

# LSO API Release Framework

## Table of Contents

- [LSO SDKs](#)
- [Overview](#)
- [Type of Release](#)
- [SDK artefacts](#)
  - [Business Requirements and Use Cases](#)
  - [API Developer Guide](#)
  - [API definitions - Swagger specifications](#)
  - [Product Specifications](#)
  - [Postman scripts](#)
  - [White Papers](#)
- [Supporting artefacts](#)
  - [Example code](#)
  - [DEV Test Service](#)
  - [Certification](#)
- [LSO Reference Points](#)
- [NETCONF and YANG Info](#)
  - [YouTube Videos](#)

## LSO SDKs

The collection of SDKs (Software Developer Kits) made available by MEF. This collection has one SDK per LSO Interface Reference Point, and currently includes:

[LSO Sonata SDK Homepage](#)   [LSO Cantata SDK Homepage](#)   [LSO Legato SDK Homepage](#)   [LSO Presto SDK Homepage](#)

A new release of each LSO SDK is made available every 6 months.

## Overview

### Overview

The MEF facilitates the development of open standardized APIs in accordance with the [LSO Reference Architecture defined in MEF 55](#). There are seven LSO Reference Points described in MEF 55.

- [LSO Sonata](#)
- [LSO Interlude](#)
- [LSO Legato](#)
- [LSO Presto](#)
- [LSO Cantata](#)
- [LSO Allegro](#)
- [LSO Adagio](#)

Each LSO Reference Point is the location of one or more APIs defined by MEF. The APIs and associated development tools for each LSO Reference Point are collected together in one SDK for that LSO Reference Point.

Each LSO SDK has its own developer community associated with it, in which developers can ask questions, make contributions under Apache 2.0 licensing, and propose new features to be put in the backlog for the next release.

## Overview

# Type of Release

MEF is providing a release every 6 months.

Particular artefacts of these releases are on different level of maturity.

The maturity of standard documents is indicated by their status at the first page. It's either a Working Draft, Draft Standard or Published Standard.

The schema files follow the semantic numbering, followed by an "-rc" suffix, where applicable.

"rc" stands for release candidate which means that the artefact is not part of a published standard yet and is a subject to change.

## SDK artefacts

The deliverables available in the LSO SDK include both the software artefacts and the documentation. The SDK contains published and draft standards covering business requirements, use cases, and attributes that serve as the basis for the associated LSO APIs and data models. LSO APIs combine service-agnostic TM Forum Open APIs with MEF 3.0 service definitions.

### Business Requirements and Use Cases

Business Requirements and Use Cases document is the MEF standard that defines the business process flows, use cases, scenarios, attributes, and state machine diagrams.

### API Developer Guide

API Developer Guide is a MEF standard specification that assists implementation of functions defined at particular Interface. This includes API definitions in Swagger 2.0 or OpenAPI 3.0 format and accompanying documentation. The document provides information on interface capabilities, operations, data models, life cycle, etc. It presents common scenarios of API usage, example interactions and implementation guidelines.

### API definitions - Swagger specifications

The Interface Implementation Specification includes the product-agnostic API definitions for functions defined at the interface. The API schemas are defined using OpenAPI 2.0 (Swagger) or OpenAPI 3.0 standard and leverage the TM Forum OpenAPIs. The Swagger specifications are the machine-readable files providing the definition of resources and operations for the APIs. These files could be used to display the API with documentation generation tools. They enable API implementers to generate code bindings for either client or server in various programming languages.

### Product Specifications

Product Specification is a MEF standard that defines the product-specific payload (e.g. Access E-Line) composed of MEF 3.0 services that can be used with the Sonata APIs. It includes a textual description of the attributes, allowable values, and rules between them and a machine-consumable definition (e.g. JSON schema) that implements these attributes.

APIs can support SD-WAN, Ethernet, IP and Optical Transport products through the delivery of technology specific product specifications.

### Postman scripts

Usage examples of the LSO APIs could be delivered as the Postman collections which include example payloads of the API requests and responses.

### White Papers

White Papers published by MEF are the documents aimed at different groups that provide guidance on MEF standards implementation and explain the definitions and attributes of MEF technical specifications.

## Supporting artefacts

There are additional artefacts and tools already created or planned to be developed that assist in the process of the LSO APIs implementation. They are not included in the SDK repository but the appropriate references are provided.

### Example code

The companies involved in the implementation of the LSO APIs could share their best practices and code examples with other LSO Developer Community members what could result in the acceleration of the LSO APIs adoption by the industry including their potential partners. This could include the example implementations of APIs, the code bindings in various programming languages, etc.

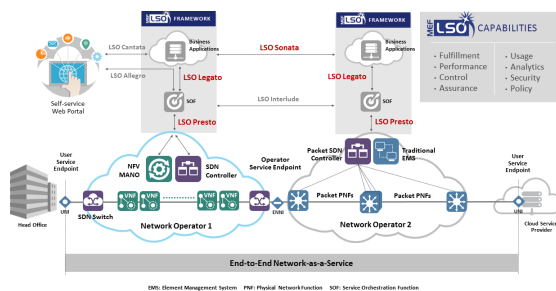
### DEV Test Service

MEF plans to develop and share the DEV Test Service which service providers can use to test their implementations of LSO APIs and speed up the integration tests.

### Certification

The MEF 3.0 LSO certification program will enable providers MEF 3.0 services to validate that their APIs used for inter-provider business transactions comply with MEF standards. Certification starts with a pilot program with an initial focus on automating ordering of MEF 3.0 CE Access E-Line services. It is expected to support APIs for serviceability (address validation, site queries, and product offering qualification), product inventory, quoting, ordering, trouble ticketing, contracts, and billing as well as additional MEF-defined services as they become available in the future.

## LSO Reference Points



Source: [LSO Reference Architecture and Capabilities](#)

# NETCONF and YANG Info

**NETCONF** is a protocol defined by the IETF in [RFC 6241](#) to “install, manipulate, and delete the configuration of network devices”. **NETCONF** operations are realized on top of a Remote Procedure Call (RPC) layer using an XML encoding and provides a basic set of operations to edit and query configuration on a network device.

**REST (Representational State Transfer)** is an architectural style that is used when building interoperable systems. The concept was first introduced in a dissertation called Architectural Styles and the Design of Network-based Software Architectures (<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>)

**YANG** is a data modelling language used to model configuration and state data manipulated by the Network Configuration Protocol (NETCONF), NETCONF remote procedure calls, and NETCONF notifications ([IETF RFC 6020](#)).

## YouTube Videos

- NETCONF by Example: <https://www.youtube.com/watch?v=N7fb11dLztA>
- YANG by Example: <https://www.youtube.com/watch?v=AhYZhbANiwE>
- PYANG: <https://www.youtube.com/watch?v=aPFTzgdYDw0>