broadband forum

Technical Report

TR-486

Interfaces for AIM

Issue: 1

Issue Date: December 2023

Notice

The Broadband Forum is a non-profit corporation organized to create guidelines for broadband network system development and deployment. This Technical Report has been approved by members of the Forum. This Technical Report is subject to change. This Technical Report is owned and copyrighted by the Broadband Forum, and all rights are reserved. Portions of this Technical Report may be owned and/or copyrighted by Broadband Forum members.

Intellectual Property

Recipients of this Technical Report 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 or use of any software code normatively referenced in this Technical Report, and to provide supporting documentation.

Terms of Use

1. License

Broadband Forum hereby grants you the right, without charge, on a perpetual, non-exclusive and worldwide basis, to utilize the Technical Report for the purpose of developing, making, having made, using, marketing, importing, offering to sell or license, and selling or licensing, and to otherwise distribute, products complying with the Technical Report, in all cases subject to the conditions set forth in this notice and any relevant patent and other intellectual property rights of third parties (which may include members of Broadband Forum). This license grant does not include the right to sublicense, modify, or create derivative works based upon the Technical Report except to the extent this Technical Report includes text implementable in computer code, in which case your right under this License to create and modify derivative works is limited to modifying and creating derivative works of such code. For the avoidance of doubt, except as qualified by the preceding sentence, products implementing this Technical Report are not deemed to be derivative works of the Technical Report.

2. NO WARRANTIES

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 BROADBAND 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.

3. THIRD PARTY RIGHTS

Without limiting the generality of Section 2 above, BROADBAND FORUM ASSUMES NO RESPONSIBILITY TO COMPILE, CONFIRM, UPDATE OR MAKE PUBLIC ANY THIRD PARTY ASSERTIONS OF PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS THAT MIGHT NOW OR IN THE FUTURE BE INFRINGED BY AN IMPLEMENTATION OF THE TECHNICAL REPORT IN ITS CURRENT, OR IN ANY FUTURE FORM. IF ANY SUCH RIGHTS ARE DESCRIBED ON THE TECHNICAL REPORT, BROADBAND FORUM TAKES NO POSITION AS TO THE VALIDITY OR INVALIDITY OF SUCH ASSERTIONS OR THAT ALL SUCH ASSERTIONS THAT HAVE OR MAY BE MADE ARE SO LISTED.

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

Issue History

Issue Number	Issue Date	Issue Editor	Changes
1	December 2023	Ken Kerpez, DZS Mauro Tilocca, TIM	Original

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

Editor:	Ken Kerpez, DZS Mauro Tilocca, TIM Antonio Marsico, Reply
Work Area Director(s):	Bruno Cornaglia, Vodafone Mengmeng Li, China Mobile
Project Stream Leader(s):	Ken Kerpez, DZS

Table of Contents

Table of Figures

Table of Tables

Executive Summary

This Technical Report specifies the Automated Intelligent Management (AIM) Interfaces, interfaces functions and information models that allow interactions between the AIM components and subsystems. This Technical Report follows TR-436 [9], which defined the AIM framework for access and home networks management with automation and intelligence.

This Technical Report specifies the services of AIM components and links them to the required information models. This Technical Report addresses both NETCONF/YANG interfaces (as defined in the referenced BBF TRs) and several other protocols in order to cover traditional FCAPS functions as well as management functions for the orchestration of AIM Pipelines and integration among AIM logical subsystems.

1. Purpose and Scope

1.1. Purpose

The TR-436 [9] defines the Automated Intelligent Management (AIM) framework for access and home networks management with automation and intelligence.

The purpose of this Technical Report is to specify the AIM Interfaces, Interfaces Functions and Information Models (IM) that allow the interactions between the AIM components and subsystems.

This Technical Report is an interface specification, following the Service Based Architecture (SBA) principles, that defines what services and DMs are exposed and what protocol requirements are supported by the specific AIM component. The exposed services and DMs depend on the set of features and/or capabilities implemented by each AIM function.

This Technical Report adopts specifications, e.g., other BBF TRs or standards published by other Standards Development Organizations that define features and/or capabilities implemented by Physical Network Function (PNF) or a Virtual Network Function (VNF).

1.2. Scope

This Technical Report specifies the services of AIM components and links them to the required Information Models.

This Technical Report addresses both NETCONF/YANG interfaces (defined in the referenced BBF TRs) and several other protocols in order to cover traditional FCAPS functions but also management functions for the orchestration of AIM Pipelines and integration among AIM logical subsystems.

This Technical Report refers, as deemed necessary, to the YANG DMs and NETCONF requirements already published, it will provide input for modifications in existing DM, and it will address the creation of new DMs and will specify the transport protocol and encoding mechanism for each AIM interface.

The protocols and DMs that are still in development by the BBF, and other Standards Development Organizations are expected to be integrated in future issues of this Technical Report as they become mature.

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 in RFC 2119 [11].

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

2.2.1. Published 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-069	CPE WAN Management Protocol	BBF	2020
[2]	TR-131i1a1	ACS Northbound Interface Requirements	BBF	2015
[3]	TR-181	Device Data Model for CWMP Endpoints and USP Agents	BBF	2022
[4]	TR-232	Bulk Data Collection	BBF	2012

[5]	TR-369	User Service Platform (USP)	BBF	2022
[6]	TR-384	Cloud Central Office Reference Architectural Framework	BBF	2018
[7]	TR-411	Definition of interfaces between CloudCO Functional Modules	BBF	2021
[8]	TR-435	NETCONF Requirements for Access Nodes and Broadband Access Abstraction	BBF	2020
[9]	TR-436	Access & Home Network O&M Automation/Intelligence	BBF	2021
[10]	TR-451	vOMCI Interface Specification	BBF	2022
[11]	RFC 2119	Keywords for use in RFCs to Indicate Requirement Levels	IETF	1997
[12]	GS NFV-MAN 001	Network Functions Virtualization (NFV); Management and Orchestration	ETSI	2014
[13]	RFC 8040	RESTCONF Protocol	IETF	2017
[14]	RFC 7950	The YANG 1.1 Data Modeling Language	IETF	2016
[15]	SOL001	Network Functions Virtualization, Protocols and Data Models; NFV descriptors based on TOSCA specification	ETSI	2020
[16]	SOL004	Network Functions Virtualization (NFV) Release 3; Protocols and Data Models; VNF Package and PNFD Archive specification	ETSI	2021
[17]	SOL006	Network Functions Virtualization (NFV) Release 3; Protocols and Data Models; NFV descriptors based on YANG Specification	ETSI	2021
[18]	SOL007	Network Functions Virtualization (NFV) Release 3; Protocols and Data Models; Network Service Descriptor File Structure Specification	ETSI	2021
[19]	GB1028	Using TM Forum APIs for End-to-End Performance Management v1.0.0	TMF	2021
[20]	IG1219	AI Closed Loop Automation – Anomaly Detection and Resolution	TMF	2021
[21]	TMF638	Service Inventory API User Guide	TMF	2020
[22]	TMF639	Resource Inventory API User Guide	TMF	2020
[23]	TMF633	Service Catalog API User Guide	TMF	2021
[24]	TMF641	Service Ordering API User Guide	TMF	2021
[25]	TMF657	Service Quality Management API User Guide	TMF	2020
[26]	TMF649	Performance Threshold API REST Specification	TMF	2017
[27]	TMF623	SLA Management API REST Specification	TMF	2015
[28]	TMF628	Performance Management API REST Specification	TMF	2015
[29]	TMF642	Alarm Management API User Guide	TMF	2020
[30]	TMF688	Event Management API User Guide	TMF	2021
[31]	GB922	Information Framework Models Suite	TMF	2022

2.2.2. 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
[32] WT-413 issue 2	SDN Management and Control Interfaces for CloudCO Network Functions	BBF	
[33] WT-477	Cloud CO Enhancement - Access Node Hardware Disaggregation	BBF	

2.3. Definitions

The following terminology is used throughout this Technical Report.

ACS
Auto-Configuration Server. This is a component in the broadband network responsible for auto-configuration of the CPE for advanced services.

ATSC
Advanced Television Systems Committee. Digital television standards primarily adopted in North America.

CPE
Customer Premises Equipment.

Cross-domain SDN Orchestrator
The orchestrating element Northbound of the Domain SDN-C systems.

In a Cloud CO context, the Cross-domain SDN Orchestrator is the CloudCO Domain Orchestrator (CCO DO) or a similar function if the deployment differs from a Cloud CO architecture.

WLAN
Wireless Local Area Network.

2.4. Abbreviations

This Technical Report uses the following abbreviations:

AI	Artificial Intelligence
AIM	Automated Intelligent Management
AIMO	AIM Orchestrator
BAA	Broadband Access Abstraction
CCO DO	CloudCO Domain Orchestrator
CF	Collection Function
CFS	Customer Facing Service
CLA	Closed Loop Automation

CNF	Containerized/Cloud Network Function
CRUD	Create Read Update Delete
CRUD-N	Create Read Update Delete Notify
CT	Collection Task
DE	Decision Element
DF	Distributor Function
DM	Data Model
DMT	Decision-Making Task
E2E	end-to-end/End-to-End
E2E AIMO	End-to-End AIM Orchestrator
IM	Information Model
ME	Managed Entity
MF	Model Function
MIB	Management Information Base
ML	Machine Learning
NFVI	Network Function Virtualization Infrastructure
NFVO	Network Function Virtualization Orchestrator
PF	Policy Function
PPF	Pre-Processing Function
PPT	Pre-Processing Task
RBAC	Rule Based Access Control
RFS	Resource Facing Service
SBA	Service Based Architecture
SBI	Southbound Interface
SDN-C	Software Defined Network Controller
SLA	Service Level Agreement
SLI	Service Level Indicator
SLO	Service Level Objective
SNMP	Simple Network Management Protocol
SRC	Source
TR	Technical Report
USP	User Service Platform
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function
VNFM	Virtual Network Function Manager
WA	Work Area
WT	Working Text
YANG	Yet Another Next Generation

3. Technical Report Impact

3.1. Energy Efficiency

The introduction of the cloud based infrastructure and resources, as described in this Technical Report and TR 436 [9], to empower the existing network and services with AI-based capabilities accounts for additional power consumption, as a general assessment.

These potential impacts shall be assessed against the advantages that those AI-based capabilities bring. Notably, AI can be used to improve monitoring and optimization of power consumption and a wealth of applications towards improving Energy Efficiency and Sustainability of broadband networks and services. In general, AI enables better analysis of network and equipment state information and makes predictions that human operators or non-AI based solutions are not capable of or not with the same amount of input data, velocity, granularity, or insight results. As an example, this may result in less truck rolls, and save energy this way up to sophisticated site power consumption management and predictions.

3.2. Security

TR-486 has no impact on security.

3.3. Privacy

TR-486 has no impact on privacy.

4. Introduction

TR-436 [9] provides an architectural framework for the integration of AIM functionalities into SDN networks.

The AIM framework supports the Logical Reference Points defined between AIM Logical Subsystems (briefly described in Section 5), and the AIM Interfaces (Section 6 provides a high level description, while Section 7 contains a detailed specification) that allow the interactions between the AIM components and subsystems.

Figure 1 and Figure 2 describe the AIM interfaces (black lines) in scope of this Technical Report. Figure 1 exemplifies the case of a hierarchy of E2E and Domain-specific Pipelines for multiple domains, and Figure 2 exemplifies a Domain-specific Pipeline scope for a single domain.

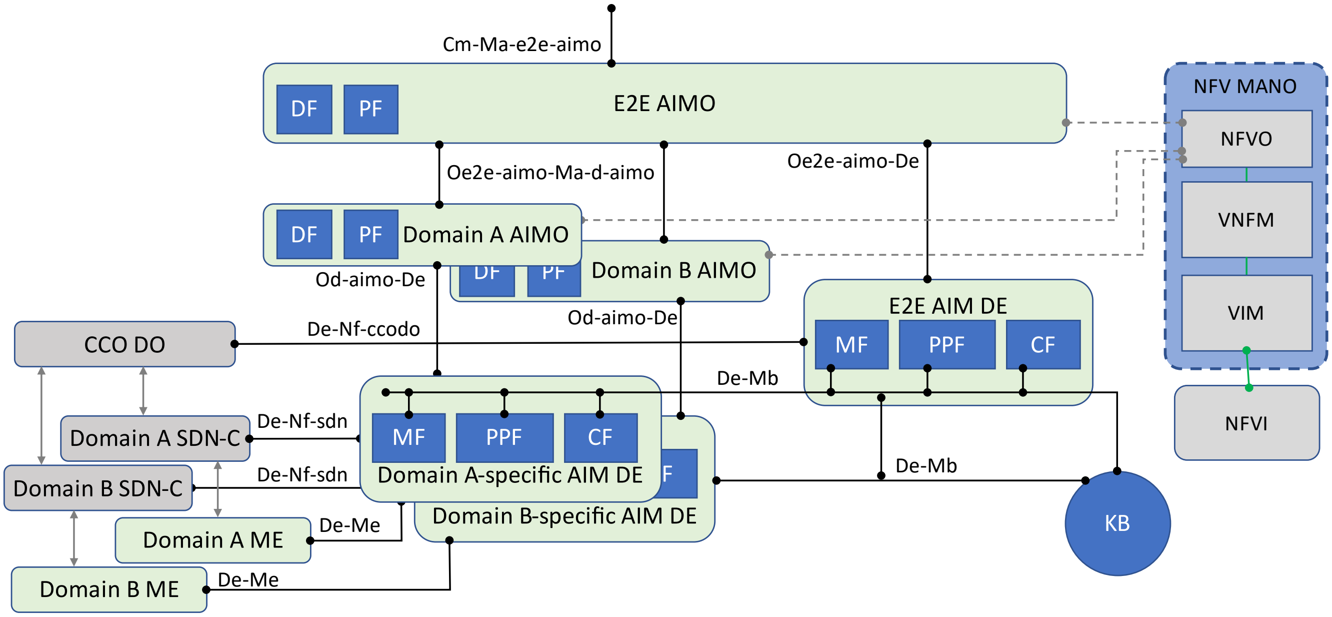

Figure 1: AIM Interfaces – E2E and Domain-specific AIM DEs hierarchy

Figure 1 shows Domain A and Domain B, each domain having a domain-specific SDN-C, a domain-specific AIMO, and domain-specific AIM DE. Each domain-specific AIM DE connects to a domain-specific ME. The E2E AIM DE performs decision analyses and configuration recommendations across the multiple domains. The E2E AIMO coordinates orchestration functions across the domains.

As shown in Figure 1, the interaction and data exchange across a hierarchy of AIM Pipelines is via the common message bus and the related De-Mb interface. This also underlines that E2E AIM Pipelines are typically fed by lower-level Domain-specific Pipelines. This can be more effective than having the Collection Function of an E2E AIM Pipeline consuming data directly from MEs via the De-Me interface which is optional (dotted line) for E2E AIM DE.

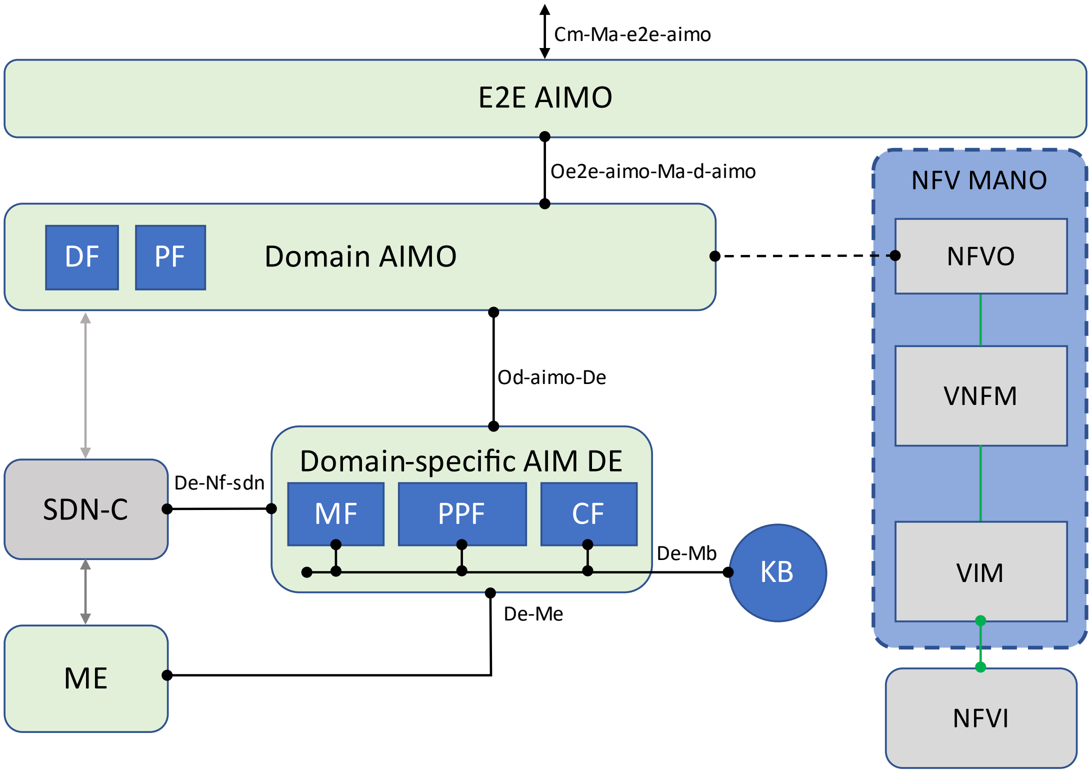

Figure 2: AIM Interfaces – Domain-specific AIM DEs scope

The Domain AIMO orchestrates AIM logical subsystems and the E2E AIM Orchestrator (E2E AIMO) provides orchestration across AIM domains.

As shown in Figure 1 and Figure 2, the AIM solution relies on the NFV Infrastructure and ETSI NFV MANO components that enable flexibility and scalability in the lifecycle and consumption of AIM components and pipelines. The SDN Controller (SDN-C) controls the physical infrastructure as well as all associated virtual/physical functions (PNF/VNF). As explained in TR-436 [9], the SDN Controller is the authoritative source for the management and configuration of MEs under its control, applying policies and resolving any conflict that may arise, assuring also one of the important key design benefits of SDN: a single “Source of Truth” for all network configurations within a domain/subdomain.

- For the closed loops confined within the PNF/VNF, the SDN Controller can delegate to the PNF/VNF itself the capability to manage and configure aspects of the MEs. In fact, these loops, like any other functionality confined within the PNF/VNF, must be configured and managed by the SDN Controller. It is indeed fundamental that the PNF/VNF timely communicates to the SDN Controller any self-applied configuration.
- For loops spanning multiple PNFs/VNFs, the SDN Controller continues to be used to manage and configure the MEs. Although delegating local configuration changes appears to deliver a fast-response in specific low-latency use-cases, such as 5G, this scenario has to prevent PNFs/VNFs from being in an out-of-sync configuration status with the SDN controller. Out-of-sync configuration issues between the SDN Controller, which has direct control over the “Source of Truth” for all management and configuration tasks, and the PNF/VNF under control of a loop would thus require complex signaling and resynchronization with the SDN Controller. Having the AIM DE steering Closed Loop recommendations through the SDN Controller avoids this out-of-sync configuration scenario.
- For closed loops whose logic and objectives spans across multiple domains, the overall structure is built as a hierarchy of pipelines (see section 5.2/TR-436 [9]), and typically the Domain-specific pipelines are also sources for a higher-level E2E pipeline. This can be appreciated in Figure 1, where the Domain-specific AIM DE can communicate with the E2E Pipeline via the common message bus.

For an E2E Pipeline the Sources providing data are the typically the lower-level Domain-specific Pipelines, albeit direct collection from the Network Resources MEs may be optionally supported.

The AIM architecture allows the usage of one or multiple Domain SDN Controllers. In this latter case, the Domain AIMO orchestrates domain-scoped Pipelines referring to the SDN-C consistently to its own domain.

Similarly, the E2E AIMO is responsible for the orchestration of end-to-end (E2E) Pipelines, which issue recommendations to the cross-domain SDN Orchestrator, e.g., the CloudCO Domain Orchestrator (CCO DO) in the case of a CloudCO SDN-NFV infrastructure (see TR-384 [6]). An end-to-end Pipeline is functionally identical to a domain-scoped Pipeline with the difference that its horizon, analyses, optimizations and collected inputs span across multiple domains.

In the remainder of this document sometimes terms like AIM Pipeline (or simply Pipeline) and AIM DE (or simply DE) are used in a generic way to refer to a Domain-scoped Pipeline, an End-to-End Pipeline or both.

Under a normative reference standpoint:

- When the term AIMO is used, the reported specifications and Requirements apply both to an E2E AIMO implementation and to a Domain AIMO implementation.
- When the terms AIMO and AIM Pipeline/DE are used in connection the reported specifications and Requirements apply to:
  - An E2E AIMO implementation and counterpart E2E AIM Components (CF, PPF, MF) communicating via the Oe2e-aimo-De interface; a MF implementation and its De-Nf-ccodo interface when operating within an end-to-end Pipeline
  - A Domain AIMO implementation and counterpart Domain-specific AIM Components (CF, PPF, MF) communicating via the Od-aimo-De interface; a MF implementation and its De-Nf-sdn interface when operating within a Domain-specific Pipeline

The AIM Logical Reference points are defined in TR-436 [9] and briefly described in Section 5 of this Technical Report.

5. AIM Subsystems Logical Reference Points

This section presents the complete picture of an AIM framework, as defined in TR-436 [9], with AIM subsystems and related logical reference points.

The AIM framework is composed of the following four logical subsystems and related reference points defined in section 6.2/TR-436 [9] and depicted in Fig.4/TR-436 [9], which is also shown in Figure 3.

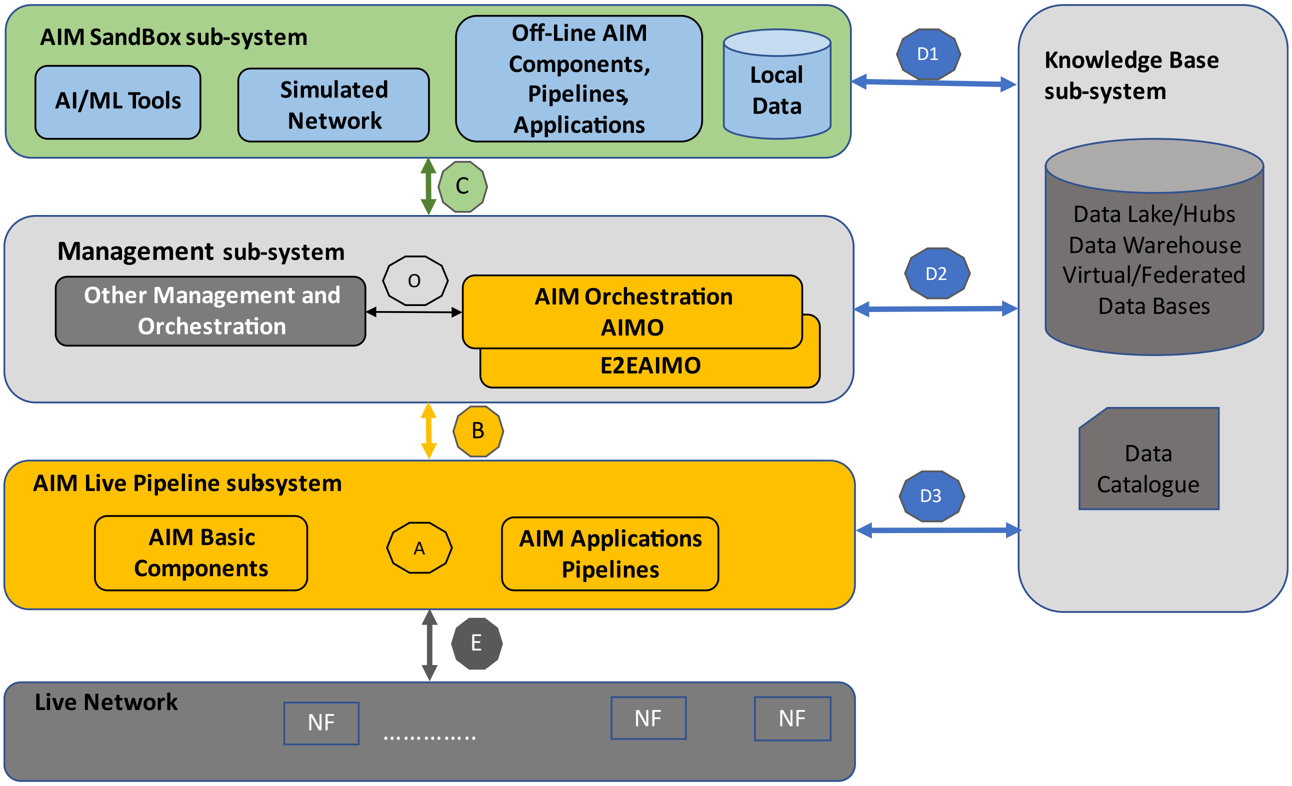

Figure 3: AIM Architecture Logical Subsystems

- **AIM Live Pipeline** subsystem is deployed and running in the operational infrastructures, thanks to the Domain AIMO and/(or) E2E AIMO that oversee the instantiation of the AIM components, set-up of the entire AIM pipelines and run-time monitoring of the AIM applications.
- **AIM Sandbox** subsystem is a dedicated off-network environment that allows the hosting of AIM components and AIM pipelines to train, test, and evaluate them before deploying them into AIM Live Pipeline logical subsystem in the operational environment, and may include AI/ML tools to train, test and evaluate ML models, simulated networks, data from live networks, access to historical data from real networks, and eventually local data.
- **Management** subsystem contains the Domain AIMOs and the E2E AIMO, i.e., the AIM specific management and orchestration functional blocks, that collaborate with other management and orchestration functions (Domain Orchestrators, E2E Orchestrator) of the SDN-NFV infrastructure to manage the AIM Live Pipeline subsystem.
- **Knowledge Base** 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.

The AIM framework supports the logical reference points between the AIM subsystems defined in TR-436 [9] and briefly described in the following sections.

Table 1 reports the mapping of AIM Interfaces to the Logical Reference Points.

Table 1: Mapping of AIM Interfaces to Logical Reference Points

Logical Reference Point	AIM Interfaces
A	De-Mb
B	Oe2e-aimo-De Od-aimo-De-xxx (Note 1)
C	Od-aimo-Sb-*xxx* (Note 1 and 2) Oe2e-aimo-Sb (Note 2)
D1	Sb-Kb (Note 2)
D2	Od-aimo-Kb-*xxx* (Note 1 and 2) Oe2e-aimo-Kb (Note 2)
D3	De-Mb
O	Cm-Ma-e2e-aimo Oe2e-aimo-Ma-d-aimo
E	De-Nf-ccodo De-Nf-sdn De-Me

Note 1: *xxx* connotates the specific domain that this Domain AIMO interface refers to.

Note 2: this interface name was not yet defined at the time of TR-436 [9] publication. Its specification is not in scope of this Technical Report.

5.1. Logical Reference Point A

The logical reference point A is internal to the AIM Live Pipeline subsystem, i.e., between the AIM Live Pipelines components. The logical reference point A is described in Section 6.3 and specified in Section 7.6 and contains the following interface:

- **De-Mb*** - located between Decision Elements, is the interface for the AIM DE and its internal components to exchange information.

* Note: in this section, only the description of the De-Mb interface internal to the AIM Live Pipeline subsystem is reported.

5.2. Logical Reference Point B

The logical reference point B is defined between the Management subsystem (particularly AIMO/E2E AIMO) and the AIM Live Pipelines subsystem regarding instantiation of AIM components, setup of AIM pipelines, AIM application execution monitoring, model performance monitoring, etc.

The logical reference point B is described in Section 6.2.2 and specified in Section 7.3 and contains the following interface:

- **Oe2e-aimo-De**, located between the E2E AIMO and AIM Decision Elements, it is the interface for the AIMO to interact with the end-to-end AIM DE functional blocks to configure, manage, and orchestrate them.
- **Od-aimo-De**, located between the Domain AIMO and AIM Decision Elements, it is the interface for the AIMO to interact with the AIM DE functional blocks to configure, manage, and orchestrate them.

5.3. Logical Reference Point C

The logical reference point C is defined between the Management subsystem (particularly the Domain AIMOs and the E2E AIMO) and the AIM Sandbox regarding transfer of trained, tested, and evaluated models ready to be deployed, and transfer of feedback on performances of the AI/ML models in live network, e.g., when the model performance falls below a predefined threshold.

The interface of Logical Reference Point C is not specified in this Technical Report.

5.4. Logical Reference Point D

The logical reference point D1 is defined between the AIM Sandbox subsystem and the Knowledge Base subsystem to get the data for the training, testing and evaluation of models in the AIM Sandbox environment.

The logical reference point D2 is defined between the Management subsystem (particularly the Domain AIMOs and the E2E AIMO) and the Knowledge Base subsystem to access any data necessary for the management and orchestration purposes.

The interfaces of Logical Reference Points D1 and D2 are not specified in this Technical Report.

The logical reference point D3 is defined between the AIM Live Pipeline subsystem and the Knowledge Base logical subsystem to access any data necessary for the run-time functionalities of the AIM Live pipelines applications.

The logical reference point D3 is described in Section 6.3 and specified in Section 7.6 and contains the following interface.

- **De-Mb***, located between the Decision Element and the Knowledge Base, is the interface for the exchange of information between DEs and the Knowledge Base and consume 3rd party services like data cataloging, etc.

* Note: in this section, only the description of the De-Mb interface between the AIM Live Pipeline subsystem and the Knowledge Base is reported.

5.5. Logical Reference Point O

The logical reference point O is 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.

The logical reference point O is described in sections 6.1 and 6.2 and specified in sections 7.1 and 7.2 and contains the following interfaces:

- **Cm-Ma-e2e-aimo** – located between the Customer Management Layer and the AIM E2E Orchestrator allows exposing an abstracted view of the AIM Domain resources to the Customer Management layer.
- **Oe2e-aimo-Ma-d-aimo –** located between the E2E AIMO and the AIMO, allows exposing an abstracted view of the AIM Domain resources to the E2E AIMO.

5.6. Logical Reference Point E

The logical reference point E is defined between the AIM Live Pipeline subsystem and the Live Network NFs to collect raw data and produce effects on NFs through configuration actions/operations. This shall be mediated by the SDN element that governs the target resource.

The logical reference point E is described in Section 6.3 and specified in sections 7.4 and 7.5 and contains the following interfaces:

- **De-Nf-ccodo** - located between the end-to-end AIM Decision Element and the cross-domain SDN Orchestrator, i.e., the CCO DO, is the interface for the AIM DE to interact with the SDN Managers and Controllers.
- **De-Nf-sdn** - located between the AIM Decision Element and the SDN Controller, is the interface for the AIM DE to interact with the SDN Managers and Controllers. This interface is derived as an extension of the Occo-Nf-sdn-xxx reference points defined by the BBF (Occo-Nf-sdn-access, Occo-Nf-sdn-edge, Occo-Nf-sdn-dc).
- **De-Me –** located between the Decision Element and the Managed Entity, is the interface for the collection function of the AIM DE to collect data from the Managed Entities.

6. AIM Interfaces

This section describes the interfaces between elements of the AIM framework as depicted in Figure 1, notably:

- Cm-Ma-e2e-aimo
- Oe2e-aimo-Ma-d-aimo
- Oe2e-aimo-De
- Od-aimo-De
- De-Me
- De-Nf-ccodo
- De-Nf-sdn
- De-Mb

6.1. E2E AIM Orchestrator

The E2E AIMO is a catalogue-driven orchestrator that delivers, in collaboration with the Domain AIMO(s), end-to-end Pipelines over an SDN-NFV infrastructure. It performs Pipeline decomposition, E2E Pipelines orchestration and the overall orchestration of the AIM domains. It oversees Pipeline hierarchical chaining and lifecycle management.

6.1.1. Component Description

The E2E AIMO is a central function in the architecture; it also exposes the AIM NBI and, in collaboration with the Domain AIMOs, a Pipeline abstraction layer, hiding the internal operations of the AIM from the NBI.

Per R-95/TR-436 [9], the E2E AIMO is implemented as a stand-alone Orchestrator or a function of the E2E Service Orchestrator.

The E2E AIMO spans over multiple AIM Domains, possibly existing in the network. Section 6.10.1 of TR-436 [9], at the top, reports a brief discussion about the rationale of having a single or multiple AIM Domains in an Operator's network.

6.1.2. Component Functionality

Key to the operations of the E2E AIMO are the AIM Pipeline Catalogue and the AIM Resource Inventory. These Catalogue and Inventory are shared with and used by the Domain AIMOs and the NFVO to manage VNFs, PNFs and NFVI resources that serve AIM purposes. The integration between the AIM Resource Inventory with the Service Inventory enables automation of assurance and management processes.

Figures 16 and 17 of TR-436 [9] show three Repositories for VNF Instances, AIM Instances, and NFVI Resource. Throughout this Technical Report, the word Inventory is used instead of Repository because it is largely adopted for these types of information bases.

The AIM Resource Inventory and the AIM Pipeline Catalogue referred to in this document correspond to the Repositories and the Catalogues shown in Figure 16 and Figure 17 of TR-436 [9]. Table 4 of TR-436 [9] synthetically describes them and, without any restriction on the implementation, hints at the following high level requirements for them:

[R-1] The AIM Pipeline Catalogue, as a whole dataset containing elements that are dual in nature MUST support attributes and information that are:

- application-layer related (i.e., AIM Pipeline Components and Pipelines)
- virtualization-layer related (i.e., VNFs and Network Services deployed on the NFVI).

[R-2] The AIM Resource Inventory, as a whole dataset containing elements that are dual in nature, MUST support attributes and information that are:

- application layer related (i.e., AIM Pipeline Components and Pipelines)
- virtualization-layer related (i.e., VNFs and Network Services deployed on the NFVI)

Figures 5 and 6 of TR-436 [9] provide a useful sketch of the overall AIM architecture. Figure 5/TR-436 [9] covers the interactions of the AIM Catalogue and the AIM Inventory with the AIM Orchestrators and other key systems involved with the AIM framework operations.

Notably, the E2E AIMO and the NFVO, for the application-layer and virtualization-layer respectively, are the primary entities that handle the elements in the Inventory and Catalogue;

These datasets can be accessed by other orchestrators and managers involved with AIM workflows on the application-layer thread (e.g., Domain AIMO, Customer Management Layer) and the virtualization-layer operations (e.g., VNFM)

[R-3] The AIM Pipeline Catalogue and AIM Resource Inventory MUST allow access to their datasets by the E2E AIMO, Domain AIMOs, and other systems external to the AIM solution (specifically, the NFVO, Customer Management Layer, VNFM) for the tasks related to the AIM workflows on the application layer thread and the virtualization layer thread.

[R-4] A Rule Based Access Control (RBAC) MUST be supported by the AIM Pipeline Catalogue and AIM Resource Inventory to grant a set of authorized activities by a requesting system.

[R-5] The RBAC mechanism on the AIM Pipeline Catalogue and AIM Resource Inventory MUST be based on a mutual authentication scheme.

[R-6] The Descriptors and Packages for on boarding and instantiating the VNFs and Network Services MUST comply with ETSI NFV ISG specifications, in particular MAN001 [12], SOL001 [15], SOL004 [16], SOL006 [17], and SOL007 [18].

[R-7] The AIM Pipeline Components and Pipelines MUST be defined via Descriptors that contain application-layer parameters.

The DE components metrics related to the xNFs status as NFVI resources (e.g., CPU-load, Memory, Storage, etc.) and related maximum quota are expected to be measured as normal operating procedures and primitives in a MANO (ETSI NFV SOL5 and SOL6 interfaces may be relevant to this) or containerized environment and are out of scope of this document.

The AIM Pipeline Catalogue is used for predefining Pipeline Templates and rules. It enables the centralization of pipeline specification and handling. In turn, the AIM Pipeline Catalogue improves automation of the end-to-end pipeline fulfillment.

The AIM Pipeline Catalogue and the AIM Resource Inventory allow to maintain and expose all the information related to the creation of the AIM pipeline components and of Pipelines. which, in turn, is functional to creating and managing the Pipeline instances so that they are created according to Pipeline specifications

The AIM Resource Inventory facilitates the end-to-end management of the AIM resources and their functions from a pipeline perspective. For example, the AIM Resource Inventory describes the running AIM components, their characteristics (e.g., role, relationships, attributes, constraints), their connectivity under an application layer perspective, etc. The AIM Resource Inventory is integrated with the AIM Pipeline Catalogue that describes what and how AIM resources are chained to form a Pipeline. In this regard, both the AIM Pipeline Catalogue and the AIM Resource Inventory allow describing a Pipeline Service and related ancillary classes in a similar way to TMF SID models which generically and broadly address the specification of any set of Customer Facing Services (CFS) and Resource Facing Services (RFS) that Service Providers need.

Pipeline definitions are modelled based on an E2E orchestration approach, where the E2E AIMO translates the CML intents into requests for domain-specific Pipelines managed by the corresponding Domain AIMO. Depending on the request from the CLM, the E2E AIMO provides the stitching across domains' Pipelines into an end-to-end Pipeline as well as oversees the whole operation of AIM capabilities in the network.

This approach enables the modelling of each AIM domain independently and allows each Domain AIMO to expose resource definitions specific to the resources' capabilities under its control (RFS according to the TMF SID model [31]).

At the E2E AIMO layer, the AIM Pipeline Catalogue contains the definition of the Pipelines' capabilities that support each specific Pipeline (CFS according to the TMF SID model).

[R-8] The E2E AIMO MUST expose, via its NBI, the CFS(s) corresponding to the tasks and services supported by the Pipelines of the AIM solution.

[R-9] The CFS(s) exposed on the E2E AIMO's NBI are defined according to TMF SID model [30].

[R-10] The Domain AIMO MUST expose, via its NBI, the RFS(s) corresponding to the tasks and services supported by the domain specific Pipelines.

[R-11] The RFS(s) exposed on the Domain AIMO's NBI are defined according to TMF SID model [30].

The Customer Management Layer will trigger the Pipeline (CFS) through the Cm-Ma-e2e-aimo reference point illustrated in Figure 4, and the E2E AIMO will use the Pipeline templates from the AIM Pipeline Catalogue to request and orchestrate end-to-end Pipeline across technology domains (federated orchestration).

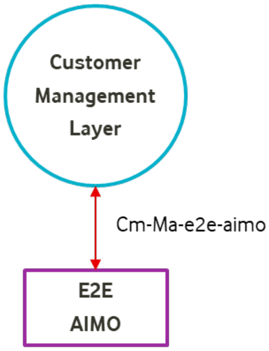

Figure 4: Customer Management Layer to E2E AIMO Reference Point

[R-12] The E2E and Domain AIMOs MUST update the AIM Resource Inventory in a timely fashion as resources are added, removed, assigned and de-assigned and as Pipelines are created, scaled, modified and remove.

[R-13] The E2E and Domain AIMOs MUST manage the lifecycle workflows for the Pipelines they are respectively responsible for, and they MUST check and reserve AIM Pipeline Components and resources before applying any change. They MUST also update the AIM Resource Inventory accordingly.

[R-14] The E2E and Domain AIMOs MUST track the assignments of resources to Pipelines, which can leverage the alarms and notifications received from Pipelines, and through the information. The stored in the AIM Resource Inventory also tracks. This is so that resource exhaustion, criticalities and failures can be quickly identified and solved. to enable policy-driven scaling and healing of Pipelines triggered by events and based on an accurate picture of current resource levels.

The above requirements allow to maintain an accurate, up-to-date inventory of all Pipeline and resource instances and to make prompt decisions as to how and where to deploy new resources.

Figure 8 illustrates the Oe2e-aimo-De reference point between the E2E AIMO and the E2E AIM DE used to interact with the AIM functional blocks (CF, PPF, and MF components) to configure, manage and orchestrate.

The AIM Resource Inventory is updated as resources are added, removed, assigned and de-assigned and as Pipelines are created, scaled, modified and removed.

Pipeline lifecycle management workflows, governed by the E2E and Domain AIMOs, check and reserve AIM resources before applying any change. The AIM Resource Inventory enables policy-driven scaling and healing of Pipelines driven by events and based on an accurate picture of current resource levels.

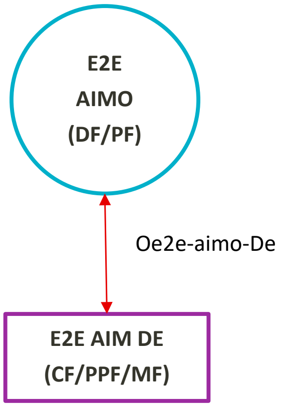

Figure 5: E2E AIMO to E2E AIM DE Reference Point

In its role of orchestrator of end-to-end Pipelines, the E2E AIMO is subject to requirements similar to those of a Domain AIMO but scoped across multiple AIM domains. The reader is invited to look at and interpret section 6.2 under this standpoint.

Here are the end-to-end Pipelines orchestration requirements applicable to the E2E AIMO.

[R-15] The E2E AIMO SHOULD implement an algorithm that optimizes the selection of the AIM components for building a Pipeline from a given Pipeline template. To allocate the resources, either to instantiate new AIM components and/or to reconfigure existing components, the algorithm SHOULD consider the computation capabilities and load status of the AIM resources as derived from usage data.

The E2E AIMO performs its role by operating or orchestrating the Domain resources through the functionalities defined by the DF, PF and the NFVO.

[R-16] The E2E AIMO MUST oversee the creation and orchestration of end-to-end Pipelines and the management of the Pipeline Templates exposed via the Pipeline Catalogue.

[R-17] The E2E AIMO MUST create a Pipeline Service (also named simply a Pipeline) upon checking the availability and usage of already instantiated AIM resources from the AIM Inventory and planning of the sequence of AIM Pipeline components

[R-18] The E2E AIMO MUST identify the Pipeline Template suitable for fulfilling the Pipeline Service request from the CML and then enrich the Pipeline Template with the information necessary to create the actual Pipeline instance.

[R-19] The E2E AIMO MUST send the appropriate requests to the NFVO (or alternative virtualized infrastructure orchestrator, e.g., a Kubernetes Controller) for the instantiation of the VNFs/CNFs instances corresponding to AIM Pipeline Components that need to be created, scaled modified and removed. The E2E AIMO MUST create the Pipeline Service by provisioning communication paths through its AIM components by leveraging message forwarding functions provided by the common message bus.

[R-20] The E2E AIMO MUST monitor virtual resources' usage levels via interaction with the Virtual Infrastructure Manager (VIM) (or alternative virtualized infrastructure manager) relevant for a given Domain.

[R-21] The Pipeline Templates contained in the AIM Catalogue MUST be provided with a set of parameters that includes:

- Deployment details of the AIM Components (e.g., its IDs from the Inventory, topology, relationships, instantiation and configuration settings, monitoring, maintenance)
- Dependencies among AIM components (if any)
- Security and data traffic requirements (e.g., encryption and authentication, latency and/or minimum bandwidth of inter-VNF connection etc.)
- Waypoints of the SBA topology
- Configuration and installation scripts

[R-22] The E2E AIMO MUST translate the declarative Intent received from the CML into end-to-end Pipeline intents and/or Domain-specific Pipeline intents.

The E2E AIMO operates its tasks via a set of standard SBIs towards the AIM DEs.

[R-23] The E2E AIMO MUST implement the DF and PF AIM Components.

[R-24] The E2E AIMO with its DF and PF functionalities MAY be implemented as a stand-alone Orchestrator or as a function of the corresponding the cross-domain SDN Orchestrator.

[R-25] When the E2E AIMO is implemented as a stand-alone Orchestrator, it MUST also implement the essential functionalities and interfaces needed to interact with the NFVO.

[R-26] When the E2E AIMO is implemented as a function of the cross-domain SDN Orchestrator, it SHOULD be responsibility of the cross-domain SDN Orchestrator to implement the functionalities and interfaces needed to interact with the NFVO.

TR-436 [9] describes the AIMO functions. The following sections focus on the DF and the PF functions as centralized functionalities generally interacting with all AIM DEs Components of instantiated Pipelines. The bullet list below describes the main capabilities of the E2E AIMO and the DF and PF integrated in it.

[R-27] Upon receipt of a Pipeline Service Intent and Pipeline Resources requests from the CML, the E2E AIMO MUST

- Retrieve the description of the Pipeline Service Intent and Pipeline Resources Intent from the AIM Pipeline Catalogue
- Verify whether the resources indicated in the Pipeline Resources Intent are already available from the Pipeline Resource Inventory. If not available, the E2E AIMO requests the instantiation of the necessary resources to the NFVO

[R-28] Based on the pipeline description from the Pipeline Service Intent, the E2E AIMO MUST request the following:

To the Distributor Function:

- to populate the Input and Output Endpoint values (e.g., topics taken from a pool the DF has access to, etc.) to be configured on AIM DE components to realize the Pipeline topology

To the Policy Function:

- to provide the access policies used by AIM components for communicating over the message bus (e.g., authentication of DE components and read/write access grants to the Message Bus)
- to define the policies to enforce control and management actions on AIM components (e.g., reconfiguration, scaling, creation of new instances, etc.)
- to define the policies under which a Pipeline is authorized to perform certain actions and the constraints associated with those actions (e.g., issuing of output recommendations to SDN-C based on a precision or recall threshold, etc.)

[R-29] To finalize the Pipeline Service Intent creation, the E2E AIMO MUST:

- Configure the AIM DE components per the Pipeline Service Intent descriptor (see Sections 6.3.2.1 and 6.3.2.2)
- Configure the AIM DE components during runtime
- Activate the pipeline per the planned schedule (e.g., immediate, on scheduled time, on event trigger)
- Pause/stop a pipeline per internal logics and/or external inputs
- Monitor pipeline's indicators (e.g., accuracy/convergence, health) and performs corrective actions
- Deactivate the pipeline (e.g., immediate, on scheduled time, on event trigger, admin trigger)

The E2E AIMO performs the stitching of Pipeline Components through its Distributor Function (DF). The DF enables the stitching of the Pipeline, which is built by using its specific Template and from the AIM Pipeline Catalogue. The Pipeline Template defines the chaining logic as a workflow of functional AIM components that corresponds to the VNF graph defined in GS NFV-MAN 001 [12] and includes the AIM components' types and template definitions to create the Pipeline (e.g., the parameters needed, their order and the connectivity requirements).

[R-30] The DF MUST configure and oversee the distribution of Components' outputs across Pipeline Components and ultimately the distribution of the output of the MF Component to the corresponding SINK nodes.

[R-31] The DF MUST identify the Output and Input Endpoints that fulfill the stitching per the Pipeline template and instantiation requested by the Domain AIMO. For example, if the communication among AIM components is implemented via a Kafka bus, the DF assigns Endpoints to Kafka topics per the appropriate pipeline topology.

6.1.2.1. Distributor Function

Whether a Distributor Function is implemented in a E2E AIMO or a Domain AIMO, its functionalities and requirements are the same.

See section 6.2.2.1 and the subsections therein.

6.1.2.2. Policy Function

Whether a Policy Function is implemented in a E2E AIMO or a Domain AIMO, its functionalities and requirements are the same.

See section 6.2.2.2 and the subsections therein.

6.1.3. Intent Base

This section presents the intents used to send a Pipeline Intent from the Tenant, i.e., the Customer Management Layer systems (e.g., an assurance platform, a billing system, a power consumption control system) in the form of an AIM CFS Intent, actualized with attributes values that fit the instantiation of that very AIM CFS request.

[R-32] Upon receipt of a Pipeline Intent request, the E2E AIMO MUST:

- Retrieve the description of the PipelineIntent from the Pipeline Catalogue
- Break down the high level PipelineIntent into the PipelineIntent for each AIM Domain involved in the Pipeline

- Map each Domain PipelineIntent to its corresponding RFS Intents. Verify if the RFS(s) are already available from the AIM Resource Inventory.
- Send to the Domain AIMO(s) the appropriate Pipeline Intent(s) to request the creation and configuration of a Pipeline (Pipeline Service Intent, refer to section 6.2.3.2) and/or of new Pipeline Resources (Pipeline Resource Intent, refer to Section 6.2.3.1).

6.1.3.1. Pipeline Intent Information Model

Figure 6 depicts the information model of a generic Pipeline Intent.

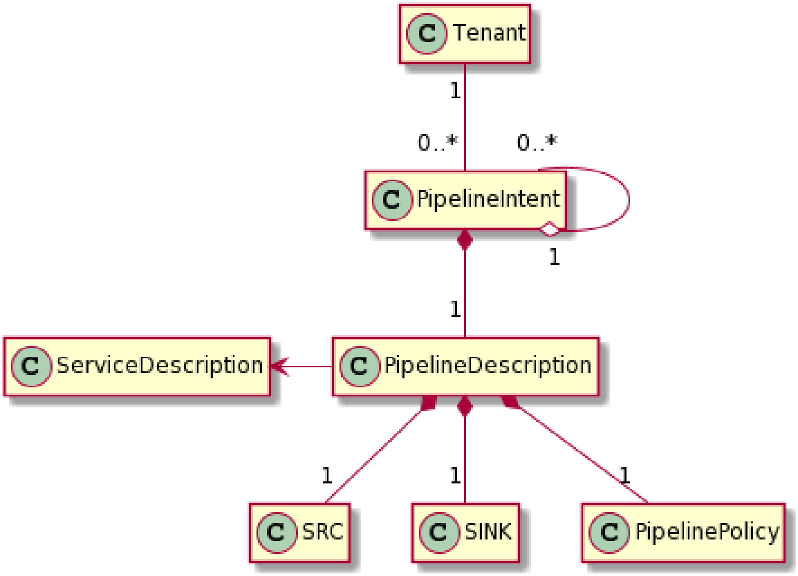

Figure 6: Pipeline Intent Information Model

[R-33] The Pipeline Intent MUST comply with the diagram in Figure 6.

The PipelineIntent can aggregate multiple PipelineIntents to accommodate simple as well as recursive creation of hierarchical Pipelines either within a single or multi-AIM domain. In fact, each PipelineIntent will be composed of a PipelineDescription which is composed of the SRC and SINK.

Additionally, the PipelineDescription depends on the ServiceDescription and is composed of the PipelinePolicy (refer to section 6.2.2.2.1) classes that connotate the requested Pipeline with service and policy information passed to the target Domain AIMO to allow the instantiation of one or more pipelines that fulfil the needs of the CFS requested by the Customer Management Layer.

[R-34] The Cm-Ma-e2e-aimo MUST expose the information elements, operations and data models described in Table 2:

Table 2: Pipeline Intent Information Model Class Description

Class Name	Description
Tenant	The owner of the Pipeline to be created
PipelineIntent	The intent used to create the Pipeline
PipelineDescription	The description of the hierarchical structure of the Pipeline (e.g., IEs that describe the Pipelines from the AIM Pipeline Catalogue of multiple domains)
ServiceDescription	The description of the generic service (e.g., an End User Service, a power consumption service, etc.) the Pipeline is attached to (e.g., the SLAs/SLOs and resources that deliver the service from the service Inventory)
SRC	The source of data that can be used as input to the Pipeline
SINK	This is the target of the output on which the Pipeline takes action
PipelinePolicy	The policy attached to a Pipeline (e.g., service policy and component policy)

The intent base used by the E2E AIMO and the Domain AIMO for orchestrating the creation of the resources for a Pipeline and its configuration are described in section 6.2.3.

6.1.4. Management Capabilities

The E2E AIMO MUST support the following AIM Components under its orchestration scope are categorized below:

[R-35] AIM Pipeline Service exposure capabilities:

- Exposure of AIM Pipeline Services to the Customer Management Layer systems via the Cm Ma e2e aimo interface

[R-36] The E2E AIMO MUST support the following AIM Pipeline Catalogue capabilities:

- Lifecycle Management of the Pipeline Templates (shared with the Domain AIMOs)
- Exposure of the Catalogue Resources
- Management of Catalogue Notifications

[R-37] The E2E AIMO MUST support the following AIM Pipeline Inventories Service exposure capabilities:

- Inventory of the Pipeline Intents (shared with Domain AIMOs)
- Inventory of Pipeline Resources, e.g.,. AIM components and related characteristics (shared with Domain AIMOs)
- Logical Topology and Connections

[R-38] The E2E AIMO MUST support the following AIM Domain Control capabilities:

- End-to-End Pipeline Intent Fulfillment
- End-to-End Pipeline Intent Check
- End-to-End Pipeline Intent Realization
- End-to-End Pipeline Policy Management

[R-39] The AIMO MUST support the following AIM Domains Monitoring capabilities:

- End-to-end Pipeline Intent Performance and Events
- End-to-end Pipeline Intent Fault and Security Events

6.2. Domain AIM Orchestrator

The Domain AIMO is a catalogue-driven orchestrator that delivers, across a specific technology domain, Pipelines over a SDN-NFV infrastructure. It performs the creation and lifecycle management of the Pipeline Resources and the Pipeline themselves and reports status and performance information to the E2E AIMO.

6.2.1. Component Description

The Domain AIMO is the core of a specific AIM domain.

The Domain AIMO NBI exposes to the E2E AIMO an abstracted view of the resources within the AIM Domain, while the Domain AIMO SBI provides management and control capabilities for the AIM components and the Pipelines. In addition, it provides the Pipeline stitching in collaboration with the NFVO which oversees their connectivity through the NFVI. The NFV elements closely follow 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 Domain AIMO operates over a hybrid physical and NFV infrastructure (NFVI). The NFV systems orchestrate and manage the virtual functions and their supporting infrastructure. The Domain AIMO, in collaboration with the NFVO, orchestrates and manages the resources required by the AIM components to build a Pipeline. These Components are the CFs, PPFs, MFs that are part of a logical DEs and the DF and PF, implemented by the Domain AIMO, which are responsible for the stitching and exchange among the AIM DEs (and their components) in a Pipeline.

To build a Pipeline, including AIM components' communication across the NFVI as well as configuring and setting up connectivity to AIM VNFs, all these resources need to be orchestrated together. This is the key role of the Domain AIMO.

[R-40] The Domain AIMO SHOULD implement an algorithm that optimizes the selection of the AIM components for building a Pipeline from a given Pipeline template. To allocate the resources, either to instantiate new AIM components and/or to reconfigure existing components, the algorithm SHOULD consider the computation capabilities and load status of the AIM resources as derived from usage data.

Indeed, the algorithm for the instantiation and configuration of AIM Components during the process of building a Pipeline shall take into consideration the actual goal of the automated closed loop, i.e., optimizing a specified cost function (e.g., energy consumption, mitigation of network congestions, etc.).

The Domain AIMO performs its role by operating or orchestrating the Domain resources through the functionalities defined by the DF, PF and the NFVO.

Per R-66/TR-436 [9], the Domain AIMO is implemented as a stand-alone Orchestrator or a function of the Domain Service Orchestrator.

6.2.2. Component Functionality

Similarly, to the E2E AIMO which orchestrates end-to-end Pipelines across multiple AIM domains, the Domain AIMOs, to fulfill the E2E AIMO requests via domain-scoped Pipelines, rely on the AIM Pipeline Catalogue and the AIM Resource Inventory which are key information bases shared among orchestration elements of the AIM solution (see Section 6.1.2). The AIM Pipeline Catalogue contains the definition of the Pipeline resources through ETSI-compliant descriptors.

While the E2E AIMO orchestrates end-to-end Pipelines across multiple AIM domains, the Domain AIMOs orchestrate domain-scoped Pipelines to fulfill the E2E AIMO requests. Through the AIM Pipeline Catalogue, each domain exposes the Pipeline Templates specific to that domain made available for higher-level end-to-end Pipelines.

The pipeline creation flow is as follows:

- A Pipeline Template from the Catalogue is mapped into a Pipeline Service, the available resources are identified from the AIM Inventory and the sequence of necessary AIM Pipeline components is planned.
- The Pipeline Service, also named simply a Pipeline, is actually created by provisioning communication paths through its AIM components by leveraging message forwarding functions provided by the common message bus.

[R-41] The Domain AIMO MUST oversee the creation and orchestration of domain-scoped Pipelines and the management of the Pipeline Templates exposed via the Pipeline Catalogue.

[R-42] The Domain AIMO MUST create a Pipeline Service (also named simply a Pipeline) upon checking the availability and usage of already instantiated AIM resources from the AIM Inventory and planning of the sequence of AIM Pipeline components

[R-43] The Domain AIMO MUST identify the Pipeline Template suitable for fulfilling the Pipeline Service request from the E2E AIMO and then enrich the Pipeline Template with the information necessary to create the actual Pipeline instance.

[R-44] The Domain AIMO MUST send the appropriate requests to the NFVO (or alternative virtualized infrastructure orchestrator, e.g., a Kubernetes Controller) for the instantiation of the VNFs/CNFs instances corresponding to AIM Pipeline Components that need to be created, scaled modified and removed. Virtual from scratch

[R-45] The Domain AIMO MUST create the Pipeline Service by provisioning communication paths through its AIM components by leveraging message forwarding functions provided by the common message bus.

[R-46] The Domain AIMO MUST monitor virtual resources’ usage levels via interaction with the Virtual Infrastructure Manager (VIM) (or alternative virtualized infrastructure manager) relevant for a given Domain.

Figure 7 illustrates the Oe2e-aimo-Ma-d-aimo reference point used for receiving declarative Intents from the E2E AIMO and for exposing an abstracted view of the AIM Domain resources to the E2E AIMO via a notification mechanism.

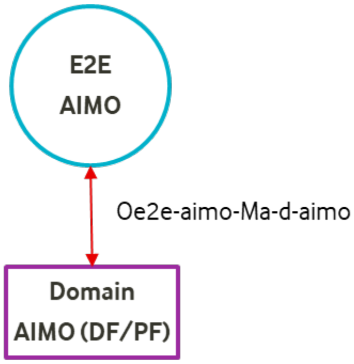

Figure 7: E2E AIMO to Domain AIMO Reference Point

[R-47] The Domain AIMO MUST translate the declarative Intent received from the E2E AIMO into imperative commands/Intents to the AIM DE.

From an SBA perspective, the Domain AIMO performs the role of service broker by interfacing with the requestor of a Pipeline (i.e. the E2E AIMO), while orchestrating the interactions with and among the AIM components with the support of the Distributor Function.

Figure 8 illustrates the Od-aimo-De reference point between the Domain AIMO and the DE used to interact with the AIM functional blocks (CF, PPF and MF components) to configure, manage and orchestrate.

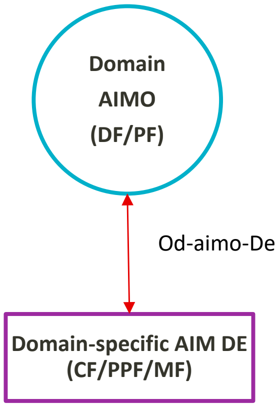

Figure 8: Domain AIMO to Domain-specific AIM DE Reference Point

The AIM components of a DE are agnostic with respect to what control loop they are included in, to allow re-usability of the same block within multiple control loops. This follows the principles of the Service Based Architecture (SBA) which defines how resources and capabilities can be handled as independent services that can be flexibly composed to provide more complex functionalities and address dynamic requirements.

The CF, PPF and MF perform a granular, and self-contained function that can be invoked through well-defined interfaces by other AIM Components. By leveraging service-based architectural principles, the AIM components can be dynamically consumed and composed to provide hierarchical and adaptive Pipelines based on specified requirements.

The Domain AIMO operates its tasks via a set of standard SBIs towards the AIM DEs.

[R-48] The Domain AIMO MUST implement the DF and PF AIM Components.

[R-49] The Domain AIMO with its DF and PF functionalities MAY be implemented as a stand-alone Orchestrator or as a function of the corresponding Domain SDN M&C.

[R-50] When the Domain AIMO is implemented as a stand-alone Orchestrator, it MUST also implement the essential functionalities and interfaces needed to interact with the NFVO.

[R-51] When the Domain AIMO is implemented as a function of the Domain Service Orchestrator, it SHOULD be responsibility of the corresponding Domain SDN M&C to implement the functionalities and interfaces needed to interact with the NFVO.

TR-436 [9] describes the AIMO functions. The following sections focus on the DF and the PF functions as centralized functionalities generally interacting with all AIM DEs Components of instantiated Pipelines. The bullet list below describes the main capabilities of the Domain AIMO and the DF and PF integrated in it.

[R-52] Upon receipt of a Pipeline Service Intent and Pipeline Resources requests from the E2E AIMO, the Domain AIMO MUST

- Retrieve the description of the Pipeline Service Intent and Pipeline Resources Intent from the AIM Pipeline Catalogue
- Verify whether the resources indicated in the Pipeline Resources Intent are already available from the Pipeline Resource Inventory. If not available, the Domain AIMO requests the instantiation of the necessary resources to the NFVO

[R-53] Based on the pipeline description from the Pipeline Service Intent, the Domain AIMO MUST request the following:

To the Distributor Function:

- to populate the Input and Output Endpoint values (e.g., topics taken from a pool the DF has access to, etc.) to be configured on AIM DE components to realize the Pipeline topology

To the Policy Function:

- to provide the access policies used by AIM components for communicating over the message bus (e.g., authentication of DE components and read/write access grants to the Message Bus)
- to define the policies to enforce control and management actions on AIM components (e.g., reconfiguration, scaling, creation of new instances, etc.)
- to define the policies under which a Pipeline is authorized to perform certain actions and the constraints associated with those actions (e.g., issuing of output recommendations to SDN-C based on a precision or recall threshold, etc.)

[R-54] To finalize the Pipeline Service Intent creation, the Domain AIMO MUST:

- Configure the AIM DE components per the Pipeline Service Intent descriptor (see Sections 6.3.2.1 and 6.3.2.2)
- Configure the AIM DE components during runtime
- Activate the pipeline per the planned schedule (e.g., immediate, on scheduled time, on event trigger)
- Pause/stop a pipeline per internal logics and/or external inputs
- Monitor pipeline's indicators (e.g., accuracy/convergence, health) and performs corrective actions
- Deactivate the pipeline (e.g., immediate, on scheduled time, on event trigger, admin trigger)

6.2.2.1. Distributor Function

The AIMO has the capabilities to build up a Pipeline on the basis of the received intent input, the interpretation of the corresponding template in the AIM Pipeline Catalogue and the stitching of the relevant Pipeline components providing the configuration of Endpoints in the CF, PPF and MF.

The AIMO performs the stitching of Pipeline Components through its Distributor Function (DF). The DF enables the stitching of the Pipeline, which is built by using its specific Template and from the AIM Pipeline Catalogue. The Pipeline Template defines the chaining logic as a workflow of functional AIM components that corresponds to the VNF graph defined in GS NFV-MAN 001 [12] and includes the AIM components' types and template definitions to create the Pipeline (e.g., the parameters needed, their order and the connectivity requirements).

[R-55] The DF MUST configure and oversee the distribution of Components' outputs across Pipeline Components and ultimately the distribution of the output of the MF Component to the corresponding SINK nodes.

[R-56] The DF MUST identify the Output and Input Endpoints that fulfill the stitching per the Pipeline template and instantiation requested by the Domain AIMO. For example, if the communication among AIM components is implemented via a Kafka bus, the DF assigns Endpoints to Kafka topics per the appropriate pipeline topology.

6.2.2.2. Policy Function

The Policy Function (PF), embedded in the AIMO, contributes to a policy-based orchestration of Pipelines and their components. Policy-based orchestration abstracts individual component parameters and instead controls their policies..

Pipeline policies provide a way to express and handle non-functional requirements. As such, a Pipeline policy consists of statements that express Service policies and Pipeline components policies, which may include several aspects of the SBA architecture like security, privacy, manageability etc. and should be The Domain AIMO MUST implement the PF.

[R-57] The PF MUST configure various types of policies pertaining the Pipeline Components and the Pipelines themselves.

[R-58] The PF MUST apply policies per Pipeline Component and/or per Pipeline.

[R-59] The PF MAY be used to monitor model performances.

[R-60] The PF MAY govern the impact of the output to a live operational environment or to other systems (e.g., an AIM Sandbox) thanks to specific rules that may be put in place and applied to the AIM components.

Figure 9 illustrates the information model that describes the Pipeline Policy.

Policies are used to define the behavior governing Pipelines and their components within a domain.

[R-61] The PF MUST support at least the following policies:

- AIM components access policies
- AIM components control and management enforcement
- Pipeline output authorization policies

[R-62] The PF MUST derive each component's policy from the Pipeline policy and, in collaboration with the DF function, optimizes the placement and the configuration of each component in the runtime environment.

Policies may be used between Components or between Pipelines and used to create templates that can be replicated in a new environment (e.g., an AIM Sandbox etc.).

6.2.2.2.1. Pipeline Policy Information Model

Figure 9 illustrates the Pipeline policy information model.

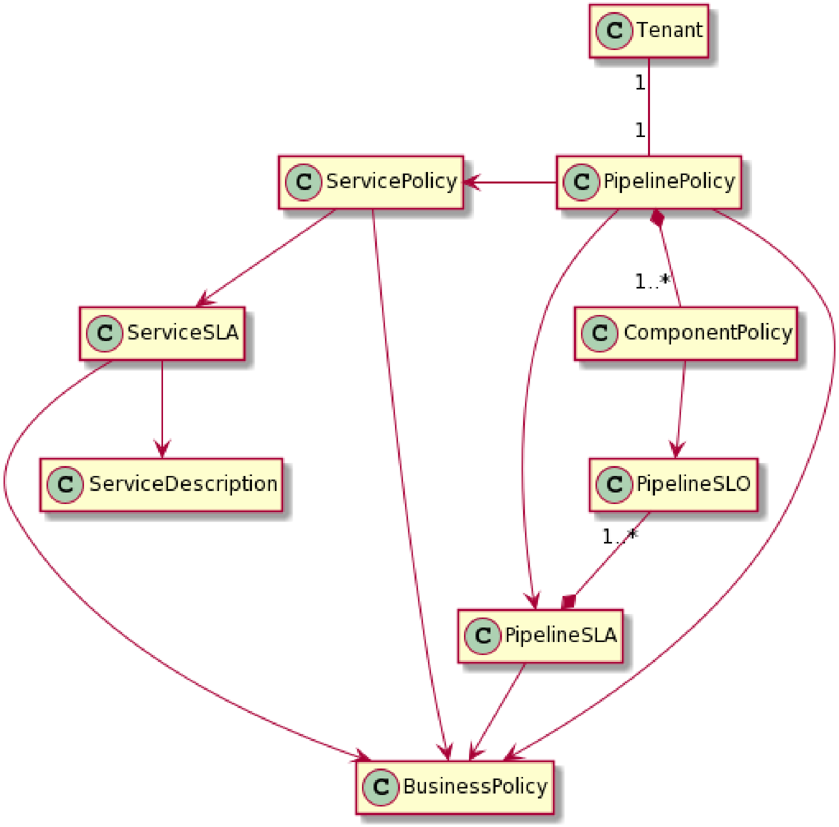

Figure 9: Pipeline Policy Information Model

[R-63] The Pipeline Policy MUST comply with the diagram in Figure 9.

The Information Model describes the Pipeline policy as a class that depends on the Service policy and aggregates 1 or more Component policy.

The service branch provides information on any policy that applies to the Service and depends on the Service SLA and the Business Policy.

The Component Policy branch provides information on the SLAs associated with the Pipeline itself and its component. In fact, it depends on the Business policies and the Pipeline SLA that translate into specific Service Level Objectives (SLO) for each Pipeline Component.

For instance, a Pipeline could require 99.999% availability 90% of the time as it monitors a critical service that would translate into similar SLOs for its components. Similarly, another Pipeline could require its AI/ML model to perform at 97% accuracy that would translate into similar SLO for its ML component.

The Component Policy also describes the authentication and authorization policies for the component.

[R-64] The PF MUST support the classes of the Pipeline policy information model as described in Table 3.

Table 3: Pipeline Policy Information Model Class Description

Class Name	Description
Tenant	The owner of the Policy to be created
ServicePolicy	The Service policy (e.g., service priority etc.)
PipelinePolicy	The policy associated to the Pipeline
ComponentPolicy	The policy associated to each component of the Pipeline
PipelineSLO	The Component SLO derived from the Pipeline SLA
PipelineSLA	The SLA that describes the Pipeline contract (e.g., the accuracy of the ML model)
ServiceSLA	The SLA that describes the Service contract (e.g., service availability etc.) and/or other constraints the Service has to fulfil.
BusinessPolicy	The Business policy (e.g., Customer data must be encrypted etc.)

[R-65] The AIMO SHOULD use the Component Policy and Pipeline SLO Information Elements to monitor a Pipeline based upon its PipelinePolicy.

6.2.3. Intent Base

This section presents the intents used by the AIMO in collaboration with the NFVO to orchestrate the creation of the resources for a Pipeline and its configuration.

The declarative intents, sent by the CML to the E2E AIMO and by the E2E AIMO to the Domain AIMO to implement a pipeline, are mapped to imperative commands used to create the required NFVI resources and to configure the AIM DE components.

Figure 10 illustrates the association between the intent used to create the pipeline (PipelineIntent), the one used to create the pipeline resources (PipelineResourceIntent) and the one used to create the pipeline service (PipelineServiceIntent). The association is realized through the Pipeline Description and the Pipeline Template in the AIM Pipeline Catalogue.

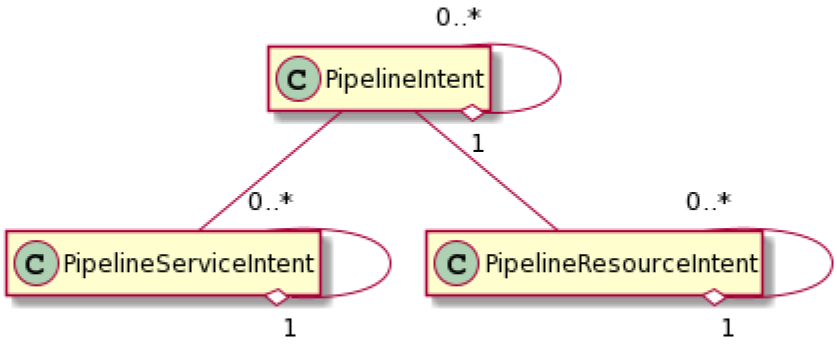

Figure 10: Pipeline Intents Association

[R-66] The Pipeline Intents Association MUST comply with the diagram in Figure 10.

6.2.3.1. Pipeline Resource Intent Information Model

The PipelineResourceIntent is used to instantiate the virtual resources required for a Pipeline by the AIMO 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., Openstack, Kubernetes) and instantiates the NFVI resources.

[R-67] The AIMO MUST implement the PipelineResourceIntent in compliance with Figure 11, to instantiate the virtual resources required for a Pipeline, 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., Openstack, Kubernetes) and instantiates the NFVI resources.

Figure 11 illustrates the Information Model used to describe the PipelineResourceIntent.

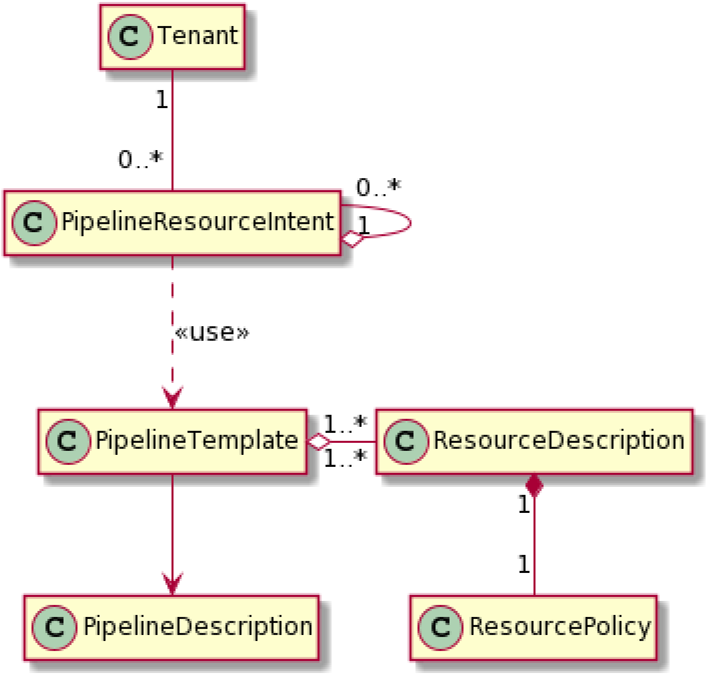

Figure 11: Pipeline Resource Intent Information Model

[R-68] The Pipeline Resource Intent MUST comply with the diagram in Figure 11.

The PipelineResourceIntent can aggregate multiple PipelineResourceIntents to accommodate simple as well as recursive creation of resources for hierarchical Pipelines either within a single or multi-AIM domain. In fact, each PipelineResourceIntent will "use" the PipelineTemplate from the Pipeline Catalogue from where it can retrieve the information and characteristics of the virtual resources required by the Pipeline and its components.

The usage relationship requires that the model element (PipelineResourceIntent) invokes another model element (PipelineTemplate) for full implementation or operation.

The PipelineTemplate depends on the PipelineDescription and aggregates multiple resources. Each resource can have an attached a policy.

Table 4 presents a description of the classes of the information model.

[R-69] The PF MUST support the classes of the PipelineResourceIntent Information Model as described in Table 4.

Table 4: Pipeline Resource Intent Information Model Class Description

Class Name	Description
Tenant	The owner of the resources to be created
PipelineResourceIntent	The intent used to create the resources
ResourceDescription	The description of the resources (e.g., IEs required to create a VNF or CNF)
PipelineTemplate	The detailed template of the Pipeline in the AIM Pipeline Catalogue (e.g., IEs required to create a Pipeline in a Catalogue)
PipelineDescription	The description of the hierarchical structure of the Pipeline (e.g., IEs that describe the Pipelines from the AIM Pipeline Catalogue of multiple domains)
ResourcePolicy	The policy attached to a resource (e.g., security and other constraints)

Table 4 describes some of the information elements, operations and information models exposed by the Oe2e-aimo-Ma-d-aimo interface for the creation of the pipeline resources.

6.2.3.2. Pipeline Service Intent Information Model

The PipelineServiceIntent is used to instantiate a Pipeline (and its components) that conforms to the specification and requirements expressed by the PipelineIntent received by the E2E AIMO.

[R-70] The AIMO MUST allow flexible placement, distribution, chaining and execution of the Pipeline components taking into account the requirements/attributes of the PipelineServiceintent and the capabilities of the underlying infrastructures.

[R-71] Upon request from its northbound orchestration system, the AIMO MUST create and activate of the Pipeline RFS via declarative Intents and map the Intent and its attributes to the RFS through the Pipeline Template in the s Pipeline Catalogue and activates the pipeline.

Figure 12 illustrates the Information Model used to describe the PipelineServiceIntent.

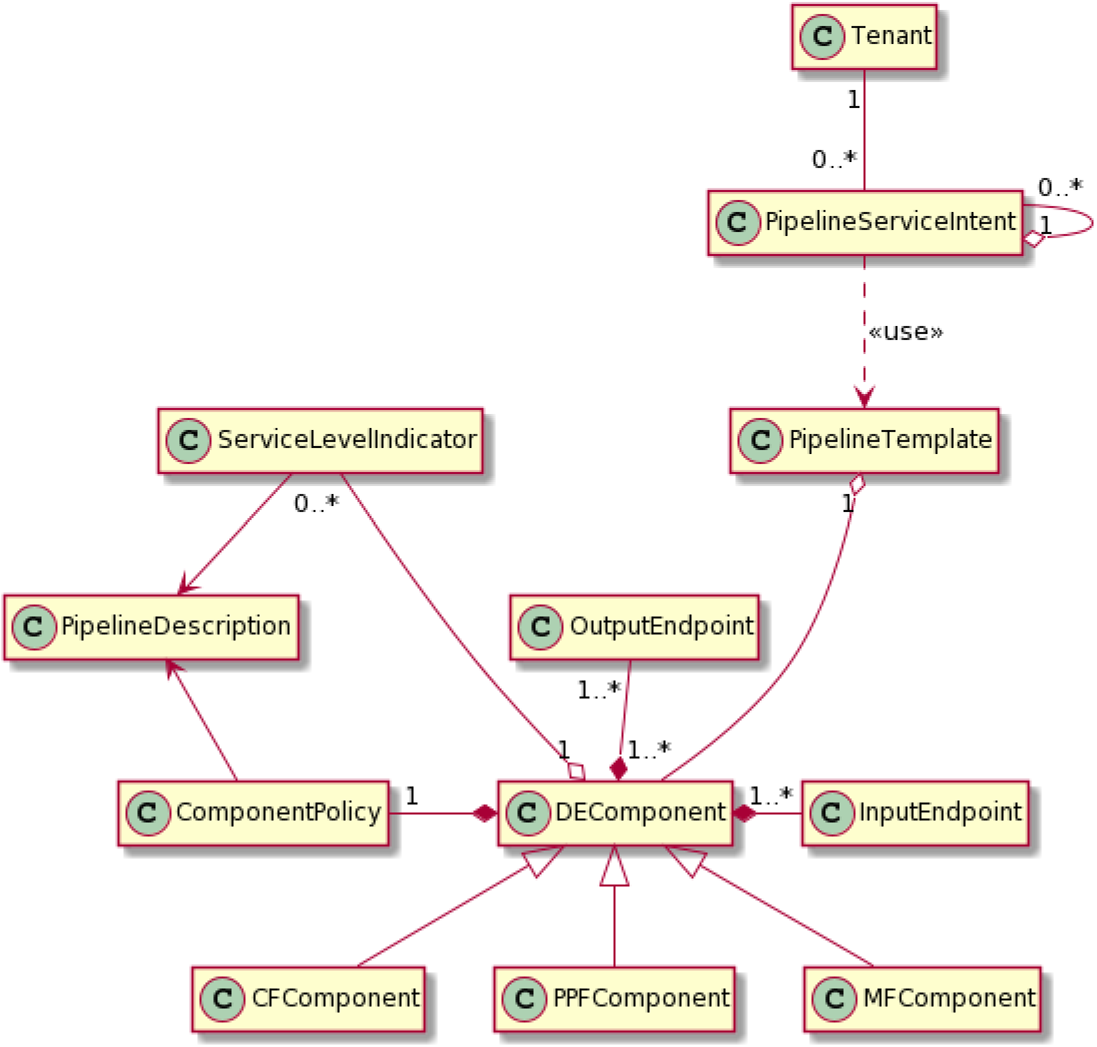

Figure 12: Pipeline Service Intent Information Model

[R-72] The Pipeline Service Intent MUST comply with the diagram in Figure 12.

The PipelineServiceIntent can aggregate multiple PipelineServiceIntent to accommodate simple as well as recursive instantiation of components for hierarchical Pipelines either within a single AIM Domain or multiple AIM domains. In fact, each PipelineServiceIntent will "use" the PipelineTemplate from the AIM Pipeline Catalogue from where it can retrieve the information and characteristics of the components required by the Pipeline.

The PipelineTemplate aggregates multiple DEComponent from where the CFComponent, the PPFComponent and the MFComponent are inherited.

The DEComponent is composed of the InputEndpoint, the OutputEndpoint and the ComponentPolicy. It also aggregates multiple ServiceLevelIndicator.

Both the ComponentPolicy and the ServiceLevelIndicator depends on the PipelineDescription

[R-73] The AIMO MUST support the classes of the PipelineServiceIntent policy information model as described in Table 5.

Table 5 presents a description of the classes of the PipelineServiceIntent Information model, and describes some of the information elements, operations and data models exposed by the E2E AIMO and the Domain AIMO across the Cm-Ma-e2e-aimo and Oe2e-aimo-Ma-d-aimo interfaces respectively, for the creation of the Pipeline service.

Table 5: Pipeline Service Intent Information Model Class Description

Class Name	Description
Tenant	The owner of the Pipeline Service to be created
PipelineServiceIntent	The intent used to create the service
ServiceLevelIndicator	The Service Level Indicator (SLI) used to monitor the SLO derived from the SLA defined in the ServiceDescription of the Pipeline Intent
DEComponent	The abstraction of the CF, PPF and CF components that aggregates common IEs like ComponentPolicy, SRC and SINK
ComponentPolicy	The policy attached to the DE Component (e.g., security etc.)
CFComponent	The specialization of the DE Component for the AIM CF
PPFComponent	The specialization of the DE Component for the AIM PPF
MFComponent	The specialization of the DE Component for the AIM MF
InputEndpoint	The IE that describes the source of the data for the DE component
OutputEndpoint	The IE that describes the destination of the outcome of the DE component

6.2.4. Management Capabilities

[R-74] The Domain AIMO MUST support the following AIM Pipeline Catalogue capabilities:

- Lifecycle Management of the Pipeline Templates (shared with the Domain AIMOs)
- Exposure of the Catalogue Resources
- Management of Catalogue Notifications

[R-75] The Domain AIMO MUST support the following AIM Pipeline Inventories capabilities:

- Inventory of the Pipeline Intents

- Inventory of the Pipeline Resources
- Inventory of AIM Components
- Logical topology and connections

[R-76] The Domain AIMO MUST support the following Domain control capabilities:

- AIM Resource creation
- AIM Components Configuration
- Domain Pipeline Intent Fulfillment
- Domain Pipeline Intent Check
- Domain Pipeline Intent Realization
- Domain Pipeline Policy Management

[R-77] The Domain AIMO MUST support the following Domain Monitoring capabilities:

- AIM Component Fault, Security and Performance Events
- Domain Pipeline Intents Performance and Events
- Domain Pipeline Intent Fault and Security Events

[R-78] If the NFVO is incorporated in the Domain AIMO, this latter MUST support the following NFVO capabilities:

- Management of Pipeline Components Descriptors and VNF packages
- Lifecycle management of Pipeline Components and VNFs
- Policy management and/or enforcement of those resources and the NFVI in general

6.3. Decision Element Interfaces

An AIM Decision Element (DE) is a logical component that implements decision-making capabilities via some degree of intelligence and reasoning over inputs mainly received from the associated Managed Entities (MEs).

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

6.3.1. Component Description

AIM DEs analyze and interpret the environment via the received inputs, suggesting actions according to inputs from other sources of information and network policies. In addition, DEs decide which recommendation fits best their automated loop purpose and constraints and sends the selected recommendations to the Controller in charge of acting upon the managed entity.

Referring to the purpose of the AIM DEs, it is useful to remind two conceptual categories defined in section 6.10 of TR-436 [9]:

- **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.

- **Holistic DE:** These AIM DEs, lay at higher-levels of the AL hierarchy and may analyze information from multiple federated domains and apply cognitive approach models to infer a holistic view and contribute to the end-to-end decision-making process (e.g., service issue, upselling/cross-selling opportunity, sophisticated customer profiling).

6.3.2. Component Functionality

The AIM DE is a logical component that includes the following pipeline components:

- **Collection Function (CF):** responsible for data collection from MEs
- **Pre-processing Function (PPF):** responsible for processing the data collected
- **Model Function (MF):** responsible for knowledge handling and recommendations generation

A Pipeline is a deployment of a combination of the above components, according to a predefined Pipeline Template, along with the interactions with the DF (see section 6.2.2.1) and PF (see section 6.2.2.2) implemented in the AIMO.

AIM DEs may also interact with the Knowledge Base (KB) through the Logical Reference Point D as described in section 5.4.

The interactions of an AIM DE with the corresponding AIMO and other elements and the related interfaces are shown in Figure 1.

Different interfaces are used to exchange data between AIM components within a DE or other DEs/components and the SDN-C, such as:

- **Oe2e-aimo-De (6.1.2):** the interface for the E2E AIMO to configure, manage and orchestrate the AIM functional blocks of an E2E AIM DE
- **Od-aimo-De (6.2.2):** the interface for the AIMO to configure, manage and orchestrate the AIM functional blocks of a Domain-specific AIM DE
- **De-Me (6.3.2.1):** the reference point for the collection function of the AIM DE to collect data from the Managed Entities (ME)
- **De-Nf-ccodo (7.4):** the interface for the AIM DE to interact with the Cross-domain Orchestrator. The recommendations from the end-to-end AIM DE are pushed to the Cross-domain Orchestrator via declarative Intents. The Cross-domain Orchestrator translates the Intent into domain-specific intents towards the appropriate domain SDN-C and then translated into atomic commands to the Managed Entity. The De-Nf-ccodo interface is derived from the Os-Ma-ccodo reference point, defined in TR-411 [7]
- **De-Nf-sdn (7.4):** the reference point for the AIM DE to interact with the SDN-C. The recommendations from the AIM DE are pushed to the SDN-C via declarative Intents. The SDN-C then translates the Intent into atomic commands to the Managed Entity in order to close the loop and fulfill its AIM purpose (e.g. self-heal and/or self-optimize the network). The De-Nf-sdn reference point is derived from the Occo-Nf-sdn-x reference point, defined in TR-411 [7]
- **De-Mb** (6.3.2.1, 6.3.2.2, 6.3.2.3): is the interface to exchange information between DEs, between components within a DE and with the Knowledge Base (KB), e.g. to consume 3rd party services like data cataloguing or to store/retrieve data generated by that DE or other DEs.

A Knowledge Base (KB) is used for basic support for storing knowledge during the AIM operation. The AIM DE components may use KB for:

- Events Notifications
- Storing data from the associated MEs. These data could be stored for a limited period, maintaining the most updated data only or could provide historical data to the DE

In addition, the De-Mb reference point is also used between AIM DE components. A possible implementation of it may be adopting a message bus.

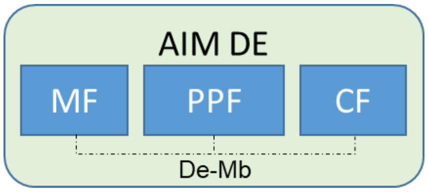

Figure 13: AIM Decision Element (DE)

The AIM DE components are independent from each other and DE components from different suppliers can be combined together in a logical AIM pipeline (i.e., a DE) and interoperate over the standard De-Mb interface specified in Section 7.6 of this Technical Report. There is merit for AIM components suppliers to propose MF and PPF components that are tightly coupled together in terms of optimizing the preprocessing and feeding data into the Model Function.

DE components operate via tasks. Figure 14 illustrates the state machine with the relationship between the status of the task and the interactions between the AIMO and the task, described in Section 7.3.3.

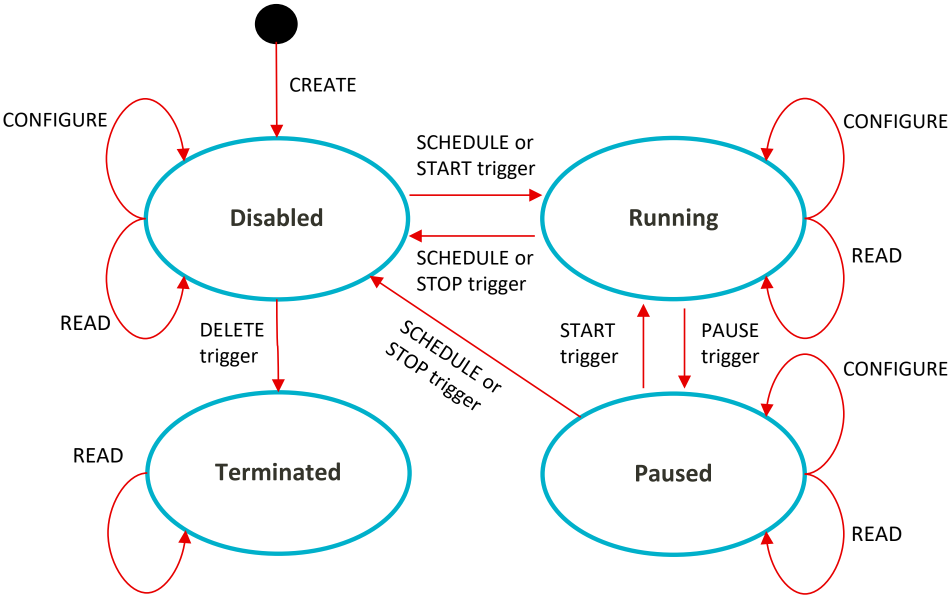

Figure 14: Task State Machine

The operations, over the Oe2e-aimo-De and the Od-aimo-De interfaces, required to configure, monitor, and orchestrate tasks in a CF, a PPF and a MF include:

- *CREATE* – creates and provides the initial configuration of the task and may activate it depending on the task schedule configuration
- *CONFIGURE* – applies updates to the task configuration
- *READ* – reads the configuration of the task, including information and/or its status

State transitions marked with *SCHEDULE*, applicable to Collection Tasks, happen when the actual time meets a configured schedule for a CT. State transition marked with *START, STOP, PAUSE* or *DELETE* triggers happen upon the receipt of a specific command from the AIMO.

The *DELETE* trigger brings the task to the Terminated status, where task metadata are kept for statistical purposes.

The operations required by the AIMO to configure, monitor, and orchestrate CF, PPF and MF and their tasks, are mapped to a stateless protocol and include “Create Read Update Delete” (CRUD) operations regarding the related information elements (see Section6.2.1).

6.3.2.1. Collection Function

The Collection Function (CF) of the AIM DE is responsible for collecting data from one or more source (SRC) node that can be used as input to the AIM pipeline. A Collection Function is configured to run collection campaigns on the SRC by the AIMO.

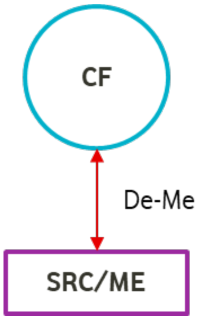

Figure 15: CF to ME Reference Point

Figure 15 illustrate the De-Me interface between the Collection Function (CF) and the Managed Entity (ME) that is the source of the data.

The Collection Function can instantiate the De-Me interface with the required collection mechanism and protocol based on the specific use-case and/or architectural/operational requirement. Independent of the collection mechanism and protocol used, the collected data is represented in a unique way using a standard data model. This requirement might limit the choice of data schema and encoding mechanism available, but it is required to avoid for instance data inconsistency and to facilitate data sharing between AIM components.

Nonetheless, when a proprietary data model or information model is used to represent the data collected from the ME, the Collection Function, before sending it to a destination, normalizes the data to the standard model for the specific type of data.

For instance, to collect data from a legacy device, the SNMP protocol with a proprietary MIB could be used and then the collected data would be converted to a standard YANG model before sharing on the common message bus.

Furthermore, when telemetry data needs to be streamed from a device that supports standard YANG models, NETCONF or gNMI or other protocols are used without requiring any data model translation.

Finally, a Managed Entity might natively support a message bus like Kafka, MQTT, RabbitMQ, Redis etc. to stream data. In this scenario, the Collection Function has to adapt to the message bus supported by the source and make the data available to be consumed via the bus of choice for the communication among the AIM components.

As an example, Figure 16 illustrate some possible alternatives of the combination of data models , encoding schema and collection protocol that the De-Me interface could implement.

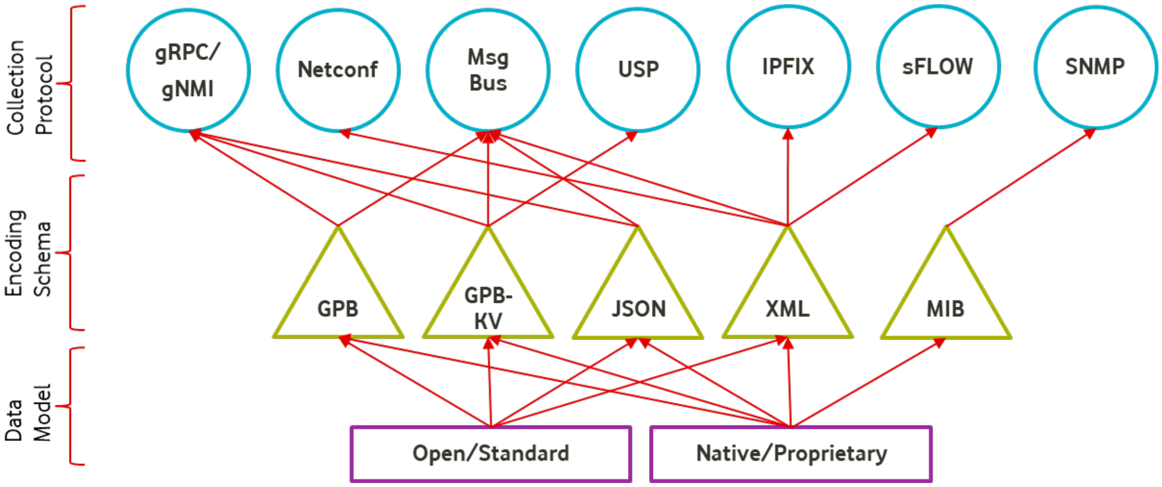

Figure 16: Data Model, Encoding Schema and Collection Protocol Combination Example

6.3.2.1.1. Collection Modes

Section 4.2 of TR-436 [9] specifies the following data collection modes to be supported by the AIM collection function: Streaming/Publish-Subscribe, Bulk, Pull and Push.

[R-79] The Collection Function MUST implement the protocols marked with X in Table 6 and SHOULD support those marked with O in Table 6 in support of the listed Collection Modes.

Note: the content of Table 6 is subject to revision in future Issues of this Technical Report.

Table 6: Collection Modes and Collection Protocols

Collection Mode / Collection Protocol	Streaming/ Publish-Subscribe	Bulk	Pull	Push
Netconf			X	X
gRPC/gNMI	O	O	X	X
Message Bus (Kafka, MQTT, RabbitMQ, Redis etc.)	X			
USP (Note 1)	O	X	X	X
IPFIX		X		X
sFLOW				O
SNMP			X	X

Note 1: USP applies to Data Collection in the Home Customer Premises domain

6.3.2.1.2. Collection Task

Data (e.g., telemetry, PM counters, alarms, notifications etc.) are typically produced by a source that publishes the data to a collector at a cadence that is needed by the consuming entity and that the source can provide.

The data collection requires configuring properties such as:

- How often the dataset is generated by the source
- How long will the dataset be generated by the source
- What information is comprised in the dataset (e.g., data model, X-Path, MIB object etc.)
- Collection mode (e.g., Streaming/Publish-Subscribe, Bulk, Pull, Push)
- The destination of the collected data

These properties are included in the Intent used to configure a CF and describe a collection task (also called campaign) of the AIM framework defined in TR-436 [9].

With respect to data collection, the AIMO manages and oversees collection tasks in a centralized and long-term way (for a given domain of sources/MEs), while AIM pipelines and DEs are created and terminated over time.

[R-80] The CF MUST operate via Collection Tasks and expose their management via the Oe2e-aimo-De and the Od-aimo-De interface.

[R-81] Collection Tasks MUST operate per the state machine described in Figure 14.

[R-82] Referring also to R-9 from TR-436 [9] the Intent used to configure a Collection Task over the Oe2e-aimo-De and the Od-aimo-De interface MUST include at least the properties in Table 7.

Table 7: Collection Task Properties

Properties	**Description**
ID	Task identifier automatically assigned
Name	Name of the Task
Description	Description of the Task. For example, this can describe the type of collection per one Operator's conventional categories. E.g., continuous background collection, troubleshooting, monitoring, analytics, trend analysis, etc.
Managed Entity	In the context of the CF, it identifies the endpoint that is the Source of the data. E.g., the name of the ME in a private DNS hosted zone or a network socket.
Data Model(s)	Specifies the data model to be used after the conversion of the data by the Collection Function before sending it to the Data Destination
Data Path	Within the Data Model, specifies the path of the data
Data Destination	Specifies where to send the data after it has been converted
Collection Interval	Specifies how often the data is collected E.g., fixed interval, variable interval, real-time, on-change etc.
Max Data	Specifies the maximum amount of data that the task can handle for security and performance reasons
Schedule	Specifies a schedule for the task according to the following parameters: • Start Time • Stop Time • Duration Interval Depending on the use cases, there are various options that can be tailored via the above parameters. ▪ None specified: Collection task is created on CF but not activated

Properties	Description
	▪ Only Start Time specified: Collection starts at “Start Time” and continues indefinitely until further change ▪ Only Stop Time specified: Collection starts immediately and stops at “Stop Time” ▪ Start and Stop Time specified: Collection starts at “Start Time” and stops at “Stop Time” ▪ Only Duration Interval Specified: Collection starts immediately and lasts for the “Duration Interval”.
Mode	Specifies the mode used to collect the data. E.g., Streaming, Bulk, Pull, Push.
Status	Specifies the status of the task E.g., Enabled, Disabled, Running, Terminated.
Policies	• Access policies • Control and management enforcement

6.3.2.2. Pre-Processing Function

The Pre-Processing Function (PPF) of the AIM DE is responsible for elaborating data collected by one or more sources, to make them in a suitable form, so that the AIM Model Function can consume it.

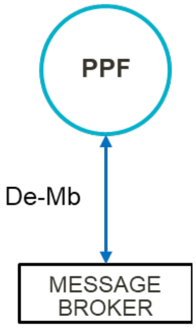

Figure 17: AIM DE Pre-Processing Function (PPF) Interfaces

The PPF communicates with other components of AIM DE by using the De-Mb reference point and its main features are:

- formatting and pre-processing/filtering of collected data
- attaching metadata to pre-processed data (e.g., to single records or to groups of them)
- aggregation of the collected data

[R-83] The PPF MUST operate via Preprocessing Tasks and expose their management via the Oe2e-aimo-De and the Od-aimo-De interface.

[R-84] Preprocessing Tasks MUST operate per the state machine described in Figure 14.

[R-85] The Intent used to configure a Preprocessing Task over the Oe2e-aimo-De and the Od-aimo-De interface MUST include at least the properties specified in Table 8.

Table 8: PPF Properties

Properties	**Description**
ID	Preprocessing Task Identifier automatically assigned
Name	Name of the Preprocessing Task
Description	Description of the Preprocessing Task
Input Endpoint	In the context of the PPF, it identifies the endpoint that is the Source of the data. E.g., the topic to subscribe to get input data from an AIM element (e.g., the CF, Knowledge Base, etc.)
Input Data Model(s)	Specifies the data model(s) used to describe the data received from the Input Endpoint Note: in principle this data model is the same used by the CF to model the data collected from MEs
Input Data Path	Within the data model, specifies the path of the data
Output Data Model(s)	Specifies the data model(s) to be used before sending data to the Output Endpoint. Note: this data model refers to the description of preprocessed data and may differ, though resemble, to the model for data collected from MEs by the CF
Output Endpoint	Specifies where to send the data. E.g., the topic where to publish processed data
Status	Specifies the status of the Preprocessing task E.g., Enabled, Disabled, Running, Terminated.
Policies	• Access policies • Control and management enforcement

6.3.2.3. Model Function

The AIM DE always embeds (at least) one Model Function (MF) which is the main component of AIM DEs and it is responsible for the decision-making process based on the interpretation of the environment inputs and for generating (not necessarily executing) feedback actions that “close the loop”.

The decision-making consists in selecting the best actions among several possible alternatives and can be improved by learning techniques for the dynamic construction of new knowledge.

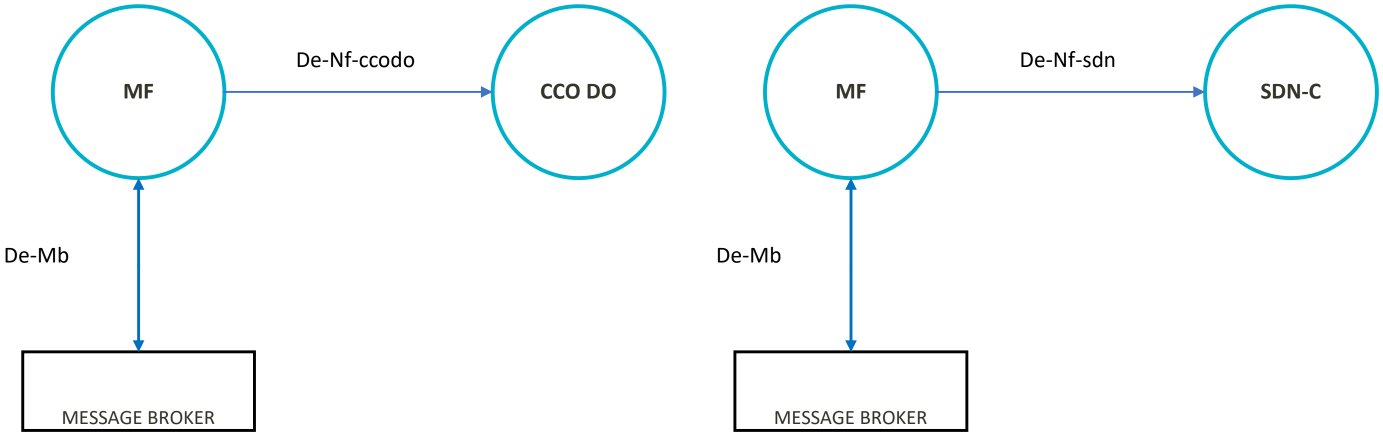

Figure 18: AIM DE Model Function (MF) Interfaces

The levels of intelligence that one DE may implement, according to the levels of complexity in building their knowledge and applying it, can be grouped in:

- **Machine Learning AI (ML AI):** Trained (supervised or unsupervised) rather than explicitly programmed rules and models
- **Symbolic AI:** Hardcoded rules crafted by programmers that do not involve any learning

The AIM framework supports DEs that implement ML AI or Symbolic AI, or both.

[R-86] The MF MUST operate via Decision-Making Tasks and expose their management via the Oe2e-aimo-De and the Od-aimo-De interface.

[R-87] Decision-Making Tasks MUST operate per the state machine described in Figure 14.

[R-88] The Intent used to configure a Decision-Making Task over the Oe2e-aimo-De and the Od-aimo-De interface MUST include at least the properties specified in Table 9.

Table 9: MF Properties

Properties	**Description**
ID	Decision-Making Task identifier automatically assigned
Name	Name of the Decision-Making Task
Description	Description of the Decision-Making Task. It can also include information such as: • Knowledge Model: ○ ML AI supervised ○ ML AI unsupervised ○ Symbolic AI • AI Model: ○ To identify the ML AI model used (e.g., name, version, developer)
Input Endpoint	In the context of the MF, it identifies the endpoint that is the Source of the data. E.g., the topic to subscribe to get input data from an AIM element (e.g., the PPF, CF, Knowledge Base, etc.)
Input Data Model(s)	Specifies the data model(s) used to describe the data received from the Input Endpoint.

	Note: in principle, this data model is the same used by the PPF to model preprocessed data or by the CF to model the data collected from MEs
Input Data Path	Within the Data Model, specifies the path of the data
Output Data Model(s)	Specifies the data model(s) to be used before sending data to the Output Endpoint.
Recommendation Endpoint	Specifies to which SDN-C element recommendations have to be sent
Output Endpoint	Specifies where to send the processed data. E.g., the topic where to publish processed data in the case of KB as output endpoint.
Status	Specifies the status of the Modeling task E.g., Enabled, Disabled, Running, Terminated.
Policies	• Access policies • Control and management enforcement • Pipeline output authorization policies

6.3.2.4. Intent Base

The MF component makes use of declarative intents to generate “recommendations” targeting the MEs.

Concerning the Reference Architecture of TR-436 [9], the MF steers its recommendations to the target MEs, through the cross-domain SDN Orchestrator or the Domain SDN-C responsible for their configuration and management, using the RFS Intent information models described in TR-411 [7] as described in Section 7.4.2.

6.3.3. Management Capability

[R-89] The CF, PPF and MF (i.e., the Pipeline Components) MUST support the following monitoring capabilities:

- AIM Components Performance
- AIM Components Diagnostic
- AIM Components Fault, Security and Performance Events
- AIM Components Fault, Security and Performance Events.

[R-90] The CF, PPF and MF MUST support the following troubleshooting capabilities:

- AIM Components Logging.

7. AIM Interfaces Specification

This section, reprising the interfaces description provided in Section 6, specifies the requirements for the interfaces between elements of the AIM framework as depicted in Figure 1, notably:

- Cm-Ma-e2e-aimo
- Oe2e-aimo-Ma-d-aimo
- Oe2e-aimo-De
- Od-aimo-De
- De-Me
- De-Nf-ccodo
- De-Nf-sdn
- De-Mb

Support for TM Forum APIs

The AIM Orchestrators' NBIs (Cm-Ma-e2e-aimo, Oe2e-aimo-Ma-d-aimo), like those of many systems at the higher layers of modern SDN hierarchical architectures, can leverage on the rich portfolio of TM Forum's APIs for intent-based interfaces.

These APIs are specified in a generic way and can then be specialized via what's called an Ontology, specific for a given business, service, or resource scope. The generic TM Forum API needs to be supported by the counterpart systems and the specific Ontology has to be defined and exchanged between them in an initial handshake phase.

The TM Forum has produced specifications and other documents on end-to-end performance management and AI-based Closed Loop Automation (CLA).

In the former, the TM Forum defined the document GB1028 [19] that is a Guidebook on how to use some TM Forum APIs to achieve end-to-end performance management.

In the latter, the TM Forum produced an Introductory Guide IG1219 [20] that defines blueprints, information models and API requirements for AI CLA focusing on Anomaly Detection and Resolution.
IG1219 [20] provides the basis for the design of an AI-based autonomous network that aims to achieve a zero-touch resolution of faults.

The adoption of TM Forum recommendations ensures a high grade of interoperability and deployability between the involved layers:

- The Customer Management Layer and the E2E AIMO (Cm-Ma-e2e-aimo interface)
- The E2E AIMO and the Domain AIMOs (Oe2e-aimo-Ma-d-aimo interfaces)

TR-436 [9] and this Technical Report are applicable to a wide set of use cases beyond the most commonly heard applications in the space of Anomaly Detection and Resolution.

As such, these Technical Reports are applicable for developing and deploying a generic AIM framework.

At the time of publication of this Technical Report, the TM Forum is working towards a set of well-defined APIs to manage a framework that exploits CLA for anomaly detection and other use cases.

The specification of the Cm-Ma-e2e-aimo and the Oe2e-aimo-Ma-d-aimo interfaces reuses the recommendations of Guidebook GB1028 [19] on how to use some of the TMForum APIs for E2E performance management, that is a prerequisite to achieve a Zero-Touch CLA.

7.1. E2E AIM Orchestrator Interfaces

The Cm-Ma-e2e-aimo is the communication interface between the Customer Management Layer and the E2E AIMO. This interface exposes to the Customer Management Layer an abstracted view of the AIM resources and the Catalogue and Inventory of AIM tasks and services of the AIM framework, specified in TR-436 [9].

This interface allows intent-based requests to flow to the E2E AIMO from systems of the Customer Management Layer (e.g., Analytics, CRM & Order management, Assurance, Billing, etc.).

This interface may be implemented either as an extension of the Os-Ma-ccodo interface defined in TR-411 [7], when the E2E AIMO is integrated in the CloudCO Domain Orchestrator, or as a stand-alone interface.

Regardless of these two options the Cm-Ma-e2e-aimo interface is required to follow the functional requirements reported in the following sections.

7.1.1 Protocol requirements

The Cm-Ma-e2e-aimo interface requires a flexible protocol to allow intent-based request for AIM tasks as well as the monitoring of the intents status. The RESTful HTTP protocol is the preferable solution since it offers a simplified way of interaction between the Customer Management Layer and the E2E AIMO.

A REST-based interface can easily be extended to support queries and the Customer Management Layer can receive notifications about specific requests and tasks.

[R-91] The Cm-Ma-e2e-aimo interface MUST be based on YAML/JSON models and grammars to expose the AIM solution resources and services via primitive service elements.

[R-92] The Cm-Ma-e2e-aimo interface MUST be an HTTP and HTTPs REST-like interface.

[R-93] The Cm-Ma-e2e-aimo interface MUST be an intent-based interface.

[R-94] The Cm-Ma-e2e-aimo interface’s service elements MUST be consumed via Create, Read, Update, Delete and Notifications (CRUD-N) actions between the Customer Management Layer and the E2E AIMO.

7.1.2 Requirements for the support of TM Forum APIs

As discussed at the beginning of Section 7, the Cm-Ma-e2e-aimo adopts the TM Forum APIs, notably:

- Service Inventory: TMF638 [21] offers Customer Management Layer systems a standard means of querying, updating, and receiving notifications from the AIM Pipeline Catalogue including the AIM pipeline service state recorded in the catalogue.
- Service Catalog Retrieval: TMF633 [23] offers the Customer Management Layer systems a standard means of querying the run-time AIM Pipeline Catalogue to determine the AIM pipeline service types available. In addition, TM633 [23] offers a complete set of CRUD-N capabilities to lifecycle manage the content of the run-time AIM Pipeline Catalogue.
- Service Fulfillment: TMF641 [24] offers the Customer Management Layer systems a standard means of ordering services from the E2E AIMOs. Notifications of changes to the AIM Pipeline intent order item(s) state and service instance are also offered. Error handling notifications are exposed northbound to the Customer Management Layer systems.
- Service Quality Management: TMF657 [25] specifies the API used to access a Service Quality Management application and extract Service Level Specifications and associated Service Level Objectives (SLO) and their thresholds. This can be used to monitor the SLA of a specific AIM pipeline service and generate alarms in case of a violation of threshold.

- Performance Thresholding TMF649 [26] offers the API to identify exceptional behavior of monitored performance indicators of an AIM Pipeline service. If a threshold is crossed, the API generates an alarm.
- Service Level Agreement: TMF623 [27] specifies the SLA API which provides a standardized interface for AIM pipeline SLA life cycle management to provide AIM pipeline services with attached SLA in the shared AIM Pipeline Catalogue. The typical SLA LCM phases are SLA configuration, SLA activation/enforcement, SLA operations, SLA violation/consequence handling, SLA reporting.
- Performance Management: TMF628 [28] specifies the REST API for Performance Management which provides a standardized mechanism for performance management such as the creation, partial or full update and retrieval of AIM pipeline resources involved in performance management
- Alarms: TMF642 [29] offers the API to manage alarms from monitored AIM pipeline resources. The API does not assume a particular management layer, so it can be either AIM pipeline resource or AIM pipeline service layer.
- Event Management: TMF688 [30] specifies the Event Management API to manage events that are coming from other TMF APIs. It uses a publish/subscribe pattern to manage messages between event producers and consumers. TMF688 [30] suggests the utilization of message brokers, such as Kafka or RabbitMQ, to implement this API.

[R-95] The E2E AIMO MUST provide the capability to support the TMF638 [21] Service Inventory REST API Specification across the Cm-Ma-e2e-aimo interface.

[R-96] The E2E AIMO MUST provide the capability to support the TMF633 [23] Service Catalog Retrieval REST API Specification across the Cm-Ma-e2e-aimo interface.

[R-97] The E2E AIMO MUST provide the capability to support the Service Fulfillment: TMF641 [24] Service Fulfillment REST API Specification across the Cm-Ma-e2e-aimo interface.

[R-98] The E2E AIMO MUST provide the capability to support the TMF657 [25] Service Quality Management REST API Specification across the Cm-Ma-e2e-aimo interface.

[R-99] The E2E AIMO MUST provide the capability to support the TMF649 [26] Performance Thresholding REST API Specification across the Cm-Ma-e2e-aimo interface.

[R-100]The E2E AIMO MUST provide the capability to support the TMF623 [27] Service Level Agreement REST API Specification across the Cm-Ma-e2e-aimo interface.

[R-101]The E2E AIMO MUST provide the capability to support the TMF628 [28] Performance Management REST API Specification across the Cm-Ma-e2e-aimo interface.

[R-102]The E2E AIMO MUST provide the capability to support the TMF642 [29] Alarm REST API Specification across the Cm-Ma-e2e-aimo interface.

[R-103]The E2E AIMO MUST provide the capability to support the TMF688 [30] Event Management REST API Specification across the Cm-Ma-e2e-aimo interface.

7.2. Domain AIMO Interfaces

The Oe2e-aimo-Ma-d-aimo interface is the communication interface between the E2E AIMO and the Domain AIMOs implemented in the AIM solution. This interface exposes to the E2E AIMO all the high-level capabilities required for provisioning, configuration, monitoring and maintenance of domains' AIM DE pipelines including the datasets contained in the AIM Pipeline Catalogue and the AIM Resource Inventory.

This interface allows intent-based requests to flow to target Domain AIMOs from the E2E AIMO to fulfill AIM task requests received from the Customer Management layer.

Theses interfaces may be implemented either as an extension of the corresponding domain interface, Occo-Nf-sdn-x defined in TR-411 [7], when the Domain AIMO is integrated in the corresponding Domain SDN-C, or as a stand-alone interface.

Regardless of these two options the Oe2e-aimo-Ma-d-aimo interface is required to follow the functional requirements reported in the following sections.

7.2.1 Protocol requirements

The Oe2e-aimo-Ma-d-aimo interface requires a flexible protocol to support the programmability of the AIM pipelines as well as the monitoring of their status. The RESTful HTTP protocol is the preferable solution since it offers a simplified way of interaction between the E2E AIMO and Domain AIMOs.

A REST-based interface can easily be extended to support queries and the E2E AIMO can receive notifications for specific information elements.

[R-104]The Oe2e-aimo-Ma-d-aimo interface MUST be based on YAML/JSON or XML models and grammar to expose the AIM solution resources and services via primitive service elements.

[R-105]The Oe2e-aimo-Ma-d-aimo interface MUST be an HTTP and HTTPs REST-like interface.

[R-106]The Oe2e-aimo-Ma-d-aimo interface MUST be an intent-based interface.

[R-107]The Oe2e-aimo-Ma-d-aimo interface’s service elements API MUST be consumed via Create, Read, Update, Delete and Notifications (CRUD-N) actions between the E2E AIMO and the Domain AIMOs.

7.2.2 Requirements for the support of TM Forum APIs

As discussed in section 7, the Oe2e-aimo-Ma-d-aimo adopts the TM Forum APIs, notably:

- Service Inventory: TMF638 [21] offers E2E AIMO entities a standard means of query, updating and receiving notifications from the AIM Pipeline Catalogue including to AIM pipeline service state recorded in the catalogue.
- Resource Inventory: TMF639 [22] offers E2E AIMO entities a standard means of query, updating and receiving notifications from the AIM Resource Inventory.
- Service Catalog Retrieval: TMF633 [23] offers the E2E AIMO entities a standard means of querying the run-time AIM Pipeline Catalogue to determine the AIM pipeline service types available. In addition, TM633 [23] offers a complete set of CRUD-N capabilities to lifecycle manage the content of the run-time AIM Pipeline Catalogue.
- Service Fulfillment: TMF641 [24] offers the E2E AIMO entities a standard means of ordering services from the Domain AIMOs. Notifications of changes to the AIM pipeline intent order item(s) state and service instance are also offered. Error handling notifications are exposed northbound to the E2E AIMO entities. When used, TMF641 [24] exposes the NFVO functionality toward the E2E AIMO entities.
- Service Quality Management: TMF657 [25] specifies the API used to access a Service Quality Management application and extract Service Level Specifications and associated Service Level Objectives (SLO) and their thresholds. This can be used to monitor the SLA of a specific AIM pipeline service and generate alarms in case of a violation of threshold.
- Performance Thresholding TMF649 [26] offers the API to identify exceptional behavior of monitored performance indicators of an AIM pipeline service. If a threshold is crossed, the API generates an alarm.
- Service Level Agreement: TMF623 [27] specifies the SLA API to provide a standardized interface for AIM pipeline SLA life cycle management between the E2E AIMO and a Domain AIMO which provides AIM pipeline services with attached SLA in the shared AIM Pipeline Catalogue. The typical SLA LCM phases are: SLA configuration, SLA activation/enforcement, SLA operations, SLA violation/consequence handling, SLA reporting.

- Performance Management: TMF628 [28] specifies the REST API for Performance Management which provides a standardized mechanism for performance management such as the creation, partial or full update and retrieval of AIM pipeline resources involved in performance management
- Alarms: TMF642 [29] offers the API to manage alarms from monitored AIM pipeline resources. The API does not assume a particular management layer, so it can be either AIM pipeline resource or AIM pipeline service layer.
- Event Management: TMF688 [30] specifies the Event Management API to manage events that are coming from other TMF APIs. It uses a publish/subscribe pattern to manage messages between event producers and consumers. TMF688 [30] suggests the utilization of message brokers, such as Kafka or RabbitMQ, to implement this API.

[R-108]The Domain AIMO MUST provide the capability to support the TMF638 [21] Service Inventory REST API Specification across the Oe2e-aimo-Ma-d-aimo reference point.

[R-109]The Domain AIMO MUST provide the capability to support the TMF639 [22] Resource Inventory REST API Specification across the Oe2e-aimo-Ma-d-aimo reference point.

[R-110]The Domain AIMO MUST provide the capability to support the TMF633 [23] Service Catalog Retrieval REST API Specification across the Oe2e-aimo-Ma-d-aimo reference point.

[R-111]The Domain AIMO MUST provide the capability to support the Service Fulfillment: TMF641 [24] Service Fulfillment REST API Specification across the Oe2e-aimo-Ma-d-aimo reference point.

[R-112]The Domain AIMO MUST provide the capability to support the TMF657 [25] Service Quality Management REST API Specification across the Oe2e-aimo-Ma-d-aimo reference point.

[R-113]The Domain AIMO MUST provide the capability to support the TMF649 [26] Performance Thresholding REST API Specification across the Oe2e-aimo-Ma-d-aimo reference point.

[R-114]The Domain AIMO MUST provide the capability to support the TMF623 [27] Service Level Agreement REST API Specification across the Oe2e-aimo-Ma-d-aimo reference point.

[R-115]The Domain AIMO MUST provide the capability to support the TMF628 [28] Performance Management REST API Specification across the Oe2e-aimo-Ma-d-aimo reference point.

[R-116]The Domain AIMO MUST provide the capability to support the TMF642 [29] Alarm REST API Specification across the Oe2e-aimo-Ma-d-aimo reference point.

[R-117]The Domain AIMO MUST provide the capability to support the TMF688 [30] Event Management REST API Specification across the Oe2e-aimo-Ma-d-aimo reference point.

7.3. Oe2e-aimo-De and Od-aimo-De Interfaces

Oe2e-aimo-De and Od-aimo-De interfaces are the communication interfaces between the AIMOs and the AIM DE types they orchestrate:

- Oe2e-aimo-De: interface between the E2E AIMO and the E2E Pipelines
- Od-aimo-De: interface between a Domain AIMO and the Pipelines scoped within that Domain.

This section describes the protocol requirements and the information elements exposed by the interface, in particular regarding the requests, responses and notifications. The information elements are divided into tables that represent the components of AIM DE (CF, PPF, MF) as well as the associated tasks.

7.3.1. Protocol Requirements

[R-118]The Oe2e-aimo-De and Od-aimo-De interfaces MUST expose the information elements via the RESTCONF protocol defined in RFC 8040 [12].

[R-119]The information elements conveyed over the Oe2e-aimo-De and Od-aimo-De interfaces MUST be encoded in the YANG data modeling language defined in RFC 7950 [13] by exploiting either JSON or XML in the request payload.

[R-120]The Oe2e-aimo-De and Od aimo-De interfaces MUST use the Subscription capabilities of the RESTCONF protocol to send notifications.

[R-121]Requests and responses across the Oe2e-aimo-De and Od-aimo-De RESTCONF/YANG interfaces MUST be scoped to specific targets (e.g., Collection Function, Pre-Processing Function, or Model Function) and information elements or sub-information elements.

[R-122]The notifications of the Oe2e-aimo-De and Od-aimo-De interfaces MUST exploit the Subscription capabilities of the RESTCONF protocol.

7.3.2. Information Elements

As described in sections 6.1.4 and 6.2.4, the E2E and Domain AIMOs perform the following over the Des under their orchestration scope:

- Control
  - AIM Resource creation
  - AIM Components Configuration
  - Pipeline Intent Fulfillment
  - Pipeline Intent Check
  - Pipeline Intent Realization
  - Pipeline Policy Management
- Monitoring
  - AIM Component Fault, Security and Performance Events
  - Pipeline Intents Performance and Events
  - Pipeline Intent Fault and Security Events

The following information elements, related to the DE components (CF, PPF and MF) and their tasks (CT, PPT and DMT), are exposed on the Oe2e-aimo-De and Od-aimo-De interfaces.

Note: At this time of publication, YANG models are not defined for the information elements in Table 10 – Table 15. Further, these tables may be subject to revision.

Task information elements related to monitoring, performance and health may be aggregated under each corresponding function for reporting to the orchestrators.

[R-123] The Oe2e-aimo-De and Od-aimo-De interfaces MUST expose the CF Information Elements specified in Table 10.

Table 10: Collection Function Information Elements

Information element name	Sub-information element name	CRUD-N	Notes/description
CF-ID		CRD	The unique identifier of the deployed CF
Tenant-ID		CRD	Optional: maybe not necessary; the AIMO may take care of this information This parameter may have privacy and regulatory implications.
Infoset	Source-domain Information-type Collection-mode NF-Type Encoding-schema Collection-protocol	R	CFs specialised for: - multiple/single network domain(*source-domain)* *Note: multiple network domains listed in this parameter applies to a CF deployed in a E2E Pipeline* - type of info (real time data, PM, alarms)
Performance-Indicators	RunningCTs_∆time TerminatedCTs_∆time Percent of successful task actions in ∆ time	R and N	∆time: time interval over which the KPI is calculated. E.g.: for 1s and 15min intervals the info elements are named: "…_1s", "…_15min"
Alarms	Alarm-Type Alarm-Detail Timestamp	R and N	
Message-bus type		R	Message Bus type that the CF sends to (e.g., Kafka, GCP message queue, etc.)

[R-124]The Oe2e-aimo-De and Od-aimo-De interfaces MUST expose the Collection Task Information Elements specified in Table 11.

Table 11: Collection Task Information Elements

Information element name	Sub-information element name	CRUD-N	Notes/description
CT-ID		CRD	Unique identifier of the deployed CT, assigned by the CF
Schedule	Start Time Stop Time Duration Interval	CRUD	The CT State Machine is described in Figure 14.
Collection Task	Name	CRUD and N	The CT State Machine is described in Figure 14.

Information element name	Sub-information element name	CRUD-N	Notes/description
	Description [Opt.] Managed Entity Data Model(s) Data Path Output Endpoint Collection Interval Reporting Interval Max Data Mode Status		Reporting Interval represents the number of samples of Collection Intervals sent forward on the pipeline. When set to be greater than one, this reduces resource consumption.
Health-status	Condition Problem Type Problem Detail	R and N	Condition: reports the health of the CT (e.g., OK, Error, Warning, etc.) Problem Type describes the type of error, if any (e.g., Connection timeout to ME, Pipeline Error, authentication error, etc.) Problem detail reports the error logs.
Policies	AccessPolicy ControlManagementEnforcement	CRUD	

[R-125]The Oe2e-aimo-De and Od-aimo-De interfaces MUST expose the PPF Information Elements specified in Table 12.

Table 12: Pre-Processing Function Information Elements

Information element name	Sub-information element name	CRUD-N	Notes/description
PPF-ID		CRD	The unique identifier of the deployed PPF
Tenant-ID		CRD	Optional: maybe not necessary; the AIMO may take care of this information This parameter may have privacy and regulatory implications.
Infoset	Source-domain Information-type Data-Transformation-type NF-Type	R	PPF specialised for: - multiple/single network domain (*source-domain)* *Note: multiple network domains listed in this parameter applies to a PPF deployed in a E2E Pipeline*

Information element name	Sub-information element name	CRUD-N	Notes/description
	Data-model PreparationRecipe		- type of info (real time data, PM, alarms) *Note: Recipe is a term used in Google Cloud Dataprep by Trifacta*
Performance-Indicators	RunningPPTs_∆time TerminatedPPTs_∆time Percent of successful task actions in ∆ time	R and N	∆time: time interval over which the KPI is calculated. E.g.: for 1s and 15min intervals the info elements are named: "…_1s", "…_15min"
Alarms	Alarm-Type Alarm-Detail Timestamp	R and N	
Message-bus type		R	Message Bus type that the PPF sends to (e.g., Kafka, GCP message queue, etc.)

[R-126]The Oe2e-aimo-De and Od-aimo-De interfaces MUST expose the Pre-Processing Task Information Elements specified in Table 13.

Table 13: Pre-Processing Task Information Elements

Information element name	Sub-information element name	CRUD-N	Notes/description
PPT-ID		R	Unique identifier of the deployed PPT, assigned by the PPF
Pre-processing Task	Name Description [Opt.] Input Endpoint Input Data Model(s) Input Data Path Output Data Model(s) Output Endpoint Status	CRUD	The PPT State Machine is described in Figure 14.
Health-status	Condition Problem Type Problem Detail	R and N	Condition is used to report the health of the PPT (e.g., OK, Error, Warning, etc.) The Problem Type describes the type of error, if any (e.g., Pipeline Error, authentication error, etc.) Problem detail reports the error logs.

Information element name	Sub-information element name	CRUD-N	Notes/description
Policies	AccessPolicy ControlManagementEnforcement	CRUD	

[R-127]The Oe2e-aimo-De and Od-aimo-De interfaces MUST expose the Pre-Processing Task Information Elements specified in Table 14.

Table 14: Model Function Information Elements

Information element name	Sub-information element name	CRUD-N	Notes/description
MF-ID		CRD	The unique identifier of the deployed MF
Tenant-ID		CRD	Optional: maybe not necessary; the AIMO may take care of this information This parameter may have privacy and regulatory implications.
Infoset	Source-domain Information-type MF-Type AI-Type NF-Type Data-Model	R	MF specialised for: - multiple/single network domain (source-domain) *Note: multiple network domains listed in this parameter applies to a MF deployed in a E2E Pipeline* - type of info (real time data, PM, alarms) MF-Type e.g.: linear-regression, logistical-regression, neural network, etc. AI-Type e.g.: ML AI supervised, ML AI unsupervised, Rule-based AI,…
Performance-Indicators	RunningDMTs_Δtime TerminatedDMTs_Δtime Percent of successful task actions in Δ time	R and N	Δtime: time interval over which the KPI is calculated. E.g.: for 1s and 15min intervals the info elements are named: "…_1s", "…_15min"
Alarms	Alarm-Type Alarm-Detail Timestamp	R and N	
Message-bus type		R	Message Bus type that the MF sends to (e.g., Kafka, GCP message queue, etc.)

[R-128]The Oe2e-aimo-De and Od-aimo-De interfaces MUST expose the Decision-Making Task Information Elements specified in Table 15.

Table 15: Decision-Making Task Information Elements

Information element name	Sub-information element name	CRUD-N	Notes/description
DMT-ID		R	Unique identifier of the deployed DMT, assigned by the MF
Decision-Making Task	Name Description [Opt.] Input Endpoint Input Data Model(s) Input Data Path Output Data Model(s) Recommendation Endpoint Output Endpoint Status Policies	CRUD	The DMT State Machine is described in Figure 14. See Table 9 for definitions.
Health-status	Condition Problem Type Problem Details	R and N	Condition is used to report the health of the DMT (e.g., OK, Error, Warning, etc.) The Problem Type describes the type of error, if any (e.g., Pipeline Error, authentication error, AI error etc.) Problem details reports the error logs.
Policies	AccessPolicy ControlManagementEnforcement PipelineOutputAuthorization	CRUD	

7.3.3. Interactions

7.3.3.1. AIMO - CF interactions

Precondition: the CF has already been instantiated by the NFVO as a xNF, with related Day-0 and Day-1 configurations, and is available as an application/service in the SBA.

[R-129]The Oe2e-aimo-De and Od-aimo-De interfaces MUST support the AIMO-Collection Function (CF)/Collection Task (CT) interactions per the primitives defined in Table 16.

Table 16: AIMO-CF/CT interactions

Interaction type	Primitive
CF Initialization/Configuration	• CF Configuration/Ack • CF Start/Ack • CF Pause/Ak Note: certain re-configurations may require to Pause the whole CF
CF Monitoring	• CF Read/Ack • CF Notification • CF Performance Monitoring • CF Alarm Reporting
CF Deletion	• CF Stop/Ack • CF Delete/Ack
CT Creation/Configuration	• CT Creation/Ack Note: the creation includes the initial configuration of the CT • CT Configuration/Ack • CT Start/Ack Note: this interaction is conditioned to the way the Schedule attribute is configured • CT Pause/Ack Note: certain re-configurations may require to Pause the CT
CT Monitoring	• CT Read/Ack • CT Notification • CT Performance Monitoring • CT Alarm Reporting
CT Deletion	• CT Stop/Ack Note: this interaction is conditioned to the way the Schedule attribute is configured • CT Delete/Ack

7.3.3.2. AIMO - PPF interactions

Precondition: the PPF has already been instantiated by the NFVO, with related Day-0 and Day-1 configurations, as a xNF and is available as an application/service in the SBA

[R-130]The Oe2e-aimo-De and Od-aimo-De interfaces MUST support the AIMO-Pre-Processing Function (PPF)/Pre-Processing Task (PPT) interactions per the primitives defined in Table 17.

Table 17: AIMO-PPF/PPT interactions

Interaction type	**Primitive**
PPF Creation/Configuration	• PPT Creation/Ack Note: the creation includes the initial configuration of the PPT • PPF Configuration/Ack • PPF Start/Ack • PPF Pause/Ack Note: certain re-configurations may require to Pause the whole PPF
PPF Monitoring	• PPF Read/Ack • PPF Notification • PPF Performance Monitoring • PPF Alarm Reporting
PPF Deletion	• PPF Stop/Ack • PPF Delete/Ack
PPT Initialization/Configuration	• PPT Configuration/Ack • PPT Start/Ack • PPT Pause/Ack Note: certain re-configurations may require to Pause the CT
PPT Monitoring	• PPT Read/Ack • PPT Notification • PPT Performance Monitoring • PPT Alarm Reporting
PPT Deletion	• PPT Stop/Ack • PPT Delete/Ack

7.3.3.3. AIMO - MF interactions

Precondition: the MF has already been instantiated by the NFVO, with related Day-0 and Day-1 configurations, as a xNF and is available as an application/service in the SBA.

[R-131]Oe2e-aimo-De and Od-aimo-De interfaces MUST support the AIMO-Model Function (MF)/Decision-Making Task (DMT) interactions per the primitives defined in Table 18.

Table 18: AIMO-MF/DMT interactions

Interaction type	Primitive
MF Initialization/Configuration	• MF Configuration/Ack Note: the initialization/configuration of the MF may include also the update of the AI/ML or Rule-based model pulled by the Domain AIMO from the Sandbox • MF Start/Ack • MF Pause/Ack Note: certain re-configurations may require to Pause the whole MF
MF Monitoring	• MF Read/Ack • MF Notification • MF Performance Monitoring • MF Alarm Reporting
MF Deletion	• MF Stop/Ack • MF Delete/Ack
DMT Creation/Configuration	• DMT Creation/Ack Note: the creation includes the initial configuration of the DMT • DMT Configuration/Ack • DMT Start/Ack • DMT Pause/Ack Note: certain re-configurations may require to Pause the DMT
DMT Monitoring	• DMT Read/Ack • DMT Notification • DMT Performance Monitoring • DMT Alarm Reporting
DMT Deletion	• DMT Stop/Ack • DMT Delete/Ack

7.4. De-Nf-ccodo and De-Nf-sdn interfaces

De-Nf-ccodo and De-Nf-sdn are the communication interface between the AIM DE and the corresponding orchestration, control and management system in the SDN-NFV deployment that the AIM solution collaborates with and empowers. AIM DEs use these interfaces to send analyses results and reconfiguration recommendations:

- De-Nf-ccodo: interface between the MF of an E2E Pipeline and the CloudCO Domain Orchestrator (or the reference SDN-NFV orchestration system)

- De-Nf-sdn: interface between the MF of a Domain-specific Pipeline and the SDN-C of that Domain.

In the AIM DE, the CF and MF and their associated tasks (i.e., Collection Task and Decision-Making Task) can use this interface to collect data and to interact with MEs via the SDN-C.

The MF uses this interface to provide recommendations to the MEs via their Domain specific SDN-C; in case of the MF of an E2E Pipeline via the chain trough both the CCO DO and Domain specific SDN-C. The recommendations are in the form of intents. Depending on the objectives and operation of a given AIM pipeline, these recommendation intents may be expressed in an abstract, high-level indication of the new recommended status of the network resource(s) or, instead, up to a quite detailed configuration.

These recommendation intents are received by the SDN-C or the CCO DO. These are SDN intents which end up being applied to MEs in the form of an imperative configuration. The CCO DO, if recipient of the recommendations from an E2E Pipeline, translates them into Domain-specific intents sent to the involved SDN-C which in turn act upon the target MEs.
The MF must have all the necessary information about network resources to generate an SDN intent for the SDN-C or the CCO DO. The SDN intents are different from the AIM intents from the Customer Management Layer to the E2E AIMO (via the Cm-Ma-e2e-aimo interface) and from the E2E AIMO and the Domain AIMO (via the Oe2e-aimo-Ma-d-aimo interface).

The MF’s recommendation intents provide a declarative input to the SDN-C or the CCO DO and are similar to the RFS intents described in TR-411 [7] in that they request a declarative configuration for target resources.

Like any SDN-based architecture, the MF shall not send direct, imperative inputs to the MEs.
This is to:

- avoid commercial MFs, developed to be widely reusable, to directly configure the network resources, which would:
  - keep the SDN-C or the chain of CCO DO and SDN-Cs out of the loop and possibly conflict with policies and configurations applicable to the SDN-C or the CCO DO
  - impose on MFs the burden of adapting their SBIs to the quite varied NBIs of the MEs
- take advantage of the already existing chain of CCO DO SBI and Domain SDN-C SBIs towards network resources avoiding the above MF dependencies
- take advantage of the existing NBIs of Domain SDN-Cs (i.e., the SDN M&C systems) and CCO DO specified in TR-411 [7] in the form of a quite flexible intent-based interface
- not violate the principle that one single element in the SDN hierarchy interacts imperatively with the network, especially in configuration mode (point-multipoint monitoring and reading of the network is less critical)

The approach of always having a single SDN system directly interacting with the network guarantees that the SDN-C (or the CCO DO on top of it) acts as the center point of any declarative requests. The SDN-C that has full visibility of the domain resources exposed in a NaaS fashion, that can apply brokerage and resource priority policies and has the ability to fulfill concurrent declarative intents by the realization of imperative commands while preserving the overall set of Domain SLAs and priorities. This:

1) prevents conflicts in concurrent configuration of a resource or anyway piped configurations (using the lock/unlock of the Netconf Datastore) that would be uncoordinated
2) guarantees that the SDN-C, which hosts the single source of truth, is the center point of declarative requests that are fed by the SDN-C to the network while the SDN-C itself updates its local Datastore
3) avoids that the MF optimization logic, applied, e.g. to a single subscriber, does optimize that service to the detriment of the overall equipment and network configuration, not necessarily taken into consideration by that MF’s pipeline (e.g. in a multi-tenant scenario)

As an alternative to using the De-Me interface for ’Es' data collection, the CF can exploit the De-Nf-sdn interface for collecting non-massive data of MEs (e.g., alarms, notifications). This interface can also be used to collect PM and real-time data over their specific network domains with the advantage of exposing a unique interface to access different network resources in the domain.

However, this approach would require Domain SDN-C elements to pass collected data from MEs to a CF. The implications on their load, operations, performance degradation on the SDN-C elements and data collection should be carefully evaluated.

7.4.1. Protocol and Data models Requirements

The De-Nf-ccodo and De-Nf-sdn interfaces reuse the protocols and data models defined respectively for the Os-Ma-ccodo and the Occo-Nf-sdn-x interfaces in TR-411 [7] and other Broadband Forum specifications, as detailed in the requirements below. These interfaces offer monitoring capabilities and performance metrics of RFS, as well as the definition and update of services via RFS Intents.

CloudCO cross-domain scope

[R-132] For end-to-end Pipelines, the De-Nf-ccodo interface's protocols and data models MUST comply with the interface requirements specified for Os-Ma-ccodo interface in TR-411 [7].

Access Domains

[R-133] For Access domain-specific Pipelines, the De-Nf-sdn interface's protocols and data models MUST comply with the interface requirements specified for Occo-Nf-sdn-access interface and generally for a Occo-Nf-sdn-x interface in TR-411 [7].

Edge Domains

[R-134]For Edge domain-specific Pipelines, the De-Nf-sdn interface's protocols and data models MUST comply with the interface requirements specified for Occo-Nf-sdn-edge and generally for a Occo-Nf-sdn-x interface in TR-411 [7].

Home Customer Premises domain

[R-135] For Home domain-specific Pipelines, the De-Nf-sdn interface's protocol MUST comply with the requirements specified at for interfacing with a USP Controller (see also The User Services Platform and refer to TR-369 [5]).

[R-136] For Home domain-specific Pipelines to interface with a CWMP (aka TR-069 [1]) ACS, the De-Nf-sdn interface's protocol MUST comply with the related northbound interface.

Note: TR-369 [5], referred to as Universal Services Platform (USP), is the next generation of CPE remote management protocols developed out of deployment experience. It is thus recommended that TR-369 [5] be considered for new deployments while TR-069 [1] be the domain of legacy deployments.

[R-137] The De-Nf-sdn interface's data models MUST comply with TR-181 [3].

7.4.2. RFS Intent Information Elements for CF and MF

As described above, the RFS intent generic model defined in TR-411 [7] can be used to send declarative requests to the MEs, via the Domain SDN-C; in case of the MF of an E2E Pipeline via the chain CCO DO/Domain specific SDN-C.

This intent-based model is composed of:

- *IntentDescription* that contains policies for the fulfillment (*ServiceLevelObjective*) and assurance (*ServiceAssurancePolicy*) of the intent
- *ServiceTarget* is the target of the RFS intent, i.e. *Nodes*, *Links*, and *Endpoints*
- A *ServiceAssurancePolicy* is associated with *Endpoints*, *Links*, and *ServiceLevelObjective* to provide the appropriate monitoring policy

The CF and MF use the RFS intents as follows:

[R-138]Referring to the generic RFS model defined in TR-411 the CF MUST use the capabilities of RFS intent monitoring to collect data from its MEs. It MUST be able to receive notifications or to poll the CCO DO or the SDN-C to get data about a specific ME that are part of the ServiceTarget based on the associated ServiceAssurancePolicy.

[R-139]Referring to the generic RFS model defined in TR-411 [7], the E2E or domain-scoped MF MUST generate and send a recommendation in the form of an RFS intent, e.g., with:

- *IntentDescription* of the network resources configuration already provisioned by the CCO DO or the SDN-C. The MF can update the relevant information elements of an RFS Intent, such as *ServiceLevelObjective* and *ServiceTarget* and the associated *ServiceAssurancePolicy*.
- *IntentDescription* providing a very detailed configuration focused on a specific set of parameters, though still representing a declarative recommendation submitted to the brokerage and policy-aware decision of the CCO DO or the SDN-C.

[R-140]For creating updated RFS intents, the MF MUST:

- be aware of the Topology and Inventory information of the associated cross-domain scope or domain scope
- be able access the CCO DO or SDN-C RFS intents Catalog.

7.5. De-Me interface

The De-Me interface is mainly used to collect data (e.g., PM, alarms, real time information, notifications) from the Managed Entities to the Collection Function in the DE which is the first component in the AIM pipeline.

For this reason, this interface needs to adapt to the standard interfaces specified for network resources, and eventually for other sources of data that are fed into the AIM pipeline.

When a proprietary data model is used to represent the data collected from the ME, the Collection Function, before sending it to a destination, normalizes the data to a standard YANG data model for the specific type of data, regardless of the collection mechanism and protocol used. The possible collection protocols usable to implement the collection modes supported by AIM are reported in Table 6.

7.5.1. Protocol and Data Model Requirements

The De-Me interface needs to align with the protocols and data models of the ME interfaces specified for a given network domain and network resources the MEs belong to.

Access domain

[R-141]For Access domain-specific Pipelines, the De-Me interface's protocols and data models MUST comply with:

- The northbound interfaces Minf-AN-type specified for Access PNFs in WT-413i2 [32] and TR-435 [8] for data collection from these network resources; note: AN-type is one of the Access PNFs specified in WT-413i2 [32].

If the Access PNFs' MEs implement data models not compliant with those standards, it is the responsibility of the Collection Function to normalize the data to the standard YANG data model of the applicable Minf-AN-type specified for Access PNFs in WT-413i2 [32] by using an adapter.

[R-142]For Access domain-specific Pipelines, the De-Me interface's protocols and data models MUST comply with the northbound interfaces BAA NAI Minf-AN-type_L1 and Minf-L2-3 specified for the BAA Layer in WT-413i2 [32] and TR-435 [8] for data collection from these network resources; note: AN-type is one of the Access PNFs specified in WT-413i2 [32].

[R-143] For Access domain-specific Pipelines, the De-Me interface's protocols and data models MUST comply with the northbound interfaces Minf specified for D-OLTs in WT-477 [33] and TR-435 [8] for data collection from these network resources.

[R-144] For Access domain-specific Pipelines, the De-Me interface's protocols and data models MUST comply with the northbound interface MVOLTMF-VOMCI specified for vOMCI functions in WT-451 [10] for data collection from these network resources.

Home Customer Premises domain

[R-145] For Home domain-specific Pipelines, the De-Me interface's protocols and data models MUST comply:

- For new deployments, with TR-369 [5] for the CPE protocols and for bulk data collection
- For legacy CWMP-based deployments, with TR-069 [1] and possibly in combination with TR-232 [4] for the CPE protocols and for bulk data collection.
- With TR-181 [3] for the CPE data models.

Note: TR-369 [5], referred to as User Services Platform (USP), is the next generation of CPE remote management protocols developed out of deployment experience. It is thus recommended that TR-369 [5] be considered for new deployments while TR-069 [1] be the domain of legacy deployments.

Furthermore the USP framework allows the lifecycle management of software modules running on the Residential Gateway’s execution environment thus allowing the ongoing trend of AIM Pipelines onboard of Network Devices.

As an alternative option to collecting data via the De-Me interface, the Collection Function MAY use the De-Nf-sdn interface to Domain SDN-C elements (defined in section 7.4).

7.6. De-Mb interface

The De-Mb interface represents the connection point within the AIM DE components (the CF, PPF, and MF), and the Knowledge base. In general, this interface is implemented as a message broker, which exploits a publish/subscribe approach to distribute messages between different components. A publisher is a component that makes a message available on the message broker, whereas a subscriber is a component that receives the message. Usually, the messages are grouped via labels named *message subjects* (or topics) providing the isolation, security, and performance control and management. Examples of message brokers are Kafka, RabbitMQ, or proprietary services provided by hyperscalers, such as Amazon MQ, Google Pub/Sub, etc.

This De-Mb interface may also carry a standard client/server interaction between the AIM DE components and KB to access specific data, such as historical data from a time series database. This section describes the De-Mb interface and protocol requirements.

7.6.1. Interface requirements

The De-Mb interface carries diverse type of information elements either as message broker or via a direct client/server interaction between the AIM DE components and the KB. It is not possible and needed to define standard information elements. The specific implementation of AIM DE components (CF, PPF, and MF) defines which data are going to be transmitted between the components, and between the components and the Knowledge base.

The following requirements apply to the De-Mb interface:

[R-146] The AIM DE components are data publishers and subscribers on the message broker, and they MUST have access to the Knowledge Base.

[R-147] The AIM DE components MAY have client/server access to the KB.

[R-148] The De-Mb interface MUST use JSON or XML data formats.

[R-149] The De-Mb interface MUST provide the capability of publishing data messages on the message bus and label them with one or more message subjects.

[R-150] The De-Mb interface MUST provide isolation, security and performance of message exchanges between specific groups of senders and receivers.

[R-151] The De-Mb interface MUST provide authentication and authorization capabilities.

[R-152] The De-Mb interface MUST provide the capability of subscribing to messages on the message bus on a particular message subject.

[R-153] The De-Mb interface MUST provide the capability to notify entities that have subscribed to a specific message subject.

End of Broadband Forum Technical Report TR-486