

Design for Edge/Fog Application Management and Orchestration

Hung-Yu Wei¹, Yao Chiang, Tse-Yu Chen
Department of Electrical Engineering, National Taiwan University

1. Edge Application Management and Orchestration²

The technology of edge computing has evolved considerably in recent years with the support of emerging key enablers like Network Function Virtualization (NFV) and Software Defined Network (SDN). NFV and SDN can be used to consolidate and deliver the networking components needed to support a fully virtualized infrastructure, such as virtual servers, storage units, and even other networks. They are applicable to any data plane processing or control plane function in both wired and wireless network infrastructures.

Also, as a maturing solution for emerging applications and scenarios, many different variants of Edge/Fog systems have been created to handle various use cases and different deployment conditions. The most famous examples include the next generation mobile network, smart factory, and Industrial Internet of Things (IIoT). Even though the designs adapted to local conditions can result in a powerful standalone product, such a large number of variants may damage the connectivity and increase communication costs between systems. In this context, the need to standardize the relevant system architecture and operational processes become evident, and IEEE P1935 standard for Edge/Fog manageability and orchestration is created in response to it.

As a standard for management and orchestration of Edge/Fog systems, IEEE P1935 covers the system architecture, resource management, application lifecycle operations, and communication specification between multiple Edge/Fog systems. The unified management and orchestration, such as the regulation of the transmissions of messages, the controlling and allocation of the resources of the edge devices, and the monitoring and arrangement in the mobile application lifecycle, can satisfy the mentioned needs and ensure better availability, flexibility, reliability, scalability, stability, service mobility, and performance. It means that the system shall be easy for multiple types of users to operate and configure, be hard to down while facing different challenges, and be as efficient as possible.

Regardless of technical details and application iterations, the primary goal of edge computing technology is to provide stable and high-quality services, or more specifically, Edge Applications. In order to provide the complete lifecycle management of Edge Applications, the system has to have knowledge about the application run-time information, and thus it is necessary to specify the procedures and the message flows between all components involved. This document describes the managing procedures and message flows of applications at Edge and pays more attention to how to run and coordinate Edge Applications in the Edge system. The main six operations in the application lifecycle, including application onboarding, application instantiation, context creation, application reconfiguration, context deletion, and application termination, are introduced in each subclause respectively. Each subclause provides a close look to the procedures, conditions, involved information, and format of requests and responses of one operation. With the help of this document, practitioners can more quickly establish a consistent and reliable Edge/Fog system and ensure their Edge Applications to function properly.

1.1 Overview of Edge Application Management

Management and orchestration (M&O) means the management and coordination of the resources, including network resources and computing resources, in order to provide the lifecycle control of the Edge/Fog applications. With such management, it is possible for Edge Service Operators (or simply

¹ Corresponding Author: Hung-Yu Wei (email: hywei@ntu.edu.tw)

² This document was submitted as technical contribution to IEEE P1935 working group in 2020. Part of the technical contents have been included in the IEEE P1935 standard.

“operators”) to manage their Edge/Fog services and applications, which is provided by Edge Service Providers (or simply “service providers”), and to make adjustments and modifications based on the scenario, network condition, and user information. An Edge/Fog service is composed of one or more Edge Applications, and an Edge Application is composed of one or more virtual functions (VFs).

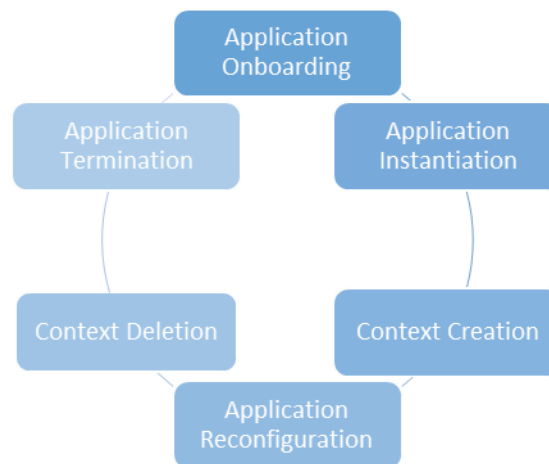


Figure 1—The diagram of lifecycle procedures of Edge Applications

As shown in Figure 1, a typical application has the following lifecycle management procedures:

- a) **Application on-boarding:** In the application on-boarding procedure, the Edge management needs to obtain the necessary software files and description files that are necessary to start running applications.
- b) **Application instantiation:** Based on the application blueprint file, which describes the management and orchestration rules for an application, the EFO is able to instantiate an application with Edge App Designer making the homing decision of the Edge Application.
- c) **Context creation:** After instantiation, the Edge Application will run on the decided Edge Platform. In the process, it is common for the Edge Service Providers to collect certain user data or for the users to upload their own data. The context, or “application context”, involves all the files that contain the custom information operators need, such as user information and statistical data.
- d) **Context deletion:** Edge Service Providers are able to remove the certain application context (such as user data) which they don’t need anymore.
- e) **Application reconfiguration:** Based on the collected data and information, the Edge Service Providers can reconfigure their Edge Applications. For example, it is possible to increase the accuracy of the service in the cost of latency to meet the requirements of the specific users.
- f) **Application termination:** If an Edge Application is no longer needed at least in the Edge system coverage, the Edge Service Operators can remove the Edge Application from the Edge Platform and even the Edge Platform Manager.

It is common for a standard Edge Application to undergo various lifecycle procedures, including on-boarding, instantiation, termination, and reconfiguration. Also for the application context, it is necessary for the Edge Service Providers and the Edge Service Operators to modify the data and information contained in their services, which corresponds to context creation and context deletion procedures. An instance is a concrete occurrence of an Edge Application which has gone through the instantiation process. It is possible to do context creation, context deletion, and application reconfiguration whenever the Edge Application is active, that is, when the Edge Application has been instantiated and hasn’t been terminated.

In the process, Edge Service Providers offer their applications and services onto the Edge/Fog system by uploading the specific manifest files, and infrastructure owners provide their Edge/Fog devices, and Edge Service Operators authorize the Edge Service Providers and the third-party users to leverage the

functionality of their Edge/Fog system. All these roles can be performed by the same group or the different groups of people.

The manifest file is necessary for the Edge Service Providers to onboard their applications and service. The manifest file contains an application package and a blueprint file. The application package is hierarchical, containing the necessary data and information of the application and its components, like VF images. The blueprint file includes the orchestration rules and the resource lifecycle management designs, etc. The Edge App Designer component in the EFO will process the files, and output the artifacts that will be distributed to the other components in the EFO.

There are also various terminologies involved in the lifecycle procedures, such as VF images and inventory. All these terms are described in Table 3 and the related subsections.

In addition, it is common for the entities and components to send requests and responses to make sure that the progress goes properly. Table 1 shows the default format of a request/response. The format of those with specific needs is described in each section respectively.

Table 1—Default format of a request/response

Element	Description
Operation status	A 200 OK code shall be returned if successful. A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed.

Table 2—The components mentioned in 1.2

Types	Components	1.2.1	1.2.2	1.2.3	1.2.4	1.2.5	1.2.6
Users	Service Provider	O					
	Operator	O	O			O	O
	End User App			O	O		
EFO	External API		O			O	O
	End User Proxy			O	O		
	Edge App Designer	O					
	VF M&O	O	O	O	O	O	O
	Edge Inventory Manager	O	O	O		O	O
	LCM Enabler	O					
	Rule Framework	O					
	EFO Controllers	O					
Control nodes	Edge Platform Manager	O	O	O	O	O	O
	IM/VIM	O	O			O	O
Compute nodes	Edge Platform		O	O	O	O	O
	Edge App		O	O	O	O	O
	VNI & Data Plane	O	O			O	

Table 3—The Files used in the Lifecycle Procedures

Terms	Descriptions	Clauses
Application context	Application context, or simply “context”, involves all the files that contain the custom information operators need, such as user information and statistical data.	1.2.3.1
Application package	Application packages are ones of the files that are involved in the application onboarding procedures. The application package is hierarchical, containing the necessary data and information of the application and its components, like VF images. An application package can be considered as an installation file that will be deployed to a compute node to launch a specific Edge Application.	1.2.1.1
Artifact	Artifacts are all the files that are generated by the Edge App Designer component and are distributed to the other components in the EFO. They are the descriptor files for the components in the EFO.	1.2.1.1
Blueprint file	A blueprint file is a file that includes the orchestration rules and the resource lifecycle management designs, etc. This file is used to configure the components in the EFO.	1.2.1.1
Inventory	Inventory are the network information, topology, and metadata that are about the Edge system environment and are stored in the Edge Inventory Manager component.	1.2.4.1
Manifest file	Manifest files are the files used in the application onboarding procedures. A manifest file contains an application package and a blueprint file.	1.2.1.1
VF image	VF images are the virtual functions providing an executing environment for the Edge Application. An application package is composed of one or more virtual functions called VF images.	1.2.1.1

1.2 Manageability and Orchestration Procedures

The details of the service and message design flow of each procedure are presented in this section. The term “application” means the Edge Applications, and the term “Context” means the application context, including user information.

The six lifecycle management and orchestration procedures are described in the following subsection in detail. Each subsection contains four necessary parts of a procedure: (1) an introduction, (2) conditions and involved information, (3) a detailed workflow, and (4) tables of the format and necessary data of the requests and responses.

1.2.1 Application On-boarding

The purpose of the application on-boarding procedure is to obtain the needed application software package and the corresponding application blueprint that describes how the application shall be deployed. The message flow of application onboarding is designed for the uploading and onboarding the necessary packages of the applications to the Edge system.

Operators and Service providers mentioned in this subclause are assumed either by the same or different people/groups/organizations. That is, if they are assumed by the same people, they don’t need to perform the query request (Step **b**) before the distribution request (Step **c**). It is also possible for the Edge Service Operators to get information of the uploaded manifest files, for example, directly from the Edge Service Providers.

The pre-condition of the procedure:

- The Edge Service Providers shall prepare the manifest files.

- Once they upload their manifest files, the Edge Service Operators are able to trigger the distribution among the EFO components.

The post-condition of the procedure:

- The artifacts are generated by the Edge App Designer and are distributed to the components in the EFO based on their functionality. The example can be found in 1.2.1.4.
- The application packages are saved to the VNI via the IM/VIM.

During the application onboarding procedures, the following files are involved:

- Manifest file: a manifest file that includes an application package and a blueprint file.
- Application package: a hierarchical file containing the necessary data and information of the application and its components, VF images. An application package is an installation file that will be deployed to a compute node to launch a specific Edge Application.
- VF image: a virtual function. An application package is composed of one or more virtual functions called VF images.
- Blueprint file: a file that includes the orchestration rules and the resource lifecycle management designs, etc. This file is used to configure the components in the EFO.
- Artifacts: all the files that are generated by the Edge App Designer component after the manifest file is passed to the Edge App Designer. They are the descriptor files for the components in the EFO and will be distributed to the other components in the EFO.

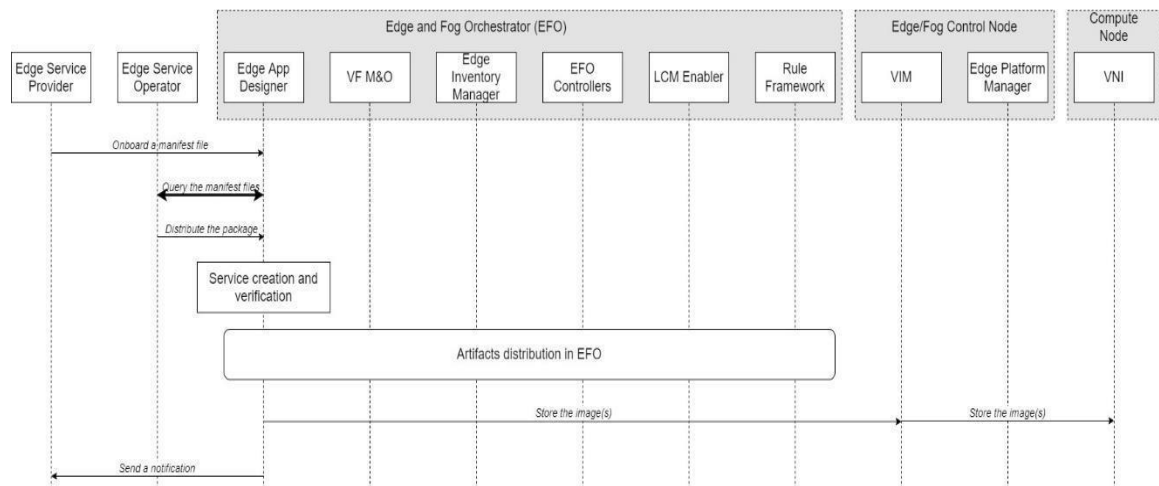


Figure 2—The workflow of application onboarding procedures

The detailed description of the flow is documented below:

- The Edge Service Provider on-boards a manifest file that includes an application package and a blueprint file via the Edge App Designer component of the EFO (Table 4). The Edge App designer checks if the Edge Service Provider is authorized to perform the uploading operation. A response shall be made if successful (Table 5).
- (Optional) the Edge Service Operator sends a query request (Table 6 and Table 7) to the Edge App Designer component to get the list of the uploaded application manifest files. Then the Edge Service Operator selects and filters these files based on their needs, and triggers the distribution of the packages and blueprint files.
- The Edge Service Operator sends a distribution request (Table 8) to trigger the distribution of the packages and blueprint files contained in the selected manifest file.
- The Edge App Designer component verifies the request, checking if the Edge Service Operator is authorized to perform the distribution operation and if the related procedures and the information are ready.

- e) If the request passes the authorization by the Edge App Designer component, the component starts to create the Edge Application. The Edge App Designer generates the artifacts as the output.
- f) All the artifacts are distributed to their responsible components in the EFO from the Edge App Designer component. The example can be found in 1.2.1.4.
- g) The application package is saved to the VNI which is responsible for storage via the corresponding IM/VIM.
- h) The Edge App Designer sends a notification to the Edge Service Operator for the success of the distribution (Table 9).

The following tables describe the elements contained in the requests and responses.

Table 4—Application manifest file uploading request

Element	Description
Service Provider ID	The Edge Service Provider identifier.
Manifest file	A manifest file that includes an application package and a blueprint file. It is the payload of the request.

Table 5—Application manifest file uploading response

Element	Description
Operation status	A 201 Created code shall be returned if successful. A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed.
Manifest file ID	The generated identifier of the uploaded manifest file. This field will be blank if the upload fails.

Table 6—Uploaded manifest files query request

Element	Description
Operator ID	The Edge Service Operator identifier. The ID shall be authorized by the EFO.
Query filter	The filter conditions for the query. Operators can only query the manifest files that satisfy these conditions and limitations.

Table 7—Uploaded manifest files query response

Element	Description
Operation status	A 200 OK code shall be returned if successful. A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed.
Query result	A list of the manifest files that satisfy the query conditions.

Table 8—Artifact distribution request

Element	Description
Operator ID	The Edge Service Operator identifier. The ID shall be authorized by the EFO.
Manifest file ID	The identifier of the manifest file which the Edge Service Operator supposes to distribute.

Table 9—Distribution success notification

Element	Description
Operator ID	The Edge Service Operator identifier.
Application ID	The packages are distributed successfully, and the application is given an identifier for the Edge Service Operators to instantiate it in the next subclause.

In addition, we describe an example of the process of artifact distribution. The artifact distribution is mentioned in Step f) of 1.2.1.2, whose purpose is to classify and send the relevant files to the corresponding component in the EFO.

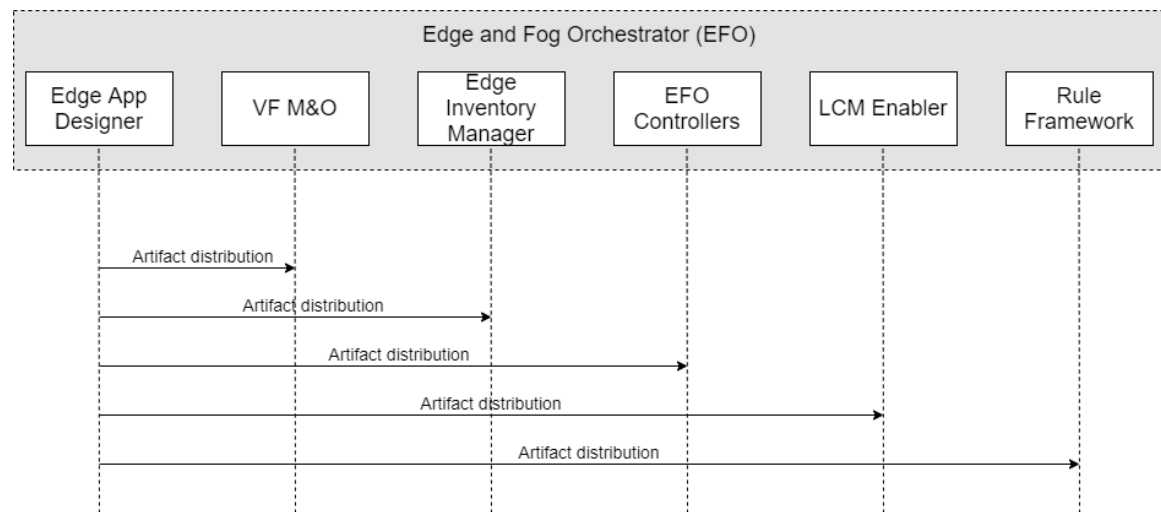


Figure 3—A schematic diagram of the workflow of artifact distribution

The resource lifecycle management designs will be distributed to the LCM Enabler component, and the orchestration rules will be distributed to the Rule Framework component. The VF M&O is responsible for the management and orchestration of Edge Applications in their run-time. EFO Controllers assist EFO to control and communicate with the entities in control nodes and compute nodes. Edge Inventory Manager stores the information and the topology of Edge Applications and the related environment. The related files contained in the manifest file will be distributed to the corresponding components according to their functionality.

1.2.2 Application Instantiation

The purpose of application instantiation is to launch the service (that is, the Edge Application) on a certain Edge Platform to serve the customers. The message flow of application instantiation and start-up is used to instantiate an application instance in the Edge/Fog system and have it properly configured.

The pre-condition of the procedure:

- The artifacts have been distributed to each component in the EFO.
- The application package has been saved to the VNI.

The post-condition of the procedure:

- The Edge Application has been instantiated following the steps described below.
- The inventory is created and updated based on the information of the created Edge Application instance.
- The Edge Platform has been configured to allocate its computing and networking resources properly.

During the application instantiation procedures, the following files are involved:

- Workload: the workload covers all the pre-work needed for the application to launch properly.
- The inventory: the network information, topology, and metadata that are about the Edge system environment and are stored in the Edge Inventory Manager component.

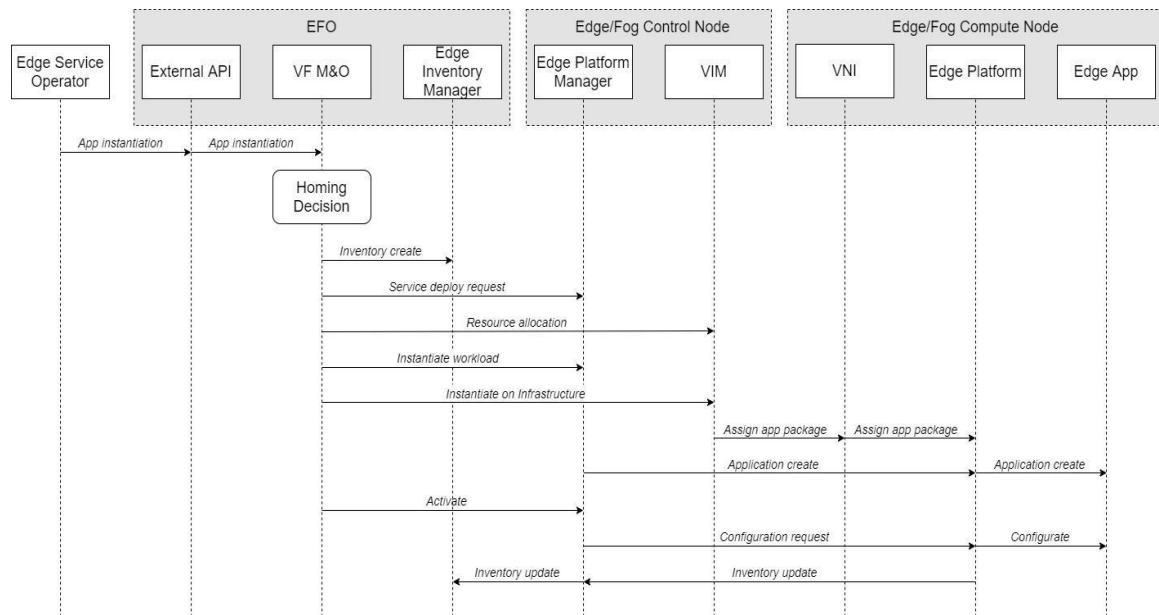


Figure 4—The workflow of application instantiation procedures

The workflow of the application instantiation procedures is documented below:

- a) The Edge Service Operator sends an application instantiation request (Table 10) to the VF M&O component through External API. A response shall be made (Table 1).
- b) The VF M&O makes the homing decision of the instance to be instantiated. The homing decision is to decide which Edge Platform the Edge Application will be instantiated on.
- c) The VF M&O creates the empty inventory in the Edge Inventory Manager (Table 11). The empty inventory will be fulfilled with real values after the Edge Application is created.
- d) The VF M&O sends a service deployment request to the target Edge Platform Manager (Table 13). A response shall be made if successful (Table 1).
- e) The VF M&O requests the IM/VIM to allocate resources, including computing resource and networking resource (Table 14). A response shall be made if successful (Table 1).
- f) The VF M&O requests the Edge Platform Manager entity to instantiate the workload (Table 15). A response shall be made if successful (Table 1).
- g) The IM/VIM deals with the instantiation work on the infrastructure (Table 16). In the process, the IM/VIM loads the needed application package from the VNI and deploy it onto the Edge Platform (Table 17 & Table 18). A response shall be made if successful (Table 1).
- h) The Edge Platform Manager informs the Edge Platform assigned by the EFO to create the Edge Application (Table 19). A response shall be made if successful (Table 1).
- i) The VF M&O requests the Edge Platform Manager to activate the instance (Table 20). The Edge Platform Manager informs the corresponding Edge Platform to deal with the configuration work on the instance (Table 21). A response shall be made if successful (Table 1).
- j) The Edge Platform returns the information to the Edge Platform Manager to update the inventory in Edge Inventory Manager (Table 12).

The following tables describe the elements contained in the requests and responses.

Table 10—Application instantiation request

Element	Description
Operator ID	The Edge Service Operator identifier. The ID shall be authorized by the EFO.
Application ID	The identifier of the application that the Edge Service Operator supposes to instantiate.

Table 11—Inventory update request from an Edge Platform to an Edge Platform Manager

Element	Description
Edge Platform ID	The Edge Platform identifier.
Application ID	The identifier of the application that has been instantiated successfully.
New inventory	The empty inventory which will be updated after instantiation finished.

Table 12—Inventory update request from an Edge Platform Manager to the Edge Inventory Manager in an EFO

Element	Description
Edge Platform Manager ID	The Edge Platform Manager identifier.
Updated inventory	The updated inventory. It shall fulfill the empty inventory that has been created with parameters and values of the real environment.

Table 13—Service deployment request

Element	Description
Edge Platform Manager ID	The Edge Platform Manager identifier.
Application ID	The identifier of the application that the Edge Service Operator supposes to instantiate.

Table 14—Resource allocation request

Element	Description
Edge Platform Manager ID	The Edge Platform Manager identifier.
VIM ID	The IM/VIM identifier.
Application ID	The identifier of the application that the Edge Service Operator supposes to instantiate.

Table 15—Application manifest file uploading request

Element	Description
Edge Platform Manager ID	The Edge Platform Manager identifier.
Application ID	The identifier of the application that the Edge Service Operator supposes to instantiate.

Table 16—Application manifest file uploading request

Element	Description
VIM ID	The IM/VIM identifier.
Application ID	The identifier of the application that the Edge Service Operator supposes to instantiate.

Table 17—Application manifest file uploading request

Element	Description
Application ID	The identifier of the application that the Edge Service Operator supposes to instantiate.
Edge Platform ID	The Edge Platform identifier that the IM/VIM tells the VNI to assign the application package to.

Table 18—Application manifest file uploading request

Element	Description
Application ID	The identifier of the application that the Edge Service Operator supposes to instantiate.
Application package	The application package for the target application to be instantiated.

Table 19—Application manifest file uploading request

Element	Description
Application ID	The identifier of the application that the Edge Service Operator supposes to instantiate.

Table 20—Application manifest file uploading request

Element	Description
Application ID	The identifier of the application that the Edge Service Operator supposes to instantiate.

Table 21—Application manifest file uploading request

Element	Description
Application ID	The identifier of the application that the Edge Service Operator supposes to instantiate.
Configuration details	The configuration of the Edge App. The details of the configuration vary depending on the system and the Edge App itself.

1.2.3 Application Context Creation

The purpose of context creation is to add necessary data and information to the system. The application context may be regarded as some information that helps the Edge Service Operators to make the decision, such as streaming resolution for each user in the case of video streaming application. If there is already some context in the system, the request is sent for the new context to join with the already existing one; if not, it is needed to instantiate a new space to store the context. For example, an End User updates its selected streaming resolution to the system.

In practice, it is possible for both end users and Edge Service Operators to do application context creation, and the procedures are different in these two cases. The End Users report and update their personal data regularly, and this data may be used to make some better decisions by the system and the Edge Service Operators. The Edge Service Operators are also able to recreate or add the information according to their own needs, like they may update the user data to reconfigure the performance of the Edge Application every season. The procedures for end users and for Edge Service Operators are described below respectively.

The pre-condition of the procedure:

- The Edge App has been instantiated.
- The inventory has been created and updated in the Edge Inventory Manager.

The post-condition of the procedure:

- The application context has been added and stored into the Edge Application.

During the application context creation procedures, the following files are involved:

- Application context: Application context involves all the files that contain the custom information operators need, such as user information and statistical data. The application context also includes the data that can assist the Edge Service Operators to make the decision.
- The inventory: the network information, topology, and metadata that are about the Edge system environment and are stored in the Edge Inventory Manager component.

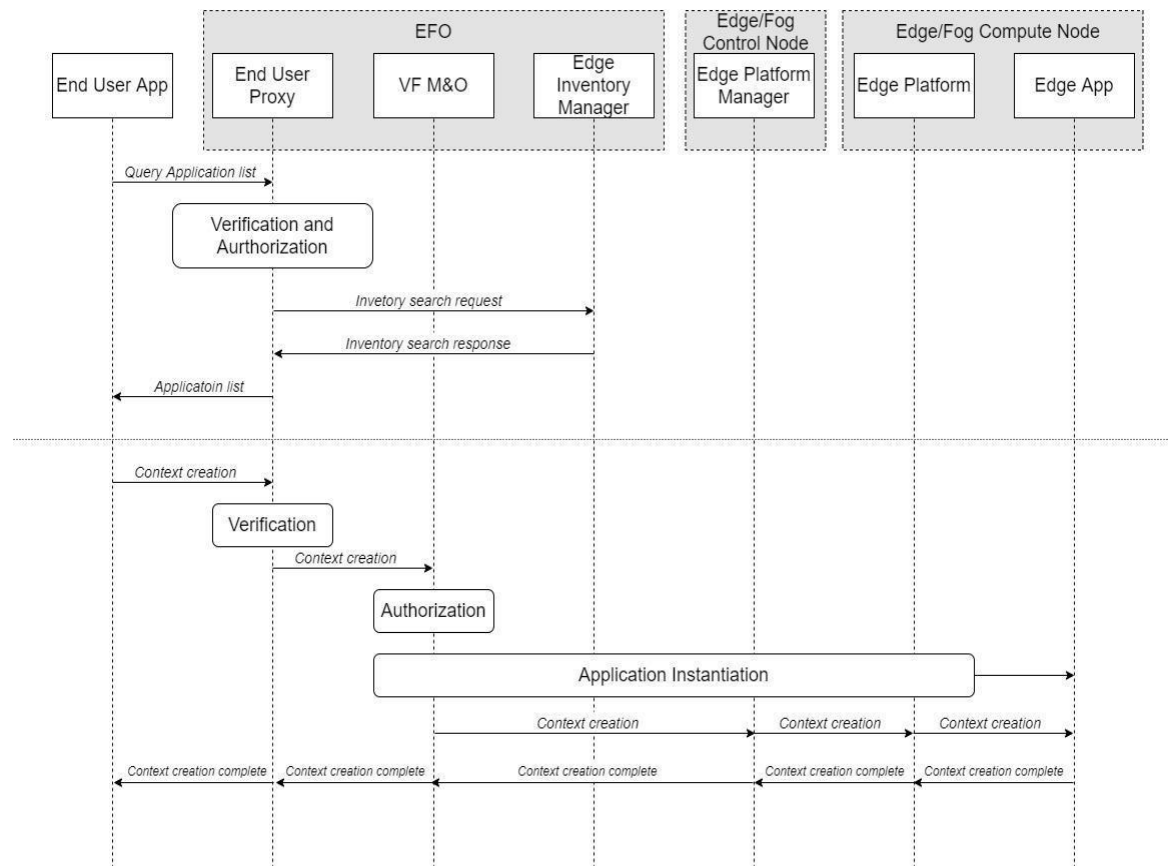


Figure 5—the workflow of application context creation procedures

The procedures of application context creation for end users are as the following:

- a) The End User App sends the application list query request (Table 22) to the End User Proxy (EUP).
- b) Upon receiving the request, the EUP verifies who sends this request and for what purpose, and authorizes the request if it is valid; that is, from a valid end user and with valid request content.
- c) If the request is valid, the EUP retrieves the list of Edge applications available to the end user app from the Edge Inventory Manager in the EFO (Table 23 & Table 24).
- d) The EUP sends the query results to the End User App (Table 25). The query results shall include a list of all applications that satisfy the end user's requirements.
- e) The End User App sends the application context creation request (Table 26) to the EUP, which contains the application context to be created.
- f) The EUP verifies the request from the user app to check who sends this request and for what purpose, and then forwards the request to the VF M&O in the EFO.
- g) The VF M&O grants the request, and if necessary (for example, there is no application available in a nearby area), does the application instantiation with the other components in the EFO and the Edge Platform Manager. The details of instantiation are the same as described in 1.2.2.2.
- h) The EFO forwards the context creation request (Table 26) to the instance, and the instance registers the user information inside.
- i) The context creation complete response is replied to the end user app (Table 27).

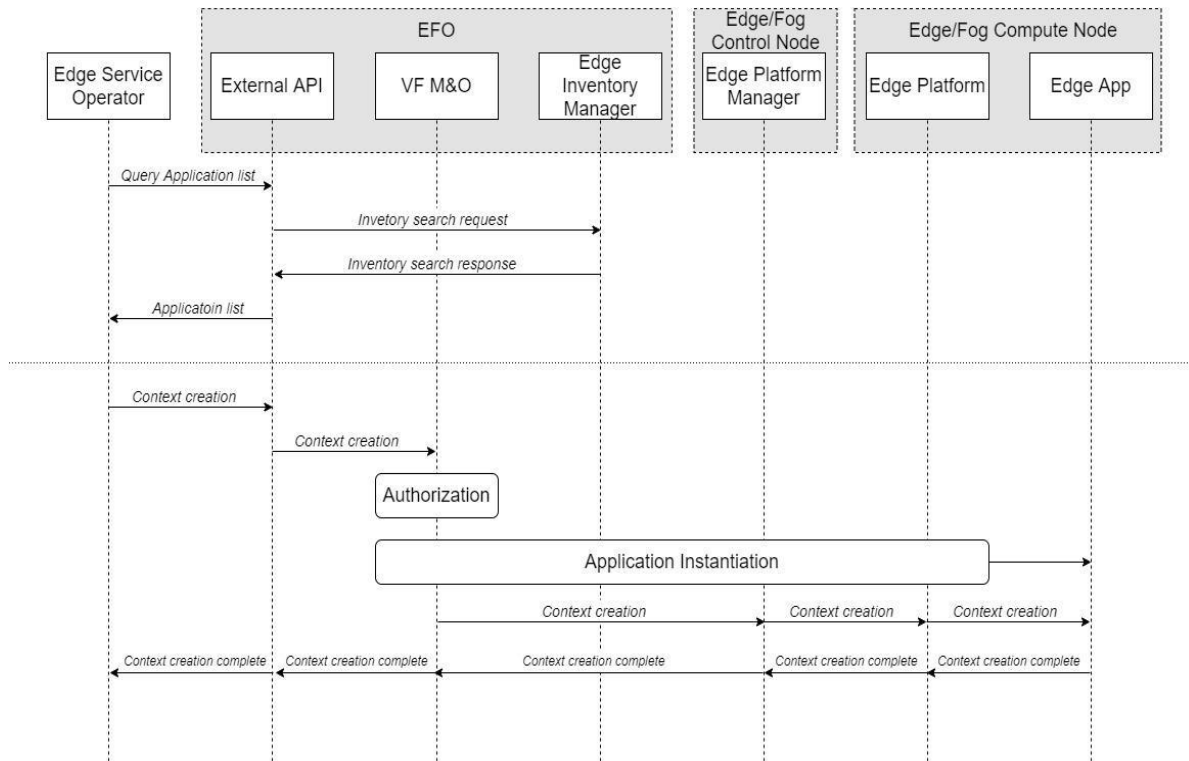


Figure 6—the workflow of application context creation procedures

The procedures of application context creation for Edge Service Operators are as the following:

- a) The Edge Service Operator sends the application list query request (Table 22) to the Edge Inventory Manager via the external API.
- b) The request retrieves the list of Edge applications available to the end user app from the Edge Inventory Manager in the EFO (Table 23 & Table 24).
- c) The query results are sent to the Edge Service Operator (Table 25). The query results shall include a list of all applications that satisfy the end user’s requirements.
- d) The Edge Service Operator sends the application context creation request (Table 26) to the VF M&O via the external API, which contains the application context to be created.
- e) The VF M&O grants the request, and if necessary (for example, there is no application available in a nearby area), does the application instantiation with the other components in the EFO and the Edge Platform Manager. The details of instantiation are the same as described in 1.2.2.2.
- f) The EFO forwards the context creation request (Table 26) to the instance, and the instance registers the user information inside.
- g) The context creation complete response is replied to the end user app (Table 27).

The following tables describe the elements contained in the requests and responses.

Table 22—Application list query request

Element	Description
End User App ID	The end user app identifier. The ID shall be authorized by the EFO.
Query filter	The filter conditions for the query. End user apps can only query the applications that satisfy these conditions and limitations.

Table 23—Inventory search request

Element	Description
Query ID	The query identifier.
Query filter	The filter conditions for the query. Operators can only query the manifest files that satisfy these conditions and limitations.

Table 24—Inventory search response

Element	Description
Query identifier	The query identifier.
Operation status	A 200 OK code shall be returned if successful. A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed.
Query result	A list of the manifest files that satisfy the query conditions.

Table 25—Application list query response

Element	Description
Operation status	A 200 OK code shall be returned if successful. A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed.
Query results	A list of the applications that satisfy the query conditions.

Table 26—Application context creation request

Element	Description
End User App ID	The end user app identifier. The ID shall be authorized by the EFO.
Application ID	The identifier of the application that the End User App supposes to do context creation.

Table 27—Application context creation response

Element	Description
Operation status	A 200 OK code shall be returned if successful. A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed.
Application ID	The identifier of the application that the End User App supposes to do context creation.

1.2.4 Application Context Deletion

The purpose of application context deletion is to delete the application context which is added or collected in the process of application context creation. The application context may be regarded as some information that helps the Edge Service Operators to make the decision, such as streaming resolution for each user in the case of video streaming application. As context creation, it is possible for both end users and Edge Service Operators to do application context deletion operation. For example, the Edge Service Operators may clean up the user data stored in the Edge/Fog system every season. The procedures for end users and for Edge Service Operators are described below respectively.

The pre-condition of the procedure:

- The Edge App has been instantiated.
- The inventory has been created and updated in the Edge Inventory Manager.
- The application context has been created in the target Edge Application.

The post-condition of the procedure:

- The application context has been removed from the system.

During the application context deletion procedures, the following files are involved:

- Application context: Application context involves all the files that contain the custom information operators need, such as user information and statistical data.
- The inventory: the network information, topology, and metadata that are about the Edge system environment and are stored in the Edge Inventory Manager component.

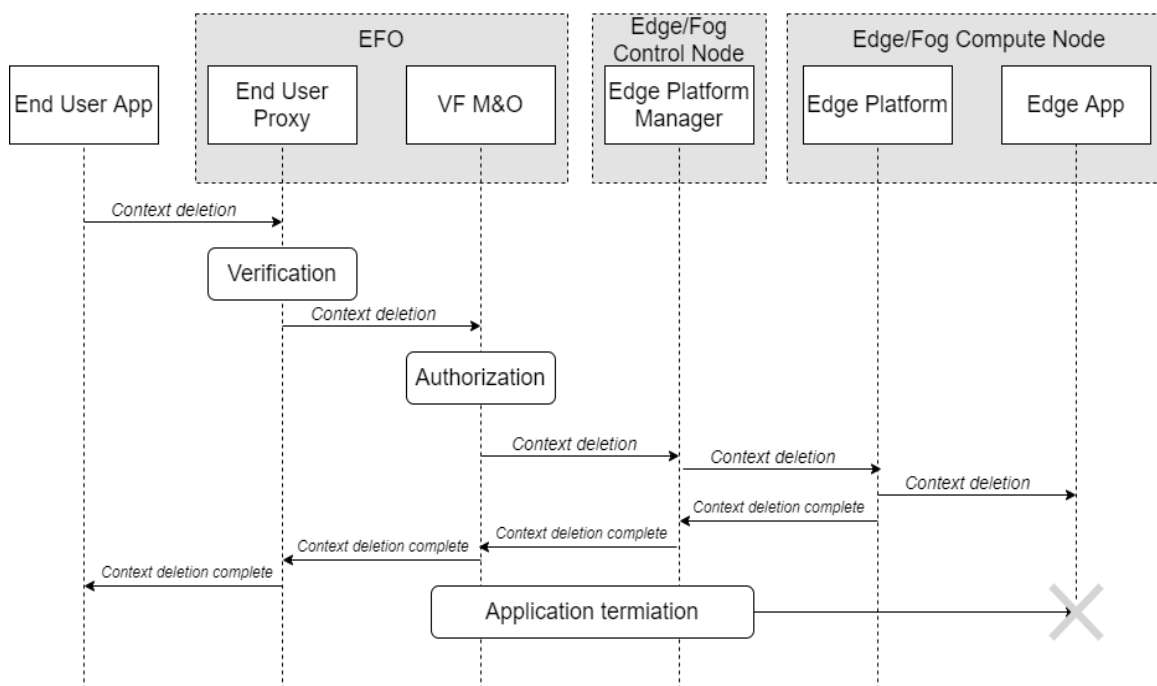


Figure 7—the workflow of application context deletion procedures

The procedures of application context deletion for end users are as the following:

- The end user app sends the application context deletion request (Table 28) to the End User Proxy (EUP), which contains the application context to be deleted.
- The EUP verifies the request from the end user app to check who sends this request and for what purpose, and then forwards the request to the VF M&O.
- The VF M&O grants the request and forwards the context deletion request to the instance, and the instance erases the related user information.
- The context creation complete response is returned to the end user app (Table 29).
- (Optional) If necessary, the EFO and the corresponding entities do the application termination as mentioned in 1.2.5.2.

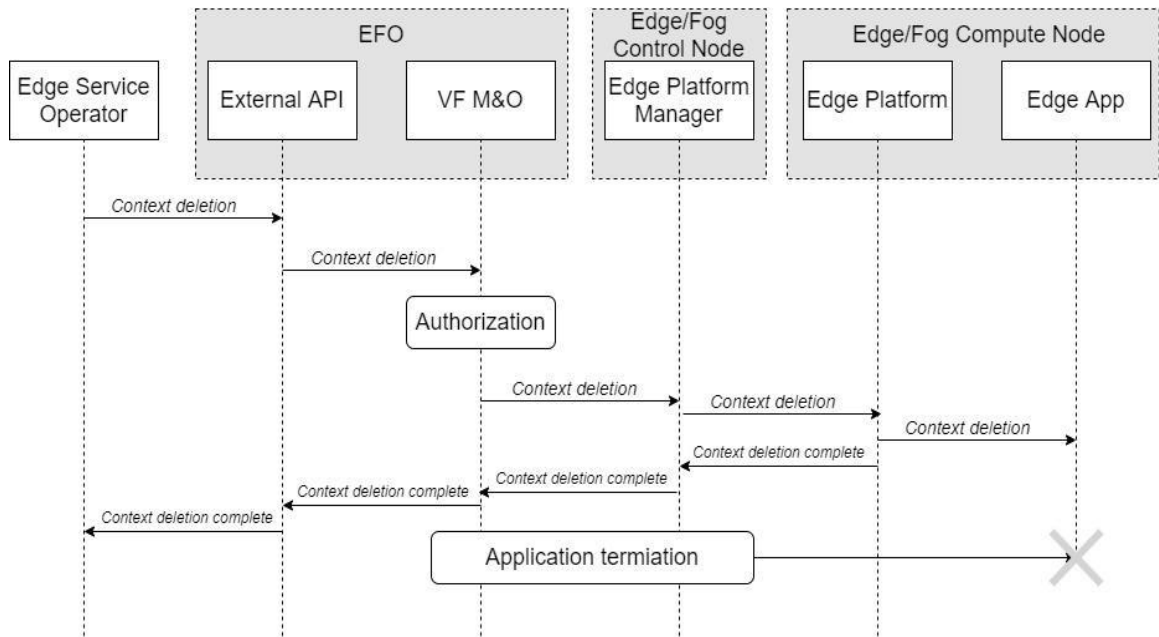


Figure 8—The workflow of application context deletion procedures

The procedures of application context deletion for Edge Service Operators are as the following:

- The end user app sends the application context deletion request (Table 28) to the VF M&O via the external API, which contains the application context to be deleted.
- The VF M&O grants the request and forwards the context deletion request to the instance, and the instance erases the related user information.
- The context creation complete response is returned to the Edge Service Operator (Table 29).
- (Optional) If necessary, the EFO and the corresponding entities do the application termination as mentioned in 1.2.5.2.

The following tables describe the elements contained in the requests and responses.

Table 28—Application context deletion request

Element	Description
End User App ID	The end user app identifier. The ID shall be authorized by the EFO.
Application ID	The identifier of the application that the End User App supposes to do context deletion.

Table 29—Application context deletion response

Element	Description
Operation status	A 200 OK code shall be returned if successful. A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed.
Application ID	The identifier of the application that the End User App supposes to do context deletion.

1.2.5 Application Termination

The purpose of application termination is to terminate an application instance in the Edge/Fog system. If the application is no longer used, that is, if there is no user in the region accessing the application, the Edge Service Operator can terminate and remove the application from the control and compute nodes.

The pre-condition of the procedure:

- The Edge App has been instantiated.
- The inventory has been created and updated in the Edge Inventory Manager.

The post-condition of the procedure:

- The Edge App has been terminated.
- The related inventory is removed from the Edge Inventory Manager.
- The artifacts, which are distributed to each component in the process of on-boarding, are still saved in each component in the EFO.

During the application termination procedures, the following files are involved:

- Workload: all the work that needs to be done before the Edge Application instance actually starts up.

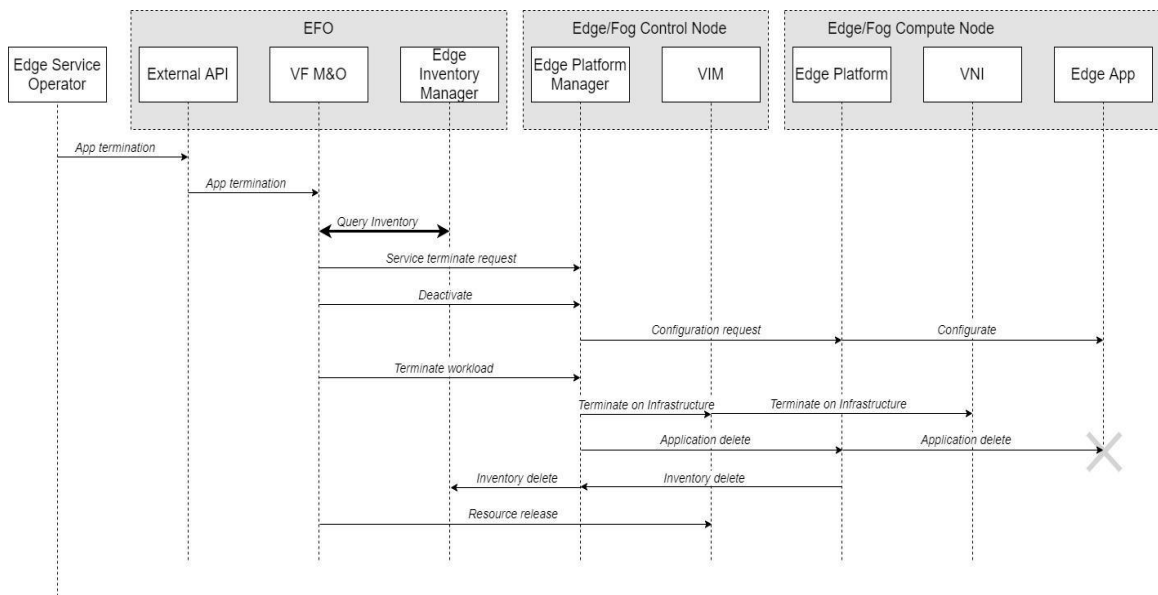


Figure 9—The workflow of application termination procedures

The detailed description of the application termination workflow is documented below:

- The Edge Service Operator sends an application termination request (Table 30) to the VF M&O component in the EFO through the External API.
- The VF M&O retrieves the location of the instance from Edge Inventory Manager (Table 31 & Table 32).
- The VF M&O sends service terminate request (Table 30) to the target Edge Platform Manager. A response shall be made if successful (Table 1).
- The VF M&O requests the Edge Platform Manager to deactivate the instance (Table 33).
- The Edge Platform Manager informs the corresponding Edge Platform to do the configuration work to the Edge Application (Table 34). The configuration may include the necessary settings which shall be made before the Edge Application is actually shut down, such as terminate the internal VFs in order. A response shall be made if successful (Table 1).
- The VF M&O requests the Edge Platform Manager to terminate the workload (Table 33).
- The Edge Platform Manager informs the IM/VIM to deal with the termination work on the infrastructure (Table 35). The IM/VIM terminates the related VNI according to the request. A

response shall be made if successful (Table 1). Note that the default response is also generated and sent to the VF M&O.

- h) The Edge Platform Manager informs the Edge Platform to actually terminate and delete the Edge Application (Table 36). A response shall be made if successful (Table 1).
- i) The Edge Platform Manager returns the information to update the inventory in Edge Inventory Manager component (Table 37).
- j) The VF M&O requests the IM/VIM to release corresponding resources, including computing and networking resources (Table 38). A response shall be made if successful (Table 1).

The following tables describe the elements contained in the requests and responses.

Table 30—Application termination request

Element	Description
Operator ID	The Edge Service Operator identifier. The ID shall be authorized by the EFO.
Application ID	The identifier of the application that the Edge Service Operator supposes to terminate.

Table 31 —Instance location retrieval request

Element	Description
Query ID	The query identifier.
Query filter	The filter conditions for the query. Operators can only query the manifest files that satisfy these conditions and limitations. In the case, there shall be only one Edge Application that satisfies the conditions.

Table 32—Instance location retrieval response

Element	Description
Query identifier	The query identifier.
Operation status	A 200 OK code shall be returned if successful. A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed.
Query result	The location information of the target Edge Application.

Table 33 —Deactivation and workload termination request

Element	Description
Application ID	The identifier of the application that the Edge Service Operator supposes to terminate. The Edge Platform Manager will deactivate and terminate the related workload according to the request.

Table 34 —Edge App configuration request

Element	Description
Application ID	The identifier of the application that the Edge Service Operator supposes to terminate.
Configuration details	The configuration of the Edge App. The details of the configuration vary depending on the system and the Edge App itself.

Table 35 —Infrastructure termination request

Element	Description
Application ID	The identifier of the application that the Edge Service Operator supposes to terminate. The IM/VIM will terminate the use of infrastructure according to the request.

Table 36—Application termination and deletion request

Element	Description
Application ID	The identifier of the application that the Edge Service Operator supposes to terminate. The Edge Platform will terminate and delete the Edge Application according to the request.

Table 37—Inventory update request

Element	Description
Edge Platform Manager ID	The Edge Platform Manager identifier.
Updated inventory	The updated inventory. After application termination, the Edge Application will be no longer available and thus the connections to its end users will be cut off, and some Edge Platform may be idle now. All this information needs to be updated.

Table 38—Resource release request

Element	Description
Application ID	The identifier of the Edge Application that has been terminated.

1.2.6 Application Configuration/Re-Configuration

The purpose of application reconfiguration is to update the settings of the Edge Applications in the Edge system. The message flow of application reconfiguration is used to update the settings and configuration of the target Edge Application.

The pre-condition of the procedure:

- The Edge App has been instantiated.
- The inventory has been created and updated in the Edge Inventory Manager.

The post-condition of the procedure:

- The configurations of IM/VIM, Edge Platform, and Edge App have been updated.
- The inventory in the Edge Inventory Manager has been updated based on the reconfiguration result.

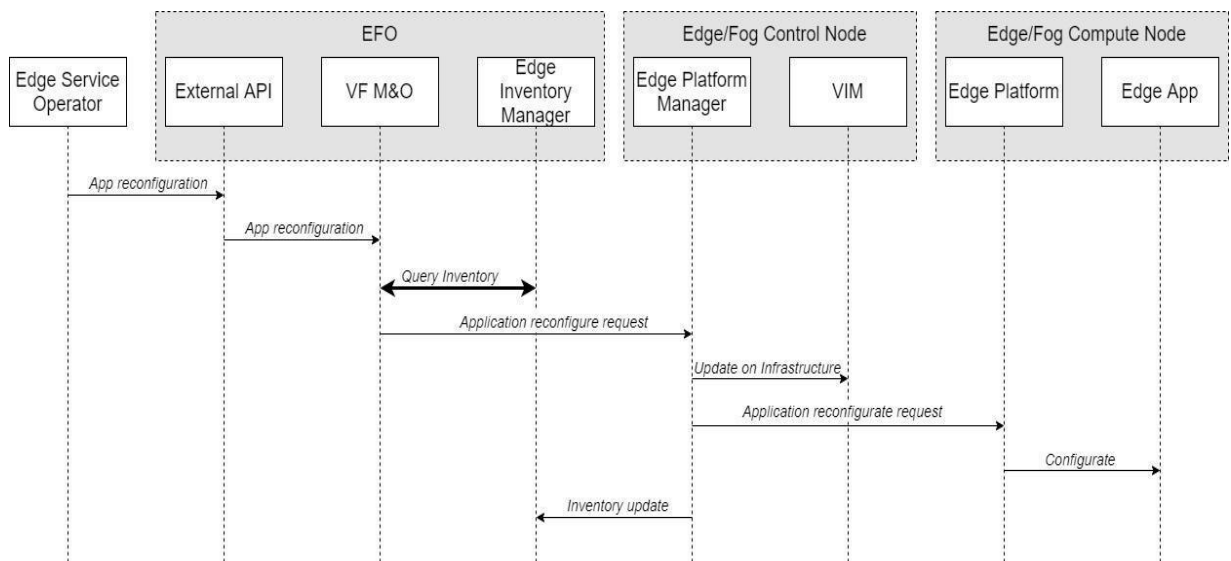


Figure 10—The workflow of application reconfiguration procedures

The detailed description of the application reconfiguration workflow is documented below:

- a) The Edge Service Operator sends an application reconfiguration request (Table 39) to the VF M&O component in the Edge and Fog Orchestrator (EFO) through the External API.
- b) The VF M&O retrieves the location of the instance from the Edge Inventory Manager (Table 40 & Table 41).
- c) The VF M&O sends an application reconfiguration request to the target Edge Platform Manager (Table 42). A response shall be made if successful (Table 1).
- d) The Edge Platform Manager informs the IM/VIM to update the configuration on the infrastructure (Table 42). A response shall be made if successful (Table 1).
- e) The Edge Platform Manager requests the corresponding Edge Platform to configure the target Edge application (Table 43). A response shall be made if successful (Table 1).
- f) (Optional) If there is any change in the system inventory, the Edge Platform Manager returns the information to update the inventory in the Edge Inventory Manager component (Table 44).

The following tables describe the elements contained in the requests and responses.

Table 39—Application termination request

Element	Description
Operator ID	The Edge Service Operator identifier. The ID shall be authorized by the EFO.
Application ID	The identifier of the application that the Edge Service Operator supposes to reconfigure.
New configuration	The new settings or the new configuration files of the application.

Table 40—Instance location retrieval request

Element	Description
Query ID	The query identifier.
Query filter	The filter conditions for the query. Operators can only query the manifest files that satisfy these conditions and limitations. In the case, there shall be only one Edge Application that satisfies the conditions.

Table 41—Instance location retrieval response

Element	Description
Query identifier	The query identifier.
Operation status	A 200 OK code shall be returned if successful. A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed.
Query result	The location information of the target Edge Application.

Table 42—Application termination request

Element	Description
Application ID	The identifier of the application that the Edge Service Operator supposes to reconfigure.
New configuration	The new settings or the new configuration files of the application.

Table 43—Application termination request

Element	Description
Edge Platform Manager ID	The Edge Platform Manager identifier.
Application ID	The identifier of the application that the Edge Service Operator supposes to reconfigure.
New configuration	The new settings or the new configuration files of the application.

Table 44—Inventory update request

Element	Description
Edge Platform Manager ID	The Edge Platform Manager identifier.
New inventory	The inventory which is updated after the application reconfiguration.

1.3 Summary

The boom in edge computing technology is complemented by the development of many applications. But the growing diversity of application scenarios also raises the question of how to design systems and operation workflows that can accommodate a large number of different use cases, which may have contrasting requirements and target users. In this context, it is particularly important to standardize the management procedures and operational instructions for applications running on the Edge/Fog systems.

This document provides a complete description of application lifecycle management procedures for Edge/Fog systems. It contains all six principal operations necessary in life cycles of Edge applications, application onboarding, application instantiation, context creation, application reconfiguration, context deletion, and application termination. The step-by-step workflow, involved information, as well as the specification of relevant requests and responses for each of these operations is stated in each sub-clause.

With the help of this document, practitioners can more quickly establish a consistent and reliable Edge/Fog system and ensure their Edge Applications to function properly. The uniform specifications introduced in the document should help the regulation and integration of Edge/Fog systems and further improve the development of related industries.