

MEF Legato Interface Profile SpecificationFebruary 2019MEF 73 draft 0.1

Working Draft

MEF 73 draft 0.1

MEF Legato Interface Profile Specification

April 2019

This draft represents MEF work in progress and is subject to change.

Disclaimer

© MEF Forum 2019. All Rights Reserved.

The information in this publication is freely available for reproduction and use by any recipient and is believed to be accurate as of its publication date. Such information is subject to change without notice and MEF Forum (MEF) is not responsible for any errors. MEF does not assume responsibility to update or correct any information in this publication. No representation or warranty, expressed or implied, is made by MEF concerning the completeness, accuracy, or applicability of any information contained herein and no liability of any kind shall be assumed by MEF as a result of reliance upon such information.

The information contained herein is intended to be used without modification by the recipient or user of this document. MEF is not responsible or liable for any modifications to this document made by any other party.

The receipt or any use of this document or its contents does not in any way create, by implication or otherwise:

1. any express or implied license or right to or under any patent, copyright, trademark or trade secret rights held or claimed by any MEF member which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
2. any warranty or representation that any MEF members will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
3. any form of relationship between any MEF member and the recipient or user of this document.

Implementation or use of specific MEF standards, specifications, or recommendations will be voluntary, and no Member shall be obliged to implement them by virtue of participation in MEF Forum. MEF is a non-profit international organization to enable the development and worldwide adoption of agile, assured and orchestrated network services. MEF does not, expressly or otherwise, endorse or promote any specific products or services.

# Table of Contents

[1 List of Contributing Members 4](#_Toc2090068)

[2 Abstract 4](#_Toc2090069)

[3 Terminology and Abbreviations 5](#_Toc2090070)

[4 Compliance Levels 6](#_Toc2090071)

[5 Introduction 7](#_Toc2090072)

[6 Use Cases and Flow 8](#_Toc2090073)

[7 Interface Requirements 8](#_Toc2090074)

[8 Interface Information Model 8](#_Toc2090075)

[8.1 Service Catalog 8](#_Toc2090076)

[8.1.1 Service Specification 8](#_Toc2090077)

[8.1.2 Service Spec Characteristic 8](#_Toc2090078)

[8.2 Service Order 8](#_Toc2090079)

[8.2.1 Service Order 8](#_Toc2090080)

[8.2.2 Service Order Item 8](#_Toc2090081)

[8.2.3 Service Order Message 8](#_Toc2090082)

[8.2.4 Related Party 8](#_Toc2090083)

[8.3 Service Inventory 8](#_Toc2090084)

[8.3.1 Service 8](#_Toc2090085)

[8.3.2 Service Characteristic 8](#_Toc2090086)

[8.3.3 Supporting Resource 8](#_Toc2090087)

[9 State Diagrams 8](#_Toc2090088)

[10 MEF Service Common Model (MSCM) Mapping 8](#_Toc2090089)

[11 References 8](#_Toc2090090)

[Appendix A Appendix Title (Informative) 9](#_Toc2090091)

[A.1 Appendix Subsection 9](#_Toc2090092)

[A.1.1 Appendix second-level subsection 9](#_Toc2090093)

# List of Figures

**No table of figures entries found.**

# List of Tables

[Table 1 – Terminology and Abbreviations 5](#_Toc2090094)

# List of Contributing Members

The following members of the MEF participated in the development of this document and have requested to be included in this list.

1. This list will be finalized before Letter Ballot. Any member that comments in at least one CfC is eligible to be included by opting in before the Letter Ballot is initiated. Note it is the MEF member that is listed here (typically a company or organization), not their individual representatives.

* ABC Networks
* XYZ Communications

# Abstract

The MEF Legato Interface Profile Specification (IPS) describes use cases, requirements and the information model (including Class diagrams and descriptions, Interface Operations, Sequence diagrams, State diagrams, etc) for the MEF Legato Service API (MLSAPI). The MLSAPI is expected to be applied at the MEF LSO Legato Interface Reference Point (IRP).

# Terminology and Abbreviations

This section defines the terms used in this document. In many cases, the normative definitions to terms are found in other documents. In these cases, the third column is used to provide the reference that is controlling, in other MEF or external documents.

In addition, terms defined in MEF X [1] are included in this document by reference, and are not repeated in the table below.

| Term | Definition | Reference |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 1 – Terminology and Abbreviations

# Compliance Levels

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 (RFC 2119 [1], RFC 8174 [2]) when, and only when, they appear in all capitals, as shown here. All key words must be in bold text.

Items that are **REQUIRED** (contain the words **MUST** or **MUST** **NOT**) are labeled as **[Rx]** for required. Items that are **RECOMMENDED** (contain the words **SHOULD** or **SHOULD** **NOT**) are labeled as **[Dx]** for desirable. Items that are **OPTIONAL** (contain the words **MAY** or **OPTIONAL**) are labeled as **[Ox]** for optional**.**

1. The following paragraph will be deleted if no conditional requirements are used in the document.

A paragraph preceded by **[CRa]<** specifies a conditional mandatory requirement that **MUST** be followed if the condition(s) following the “<” have been met. For example, “**[CR1]<**[D38]” indicates that Conditional Mandatory Requirement 1 must be followed if Desirable Requirement 38 has been met. A paragraph preceded by **[CDb]<** specifies a Conditional Desirable Requirement that **SHOULD** be followed if the condition(s) following the “<” have been met. A paragraph preceded by **[COc]<** specifies a Conditional Optional Requirement that **MAY** be followed if the condition(s) following the “<” have been met.

# Introduction

The MEF Legato IPS describes use cases, requirements and the information model for the MEF Legato Service API (MLSAPI). The MLSAPI is expected to be applied at the MEF LSO Legato Interface Reference Point (IRP).

The MLSAPI is agnostic of any specific MEF product, service or technology specification. The different MEF services will be described as Service Specification instances (functional templates) retrievable from a Service Catalog using MLSAPI. The schema for the different types of MEF services is not specified in this document and will be done in the related MEF project – MEF Services Common Model (MSCM) The MLSAPI information model is defined to be aligned with the MSCM as well as MEF Core Model (MCM).

# Use Cases and Flow

# Interface Requirements

# Interface Information Model

<config services=’TagFileBuffer’><drop/>  
<output path=’C:\Users\217216X710581\WorkspaceMLSAPI\MEF-LSO-Legato-SDK\experimental\doc\MLSAPI\_Gendoc\_Model.docx' /><drop/>  
</config> <drop/>

<context model=’C:\Users\217216X710581\WorkspaceMLSAPI\MEF-LSO-Legato-SDK\experimental\uml\LegatoApi.notation' element=’{0}’ importedBundles='gmf;papyrus' /><drop/>

<gendoc><drop/>

## Class Diagrams

[for (d : notation::Diagram |notation::Diagram.allInstances()->sortedBy(name))]<drop/>

[if (d.name.contains(‘LSAPI’))]<drop/>

### [d.name/]

<image object='[d.getDiagram()/]' maxW='true' keepH='false' keepW=’false’ ><drop/>

</image><drop/>

[/if] <drop/>

[/for]<drop/>

</gendoc><drop/>

<context model=’C:\Users\217216X710581\WorkspaceMLSAPI\MEF-LSO-Legato-SDK\experimental\uml\LegatoApi**.**uml’ element=’{0}’ importedBundles='gmf;papyrus' />

<gendoc><drop/>

## Class Descriptions

[for (cl:Class | self.eAllContents(Class)->sortedBy(name))]<drop/>

### [cl.name/]

[for (co:Comment | cl.ownedComment)]<drop/>

<dropEmpty>[co.\_body.clean()/]</dropEmpty>

[/for]<drop/>

Applied stereotypes:

[for (st:Stereotype | cl.getAppliedStereotypes())]<drop/>

* [st.name/]

[for (oa:Property|st.ownedAttribute)]<drop/>

* [if (not oa.name.contains('base'))][oa.name/]: [if (not cl.getValue(st, oa.name).oclIsUndefined())][if oa.name.contains('condition')][cl.getValue(st, oa.name).oclAsType(String)/] [else][cl.getValue(st, oa.name).oclAsType(EnumerationLiteral).name/][/if][else]<drop/>[/if]

[/if] <drop/>

[/for]<drop/>

[/for]<drop/>

[if cl.allAttributes()->notEmpty()]<drop/>

<table><drop/>

| **Attribute Name** | **Type** | **Mult.** | **Access** | **Stereotypes** | **Description** |
| --- | --- | --- | --- | --- | --- |

[for (p:Property|cl.allAttributes())]<drop/>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [p.name/] | [p.type.name/] | [if(p.lower=p.upper)]1[else][p.lower/]..[if(p.upper=-1)]\*[else][p.upper/][/if][/if] | [if(p.isReadOnly)]R[else]RW[/if] | [for (st:Stereotype | p.getAppliedStereotypes())]<drop/>  [st.name/]  [for(oa:Property|st.ownedAttribute)]<drop/>   * [if oa.name.contains('attribute')]AVC: [p.getValue(st, oa.name).oclAsType(EnumerationLiteral).name/]   [else]<drop/>   * [if oa.name.contains('isInvariant')]isInvariant: [p.getValue(st, oa.name).oclAsType(Boolean)/]   [else]<drop/>   * [if oa.name.contains('value')]valueRange: [if (not p.getValue(st, oa.name).oclIsUndefined())][p.getValue(st, oa.name).oclAsType(String).clean()/][else] no range constraint [/if]   [else]<drop/>   * [if oa.name.contains('support')]support: [p.getValue(st, oa.name).oclAsType(EnumerationLiteral).name/]   [else]<drop/>   * [if oa.name.contains('condition')][if (not p.getValue(st, oa.name).oclIsUndefined())]condition:[p.getValue(st, oa.name).oclAsType(String).clean()/][else] <drop/> [/if]   [else]<drop/>   * [if oa.name.contains('passedByRef')] [if (not p.getValue(st, oa.name).oclIsUndefined())][p.getValue(st, oa.name).oclAsType(Boolean)/][else] undefined [/if]   [else]<drop/>   * [if oa.name.contains('reference')][if (not p.getValue(st, oa.name).oclIsUndefined())]reference:[p.getValue(st, oa.name).oclAsType(String).clean()/][else] <drop/> [/if]   [/if]<drop/>  [/if]<drop/>  [/if]<drop/>  [/if]<drop/>  [/if]<drop/>  [/if]<drop/>  [/if]<drop/>  [/for]<drop/>  [/for]<drop/> | [for (c:Comment | p.ownedComment)] <drop/>  [c.\_body.clean()/]  [/for] |

[/for]<drop/>

</table><drop/>

[else][/if]<drop/>

[/for]<drop/>

## Data Types and Enumeration

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

[if dt.oclIsTypeOf(DataType)]<drop/>

### Data Type: [dt.name/]

[for (co:Comment | dt.ownedComment)]<drop/>

<dropEmpty>[co.\_body.clean()/]</dropEmpty>

[/for]<drop/>

[if dt.ownedAttribute->notEmpty()]<drop/>

<table><drop/>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Mult.** | **Access** | **Stereotypes** | **Description** |

[for (p:Property|dt.allAttributes())]<drop/>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [p.name/] | [p.type.name/] | [if(p.lower=p.upper)]1[else][p.lower/]..[if(p.upper=-1)]\*[else][p.upper/][/if][/if] | [if(not(p.isReadOnly))]RW[else]R[/if] | [for (st:Stereotype | p.getAppliedStereotypes())]<drop/>  [st.name/]  [for(oa:Property|st.ownedAttribute)]<drop/>   * [if oa.name.contains('attribute')]AVC: [p.getValue(st, oa.name).oclAsType(EnumerationLiteral).name/]   [else]<drop/>   * [if oa.name.contains('isInvariant')]isInvariant: [p.getValue(st, oa.name).oclAsType(Boolean)/]   [else]<drop/>   * [if oa.name.contains('value')]valueRange: [if (not p.getValue(st, oa.name).oclIsUndefined())][p.getValue(st, oa.name).oclAsType(String).clean()/][else] no range constraint [/if]   [else]<drop/>   * [if oa.name.contains('support')]support: [p.getValue(st, oa.name).oclAsType(EnumerationLiteral).name/]   [else]<drop/>   * [if oa.name.contains('condition')][if (not p.getValue(st, oa.name).oclIsUndefined())]condition:[p.getValue(st, oa.name).oclAsType(String).clean()/][else] <drop/> [/if]   [else]<drop/>   * [if oa.name.contains('passedByRef')] [if (not p.getValue(st, oa.name).oclIsUndefined())][p.getValue(st, oa.name).oclAsType(Boolean)/][else] undefined [/if]   [else]<drop/>  [/if]<drop/>  [/if]<drop/>  [/if]<drop/>  [/if]<drop/>  [/if]<drop/>  [/if]<drop/>  [/for]<drop/>  [/for]<drop/> | [for (c:Comment | p.ownedComment)] <drop/>  [c.\_body.clean()/]  [/for] |

[/for]<drop/>

</table><drop/>

[else][/if]<drop/>

[else][/if]<drop/>

[/for]<drop/>

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

[if dt.oclIsTypeOf(Enumeration)]<drop/>

### Enumeration: [dt.name/]

[for (co:Comment | dt.ownedComment)]<drop/>

<dropEmpty>[co.\_body.clean()/]</dropEmpty>

[/for]<drop/>

Contains Enumeration Literals:

[for (e:EnumerationLiteral|dt.oclAsType(Enumeration).ownedLiteral)]<drop/>

* [e.name/]:
  + [for (co:Comment | e.ownedComment)]<drop/>
  + <dropEmpty>[co.\_body.clean()/]
  + </dropEmpty>[/for]<drop/>

[/for]<drop/>

[else] [/if]<drop/>

[/for]<drop/>

[for (dt:DataType | self.eAllContents(DataType)->sortedBy(name))]<drop/>

[if dt.oclIsTypeOf(PrimitiveType)]<drop/>

### Primitive: [dt.name/]

[for (co:Comment | dt.ownedComment)]<drop/>

<dropEmpty>[co.\_body.clean()/]</dropEmpty>

[/for]<drop/>

[else] [/if]<drop/>

[/for]<drop/>

## Interface Operations

[for (it:Interface | self.eAllContents(Interface)->sortedBy(name))]<drop/>

### [it.name/]

[for (co:Comment | it.ownedComment)]<drop/>

<dropEmpty>[co.\_body.clean()/]</dropEmpty>

[/for]<drop/>

Applied stereotypes:

[for (st:Stereotype | it.getAppliedStereotypes())]<drop/>

* [st.name/]

[for (oa:Property|st.ownedAttribute)]<drop/>

* [if (not oa.name.contains('base'))][oa.name/]: [if (not it.getValue(st, oa.name).oclIsUndefined())][if oa.name.contains('condition')][it.getValue(st, oa.name).oclAsType(String)/] [else][it.getValue(st, oa.name).oclAsType(EnumerationLiteral).name/][/if][else]<drop/>[/if]

[/if] <drop/>

[/for]<drop/>

[/for]<drop/>

[for (op:Operation | self.eAllContents(Operation)->sortedBy(name))]<drop/>

#### Operation: [op.name/]()

[for (co:Comment | op.ownedComment)]<drop/>

<dropEmpty>[co.\_body.clean()/]</dropEmpty>

[/for]<drop/>

Applied stereotypes:

[for (st:Stereotype | op.getAppliedStereotypes())]<drop/>

* [st.name/]

[for (oa:Property|st.ownedAttribute)]<drop/>

* [if (not oa.name.contains('base'))][oa.name/]:[if (not op.getValue(st, oa.name).oclIsUndefined())][if oa.name.contains('isOperation')][op.getValue(st, oa.name).oclAsType(Boolean)/][else][if oa.name.contains('condition')][op.getValue(st, oa.name).oclAsType(String)/][else][op.getValue(st, oa.name).oclAsType(EnumerationLiteral).name/][/if][/if]

[/if]<drop/>

[/if]<drop/>

[/for]<drop/>

[/for]<drop/>

[if op.ownedParameter->notEmpty()]<drop/>

<table><drop/>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter Name** | **Type** | **Dir.** | **Multi.** | **Stereotypes** | **Description** |

[for (p:Parameter|op.ownedParameter)]<drop/>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [p.name/] | [p.type.name/] | [p.direction/] | [if(p.lower=p.upper)]1[else][p.lower/]..[if(p.upper=-1)]\*[else][p.upper/][/if][/if] | [for (st:Stereotype | p.getAppliedStereotypes())]<drop/>  [st.name/]  [for(oa:Property|st.ownedAttribute)]<drop/>   * [if oa.name.contains('value')]valueRange: [if (not p.getValue(st, oa.name).oclIsUndefined())][p.getValue(st, oa.name).oclAsType(String)/][else] no range constraint [/if]   [else]<drop/>   * [if oa.name.contains('support')]support: [p.getValue(st, oa.name).oclAsType(EnumerationLiteral).name/]   [else]<drop/>   * [if oa.name.contains('condition')][if (not p.getValue(st, oa.name).oclIsUndefined())]condition:[p.getValue(st, oa.name).oclAsType(String)/][else] <drop/> [/if]   [else]<drop/>  [/if]<drop/>  [/if]<drop/>  [/if]<drop/>  [/for]<drop/>  [/for]<drop/> | [for (c:Comment | p.ownedComment)] <drop/>  [c.\_body.clean()/]  [/for] |

[/for]<drop/>

</table><drop/>

[else][/if]<drop/>

[/for]<drop/>

[/for]<drop/>

</gendoc><drop/>

# MEF Service Common Model (MSCM) Mapping

# References

1. IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, March 1997
2. IETF RFC 8174, *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*, May 2017

1. Appendix Title (Informative)

Appendix content.

* 1. MEF Access E-Line Service Specification

More content

* + 1. Appendix second-level subsection

Yet more content.