

Workgroup: OPSA Working Group
Internet-Draft: draft-palmero-opsawg-dmlmo-09
Published: 17 January 2023
Intended Status: Standards Track
Expires: 21 July 2023

Authors: M. Palmero F. Brockners S. Kumar
 Cisco Systems Cisco Systems NC State University
 S. Bhandari C. Cardona D. Lopez
 Thoughtspot NTT Telefonica I+D

Data Model for Lifecycle Management and Operations

Abstract

This document motivates and specifies a data model for lifecycle management and operations. It describes the motivation and requirements to collect asset-centric metrics including but not limited to asset adoption and usability, licensing, supported features and capabilities, enabled features and capabilities, etc.; with the primary objective to measure and improve the overall user experience along the lifecycle journey, from technical requirements and technology selection through advocacy and renewal, including the end of life of an asset.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 21 July 2023.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [1.1. Requirements language](#)
- [2. Terminology](#)
- [3. Motivation](#)
- [4. Use Cases](#)
 - [4.1. Entitlement Inventory and Activation](#)
 - [4.2. Features in Use](#)
 - [4.3. Assets in Use](#)
 - [4.4. Risk Mitigation Check \(RMC\)](#)
 - [4.5. Errata](#)
 - [4.6. Security Advisory](#)
 - [4.7. Optimal Software Version \(OSV\)](#)
 - [4.7.1. Software Conformance](#)
 - [4.7.2. Risk Trend Analysis](#)
 - [4.7.3. What-if Analysis](#)
 - [4.8. Asset Retirement - End of Life \(EOL\)](#)
- [5. Information Model](#)
- [6. Data Models](#)
 - [6.1. Tree Diagrams of the modules that form LMO](#)
 - [6.1.1. Aggregated Asset](#)
 - [6.1.2. Entitlements](#)
 - [6.1.3. Features](#)
 - [6.1.4. Usage](#)
 - [6.1.5. Incident Management](#)
 - [6.1.6. Organization](#)
 - [6.1.7. User](#)
 - [6.2. LMO Modules](#)
 - [6.2.1. LMO Common Module](#)
 - [6.2.2. LMO Module](#)
 - [6.2.3. Aggregated Asset](#)
 - [6.2.4. Entitlements](#)
 - [6.2.5. Features](#)
 - [6.2.6. Usage](#)
 - [6.2.7. Incident Management](#)
 - [6.2.8. Organization](#)
 - [6.2.9. User](#)
- [7. Deployment Considerations](#)
- [8. Security Considerations](#)
- [9. IANA Considerations](#)
 - [9.1. The IETF XML Registry](#)
 - [9.2. The YANG Module Names Registry](#)
- [10. References](#)
 - [10.1. Normative References](#)
 - [10.2. Informative References](#)
- [Change log](#)
- [Acknowledgments](#)
- [Appendix A](#)
- [Authors' Addresses](#)

1. Introduction

The virtualization of hardware assets and the development of applications using microservice architecture for cloud-native infrastructure created new consumption and licensing models. Any service can be deployed by composing multiple assets together where an asset refers to hardware, software, application, system or

service. For example, cloud-native infrastructure from one vendor may be hosted on the physical server from another vendor or a combination of multiple cloud-native functions from one or more vendors can be combined to execute any service.

This introduces challenges for both lifecycle and adoption management of the assets. For example, a user may need to identify the capability availability of different assets or measure the usage of each capability (or the combination) from any specific asset to measure its optimal potential. Moreover, the user could pinpoint the reason: the software application could not be optimally deployed, or is not simple to use, or is not well documented, etc. The user may use feed such measurements and analysis metrics back to the support engineers and the developers, so they can focus their work effort only on features that users are adopting, or even determine when the lifecycle of the development could end.

This creates the need to collect and analyze asset-centric lifecycle management and operations data. From now on this data will be referred as Lifecycle Management and Operations (LMO); where LMO is not limited to virtualized or cloud environments, it covers all types of networking environments in which technology assets are deployed.

LMO data constitutes data needed to measure asset-centric lifecycle metrics including but not limited to asset adoption and usability, licensing, supported features and capabilities, enabled features and capabilities, etc. The primary objective is to facilitate the asset lifecycle management from the initial asset selection and positioning, licensing, feature enablement and usage, and beyond renewal to improve the overall user experience.

The main challenge in collecting LMO-related data, especially in a multi-vendor environment, relies on the ability to produce and consume such data in a vendor-agnostic, consistent and synchronized manner. APIs or telemetry are meant to collect and relay this data to receiving equipment for storing, analysis and/or visualization.

This document describes the motivation behind LMO, lists use cases, followed by the information model and data model of LMO. The list of use cases describes the need for new functional blocks and their interactions. The current version of this draft is focused on assets, entitlement information, feature usage and incident management. This draft specifies four YANG modules [[RFC7950](#)] focused on LMO, including:

- * Entitlement,
- * Assets,
- * Usage level of Asset features, and
- * Incident Management.

This document is organized as follows. Section 2 establishes the terminology and abbreviations. In Section 3, the goals and motivation of LMO are discussed. In Section 4, use cases are introduced. Section 5 specifies the information model and the data models for LMO.

1.1. Requirements language

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 [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

2. Terminology

Terminology and abbreviations used in this document:

- *Asset: refers to hardware, software, applications, or services. An asset can be physical or virtual.
- *Consumer: refers to an entity that utilizes the outcomes of LMO. A consumer can be a user, a developer or some other interested third party.
- *Developer: refers to the entity that creates or develops the entire asset or the part of the asset.
- *EOL: End of Life.
- *Features: are options or functional capabilities available in an asset.
- *Entitlement: also known as license, is issued by an entity such as the developer or the Open Source community and allows the user to operate the asset. Entitlements determine how the asset can be leveraged and what is required in cases the asset is changed.
- *LMO: Lifecycle Management and Operations.
- *Optimal Software Version(OSV): refers to the elected software version considered optimal in the user environment.
- *PID: Product Identifier.
- *Usage: refers to how features of the asset are used.
- *User: refers to owner or consumer of the asset. User belongs to an organization. Within the organization there are entities that:
a) use the assets in their operations, b) manage the assets.
- *User Experience: how a user interacts with and experiences a particular asset. It includes a user's perceptions of ease of use, efficiency, and utility of an asset.

3. Motivation

The user experience with a specific asset can be organized into four classes:

1. Asset characteristic class, covering anything related to asset, entitlement, features, etc.

2. Utilization class, to measure how the assets and features are used, duration of usage, uptime, etc.
3. Notification class, covering any security advisory, retirement, etc.
4. Incident class, to record and report any problem the user has faced with the asset.

The ability to measure, produce and consume LMO could benefit the user organization in addressing issues such as:

*Entitlements may not have been obtained at the optimum level for a given feature, where a user might have bought entitlements that are not activated.

*Features of an asset might not be used as needed in all deployments within the organization.

*Resolution of incidents involving the asset and the developer of the technology used within the asset.

In addition to the resolution of incidents, LMO could allow developer organizations to optimize the features they offer. For example, they could consider deprecating features that are used infrequently or focus on introducing more features for the assets that are widely deployed in various infrastructures.

LMO also covers the need of communication between users and the developer. LMO can provide the capability for users to provide feedback about any asset (e.g., potential deficiency of a feature, feature enhancement request). An administrator in the user organization may include specific metrics that identify a potential problem of that specific feature or a capability of the asset. An engineer in the developer organization can determine the impact of the potential deficiency from the number of users providing feedback. Note that this channel is different from a "call to a Technical Assistance Center" in which the user may request help in resolving operational issues with the asset.

4. Use Cases

4.1. Entitlement Inventory and Activation

An operations engineer would like to understand which entitlements are activated and which are used and/or consumed. It is also important for asset users to understand which features within their assets might need an entitlement and how to activate them.

It is relatively straightforward to have an inventory of existing entitlements when there is only one asset developer (providing the asset) and one asset family.

But complexity grows when there are many different developers, systems and processes involved. New service offerings have introduced new attributes and datasets and require alignment with new business models (pay-per-product, subscription model, pay-as-you-go model, etc.). They might support different entitlement types

and models: asset activation keys, trust-based model, systems that act as proxy from the back end owned by the asset developer to support the control of entitlements, etc.

Sometimes it is a challenge to report which entitlements have been bought by the asset user, or who in the user organization owns that entitlement because that information might rely on different asset developers; even within the same asset developer, entitlements may correspond to different types or groups of assets. Asset users often need to interact with different entitlement systems and processes.

Information on how assets are entitled could be delivered from a combination of attributes such as: sales order, purchase order, asset activation key, serial number, etc.

If there is no consistency on how to deal with those data points, complexity increases for the consumer, potentially requiring manual steps. Automating those manual steps or exceptions becomes time-consuming, eventually leading to higher costs for the asset consumer.

Having a common data model for LMO eases the integration between different data sources, processes, and consolidation of the information under a common reference.

4.2. Features in Use

Feature logic is required to identify the configured features from the running configuration and determine how they might be used. There is often a lack of an easy method to list any configured features available in the current asset.

This information is extracted from the running configuration many times, implemented by a rule system without having an easy method to list any configured features available in the current asset.

Some of these use cases need to be built on top of others, and from them, other more complex use cases could be created. For instance, Software Compliance use cases can be automated, based on use cases like security advisory, errata, End of Life(EOL), etc.

All this brings a complete set of use cases that fulfills Lifecycle Management of assets, complementing and providing metrics on how asset users are using assets and how their experience from using those assets can be improved.

4.3. Assets in Use

Current approach to quantify how an asset is used, requires volume or aggregated usage/consumption metrics related to deployed assets, functions, features, integrations, etc. Also the need to quantify which metrics might be associated to a user, an organization, to specific services and how often are used; while others may be based on pre agreed profile (contractual or usage) of intended use. Examples include:

*Number of search/queries sent by the user.

*Amount of data returned to the user.

*Amount of active time spent using the asset/feature.

*Number of concurrent users accessing the asset/feature.

*Number of features in use.

*Number of users or sites using those features, etc.

The information models and data models for LMO include data fields to support metrics that might be required by consumption-based charging and licensing of asset usage.

4.4. Risk Mitigation Check (RMC)

Network, software and cloud engineers would like to be aware of known issues that are causing assets to crash so that they can act to remediate the issue quickly, or even prevent the crash if alerts are triggered on time. There are analytics tools that can process memory core dumps and crash-related files, providing the ability to the asset developers to determine the root cause.

Accordingly, asset users can remediate the problem, automate the remedy to enable incident deflection, allowing the support staff to focus on new problems. The goal of introducing normalization is not to define attributes for each of the elements being part of the crash information, but the results of RMC should be normalized and registered.

Risk Mitigation Check could also include the possibility to be aware of current and historical restarts allowing network and software engineers to enhance the service quality to asset users.

4.5. Errata

Both hardware and software critical issues or Errata need development to automate asset user matching:

*Hardware Errata match on product identifiers (PIDs) + serial numbers along with additional hardware attributes.

*Software Errata match on software type and software version along with some additional device attributes.

Engineering might develop the logic to check whether any critical issue applies to a single serial number or a specific software release.

The information to be correlated includes customer identification, entitlement, and asset information that the asset user might own. All this information needs to be correlated with hardware and software Errata, and EOL information to show which part of the asset inventory might be affected.

4.6. Security Advisory

The Security Advisory use case automates the matching of asset user data to security bulletins published by asset developers. Security Advisory logic implemented by developers could apply to a specific software release.

4.7. Optimal Software Version (OSV)

The objective of the Optimal Software Version (OSV) use case is that consumers can mark software images as OSV for their assets; based on this, it is easier for them to control and align their hardware and software assets to the set of OSVs.

Based on the logic of OSV, use cases like software compliance, risk trend analysis, acknowledge bugs, security advisories, errata, what-if analysis, etc., could be realized.

4.7.1. Software Conformance

All the assets should be at their latest recommended software version in case a security update is required to address a security issue of a specific feature.

The Software Conformance use case provides a view to the asset users and informs the users whether the assets that belong to a specific group conforms to the OSV or not. It can provide the users with a report, including a representation of software compliance for the entire network and software applications. This report could include the current software version running on the asset and the recommended software version. The report could enable users to quickly highlight which group of assets might need the most attention to inspire appropriate actions.

The Software Conformance use case uses data that might not be provided by the asset itself. Data needs to be provided and maintained also by the asset developers, through e.g., asset catalog information. Similar logic applies to a feature catalog, where the asset developer maintains the data and updates it adequately based on existing bugs, security advisories, etc.

The Software Conformance process needs to correlate the Software catalog information with the software version running on the asset.

4.7.2. Risk Trend Analysis

The Risk Trend Analysis use case provides customers with a risk trend analysis, summarizing what might change before applying changes, including registered bugs, security advisories and errata.

4.7.3. What-if Analysis

The What-if Analysis use case allows asset users to plan for new hardware or software, giving them the possibility to change the config parameters or model how new hardware or software might change the software suggestions generated by OSV.

OSV and the associated use cases involve dependencies on attributes that might need to be collected from assets directly, including related inventory information (serial numbers, asset identifiers, software versions, etc.), but also dynamic information could be required, like:

- *Information on features that might be enabled on the particular asset.

- *Catalogs, that might include information related to release notes. For example, consider a feature catalog. This catalog could include software versions that support a specific feature; the software releases that a feature is supported in; or the latest version that a feature is supported in, in case the feature is EOL.

- *Data sources to correlate information coming from reports on critical issues or errata, security advisory, End of Life, etc.

Those catalogs and data sources with errata information, EOL, etc. need to be maintained and updated by asset developers, making sure, that the software running on the assets is safe to run and up to date.

4.8. Asset Retirement - End of Life (EOL)

Hardware EOL reports need to map Hardware EOL PIDs, focusing on base PIDs so that bundles, spares, non-base PIDs, etc., do not provide false EOL reporting to asset users. Software EOL reports are used to automate the matching of user software type and software version to software EOL bulletins.

5. Information Model

The broad metric classes defined in section 3 that quantify user experience can be modeled as shown in Figure 1. There is an inventory of all assets that the user possesses. Each asset in the inventory may be entitled to one or more entitlements; a entitlement may contain one or more sub-entitlements. The level of usage for each feature and entitlement associated with the asset is measured. For every asset, a list of incidents could be created.

For example, a user needs to measure the utilization of a specific entitlement for a specific type of asset. The information about the entitlement may reside in a entitlement server. The state (activated or not) of the entitlement may reside with the asset itself or a proxy. They can be aggregated/correlated as per the information model shown in Figure 1 to give information to the user regarding the utilization of the entitlements. The user experience is thus enhanced by having accurate knowledge about the utility of the given entitlement.

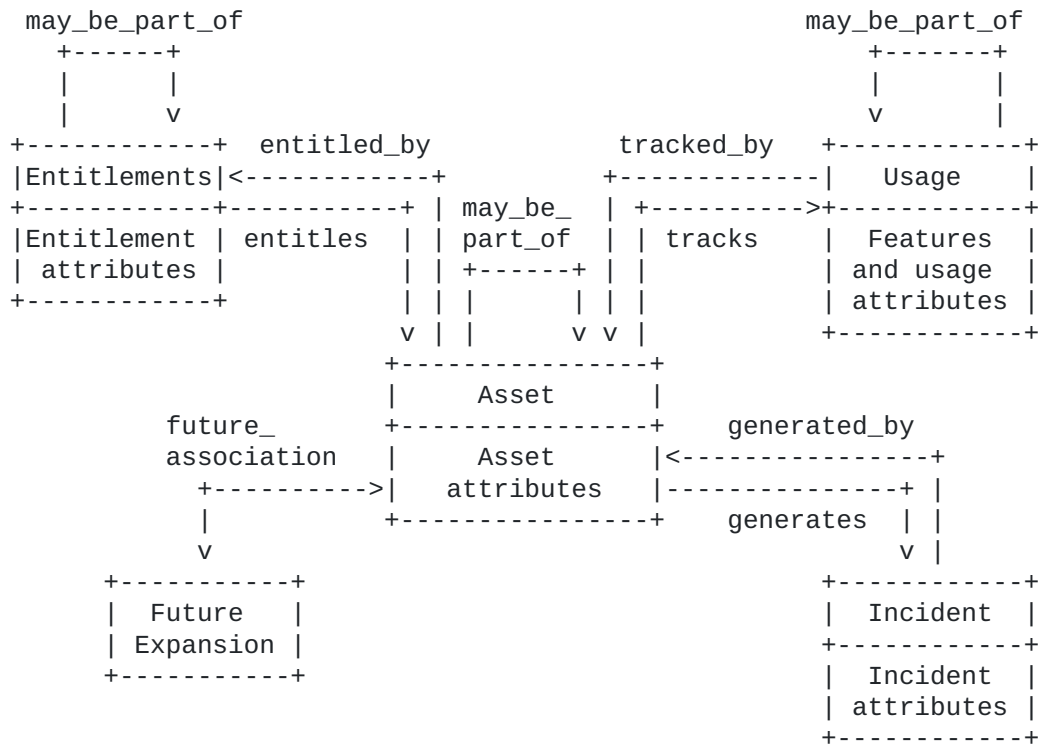


Figure 1: Information Model

The model allows for future expansion by new metrics that will quantify user experience. Notice that future association relationship and future expansion might be linked to asset or to one of the other datasets: Incident, Feature Usage or Entitlements. Two new YANG modules have been created for User and Organization. For sake of simplicity they have not been included in Figure 1; they have direct association with asset. Also, feature usage are split into two different YANG modules: Feature and Usage.

6. Data Models

6.1. Tree Diagrams of the modules that form LMO

6.1.1. Aggregated Asset

This specification uses [\[RFC9179\]](#), [\[I-D.draft-ietf-opsawg-sbom-access-10\]](#) module: ietf-lmo-assets

```

augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst:
  +--rw role?                identityref
  +--rw aggregation?         boolean
  +--rw number-of-instances? uint32
  +--rw platform-dependency-os? identityref
  +--rw install-location
  | +--rw geo-location
  | | +--rw reference-frame
  | | | +--rw alternate-system? string {alternate-systems}?
  | | | +--rw astronomical-body? string
  | | | +--rw geodetic-system
  | | |   +--rw geodetic-datum? string
  | | |   +--rw coord-accuracy? decimal64
  | | |   +--rw height-accuracy? decimal64
  | | +--rw (location)?
  | | | +--:(ellipsoid)
  | | | | +--rw latitude? decimal64
  | | | | +--rw longitude? decimal64
  | | | | +--rw height? decimal64
  | | | +--:(cartesian)
  | | |   +--rw x? decimal64
  | | |   +--rw y? decimal64
  | | |   +--rw z? decimal64
  | +--rw velocity
  | | +--rw v-north? decimal64
  | | +--rw v-east? decimal64
  | | +--rw v-up? decimal64
  | +--rw timestamp? yang:date-and-time
  | +--rw valid-until? yang:date-and-time
  +--rw deployment-mode? identityref
  +--rw activation-date? yang:date-and-time
  +--rw software-version? string
  +--ro hotfixes
  | +--ro hostfix* []
  | | +--ro version? identityref
  | | +--ro order? uint8
  +--rw software-type? string
  +--rw sign-of-life-timestamp? yang:date-and-time
  +--rw tags? string

```

6.1.2. Entitlements

module: ietf-lmo-entitlements

```
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst:
  +--rw uid?                string
  +--rw (all-1-asset)?
  |   +--:(all-assets)
  |   |   +--rw all-assets?  boolean
  |   +--:(assets)
  |       +--rw assets
  |           +--rw asset* [lmo-class id]
  |               +--rw lmo-class    -> /ietf-lmo:lmos/lmo/lmo-class
  |               +--rw id          -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-cl
+--rw resource* [id]
  |   +--rw id                string
  |   +--rw name?            string
  |   +--rw summary?        string
  |   +--rw characteristic* [id]
  |       +--rw id            string
  |       +--rw name?        string
  |       +--rw description?  string
  |       +--rw unit?        string
  |       +--rw value?       yang:counter64
  |       +--rw value-max?   yang:counter64
+--rw features
  |   +--rw feature* [lmo-class id]
  |       +--rw lmo-class    -> /ietf-lmo:lmos/lmo/lmo-class
  |       +--rw id          -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-class =
+--rw state?                ietf-lmo-common:entitlement-state-t
+--rw renewal-profile
  |   +--rw activation-date? yang:date-and-time
  |   +--rw expiration-date? yang:date-and-time
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst:
  +--rw entitlements
  |   +--rw lmo-class?      -> /ietf-lmo:lmos/lmo/lmo-class
  |   +--rw id?            -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-class = cu
```

6.1.3. Features

module: ietf-lmo-feature

```
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst:
  +--rw features
    +--rw feature* [lmo-class id]
      +--rw lmo-class -> /ietf-lmo:lmos/lmo/lmo-class
      +--rw id -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-class =
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst:
  +--rw name? string
  +--rw summary? string
  +--rw category? string
  +--rw entitlement? string
  +--rw first-available-version? string
  +--ro backported-versions
  | +--ro backported-version* []
  | +--ro version? identityref
  +--rw scope? identityref
  +--rw config-options* [id]
  | +--rw id string
  | +--rw name? string
  | +--rw summary? string
  | +--rw characteristic* [id]
  | +--rw id string
  | +--rw name? string
  | +--rw value? string
  +--rw asset
  | +--rw lmo-class? -> /ietf-lmo:lmos/lmo/lmo-class
  | +--rw id? -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-class = cur
  +--rw subfeatures
    +--rw subfeature* [lmo-class id]
      +--rw lmo-class -> /ietf-lmo:lmos/lmo/lmo-class
      +--rw id -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-class =
```

6.1.4. Usage

module: ietf-lmo-usage

```
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst:
  +--rw feature
  | +--rw lmo-class?   -> /ietf-lmo:lmos/lmo/lmo-class
  | +--rw id?         -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-class = cu
+--rw name?           string
+--rw summary?       string
+--rw uri?           string
+--rw deployment-mode? identityref
+--rw scope?         identityref
+--rw activation-status? string
+--rw instances?     uint32
+--rw count-type?    identityref
+--rw timestamp?     yang:date-and-time
+--rw count?         uint32
+--rw frequency* [name]
  | +--rw name        string
  | +--rw type-freq?  string
  | +--rw value?      yang:counter64
+--rw resource-consumption* [id]
  +--rw id            string
  +--rw name?        string
  +--rw summary?     string
  +--rw characteristic* [id]
    +--rw id          string
    +--rw name?       string
    +--rw unit?       string
    +--rw value?      yang:counter64
    +--rw value-max? yang:counter64
```

6.1.5. Incident Management

```
module: ietf-lmo-incident-management
```

```
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst:
  +--rw id?                string
  +--rw title?             string
  +--rw summary?          string
  +--rw severity?         string
  +--rw status?           string
  +--rw created?          yang:date-and-time
  +--rw last_updated?     yang:date-and-time
  +--rw capability?       string
  +--rw technology?       string
  +--rw subtechnology?    string
  +--rw problem-type?     string
  +--rw resolution?       string
  +--rw owner?            string
  +--rw support-engineer? string
  +--rw asset
  | +--rw lmo-class?      -> /ietf-lmo:lmos/lmo/lmo-class
  | +--rw id?            -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-class = cu
+--rw feature
  | +--rw lmo-class?      -> /ietf-lmo:lmos/lmo/lmo-class
  | +--rw id?            -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-class = cu
+--rw contract-number?    string
```

6.1.6. Organization

```
module: ietf-lmo-organization
```

```
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst:
  +--rw address?          string
  +--rw department?       boolean
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst:
  +--rw organization
  | +--rw lmo-class?      -> /ietf-lmo:lmos/lmo/lmo-class
  | +--rw id?            -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-class = cu
```

6.1.7. User

```
module: ietf-lmo-user
```

```
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst:
  +--rw billing-account?  uint32
  +--rw represents
  | +--rw lmo-class?      -> /ietf-lmo:lmos/lmo/lmo-class
  | +--rw id?            -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-class = cu
  +--rw authority?        enumeration
  +--rw email?            string
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst:
  +--rw user
  | +--rw lmo-class?      -> /ietf-lmo:lmos/lmo/lmo-class
  | +--rw id?            -> /ietf-lmo:lmos/lmo[ietf-lmo:lmo-class = cu
```

6.2. LMO Modules

6.2.1. LMO Common Module

<CODE BEGINS> file "ietf-lmo-common@2023-01-16.yang"

```
module ietf-lmo-common {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lmo-common";
  prefix ietf-lmo-common;
  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/opsawg/>
    WG List:  <mailto:opsawg@ietf.org>
    Editor:   Marisol Palmero
              <mailto:mpalmero@cisco.com>
    Editor:   Josh Suhr
              <mailto:josuhr@cisco.com>
    Editor:   Sudhendu Kumar
              <mailto:skumar23@ncsu.edu>";
  description
    "This YANG module defines a collection of useful data types
    and identity for Lifecycle Management and Operations (LMO).

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";

  revision 2023-01-16 {
    description
      "rename license by entitlement";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2022-02-28 {
    description
      "Introduced flexible root structure";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2021-08-23 {
    description
      "Initial revision for Common Module as part of the LMO
      YANG Model";
    reference
      "RFC XXXX: LMO YANG Model";
  }
}

typedef entitlement-id-t {
  type string;
```

```

description
    "Entitlement ID Type";
}
typedef entitlement-model-t {
type enumeration {
enum perpetual {
description
    "Perpetual entitlement";
}
enum subscription {
description
    "Subscription entitlement";
}
enum usage-based {
description
    "Usage-based entitlement";
}
enum other {
description
    "Undefined entitlement type";
}
}
description
    "Entitlement Model Type";
}
identity entitlement-buying-program-t {
description
    "Entitlement Buying Program that contains the plan to generate
    revenue for specific asset";
}
identity enterprise-agreement {
base entitlement-buying-program-t;
description
    "Enterprise Agreement";
}
identity managed-service-entitlement-agreement {
base entitlement-buying-program-t;
description
    "Managed Service Entitlement Agreement";
}
identity service-provider-network-agreement {
base entitlement-buying-program-t;
description
    "Service Provider Network Agreement";
}
identity collab-active-user {
base entitlement-buying-program-t;
description
    "Collaboration Active User";
}
identity service-full-coverage {
base entitlement-buying-program-t;
description
    "Service Full-Coverage";
}
identity offer-type-t {
description
    "License Offer Type, part of the plan to generate revenue

```

```

        for specific asset";
    }
identity perpetual-software {
    base offer-type-t;
    description
        "Perpetual softwar gives the user the right to use the
        program indefinitely";
}
identity standalone-hardware {
    base offer-type-t;
    description
        "Standalone hardware is able to function independently
        of other hardware";
}
identity on-premise-software-subscription {
    base offer-type-t;
    description
        "On-Premise software subscription, relates to a temporary
        on-prem licencing model, allowing users to pay a per user
        fee";
}
identity cloud-software-saas-subscription {
    base offer-type-t;
    description
        "Cloud Software (SaaS) subscription is a service busines
        model where the user is entitled to use the cloud software
        for a specific time period";
}
identity third-party-software {
    base offer-type-t;
    description
        "It includes licenses, entitlements, agreements, obligations
        or other commitment under which the user can use the asset
        not directly sold by the manufacturer";
}
identity flex-cloud-prem-subscription {
    base offer-type-t;
    description
        "Flex Cloud-Prem subscription allows software vendors to
        limit the number of entitlements for the use of the specific
        asset";
}
typedef purchase-order-t {
    type string;
    description
        "License purchase order number";
}
typedef entitlement-state-t {
    type enumeration {
        enum inactive {
            description
                "Inactive State";
        }
        enum active {
            description
                "Active State";
        }
    }
    enum unknown {

```

```

        description
            "Unknown State";
    }
}
description
    "Entitlement State Type";
}

typedef asset-id {
    type string;
    description
        "Asset ID Type";
}

typedef vendor-id {
    type enumeration {
        enum cisco {
            description
                "Vendor-id is Cisco";
        }
        enum other {
            description
                "Vendor-id is not determined";
        }
    }
    description
        "Vendor identifier";
}

identity asset-type {
    description
        "type of the asset: hardware, software, software cloud, ...";
}
identity hw {
    base asset-type;
    description
        "Hardware refers to any physical device";
}
identity sw {
    base asset-type;
    description
        "Software refers to a collection of code installed on a
        hardware asset";
}
identity sw-cloud {
    base asset-type;
    description
        "Cloud-based software, that allows users access to software
        application that run on a shared computing resources via
        Internet";
}
identity phone {
    base asset-type;
    description
        "Mobile telephone or a handheld two-way communication device
        over a cellular network.";
}
identity other {

```

```

base asset-type;
description
    "Different or additional type not specified as part of another
    defined asset-type.";
}
identity asset-subtype {
    description
        "subtype of the asset: router, switch, wireless,
        controller, ...";
}
identity router {
    base asset-subtype;
    description
        "Network connecting device. It operates at layer-3 of the OSI
        model.";
}
identity switch {
    base asset-subtype;
    description
        "Network connecting device. It operates at layer-2(Data Link
        Layer) of the OSI model.";
}
identity wireless {
    base asset-subtype;
    description
        "Network connecting device. It creates a wireless local area
        network. It connects to a wired router, switch, or hub via an
        Ethernet cable, and projects a Wi-Fi signal to a designated
        area";
}
identity controller {
    base asset-subtype;
    description
        "Centralized device in the network which is used in combination
        with network connection devices, when there is a need to manage
        them in large quantities.";
}
identity board {
    base asset-subtype;
    description
        "Electronic circuit board in an asset which interconnects
        another hardware assets attached to it.";
}
identity p-supply {
    base asset-subtype;
    description
        "Power supply, as it might have independent identity.";
}
identity transceiver {
    base asset-subtype;
    description
        "Device that is both a transmitter and a receiver. Usually
        it's in a single device.
        This is commonly used as a modular network interface";
}
identity others {
    base asset-subtype;
    description

```

```

        "Different or additional type not specified as part of another
        defined asset-subtype.";
    }
    identity version {
        description
            "Base identity for all version types";
    }
    identity version-sw {
    base version;
        description
            "Version release of the operating system that runs on the
            asset";
    }
    identity platform-dependency-os {
        description
            "Operating system that creates an environment for the asset
            to get deployed. Enum of options covering OS platform
            dependency.";
    }
    identity linux {
        base platform-dependency-os;
        description
            "UNIX like operating system";
    }
    identity windows {
        base platform-dependency-os;
        description
            "Windows operating system";
    }
    identity macOS {
        base platform-dependency-os;
        description
            "Mac operating system develop by Apple, Inc.";
    }
    identity darwin {
        base platform-dependency-os;
        description
            "Open-source Unix-like operating system first released by Apple
            Inc.";
    }
    identity ubuntu {
        base platform-dependency-os;
        description
            "Linux distribution, used in desktop distribution";
    }
    identity red-hat {
        base platform-dependency-os;
        description
            "Red Hat Enterprise Linux, released in multiple server and
            desktop versions";
    }
}
// NEED to extend and include iOS, Android, etc.;

identity role {
    description
        "What the role of a given device/component is in the network.
        This attribute normally will be configured on the specific
        component during setup. This attribute normally will be

```

```

        configured on the specific component during setup";
    }
identity border-router {
    base role;
    description
        "Router that provides connectivity between interior and
        exterior network routers or to the cloud";
}
identity access {
    base role;
    description
        "Router that provides access to a larger communication network
        of some sort.";
}
identity control-plane {
    base role;
    description
        "Network component that controls how data packets are
        forwarded";
}
identity edge {
    base role;
    description
        "Router that provides an entry point into enterprise or service
        provider core networks";
}
identity core {
    base role;
    description
        "Component part of the high-speed backbone of the network. It
        provides fast and efficient data transport";
}
identity datacenter {
    base role;
    description
        "Component placed in the data center, mantaining and housing
        back-end IT system and data stores";
}
identity branch {
    base role;
    description
        "Router in a remote branch of an enterprise's network";
}
}
identity deployment-mode {
    description
        "This attribute will denote the configured deployment mode
        for the asset and features, if applicable; e.g.,
        High Availability(HA) or Faiover cluster, virtual appliance,
        etc.";
}
identity primary {
    base deployment-mode;
    description
        "Asset or featurrs that support critical applications to
        minimize system downtime, to achieve high availabiilty or
        failover";
}
}

```



```

identity secondary {
  base deployment-mode;
  description
    "Redundant asset or feature, that is triggered when the
    primary encounters performance issues, to achieve high
    availability or failover";
}
identity cloud {
  base deployment-mode;
  description
    "Especially it refers to remote, distributed and shared asset
    resources (i.e. data storage, computing power, etc.), which
    are hooked together and meant to operate as a single
    ecosystem.";
}
identity virtual-appliance {
  base deployment-mode;
  description
    "pre-configured virtual machine image, ready to run on a
    hypervisor";
}
identity container {
  base deployment-mode;
  description
    "Standard unit of software that packages up code and all its
    dependencies so the application runs quickly and reliably from
    one computing environment to another";
}
identity undeployed {
  base deployment-mode;
  description
    "it refers to an asset that is undeployed";
}
identity counter-type {
  description
    "Specify the different type of counters, i.e accumulated-count,
    average-count, last-count, high-water mark count, low-water
    mark count" ;
}
identity accumulated {
  base counter-type;
  description
    "monotonically increasing counters. They're useful for
    aggregating metric information such as the number of hits
    on a web page, how many users log into a portal, etc.";
}
identity average {
  base counter-type;
  description
    "typical value in a set of metrics, in particular the mean,
    which is calculated by dividing the sum of the values in the
    set by their number.";
}
identity last {
  base counter-type;
  description

```

```

    "Last value measured and collected for specific metric.";
}
identity high-water-mark {
    base counter-type;
    description
        "Highest level of value in a set of metrics.";
}
identity low-water-mark {
    base counter-type;
    description
        "Lowest level of value in a set of metrics.";
}
identity feature-scope {
    description
        "Optional tag that could apply to any usage feature, so that
        if there are multiple dimensions of reporting that need to
        be accommodated (i.e., report feature usage by 'site')";
}
identity site {
    base feature-scope;
    description
        "Single location, part of the network";
}
identity network {
    base feature-scope;
    description
        "scope limited to the networking assets";
}
typedef feature-usage-type {
    type enumeration {
        enum none {
            description
                "No Usage";
        }
        enum low {
            description
                "Usage meeting the Low Threshold";
        }
        enum medium {
            description
                "Usage meeting the Medium Threshold";
        }
        enum high {
            description
                "Usage meeting the High Threshold";
        }
        // NEED to elaborate more on this list, based on use case
        // validation
    }
    description
        "feature usage % 0-25-50-75-100";
}

identity lmo-class {
    description "Base identity for classes of LMOs";
}
}
<CODE ENDS>

```

6.2.2. LMO Module

```

<CODE BEGINS> file "ietf-lmo@2022-12-20.yang"

module ietf-lmo {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lmo";
  prefix ietf-lmo;
  import ietf-lmo-common {
    prefix ietf-lmo-common;
  }
  import ietf-yang-types {
    prefix yang;
  }
  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
    WG List: <mailto:opsawg@ietf.org>
    Editor: Jan Lindblad
           <mailto:jlindbla@cisco.com>
    Editor: Marisol Palmero
           <mailto:mpalmero@cisco.com>";
  description
    "This YANG module add the flexibility to define its own
    and extensible set of lmo classes.

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";
  revision 2022-12-20 {
    description
      "deref reference removed";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2022-03-01 {
    description
      "Initial revision for LMO Module as part of the
      LMO YANG Model";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  container lmos {
    //config false; //temporarily commented out for easy testing
    description
      "Container that includes instances for all types of datasets
      which are part of DMLMO";
  }
}

```

```

list lmo {
  key lmo-class;
  description
    "list that includes the instances for all DMLMO datasets";
  leaf lmo-class {
    type identityref {
      base ietf-lmo-common:lmo-class;
    }
    description
      "lmo-class defining the type of instance";
  }
}
list inst {
  key id;
  description
    "defines the specific instance";
  leaf id {
    type string;
    description
      "Identifies the instance";
  }
  container parent {
    description
      "identifies the hierarchy structure for a given instance";
    leaf lmo-class {
      type leafref {
        path /lmos/lmo/lmo-class;
      }
      description
        "lmo-class that defines the parent type of instance";
    }
    leaf id {
      type leafref {
        path "/ietf-lmo:lmos/ietf-lmo:lmo[ietf-lmo:lmo-class]"
        + "=current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
      }
      description
        "identifies the parent type of instance";
    }
  }
}
container capture-info {
  // Moved capture-info to the instance level, as
  // asset/... data will generally be collected
  // from one source at one time.
  description
    "Capture information for this data";
  leaf collected-on {
    type yang:date-and-time;
    description
      "Time at which this data was collected";
  }
  leaf collected-from {
    type string;
    description
      "Identifier for original source of this data";
  }
}
}
}
}

```

```
}  
}  
<CODE ENDS>
```

6.2.3. Aggregated Asset

<CODE BEGINS> file "ietf-lmo-assets@2023-01-16.yang"

```
module ietf-lmo-assets {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lmo-assets";
  prefix ietf-lmo-asset;
  import ietf-lmo-common {
    prefix ietf-lmo-common;
  }
  import ietf-lmo {
    prefix ietf-lmo;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-geo-location {
    prefix ietf-geo-location;
  }
  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
    WG List: <mailto:opsawg@ietf.org>
    Editor: Marisol Palmero
           <mailto:mpalmero@cisco.com>
    Editor: Josh Suhr
           <mailto:josuhr@cisco.com>
    Editor: Sudhendu Kumar
           <mailto:skumar23@ncsu.edu>";
  description
    "This YANG module includes the concept asset aggregation
    and platform dependency of an asset.

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";
  revision 2023-01-16 {
    description
      "Removed inventory suffix from the YANG model";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2022-11-25 {
    description
      "Removed inventory information from this module, in order to let
      such information be mapped from other inventory models instead";
    reference
```



```

    "RFC XXXX: LMO YANG Model";
}
revision 2022-07-07 {
    description
        "fixed YANG statements";
    reference
        "RFC XXXX: LMO YANG Model";
}
revision 2022-02-22 {
    description
        "Introduced flexible root structure";
    reference
        "RFC XXXX: LMO YANG Model";
}
revision 2021-10-25 {
    description
        "Initial revision for Aggregated Assets Inventory as part of
        the LMO YANG Model";
    reference
        "RFC XXXX: LMO YANG Model";
}

identity asset {
    base ietf-lmo-common:lmo-class;
    description
        "An asset is a class of lmo that represents a
        hardware, software, application, or service component.
        An asset can also be physical or virtual";
}

augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst {
    when "derived-from-or-self(..../ietf-lmo:lmo-class,'ietf-lmo-asset:ass
    description
        "Assets container includes attributes that add the aggregated
        view";
    // Removed vendor, name, description, pid, serial-number, vid,
    // mac-address, ip-address, entity-name, product-description, udi,
    // transparency-info, from version 08;
    // as these and similar properties are expected to be managed using
    // other inventory mechanisms. Maybe more leafs should be removed.
    leaf role {
        type identityref {
            base ietf-lmo-common:role;
        }
        description
            "What the role of a given device/component is in the
            network. This attribute normally will be configured on the
            specific component during setup";
    }
    leaf aggregation {
        type boolean;
        must "../number-of-instances = 1 or current() = 'true'";
        default false;
        description
            "Asset aggregation; e.g., false (default) or true";
    }
    leaf number-of-instances {
        type uint32;
        default 1;
    }
}

```

```

description
    "Number of instances or endpoints covered by the aggregated
    asset. If different from 1, it should enforce that
    aggregation attribute is set to true";
}
leaf platform-dependency-os {
    type identityref {
        base ietf-lmo-common:platform-dependency-os;
    }
    description
        "Operating system for the asset to be deployed.";
}
container install-location {
    uses ietf-geo-location:geo-location;
    description
        "Physical installed location of the product. Location is
        provided based on what customer/user configures";
}
leaf deployment-mode {
    type identityref {
        base ietf-lmo-common:deployment-mode;
    }
    description
        "Deployment mode for the asset, if applicable; e.g.,
        HA cluster, virtual appliance, etc.";
}
leaf activation-date {
    type yang:date-and-time;
    description
        "Date of asset activation or initial contact";
}
leaf software-version {
    type string;
    description
        "Software version running on the hardware device or
        software component";
}
container hotfixes {
    config false;
    description "list of hotfixes";
    list hotfix {
        description
            "List of hotfixes that have been installed";
        leaf version {
            type identityref {
                base ietf-lmo-common:version;
            }
            description
                "It includes the first hotfix installed";
        }
        leaf order {
            type uint8;
            description
                "It refers to the order of how the hotfixes have been
                installed, range 0..100";
        }
    }
}
}

```

```
leaf software-type {
  type string;
  description
    "Software type or Operating System";
}
leaf sign-of-life-timestamp {
  type yang:date-and-time;
  description
    "Date of last contact";
}
leaf tags {
  type string;
  description
    "Comma-separated descriptive tags for this asset";
}
}
```

<CODE ENDS>

6.2.4. Entitlements

<CODE BEGINS> file "ietf-lmo-entitlements@2022-12-20.yang"

```
module ietf-lmo-entitlements {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lmo-entitlements";
  prefix ietf-lmo-entitlements;
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-lmo-common {
    prefix ietf-lmo-common;
  }
  import ietf-lmo {
    prefix ietf-lmo;
  }
  import ietf-lmo-assets {
    prefix ietf-lmo-asset;
  }
  import ietf-lmo-feature {
    prefix ietf-lmo-feature;
  }
  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
    WG List: <mailto:opsawg@ietf.org>
    Editor: Marisol Palmero
           <mailto:mpalmero@cisco.com>
    Editor: Josh Suhr
           <mailto:josuhr@cisco.com>
    Editor: Sudhendu Kumar
           <mailto:skumar23@ncsu.edu>";
  description
    "This YANG module includes the entitlement attributes of a
    product.

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";
  revision 2022-12-20 {
    description
      "license(s) renamed to entitlement(s)";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2022-09-20 {
    description
```

```

        "fixed YANG statements";
    reference
        "RFC XXXX: LMO YANG Model";
}
revision 2022-07-07 {
    description
        "fixed YANG statements";
    reference
        "RFC XXXX: LMO YANG Model";
}
revision 2022-02-28 {
    description
        "Introduced flexible root structure";
    reference
        "RFC XXXX: LMO YANG Model";
}
revision 2021-10-25 {
    description
        "Initial revision for Licenses Module as part of the LMO YANG
        Model";
    reference
        "RFC XXXX: LMO YANG Model";
}

// Can we capture licensing ties to API access where we may be
// entitled on events queries per second, minute, hour, etc.
// This is a popular model in the cloud space for example the Google
// MAPs API??

identity entitlement {
    base ietf-lmo-common:lmo-class;
    description "A entitlement is a class of lmo that represents how
    the asset(s) or feature(s) can be leveraged and what is required
    in cases the asset(s) or feature(s) are changed.";
}
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst {
    when "derived-from-or-self(..ietf-lmo:lmo-class, "+
        "'ietf-lmo-entitlements:entitlement)";
    description
        "entitlements container includes attributes for entitlements";
    leaf uid {
        type string;
        description
            "Unique License Identifier";
    }
    choice all-1-asset{
        description
            "Considering entitlement is linked to all or explicitly a
            one/few assets";
        leaf all-assets {
            type boolean;
            default false;
            description
                "License apply to all assets; e.g., false (default) or
                true";
        }
    }
    container assets {
        description

```

```

    "Assets to which this entitlement are attached";
list asset {
  key "lmo-class id";
  description
    "list of assests";
  leaf lmo-class {
    type leafref {
      path "/ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:lmo-class";
    }
    must "derived-from-or-self(current(), "+
      " 'ietf-lmo-asset:asset')";
    description
      "Asset class to which this entitlement is attached";
  }
  leaf id {
    type leafref {
      path "/ietf-lmo:lmos/ietf-lmo:lmo[ietf-lmo:lmo-class "+
      " = current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
    }
    description
      "Asset to which this entitlement is attached";
  }
}
}
}
list resource {
  key "id";
  description
    "Resource profile";
  leaf id {
    type string;
    description
      "Identify resource for entitlement consumption metric";
  }
  leaf name {
    type string;
    description
      "Friendly name of the resource";
  }
  leaf summary {
    type string;
    description
      "Brief description of the resource";
  }
}
list characteristic {
  key "id";
  description
    "Characteristic of resource consumption, i.e.,
    number of cpu's, limit BW.";
  leaf id {
    type string;
    description
      "Identifier for resource consumption characteristic";
  }
  leaf name {
    type string;
    description
      "Friendly name for resource consumption

```

```

        characteristic";
    }
    leaf description {
        type string;
        description
            "Description for resource consumption
            characteristic";
    }
    leaf unit {
        type string;
        description
            "unit of measurement for the characteristic";
    }

    // NEED to define identity type for unit: min, hour, sec,
    // days, ...

    leaf value {
        type yang:counter64;
        description
            "Resource consumption characteristic measurement";
    }
    leaf value-max {
        type yang:counter64;
        description
            "Maximum resource consumption characteristic value";
    }
}
}
}
container features {
    description
        "Features to which this entitlement are attached";
    list feature {
        key "lmo-class id";
        description
            "list of features";
        leaf lmo-class {
            type leafref {
                path "/ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:lmo-class";
            }
            must "derived-from-or-self(current(), "+
                "'ietf-lmo-feature:feature')";
            description
                "feature to which this entitlement is attached";
        }
        leaf id {
            type leafref {
                path "/ietf-lmo:lmos/ietf-lmo:lmo[ietf-lmo:lmo-class = "+
                    " current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
            }
            description
                "Feature to which this entitlement is attached";
        }
    }
}
}
leaf state {
    type ietf-lmo-common:entitlement-state-t;
    description

```



```

    "Entitlement state; e.g., active, inactive, or unknown";
}
container renewal-profile {
  description
    "Profile of entitlement renewal status and information";
  leaf activation-date {
    type yang:date-and-time;
    description
      "Activation Date";
  }
  leaf expiration-date {
    type yang:date-and-time;
    description
      "Expiration Date";
  }
}
}
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst {
  when "derived-from-or-self(..ietf-lmo:lmo-class, "+
    " 'ietf-lmo-asset:asset')";
  description
    "assets attributes related to entitlements";

  container entitlements {
    description
      "entitlement attributes";
    leaf lmo-class {
      type leafref {
        path "/ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:lmo-class";
      }
      must "derived-from-or-self(current(), "+
        " 'ietf-lmo-entitlements:entitlement')";
      description
        "Asset class to which this entitlement is attached";
    }
    leaf id {
      type leafref {
        path "/ietf-lmo:lmos/ietf-lmo:lmo[ietf-lmo:lmo-class = "+
          " current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
      }
      description
        "Asset to which this entitlement is attached";
    }
  }
}
//Fill more leafs for entitlement if required...
}
}
<CODE ENDS>

```

6.2.5. Features

```

<CODE BEGINS> file "ietf-lmo-feature@2022-12-20.yang"

module ietf-lmo-feature {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lmo-feature";
  prefix ietf-lmo-feature;
  import ietf-lmo-common {
    prefix ietf-lmo-common;
  }
  import ietf-lmo {
    prefix ietf-lmo;
  }
  import ietf-lmo-assets {
    prefix ietf-lmo-asset;
  }
  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
    WG List: <mailto:opsawg@ietf.org>
    Editor: Marisol Palmero
           <mailto:mpalmero@cisco.com>
    Editor: Josh Suhr
           <mailto:josuhr@cisco.com>
    Editor: Sudhendu Kumar
           <mailto:skumar23@ncsu.edu>";
  description
    "This YANG module includes the different attributes that define
    description, usage and resource consumption for specific
    features or capabilities of assets.

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    (https://trustee.ietf.org/license-info)."

    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";
  revision 2022-12-20 {
    description
      "fixed YANG statements";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2022-09-20 {
    description
      "fixed YANG statements";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2022-07-07 {

```

```

description
  "fixed YANG statements";
reference
  "RFC XXXX: LMO YANG Model";
}
revision 2022-02-28 {
  description
    "Introduced flexible root structure";
  reference
    "RFC XXXX: LMO YANG Model";
}
revision 2021-10-25 {
  description
    "Initial revision for Product Usage Module as part of the
    LMO YANG Model";
  reference
    "RFC XXXX: LMO YANG Model";
}

identity feature {
  base ietf-lmo-common:lmo-class;
  description
    "A feature is a class of lmo that represents
    options or functional capabilities available in an asset";
}

augment /ietf-lmo:mos/ietf-lmo:lmo/ietf-lmo:inst {
  when "derived-from-or-self(..ietf-lmo:lmo-class, 'ietf-lmo-asset:as
  description
    "feature attributes related to assets";

  container features {
    description
      "Features to be part of an asset";
    list feature {
      key "lmo-class id";
      description
        "Subfeature ID";
      leaf lmo-class {
        type leafref {
          path "/ietf-lmo:mos/ietf-lmo:lmo/ietf-lmo:lmo-class";
        }
        must "derived-from-or-self(current(), "+
        " 'ietf-lmo-feature:feature')";
        description
          "Reference to lmo-class";
      }
      leaf id {
        type leafref {
          path "/ietf-lmo:mos/ietf-lmo:lmo[ietf-lmo:lmo-class = "+
          " current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
        }
        description
          "Reference to Feature ID";
      }
    }
  }
}

//Fill more leafs for license when organization
//module is present here...

```

```

}
augment /ietf-lmo:lmo/ietf-lmo:lmo/ietf-lmo:inst {
  when "derived-from-or-self(..ietf-lmo:lmo-class, "+
    " 'ietf-lmo-feature:feature')";
  description
    "feature attributes";

  leaf name {
    type string;
    description
      "Friendly name of the feature";
  }
  leaf summary {
    type string;
    description
      "Brief description of the feature";
  }
  leaf category {
    type string;
    description
      "Feature category or tag list (as applicable to the
        product)";
  }
  leaf entitlement {
    type string;
    description
      "Minimum entitlement level, subscription, or license level
        required for the feature";
  }
  leaf first-available-version {
    type string;
    description
      "The first version in which this feature was enabled";
  }
  container backported-versions {
    config false;
    description
      "software patch or update is taken from a recent software
        version and applied to an older version of the same
        software";
    list backported-version {
      config false;
      description
        "Backport releases to older release";
      leaf version {
        type identityref {
          base ietf-lmo-common:version;
        }
        description
          "version of the backport release";
      }
    }
  }
  leaf scope {
    type identityref {
      base ietf-lmo-common:feature-scope;
    }
    description

```

```

    "Optional tag that could apply to any of the usage
    features, so that if there are multiple dimensions of
    reporting that need to be accommodated (i.e., report
    feature usage by 'site')";
}
list config-options {
  key "id";
  description
    "Feature configuration profile - optional, for features
    that require configuration beyond enable/disable";
  leaf id {
    type string;
    description
      "Identify feature configuration option uniquely across
      all products";
  }
  leaf name {
    type string;
    description
      "Friendly name of the feature option";
  }
  leaf summary {
    type string;
    description
      "Brief description of the feature option";
  }
}
list characteristic {
  key "id";
  description
    "Characteristics of feature configuration options,
    i.e. value=enabled/disabled";
  leaf id {
    type string;
    description
      "Identifier for feature option configuration
      characteristic";
  }
  leaf name {
    type string;
    description
      "Friendly name for feature option configuration
      characteristic";
  }
  leaf value {
    type string;
    description
      "Configuration characteristic value; describes how
      this feature option characteristic is configured";
  }
}
}
container asset{
  description
    "Asset that this feature is attached to";
  leaf lmo-class {
    type leafref {
      path "/ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:lmo-class";
    }
  }
}

```

```

    must "derived-from-or-self(current(), "+
        " 'ietf-lmo-asset:asset')";
    default ietf-lmo-asset:asset;
    description
        "Asset class to which this feature is attached";
}
leaf id {
    type leafref {
        path "/ietf-lmo:lmos/ietf-lmo:lmo[ietf-lmo:lmo-class = "+
            "current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
    }
    description
        "Asset to which this feature is attached";
}
}
container subfeatures {
    description
        "Sub-features to the top-level feature";
    list subfeature {
        key "lmo-class id";
        description
            "Subfeature ID";
        leaf lmo-class {
            type leafref {
                path "/ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:lmo-class";
            }
            must "derived-from-or-self(current(), "+
                " 'ietf-lmo-feature:feature')";
            description
                "lmo-class reference";
        }
        leaf id {
            type leafref {
                path "/ietf-lmo:lmos/ietf-lmo:lmo[ietf-lmo:lmo-class = "+
                    "current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
            }
            description
                "Reference to Feature ID";
        }
    }
}
}
}
}
}
<CODE ENDS>

```

6.2.6. Usage


```
<CODE BEGINS> file "ietf-lmo-usage@2022-09-20.yang"

module ietf-lmo-usage {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lmo-usage";
  prefix ietf-lmo-usage;
  import ietf-lmo-common {
    prefix ietf-lmo-common;
  }
  import ietf-lmo {
    prefix ietf-lmo;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-lmo-feature {
    prefix ietf-lmo-feature;
  }
  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
    WG List: <mailto:opsawg@ietf.org>
    Editor: Jan Lindblad
           <mailto:jlindbla@cisco.com>
    Editor: Marisol Palmero
           <mailto:mpalmero@cisco.com>";
  description
    "This YANG module includes the different attributes that define
    description, usage and resource consumption for specific
    features or capabilities of assets.

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    (https://trustee.ietf.org/license-info)."

    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";

  revision 2022-09-20 {
    description
      "fixed YANG statements";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2022-07-07 {
    description
      "fixed YANG statements";
    reference
      "RFC XXXX: LMO YANG Model";
  }
}
```

```

}
revision 2022-02-22 {
  description
    "Introduced flexible root structure";
  reference
    "RFC XXXX: LMO YANG Model";
}
revision 2021-10-25 {
  description
    "Initial revision for Product Usage Module as part of the
    LMO YANG Model";
  reference
    "RFC XXXX: LMO YANG Model";
}

identity usage {
  base ietf-lmo-common:lmo-class;
  description "A usage is a class of lmo that represents how
  features of an asset are used";
}
augment /ietf-lmo:mos/ietf-lmo:lmo/ietf-lmo:inst {
  when "derived-from-or-self(..ietf-lmo:lmo-class, "+
    " 'ietf-lmo-usage:usage')";
  description
    "usage attributes";
  container feature {
    leaf lmo-class {
      type leafref {
        path "/ietf-lmo:mos/ietf-lmo:lmo/ietf-lmo:lmo-class";
      }
      must "derived-from-or-self(current(), "+
        " 'ietf-lmo-feature:feature')";
      default ietf-lmo-feature:feature;
      description
        "feature to which this license is attached";
    }
    leaf id {
      type leafref {
        path "/ietf-lmo:mos/ietf-lmo:lmo[ietf-lmo:lmo-class = "+
          " current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
      }
      description
        "Feature to which this license is attached";
    }
  }
  description
    "usage link to a feature";
}
leaf name {
  type string;
  description
    "Name of feature usage characteristic";
}
leaf summary {
  type string;
  description
    "Brief description of feature usage characteristic";
}
leaf uri {

```

```

    type string;
    description
        "Target URI of feature characteristic, if applicable - e.g.,
        for clickstream or API";
}
leaf deployment-mode {
    type identityref {
        base ietf-lmo-common:deployment-mode;
    }
    description
        "Deployment mode for the feature. When applicable, feature
        might be independent of the deployment mode in the asset;
        e.g., cloud, HA cluster, virtual appliance, etc.";
}
leaf scope {
    type identityref {
        base ietf-lmo-common:feature-scope;
    }
    description
        "Optional tag that could apply to any of the usage
        features, so that if there are multiple dimensions of
        reporting that need to be accommodated, (i.e., report
        feature usage by 'site')";
}
leaf activation-status {
    type string;
    description
        "Feature activation status for this instance of the
        product (on/off; active/inactive; enabled/disabled)";
}
leaf instances {
    type uint32;
    description
        "Number of instances or end-points using this feature";
}

leaf count-type {
    type identityref {
        base ietf-lmo-common:counter-type;
    }
    description
        "Specify the counter type i.e accumulated-count,
        average-count, last-count, high-water mark count
        (+time stamp), low-water mark count (+time stamp)";
}
leaf timestamp {
    type yang:date-and-time;
    description
        "Some counters will benefit from timestamp based on the
        time when the counter has been collected";
}
leaf count {
    type uint32;
    units "times";
    description
        "Count of times the feature has been used";
}
list frequency {

```

```

key "name";
description
    "Frequency with which the feature is used";
leaf name {
    type string {
        length "1..64";
    }
    description
        "reference in case that feature is for different
        purpose of usage";
}
leaf type-freq {
    type string;
    description
        "Frequency type, i.e daily, weekly, monthly";
}
leaf value {
    type yang:counter64;
    description
        "Value collected for the usage";
}
}
list resource-consumption {
    key "id";
    description
        "Resource consumption profile";
    leaf id {
        type string;
        description
            "Identify resource for consumption measurement";
    }
    leaf name {
        type string;
        description
            "Friendly name of the resource";
    }
    leaf summary {
        type string;
        description
            "Brief description of the resource";
    }
}
list characteristic {
    key "id";
    description
        "Characteristic of resource consumption";
    leaf id {
        type string;
        description
            "Identifier for resource consumption characteristic";
    }
    leaf name {
        type string;
        description
            "Friendly name for resource consumption
            characteristic";
    }
}
leaf unit {

```


6.2.7. Incident Management

```

<CODE BEGINS> file "ietf-lmo-incident-management@2022-09-20.yang"

module ietf-lmo-incident-management {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lmo-incident-management";
  prefix ietf-lmo-incident;

  import ietf-lmo-common {
    prefix ietf-lmo-common;
  }
  import ietf-lmo {
    prefix ietf-lmo;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-lmo-assets {
    prefix ietf-lmo-asset;
  }
  import ietf-lmo-feature {
    prefix ietf-lmo-feature;
  }
  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
    WG List: <mailto:opsawg@ietf.org>
    Editor: Marisol Palmero
           <mailto:mpalmero@cisco.com>
    Editor: Josh Suhr
           <mailto:josuhr@cisco.com>
    Editor: Sudhendu Kumar
           <mailto:skumar23@ncsu.edu>";
  description
    "This YANG module includes the incident management attributes
    to handle incidents.

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";
  revision 2022-09-20 {
    description
      "fixed YANG statements";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2022-07-07 {

```

```

description
  "fixed YANG statements";
reference
  "RFC XXXX: LMO YANG Model";
}
revision 2022-02-28 {
  description
    "Introduced flexible root structure";
  reference
    "RFC XXXX: LMO YANG Model";
}
revision 2021-10-25 {
  description
    "Initial revision for Incident Management as part of
    the LMO YANG Model";
  reference
    "RFC XXXX: LMO YANG Model";
}

identity incident {
  base ietf-lmo-common:lmo-class;
  description
    "An incident is a class of lmo that represents an event
    that is not part of normal operations that disrupts
    operational processes.";
}

augment /ietf-lmo:mos/ietf-lmo:lmo/ietf-lmo:inst {
  when "derived-from-or-self(..ietf-lmo:lmo-class, "+
    "'ietf-lmo-incident:incident')";
  description
    "licenses container includes attributes for licenses";

  leaf id {
    type string;
    description
      "Technical Support Center Case Number id";
  }
  leaf title {
    type string;
    description
      "headline Incident Case";
  }
  leaf summary {
    type string;
    description
      "Problem description summary";
  }
  leaf severity {
    type string;
    description
      "severity, in some cases also defined as priority";
    // NEED to define different levels of severity, from
    // severity 1 to 6, i.e. Sev1: network down, Sev6: enhancement
  }
  leaf status {
    type string;
    description
      "case status, i.e. Customer-Pending, Customer

```



```

    Engineer-Pending, Developer-Pending, Closed, Open, etc.";
    // NEED to define different status
}
leaf created {
    type yang:date-and-time;
    description
    "created date-and-time";
}
leaf last_updated {
    type yang:date-and-time;
    description
    "last updated date-and-time";
}
leaf capability {
    type string;
    description
    "i.e. to reveal associated RMA items";
}
leaf technology{
    type string;
    description
    "Technology related";
}
leaf subtechnology{
    type string;
    description
    "Subtechnology related";
}
leaf problem-type{
    type string;
    description
    "Problem type definition, i.e. network, faulty hardware,
    performance, security, etc. ";
}
leaf resolution{
    type string;
    description
    "code (closed cases only)";
}
leaf owner{
    type string;
    description
    "Customer in charge of the case";
    // NEED to refer to user lmo-class
}
leaf support-engineer{
    type string;
    description
    "Customer Support Engineer in charge of the case";
    // NEED to refer to user lmo-class
}
container asset {
    description
    "Asset container to which this incident is attached";
    leaf lmo-class {
        type leafref {
            path "/ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:lmo-class";
        }
    }
}

```

```

    must "derived-from-or-self(current(), "+
        " 'ietf-lmo-asset:asset')";
    default ietf-lmo-asset:asset;
    description
        "Asset class to which this incident is attached";
}
leaf id {
    type leafref {
        path "/ietf-lmo:lmos/ietf-lmo:lmo[ietf-lmo:lmo-class = "+
            " current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
    }
    description
        "Asset to which this incident is attached";
}
}
container feature {
    description
        "Feature to which this incident is attached";
    leaf lmo-class {
        type leafref {
            path "/ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:lmo-class";
        }
        must "derived-from-or-self(current(), "+
            " 'ietf-lmo-feature:feature')";
        default ietf-lmo-feature:feature;
        description
            "Feature class to which this incident is attached";
    }
    leaf id {
        type leafref {
            path "/ietf-lmo:lmos/ietf-lmo:lmo[ietf-lmo:lmo-class = "+
                " current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
        }
        description
            "Feature to which this incident is attached";
    }
}
}
leaf contract-number {
    type string;
    description
        "Support contract number";
    // NEED to evaluate if it should be independent container. It
    // should be associated to asset and license
}
}
}
}
<CODE ENDS>

```

6.2.8. Organization

<CODE BEGINS> file "ietf-lmo-organization@2022-12-20.yang"

```
module ietf-lmo-organization {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lmo-organization";
  prefix ietf-lmo-organization;
  import ietf-lmo-common {
    prefix ietf-lmo-common;
  }
  import ietf-lmo {
    prefix ietf-lmo;
  }
  import ietf-lmo-entitlements {
    prefix ietf-lmo-entitlements;
  }
  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
    WG List: <mailto:opsawg@ietf.org>
    Editor: Jan Lindblad
           <mailto:jlindbla@cisco.com>
    Editor: Marisol Palmero
           <mailto:mpalmero@cisco.com>";
  description
    "This YANG module defines the organization module and the
    different attributes that defines it.

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";
  revision 2022-12-20 {
    description
      "license(s) renamed to entitlement(s)";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2022-09-20 {
    description
      "fixed YANG statements";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2022-07-07 {
    description
      "fixed YANG statements";
    reference
```

```

    "RFC XXXX: LMO YANG Model";
}
revision 2022-03-01 {
    description
        "Initial revision for Organization Module as part of the
        LMO YANG Model";
    reference
        "RFC XXXX: LMO YANG Model";
}

identity organization {
    base ietf-lmo-common:lmo-class;
    description
        "An organization is a class of lmo that represents an entity
        such as a company, an institution, or an association,
        comprising one or more people and having a particular
        purpose";
}
augment /ietf-lmo:mos/ietf-lmo:lmo/ietf-lmo:inst {
    when "derived-from-or-self(..../ietf-lmo:lmo-class, "+
        " 'ietf-lmo-organization:organization')";
    description
        "organization attributes";

    leaf address {
        type string;
        description
            "organization address";
    }
    leaf department {
        type boolean;
        default false;
        description
            "How to categorize the organization";
    }
    //Fill more leafs for organizations here...
}
augment /ietf-lmo:mos/ietf-lmo:lmo/ietf-lmo:inst {
    when "derived-from-or-self(..../ietf-lmo:lmo-class, "+
        " 'ietf-lmo-entitlements:entitlement')";
    description
        "entitlement attributes related to organization";

    container organization {
        //NEED to determine how the model will consume
        //organization/service/user
        //Leaving it here for the moment
        description
            "Organization container";
        leaf lmo-class {
            type leafref {
                path "/ietf-lmo:mos/ietf-lmo:lmo/ietf-lmo:lmo-class";
            }
            must "derived-from-or-self(current(), "+
                " 'ietf-lmo-organization:organization')";
            description
                "Organization class to which this entitlement is attached";
        }
    }
}

```

```
leaf id {
  type leafref {
    path "/ietf-lmo:lmos/ietf-lmo:lmo[ietf-lmo:lmo-class = "+
      " current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
  }
  description
    "Organization to which this entitlement is attached";
}
}
//License module can be extended here when organization is
//present...
}
}
<CODE ENDS>
```

6.2.9. User

```

<CODE BEGINS> file "ietf-lmo-user@2022-12-20.yang"

module ietf-lmo-user {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lmo-user";
  prefix ietf-lmo-user;
  import ietf-lmo-common {
    prefix ietf-lmo-common;
  }
  import ietf-lmo {
    prefix ietf-lmo;
  }
  import ietf-lmo-entitlements {
    prefix ietf-lmo-entitlements;
  }
  import ietf-lmo-organization {
    prefix ietf-lmo-organization;
  }
  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
    WG List: <mailto:opsawg@ietf.org>
    Editor: Jan Lindblad
           <mailto:jlindbla@cisco.com>
    Editor: Marisol Palmero
           <mailto:mpalmero@cisco.com>";
  description
    "This YANG module includes the user module and the different
    attributes that defines it.

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";

  revision 2022-12-20 {
    description
      "license(s) renamed to entitlement(s)";
    reference
      "RFC XXXX: LMO YANG Model";
  }
  revision 2022-10-11 {
    description
      "fixed definition of users";
    reference
      "RFC XXXX: LMO YANG Model";
  }
}

```



```

revision 2022-09-20 {
  description
    "fixed YANG statements";
  reference
    "RFC XXXX: LMO YANG Model";
}
revision 2022-07-07 {
  description
    "fixed YANG statements";
  reference
    "RFC XXXX: LMO YANG Model";
}

revision 2022-03-01 {
  description
    "Initial revision for User Module as part of the
    LMO YANG Model";
  reference
    "RFC XXXX: LMO YANG Model";
}

identity user {
  base ietf-lmo-common:lmo-class;
  description
    "A user is a class of lmo that refers to owner or consumer
    of the asset. User belongs to an organization. Within the
    organization there are entities that: a) use the assets
    in their operations, b) manage the assets. ";
}

augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst {
  when "derived-from-or-self(..ietf-lmo:lmo-class, "+
    " 'ietf-lmo-user:user')";
  description
    "user attributes";

  leaf billing-account {
    type uint32;
    description "billing account information";
  }
  container represents {
    description
      "Organization to which this user is attached";
    leaf lmo-class {
      type leafref {
        path "/ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:lmo-class";
      }
      must "derived-from-or-self(current(), "+
        " 'ietf-lmo-organization:organization')";
      description
        "Organization class to which this user is attached";
    }
  }
  leaf id {
    type leafref {
      path "/ietf-lmo:lmos/ietf-lmo:lmo[ietf-lmo:lmo-class = "+
        " current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
    }
    description
      "Organization to which this user is attached";
  }
}

```

```

    }
  }
  leaf authority {
    type enumeration {
      enum super-user {
        description
          "super-user";
      }
      enum standard-user {
        description
          "standard-user";
      }
      enum restricted-user {
        description
          "restricted-user";
      }
      enum admin {
        description
          "admin";
      }
    }
    default standard-user;
    description
      "authority assign to the user";
  }
  leaf email {
    type string;
    description
      "user email";
  }
  // Fill in more leafs you want on organizations here...
}

augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst {
  when "derived-from-or-self(..ietf-lmo:lmo-class, "+
    " 'ietf-lmo-entitlements:entitlement')";
  description
    "entitlement attributes related to user";

  container user {
    //NEED to determine how the model will consume
    //organization/service/user. Leaving it here for the moment
    description
      "user container";
    leaf lmo-class {
      type leafref {
        path "/ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:lmo-class";
      }
      must "derived-from-or-self(current(), "+
        " 'ietf-lmo-user:user')";
      description
        "User class to which this entitlement is attached";
    }
  }
  leaf id {
    type leafref {
      path "/ietf-lmo:lmos/ietf-lmo:lmo[ietf-lmo:lmo-class = "+
        " current()/../lmo-class]/ietf-lmo:inst/ietf-lmo:id";
    }
  }
}

```

```
description
  "User to which this entitlement is attached";
}
}
}
}
<CODE ENDS>
```

7. Deployment Considerations

LMO Data Models defines the data schemas for LMO data. LMO Data Models are based on YANG. YANG data models can be used independent of the transport and can be converted into any encoding format supported by the network configuration protocol. YANG is a protocol independent.

To enable the exchange of LMO data among all interested parties, deployment considerations that are out of the scope of this document, will need to include:

- *The data structure to describe all metrics and quantify relevant data consistently, i.e. specific formats like XML or JSON encoded message would be deemed valid or invalid based on LMO models.

- *The process to share and collect LMO data across the consumers consistently, including the transport mechanism. The LMO YANG models can be used with network management protocols such as NETCONF [[RFC6241](#)], RESTCONF [[RFC8040](#)], streaming telemetry, etc. OpenAPI specification might also help to consume LMO metrics.

- *How the configuration of assets should be done.

8. Security Considerations

The security considerations mentioned in section 17 of [[RFC7950](#)] apply.

LMO brings several security and privacy implications because of the various components and attributes of the information model. For example, each functional component can be tampered with to give manipulated data. LMO when used alone or with other relevant data, can identify an individual, revealing Personal Identifiable Information (PII). Misconfigurations can lead to data being accessed by unauthorized entities.

Methods exist to secure the communication of management information. The transport entity of the functional model MUST implement methods for secure transport. This document also contains an Information model and Data-Model in which none of the objects defined are writable. If the objects are deemed sensitive in a particular environment, access to them MUST be restricted using appropriately configured security and access control rights. The information model contains several optional elements which can be enabled or disabled for the sake of privacy and security. Proper authentication and audit trail MUST be included for all the users/processes that access the LMO.

9. IANA Considerations

9.1. The IETF XML Registry

This document registers URIs in the IETF XML registry [[RFC3688](#)]. Following the format in [[RFC3688](#)], the registrations defined below are requested:

URI: urn:ietf:params:xml:ns:yang:ietf-lmo
Registrant Contact: The OPSA WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-lmo-common
Registrant Contact: The OPSA WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-lmo-assets
Registrant Contact: The OPSA WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-lmo-entitlements
Registrant Contact: The OPSA WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-lmo-feature
Registrant Contact: The OPSA WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-lmo-usage
Registrant Contact: The OPSA WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-lmo-incident-management
Registrant Contact: The OPSA WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-lmo-organization
Registrant Contact: The OPSA WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-lmo-user
Registrant Contact: The OPSA WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

9.2. The YANG Module Names Registry

This document registers YANG modules in the YANG Module Names registry [[RFC7950](#)]. Following the format in [[RFC7950](#)], the registrations defined below are requested:

name: ietf-lmo
namespace: urn:ietf:params:xml:ns:yang:ietf-lmo
maintained by IANA: N
prefix: ietf-lmo
reference: RFC XXXX

name: ietf-lmo-common
namespace: urn:ietf:params:xml:ns:yang:ietf-lmo-common
maintained by IANA: N
prefix: ietf-lmo-common
reference: RFC XXXX

name: ietf-lmo-asset-inventory
namespace: urn:ietf:params:xml:ns:yang:ietf-lmo-assets
maintained by IANA: N
prefix: ietf-lmo-asset
reference: RFC XXXX

name: ietf-lmo-entitlements
namespace: urn:ietf:params:xml:ns:yang:ietf-lmo-entitlements
maintained by IANA: N
prefix: ietf-lmo-entitlements
reference: RFC XXXX

name: ietf-lmo-feature
namespace: urn:ietf:params:xml:ns:yang:ietf-lmo-feature
maintained by IANA: N
prefix: ietf-lmo-feature
reference: RFC XXXX

name: ietf-lmo-usage
namespace: urn:ietf:params:xml:ns:yang:ietf-lmo-usage
maintained by IANA: N
prefix: ietf-lmo-usage
reference: RFC XXXX

name: ietf-lmo-incident-management
namespace: urn:ietf:params:xml:ns:yang:ietf-lmo-incident-management
maintained by IANA: N
prefix: ietf-lmo-incident
reference: RFC XXXX

name: ietf-lmo-organization
namespace: urn:ietf:params:xml:ns:yang:ietf-lmo-organization
maintained by IANA: N
prefix: ietf-lmo-organization
reference: RFC XXXX

name: ietf-lmo-user
namespace: urn:ietf:params:xml:ns:yang:ietf-lmo-user
maintained by IANA: N
prefix: ietf-lmo-user
reference: RFC XXXX

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

10.2. Informative References

- [I-D.draft-ietf-opsawg-sbom-access-10] Lear, E. and S. Rose, "Discovering and Retrieving Software Transparency and Vulnerability Information", Work in Progress, Internet-Draft, draft-ietf-opsawg-sbom-access-10, 28 September 2022, <<https://www.ietf.org/archive/id/draft-ietf-opsawg-sbom-access-10.txt>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC9179] Hopps, C., "A YANG Grouping for Geographic Locations", RFC 9179, DOI 10.17487/RFC9179, February 2022, <<https://www.rfc-editor.org/info/rfc9179>>.

Change log

RFC Editor Note: This section is to be removed during the final publication of the document.

version 09

*Rename "license" to "entitlement".

*renamed ietf-lmo-assets-inventory to ietf-lmo-assets.

*ietf-lmo-assets provides capability of integration and extension for a different approach on how to address inventory use cases. Process is explained in the Appendix A.

*ietf-lmo-example-mapping-XXX YANG modules accommodates the ietf-lmo-assets YANG module to any other inventory which will be required in the future to be referenced.

version 08

*fixing errors shown in YANG validation

version 07

*fixing references

version 06

*fixing errors shown in YANG validation

version 05

*introduce fixes for YANG statements

version 04

*Remove ietf-lmo-service YANG module, as service is considered within the asset concept

*Fix introduced to the .xml and .txt avoiding a compiling issue on the YANG modules.

version 03

*Flexible root structure has been introduced by the ietf-lmo YANG module: Modules are arranged into layers, with ietf-lmo-common and ietf-lmo at the core. Other modules can be added in layers on top. This structure allows flexibility and the option to be enhanced by vendor implementation. The new structure allows to include other lmo classes, or exclude current lmo classes.

*Feature and Usage containers have been split in two independent modules. Where Usage relates to runtime data.

*Organization attribute, has been enhanced to an independent YANG module, adding flexibility and the option to be called independently and enhanced.

*Service and User YANG modules, have been also introduced in a similar flexible structure, being part of new lmo classes.

*Information Model, has been enhanced with new modules: Organization, Service and User modules. On this version the new lmo classes can be called independently or from the entitlements module. There is no restriction to be called from any of the other YANG modules.

version 02

- *"Support case" renamed to "incident".

- *Add MAC address and IP address attributes under asset-inventory YANG module.

- *Link among objects & YANG modules (notably with feature).

- *New text about asset usage.

version 01

- *Fixes for YANG validator and idnits warnings.

version 00

- *Initial version.

Acknowledgments

The ideas in this document originate from early work by Tony Colon, Carlos Pignataro, and Yenu Gobena originally referred to as Experience Telemetry.

This document was created by meaningful contributions from Josh Suhr, Eric Vyncke, Yannis Viniotis, Nagendra Kumar Nainar, Yenu Gobena, Dhiren Tailor and Jan Lindblad.

The authors wish to thank Gonzalo Salgueiro, Martin Beverley, Ignacio Dominguez Martinez and many others for their helpful comments and suggestions.

Appendix A

Hardware network inventory is described as part of network topology which is defined in [[RFC8345](#)], it has been explored in several IETF work as it might need an extension for some of the use cases that need to consume inventory information. This is the case for DMLMO, as assets are defined as hardware, software or even service instances.

This section summarizes and provides an example with the changes to make DMLMO compatible to any future changes that will come as part of the current inventory discussions and decisions.

DMLMO version -09 provides the approach to make DMLMO independent from the network inventory discussions, providing a way to consume any inventory management module(s). Version -09 contains changes to accommodate ietf-lmo-assets, aka ietf-lmo-assets-inventory in previous versions, to any other inventory module that might be required.

The following example considers iana-hardware and ietf-network-inventory YANG modules as inventory YANG modules to consider. It could include others, i.e., openconfig-platform.

```

<CODE BEGINS> file "iana-hardware@.yang"

module iana-hardware {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:iana-hardware";
  prefix ianahw;

  organization "IANA";
  contact
    "      Internet Assigned Numbers Authority

    Postal: ICANN
           12025 Waterfront Drive, Suite 300
           Los Angeles, CA 90094-2536
           United States of America

    Tel:    +1 310 301 5800
    E-Mail: iana@iana.org>";

  description
    "IANA-defined identities for hardware class.

    The latest revision of this YANG module can be obtained from
    the IANA website.

    Requests for new values should be made to IANA via
    email (iana@iana.org).

    Copyright (c) 2018 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License

    set forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    The initial version of this YANG module is part of RFC 8348;
    see the RFC itself for full legal notices.";
  reference
    "https://www.iana.org/assignments/yang-parameters";

  revision 2018-03-13 {
    description
      "Initial revision.";
    reference
      "RFC 8348: A YANG Data Model for Hardware Management";
  }

  /*
  * Identities
  */

  identity hardware-class {
    description
      "This identity is the base for all hardware class

```

```
        identifiers.";
    }

identity unknown {
    base ianahw:hardware-class;
    description
        "This identity is applicable if the hardware class is unknown
        to the server.";
}

identity chassis {
    base ianahw:hardware-class;
    description
        "This identity is applicable if the hardware class is an
        overall container for networking equipment. Any class of
        physical component, except a stack, may be contained within a
        chassis; a chassis may only be contained within a stack.";
}

identity backplane {
    base ianahw:hardware-class;
    description
        "This identity is applicable if the hardware class is some sort
        of device for aggregating and forwarding networking traffic,
        such as a shared backplane in a modular ethernet switch. Note
        that an implementation may model a backplane as a single
        physical component, which is actually implemented as multiple
        discrete physical components (within a chassis or stack).";
}

identity container {
    base ianahw:hardware-class;
    description
        "This identity is applicable if the hardware class is capable
        of containing one or more removable physical entities,
        possibly of different types. For example, each (empty or
        full) slot in a chassis will be modeled as a container. Note
        that all removable physical components should be modeled
        within a container component, such as field-replaceable
        modules, fans, or power supplies. Note that all known
        containers should be modeled by the agent, including empty
        containers.";
}

identity power-supply {
    base ianahw:hardware-class;
    description
        "This identity is applicable if the hardware class is a
        power-supplying component.";
}

identity fan {
    base ianahw:hardware-class;
    description
        "This identity is applicable if the hardware class is a fan or
        other heat-reduction component.";
}
```

```

identity sensor {
  base ianahw:hardware-class;
  description
    "This identity is applicable if the hardware class is some sort
    of sensor, such as a temperature sensor within a router
    chassis.";
}

identity module {
  base ianahw:hardware-class;
  description
    "This identity is applicable if the hardware class is some sort
    of self-contained sub-system. If a module component is
    removable, then it should be modeled within a container

    component; otherwise, it should be modeled directly within
    another physical component (e.g., a chassis or another
    module).";
}

identity port {
  base ianahw:hardware-class;
  description
    "This identity is applicable if the hardware class is some sort
    of networking port capable of receiving and/or transmitting
    networking traffic.";
}

identity stack {
  base ianahw:hardware-class;
  description
    "This identity is applicable if the hardware class is some sort
    of super-container (possibly virtual) intended to group
    together multiple chassis entities. A stack may be realized
    by a virtual cable, a real interconnect cable attached to
    multiple chassis, or multiple interconnect cables. A stack
    should not be modeled within any other physical components,
    but a stack may be contained within another stack. Only
    chassis components should be contained within a stack.";
}

identity cpu {
  base ianahw:hardware-class;
  description
    "This identity is applicable if the hardware class is some sort
    of central processing unit.";
}

identity energy-object {
  base ianahw:hardware-class;
  description
    "This identity is applicable if the hardware class is some sort
    of energy object, i.e., it is a piece of equipment that is
    part of or attached to a communications network that is
    monitored, it is controlled, or it aids in the management of
    another device for Energy Management.";
}

```

```
identity battery {
  base ianahw:hardware-class;
  description
    "This identity is applicable if the hardware class is some sort
    of battery.";
}

identity storage-drive {
  base ianahw:hardware-class;
  description
    "This identity is applicable if the hardware class is some sort
    of component with data storage capability as its main
    functionality, e.g., hard disk drive (HDD), solid-state device
    (SSD), solid-state hybrid drive (SSHD), object storage device
    (OSD), or other.";
}
}

<CODE ENDS>
```

The YANG modules `ietf-lmo-example-mapping-ietf-network-inventory` and `ietf-lmo-example-mapping-openconfig-platform` make the import of the inventory module(s) and augment the `ietf-lmo-assets` YANG module to include inventory attributes to the asset identity.

For this practice, `ietf-lmo-assets.yang`, removes `vendor`, `name`, `description`, `pid`, `serial-number`, `vid`, `mac-address`, `ip-address`, `entity-name`, `product-description`, `udi`, `transparency-info` as these and similar properties are expected to be managed using other inventory mechanism.

This process requires to include a mapping YANG module per imported inventory YANG module.

Module `ietf-lmo-example-mapping-ietf-network-inventory`, makes the mapping between `ietf-network-inventory` and `ietf-lmo-assets`, augmenting asset identity:

```

<CODE BEGINS> file "ietf-lmo-example-mapping-ietf-network-inventory@.yan
module ietf-lmo-example-mapping-ietf-network-inventory {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lmo-example-mapping-ietf-n
  prefix ietf-lmo-example-map-ietf;

  import ietf-lmo-common {
    prefix ietf-lmo-common;
  }
  import ietf-lmo {
    prefix ietf-lmo;
  }
  import ietf-lmo-assets {
    prefix ietf-lmo-asset;
  }
  import ietf-network-inventory {
    prefix ni;
  }
  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/opsawg/>
    WG List: <mailto:opsawg@ietf.org>
    Editor: Marisol Palmero
           <mailto:mpalmero@cisco.com>
    Editor: Josh Suhr
           <mailto:josuhr@cisco.com>
    Editor: Sudhendu Kumar
           <mailto:skumar23@ncsu.edu>";
  description
    "This YANG module maps the IETF LMO asset concept to the
    IETF network inventory framework.

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";

  revision 2022-11-25 {
    description
      "First version of mapping to IETF asset inventory modules.";
    reference
      "RFC XXXX: mapping inventory and LMO YANG Model";
  }

  augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst {
    when "derived-from-or-self(..ietf-lmo:lmo-class,'ietf-lmo-asset:ass

```

```

choice mapping-type {
  // config true;
  description
    "mapping type description";
  case network-element {
    leaf network-element-ref {
      type leafref {
        path "/ni:network-inventory/ni:network-elements/"
          + "ni:network-element/ni:uuid";
        description
          "network element reference description";
      }
    }
  }
  case component {
    leaf component-network-element-ref {
      type leafref {
        path "/ni:network-inventory/ni:network-elements/"
          + "ni:network-element/ni:uuid";
        description
          "component network element reference description";
      }
    }
    leaf component-ref {
      type leafref {
        path "/ni:network-inventory/ni:network-elements/"
          + "ni:network-element"
          + "[ni:uuid = current()/../network-element-ref]/"
          + "ni:components/ni:component/ni:uuid";
        description
          "component reference description";
      }
    }
  }
  case rack {
    leaf rack-equipment-room-ref {
      type leafref {
        path "/ni:network-inventory/ni:equipment-rooms/"
          + "ni:equipment-room/ni:uuid";
        description
          "rack equipment room reference description";
      }
    }
    leaf rack-ref {
      type leafref {
        path "/ni:network-inventory/ni:equipment-rooms/"
          + "ni:equipment-room"
          + "[ni:uuid = current()/../rack-equipment-room-ref]/"
          + "ni:racks/ni:rack/ni:uuid";
        description
          "rack reference description";
      }
    }
  }
}

description
  "This adds a reference from LMO instances of class 'asset'

```



```
    to the IETF network inventory tree.";  
  }  
}  
<CODE ENDS>
```

Module `ietf-lmo-example-mapping-openconfig-platform`, includes the mapping between `openconfig-platform` and `ietf-lmo-assets`, augmenting asset identity:

```

<CODE BEGINS> file "ietf-lmo-example-mapping-openconfig-platform@2023-01
module ietf-lmo-example-mapping-openconfig-platform {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lmo-example-mapping-openco
  prefix ietf-lmo-example-map-oc;

  import ietf-lmo {
    prefix ietf-lmo;
  }
  import ietf-lmo-assets {
    prefix ietf-lmo-asset;
  }
  // For the show case, import statement should be uncommented,
  // import openconfig-platform {
  //   prefix oc-platform;
  // }
  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/opsawg/>
    WG List:  <mailto:opsawg@ietf.org>
    Editor:   Marisol Palmero
              <mailto:mpalmero@cisco.com>
    Editor:   Josh Suhr
              <mailto:josuhr@cisco.com>
    Editor:   Sudhendu Kumar
              <mailto:skumar23@ncsu.edu>";
  description
    "This YANG module maps the IETF LMO asset concept to the
    OpenConfig platform framework.

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.";
  revision 2023-01-16 {
    description
      "import openconfig-platform statement commented out.";
    reference
      "RFC XXXX: mapping openconfig inventory and LMO YANG Model";
  }
  revision 2022-11-25 {
    description
      "First version of mapping to OC platform modules.";
    reference
      "RFC XXXX: mapping openconfig inventory and LMO YANG Model";
  }
}

```

```
augment /ietf-lmo:lmos/ietf-lmo:lmo/ietf-lmo:inst {
  when "derived-from-or-self(..ietf-lmo:lmo-class,'ietf-lmo-asset:ass

  leaf oc-component-ref {
    type string;
    config true;
    // leafref path "/oc-platform:components/oc-platform:component/"
    // + "oc-platform:name";
    // this is simplified version to show case
    description "simplified example to include component reference";
  }

  description
    "This adds a reference from LMO instances of class 'asset'
    to the OpenConfig platform tree.";
}
}
<CODE ENDS>
```

openconfig-platform.yang is not included in the section for simplicity of the output.

Once compilation is applied to the YANG modules, the following configuration, considers network element "router2" as a hardware network element, which is described under network-inventory YANG module:

```
network-inventory network-elements network-element 2222 name
router2 hardware-rev 1.1 software-rev 17.1 mfg-name cisco serial-
number AF123456 product-name ASR1k components component fan part-
number 678678 components component psu part-number 654321
```

"router2" asset identity is augmented including attributes from ietf-network-inventory(i.e. rack-equipment-room-ref, rack-ref, network-element-ref, etc) and any other imported YANG module, i.e. openconfig-platform inventory YANG modules, with oc-component-ref.

```
lmo@(config)#lmos lmo asset inst router2 ? Possible completions:
activation-date age aggregation capture-info component-network-
element-ref
component-ref ietf-lmo-asset:deployment-mode ietf-lmo-
feature:features install-location interfaces licenses network-
element-ref number-of-instances oc-component-ref parent
platform-dependency-os rack-equipment-room-ref rack-ref role sign-
of-life-timestamp
software-type software-version tags
```

Changes in future versions of DMLMO, might require one unique import statement in the mapping YANG module, from another inventory YANG module.

Authors' Addresses

Marisol Palmero
Cisco Systems

Email: mpalmero@cisco.com

Frank Brockners
Cisco Systems

Email: fbrockne@cisco.com

Sudhendu Kumar
NC State University

Email: skumar23@ncsu.edu

Shwetha Bhandari
Thoughtspot

Email: shwetha.bhandari@thoughtspot.com

Camilo Cardona

NTT

Email: camilo@ntt.net

Diego Lopez
Telefonica I+D

Email: diego.r.lopez@telefonica.com