ETSI GR NFV-IFA 023 V3.1.1 (2017-07)

Network Functions Virtualisation (NFV); Management and Orchestration; Report on Policy Management in MANO; Release 3

Disclaimer

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

Reference
DGR/NFV-IFA023

Keywords
MANO, NFV, policy management

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document provides a study on managing policies in the NFV MANO architecture. High-level use cases, key functionality analysis and potential architectural options for fulfilling this objective are proposed in the present document. As an output of the study, recommendations for the normative work are provided.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

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

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

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

[i.1] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for main concepts in NFV".

[i.2] ETSI GS NFV-MAN 001: "Network Functions Virtualisation (NFV); Management and Orchestration".

[i.3] ETSI GS NFV-IFA 007: "Network Functions Virtualisation (NFV); Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".

[i.4] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV); Management and Orchestration; VNF Packaging Specification".

[i.5] TM Forum TR235: "Report on ZOOM policy model and architecture snapshot".

[i.6] IETF draft-ietf-supa-policy-based-management-framework-00: "SUPA policy-based management framework".

[i.7] OpenStack® Releases.

NOTE: Available at https://releases.openstack.org/teams/congress.html.

[i.8] ETSI GS NFV-IFA 014: "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Network Service Templates Specification".

[i.9] IETF draft-ietf-supa-generic-policy-info-model-02: "Generic Policy Information Model for Simplified Use of Policy Abstractions (SUPA)".

[i.10] IETF draft-ietf-supa-generic-policy-data-model-03: "Generic Policy Data Model for Simplified Use of Policy Abstractions (SUPA)".

[i.11] IETF RFC 3060: "Policy Core Information Model -- Version 1 Specification".

[i.12] IETF RFC 3460: "Policy Core Information Model (PCIM) Extensions".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI GS NFV 003 [i.1] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS NFV 003 [i.1] and the following apply.

NOTE: An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in ETSI GS NFV 003 [i.1].

PAP Policy Administration Point
PF Policy Function

4 Background and Overview

Policy is one of the key enablers for constructing flexible management and orchestration functions in the NFV MANO architecture. Assisted with policies, MANO functions may be provided with more automatic characteristics which fit in with the dynamic requirements of resource management and network service orchestration in the virtualized network environment.

Although the importance of policy management in MANO has been identified by ETSI GS NFV-MAN 001 [i.2], there is not a systematic study on how to integrate policy management in NFV MANO architecture during the timeframe of releases 1 and 2. Following a model consisting of a Policy Administration Point (PAP), which defines policies, and a Policy Function (PF), which evaluates policies and makes the policy decisions, the following aspects are to be investigated in the present document:

- The model of PAP and PF can be applied to the management of MANO policy. Particularly, a MANO entity such as the NFVO can act as both the PAP and the PF (i.e. acting as the former on one policy and as the latter on another policy).
- There can be multiple sources that create policy information enforced by a PF, e.g. the policy information on scaling for a certain VNF instance can be created by either the OSS/BSS, the NFVO or be pre-defined in the VNF descriptor.
- Potential conflicts of policy enforcement can occur during the enforcement of MANO policies distributed in different layers.
- From management point of view, MANO policy can apply to lifecycle management and resource orchestration.

Therefore, the study in the present document is targeted to deliver recommendations on a functional framework of policy management with the analysis on the above aspects.

5 Use cases

5.1 General

The following use cases describe the processes involving MANO policy management. In clauses 5.2 to 5.7, rather than the method of designating assigned management functions to one or more MANO functional block(s), the abstracted roles are introduced (see table 5.1-1) in the use case description for improving the flexibility of functionality mapping to MANO architecture. These roles capture the generic processes as well as the action required to be performed by different actors. How does each role be mapped to the actual MANO functional block(s) is addressed under clause 6 of key issue analysis.

Table 5.1-1: Use case roles and description

Role	Description
Policy Administration Point (PAP)	The PAP is responsible for defining MANO policies. It is not in the scope of this document to define a jurisdiction (one administrative domain or multiple administrative domains) for a PAP.
Policy Function (PF)	The PF is responsible for making policy decisions, i.e. to evaluate whether the policy conditions are met and to define what is to be done when they are met. These responsibilities are referred to as "policy enforcement" in the presentdocument.
NOTE: Both the PAP and the PF can be mapped to MANO functional blocks (e.g. NFVO, VNFM, VIM) in different ways.	

Clause 5.8 provides NFV MANO specific use cases which are applied to specific VNF/NS lifecycle management or resource management scenarios. Finally, clause 5.9 proposes the recommendation of potential requirements which are derived from the use cases.

5.2 Policy creation and transfer

5.2.1 Use case description

This use case describes the process of creating a MANO policy, which is initiated by the PAP, along with the consequent process of transferring the created MANO policy to the PF for storage. It is assumed that policy creation occurs only inside the PAP and is not exposed to the PF.

5.2.2 Actors and conditions

Table 5.2.2-1: Actors

#	Actor
1	PAP
2	PF

Table 5.2.2-2: Pre-conditions

#	Pre-conditions	Comment
1	The PAP knows about the MANO functional entities and the necessary information for creating the MANO policy.	

Table 5.2.2-3: Post-conditions

#	Post-conditions	Comment
1	The PAP is aware of the acceptance of created MANO policy.	

5.2.3 Base flow

Table 5.2.3-1: Base flow

#	Actor	Action/Description
1	PAP	The PAP creates a certain MANO policy (e.g. NS LCM policy, VNF LCM policy, resource management policy or placement policy, etc.) and transfers the policy to the PF for acceptance.
2	PF	The PF checks the format and the parameters in the policy information and responds to the PAP with a response of acceptance. The PF stores the created MANO policy in its local database.

5.3 Policy deletion

5.3.1 Use case description

This use case describes the process of deleting one or multiple MANO policy(ies), which is initiated by the PAP, along with its consequent interaction with the PF.

5.3.2 Actors and conditions

Table 5.3.2-1: Actors

#	Actor
1	PAP
2	PF

Table 5.3.2-2: Pre-conditions

#	Pre-conditions	Comment
1	The target policy(ies) has/have been delivered to the PF.	

Table 5.3.2-3: Post-conditions

#	Post-conditions	Comment
1	The PAP is aware of the status of policy deletion.	

5.3.3 Base flow

Table 5.3.3-1: Base flow

#	Actor	Action/Description
1	PAP	The PAP initiates a deletion request for one or multiple policies, and sends the request to the PF.
2	PF	The PF checks the usage status of the policy(ies) to be deleted. The policy(ies) is/are deleted from the database of the PF. The PF notifies the PAP about the policy deletion.

5.4 Policy update

5.4.1 Use case description

This use case describes the process of updating a MANO policy, which is initiated by the PAP, along with its consequent interaction with the PF. Policy update can also be a transactional representation of deactivation, deletion, transfer and activation of the policy.

5.4.2 Actors and conditions

Table 5.4.2-1: Actors

#	Actor
1	PAP
2	PF

Table 5.4.2-2: Pre-conditions

#	Pre-conditions	Comment
1	The target policy has been delivered to the PF.	

Table 5.4.2-3: Post-conditions

#	Post-conditions	Comment
1	The PAP is aware of the acceptance of updated MANO policy.	

5.4.3 Base flow

Table 5.4.3-1: Base flow

#	Actor	Action/Description
1	PAP	The PAP updates a certain MANO policy and transfers the updated policy to the PF for acceptance.
2	PF	The PF checks the format and the parameters in the policy and responds to the PAP with a response of acceptance. (See note)
NOTE: The updated MANO policy may need to be activated in order to become enabled in the PF.		

5.5 Policy query

5.5.1 Use case description

This use case describes the process of querying a MANO policy, which is initiated by the PAP, along with its consequent interaction with the PF.

5.5.2 Actors and conditions

Table 5.5.2-1: Actors

#	Actor
1	PAP
2	PF

Table 5.5.2-2: Pre-conditions

#	Pre-conditions	Comment
1	The target policy has been delivered to the PF.	

Table 5.5.2-3: Post-conditions

#	Post-conditions	Comment
1	The PAP receives the queried MANO policy.	

5.5.3 Base flow

Table 5.5.3-1: Base flow

#	Actor	Action/Description
1	PAP	The PAP sends a request of querying MANO policy (e.g. the information of conditions or actions contained by the MANO policy) to the PF.
2	PF	The PF checks availability of the queried policy and responds to the PAP with the details of policy.

5.6 Policy activation

5.6.1 Use case description

This use case describes the process of activating a MANO policy, which is initiated by the PAP, along with its consequent interaction with the PF. This use case assumes that the policy has an administrative state in the PF and requires explicit activation. The policy activation can also be implicitly performed when the PAP pushes the policy to the PF.

5.6.2 Actors and conditions

Table 5.6.2-1: Actors

#	Actor
1	PAP
2	PF

Table 5.6.2-2: Pre-conditions

#	Pre-conditions	Comment
1	The target policy(ies) has/have been disabled or have not been enabled yet in the PF.	

Table 5.6.2-3: Post-conditions

#	Post-conditions	Comment
1	The target policy(ies) has/have been enabled in the PF.	

5.6.3 Base flow

Table 5.6.3-1: Base flow

#	Actor	Action/Description
1	PAP	The PAP enables the policy to the PF for acceptance.
2	PF	The policy(ies) is/are enabled at the PF. The PF notifies the PAP about the policy status change result.

5.7 Policy deactivation

5.7.1 Use case description

This use case describes the process of deactivating a MANO policy, which is initiated by the PAP, along with its consequent interaction with the PF. This use case assumes that the policy has an administrative state in the PF and requires explicit activation and deactivation.

5.7.2 Actors and conditions

Table 5.7.2-1: Actors

#	Actor
1	PAP
2	PF

Table 5.7.2-2: Pre-conditions

#	Pre-conditions	Comment
1	The target policy(ies) has/have been enabled in the PF.	

Table 5.7.2-3: Post-conditions

#	Post-conditions	Comment
1	The target policy(ies) has/have been disabled in the PF.	

5.7.3 Base flow

Table 5.7.3-1: Base flow

#	Actor	Action/Description
1	PAP	The PAP disables the policy to the PF.
2	PF	The policy(ies) is/are disabled at the PF. The PF notifies the PAP about the policy status change result.

5.8 NFV MANO specific policy management use cases

5.8.1 Overview

In this clause, several detailed use cases on applying policies in NFV MANO functionality are defined. It is not intended to enumerate all the potential use cases of policies applied in MANO functional blocks, but to illustrate how MANO policy is applied to specific VNF/NS lifecycle management or resource management scenarios.

5.8.2 Granting policy in VNF lifecycle management operation

5.8.2.1 Use case description

This use case describes the process of applying granting policy in the VNF LifeCycle Management (LCM) operations. Granting policy is a kind of MANO policy enforced by the NFVO, to assist for the NFVO on determining to approve or reject the grant request from the VNFM.

Before applying the granting policy, there is no differentiation in handling for the NFVO to respond to the grant requests for VNF instances which are managed by the VNFM. The NFVO checks the resource availability of the NS instance that one VNF instance belongs to, as well as the impacts on the NS instance that the VNF instance belongs to, and then determines to approve or reject the grant request after comparing the available resources with the requested resources in the request message.

The service provider plans to change this situation, for instance to reduce the rejection possibility of grant requests for certain types of VNFs in case of resource contention. Therefore, the service provider creates the granting policy and makes it into effect in the NFVO. In the policy design phase, the designer (belonging to the organization of the service provider) determines the information of granting policy for some given types of VNFs. The semantics of the granting policy may vary with the requirement changes of the service provider, and some examples are listed to illustrate what information is contained by the granting policy:

- the granting policy may indicate for some given types of VNF, the grant request is always permitted (see note);
- the granting policy may indicate for the VNF instances belonging to certain NS types, the grant request is always permitted (see note);
- the granting policy may allow to accept the grant request for the given type of VNF when some conditions are met, e.g. when the usage ratio of the compute resource is under a predefined threshold (see note).
- the granting policy may indicate for each VNFM, one of the following resource reservation mode can be selected: GRANT_RESERVE_MULTI, GRANT_RESERVE_SINGLE or GRANT_APPROVE (as specified in clause 6.3.2.1 of ETSI GS NFV-IFA 007 [i.3]) (see note).

NOTE: The NFVO considers in the granting operation both the granting policy and the availability of resources.

The service provider may further validate whether the newly created granting policy is in conflict with the existing policies in a centralized policy database, this step may also include the comparison with pre-defined policies in the NSD. After that, the granting policy is transferred to the NFVO. The NFVO stores the policy in its local policy database and activates it before the instantiation of VNF belonging to the given types. The NFVO may also subscribe to the notifications of consumed capacity for the given types of VNFs which will be monitored as whether the condition of the policy is met.

When the given type VNF is initially instantiated and during the subsequent lifecycle management operations like VNF scaling or VNF healing, the NFVO enforces the granting policy when it receives the grant request from the VNFM. To enforce a granting policy with the condition, the NFVO monitors the consumed capacity for the given types of VNFs and determine whether the condition of the granting policy is fulfilled. The decision of grant response (approve or reject) may be changed by applying the policy, compared with the case that does not apply the policy.

By applying the policy in the Grant VNF lifecycle operation procedure, the service provider possesses a more flexible way in influencing the enforcement result of the management functions in the NFVO, and makes the result more consistent with the high-level requirements reflected by the policy.

5.8.3 Optimization of NFVI-PoPs resource utilization

5.8.3.1 Use case description

NFVI-PoPs' resource utilization level vary over time due to a number of reasons: e.g. new VNFs are instantiated while some of the existing ones are either scaled or terminated.

It is reasonabe assuming that an Operator would pursue to maintain an optimal resource utilization level across its own infrastructure. Irrespective of the connotation each Operator may assign to the term "optimal", such intent is likely to be implemented in practice by sampling the instantaneous NFVI-PoPs resource utilization level at a given rate and, based on the observations, migrating one or more VNFs from one NFVI-PoP to another.

At least, two resource utilization criteria would be applicable:

1) **Load balancing**, see figure 5.8.3.1-1. If this resource utilization criterion is chosen then the Operator desires to maintain a uniform resource utilization level across its own infrastructure. Either instantaneous observations or forecasts based on historic trends can be exploited in order to select the NFVI-PoPs to be affected by the VNFs migration process. In the former case, a suitable approach would be to migrate one or more VNFs from the currently most-loaded NFVI-PoP to the currently least-loaded NFVI-PoP; in the latter case, a possibility would be to migrate one or more VNFs from the NFVI-PoP which is going to be the most-loaded one within a given time T to the NFVI-PoP which is going to be the least-loaded one within time T.

2) **Consolidation**, see figure 5.8.3.1-2. If this resource utilization criterion is chosen then the Operator desires to maximize the resource utilization level of one NFVI-PoP, thus mitigating the load on the others. A portion or the totality of an unloaded NFVI-PoP could then be switched off in order to obtain energy savings. When maximizing the resource utilization level of a given NFVI-PoP, the proper operation of the VNFs deployed on the NFVI-PoP itself is not expected to be compromised. Again, either instantaneous observations or forecasts based on historic trends can be exploited in order to select the NFVI-PoPs to be affected by the VNFs migration process. In the former case, a suitable approach would be to migrate one or more VNFs from the currently least-loaded NFVI-PoP to the currently most-loaded NFVI-PoP; in the latter case, a possibility would be to migrate one or more VNFs from the NFVI-PoP which is going to be the least-loaded one within a given time T to the NFVI-PoP which is going to be the most-loaded one within time T.

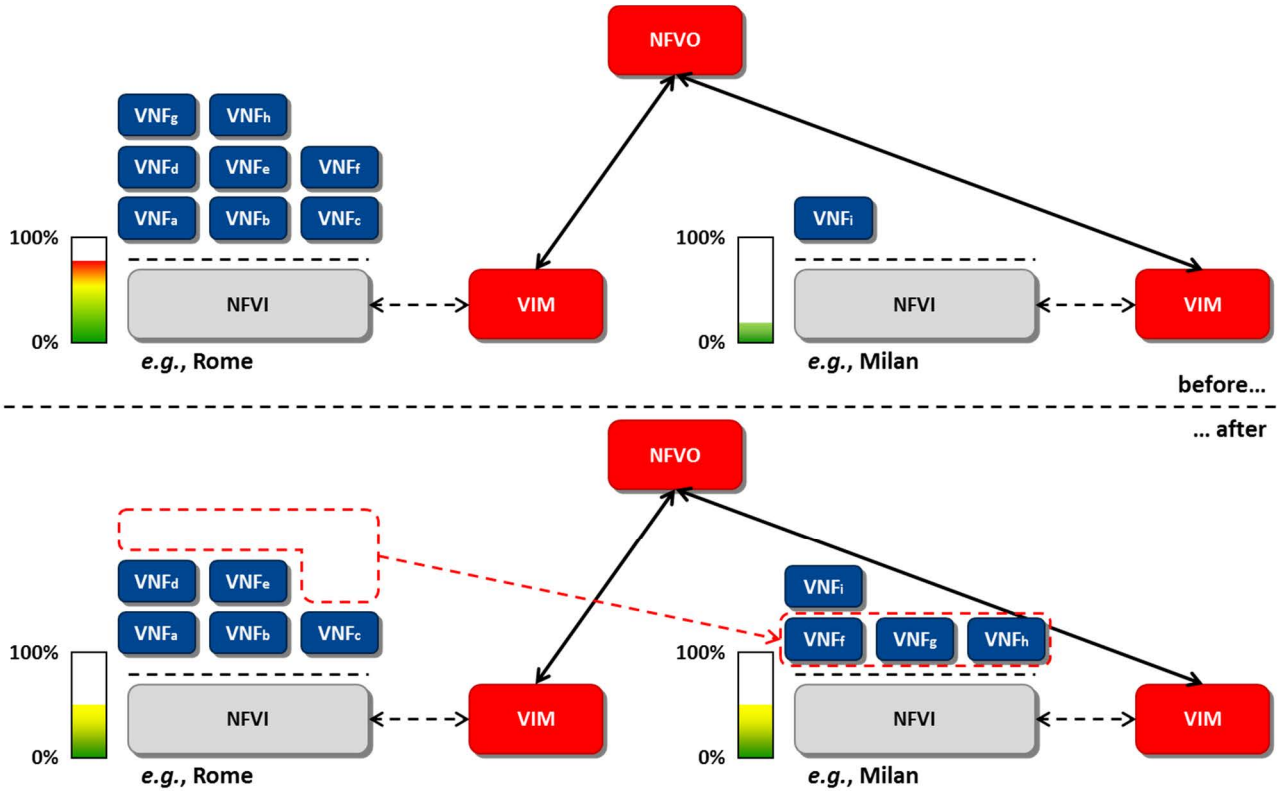

Figure 5.8.3.1-1: Load balancing resource utilization criterion

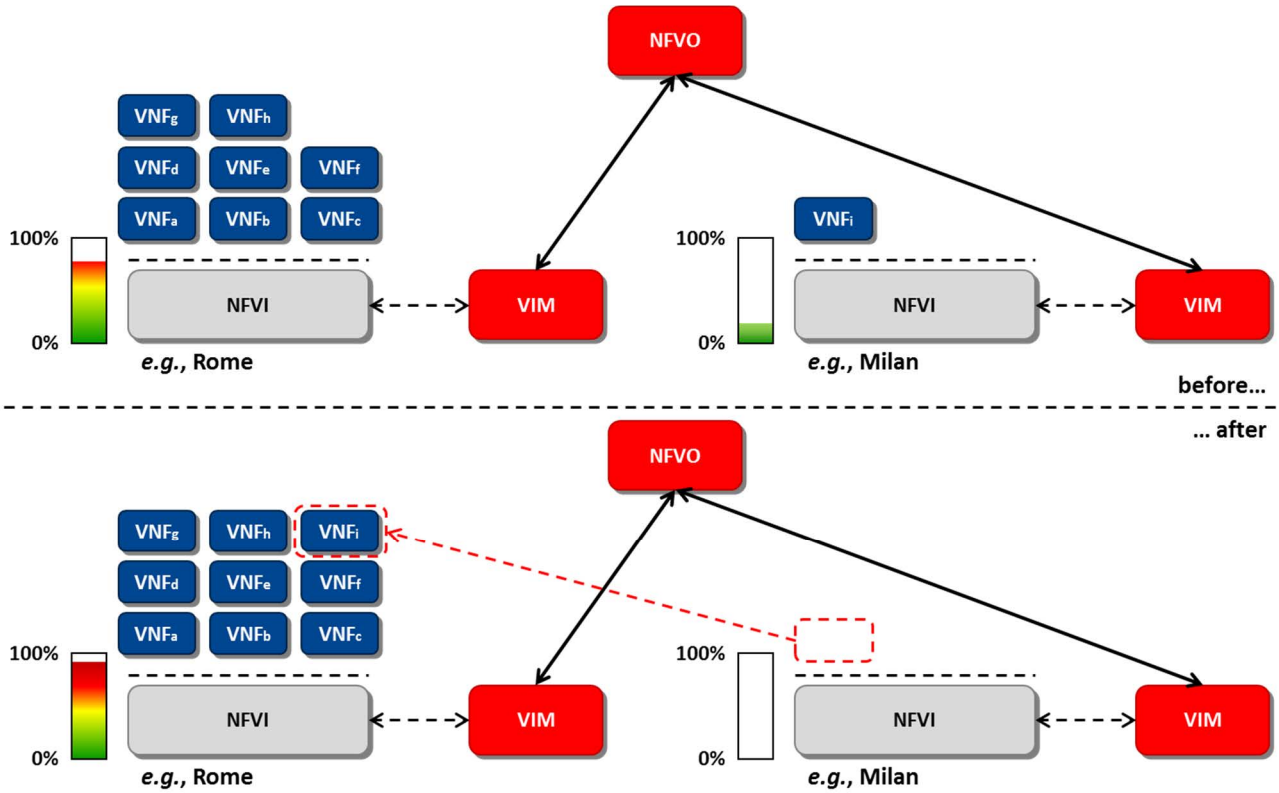

Figure 5.8.3.1-2: Consolidation resource utilization criterion

Policies might affect several aspects of the above description, including but not limited to:

- selection of the overall resource utilization criterion (e.g. load balancing vs. consolidation);

- selection of the sampling rate used to gather instantaneous resource utilization levels;
- selection of the window size T adopted for the prediction, when historic trends are to be leveraged;
- selection of the VNFs to be migrated (e.g. non-mission critical VNFs, third-party VNFs, etc.).

NOTE 1: The information contained in the descriptors (e.g. affinity/anti-affinity rules) are expected to take precedence over the policies governing the aforementioned behaviour.

NOTE 2: The operation Migrate VNF is not supported by NFV Release 2; this use case aims at highlighting the potential value of such operation.

In the following, the actors relevant for this use case and the role(s) they play among the ones described in clause 5.1 are identified. Additionally, a base flow showing how a policy in line with the above description might affect the run-time behaviour of MANO functional blocks is presented along with its corresponding pre- and post-conditions.

It is worth mentioning that the base flows described in clauses 5.2, 5.3, 5.4, 5.5, 5.6 and 5.7 are also applicable to this use case.

5.8.3.2 Actors and roles

Table 5.8.3.2-1 describes the actors relevant for this use case and the role(s) they play among the ones identified in clause 5.1.

Table 5.8.3.2-1: Optimization of NFVI-PoPs resource utilization actors and roles

#	Actor	Role(s)	Description
1	Policy Designer	N/A (see note)	Operator's function responsible for defining policies.
2	NFVO	PAP, PF (see note)	The Network Function Virtualisation Orchestrator, as defined in [i.1].
3	VNFM	N.A.	The Virtualised Network Function Manager, as defined in [i.1].
NOTE: In this use case, the PF role is played by the NFVO functional block. In addition, assuming that: 1) the Policy Designer leverages on a tool providing technical means (e.g. graphic editor, wizard, etc.) to ease the definition of policies; and 2) such tool is embedded in the product implementing the NFVO functionalities; then the PAP role is played by the NFVO functional block as well. However, scenarios where the PAP role is played by the OSS/BSS functional block are not prevented.			

5.8.3.3 Base flow

Table 5.8.3.3-1 describes the pre-conditions relevant for this use case.

Table 5.8.3.3-1: Optimization of NFVI-PoPs resource utilization pre-conditions

#	Pre-conditions	Comment
1	A sub-optimal NFVI-PoPs resource utilization level is detected.	N.A.

Table 5.8.3.3-2 describes the post-conditions relevant for this use case.

Table 5.8.3.3-2: Optimization of NFVI-PoPs resource utilization post-conditions

#	Post-conditions	Comment
1	The optimal NFVI-PoPs resource utilization level is re-established.	N.A.

Table 5.8.3.3-3 describes the base flow relevant for this use case.

Table 5.8.3.3-3: Optimization of NFVI-PoPs resource utilization base flow

#	Actor	Action/Description
1	NFVO	The NFVO monitors instantaneous NFVI-PoPs resource utilization level applying the sampling rate dictated by the policy.
2	NFVO	Based on the observations, the NFVO detects a sub-optimal NFVI-PoPs resource utilization level which violates the overall resource utilization criterion dictated by the policy (e.g. load balancing, consolidation, etc.).
3	NFVO	According to the selection method dictated by the policy (e.g. non-mission critical VNFs, third-party VNFs, etc.), the NFVO identifies the VNF(s) to be migrated, the corresponding VNFM(s) and the target NFVI-PoP.
4	NFVO	The NFVO triggers the VNF migration process by issuing the corresponding request towards the relevant VNFM. See notes 1 and 2.
5	VNFM	The VNFM performs the VNF migration procedure according to the directives provided by the NFVO (e.g. affected VNF instances, target NFVI-PoP, etc.). See note 1.
NOTE 1: This step is to be repeated for each VNF identified by the NFVO in step 3. NOTE 2: Assessing and addressing potential impacts of the VNF migration process on service continuity is outside the scope of this use case.		

5.8.4 Transferring NS healing policy to the NFVO

5.8.4.1 Use case description

This use case describes the process of transferring NS healing policy for a NS instance. In this use case, NFVO as the PF enforces the NS healing policy (i.e. evaluates whether the policy conditions are met and define what is to be done) created by the OSS/BSS, who is the PAP of the policy management. The NS healing policy is a kind of MANO policy to assist for the NFVO on determining what specific healing actions need to be taken within the NS instance, when one or more faulty VNFs and/or VLs in the NS instance are detected.

The OSS/BSS have the information of available VNF instances, for the sake of setting NS healing policy information. One possible example of NS healing policy may contain the following information:

- If the faulty information of a specific VNF or VL is reported to NFVO, according to the severity of the fault, the NFVO determines whether to migrate the VNF to other VM(s) or any other healing actions.

After the policy is received by NFVO, the NFVO enforces the NS healing policy during the lifecycle of NS instance. Each time the NFVO detects the occurrence of the condition, the NFVO executes the actions as the policy indicated. By applying the NS healing policy in the lifecycle management, the operation results are more consistent with the high-level requirements reflected by the policy.

5.8.5 Auto-scaling policy pre-defined in the VNFD

5.8.5.1 Use case description

This use case describes the process of applying auto-scaling policy in the VNF scaling operation. Auto-scaling policy is pre-defined in the VNFD by the VNF provider (who acts as the PAP) and enforced by the VNFM (who acts as the PF). The auto-scaling policy is known as auto-scale rules defined in ETSI GS NFV-IFA 011 [i.4].

Policy designer (belonging to the organization of VNF provider) determines the information of auto-scaling policy for a given type of VNF and encapsulates the policy information in the VNFD. Policy designer optionally receives the requirements from the operator as the input for determining auto-scaling policy. One example of auto-scaling policy may contain the following information:

- If certain VNF indicator values changes are detected, scaling out is then performed for a VNF instance with the given number of steps.

After the VNFD is on-boarded, the VNFM accesses the VNFD and gets the information of auto-scaling policy. The VNFM applies the auto-scaling policy during the lifecycle of VNF instance which is instantiated by using the VNFD. The VNFM monitors the condition of auto-scaling policy. On each time the VNFM detects the occurrence of the condition, the VNFM executes the action as the policy indicated, e.g. scale out with the number of steps in the above example of auto-scaling policy.

NOTE: The enforcement of auto-scaling policy has the dependency on the configuration of the VNF autoScalable attribute in the VNFD (see ETSI GS NFV-IFA 011 [i.4], clause 7.1.12). If auto-scaling functionality is disabled by this attribute, the policy will not be applied.

5.8.6 Associating VNF LCM policies with VNF instances

5.8.6.1 Use case description

A MANO policy may be created before the instantiation of NS instances or VNF instances to which it is applicable. In this case, the PAP determines the association of the MANO policy and instantiated NSs or VNFs, and transfers this association to the PF. It is also possible that the PAP indicates some criteria to the PF and the PF associates the MANO policy with the NS or VNF instances based on those criteria. This use case provides an example of the scenario concerning the association of a VNF LCM policy with specific VNF instances, in which the NFVO acts as the PAP and the VNFM acts as the PF.

The NFVO has previously created a VNF scaling policy and sent this policy to the VNFM for activation. One example of the information contained in a VNF scaling policy for auto-scaling is as follows:

"*If the monitored bytes of the VNF instance have crossed a threshold, then a scale out operation is performed for the VNF instance with a given number of scaling steps*".

The service provider does not want to apply this policy to all of the VNF instances managed by the VNFM, but to apply it to VNF instances belonging to the NS instance with a certain NS type. Therefore, after the instantiation of NS with this NS type, the NFVO further decides to associate this VNF scaling policy with the constituent VNF instances of the new NS instance. When the association is completed, the NFVO transfers this association information to the VNFM. The VNFM enforces this VNF scaling policy during the lifecycle of its associated VNF instances.

5.8.7 Disassociating VNF LCM policies from VNF instances

5.8.7.1 Use case description

The association of a MANO policy and the NS instances or VNF instances that the policy is applicable to may be released, due to the termination of the associated instances, the deletion of the policy, or a disassociation decision from the service provider. In this case, the PAP determines the disassociation of the MANO policy from NS instances or VNF instances that this PAP has previously associated, and transfers this disassociation to the PF. It is also possible that the PF disassociates the MANO policy from NS or VNF instances based on criteria indicated by the PAP. This use case provides an example of the scenario for disassociating a VNF LCM policy from its associated VNF instances, in which the NFVO acts as the PAP and the VNFM acts as the PF.

The NFVO previously established the association of a VNF scaling policy and the VNF instances that it is applicable to. One example of the information contained in a VNF scaling policy for auto-scaling is as follows:

"*If the monitored bytes of the VNF instance have crossed a threshold, then a scale out operation is performed for the VNF instance with a given number of scaling steps*".

The NFVO decides to disassociate this VNF scaling policy for auto-scaling from the VNF instances of the updated NS instance. When the disassociation is completed, the NFVO transfers this disassociation information to the VNFM. The VNFM internally releases the association between the VNF scaling policy and the VNF instances that the policy is applicable to. The VNFM does not apply the VNF scaling policy to those VNF instances any more.

5.8.8 Policy conflict detection in the PF

5.8.8.1 Use case description

This use case describes one possible process of policy conflict detection after the created policy is sent to the PF. Policy conflict detection is needed in a MANO system acting as a PF in order to allow the resolution of conflicts between/amongst multiple different policies.

A MANO policy for a specific PF (e.g. the VNFM) may have various sources (e.g. created by the operator, loaded from multiple different NSDs/VNFDs, created by the NFVO, etc.). Given multiple policies in the local PF, these policies may be in conflict. An example of two VNF scaling policies for one VNFM is used to explain the process of policy conflict detection, where the NFVO acts as the PAP and the VNFM acts as the PF.

The NFVO creates a VNF scaling policy and sends it to the VNFM. One example of a VNF scaling policy for auto-scaling may contain the following information:

- If certain VNF indicator values are detected, scaling out to level A is then performed for a VNF instance.

After this VNF scaling policy is sent, it may conflict with an existing scaling policy in the local VNFM, which is also created by this NFVO. An example of an existing VNF scaling policy for auto-scaling in the VNFM may contain the following information:

- If certain VNF indicator values are detected, scaling out to level B is then performed for a VNF instance.

In this situation, two policies have the same triggering condition, but different actions, which causes the policy conflict. After the newly created scaling policy is received by the VNFM, the VNFM compares the newly created scaling policy with the existing policies and detects that there is a conflict in the enforcement of both polices. The information of conflicted polices is further reported to the PAP (e.g. NFVO).

The NFVO may analyse various information. A decision is made by the NFVO regarding the policy conflict. Possible results might be:

- the NFVO decides to keep the newly created VNF scaling policy: Therefore, the NFVO activates the newly created VNF scaling policy while deactivates the existing VNF scaling policy; or
- the NFVO decides to keep the existing scaling policy and then the newly created VNF scaling policy is not activated by the NFVO.

The above use case of policy conflict detection can be further extended to a scenario that multiple PAPs interact with one PF for policy management. For example, the NFVO acting as the PAP sends a VNF scaling policy to the VNFM which acts as the PF. Meanwhile, the EM acting as the PAP sends another VNF scaling policy to the same VNFM. These two VNF scaling polices may be in conflict with each other in the target actions triggered by the same condition, and the conflict is detected by the VNFM. In this case, the VNFM reports the event of policy conflict to the NFVO and the EM respectively. It depends on the coordination between the PAPs (the NFVO and the EM in this example) to determine which policy (VNF scaling policy for auto-scaling in this example) will be performed with a higher priority.

5.8.9 Policy conflict detection in the PAP

5.8.9.1 Use case description

This use case describes one possible process of policy conflict detection in the PAP before the created policy is sent to the PF. An example of two VNF scaling policies created by the same NFVO for one VNFM is used to explain the process of policy conflict detection, where the NFVO acts as PAP and the VNFM acts as PF.

The NFVO creates a VNF scaling policy for the VNFM. One example of a VNF scaling policy for auto-scaling may contain the following information:

- If certain VNF indicator values are detected, scaling out to level A is then performed for a VNF instance.

Before the newly created policy is sent to the VNFM, the NFVO queries the policies existing in the local VNFM. An example of an existing VNF scaling policy for auto-scaling in the VNFM may contain the following information:

- If certain VNF indicator values are detected, scaling out to level B is then performed for a VNF instance.

In this situation, two policies have the same triggering condition, but different actions, which causes the policy conflict. The NFVO compares the newly created VNF scaling policy with the existing policies queried from VNFM and detects that there is a conflict in the enforcement of both policies. The NFVO may analyse various information. A decision is made by the NFVO regarding the policy confliction. Possible results might be:

- the NFVO decides to keep the newly created VNF scaling policy: Therefore, the NFVO sends the newly created VNF scaling policy to the VNFM and activates it, while deactivates the existing VNF scaling policy; or
- the NFVO decides to keep the existing VNF scaling policy and the newly created VNF scaling policy is not sent to the VNFM.

5.8.10 Enforcement among multi-layer polices

5.8.10.1 Use case description

MANO policies are deployed and enforced in the functional blocks which are located in different layers of MANO architecture, and the functional blocks include the NFVO, the VNFM and the VIM. Considering a scenario of performing a MANO procedure with applicable polices at different steps in the procedure, it can be foreseen the need of certain relationship between the policies at multiple NFV-MANO layers. The higher layer the MANO procedure step (e.g. an NS scaling workflow) is located in, the more layers of MANO polices are involved in the enforcement during the procedure. This use case uses NS scaling as an example of MANO functional procedure for illustrating the scenario of enforcement of multi-layer policies.

When a NS scaling is applied to a certain NS instance, multiple layers of policies can be already created and applied to the NS scaling workflow as well:

a) One or multiple NS scaling policies are activated in the NFVO.

b) For each constituent VNF instance of the NS instance, the VNFM applies corresponding VNF scaling policies.

c) Other placement constraints related policies like Affinity/Anti-Affinity rules among the constituent VNF instances or dependency polices among the constituent VNF instances are applied in the NS scaling workflow.

These multi-layered polices work together in the same NS scaling workflow, while each policy may focus on a certain point of the whole procedure and takes a certain action for optimizing or resolving that point. It is expected that a mechanism is needed for guaranteeing the enforcement of these policies in the NS scaling workflow.

5.8.11 Transferring VNF healing policy from the EM

5.8.11.1 Use case description

This use case describes the process of transferring VNF healing policy from the EM, in which the EM acts as the PAP and the VNFM acts as the PF. The VNF healing policy is a kind of MANO policy to assist for the VNFM on determining what specific healing actions be taken within the VNF instance(s) managed by the VNFM, when one or more faulty events in the VNF instance(s) are detected.

The EM has the information of available VNF instances, for the sake of setting VNF healing policy information. One possible example of VNF healing policy may contain the following information:

- If the faulty information of a specific VNF instance is reported to VNFM, according to the severity of the fault, the VNFM determines whether to migrate the VNF instance to other VM(s) or any other healing actions.

After the policy is created by the EM and transferred to the VNFM, the VNFM enforces the VNF healing policy during the lifecycle of VNF instance. Each time the VNFM detects the occurrence of the condition, the VNFM executes the actions as the policy indicated.

In cases where VNF provider defined auto-healing policy already exists in the VNFD (VNF provider is the PAP), it is the responsibility of the EM to resolve potential conflicts between the VNF healing policy created by EM and VNF auto-healing policy in the VNFD. The conflict resolution methods may include, but are not limited to: update of the VNF healing policy created by EM, disabling of VNF auto-healing functionality.

NOTE: This use case does not preclude scenarios where the EM performs or initiates the VNF healing directly by using the existing MANO functions. For example, the EM may request the healing / auto-healing script to be executed at the VNFM. Or, the EM may provide a set of specific VNF healing actions as parameter of the VNF heal operation.

5.8.12 Transferring virtualized resource allocation policy for the VIM

5.8.12.1 Use case description

Currently the information of locationConstraints is used as an input parameter in each operation of virtualized resource allocation (compute, storage, network). However, it can also be combined in a MANO policy on virtualized resource management and be transferred to the VIM for the enforcement automatically. This use case describes a process of transferring a virtualized resource allocation policy which makes effect on locationConstraints. In this use case, the NFVO acts as the PAP and the VIM acts as the PF.

The NFVO possesses the information on location constraints for each type of network resources and creates a policy correspondingly. One possible example of virtualized resource management policy may contain the following information:

- If the network resource type equals to "Subnet", then network resources are allocated in a particular resource zone A; If network resource type equals to "Network", then network resources are allocated in a particular resource zone B.

After the policy is created in the NFVO, the NFVO transfers the policy to the VIM. At each time the VIM receives a request of virtualized network resource allocation from the NFVO, the VIM applies this policy by reading the networkResourceType information in the input parameters of the request message and adapts to use location constraints according to the value of network resource type.

NOTE: The location constraints contained in specific virtualized network resource allocation request is expected to take precedence over the policies containing the behaviour on location constraints after the policy is activated.

5.8.13 Distribution of energy efficiency policies

5.8.13.1 Use case description

The service provider wants to influence the placement of virtualized resources to reduce the cost of energy and/or reduce energy consumption.

Cost of energy reduction

The service provider creates an energy cost reduction policy and makes it into effect in the NFVO. The NFVO uses the policy rules as input to the selection of NFVI PoPs when answering a Grant VNF lifecycle operation request. Each policy rule associates a condition to one or more NFVI PoPs to be selected. The policy can be static in which case the designer of the policy is assumed to be aware of the price of energy in each NFVI PoP or dynamic in which case the policy rules are generated by a policy engine that collects, through one or more APIs, cost information from each of the energy providers serving the PoPs. The condition in a policy rule can be related to the type of NS and/or VNF considered and/or on the time the request is received (e.g. in case of time-varying energy price). The selection of the NFVI PoPs is determined by a combination of these rules with other rules that belong to other categories of policies. The weight and priority of energy efficiency policy rules compared to other policy rules are configured by the service provider.

Energy consumption reduction

The service provider creates an energy consumption reduction policy and makes it into effect in the NFVO and the VIM(s).

Energy consumption can be reduced by consolidating workloads on a smaller number of NFVI PoPs so that a portion or the totality of an unloaded NFVI-PoP can then be switched off. In this case the energy consumption reduction policy is a particular incarnation of a resource optimization policy as described in clause 5.8.3.

5.9 Recommendations

5.9.1 Overview

A similar process can be applied inside an NFVI PoP by consolidating the workload on a smaller number of NFVI nodes so that a portion or the totality of unloaded NFVI nodes in the PoP can then be switched off. In this case the policy information is made available to the VIM.

In this clause, recommendations are proposed from the above use cases, which can be regarded as an input to trigger the architectural option discussion and solution selection for fulfilling MANO policy management in the subsequent clauses. The following clauses provide recommendations on management requirements to be specified.

5.9.2 General recommendations

Table 5.9.2-1 provides the general recommendations for MANO policy management.

Table 5.9.2-1: General recommendations for MANO policy management

Rec Number	Recommendation Description	Comments
POLICY.MGT.001	It is recommended that a requirement be specified for the NFVO to support the lifecycle management of MANO policies (see note).	*Derived from use cases in clauses 5.2, 5.3, 5.4, 5.5, 5.6 and 5.7.*
POLICY.MGT.002	It is recommended that a requirement be specified for the VNFM to support the lifecycle management of MANO policies (see note).	*Derived from use cases in clauses 5.2, 5.3, 5.4, 5.5, 5.6 and 5.7.*
POLICY.MGT.003	It is recommended that a requirement be specified for the VIM to support the lifecycle management of MANO policies (see note).	*Derived from use cases in clauses 5.2, 5.3, 5.4, 5.5, 5.6 and 5.7.*
POLICY.MGT.004	It is recommended that a requirement be specified for the Lifecycle management of MANO policies to be fulfilled by using the existing MANO functional blocks and reference points (see note).	*Derived from use cases in clauses 5.8.2 and 5.8.4.*
POLICY.MGT.005	It is recommended that a requirement be specified for the NFVO acting as a PF to be able to parse a policy.	*Derived from use cases in clause 5.2, 5.4, 5.6 and 5.7.*
POLICY.MGT.006	It is recommended that a requirement be specified for the VNFM acting as a PF to be able to parse a policy.	*Derived from use cases in clauses 5.2, 5.4, 5.6 and 5.7.*
POLICY.MGT.007	It is recommended that a requirement be specified for the VIM acting as a PF to be able to parse a policy.	*Derived from use cases in clauses 5.2, 5.4, 5.6 and 5.7.*
POLICY.MGT.008	It is recommended that a requirement be specified for the NFVO acting as a PF to report the conflicted policies it detects.	*Derived from use case in clauses 5.8.8.*
POLICY.MGT.009	It is recommended that a requirement be specified for the VNFM acting as a PF to report the conflicted policies it detects.	*Derived from use case in clause 5.8.8.*
POLICY.MGT.010	It is recommended that a requirement be specified for the VIM acting as a PF to report the conflicted policies it detects.	*Derived from use case in clause 5.8.8.*
POLICY.MGT.011	It is recommended that a requirement be specified for the NFVO acting as a PAP to resolve the conflicted policies it creates.	*Derived from use cases in clauses 5.8.8 and 5.8.9.*
NOTE: It is possible that the collective functions of the NFVO, the VNFM and the VIM plays only a partial role in the management of MANO policies, e.g. the collective functions of the NFVO, the VNFM and the VIM system acts as the PAP or the PF for a particular policy or a category of policies.		

5.9.3 Recommendations for policy lifecycle management

Table 5.9.3-1 provides the recommendations for potential requirements related to MANO policy lifecycle management.

Table 5.9.3-1: Recommendations for MANO policy lifecycle management (see note 1)

Rec Number	Recommendation Description	Comments
POLICY.LCM.001	It is recommended that a requirement be specified for the NFVO acting as a PAP to support the creation of MANO policies.	*Derived from use case in clause 5.2.*
POLICY.LCM.002	It is recommended that a requirement be specified for the VNFM acting as a PAP to support the creation of MANO policies.	*Derived from use case in clause 5.2.*
POLICY.LCM.003	It is recommended that a requirement be specified for the NFVO acting as a PAP to support the transfer of MANO policies to the VNFM or the VIM acting as a PF.	*Derived from use case in clause 5.2.*
POLICY.LCM.004	It is recommended that a requirement be specified for the VNFM acting as a PAP to support the transfer of MANO policies to the VIM acting as a PF.	*Derived from use case in clause 5.2.*
POLICY.LCM.005	It is recommended that a requirement be specified for the NFVO to support the deletion of MANO policies.	*Derived from use case in clause 5.3.*
POLICY.LCM.006	It is recommended that a requirement be specified for the VNFM to support the deletion of MANO policies.	*Derived from use case in clause 5.3.*
POLICY.LCM.007	It is recommended that a requirement be specified for the VIM to support the deletion of MANO policies.	*Derived from use case in clause 5.3.*
POLICY.LCM.008	It is recommended that a requirement be specified for the NFVO to support the update of MANO policies.	*Derived from use case in clause 5.4.*
POLICY.LCM.009	It is recommended that a requirement be specified for the VNFM to support the update of MANO policies.	*Derived from use case in clause 5.4.*
POLICY.LCM.010	It is recommended that a requirement be specified for the VIM to support the update of MANO policies.	*Derived from use case in clause 5.4.*
POLICY.LCM.011	It is recommended that a requirement be specified for the NFVO to support the query of MANO policies.	*Derived from use case in clause 5.5.*
POLICY.LCM.012	It is recommended that a requirement be specified for the VNFM to support the query of MANO policies.	*Derived from use case in clause 5.5.*
POLICY.LCM.013	It is recommended that a requirement be specified for the VIM to support the query of MANO policies.	*Derived from use case in clause 5.5.*
POLICY.LCM.014	It is recommended that a requirement be specified for the NFVO to support the activation of MANO policies. See note 2.	*Derived from use case in clause 5.6.*
POLICY.LCM.015	It is recommended that a requirement be specified for the VNFM to support the activation of MANO policies. See note 2.	*Derived from use case in clause 5.6.*
POLICY.LCM.016	It is recommended that a requirement be specified for the VIM to support the activation of MANO policies. See note 2.	*Derived from use case in clause 5.6.*
POLICY.LCM.017	It is recommended that a requirement be specified for the NFVO to support the deactivation of MANO policies. See note 2.	*Derived from use case in clause 5.7.*
POLICY.LCM.018	It is recommended that a requirement be specified for the VNFM to support the deactivation of MANO policies. See note 2.	*Derived from use case in clause 5.7.*
POLICY.LCM.019	It is recommended that a requirement be specified for the VIM to support the deactivation of MANO policies. See note 2.	*Derived from use case in clause 5.7.*
POLICY.LCM.020	It is recommended that a requirement be specified for the NFVO acting as a PAP to support associating VNF LCM policies it creates with corresponding VNF instances.	*Derived from use case in clause 5.8.6.*
POLICY.LCM.021	It is recommended that a requirement be specified for the NFVO acting as a PAP to support disassociating VNF LCM policies it creates from corresponding VNF instances.	*Derived from use case in clause 5.8.7.*

NOTE 1: It is allowed to apply the above requirements to multiple PAPs per case by case basis but it does not apply to the creation of policies, e.g. the policy management function provided by the VNFM is consumed by the NFVO or the EM.

NOTE 2: This requirement is valid for the case the policy has an administrative state in the PF. In other cases the policy is activated implicitly by pushing it to the PF and deactivated by deleting it.

6 Key issue analysis

6.1 Key Issue 1: Mapping of PAP and PF to MANO functional blocks

6.1.1 Overview

The PAP and the PF are the abstracted roles applied in high-level use case description of policy management. In the context of MANO architecture, the PAP and the PF are mapped to MANO functional blocks in different ways. The subsequent clauses elaborate the cases of PF mapping in the MANO architecture and summarize the PAP mapping from those cases at the end.

6.1.2 Mapping the PF to the NFVO and the PAP to the OSS/BSS

When the MANO policy is applied for NS LCM operations and orchestration of resources managed across VIM, the PF may be mapped to the NFVO and the PAP may be mapped to the OSS/BSS. Those NS LCM operations include but not be limited to: NS instantiation, NS scaling, NS healing and NS termination, and examples of resource orchestration are (but not limited to): virtualised resource reservation, and virtualised resource quota management. The mapping relationship is shown in figure 6.1.2-1. This mapping is exemplified by the use case in clause 5.8.4, when creating NS healing policy.

The policies applicable for this mapping include NS LCM-type functionality, such as: NS instantiation policy, NS scaling policy, NS updating policy, NS healing policy, NS termination policy and virtualised resource orchestration-type functionality, such as: virtualised resource capacity management, virtualised resource reservation management, and virtualised resource quota management.

Figure 6.1.2-1: Mapping the PF to the NFVO and the PAP to the OSS/BSS

6.1.3 Mapping the PF to the VNFM and the PAP to the NFVO and/or the EM

When the MANO policy is applied for VNF LCM operations, the PF may be mapped to the VNFM and the PAP may be mapped to the NFVO or the EM. Those VNF LCM operations include but not be limited to: VNF instantiation, VNF scaling, VNF healing and VNF termination. The mapping relationship is shown in figure 6.1.3-1. This mapping is exemplified by the use case in clause 5.8.4, when creating NS healing policy.

The policies applicable for this mapping include VNF LCM-type functionality, such as: VNF instantiation policy, VNF scaling policy, VNF healing policy and VNF termination policy.

NOTE: Since the VNFM is not the trigger of VNF instantiation, the applicability of VNF instantiation policy to be enforced at the VNFM is FFS.

Figure 6.1.3-1: Mapping the PF to the VNFM and the PAP to the NFVO and/or the EM

6.1.4 Mapping the PF to the VIM and the PAP to the NFVO or VNFM

When the MANO policy is applied for virtualized resource management operations, the PF may be mapped to the VIM and the PAP may be mapped to the NFVO. The mapping relationship is shown in figure 6.1.4-1.

The policies applicable for this mapping include: virtualised resource allocation policy, and virtualised resource reservation policy (only applicable if the NFVO is the PAP), virtualised resource quota policy (only applicable if the NFVO is the PAP), and virtualised resource capacity policy (only applicable if the NFVO is the PAP).

Figure 6.1.4-1: Mapping the PF to the VIM and the PAP to the NFVO

When the MANO policy is applied for virtualized resource management operations, the PF may be mapped to the VIM and the PAP may also be mapped to the VNFM. The mapping relationship is shown in figure 6.1.4-2. This case is only applicable to the virtual resource(s) used by the VNF which the VNFM manages.

Figure 6.1.4-2: Mapping the PF to the VIM and the PAP to the VNFM

6.1.5 Summary of PAP mapping

Within MANO, the PAP is mapped to the NFVO, based on the use cases developed in clause 5.8. The PAP mapped to the VNFM is only applicable to the virtual resource used by the VNF which the VNFM manages. The PAP can also be mapped to OSS/BSS or the EM but they are both out of the scope of MANO functional blocks.

For the cases of pre-defined policies in the VNFD (see the use case in clause 5.8.5), the PAP is inside the domain of VNF provider by setting the policy in the VNFD.

6.2 Key Issue 2: Categorizing policies in MANO architecture

6.2.1 Overview

From the use cases on NFV MANO specific policy management under clause 5.8, it can be seen that MANO policies are mainly applicable to MANO reference points to assist for corresponding MANO functions like NS LCM, VNF LCM or resource management. Categorizing those policies will provide an explicit view on the scope of policies under this study, and further benefit for the modelling of policy information in MANO architecture.

6.2.2 MANO policy categories

MANO policies can be grouped to the categories as shown in table 6.2.2-1. Different rows of the same policy category indicates different cases of PAP/PF mapping for applying the polices in this category.

Table 6.2.2-1: MANO policy categories

Policy Category	PF	PAP
NS instantiation policy	NFVO	OSS
	NFVO	NSD (see note)
NS scaling policy	NFVO	OSS
	NFVO	NSD (see note)
NS update policy	NFVO	OSS
	NFVO	NSD (see note)
NS healing policy	NFVO	OSS
	NFVO	NSD (see note)
NS termination policy	NFVO	OSS
	NFVO	NSD (see note)
VNF instantiation policy	VNFM	NFVO
	VNFM	EM
	NFVO	NSD (see note)
	VNFM	VNFD (see note)
VNF scaling policy	VNFM	NFVO
	VNFM	EM
	NFVO	NSD (see note)
	VNFM	VNFD (see note)
VNF healing policy	VNFM	NFVO
	VNFM	EM
	NFVO	NSD (see note)
	VNFM	VNFD (see note)
VNF termination policy	VNFM	NFVO
	VNFM	EM
	NFVO	NSD (see note)
	VNFM	VNFD (see note)
Virtualised resource allocation policy	VIM	NFVO
	VIM	VNFM
Virtualised resource reservation policy	VIM	NFVO
	NFVO	OSS
Virtualised resource quota (management) policy	VIM	NFVO
	NFVO	OSS
Virtualised resource capacity (management) policy	VIM	NFVO
	NFVO	OSS
NOTE: NSD and VNFD are not actors, the PAPs are the creators of those artifacts.		

6.3 Key Issue 3: Principles on information modelling of MANO policies

6.3.1 Overview

Information modelling is an elementary aspect for fulfilling policy management in NFV MANO. Instead of determining the detailed information model of MANO policies which is out of the scope of the present document, this clause provides some general analysis on the principles which can be used in guiding the information modelling of MANO policies in the normative work. Information modelling of policies has amply been studied by other SDOs IETF RFC 3060 [i.11] and IETF RFC 3460 [i.12].

6.3.2 Lessons learnt from other work

Some investigation is made on the policy model work in TM Forum's Zero-touch Orchestration, Operations and Management (ZOOM) policy model [i.5], IETF Simplified Use of Policy Abstraction (SUPA) work [i.6], [i.9] and [i.10] and OpenStack® Congress project [i.7]. The lessons on policy information modelling are summarized as follows:

1) Policy exists at multiple levels of abstraction. Each level of abstraction serves a particular constituency.

2) A policy at a high level of abstraction can be implemented as declarative versus a policy at lower levels of abstraction can be implemented as imperative, like the Event-Condition-Action (ECA) model.

3) Any real policy management system needs a formal notion of what a "policy conflict" is and how to detect and resolve the conflict.

6.3.3 Principles on MANO policy information modelling

MANO policies can be either declarative or imperative ones. For example, policies related to VNF placement constraints (e.g. Affinity/Anti-Affinity rules) are declarative, and NS LCM policies or VNF LCM polices which are transferred over MANO reference points are imperative. To construct the information model of MANO polices, some generic models can be referenced, like the ECA model (Event, Condition and Action) for imperative policies. It is also necessary to analyze whether other information modelling aspects specific to NFV MANO will be applied.

- Lifecycle aspects of a MANO policy.

A MANO policy has its own lifecycle and can be dynamically created, updated and deleted. Unlike the management policies applied to the traditional Physical Network Function(s) (PNF), in which once the policy is created, it is enforced for a long time duration with seldom change, a MANO policy is more dynamic and frequently associated with or disassociated from the objects (e.g. NS instance(s), VNF instance(s)) it is applied to. Therefore, the first observation on the information modelling of a MANO policy is that it can be managed as the policy instance.

Principle 1: With regards to the dynamics of a MANO policy during its lifecycle, it can be modelled as a policy instance.

- Relationship with the associated objects.

The next observation on modelling MANO policy is how to model the policy instance with regards to its associated NS or VNF instances. One possibility is to model the policy instance and its associated NS or VNF instance(s) separately and establish their associations when those instances are available. This will benefit for the separate design and optimization of the elementary part (e.g. basic LCM and virtualized resource allocation functions) and the value-added part (e.g. applying a policy) of the associated objects and make each part consistent in its internal logic.

Principle 2: A MANO policy can be modelled separately from its associated objects (e.g. NS or VNF instance(s)) and establish the association among instances when they are available.

- Applying policy descriptor in modeling MANO policy.

The concept of descriptor is widely used for information modelling in the scope of NFV MANO architecture, e.g. the NSD, VNFD, PNFD, VLD and VNFFGD as specified in ETSI GS NFV-IFA 014 [i.8] or ETSI GS NFV-IFA 011 [i.4].

Following Principle 1, MANO policy can be modelled as a policy instance which is created by the PAP and transferred to the PF for the enforcement. Furthermore, the PAP can apply policy descriptor to the lifecycle management of policy instance(s) it creates. Policy descriptor is associated with a certain policy instance after the policy instance is created by the PAP, and it includes the meta-data information (e.g. the category of the MANO policy, the organization generating the policy descriptor, basic description of the policy) of policy instance which further facilitates the lifecycle management of the policy instance between the PAP and the PF.

Principle 3: A MANO policy instance can be managed by using policy descriptor which contains the meta-data information for modelling MANO policy.

NOTE: In this study, only the principle of applying policy descriptor in MANO policy management is determined. It is recommended to further define policy descriptor management related use cases and requirements in the normative work.

7 Architectural options

7.1 Introduction

The use cases determined in clause 5 derive a series of potential requirements for managing policies in MANO architecture, which are eventually fulfilled by corresponding management functions. Those management functions, can be placed or distributed in MANO architecture in different ways, according to the potential requirements in clause 5.9 which are derived from clause 5 use cases. This clause proposes the potential architectural options for supporting the placement of policy management functions in MANO architecture.

7.2 Architectural option #1: Managing policies by using existing MANO reference points

7.2.1 Overview

This architectural option aligns with the recommendation POLICY.MGT.002 in clause 5.9.2, in which the existing MANO reference points are used and enhanced with newly added policy management interface(s) (e.g. policy lifecycle management interface) to support the management of policies enforced by the NFVO, VNFM or VIM.

7.2.2 Managing MANO policies using Os-Ma-nfvo

In this case, the reference point Os-Ma-nfvo is enhanced with a newly added policy management interface. The OSS acts as the PAP, which is the consumer of the policy management interface. The NFVO acts as the PF, which is the producer of the policy management interface, as shown in figure 7.2.2-1.

The policies applicable for this mapping include NS LCM-type functionality, such as: NS instantiation policy, NS scaling policy, NS updating policy, NS healing policy and NS termination policy, and virtualised resource orchestration-type functionality, such as: virtualised resource capacity management, virtualised resource reservation management, and virtualised resource quota management policies.

NOTE: The new policy management interface is not applicable to policy rules specified in the NSD and VNFD.

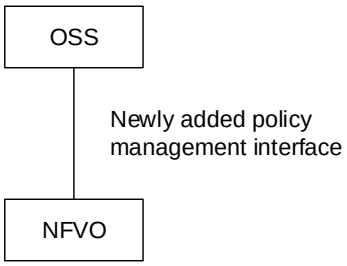

Figure 7.2.2-1: Managing MANO policies using Os-Ma-nfvo

7.2.3 Managing MANO policies using Or-Vnfm

In this case, the reference point Or-Vnfm is enhanced with a newly added policy management interface. The NFVO acts as the PAP, which is the consumer of the policy management interface. The VNFM acts as the PF, which is the producer of the policy management interface, as shown in figure 7.2.3-1.

Policy rules transferred using the new policy management interface are either generated by the NFVO, based on policy rules received from the OSS, or are transparently relayed from the OSS to the VNFM.

The policies applicable for this mapping include: VNF instantiation policy, VNF scaling policy, VNF healing policy and VNF termination policy.

NOTE 1: The new policy management interface is not applicable to the transfer of policy rules specified in the NSD and VNFD.

NOTE 2: Since the VNFM is not the trigger of VNF instantiation, the applicability of VNF instantiation policy to be enforced at the VNFM is FFS.

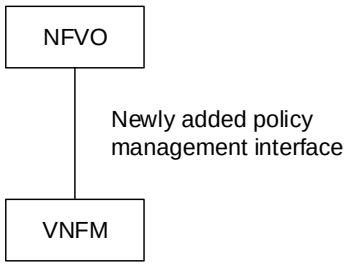

Figure 7.2.3-1: Managing MANO policies using Or-Vnfm

7.2.4 Managing MANO policies using Or-Vi

In this case, the reference point Or-Vi is enhanced with a newly added policy management interface. The NFVO acts as the PAP, which is the consumer of the policy management interface. The VIM acts as the PF, which is the producer of the policy management interface, as shown in figure 7.2.4-1.

Policy rules transferred using the new policy management interface are either generated by the NFVO, based on policy rules received from the OSS, or are transparently relayed from the OSS to the VIM.

The policies applicable for this mapping include: virtualised resource allocation policy, and virtualised resource reservation policy (only applicable if the NFVO is the PAP), virtualised resource capacity policy (only if the NFVO is the PAP), and virtualised resource quota policy (only if the NFVO is the PAP).

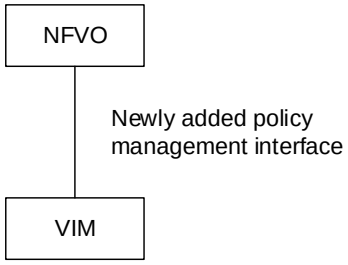

Figure 7.2.4-1: Managing MANO policies using Or-Vi

7.2.5 Managing MANO policies using Ve-Vnfm

In this case, the reference point Ve-Vnfm is enhanced with a newly added policy management interface. The EM acts as the PAP, which is the consumer of the policy management interface. The VNFM acts as the PF, which is the producer of the policy management interface, as shown in figure 7.2.5-1.

The policies applicable for this case include: VNF instantiation policy, VNF scaling policy, VNF healing policy and VNF termination policy.

NOTE 1: The new policy management interface is not applicable to the transfer of policy rules specified in the VNFD.

NOTE 2: Since the VNFM is not the trigger of VNF instantiation, the applicability of VNF instantiation policy to be enforced at the VNFM is FFS.

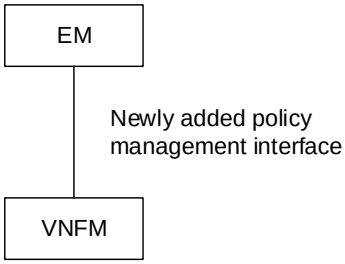

Figure 7.2.5-1: Managing MANO policies using Ve-Vnfm

7.2.6 Managing MANO policies using Vi-Vnfm

In this case, the reference point Vi-Vnfm is enhanced with a newly added policy management interface. The VNFM acts as the PAP, which is the consumer of the policy management interface. The VIM acts as the PF, which is the producer of the policy management interface, as shown in figure 7.2.6-1. This case is only applicable to the virtual resource used by the VNF which the VNFM manages.

Policy rules transferred using the new policy management interface are either generated by the VNFM, based on policy rules received from the NFVO, or are transparently relayed from the NFVO to the VNFM. In the former case the VNFM is the PAP, while in the latter case the NFVO is the PAP.

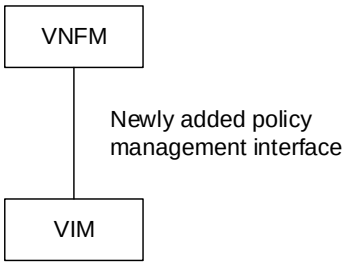

Figure 7.2.6-1: Managing MANO policies using Vi-Vnfm

7.2.7 Impact on MANO Architecture

7.2.7.1 Introduction

In the current MANO architecture, the NFVO plays an important role in MANO policy management. The NFVO has valid reference points to all the other MANO functional blocks like the VNFM and the VIM as well as to OSS/BSS which is out of the scope of MANO functional block. Hence, the NFVO could be evolved to have the capability of policy management, via enhancing the corresponding Or-Vnfm and Or-Vi reference points with policy management interfaces.

7.2.7.2 Policy conflict avoidance

The NFVO may create VNF lifecycle management policies, resource management policies and activates those policies in the VNFM and the VIM, respectively. During the design and creation of those policies, the NFVO is responsible for examining the newly created polices with the existing policies in its policy database to avoid the potential conflicts of enforcement for those policies in advance.

7.2.7.3 Policy conflict detection and resolution

After the policies created by the NFVO are activated in the VNFM or the VIM, the NFVO is responsible for resolving potential conflicts between the policies. As described in use cases under clauses 5.8 and 5.9, no matter whether the VNFM or the VIM (acting as the PF) or the NFVO (acting as the PAP) detects the conflicted polices during their enforcement phase, the NFVO is aware of the conflict and provides the resolution for the conflicted polices. The NFVO plays the role of coordinator of policy enforcement for those policies enforced by the VNFM or the VIM.

7.2.7.4 Summary

As a summary on the impact on MANO architecture, MANO policy management functions are integrated in the NFVO. Apart from basic functions of creating, updating, deleting and querying MANO policies, the NFVO is also capable of avoiding the policy confliction during the policy designing phase and coordinating the policies in MANO architecture during the lifecycle of the NS instances it managed.

7.3 Analysis

Other architectural options than Option#1 are also discussed in this study. A proposal is addressed on potential architectural option for managing MANO policies by a standalone PAP (centralized architecture for MANO policy management), in which a standalone entity of PAP interacts with the PFs located in the NFVO, VNFM and the VIM respectively, as shown in figure 7.3-1.

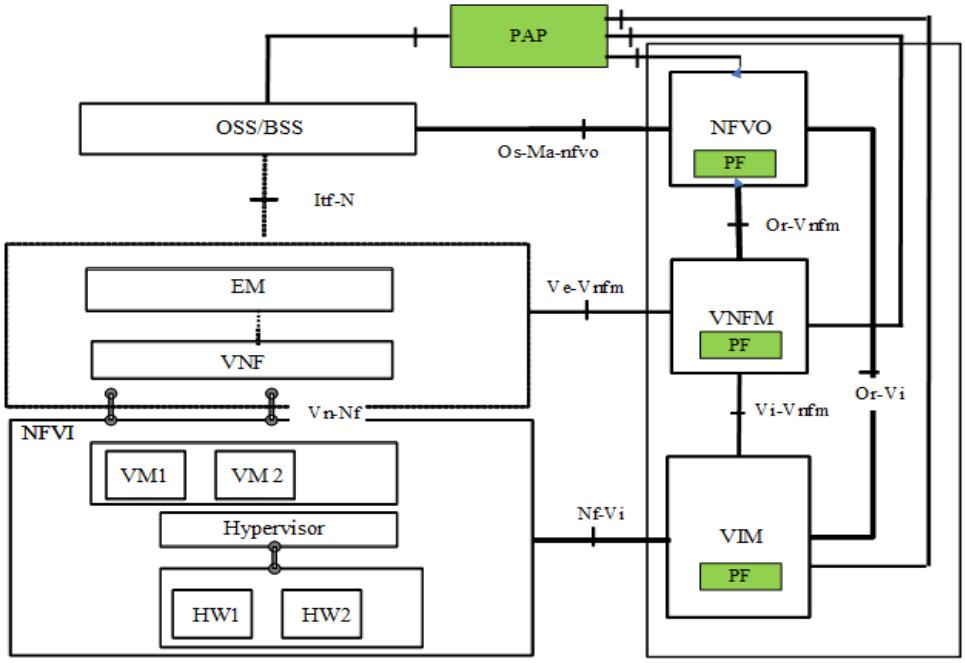

Figure 7.3-1: Centralized MANO policy management

The concern for introducing this architectural option mainly includes the following aspects:

- In a multi-PAP environment, each PAP is only aware of the enforcement status of policies which are created by itself. If the PAP wants to know the enforcement status of polices created by another PAP, it has to interact with that PAP for capturing the status.
- If the coordination among multiple policies is needed, or a conflict among multiple policies is detected, in which those policies are created by multiple PAPs, it is unclear which PAP is responsible for the final decision of the conflict resolution;

There is no explicit use case to resolve the above two concerns. For example, in a multi-PAP environment, it is not clear under what cases does a PAP need to know the enforcement status of policies created by another PAP, or under what cases do two PAPs interact with each other for policy conflict resolution.

However, the issues pointed out above are not exclusively related to stand-alone PAPs, they could also occur in architectural option #1 e.g. if the EM and NFVO create policies within the same jurisdiction area. A single stand-alone PAP within a policy jurisdiction area would not face the problems mentioned above.

Stand-alone PAPs are a valid architectural option but it is recommended to limit this option to the case in which it does not impact existing and does not create new reference points in NFV architecture.

8 Recommendations

The present document makes the investigation on the use cases, potential functional requirements, key issues analysis and architectural options on policy management in NFV MANO architecture. Based on the output of the study, normative work for MANO policy management is needed in the following aspects:

- Architectural aspect: The existing MANO architecture is reused to support policy management, in which Os-Ma-Nfvo, Or-Vnfm, Or-Vi, Vi-Vnfm and Ve-Em-Vnfm reference points are enhanced with a new policy management interface. Policy management functions are integrated in the NFVO, VNFM and the VIM respectively, which acts as the PF to provide interface functions. The OSS/BSS, NFVO, VNFM or the EM which acts as the PAP consumes corresponding functions provided by policy management interface. Further consideration to work done in other fora (TM Forum TR235 [i.5]) should be given when defining the interfaces.

- Information model aspect: MANO policy is modeled as policy instance(s), and modeled separately from its associated objects. The policy descriptor is used to record the meta-data of policy information for managing MANO policies.

- Functional aspect: Policy transfer, deletion, update, query, activation and deactivation are the basic set of functions to be supported by policy management interface. Policy conflict detection and resolution in the enforcement phase is necessary to be supported as well. Coordination among multiple PAPs in policy conflict detection and resolution is out of the scope of work in this release.

Annex A:
Authors & contributors

The following people have contributed to the present document:

Rapporteur:
Xia Haitao, Huawei Technologies

Other contributors:
Jiang Junyi, Huawei

Zhao Peng, China Mobile

Nicola Santinelli, TELECOM ITALIA S.p.A

Miao Jie, China Unicom

He Gang, China Unicom

Tong Junjie, China Unicom

Stephen Fratini, Ericsson

Wang Jian, Huawei

Bruno Chatras, ORANGE

Janusz Pieczerak, ORANGE

Joan Triay, DOCOMO Communications Lab

Kazuaki Obana, DOCOMO Communications Lab

Ashiq Khan, DOCOMO Communications Lab

Gerald Kunzmann, DOCOMO Communications Lab

Bertrand Souville, DOCOMO Communications Lab

Arturo Martin de Nicolas, Ericsson LM

Cristina Badulescu, Ericsson LM

History

Document history		
V3.1.1	July 2017	Publication