Insights for Edge Software Developers

Presented by: **Lijuan Chen**
(ZTE Corporation, ETSI
MEC delegate)

For: **everyone**

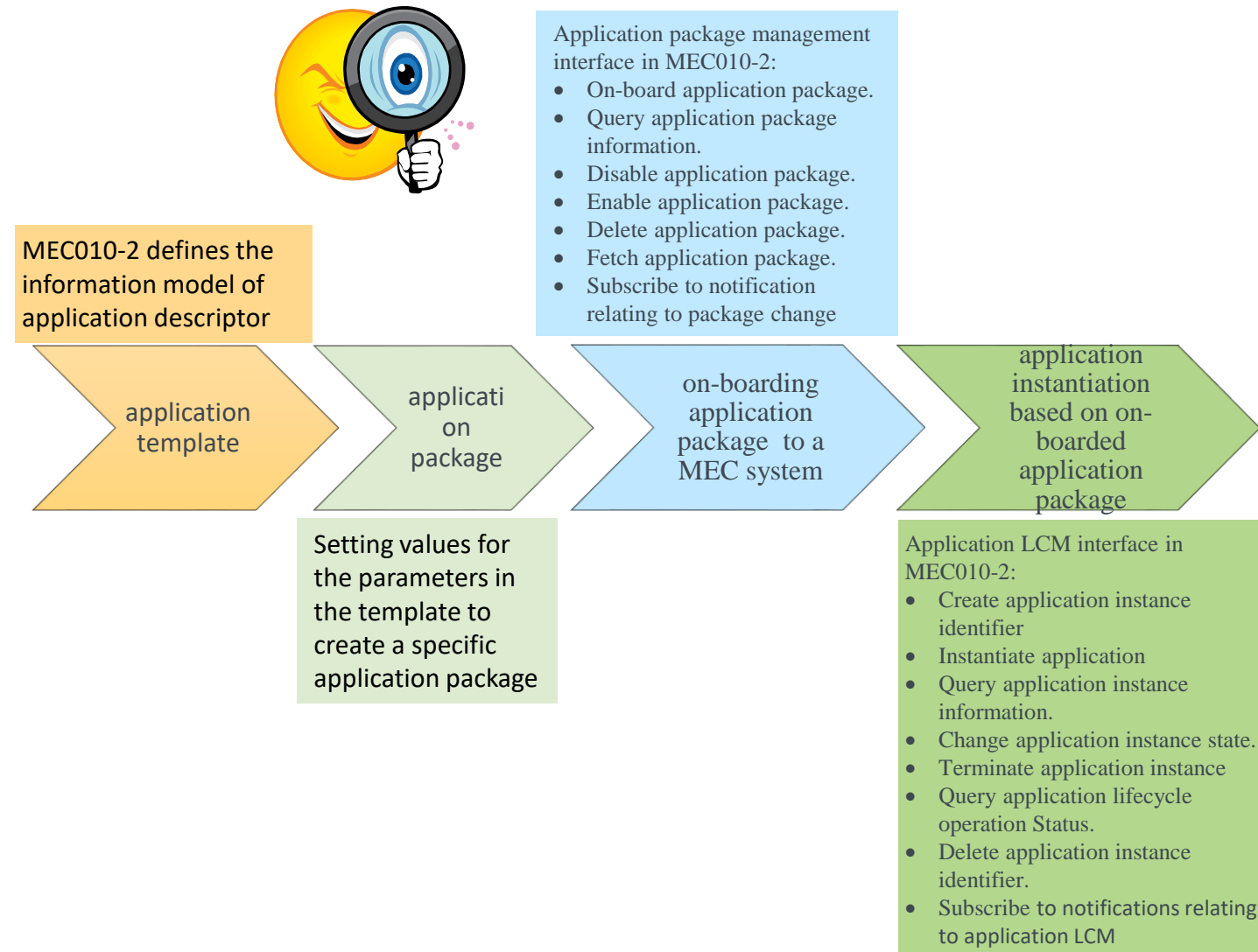*Episode* **#13 –** **Application lifecycle, rules and requirements management**

In this episode …

- ## We will learn:

  As a client (application provider),

  - ## Why a MEC application descriptor and package needs to be provided.

  - ## How to use the MEC application package management interface

  - ## How to use the MEC application instance lifecycle management interface

MEC010-2 defines the information model of application descriptor

Application package management interface in MEC010-2:
- On-board application package.
- Query application package information.
- Disable application package.
- Enable application package.
- Delete application package.
- Fetch application package.
- Subscribe to notification relating to package change

application template

application package

Setting values for the parameters in the template to create a specific application package

on-boarding application package to a MEC system

application instantiation based on on-boarded application package

Application LCM interface in MEC010-2:
- Create application instance identifier
- Instantiate application
- Query application instance information.
- Change application instance state.
- Terminate application instance
- Query application lifecycle operation Status.
- Delete application instance identifier.
- Subscribe to notifications relating to application LCM

ETSI MEC management specifications

Part 1: "System, host and platform management"
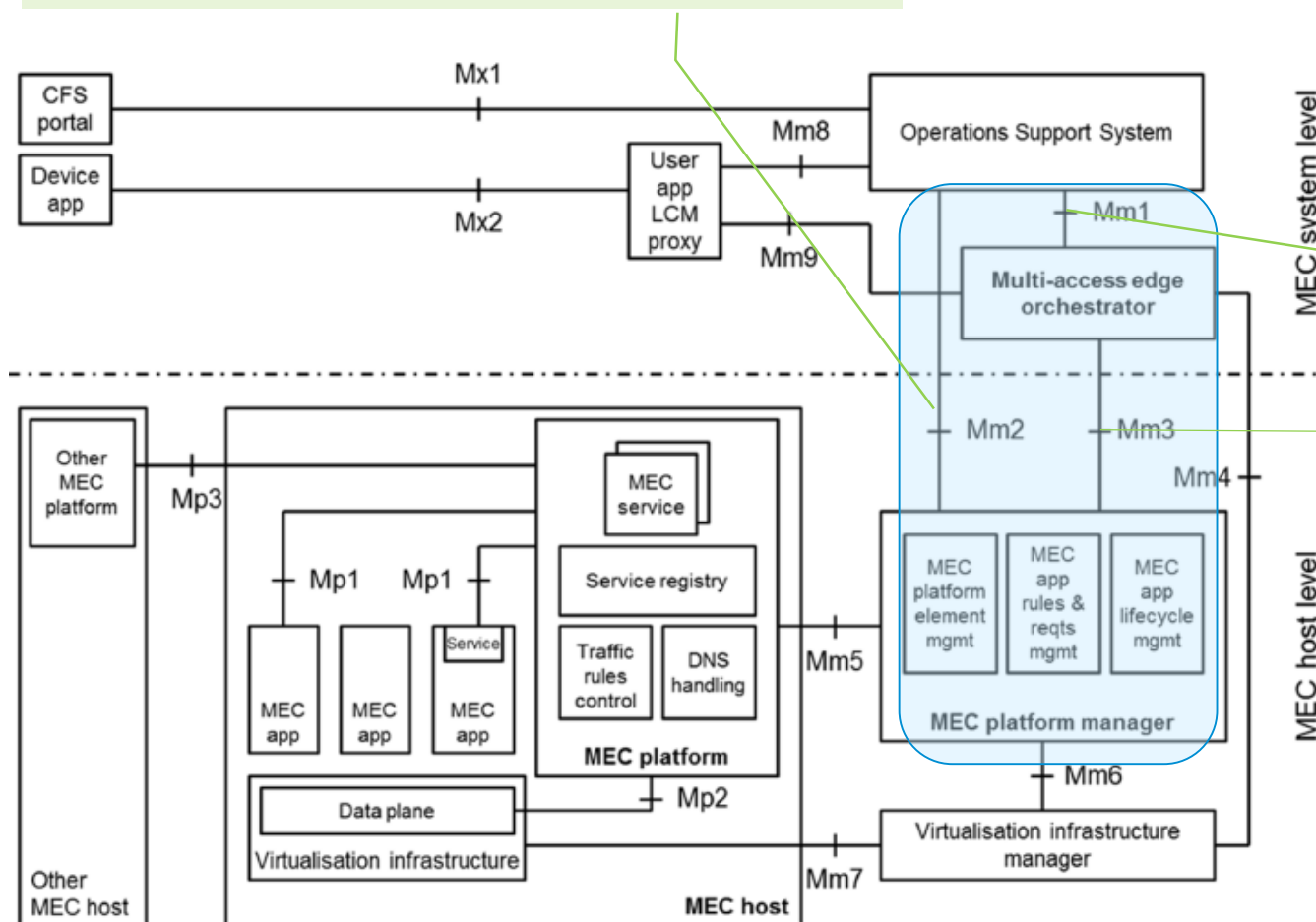
Part 2: "Application lifecycle, rules and requirements management".

3

As a start point of using MEC management system, the client needs to set values of the parameters in the template to create a specific application package.

Follow specifications:
- ETSI MEC010-2 defines the information model of application descriptor.
- ETSI MEC037 defines Application Package and Descriptor Format.

application package

- package metadata
  - manifest
  - application descriptor
    - requirements of a MEC application
    - application rules
- artifacts or URIs to artifacts
  - software image(s)
  - optionally other files

Table 6.2.1.2.2-1: Attributes of AppD

Attribute name	Cardinality	Data type	Description
appDId	1	String	Identifier of this MEC application descriptor. This attribute shall be globally unique. See note 1.
appName	1	String	Name to identify the MEC application.
appProvider	1	String	Provider of the application and of the AppD.
appSoftVersion	1	String	Identifies the version of software of the MEC application.
appDVersion	1	String	Identifies the version of the application descriptor.
mecVersion	1	String	Identifies version(s) of MEC system compatible with the MEC application described in this version of the AppD. The value shall be formatted as comma-separated list of strings. Each entry shall have the format <x>.<y>.<z> where <x>, <y> and <z> are decimal numbers representing the version of the present document. Whitespace between list entries shall be trimmed before validation.
appInfoName	0..1	String	Human readable name for the MEC application.
appDescription	1	String	Human readable description of the MEC application.
virtualComputeDescriptor	0..1	VirtualComputeDescriptor	Describes CPU and memory requirements, as well as optional additional requirements, such as disk and acceleration related capabilities, of the single VM used to realize this MEC application. See note 5.
osContainerDescriptor	0..N	OsContainerDescriptor	Describes CPU, memory requirements and limits, and software images of the OS Containers realizing this MEC application corresponding to OS Containers sharing the same host and same network namespace. See notes 5 and 7.
swImageDescriptor	1..N	SwImageDescriptor	Describes the descriptors of the software image to be used by the virtualisation container used to realize this MEC application. See note 5.
virtualStorageDescriptor	0..N	VirtualStorageDescriptor	Defines descriptors of virtual storage resources to be used by the MEC application.
appExtCpd	1..N	AppExternalCpd	Describes external interface(s) exposed by this MEC application. See note 4.
appServiceRequired	0..N	ServiceDependency	Describes services a MEC application requires to run.
appServiceOptional	0..N	ServiceDependency	Describes services a MEC application may use if available.
appServiceProduced	0..N	ServiceDescriptor	Describes services a MEC application is able to produce to the platform or other MEC applications. Only relevant for service-producing apps.
appFeatureRequired	0..N	FeatureDependency	Describes features a MEC application requires to run.
appFeatureOptional	0..N	FeatureDependency	Describes features a MEC application may use if available.
transportDependencies	0..N	TransportDependency	Transports, if any, that this application requires to be provided by the platform. These transports will be used by the application to deliver services provided by this application. Only relevant for service-producing apps. See note 2.
appTrafficRule	0..N	TrafficRuleDescriptor	Describes traffic rules the MEC application requires.
appDNSRule	0..N	DNSRuleDescriptor	Describes DNS rules the MEC application

4

2 – Uploading MEC application package and deploy application

After uploading a MEC application package to the MEO via OSS, and enabling it, the client can deploy an application instance based on the application package.
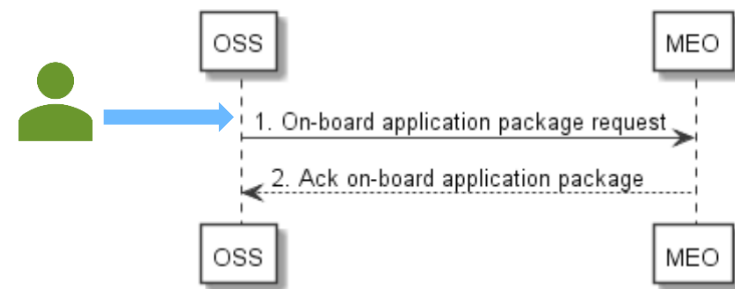

Figure 5.2.2-1: On-board application package

MEO checks the application package, e.g. the MEO checks for the existence of mandatory elements within the application package, validates the authenticity and integrity of the application package
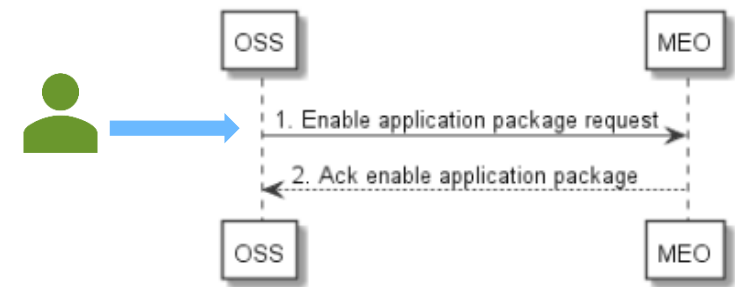

Figure 5.2.5-1: Enable application package

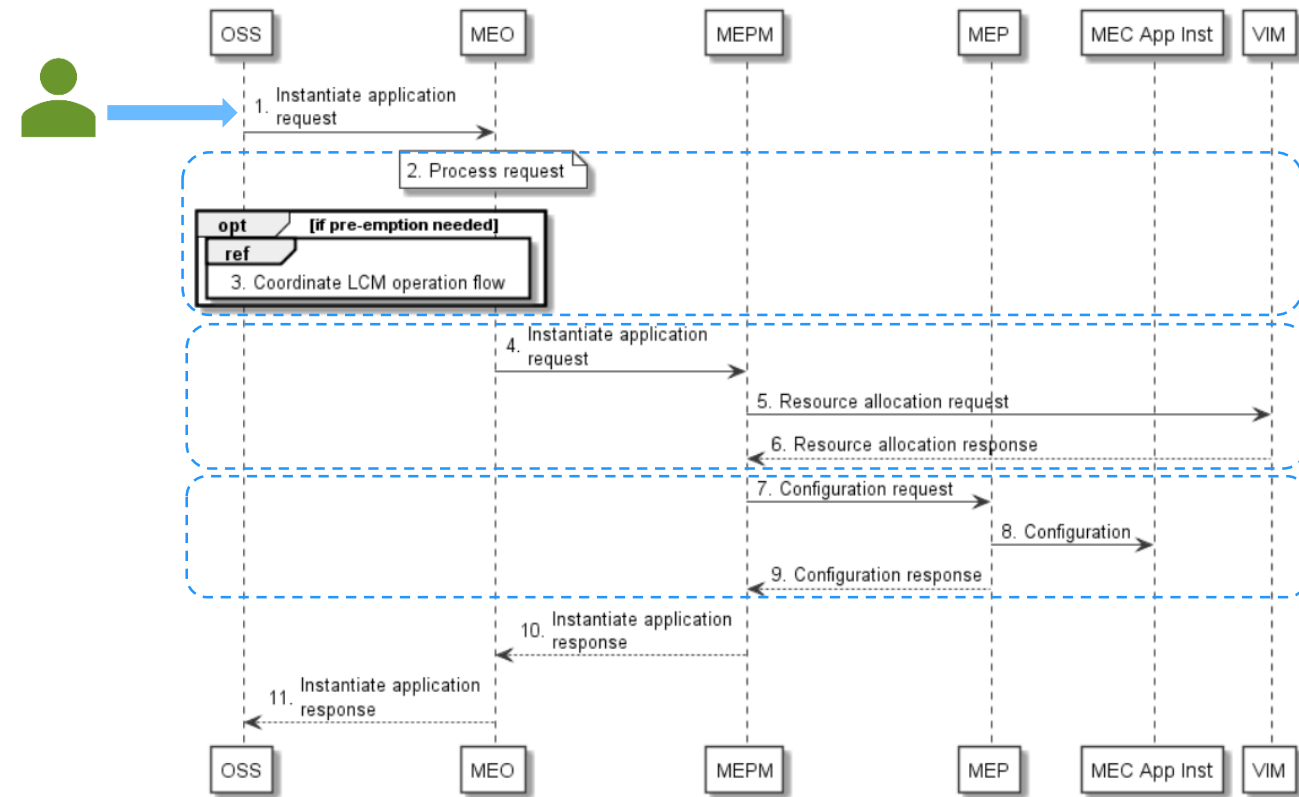MEO marks an application package as enabled in the MEC system

Figure 5.3.1-1: Application instantiation flow

Other MEC management interfaces

Application package management interface:

- Fetch application package.
- Query application package information.
- Subscribe to notification relating to package change
- Disable application package.
- Delete application package.

Application instance management interface:

- Query application instance information.
- Change application instance state.
- Query application lifecycle operation Status.
- Terminate application instance
- Delete application instance identifier.
- Subscribe to notifications relating to application LCM

Conclusions and further resources

What we have learnt:

- The definition of MEC application descriptor and package.

- How to on-board MEC application package and instantiate MEC application instance

- Specified application package and instance lifecycle management interfaces

Interested to learn more?

- Look for yourself at the available MEC010-2 APIs at forge.etsi.org/rep/mec

- Episode #3 for interaction between MEP and application instance

- Follow also the [next episodes of the MEC TECH Series](#) ☺

Enjoy the

https://mecwiki.etsi.org/index.php?title=MEC_Tech_Series