the management and orchestration capabilities and mechanisms used already for SDN networks to AIM components, pipelines and applications.
logical subsystem The AIMO and E2EAIMO work in coordination with the other SDN-NFV orchestration functions to manage the AIM Live Pipeline Logical subsystem.

The interaction between AIM framework and SDN-NFV Orchestration is achieved by a Service-Based Architecture (SBA).

Being that the Model Function (MF) is one of the core components of the AIM framework there are some specific model lifecycle management functions that have to be implemented in the AIM management subsystem.

R-49 The AIMO and E2EAIMO MUST support Model Deployment and Operation - Trained, tested and evaluated models can be deployed in the operational environment. Deployed model generates outputs (actions/recommendations), based on accumulated knowledge and on the live data collected from the operational environment.

R-50 The AIMO and E2EAIMO MUST support Model Performance Monitoring - Deployed model can generate feedbacks or reports, as secondary but very precious outputs, to monitor its own performances enabling an update and/or re-train of the model. Model performances may be checked against performance thresholds and may lead to warning and notifications recommending, for example, to update or to retrain the model.

R-51 The AIMO and E2EAIMO MUST support management and orchestration functions of the AIM components of AIM pipelines and AIM applications deployed in the operational environment based on AIM Intent and dynamic conditions.

R-52 The AIMO and E2EAIMO MUST support AIM Model Selection at the set-up time of the AIM application - advances in artificial intelligence and machine learning provide models with various characteristics that are appropriate for different kind of problems and having different characteristics (as described in5.4 AIM Degrees of Intelligence)

R-53 The AIMO and E2EAIMO MUST support AIM Model Training and Model Updates without impacting the live operational environment.
ML model training:

- MAY use dedicated hardware to optimize certain AIM KPIs (e.g., decision algorithm convergence speed, decision reliability, etc.);
- MAY require availability of huge amounts of data; and
- MAY require lot of parameter optimizations, etc.

There could be various training techniques producing complex models (as described in section5.4. The performances of such models are determined and evaluated in a dedicated environment, i.e., AIM Sandbox, prior to be selected for specific AIM applications. Model training, testing and updates are performed in a way to avoid impact on live operational environment.

R-54 The AIMO and E2EAIMO MUST support AIM model transfer from AIM Sandbox to AIM Live Pipeline subsystem, i.e., in real operational environment, according to the specific AIM application lifecycle management.

R-55 The AIMO and E2EAIMO MUST support flexible, multilevel and multi-domain chaining of AIM components.
Chaining functionality, i.e., connecting AIM components together to form an AIM pipeline is about hosting and positioning AIM components on different NFs and levels/domains, enabling the hierarchical and/or distributed AIM applications. Chaining of AIM components is used to build complex AIM applications.

R-56 The AIMO and E2EAIMO MUST support instantiation, set-up and orchestration of the AIM live pipelines in coordination with the other SDN-NFV orchestration functions, based on the infrastructure capabilities and requirements of the AIM application.

R-57 The AIMO and E2EAIMO MUST support monitoring of the effects of AIM applications on network/service operations – monitoring includes evaluation of network/service performance along with performance of models and algorithms.
Various network/service KPIs are measured constantly and the impact of AIM applications on them. AIM components themselves have to be monitored continuously and corrected or updated if needed (e.g., when the model performance falls below a predefined threshold) to be ready for future recognition and handling of such scenarios.

R-58 The AIMO and E2EAIMO MUST support receiving feedback from the AIM Live Pipeline logical subsystem and from the SDN-NFV orchestration functions, along with feedback regarding the output of the MF based on model performances, to be aware of the performances of deployed models in a real operational environment.

R-59 The AIMO and E2EAIMO SHOULD support orchestration of the AIM pipeline components in AIMO Sandbox:
For some AIM applications it MAY be required to test and evaluate the complete pipeline made of several AIM components, and not just to train, test and evaluate some of them such as ML-based Model Functions. In that case AIMO/E2EAIMO may orchestrate the complete AIM pipeline in AIM Sandbox environment.

6.2.4 Knowledge Base subsystem

A Knowledge Base logical subsystem is used for storing knowledge and data in various types of repositories (Data Lakes/Data Hubs, Virtual/Federated Data Bases, Data Warehouses) accumulated also over time.

R-60 Data catalogue and other management systems MAY be supported to facilitate data integration empowered by data governance.

R-61 The Knowledge Base logical subsystem MAY be supported even as part of global data management strategy related also to other processes than network/service management as defined by each Operator.

6.2.5 Logical Reference Points between AIM Architecture Logical Subsystems

R-62 The AIM framework MUST support the following logical Reference Points defined between AIM Architecture logical subsystems (see Figure 4 and Section 6.8; this latter reports a detailed description):

- **A** - internal to the AIM Live Pipeline subsystem, i.e., between AIM live pipelines components
- **B** - between AIM orchestration logical subsystem (particularly AIMO/E2EAIMO) and AIM Live Pipelines logical subsystem regarding instantiation of AIM components, set-up AIM pipelines, AIM application execution monitoring, model performance monitoring etc.
- **C** - between Management logical subsystem (particularly AIMO/E2EAIMO) and AIM Sandbox. It is used to transfer trained, tested and evaluated models ready to be deployed; it is also used to transfer feedback on performances of the AIM functionalities in live network, e.g., when the model performance falls below a predefined threshold.
- **D1** - between AIM Sandbox logical subsystem and Knowledge Base logical subsystem to get the data for the training, testing and evaluation of models in the AIM Sandbox environment
- **D2** - between Management subsystem, particularly AIMO/E2EAIMO, and Knowledge Base subsystem to access any data necessary for the management and orchestration purposes
- **D3** – between AIM Live Pipeline logical subsystem and Knowledge Base logical subsystem to access any data necessary for the run-time functionalities of the AIM live pipelines applications
- **E** – between AIM Live Pipeline logical subsystem and Live Network NFs to collect row data and produce effects on NFs thorough configuration actions/operations
  This shall be mediated by the SDN element that governs the target resource
- **O** – internal to the Management subsystem, to integrate and interact with other management and orchestration functions (Domain Orchestrator, E2E Orchestrator), for the management and orchestration purposes

6.2.6 Additional Architecture Aspect of Logical Subsystems

In addition to the above high-level architectural components and logical subsystems, the following architectural aspects are to be noted:

- **Declarative specification** (i.e., intent-based) and corresponding translation into configurations for AIM applications. Interpretation of the declarative specification, done by AIMO and/or E2EAIMO, allows translating the specification into the configuration and deployment in the run-time operational environment.
- **Service-based architecture (SBA)** is the approach for interfacing between AIM live pipelines applications and network (NFs), between AIM Live pipeline components themselves and may be used by the AIMO and E2EAIMO to orchestrate and manage the AIM functionalities.
- **Data handling reference points** are defined between AIM Architecture logical subsystems, as well as with the live network (NFs). Any impact to the live network is localized to the source of data and target of configurations, so the extensions of the existing protocols may be used to minimize the impact to the networks.
- **Domain federation -** the deployment of an AIM application may span different domains, e.g., distribute the AIM pipeline components across different domains (e.g., CPE, AN and Edge). In this case, integration and interfacing between components, pipelines located in different domains are based on domain federation principles (see Section 5.3) and detailed in Section 6.10.
- **Third-party integration -** enables the third-party solution providers to integrate with AIM Architecture providing AIM application, the complete AIM pipeline or some of the components of AIM pipelines, such as Model Function (MF) component.

6.3 High Level Architecture

The AIM framework specializes in management, control, and orchestration of control loop functionality in a pipeline. Figure 5 illustrates how the AIM framework fits into an Operator's larger overall network.

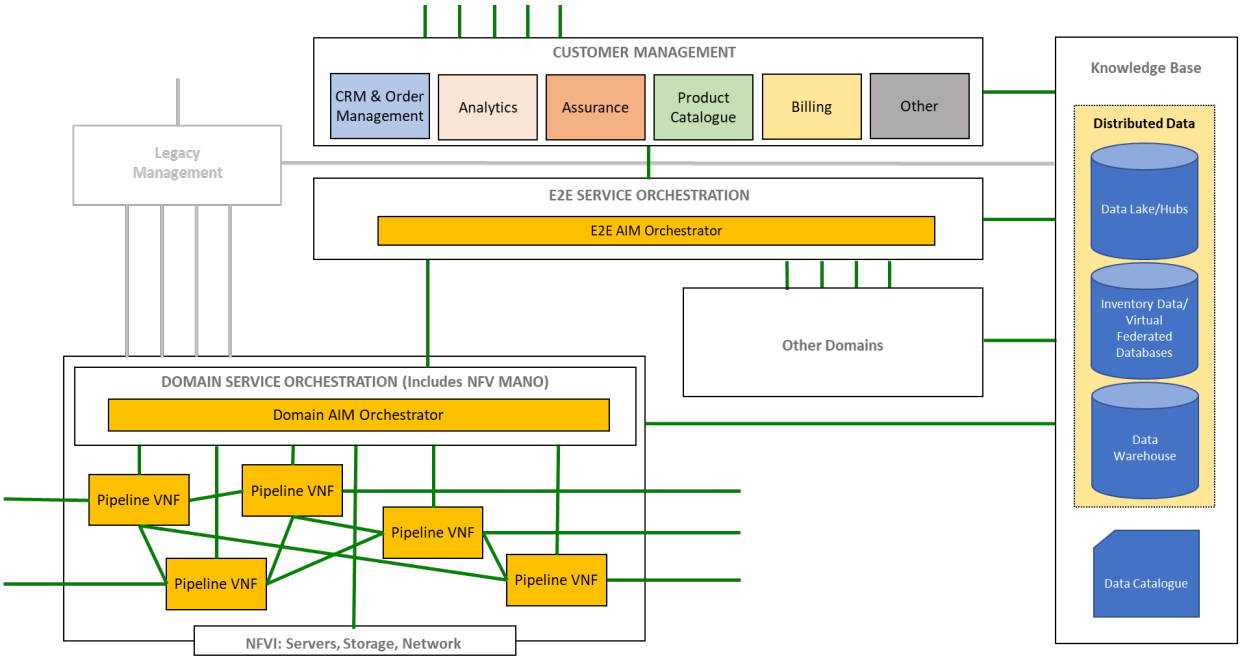

Figure 5: AIM High Level Architecture

The AIM Domain is accessed through a higher level entity, the End-to-End AIM Orchestrator (E2E AIMO), which has broader scope and visibility, and is in a position to resolve multiple accesses to the AIM resources including the sharing of information by granting applications access to the Knowledge Base. This broader scope spans from the given AIM Domain to other possible AIM Domains within the same provider's network, to non-AIM domains, and to peer domains with partner carriers or providers.

R-63 The End-to-End AIM Orchestrator function MUST coordinate all access interfaces, their privileges and views, and resolve any contention that may arise.

R-64 The End-to-End AIM Orchestrator functions MAY be implemented as a stand-alone End-to-End orchestrator or as a function of the End-to-End Service Orchestrator.

R-65 Each AIM Domain MUST be managed, controlled and orchestrated by the Domain AIM Orchestrator (Domain AIMO)

R-66 The Domain AIM Orchestrator (Domain AIMO) functions MAY be implemented as:

- a stand-alone orchestrator or
- as a function of a separate Domain Service Orchestrator governing the overall SDN-NFV infrastructure.

R-67 The AIM Domain NBI MUST allow on-boarding of control loops.

Control Loops are implemented using VNFs, which are logically interconnected by a given pipeline, and use declarative Intents to hide the complexity of the API. Furthermore, the widespread distribution of NFVI throughout network sites allows deployment of AIM functionalities flexibly in space and dynamically in time.

R-68 The AIM Domain SBIs MUST be used to instantiate, set up and manage the AIM pipelines in coordination with the other management and orchestration functions implemented by the NFVO.

R-69 The AIM Domain SBIs MUST map the pipeline Intents to diverse technology-specific network functions.

A single instance of the Domain AIM Orchestrator could orchestrate, manage and control the entire AIM pipelines or there might be constraints/needs that warrant deploying multiple, unique AIM Domains, where the respective Domain AIM Orchestrators are orchestrated and federated using the E2E AIM Orchestrator.

R-70 A Knowledge Base (KB) MUST be used for storing data and knowledge (e.g., insights, inferences, etc.) from the supported AIM domains and its evolution over time.

The Knowledge Base (KB) is implemented according to each Operator's database integration strategy.

R-71 The Knowledge Base (KB) SHOULD support strong data governance mechanisms and an Identity Access Management (IAM) system. It encompasses distributed data sources and other resources like data catalogue.

R-72 The distributed data resources MAY be implemented leveraging the following technologies or a combination of both:

- Data Lakes: Data from disparate silos is moved into one system (for instance Hadoop/HDFS). The data from disparate siloes is not harmonized or re-indexed. A Data Lake holds a vast amount of raw data in its native format, including structured, semi-structured and unstructured data.
- Data Hubs: Data is physically moved and re-Indexed into a new system. To be a Data Hub (vs. a Data Lake), this system would support discovery, indexing and analytics.
- Virtual/Federated Databases: Data stays in separate siloes. A Virtual Database accepts queries and pretends to be a big database that includes many disparate siloes data sets. It queries the back-end (live, production or warehouse) systems in real time and converts the data to a common format as it is queried. Inventory Data falls in this category.
- Data Warehouses: They are used for reporting and data analysis, and is considered a core component of business intelligence. Data warehouses are central repositories of integrated data from one or more disparate sources. As opposed to Data Lakes, Data Warehouses holds structured and processed data sets.

R-73 The data catalogue SHOULD expose an unambiguous characterization of the data held in the Knowledge Base.

The AIM framework could implement its own data catalogue to help to discover, describe, assemble and govern data sets, or it could integrate with a 3rd party catalogue and data governance system via standard APIs.

6.4 Domain Implementation

While Section 6.3 and Figure 5 illustrate the AIM framework in its wider context, this section and Figure 6 provide a first level view of the detail inside AIM Domain and how the AIM Domain is physically implemented in an AIM node.

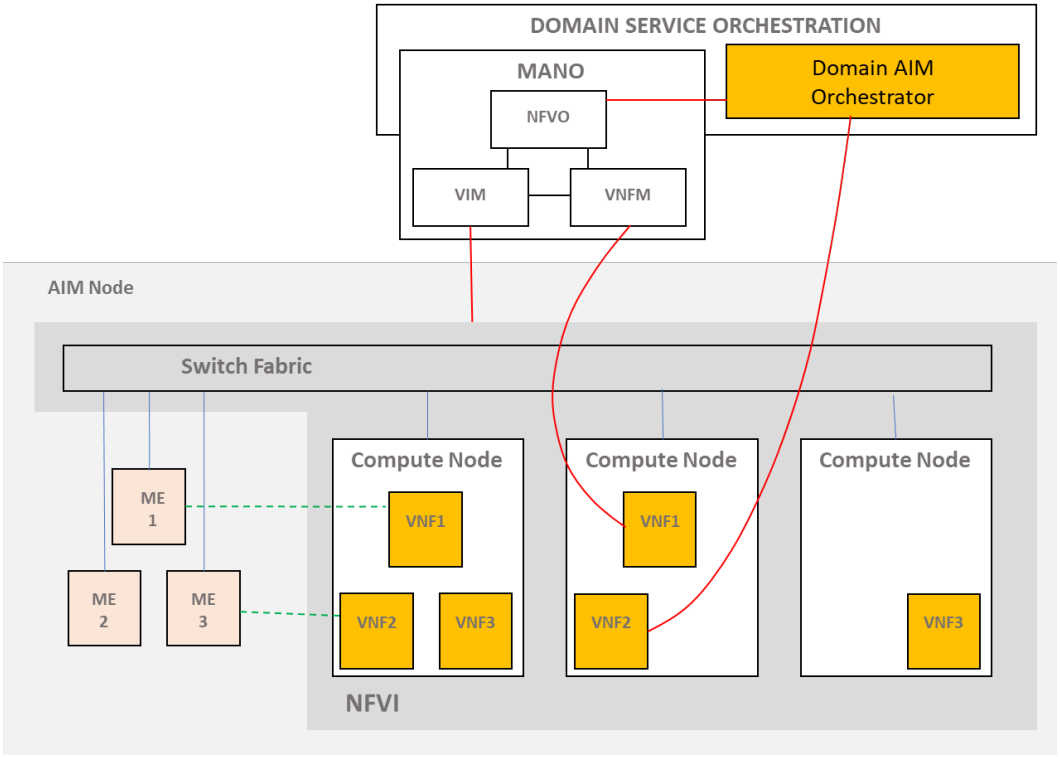

Figure 6: AIM Node structure

Functional blocks in the AIM include the Domain AIM Orchestrator with an NFV Orchestrator (NFVO), an NFVI, a Managed Entity and the AIM functional blocks, deployed as VNFs, that are chained and orchestrated in a pipeline/control loop.

Note that for the purposes of this architecture, the Domain AIM Orchestrator is a conceptual container for management, control and orchestration entities some of which may be implemented as VNFs.

The dark red lines in Figure 6 exemplify management and control associations to the various components. For example, the VIM is responsible for the management and control of the NFVI. Likewise, the VNF Manager (VNFM) provides management and control responsibility for the lifecycle and virtualization attributes of the VNFs, and the AIM Pipeline Domain Orchestrator is responsible for all AIM functional blocks inside the NFVI.

For a complete list of NFVO, VIM and VNFM capabilities refer to Section 5.4 of [7].

6.5 Functional Blocks

R-74 The AIM Functional Blocks (FB) SHOULD be implemented as micro-services and deployed as VNFs.

R-75 The AIM Functional Blocks (FB) MUST communicate using a message bus and standard APIs and data models.

The AIM framework inherits the guidelines defined in ITU-T Y.3172 [17] and in particular the requirement to break down the pipeline in the seven different logical components described in Section 6.1 and the requirement to orchestrate the logical components in a pipeline.

R-76 The AIM Functional Blocks MUST be implemented as part of a Service-Based Architecture (SBA) which supports the following characteristics/features:

- An environment where AIM applications can be deployed using components of varying sources and suppliers;
- Control Plane functionality and common repositories of components/Network Functions: VNF Catalogue(s) and Repository(ies),
- AIM applications are delivered by way of a set of interconnected Network Functions (NFs), with the flexibility and granularity of authorizing each NF to access to other NFs' services
- Network Functions are self-contained, independent and reusable and they can assume the role of either Service Consumer or Service Producer
- Network Function service exposes its functionality through a Service Based Interface (SBI), which employs a request-response and/or subscribe-notify interface model.
- Network Functions are interconnected via common buses
- Orchestration and VNF Catalogue(s) and Repository(ies) maintain a record of available NF instances with their supported services, allowing discovery and registration of new functions

R-77 The pipelines SHOULD be orchestrated to include any combination and any number of the AIM components.

Figure 7 presents five examples of pipelines. The topmost pipeline in Figure 7 is the most generic pipeline and includes all components defined by ITU-T Y.3172 [17]. The bottommost pipeline in Figure 7 is the simplest pipeline that may be built and includes a source (SRC), a collection (CF), a distributor (DF) and a SINK.

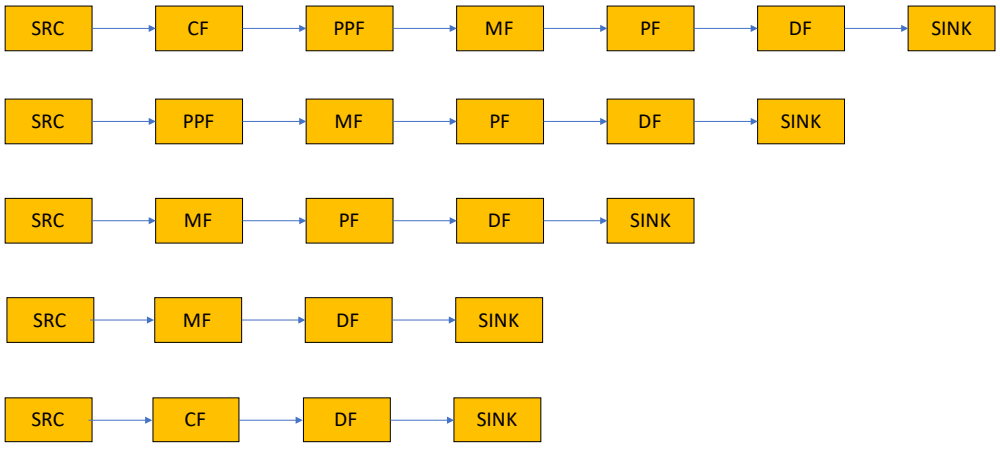

Figure 7: AIM Pipelines examples

R-78 The pipelines presented in Figure 7 MAY be chained and orchestrated hierarchically in order to implement the hierarchical control loop approach described in the ETSI GANA framework [4]. This is described in Section 6.7

The AIM framework goes beyond the descriptions of the logical components of a pipeline provided in ITU-T Y.3172 [17] and specifies how the logical components are implemented by the AIM functional blocks or components. Nonetheless, apart from the hierarchical control loop approach, the AIM inherits some additional

key concepts and definitions from the ETSI GANA framework [4]: Managed Entities (ME), Decision Element (DE) and Knowledge Base (KB) etc. described in Section I.1.

When a similarity or a match exists between the ITU-T Y.3172 and the ETSI GANA components and/or concepts, the AIM framework described in this Technical Report has adopted the terminology of either depending on the context. As a general rule, the most specific term is preferred.

R-79 The PPF and MF components MUST implement the different levels of intelligence described in Section 5.4.
Therefore, they are part of the logical ETSI GANA DE as explained in Section 6.6.

R-80 The Policy Function (PF) component MUST be implemented by the Domain AIMO in charge of enforcing policies on the output of the Model Function (MF) to the SINK.

R-81 The Distributor Function (DF) logical component MUST be implemented as a function of the Domain AIM Orchestrator.

R-82 The AIM functional blocks or components MUST be agnostic with respect to what control loop they are included in, to allow re-usability of the same block within multiple control loops.

R-83 The Domain AIM Orchestrator MUST configure the AIM functional blocks with the information, among others, on where to fetch the needed data from, and where to send their output to.

R-84 AIM DEs MAY implement a Local Data Storage functional block.
This concept is inherited from the ETSI GANA internal Knowledge Base (KB). The Local Data Storage is responsible for storing the data received by the data collector (CF) for a limited period and for applying a sliding window mechanism in order keep the most updated data only.

R-85 The Local/Temporary Data Storage of R-84 SHOULD provide the AIM Functional Blocks with the historical data they need, mainly to implement fast control loops.

Figure 8 provides a first level view of the detail inside the AIM Domain and how the AIM Domain is logically implemented. It also presents the AIM closest implementation of the ETSI GANA Decision Element (DE).

R-86 The AIM DE MUST include the Model Function (MF), the Pre-processing Function (PPF) and the data Collector Function (CF).

The AIM DE is further described in Section 6.6.

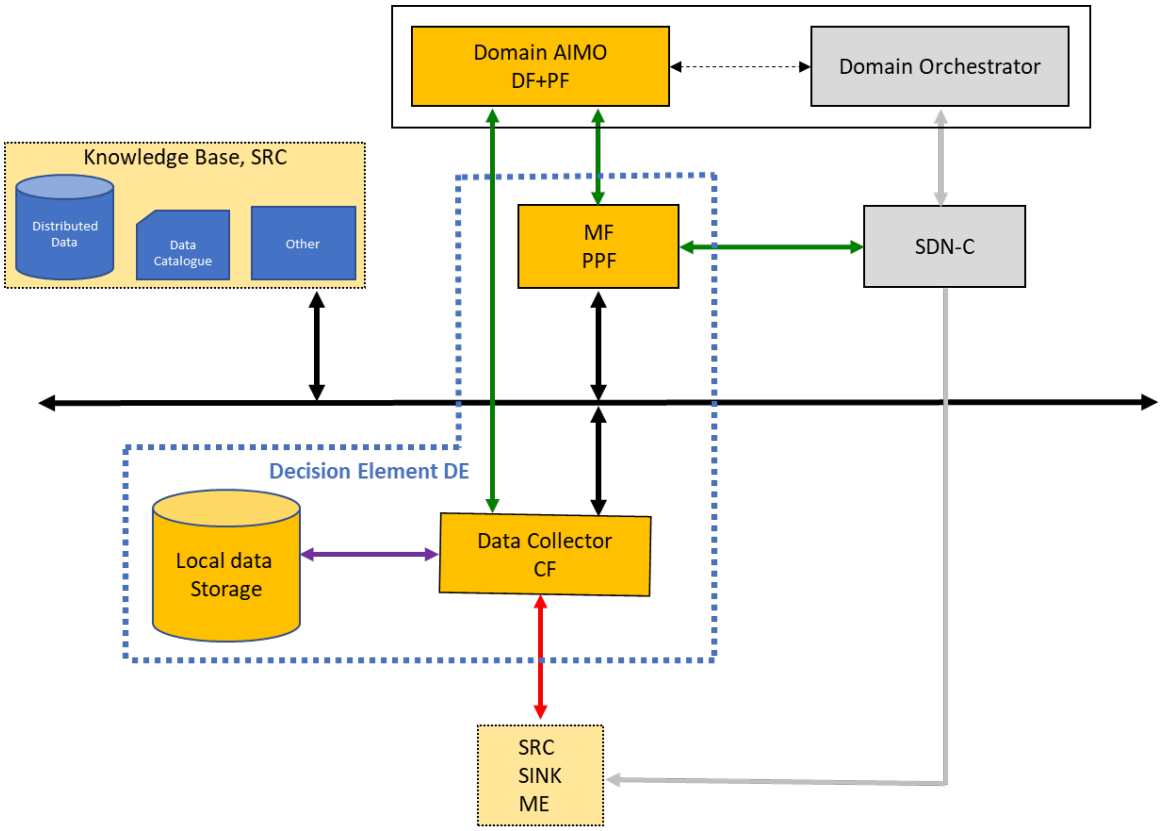

Figure 8: AIM Functional Blocks - Logical Diagram

The green lines represent management, control and orchestration function.

The purple line represents write and read operations to/from the local storage.

The red line represents collection functions.

Finally, the black line represents the message bus used to exchange information between the AIM blocks and the Knowledge Base.

The grey lines represent management functions that are not in the scope of the AIM. As an example, Figure 9 provides a more detailed explanation on how the AIM blocks may be orchestrated to implement the most generic ITU-T pipeline where the PPF and MF components are implemented separately.

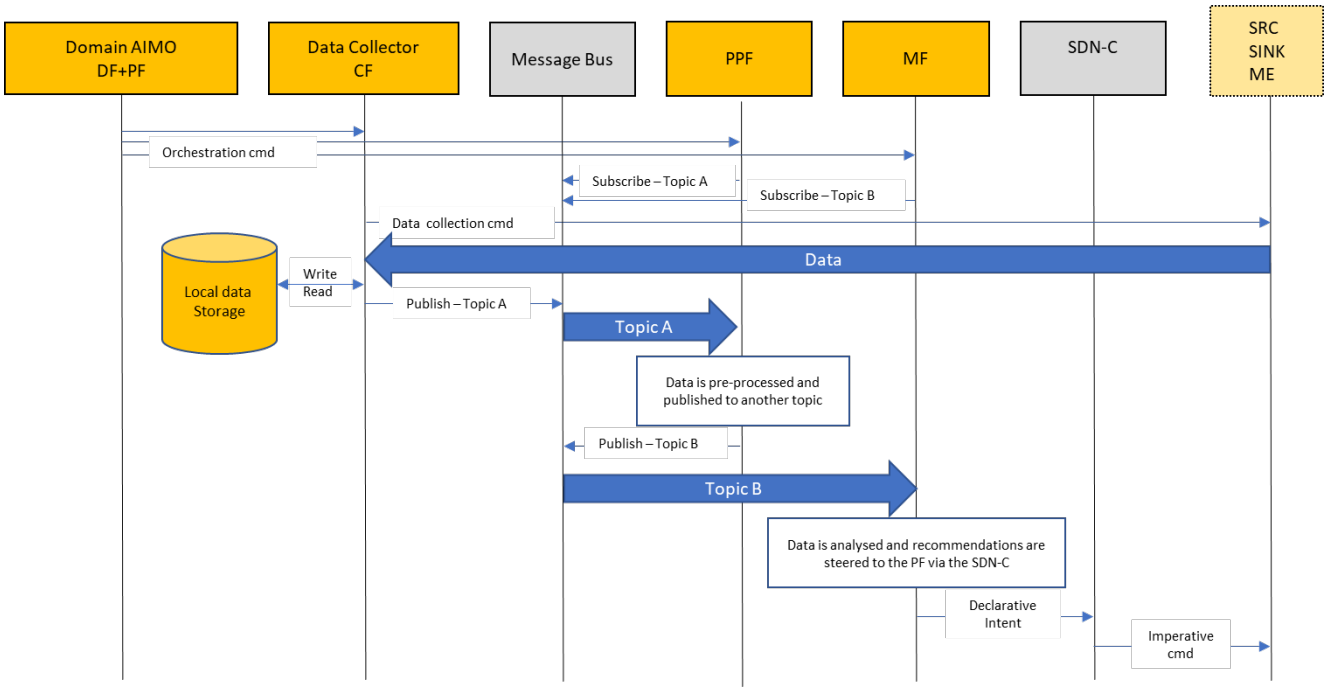

Figure 9: AIM Pipeline Message Flow Example

The Pipeline described in Figure 9 provides an example of an automated closed loop with one single level of intelligence. Once the data is collected from the Managed Entity, The PPF component pre-process the data for the MF component. The MF component will provide its recommendations to the controller via declarative Intent. The controller will translate the Intent into atomic commands to the Managed Entity in order to close the loop and self-heal and/or self-optimize the network.

The logical flow of the messages between the AIM functional blocks is explained:

1. The Domain AIM Orchestrator configures the AIM functional blocks and activate the pipeline. In particular:
    a. It orchestrates the data collection jobs on the CF and specifies what Topic to use to publish the collected data on the message bus;
    b. It specifies what Topic the PPF must use to fetch the data and what Topic to use to publish the processed data;
    c. It specifies what Topic the MF should use to read the data and where to send its recommendation;
    d. It may enforce some policies on the AIM components.
2. The PPF subscribes to “Topic A”
3. The MF subscribes to “Topic B”
4. The CF sends a data collection command to the Managed Entity and starts collecting the data
5. The CF may also write/read to/from the Local Data Storage
6. The CF publishes the collected data to “Topic A”
7. The PPF reads the data from “Topic A”, processes the data and publish to “Topic B”
8. The MF reads the data from “Topic B”, analysis the data and provide a recommendation to the controller using a declarative Intent
9. The controller translates the declarative Intent into specific configuration changes to the ME using imperative commands

6.6 Decision Element

According to the logical implementation of the AIM DE presented in Figure 8, the AIM logical diagram may be simplified as described in Figure 10. The AIM framework does not prescribe any particular implementation and the Local Data Storage, the CF, the PPF and the MF can possibly be implemented as a single monolithic software function.

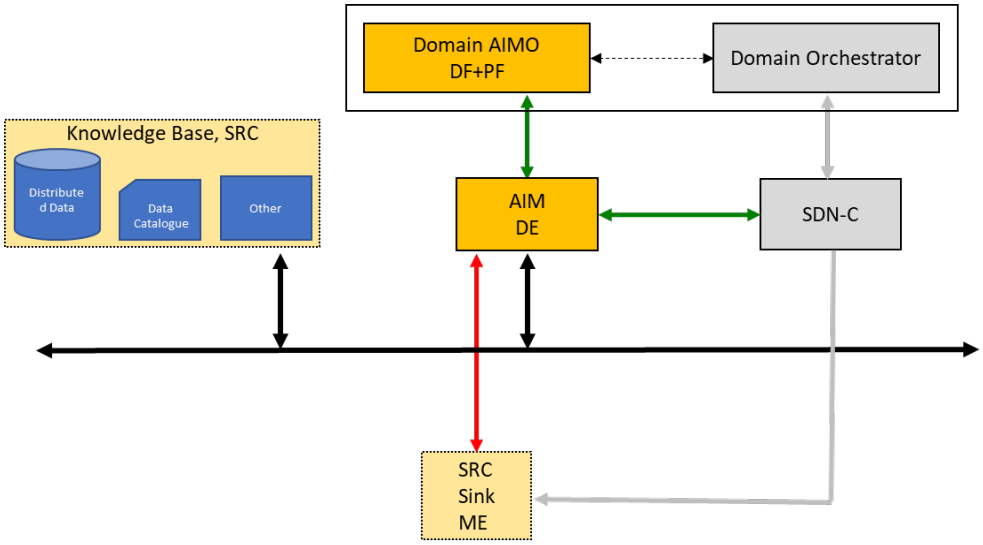

Figure 10: AIM Functional Blocks - Simplified Logical Diagram

R-87 The Local Data Storage, the CF, the PPF and the MF SHOULD implemented as micro-services packaged in containers and deployed as VNF in order to optimize the sharing of the resources, support In Service Software Update (ISSU) and automatic scaling up/down.

In particular, Figure 11 shows how three different AIM DEs may be logically implemented by sharing the same Local Data Storage and the same CF component. Furthermore, as described in the previous sections, the AIM functional blocks are agnostic with respect to what control loop they serve and expose their services to the other AIM components or 3rd party applications thus implicitly being able to be pipelined into multiple control loops at a time.

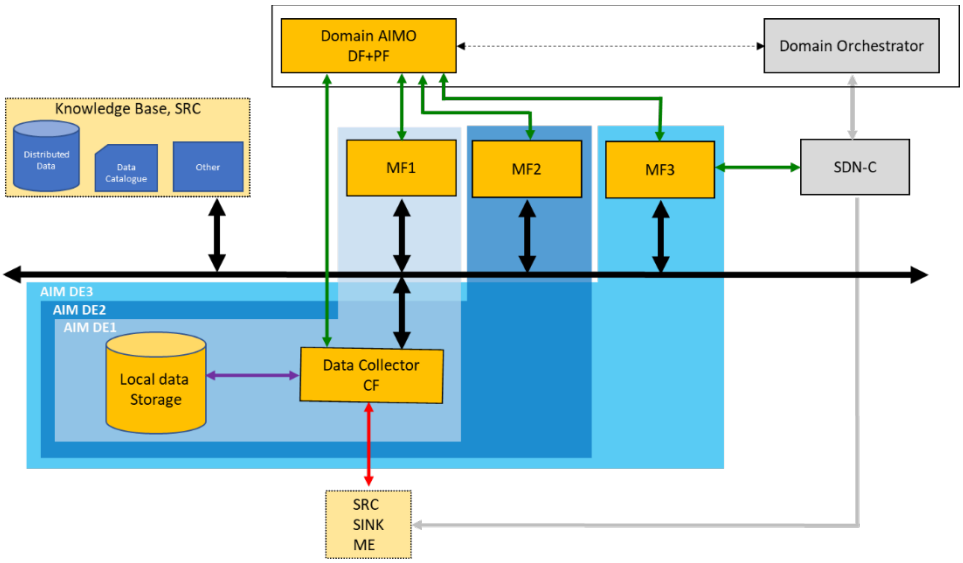

Figure 11: AIM DE - Resource Sharing

R-88 Regardless of the specific implementation of the AIM DE, the exchange of information and the access to the services exposed by the Knowledge Base, including but not limited to Data Catalogue and Distributed Data, SHOULD be implemented using a message bus via a publish/subscribe or request/response mechanism

6.7 Control Loop and Hierarchical Pipeline

As described in Sections 6.5 and 6.6, the AIM functional blocks are chained and orchestrated in a pipeline to implement a control loop. Pipelines can possibly include any combination and any number of the ITU-T components, and, the number of AIM functional blocks and/or how they are chained within a pipeline varies and it depends on the specific implementation and on the specific use-case.

Figure 7 provides some examples of basic pipelines but, for instance, a more complex logical AIM DE may be implemented chaining multiple PPF and MF component in the same pipeline (for example to break down a complex task into multiple simple tasks) as illustrated in Figure 12.

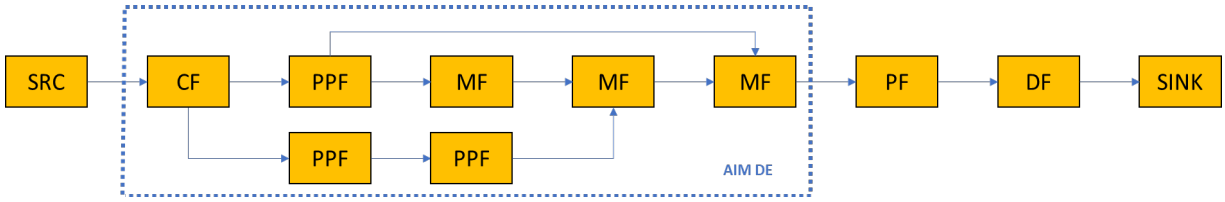

Figure 12: Complex DE Example

While multiple logical AIM DEs may be chained within a pipeline, a pipeline that starts with an SRC and ends with a SINK implements a control loop. Closed Loop Application (CLA) and/or Open Loop Application (OLA) may be implemented by a single control loop or by nested hierarchical control loops that may share their outcomes through the message bus or through the Knowledge Base. The AIM inherits the four hierarchical control loops levels defined by ETSI GANA and described in I.1.

Figure 13 illustrates an example of hierarchical pipeline with two levels of control loops.

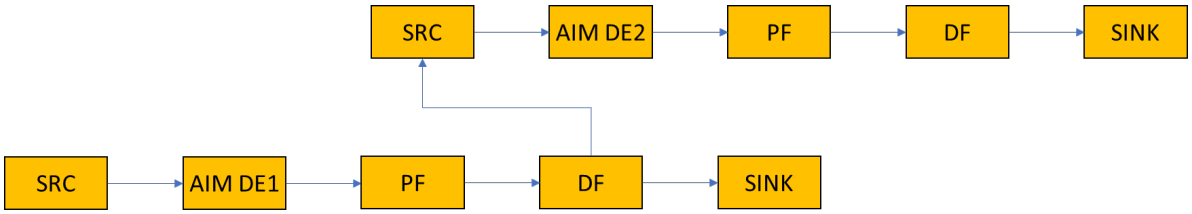

Figure 13: AIM Hierarchical Pipeline

R-89 The Domain AIM Orchestrator MUST orchestrate the nested control loops in the hierarchical pipeline. The output of the lower control loop implemented with the AIM DE1, will be part of the input data (SRC component) of the higher control loop implemented with the AIM DE2.

Similarly to the AIM functional blocks that are agnostic with respect to what control loop they are part of,

R-90 Control loops MUST be agnostic with respect to the CLA/OLA hierarchy they belong to.

R-91 The Domain AIM Orchestrator MUST provide control loops with the information, among others, on where to publish/store their outcomes and from where to fetch the outcomes of the inner loops they need to operate with (if any).

Figure 14 provides an example of how a hierarchical pipeline can be logically implemented. In this particular example, two hierarchical levels of control loops are implemented, and both provide their recommendations to the controller and publish their results to the message bus. The inner control loop implemented with the AIM DE1 operates independently from the higher control loop while the higher control loop implemented with the AIM DE2 consumes the results of the inner loop to provide its recommendations. Furthermore, both DEs analyses the data collected from the Managed Entities respectively ME1 and ME2.

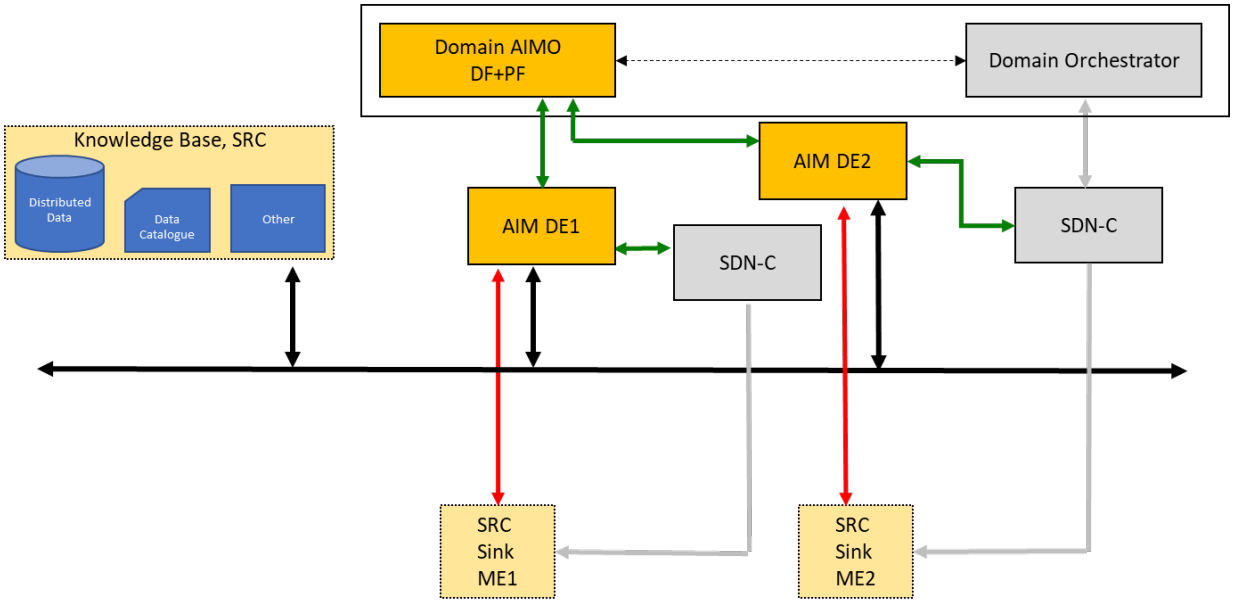

Figure 14: AIM Hierarchical Pipeline - Logical Diagram

6.8 Reference Points

This section specifies the AIM reference points as part of the detailed logical and functional AIM architecture. Figure 15 represents the logic and functional relationships among the blocks of the AIM architecture, via relevant reference points and related interfaces. The reference points are represented with respect to the most generic AIM functional block the AIM DE, but the same reference points apply to all AIM functional blocks.

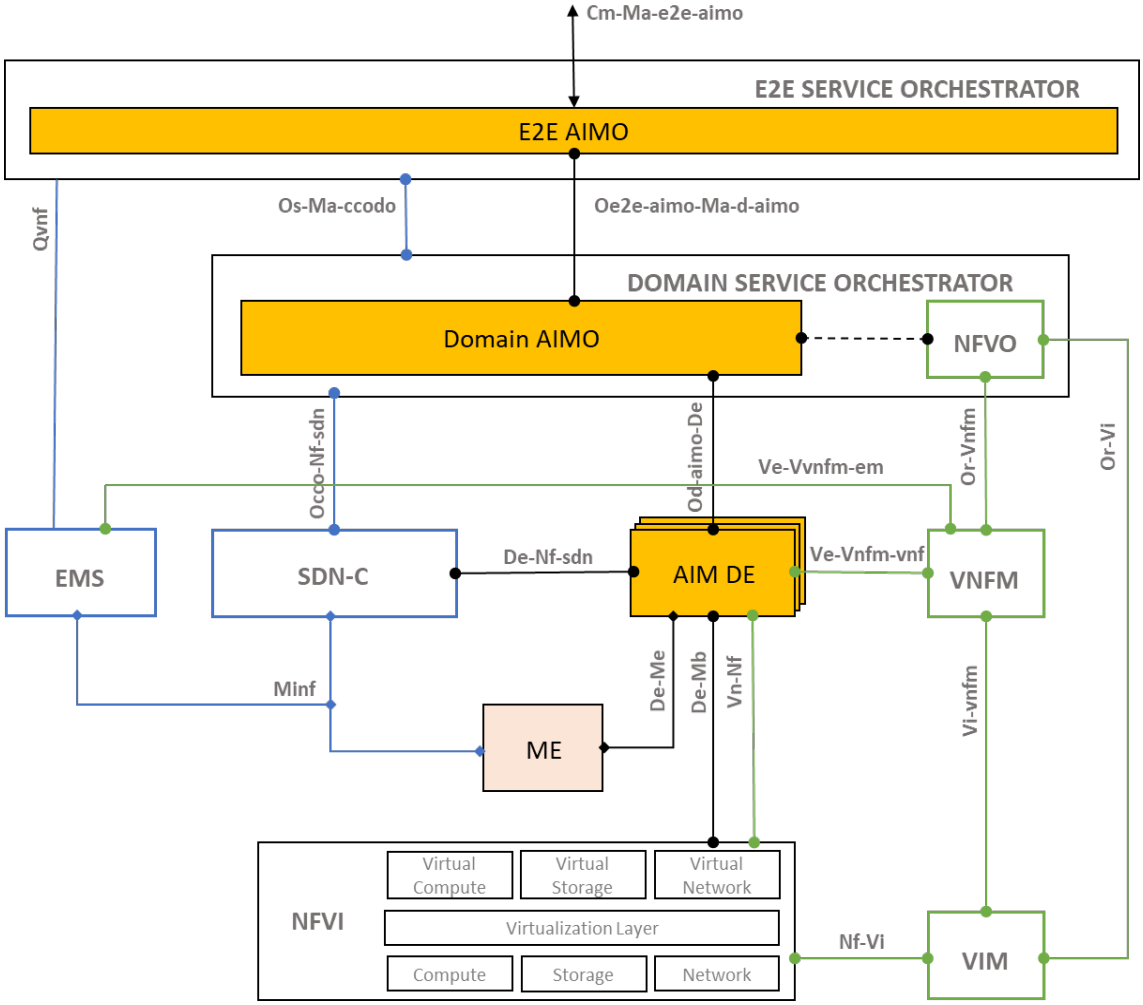

Figure 15: AIM Reference architecture

The green colored boxes and lines are related to NFV reference points as identified mainly in ETSI standards. The blue boxes and lines are related SDN reference points as identified in BBF standards.

R-92 The AIM NBI is labelled as Cm-Ma-e2e-aimo and it SHOULD be implemented as an extension of the TMF641 [19], TMF645 [22] reference points, to encompass the Customer Management to AIM E2E Orchestration layer relationship.

The AIM reference architecture is a combination of an SDN and NFV architecture applied over an NFV infrastructure. The NFV portion is used to orchestrate and manage the AIM virtual functions and their supporting infrastructure, while the SDN portion may be used to manage and control the AIM Managed Entities. In order to build pipelines, including configuring and setting up connectivity to AIM VNFs on-boarded on the NFVI, all these resources have to be orchestrated together.

This is the role of the Domain AIM Orchestrator function in collaboration with the E2E AIM Orchestrator function whose detailed functional requirements are defined in Section 6.2.3 alongside those below.

The E2E AIMO function is a central function in the architecture and it is also home to the AIM NBI, and as such, in collaboration with the Domain AIMO, it delivers the necessary Service Abstraction Layer, hiding the internal operations of the AIM from the NBI.

R-93 The E2E AIMO function MAY be implemented as a stand-alone Orchestrator or a function of the E2E Service Orchestrator.

R-94 The Domain AIMO function MAY be implemented as a stand-alone Orchestrator or as a function of the Domain Service Orchestrator.

R-95 When the Domain AIMO is stand-alone element, it MUST also implement the essential functionalities and interfaces needed to interact with the embedded NFVO.

R-96 When the Domain AIMO is a function of the Domain Service Orchestrator, it SHOULD be responsibility of the Domain Service Orchestrator to implement the functionalities and interfaces needed to interact with the NFVO.

This integration is key to guarantee effective operation of the AIM DEs over the management and control plane, while the MANO components ensure that the VNFs over the user plane evolve in space and time as required by the state and configuration transitions of the AIM DEs.

R-97 The Domain AIMO MUST operate its tasks via a set of standard SBI towards the AIM DE and as directly instructed via the E2E AIMO.

As shown in Figure 11 these Southbound Interfaces (SBIs) govern the Management & Control and User Plane of the AIM DE.

R-98 The Domain AIMO MUST translate the declarative Intent received from the E2E AIMO into imperative commands to the AIM DE.

The NFV part closely follows the ETSI Reference model (NFVI and MANO). The ETSI architecture allows one or multiple VIMs per NFVO. Various VIMs and VNFMs may be deployed as part of the solution depending on the requirements and the Operator's implementation strategy.

The SDN Management and Control elements are not in the scope of the AIM framework specified in this Technical Report, although the logical interface/reference point between the AIM DE and the SDN controller is in scope. An SDN controller should control the physical infrastructure as well as all associated virtual function described as AIM Managed Entities. The SDN Controller is the sole in charge of managing and configuring the Managed Entities under its control and applies polices to resolve any conflict that may arise. The architecture allows one or multiple SDN Controllers, in which case the AIM Domain Orchestrator will orchestrate the Pipeline to use the correct one.

R-99 The AIM DE MUST operate its tasks as instructed by the Domain AIMO and it MUST communicate with other DEs and the Knowledge Base using the message bus exposed on the NFVI.

R-100 The AIM DEs, through its CF component, SHOULD be the sole in charge of running collection campaigns to retrieve operational and telemetry data from the Managed Entities.

Finally, and as explained in the previous paragraph:

R-101 The AIM DE MUST communicate with the SDN Controller using Intents in order to implement its recommendations on the Managed Entities.

An Intent is a declarative description of an application, which is used to abstract any technology-specific network functions. Furthermore, the De-Nf-sdn reference point may be derived as an extension of the Occo-Nf-sdn reference points defines by the BBF (Occo-Nf-sdn-pnf, Occo-Nf-sdn-vnf, Occo-Nf-sdn-dc).

The ETSI-NFV reference points relevant for the AIM architecture are defined in [7], [8], [9], [10], [11], [12] and [13].

The BBF CloudCO reference points that are relevant for the AIM architecture are defined in [2] and [23].

Table 2 specifies additional reference points required for the interactions among the blocks defined in the AIM architecture.

R-102 The reference points in Table 2 MUST be supported by the corresponding elements shown in Figure 15.

Table 2: AIM Reference Points

Reference Point	Location	Description
Cm-Ma-e2e-aimo	Between the Customer Management layer and the AIM E2E Orchestrator	The Cm-Ma-e2e-aimo reference point allows exposing an abstracted view of the AIM Domain resources to the Customer Management layer. This reference point may be an extension of, and expose directly the functions associated to TMF641 [21], TMF645 [22], TMF633 [19], TMF638 [20] like Service Catalogue, Service Inventory, Service Ordering and Activation and Service Qualification.
Oe2e-aimo-Ma-d-aimo	Between the E2E AIMO and the AIMO	The Oe2e-aimo-Ma-d-aimo reference point allows exposing an abstracted view of the AIM Domain resources to the E2E AIMO. Given the AIMO contains a NFVO, this reference point may expose directly to the E2E AIMO the functions associated to Os-Ma-Nfvo reference point: management of Network Service Descriptors and VNF packages; lifecycle management of Network Services and VNFs; policy management and/or enforcement of those resources and the NFVI in general. Alternatively, the Os-Ma-Nfvo reference point may be embedded in the AIMO, which would be then the only block to interface with the E2E AIMO. This reference point may be an extension of, and expose directly the functions associated to TMF641, TMF633, and TMF638 like Service Catalogue, Service Inventory and Service Ordering and Activation.
Od-aimo-De	Between the AIM Domain Orchestrator and the AIM Decision Element	The Od-aimo-De is the reference point for the AIMO to interact with the AIM functional blocks to configure, manage and orchestrate.
De-Nf-sdn	Between the AIM Decision Element and the SDN Controller	The De-Nf-sdn is the reference point for the AIM DE to interact with the SDN Managers and Controllers. This reference point may be derived as an extension of the Occo-Nf-sdn-xxx reference points defines by the BBF (Occo-Nf-sdn-pnf, Occo-Nf-sdn-vnf, Occo-Nf-sdn-dc)
De-Me	Between the Decision Element and the Managed Entity	The De-Me is the reference point for the collection function of the AIM DE to collect data from the managed Entities via streaming, polling etc.
De-Mb	Between the Decision Element and the Knowledge Base and between Decision Elements	The De-Mb is the reference point for the AIM DE and its internal components to exchange information between DEs and the Knowledge Base and consume 3rd party services like data cataloguing etc.

Table 3 and Table 4 describe the ETSI NFV and additional BBF repositories that support the NFVO function operating in the context of the AIM framework.
The tables represent examples that should be refined at the stage of actual software implementation.

The catalogue of orchestration, control, and management packages in Table 5 assumes that these functions are implemented as VNFs.

Table 3: ETSI NFV Data Repositories

Catalog	Primary Entity	Description
NS Catalog	NFVO	Represents the specifications of all of the on-boarded Network Services, VNFs and NFVI Resources. The NS Catalog is further described in clause 5.4.4 of [7]
VNF Catalog	NFVO, VNFM	Represents the specifications of all of the on-boarded VNF Packages, supporting the creation and management of the VNF Package (VNF Descriptor (VNFD), software images, manifest files, etc.). The VNF Catalog is further described in clause 5.4.5 of [7].
VNF Instance Repository	NFVO	The VNF Instances repository holds information of all VNF instances and Network Service instances. Each VNF instance is represented by a VNF record, and each NS instance is represented by an NS record. Those records are updated during the lifecycle of the respective instances, reflecting changes resulting from execution of NS lifecycle management operations and/or VNF lifecycle management operations. The VNF Instance Repository is further described in clause 5.4.6 of [7].
NFVI Resource Repository	NFVO	Represents information about available/reserved/ allocated NFVI resources as abstracted by the VIM across Operator's Infrastructure Domains. The NFVI Resource Repository is further described in clause 5.4.7 of [7].

R-103 The AIM data repositories in Table 4 MUST be supported by the Primary Entities indicated in the table.

Table 4: AIM Data Repositories

Catalog	Primary Entity	Description
AIM Pipeline Catalog	E2E AIMO, NFVO	Represents the specifications of all of the on-boarded AIM Pipelines (in the E2E AIMO) and AIM VNFs (in the NFVO).
AIM VNF Catalog	NFVO, NFVM	Represents the specifications of all of the on-boarded AIM VNF Packages, supporting the creation and management of the VNF Package (VNF Descriptor (VNFD), software images, etc.).
AIM Instance Repository	NFVO, E2E AIMO	It holds information of all AIM VNF instances (in the NFVO) and AIM Pipeline instances (in the E2E AIMO).

6.9 Pipeline Orchestration

This section presents two key phases in the operation of an AIM framework that require orchestration by the E2E and Domain AIMO in collaboration with the NFVO. The two phases are presented with the sole intent to better explain how the components of the architecture presented throughout Section 6 are orchestrated in a pipeline. It further specifies the use of the Data Repositories described in Table 4.

6.9.1 AIM VNF Deployment

This phase, represented in Figure 16, describes how the Domain AIMO typically performs the instantiation of the AIM VNF components in collaboration with the other management and orchestration functions implemented by the NFVO and the functionalities exposed by the VNFM, which configures VIM (Appropriate VIM API e.g.: VMware, Openstack, Kubernetes) and instantiates the NFVI resources.

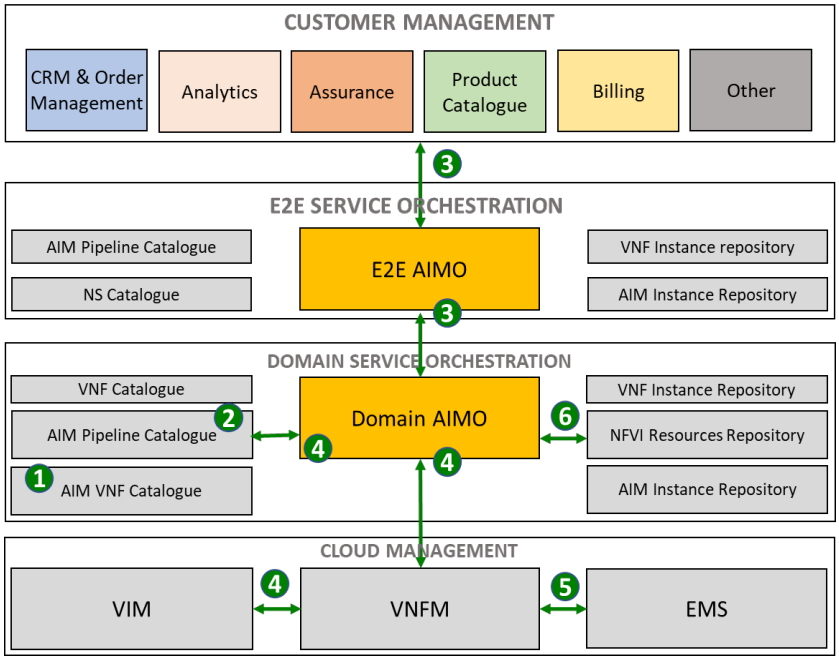

Figure 16: AIM VNF Deployment

The logical message flow in Figure 16 is explained:

1) A VNF Descriptor (VNFD) is created
2) A VNF package is on-boarded and an AIM Component is created (e.g., with VNF & connectivity). The AIM NS template is added to the AIM Pipeline Catalogue, ready for consumption
3) An AIM Order is raised from the BSS (for instance an assurance order)
4) The NFVO retrieves AIM NS template from the AIM Pipeline Catalogue & distributes to VNFM, which configures VIM (Appropriate VIM API e.g.: VMware, Openstack, Kubernetes)
5) The EMS performs application level configuration on the created VNFs per Life Cycle Management (LCM) operation
6) The Domain AIMO updates the NFVI Resources Repository

6.9.2 AIM Pipeline Instantiation

This phase, presented in Figure 17, describes how the Domain AIMO receives a declarative Intent from the E2E AIMO and creates an AIM pipeline that conforms to the specification and requirements of the AIM Order sent to the E2E AIM Orchestrator by a Customer Management layer application. A pipeline is realized by

instantiating AIM components (e.g., SRC, CF, PPF, MF, SINK) and taking into account the requirements/attributes of the Intent and the capabilities of the underlying infrastructures. The Domain AIMO allows flexible placement, distribution, chaining and execution of its components to facilitate the implementation of complex AIM Pipelines.

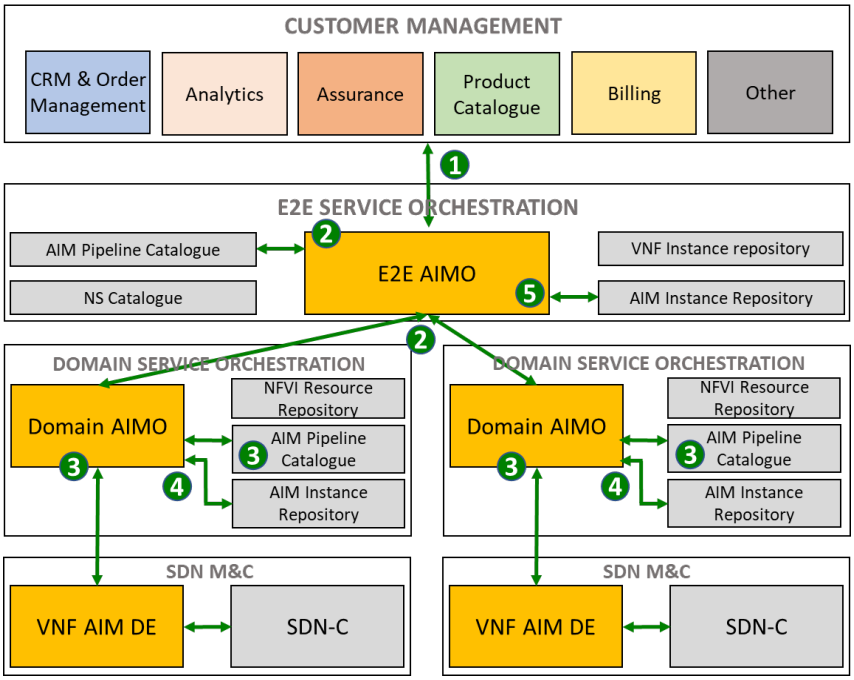

Figure 17: AIM Pipeline Instantiation

The logical message flow in Figure 17 is explained:

1) An AIM Order is raised from the BSS (for instance an assurance order)
2) The AIM Order is mapped to the CFS and the E2E AIMO requests the Domain AIMO the creation and activation of the RFS via declarative Intent
3) The Domain AIMO maps the Intent and the AIM Order's attributes to the RFS from the AIM Pipeline Catalogue and activates the pipelines
4) The Domain AIMO updates the AIM Instance Repository
5) The E2E AIMO updates the AIM Instance Repository

6.10 Domain Federation

Section 5.3, introduces the concept of domain federation focusing on the cross-domain interfaces between domain-specific DEs.
Generally speaking, a multi-domain SDN architecture can take advantage of network domain federation to improve automation and operational agility of its management and control chains.

Similarly, the deployment of the AIM framework as an empowerment of such SDN architecture can leverage on domain federation techniques especially for higher-level Automated Loops (see Section 6.7) whose decision scope spans across multiple domains.

One example is the decision making around the root cause of a network problem observed from multi-domain perspective and to provide the best possible diagnosis and recommendations thereof.

R-104 The AIM DEs within a network domain SHOULD analyze the information fetched (and possibly rationalized) from other domains alongside the detailed information derived from the "monitored" MEs implementing a cognitive behavior.

The cognitive behavior implemented by the AIM DEs, takes into consideration the environment, suggests actions according to input from other sources of information and network policies, decides which recommendation fits best its end-to-end purpose, and finally acts on the recommendation.

R-105 A two-phase commit approach SHOULD be implemented to coordinate the DEs in a pipeline spanning across multiple domains. Commit or roll back of the provided recommendations become then a matter also of the consensus/feedback obtained from federated domains.

Borrowing again from the previous multi-domain service assurance and troubleshooting example: the effectiveness in detecting a real or potential fault heavily relies on the ability to perform diagnostics to get additional perspectives on faults. This should be done before reaching a conclusion on the type of problem and determining which action(s) to take. This behavior becomes more important as the (trouble)shooting angle gets higher and broader as well as the network becomes more complex and autonomous.

Referring to the end-to-end purpose of the AIM DEs, it is useful to identify two conceptual categories:

- **Domain-centered DE**: These AIM DEs analyze information from federated domains in order to assist with the identification and resolution of issues in their own domain. Furthermore, a domain-centered DE, by definition, must not steer recommendations to other domain, even if some federated and trusted relationship is structurally established between the two.
- **Holistic DE**: These AIM DEs, lay at higher-levels of the AL hierarchy and may analyze information from federated domains and apply cognitive approach models to infer an holistic view and contribute to the end-to-end decision-making process (e.g., service issue, upselling/cross-selling opportunity, sophisticated customer profiling).

The OASIS Web Services Federation Language specification describes that "*The goal of federation is to allow security principal identities and attributes to be shared across trust boundaries according to established policies.*" It involves having common standards and protocols to manage and map user and/or application identities (Entities) between Identity Providers across organizations (and security domains) via trust relationships.

In such federated system, the concept of Identity Provider (IdP) is responsible for the authentication, and a Service Provider (SP), intended as a micro-service or an application, controls access to resources. The SP trusts the IdP by administrative agreement and configuration, to authenticate Entities and relies on the information provided by the IdP about them. After authenticating the Entities, the IdP sends the SP a message, called an assertion, containing the Entity's sign-in name and other attributes that the SP needs to establish a session with the Entity and to determine the scope of resource access that the SP should grant.

Besides information exchange across network and/or business domains inside an organization, federation relies on access control systems.

In the AIM framework described in the previous sections, the potentially high number of DEs and other components that exchange information calls for managing their identities via a central IdP and for swiftly governing their access to multiple applications and services acting as SPs.

Concerning the communication between the AIM components implemented as micro services:

R-106 An identity MUST be provided for those services that provide services to other services.
3GPP TS 23.501 [14] defines a Service Based Architecture (SBA) as a system architecture based on a set of NFs providing services to other NFs authorized to access their services. Furthermore, Section 6.7.3 of ETSI TS 129 500 [15] specifies how the NFs may be authorized to access the services provided by the NF service providers.

Finally, according to ITU-T Y.3172 [17], a "Service-based architecture (SBA) [b-ETSI TS 129 500] may be used to interface ML functionalities with ML underlay networks. Similarly, for the ML pipeline in the sandbox, SBA may be used to interface the ML functionalities with the simulated ML underlay networks."

Although how to accomplish authentication and authorization between microservices, and a federation relationship among different entities, out of scope for this Technical Report, it is noted that several protocols and standards are available in the industry to achieve those goals depending on the implemented request-response or message/event based architecture. It can also take different forms depending on how many IdPs and SPs are involved and how they relate with each other. The Operator should decide on the best option for them according to their own requirement and security policies.

Some examples of authentication and authorization protocols are: Simple Authentication and Security Layer (SASL), JSON Web Token (JWT), service-mesh, Security Assertion Markup Language (SAML), OAuth2 etc.

The following sections describe the interaction of the AIM components across federated network domains assuming the necessary trust relationships have already been established.
Section 6.10.1 explains how cross-federation can be leveraged to implement Domain-centered DEs.
Section 6.10.2 applies similar concepts to the implementation of Holistic DEs.

6.10.1 Domain-centered DEs

AIM domains are best defined as **a grouping of AIM components under the same administrative and security control.** The Operator may decide to organize its business and operational workflows based on network and technology domains (Access, Transport, Core etc.) and/or business domains (Sales, Assurance, Engineering etc.) or any other organizational structure they design for.

As discussed in Section 5.3, under an AIM perspective, there is merit in federating network domains to steadily exchange information among them to improve the decision process of a Dom1-DE based not only on the inputs from Dom1-MEs but also on inputs received from Dom2-DE and vice-versa.
An Operator could decide to group the DEs under one single overall domain. Although this is not considered as a best practice, it may suit the requirements of a small Operator or a Virtual Network Operator who does not own the infrastructure and have a very lean organizational structure.
For broader and domain-structured deployments, it is typical to have AIM DEs deployed in each domain, having a domain specific scope and acting upon a set of Managed Entities under their own domain.

In such typical AIM architectures (well described in the previous sections):

R-107 Cross-domain sharing of information, aimed at decision and automation improvements, SHOULD be achieved through domain federation.

Secured, policy-based and effective information exchange among intra- and cross-domain entities, hence becomes a further aspect.

R-108 Security and Access policies MUST be enacted and enforced via the E2EAIMO and the Domain AIMO.

R-109 Intra- and cross-domain AIM pipeline communications MUST be based on permissions granted to each AIM component involved.

R-110 Data generated by the operation of pipelines MUST be shared among AIM components without involving any AIM orchestration function.

R-111 The sharing of information (blue lines shown in Figure 18) among federated AIM domains SHOULD be enabled by connecting the domains and the Knowledge Base to a common message bus.

Although how to architect the message bus is out of scope for this Technical Report, in the particular scenario presented in Figure 18, each domain uses its own message bus (black lines) for internal exchange of information among AIM components while a separate message bus is set up for inter-domain and domain to Knowledge Base communication.

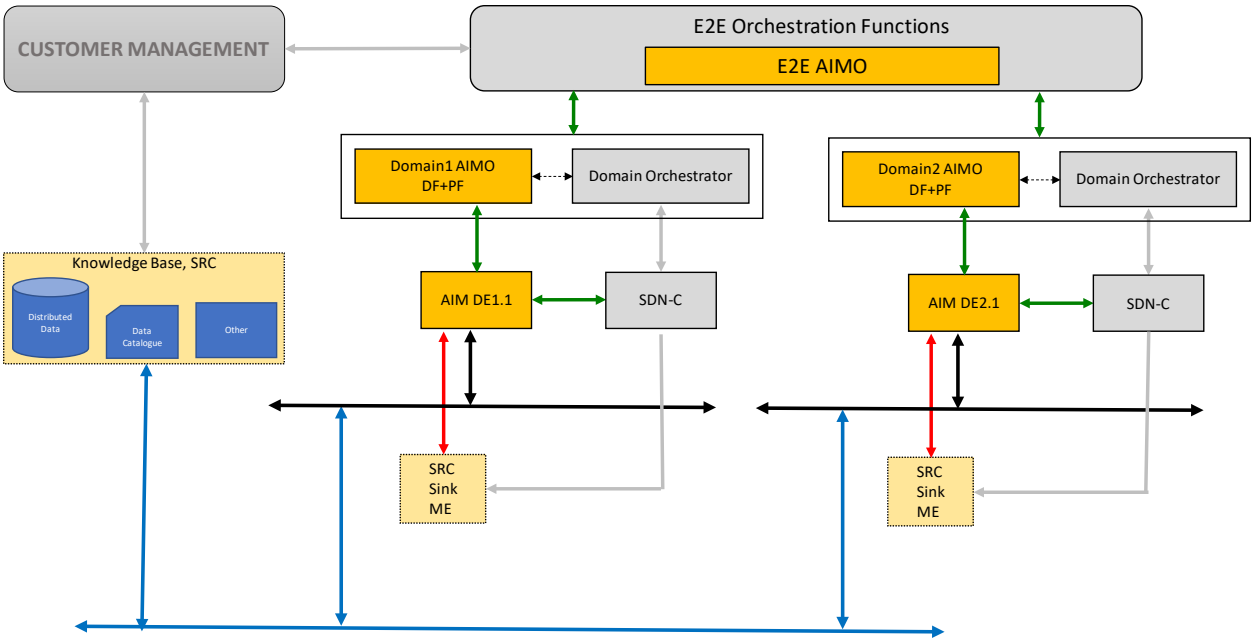

Figure 18: Inter AIM Domains federation

There are mainly two security aspects related to the service-to-service communication paradigm. The first is to make sure that a microservice is allowed to communicate with another microservice by means of cloud-based security enforcement, which is a well consolidate industry practice. The second is to make sure the identity of the microservice is known in order to enforce Authentication, Authorization and Accounting (AAA).

R-112 Access to domain resources stored in the Knowledge Base through the common message bus SHOULD be authorized by an IdP in collaboration with functions implemented at the Customer Management layer.

Figure 19 shows an example of a trusted-relationship scheme among DEs that should be established in order to allow such communication. The example shows a 2-level Automated Loop hierarchy in Domain 1 and in particular DE1.1 and DE1.3 are allowed to share information with DE2.1 on the common message bus.

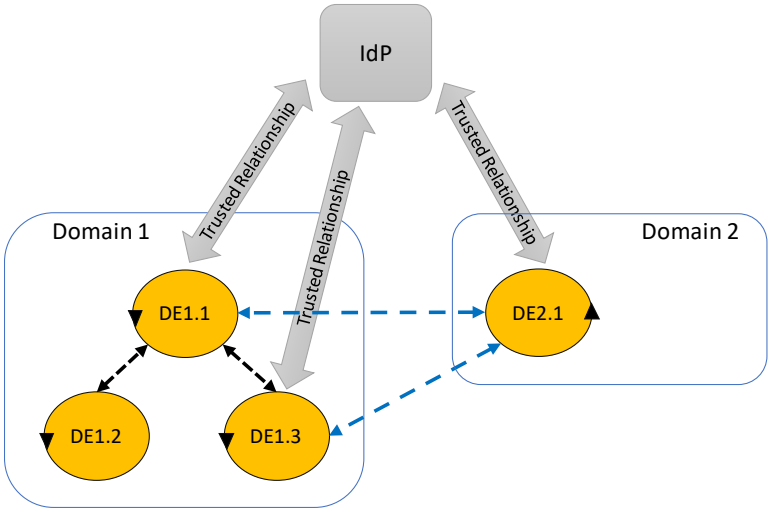

Figure 19: Inter AIM Domains trusted relationships

R-113 The reference point “De-Mb” specified in Section 6.8 MUST be used for the exchange of information among federated domains.

As a recommended architectural principle, the streaming of data across federated domains/DEs must not involve the orchestration functions whose complexity and load shall be focused on the AIM pipelines lifecycle, orchestration and optimization rather than on the data transfer tasks carried out by the components of the AIM pipelines themselves.

According to point 2) in Section 5.3, the Federation is achieved via the coordination of the E2E orchestration functions that take part (but not limited to) in the pipeline components instantiation and deployment workflows. In particular:

R-114 The E2E orchestration functions MUST take care of configuring the data needed to determine the identity and the scope of access of the AIM components via the “Cm-Ma-e2e-aimo” reference point described in Section 6.8.

6.10.2 Holistic DEs

As mentioned in the Section 6.10.1, one aspect related to end-to-end scoped AIM DEs is to analyze information from federated domains to achieve a holistic view and a more effective decision making.

Section 6.9.2 describes how the instantiation of the AIM Pipelines starts from an AIM order originated from the Customer Management layer, for instance originated by a Service Assurance platform. The Customer Management layer is the only one who has knowledge of the Customer Facing Service (CFS), complete view of its facets and, when applicable, of its associated Service Level Agreement (SLA) including Service Level Objects (SLO) and Service Level Indicators (SLI). Furthermore, an end-to-end decision-making process can be fully understood and effectively carried out at the layer where the AIM order was originated.

The Customer Management layer may include functions like workflow/intent engine, Analytics & AI, Correlation, end-to-end root cause analysis, customer experience accounting, topology discovery etc.

R-115 The Customer Management layer functions SHOULD interact via the “Cm-Ma-e2e-aimo” with E2EAIMO and use the information published by Holistic DEs on the common message bus or through the Knowledge Base.

R-116 The Customer Management functions that require access to the data from the Domain-scoped DEs MUST establish a trust relationship and federate with those DEs via an IdP in a similar way to what described in section 6.10.1 for the federation among AIM domains.

There is not a substantial difference between establishing a trust relationship between DEs or between a DE and any service/function in other domain.

The main difference is on the purpose and implementation of the Holistic DE.:

R-117 The Holistic DE SHOULD provide information on the health of its Managed Entities (“holistic” information), how it contributes to the health of the end-to-end service and how it may be impacted by decisions/actions taken in other domains from the perspective of its own domain.

R-118 The “holistic” information MUST be published on a common message bus to be consumed by the higher control loops in the hierarchy implemented at the customer management layer.

Figure 20 shows an example on the trusted relationship among DEs and other domains that should be established in order to allow such communication. The example shows a 2-level control loop hierarchy in domain 1 and in particular DE1.1 and DE2.1 are allowed to share information with the higher loop implemented at the Customer Management domain on the common message bus.

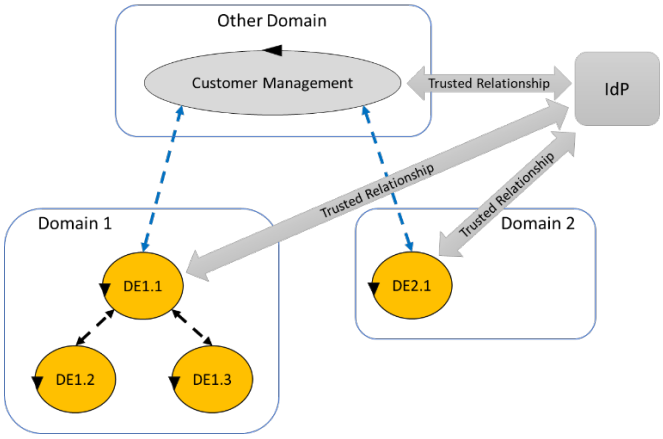

Figure 20: AIM Domains and other Domains trusted relationship

The functions implemented at the Customer Management layer may be authorized to access information stored in the Knowledge Base by each AIM domain and vice-versa. For instance, the DEs in a domain may be authorized to access information stored in the Knowledge Base by a ticketing system etc. via the common message bus (blue lines) as shown in Figure 21.

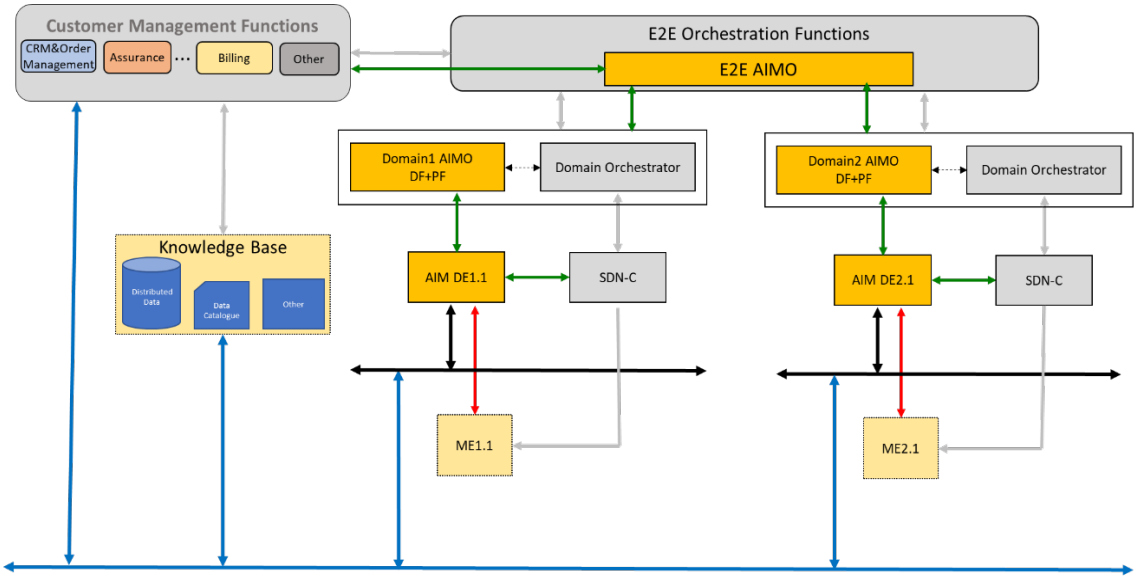

Figure 21: AIM and other Domains federation

The AIM framework specified in this Technical Report, does not specify how the Automated Loops instantiated at the Customer Management layer should work or what type of intelligence they should implement.

R-119 The Refence Points between Customer Management layer and this AIM framework MUST be the “De-Mb” and “Cm-Ma-e2e-aimo” reference points described in Section 6.8 and the interactions enabled by them.

6.10.3 AIM Sandbox Logical Subsystem

The main functionalities of the AIM Sandbox Logical subsystem are presented in Section 6.2.2, while Section 6.2.3 explains how most of those functions are orchestrated and managed through the Management Subsystem.

The scope and the complexity of the AIM Sandbox could start with offering the capability of training an algorithm before deploying it or it can become a replica of the AIM Live Pipeline deployment, including model(s) selection, design, training, testing, update and lifecycle management functionalities.

Furthermore, the implementation of the AIM Sandbox may vary depending on the Operator's business and technical requirements.:

R-120 The AIM Sandbox functionalities MAY be implemented (but not limited to) as:

1. An AIM Domain
2. A number of AIM Domains
3. Functionalities and/or services offered by the Customer Management Layer
4. Functionalities and/or services offered by a 3rd party organization
5. A combination of the above

According to ITU-T Y.3172 [17], the AIM Sandbox should be implemented as an isolated domain, which allows the hosting of AIM components and pipelines for the aforementioned purpose before live deployment. Therefore, regardless of the specific implementation of the AIM Sandbox, it must federate with the AIM Domains it is supporting, following the same principles described in sections 6.10.1 and 6.10.2.

In its more generic implementation:

R-121 The AIM Sandbox MAY be a federated space on its own as described in Section 6.10.1 that federates with the AIM Domains and the other domains like the Customer Management layer in order to achieve its goals.

Figure 22 shows the trusted relationship established between two federated domains, the AIM Sandbox and the AIM Live Functions, as an example of a possible implementation of a replica of a live environment in a Sandbox.

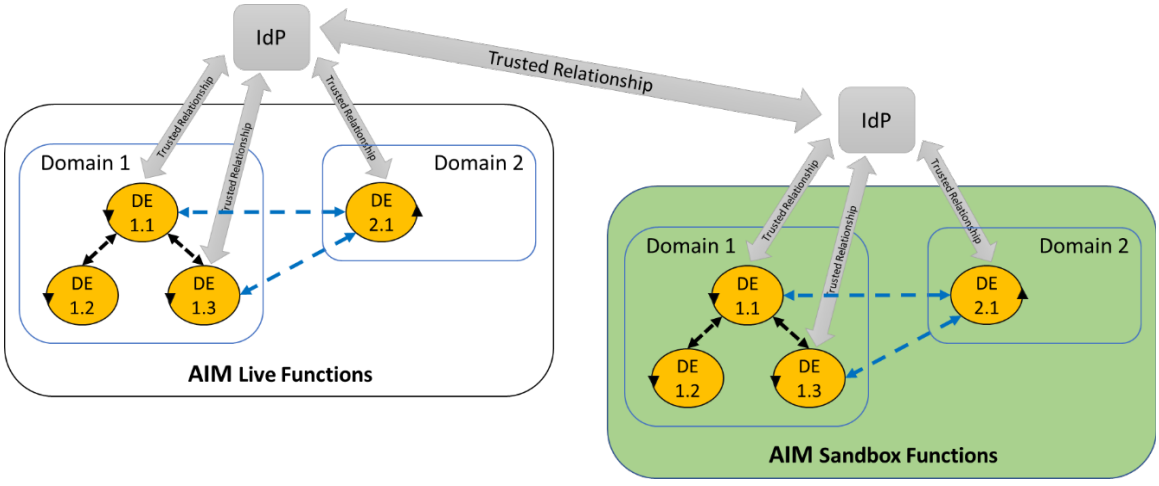

Figure 22: An example AIM Sandbox and Live Subsystems trusted relationship

Figure 23 presents a detail of the logical reference points illustrated in Figure 4 of Section 6.2.

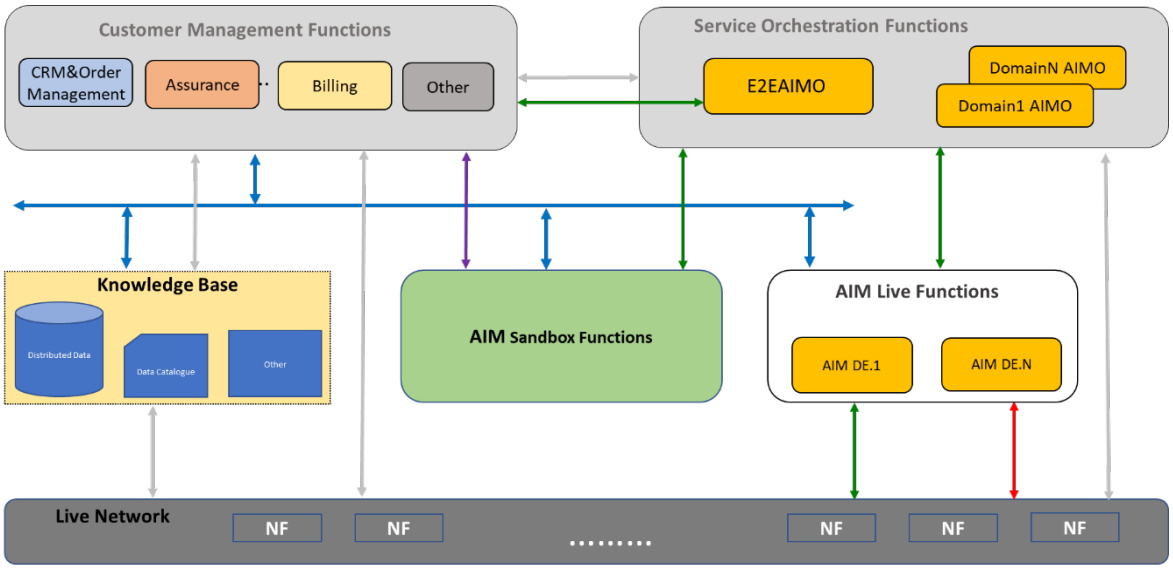

Figure 23: Sandbox Logical subsystem Federation

The “common message bus” (blue lines) is used to access and share information while:

R-122 The interface between the AIM Sandbox Functions to the E2EAIMO/DomainAIMO Orchestration Functions (green line) MUST be used to orchestrate and manage functionalities of AIM Sandbox.

R-123 In particular the “Od-aimo-De” reference point described in Section 6.8 SHOULD be used to:

- Deploy models from the AIM Sandbox to the AIM Live Pipeline subsystem, i.e., in the live operational environment, according to the specific AIM DEs lifecycle management
- Monitor the effects of AIM DEs on network/service operations
- Manage feedbacks regarding the model performances, when the model performance falls below a predefined threshold.

R-124 The interface between the AIM Sandbox Functions and the Customer Management Functions (purple line) SHOULD be used for the management of the functionalities implemented by the AIM Sandbox Logical subsystem like model selection, design, update and lifecycle management.

This interface implements the business logics adopted by an Operator, possibly developing its own AIM Sandbox solution or deploying a 3rd party AIM Sandbox. The standardization of this reference point is out of scope for this Technical Report although.

This AIM Sandbox provides developers with the environment for editing, composing, integrating, packaging, training, and ultimately deploying AI-based microservices.

Appendix I. Reference concepts from other frameworks

I.1 GANA Loop Automation hierarchy

The support of hierarchical OLA/CLA is a capability required to the AIM framework. The four basic hierarchical control loops levels and associated Decision Elements (DEs) defined in ETSI GANA [4] are a good reference for application to CloudCO Managed Entities:

Table 5: GANA framework in a nutshell: hierarchical levels, DEs and intelligence degrees

Control-loop levels	Decision Element description	Degree of intelligence
Protocol-Level	Level-1 DE: any Managed Entity (ME) that may exhibit intrinsic control-loops and associated inbuilt DE Logic	Zero to much uncomplex Cognitive Algorithms for AI
Function-Level	Level-2 DE: a DE for collective autonomic management and control of "networking function" or a "management/control function"	Uncomplex Cognitive Algorithms for AI
Node-Level	Level-3 DE: a DE for autonomic management and control of the node as a whole, as well as the orchestration and policing of the "Function Level-DE"	Moderately complex Cognitive Algorithms for AI (e.g., Machine Learning, Deep Learning
Network-Level	Level-4 DE: a DE for network-wide autonomic management and control of lower level DEs and the MEs.	Complex Cognitive Algorithms for AI (e.g., Machine Learning, Deep Learning

I.2 Machine Learning Pipeline Components

Any DE within the AIM framework may include functions performing ML for AI analysis. A function performing ML can be decomposed into an ML pipeline, wherein the ML pipeline consists of a number of constituent components. This is similar to decomposing a Virtual Network Function (VNF) into a number of constituent VNF Components (VNFCs).

This section reports the definitions of Machine Learning components of provided in ITU-T Y.3172 [17] and ETSI ENI [6], which are functionally quite similar.

For unification purposes, this Technical Report, has been adopted the ITU-T Y.3172 [17] nomenclature.

ITU-T Y.3172 [17] functional definitions have inspired the specifications in this Technical Report but they shall not be construed as constrained by the ITU-T Y.3172 [17] and ETSI ENI [6] framework.

Instead the AIM framework specified in this Technical Report, goes beyond them to best fit into Broadband Forum's portfolio of specification for network transformation towards Cloud, SDN, and NFV based architectures.

I.2.1 ETSI ENI System Functional Blocks

ETSI Experiential Networked Intelligence (ENI) [6] defines the following system functional blocks:

- Input processing
  - Data ingestion
  - Normalization

  Data typically comes from different sources, is created using different applications and programming languages, and typically is ingested using different protocols. It is possible to combine the Data ingestion and Normalization functional blocks into a single functional block.
- Analysis

    - Knowledge management and processing
    - Context awareness
    - Cognition management

  The analysis functional block may represent cognitive systems beyond ML.
- Policy generation
    - Situation awareness
    - Model driven engineering
    - Policy management

  Situation awareness can provide focus with high information flows. Model driven engineering may further analyses with reusable models of concepts and interactions between those concepts. Policies may manage goals, recommendations, or commands to the assisted system.
- Output generation
    - Denormalization
    - Output generation

  It is possible to combine Denormalization and Output Generation into a single Functional Block.

I.2.2 ITU-T Y.3172 ML Pipeline Nodes

ITU-Y Y.3172 [17] defines the following nodes of an ML pipeline:

- SRC (source): This node is the source of data that can be used as input to the ML pipeline. There can be multiple SRC nodes.
- C (collector): This node is responsible for collecting data from one or more SRC nodes. A collector node may have the capability to configure SRC nodes; such configurations may be used to control the nature of data, its granularity and periodicity.
- PP (pre-processor): This node is responsible for cleaning data, aggregating data or performing any other pre-processing needed for the data to be in a suitable form so that the ML model can consume it.
- M (model): This is a machine learning model, in a form which is usable in a machine learning pipeline.
- P (policy): This node enables the application of policies to the output of the model node. Specific rules can be put in place by a network operator to safeguard the sanity of the network, e.g., major upgrades may be done only at night time or when data traffic in the network is low.
- D (distributor): This node is responsible for identifying the SINK(s) and distributing the output of the model node to the corresponding SINK nodes. It may have the capability to configure SINK nodes.
- SINK: This node is the target of the ML output on which it takes action. There can be multiple SINK nodes.

End of Broadband Forum Technical Report TR-436

