Software Identification and Entitlement Metrics

Information for Work-in-Progress version:
IMPORTANT: This specification is not a standard. It does not necessarily reflect the views of the DMTF or all of its members. Because this document is a Work in Progress, this specification may still change, perhaps profoundly. This document is available for public review and comment until the stated expiration date.
It expires on: 2012-03-15
Provide any comments through the DMTF Feedback Portal:
http://www.dmtf.org/standards/feedback

Version 0.9.2

Status: Informational

Publication Date: 12/9/2011

Expiration Date: 03/15/2012

DSP-IS0301

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit http://www.dmtf.org/about/policies/disclosures.php.

Software Identification and Entitlement Metrics
Software License Management Incubator

Version 0.9.2

Publication Date: 12/9/2011

DSP-IS0301

Status: Informational

Table of Contents

1 Abstract

1.1 Executive Summary

The emergence of cloud computing along with virtualization adds additional complexity to software license management for platform vendors, application providers and their customers.

The increased mobility of workloads and the ability to clone virtualized systems increases the challenge for organizations to track software license compliance associated with the virtualized instances. The mobility of licensed software to public and private clouds makes it more difficult to accurately identify and inventory deployed software, to trace its use and correlate the use to an entitlement. These challenges, however, create an opportunity to address customer pain points and to unlock the value and realize the efficiencies offered by these new virtualized and cloud technologies.

In order to fully realize the value of virtualization and cloud technologies standards are needed to sufficiently identify licensed software products, and to trace and gather the use of the software and other entitlement metrics across the span of deployments.

To effectively manage their licensed software product(s) and product usage, customers have the need to:

- Record and enumerate software product usage. This could encompass what instances, users, CPUs or other measurable units that may be running, where (e.g. whether in an operating system on hardware server, or a virtualized or cloud computing environment), with what device, and by whom (i.e. on which processor of a given hardware server).
- Uniquely identify the software licensed product(s) associated with a particular usage.
- Technically express product usage information of licensed product(s) for pre-deployment or reporting purposes. For example, software entitlement metrics requirements in a package such as Open Virtualization Format (OVF).

1.1.1 Scope

To achieve these goals, the Software License Management (SLM) Incubator was created and its charter approved in December 2010.

The Incubator was formed to develop recommendations focused on the challenges surrounding software licensing management and to move the industry in a direction to effectively manage licensed software product(s), and work toward interoperable solutions.

The intent of this work is to be applicable to licensed software. It is applicable to software products that are developed in various ways, including the use of open source software.

This white paper outlines the technical aspects required to address the requirements, use cases, scenarios and solutions identified. For example:

- The representation of the identity of a licensable product (i.e. virtual machine instance, on premise product, etc.)
- How it is associated with a running instance or a particular operating system

- Who and what (device) are assessing that instance, and
- The ability to discover if and where the product instance is running.

1.2 Acknowledgments

The authors acknowledge the contributions from the members of the DMTF Software License Management Incubator. The following persons were instrumental in the development of this specification:

- (Editor) Mr. Howard Hastings, Blazent, Inc.
- (Editor) John Parchem, Microsoft Corporation
- Mr. Shishir Pardikar, Citrix System Inc.
- Sharon Pitt, George Mason University
- Arul Murugan Alwar, HP
- Dr. Brad Topol, IBM
- Mr. Curt Barrentine, JP Morgan Chase
- Stephen DiGianno, JP Morgan Chase
- Ms. Monica Martin, Microsoft Corp
- Winston Bumpus, VMware Inc.

1.3 Recommendations

The SLM Incubator has identified four key recommendations for future work. These recommendations seek address the requirements identified in the preceding summary and suggest the development of:

- Standard for the identity of a licensed product offering

- Standard format for capturing the core entitlement metric requirements that reflect measurable product use rights,

- Standard log format and a normative schema to capture the consumption of an entitlement.

- Describe a process and use cases utilizing the above three standards that enable the automation of the core licensing management use cases and to enable the determination of the state of compliance to the corresponding license terms.

One intended usage of the above standards is that the product identity and the core metric requirements can be carried in an Open Virtualization Format (OVF) package for use by an automated deployment system or as part of a private or public cloud deployment package.

2 Common Terminology

Term	Definition	DMTF Reference	External Reference
Application	Software that provides functions that are required by an IT Service. Each application may be part of more than one IT Service. An application instance can run on one or more computing systems.	CIM_ApplicationSystem	
Bundle	A grouping of products which is the result of a marketing/licensing strategy to sell entitlements to multiple products as one purchased item. As these are multiple licensed products there may be no way to determine that an individual installed product was purchased as a bundle.	CIM_SoftwareFeature	
Client Software	The part of a client-server Application that the user directly interfaces with. For example: an email client.		
Client Access License (CAL)	A software license that legally permits client computers to connect to server software. CALs apply to either a "device" (as defined in the license agreement) or a "user". A Per-User CAL allows one user to connect to the server software. Any user can connect, but only one user may use a given CAL at any given time. Any number of devices may connect to the server software, but only a set number of users can connect to it at once. A Per device CAL operates in much the same way, but limits connections made by devices, rather than users. One CAL enables one device to connect to and use the server software, regardless of how many users are connecting.		http://en.wikipedia.org/wiki/Client_Access_License
Computing Device	The hardware technology upon which the software is installed or executed.		
Computing System	One or more virtual or physical computing devices including applicable operating system or firmware that support installation and execution of applications.		
Consumer	Consumer is a legal entity that purchases, installs, deploys or uses a product.		
Central Processing Unit (CPU)	An integrated circuit chip installed in a computing device comprised of one or more processors that perform the instructions of a computer's programs. Modern CPUs usually contain on-chip memory referred to as "level 1" cache.		
Data Center	A data center is a physical location that provides computing resources and may contain physical and virtual systems, storage and networking.		
Deployment	The process of installing a service instance in a reserved or prepared environment.	CIM_Action	
Deployment System	The system that installs a software package or appliance.		

Term	Definition	DMTF Reference	External Reference
Entitlement (Software)	Legal ownership of software license use rights as defined through agreements between a software purchaser and the software copyright holder.		
Feature (Software)	A collection of software elements that performs a particular function or role of a software product. This level of granularity is intended to be meaningful to a consumer or user of the application to choose. This concept allows software products or application systems to be decomposed into units that have a meaning to users rather than units that reflect how the product or application was built (i.e., software elements).	DMTF Application Management Model	
Globally Unique Identifier (GUID)	A unique reference number used as an identifier in computer software.		
Guest Software	The software running on a virtual machine, stored on the virtual disks, that runs when a virtual machine is powered on The guest is typically an operating system and some user-level applications and services.		
Identity	A name that is used to uniquely identify a user or person for the purposes of granting/assigning software use rights. Example might be the username "SmithJ".		
Image	Exact copy of the storage (disk) contents of a computing device for the purposes of recovery or provisioning of a duplicate system. This co compasses the full instantiation of a deployed operating environment, all applications, data and configuration settings.		
Infrastructure as a Service (IaaS)	A service delivery model where the capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).		http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909616
Instance (Software)	An installed copy of a software product or application whose presence can be identified through manual or automated means.		

Term	Definition	DMTF Reference	External Reference
IT Service	A set of related functions provided by IT systems in support of one or more business areas, which in turn may be made up of software, hardware and communications facilities, perceived by the customer as a coherent and self-contained entity. An IT service may range from access to a single application, such as a general ledger system, to a complex set of facilities including many applications, as well as office automation that might be spread across a number of hardware and software platforms.		http://www.knowledgetransfer.net/dictionary/ITIL/en/IT_Service.htm
License (Software)	Legal rights to use software in accordance with terms and conditions specified by the software copyright owner		
Licensee (Software)	A legal entity, typically a person or organization, contractually bound to a given software license agreement that provides rights to use the associated software in accordance with the terms and conditions as specified by the copyright owner.		
Physical Location	A physical place associated with a specific geographical reference.		
Platform	A combination of hardware and software operating environment upon which applications can be installed and operate.		
Platform as a Service (PaaS)	A service delivery model where the capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.		http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909616
Processor	The set of logic circuitry within a CPU that responds to and processes the basic instructions that perform the intended functions of a computing device. Modern processors may contain more than one core and/or have multithread capabilities that allow for execution of multiple instructions.		
Product Activation	Activation associates an event that recognizes the intended use of a software product with a specific device or system.		
Product Edition	A specific edition (i.e. a SKU variation) related to a specific version of a licensed software product.		
Product Version	A specific release of a licensed software product.		
Provisioning (Software)	The process of selecting, reserving resources, or creating an instance of a service offering.		

Term	Definition	DMTF Reference	External Reference
Relationship (Software)	A connection or interaction between one or more products, solutions, software components, applications or IT Service. Suites and bundles are clear examples of where knowledge of the relationships involved is important to properly manage a product or solution during packaging, provisioning, or addressing license compliance. A more complex example is an IT Service and its requisite applications, where documenting and maintaining the relationships between all the various elements involved is crucial for proper change and performance management.		
Resource	A generic term that includes IT infrastructure, people, money, hardware components or anything else that might help to deliver an IT Service.		
Server (Software)	The part of a client-server application that the client software interacts with. For example: an email server.		
Service Delivery Model	The approach used to deliver a given service to its intended consumers. As it applies to software, Service Delivery Models vary to address different ways in which software can be sold, managed and accessed, such as Software as a Service vs. software that must be installed by the customer on their own server, etc.		
Service Catalog	Self-service portals and/or eProcurement systems that contain a list of available services and products that can be requested/ordered in an automated manner.		
Software as a Service (SaaS)	A service delivery model where the capability provided to the consumer is to use the provider's application(s) running on a cloud infrastructure accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.		http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909616
Software Catalog	A subset of the service catalog that contains the list of software titles available for request/order through self-service portals and/or eProcurement systems.		
Software Element	A general term that is used to mean one discrete software part of a more complex software product or application.	CIM_SoftwareElement	
Software Product	One or more applications governed by one license, which may include procedures, documentation and data, commercially available as a single item for a fee to a licensee.	CIM_Product	

Term	Definition	DMTF Reference	External Reference
Software Package	A set of related software components that are combined into a single payload or a distributable installable item. For example, a software package is a set of files that can be used to install software on a computing device and can be distributed via CD or electronic means. An Open Virtualization Format (OVF) package is an example of a package for cloud deployment.		
Software Suite	A set of individually licensable software products or software features, that is combined and licensed as a separate single product.		
Solution	A combination of one or more applications, which may also include one or more computing systems, made available as a single IT Service.	CIM_Application System	
Solution Multiplexing	Correlation of use of multiple individually licensed software components that compose a licensed application. For example, a web tier application with a single sign on to a data tier application may need to track those requests on a user's behalf that result in requests to the data tier		
Stock-Keeping Unit (SKU)	A number or string of alpha and numeric characters that uniquely identify a product. SKUs are often called part numbers, product numbers, and product identifiers, and may be represented by a universal number such as a UPC.		http://www.techterms.com/definition/sku
Thread	In programming, a part of a program that can execute independently of other parts. Operating systems that support multithreading capabilities of processors enable programmers to design programs whose threaded parts can execute concurrently.		
Uniform Resource Identifier (URI)	A string of characters used to identify a name or a resource usually on the internet.		
User	A person who uses an IT service. Users are distinct from customers, as some Customers do not use IT services directly.		
Virtual	As it applies to information technology, not physically existing as such but made by software to appear to do so.		
Virtualization Platform	Infrastructure enabling virtualization provided by a host system that enables the deployment of virtual systems.		
Virtual Machine (VM)	The virtual representation of a computing device including the CPU, memory, controllers, network interfaces, and storage that supports the execution of guest software in a virtualized environment.		

Term	Definition	DMTF Reference	External Reference
Virtual System	A virtual operating system environment that includes virtual machine(s), the operating systems and applications. The virtual system is a computer system operated in a virtualized environment that includes its software running in that environment.		

3 Software Licensing Concepts and Environment

Licensed software products can be packaged or made available in many different ways for a consumer. The variability of packaging and distribution methods increases the complexity to uniquely identify and to track usage for a licensed software product(s). Licensing models, programs, and licensing terms also may influence how a licensed software product is packaged or made available.

Licensed software products are packaged or made available in some of the following ways:

- End user acquired products
- Organizational acquired products
- Single Executable
- Single product
- Suite
- Server offering
- Product bundling

Software products are licensed based on factors such as: its use, the party that will use or access it, on what device, number of processors, amount of system memory, running location, and what other products are required to run it.

Licensed software products are to be used based on an entitlement(s). Common entitlements include:

- End User License Agreement (EULA)
- Site License
- Subscription
- Type of use or consumption (personal/business)
- Client Access License (CAL)
    - Device
    - User
    - Concurrent user
    - Internal, external
- Instance license
- License specific to a product

The use rights for and consumption of a licensed software product are tied to its entitlement. Use rights typically provide boundaries for:

- Operating System requirements
- License Life span
- Transfer rights
    - Computer System
    - Physical Location
- Number of unique Users
- Number of Installations
- Number of unique Devices
- Maximum number of processors\ virtual processor

- Maximum amount of system memory
- Location
- Type of Device\system (desktop, server, phone, …)

Entitlement metric are events that measure the installation and use of a software product instance. Entitlement metrics may be gathered on the consumption of a licensed software product instance. The characteristics of and packaging for a licensed software product are defined to identify and track consumption of that product instance.

A licensed software product and its constituent components, where applicable, should be normatively identifiable to enable traceability through its lifecycle for identification and consumption purposes to correlate entitlement metrics.

Today products are delivered to a desktop, through a virtualized environment, from the cloud and through an enterprise data center. The characteristics of and consumption of a licensed software product instance should be traceable regardless of the environment of which it may be made available.

Application virtualization is where the technology isolates and packages applications in a way that they are dissociated from the underlying machine and operating system. Correspondingly, desktop virtualization is where the technology isolates the entire user experience (or desktop) from a physical machine and makes it available across one or more client devices.

The characteristics of a licensed software product may be captured in a computable package such as an Open Virtualization Format (OVF) package. OVF 1.1 is a DMTF Standard, and very recently was approved as an ISO/IEC International Standard. OVF is a standard format for packaging virtual appliances or machines. This allows the virtual machine to be more easily moved and deployed across virtualization platforms.

Licensed software products may also be available or deployed in the cloud. As shown in Figure 1 there are currently several acknowledged categories of clouds: private, public, community, and hybrid.

- Private Cloud- A private cloud is one that serves a single organization. Private clouds facilitate security, compliance, and quality of service improvements due to network optimization and isolation.
- Public Cloud- A public cloud is one that is available to the general public and is owned by an organization selling cloud services. Public clouds provide efficiencies through large economies of scale.

- Community Cloud- A Community cloud shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally. The costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the benefits of cloud computing are realized
- Hybrid Cloud- A hybrid cloud is a composition of two or more clouds (private, public, or community) that remain separate and autonomous but allow for data and/or application portability between themselves via standard or proprietary technologies.

The figure evidences the complexity of identifying and tracking consumption for licensed software product instances through the product lifecycle.

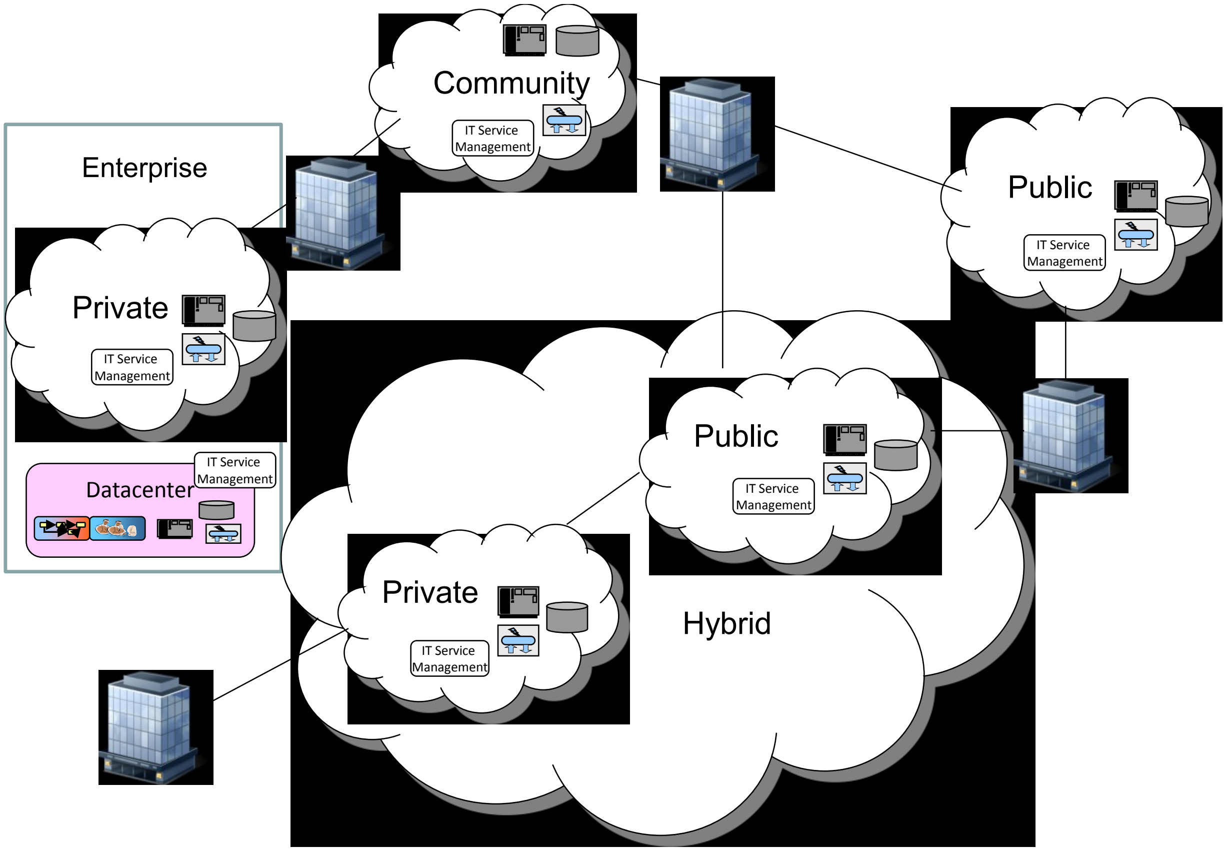


Figure 1 Cloud Deployment Environments

In the following sections, we identify several key scenarios that further describe the identification of and consumption for a licensed software product instance through its lifecycle.

4 Scenarios

This section describes key entitlement metric scenarios that focus on top level product deployment scenarios. The entitlement metric scenarios can be summarized: What software do I have; who or what is using the software and where and when is the software being accessed. The scenarios are written to illustrate the requirements needed to answer these questions in a normative process across a range of deployment scenarios. The uses cases derived from the deployment scenarios place requirements on software packaging, deployment, installation and upgrade, and runtime logging by the software. These requirements are the focus of this white paper.

Also note that parts of the included scenarios are outside of the scope of Software License Management Incubator and their inclusion is not an effort to make a recommendation for standardization. They are listed to explore and illustrate the requirements for the data artifacts required to standardize product identification and entitlement metrics.

The scenarios focus on separate environments with some overlapping requirements.

1. The packaging and development of software, solutions or applications for deployment in a cloud.
2. Desktop deployment of software in an enterprise
3. Software delivered as a service
4. Application and desktop virtualization
5. The packaging and installation of software on a server in a data center. It is possible that the data center is an enterprise data center or a cloud data center.
6. Cloud deployment scenarios

The packaging and development requirements focus on software identity and a manifest of licensable software within a package. The data center use cases utilizes a software identity artifact, but also focuses on the entitlement metrics to track the “who”, “what”, “when” and “where” of an installed software instance throughout its lifecycle. Entitlement metrics should be tracked, discoverable and available for reporting. The cloud deployment scenarios focus on the additional requirements of tracking software utilization or entitlement metrics in a remote data center. The scenarios are described from the perspective of the described set of actors.

4.1 Scenario Actors

A group of actors are identified that may participate in the scenarios defined in this document:

- IT Pro/Administrator
  a. Software or Asset Administrator: Corporate software assets – match assets against the licenses acquired [Persona: IT Pro]
  b. System Administrator [Persona: IT Pro]
  c. Deployment Manager – Deployment on virtual machine [Persona: IT Pro]
- Product Provider
- Procurement Manager [Persona: Business Development Manager]
- Business Manager: Vendor or contract manager [Persona: Business Development Manager]
- Compliance officers
  a. Compliance Manager: Assesses reports against regulatory, corporate and other requirements (i.e., Business Conduct). Responds to information provided by IT Auditor. Compliance Manager develops set of criteria for IT auditor. [Persona: Business Development Manager]
  b. IT Auditor (internal) [Persona: Business Development Manager]
  c. Auditor (external): Reviews Compliance Manager's output and verifies or certifies the results, and approves the compliance plans.
- Security Manager: Access control [Persona: none defined]
- Service Manager [Persona: IT Pro]
- End user

4.2 Usage Scenarios

The usage scenarios focus on the usage of licensed software products. Usage scenarios are identified for on-premise and enterprise or cloud data centers.

4.2.1 End user scenarios

The end user scenarios are a set of scenarios that focus on software used on a desktop computer, terminals or mobile devices. Multiple core scenarios are in practical use today:

- On-premise deployment of end user requested desktop software from a software offering catalog
- End user use of virtual applications streamed from a server onto a client device.
- End user access of server resources.
- End user use of a virtualized desktop.

4.2.2 End User Request Software for a Desktop System

An end user requests software from an enterprise software catalog for use on a desktop target system. The software is licensed by the enterprise and available for use. The software is provided, installed and then used by the end user.

Entitlement, utilization, and product metrics are gathered from both the software catalog and the user's desktop system to track the lifecycle of the software instance including additions, upgrades and removal. Refer to 8 Appendix A for a detailed look at the use cases and the need for entitlement metrics across the life cycle of a product for this scenario.

4.2.3 End User Entitlement to Software use via a Licensing Server

An enterprise acquires software and a number of licenses for a software product and ties the acquired licenses to a licensing server.

End user software is deployed and installed though any means, CD, software download or preload in an image file. As part of installation a license server is discovered or the user is prompted to provide the address of the license server.

When a client uses the software the software connects to the license server, and requests a use license.

- If the customer has a "Concurrent User License" and the software is available for use, the client is granted access.
- If the customer has "Per User/Device License" and if a license is available, the client is granted access.
- If the license server is not found, the client is either not allowed to run the software or given a temporary use license.

Entitlement metrics should be gathered on the acquisition and release or expiry of a license and the products use. Entitlement metrics are also generated on the licensing server.

The license is validated per usage. User access is based on consumption (per-use).

4.2.4 End User Access to Software as a Service

Software may also be provided through an application delivery system or appliance which provides enterprise level services to an end user commonly through a web based interface. The software is delivered as a service to the end user.

In this scenario the software mainly exists on the servers providing the service. Often a thin software client (i.e. applet, or ActiveX control) is downloaded or installed on the client. Entitlement metrics are gathered on the servers providing the service.

4.2.5 End User use of virtualized application

Stateless client application can be virtualized on a server and run on demand by a user and delivered to the user through a remote console protocol like Microsoft Remote Desktop Services or Citrix XenApp. In these cases it is assumed that the license tracking would be performed on the server that is hosting the application.

4.3 Packaging and Deployment Scenarios

Licensed software products can be packaged to form solutions, suites, bundles and virtual appliances. To ease the burden of licensing management the deployment scenarios place requirements on the software developer, systems integrator or software vendor to include the product identification for each licensable product in the package in a normative and machine readable format. The following scenarios illustrate the value of a normative list of licensable products and required entitlement metrics contained in the software deployment package.

4.3.1 Packaging for data center deployments

Broader use of virtualization in enterprise data centers or private clouds has changed the typical data center deployments from an install in place or provisioning of software on a server to the deployment of fully provisioned images. Without the knowledge of the licensed software contained in the image it is hard for the IT Pro to assure licensing compliance. A normative manifest in the deployment package could contain both a list of the installed products and the metric requirements for those products to complete the following scenario.

A standards compliant OVF deployment package is scheduled for deployment in a data center. Before deployment the software administrator opens the package and extracts the product identification section from the package and if available any packaged license entitlements to assure that the proper entitlements are available to comply with the licensing requirements of the products in the package.

Based on the available entitlements for each product the software administrator either appends to an existing set of entitlement metric requirements or places an entitlement metric requirements data structure into the package for deployment by the deployment manager.

4.3.2 Cloud deployment

Three cloud deployment scenarios have been identified.

Product provider packages a solution targeted at a cloud deployment that contains multiple separately licensed products. Included in the package is a manifest that contains the software Identification of all of the products contained in the product.

Deployment manager receives a packaged solution to deploy into an enterprise private cloud. Before deployment the licensed products in the package are inventoried and checked against available entitlements to assure license usage compliance. The deployment manager determines whether to constrain a deployment, migration or movement of a package.

An IT Pro wants to move a line of business application from a set of dedicated servers to the enterprises private cloud. The IT Pro queries each of the servers to obtain a list of the software products contained in the servers that are used in the application. The IT Pro includes this list in the application package being developed to deploy the application in the private cloud. The IT Pro also delivers the list to the asset manager to obtain free up or transfer the entitlements that are required to run the package in the private cloud. IT Pro also places a data structure into the package that includes the entitlement metric requirements before deploying the package on the cloud service.

4.3.3 Product deployment in an enterprise data center

Three enterprise data center scenarios have been identified.

Compliance Manager takes an inventory of all the licensable software products in a Data Center. For each of the discovered products, entitlement metrics are harvested from the system logs identifying life cycle and usage events for the product. The Compliance Manager correlates the lifecycle and usage logs against the relevant entitlements to assure that the data center is in compliance with the entitlements for the products. This includes inventory usage for product instances accessed indirectly through another licensed software product on behalf of a user.

The Software Administrator uses the correlation between the entitlements and the installed products, and entitlement metrics to forecast, adjust the continuing license requirements. For example, product usage and activations could determine the necessity to adjust licensing requirements.

An IT Pro set up an automated system to track product usage against a set of entitlement policies set by the Software Administrator. An example is a service that requires a Client Access License (CAL) per unique user or client device. The IT Pro registers for the relevant indications (events) based on the standard set of entitlement metrics delivered by each software product instance. Based on the dynamic usage events received each month the IT Pro is able to deliver a report to the Software Administrator with recommendations to increase or decrease the available entitlements or the type of entitlement required. For example, the IT Pro uses an automated system to differentiate per-processor or per-server product usage for the same license type and different entitlement metrics. Or, the IT Pro tracks product usage based on access to domain and member servers irrespective of where the usage occurs.

4.3.4 Datacenter/Server software deployment

During deployment of software or a packaged software solution, suite, or bundle, a deployment manager is able to discover the products contained in the package and the required entitlement metrics before the installation on the server(s). Entitlement metrics requirements are expressed in a standard way.

The Deployment Manager uses the available metric requirements to select a suitable server and/or create a virtual machine template that matches the entitlement metrics requirements.

When a licensed software product(s) are deployed, an artifact for each product is created that captures the relevant entitlement metrics relating to the environment into which the software is deployed (i.e. location, hardware definition, and VM definition).

As part of the deployment the Deployment Manager may also configure any settable system configurations parameters that are expressed in the entitlement metric requirements as packaged by the software asset manager.

4.3.5 Administrator Deployed Software on Desktop System

A software administrator deploys software products to an individual or a group of desktop system. Entitlement metrics are generated when the software is actually deployed, used or removed from a system. An example would be an IT managed installation of an antivirus product across an enterprise's desktop systems.

5 Management Data Artifacts Requirements

To support the monitoring and management of relevant events related to software entitlements, events for the installed instance should be logged according to the defined standards for entitlement metrics. The Software Usage Lifecycle section lists the relevant but not exhaustive list of software life cycle events that a system should capture.

5.1 Software Usage Lifecycle

To support the monitoring of relevant events related to software entitlements these events for the software instance should be logged according to a to be defined standard for entitlement metrics. The following list shows the relevant software life cycle events that should be captured.

- Request for software
- Acquisition of software and/or entitlement
- Addition to software offering catalog or software made available for use (consumption).
- Deployment or installation
- Product use
- Application migration
- VM migration
- Removal
- Modification of the software instance
- Upgrade or Downgrade
- Retirement

An example of a detailed description of a complete lifecycle, the end user request for software scenario is included in Section 8 Appendix A. The following two sections Product Identification and Entitlement Metric Requirements describe the relevant aspects that should be captured in each event.

5.2 Artifact Overview

5.2.1 Product Identification

A licensed software product instance should be identifiable by a normative set of properties. This structure should contain all of the information required to completely identify the software product and optionally to describe the entitlement metrics that the product generates through its life cycle. For example:

- Vendor
- Software ID
- Software ID Type
- Product Title
- Product Category
- Product Family

- Edition
- Release Date
- Software version
- Software edition
- Version Type
- Patch Level
- Product Dependencies
- Entitlement Metric Requirements[] (usage, user, processors, OS version, operational state …)
- Certificates[]
- Security Token

5.2.2 Entitlement Metrics

Entitlement metrics capture the relevant measureable or discoverable events in a system that relate to the licensing of a product.

- Product identification as described in section 5.2.1 .
- The environment, unique users, devices, and usages related to the licensed software product instance
    - Number and type of virtual processors\cores
    - Amount of memory
    - Number and type of underlying hardware processors\cores
    - Timestamp
    - Administration domain
    - File location of binaries\executable
    - User Identity,
    - User privilege
    - privilege level ,
    - Client device identity and\or
    - server identity
    - Tenant
    - Server or device type
    - Operating system Identity
    - Event type (reflect the life cycle)
        - Use
        - Operation state change (running, paused, stopped …)
        - Installation
        - Uninstall
        - Migration departure
        - Migration arrival
        - Upgrade/
        - Servicing
        - …
    - …
- Usage Metrics (running time .., processor time, memory usage, timer based metrics)

6 Relevant Standards

Organization	Specification	Date	Description
DMTF	DSP0140 Application White Paper	June 2003	The CIM Application Management Model is an information model that describes the details commonly required to manage software products and applications. This model can describe applications with various structures – ranging from standalone desktop applications to a sophisticated, multi-platform distributed, Internet-based application. Likewise, the model can be used to describe a single software product as well as a group of interdependent software products that form a business system.
DMTF	DSP0243 Open Virtualization Format (OVF)	January 2010	The Open Virtualization Format (OVF) Specification describes an open, secure, portable, efficient and extensible format for the packaging and distribution of software to be run in virtual machines.
DMTF	DSP1054 1.2.0 Indications Profile	June 2011	The Indications Profile defines the CIM elements that are used to subscribe for indications of unsolicited events, to advertise the possible indications, and to represent indications used to report events in a managed system.
OASIS	Solution Deployment Descriptor (SDD)	Sept. 2008	This specification defines schema for two XML document types: Package Descriptors and Deployment Descriptors. Package Descriptors define characteristics of a package used to deploy a solution. Deployment Descriptors define characteristics of the content of a solution package, including the requirements that are relevant for creation, configuration and maintenance of the solution content. The semantics of the descriptors are fully defined, allowing software implementations to precisely understand the intent of the descriptor authors and to use the information provided in the descriptors to support solution deployment.
IETF	Application Management MIB	May 1995	This specification defines an experimental portion of the Management Information Base (MIB) for use with network management protocols in the Internet Community. In particular, it defines objects used for the management of applications. This MIB complements the System Application MIB, providing for the management of applications' common attributes which could not typically be observed without the cooperation of the software being managed
ISO/IEC	19770-2:2009 Software Identification Tag Standard	2009	ISO/IEC 19770-2:2009 establishes specifications for tagging software to optimize its identification and management.

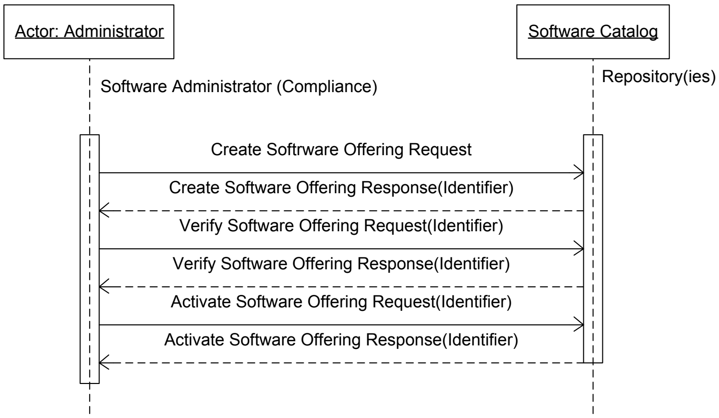
7 Standards Currently Under Development

ISO/IEC	19770-3 Software Entitlement Tag Standard	Under development	ISO/IEC 19770-3 focuses on capturing and defining the information necessary to describe how software may be used, known as the entitlement. This standard will provide a framework and criterion of measurement for creating unambiguous definitions of entitlements.

8 Appendix A

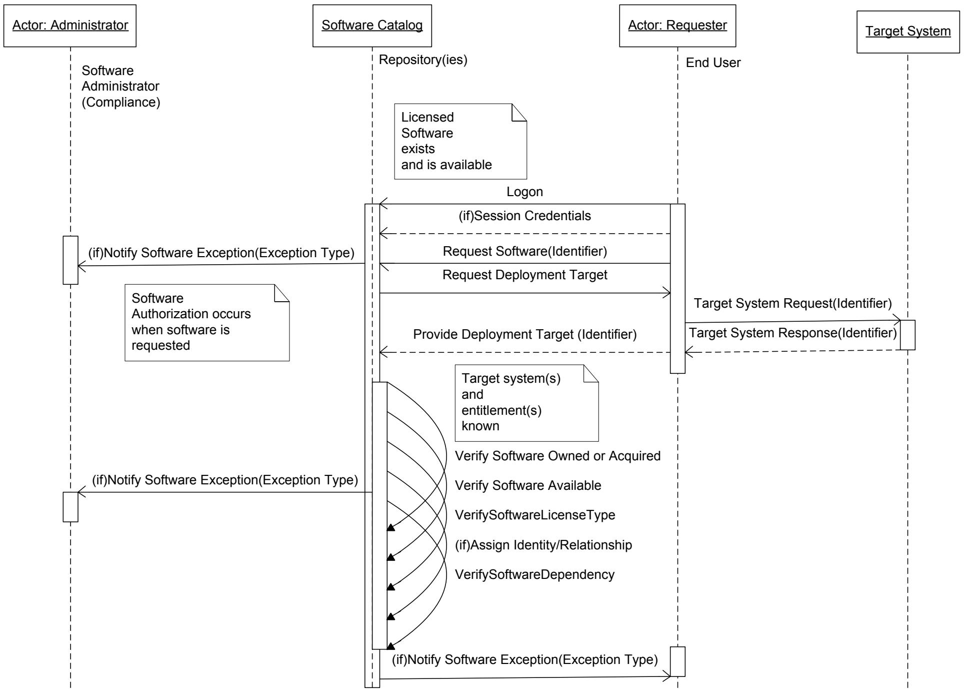
The following are the use cases for just one of the above scenarios, automated end user requested software for their desktop system. The diagrams are shown only as an illustration of the type of interactions and the metrics required completing the scenario and track and managing the software licenses involved in the scenario. The scenario and the use cases described were used to determine the necessity and the requirements for software license management. Although this use case addresses a desktop deployment scenario, many of the use cases around the usage metrics and licensable events are true for the server scenarios that were discussed in the incubator.

Make Software Available
An administrator adds a software offering to a software catalog and makes it available for request. An administrator may verify the software offering and activate it in the software catalog.


Actor: Administrator
Software Catalog
Software Administrator (Compliance)
Repository(ies)
Create Softrware Offering Request
Create Software Offering Response(Identifier)
Verify Software Offering Request(Identifier)
Verify Software Offering Response(Identifier)
Activate Software Offering Request(Identifier)
Activate Software Offering Response(Identifier)

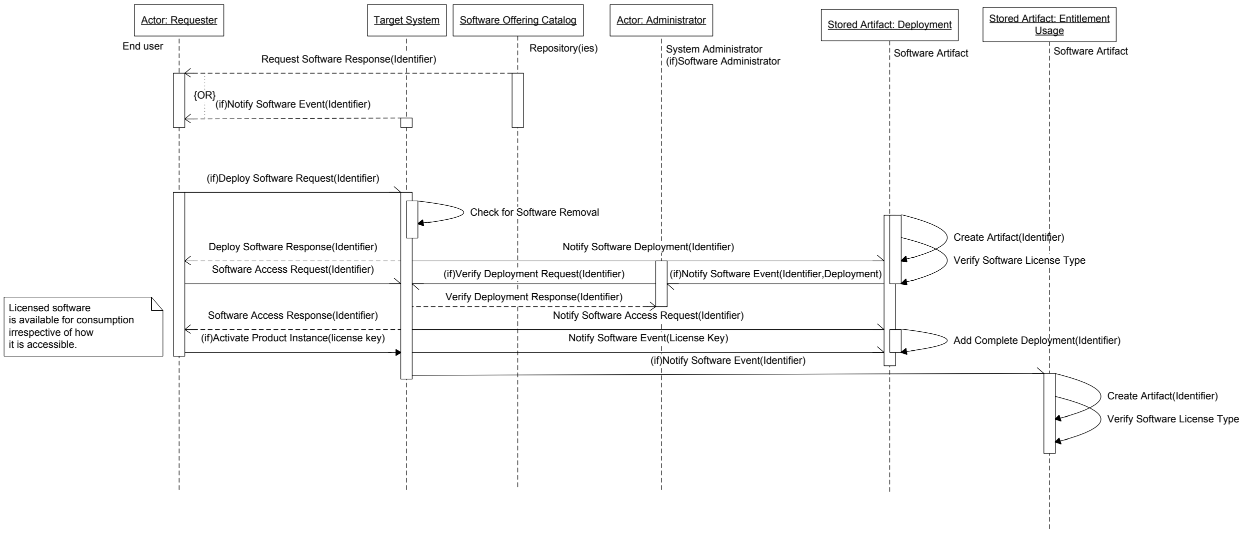
Request Software

A requestor (actor) requests software from a software catalog to deploy to a desktop system. The software request and deployment environment are used to complete the request. Software should be owned and available in the software offering to fulfill the request. As this is a licensable event, other licensing and software dependency checks are made before the software request can be completed. An authorization occurs when the request for software and other checks are made.


Actor: Administrator
Software Catalog
Actor: Requester
Target System
Software Administrator (Compliance)
Repository(ies)
End User
Licensed Software exists and is available
Logon
(if)Session Credentials
Request Software(Identifier)
Request Deployment Target
(if)Notify Software Exception(Exception Type)
Software Authorization occurs when software is requested
Target System Request(Identifier)
Target System Response(Identifier)
Provide Deployment Target (Identifier)
Target system(s) and entitlement(s) known
Verify Software Owned or Acquired
Verify Software Available
VerifySoftwareLicenseType
(if)Assign Identity/Relationship
VerifySoftwareDependency
(if)Notify Software Exception(Exception Type)
(if)Notify Software Exception(Exception Type)

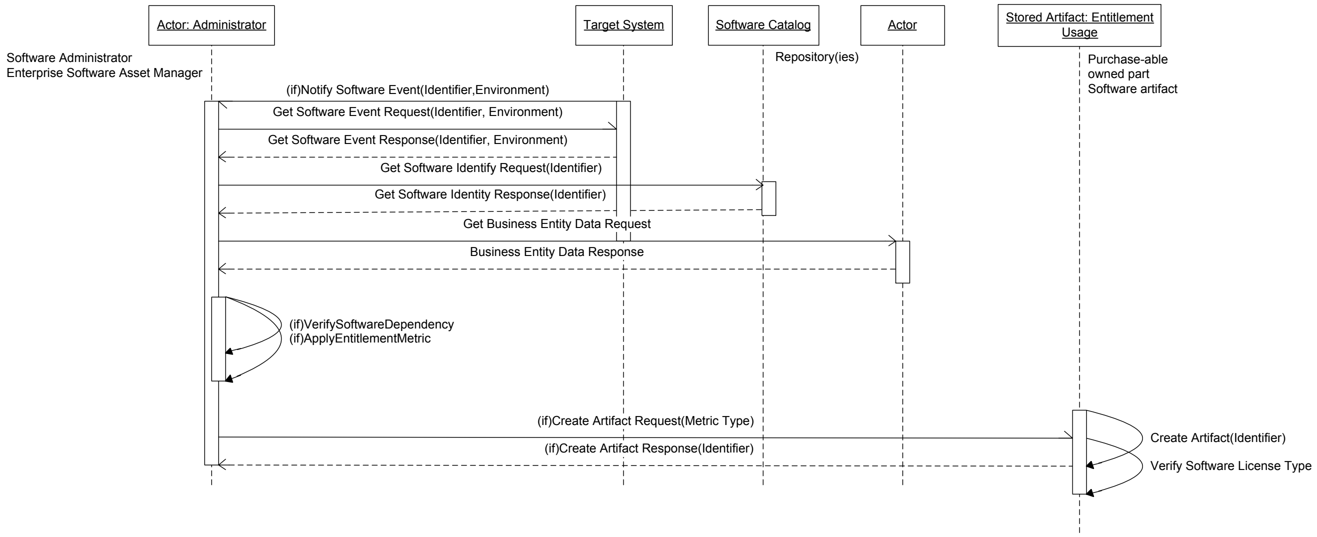
Deploy Software

After a software request is complete, software is delivered to the desktop system for deployment. Deployment can be by manual or automated means. Entitlement metrics are captured throughout the deployment of the software. A licensing key may be required to activate the software instance.


Actor: Requester
Target System
Software Offering Catalog
Actor: Administrator
Stored Artifact: Deployment
Stored Artifact: Entitlement Usage
End user
Repository(ies)
System Administrator
(if)Software Administrator
Software Artifact
Software Artifact
Request Software Response(Identifier)
{OR}
(if)Notify Software Event(Identifier)
(if)Deploy Software Request(Identifier)
Check for Software Removal
Create Artifact(Identifier)
Verify Software License Type
Deploy Software Response(Identifier)
Notify Software Deployment(Identifier)
Software Access Request(Identifier)
(if)Verify Deployment Request(Identifier)
(if)Notify Software Event(Identifier,Deployment)
Verify Deployment Response(Identifier)
Licensed software
is available for consumption
irrespective of how
it is accessible.
Software Access Response(Identifier)
Notify Software Access Request(Identifier)
(if)Activate Product Instance(license key)
Notify Software Event(License Key)
Add Complete Deployment(Identifier)
(if)Notify Software Event(Identifier)
Create Artifact(Identifier)
Verify Software License Type

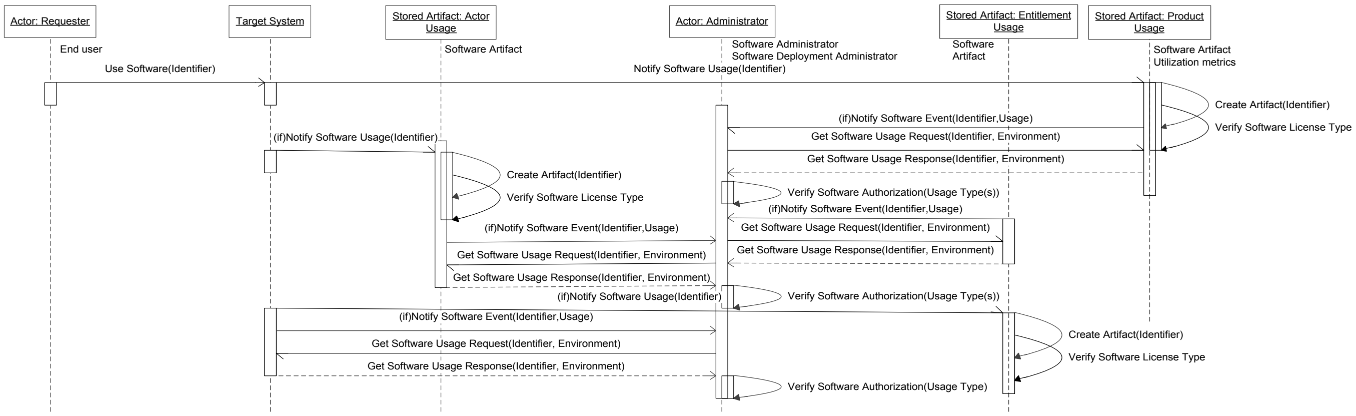
Identify Relationship

A software request, and deployment and access are related to and accounted for by a business entity. Entitlement metrics are created and captured on the software instance once the business entity-software instance relationship is set. The business entity is the consumer of the entitlement. Creating the relationship between the software request, the business entity, and the entitlement could occur in parallel to or in conjunction with other phases (i.e. Request Software, Deploy Software).


Actor: Administrator
Target System
Software Catalog
Actor
Stored Artifact: Entitlement Usage
Software Administrator
Enterprise Software Asset Manager
Repository(ies)
Purchase-able
owned part
Software artifact
(if)Notify Software Event(Identifier,Environment)
Get Software Event Request(Identifier, Environment)
Get Software Event Response(Identifier, Environment)
Get Software Identify Request(Identifier)
Get Software Identity Response(Identifier)
Get Business Entity Data Request
Business Entity Data Response
(if)VerifySoftwareDependency
(if)ApplyEntitlementMetric
(if)Create Artifact Request(Metric Type)
(if)Create Artifact Response(Identifier)
Create Artifact(Identifier)
Verify Software License Type

Use Software

Entitlements metrics are captured when a software instance is accessed and used for the lifecycle of the software instance.

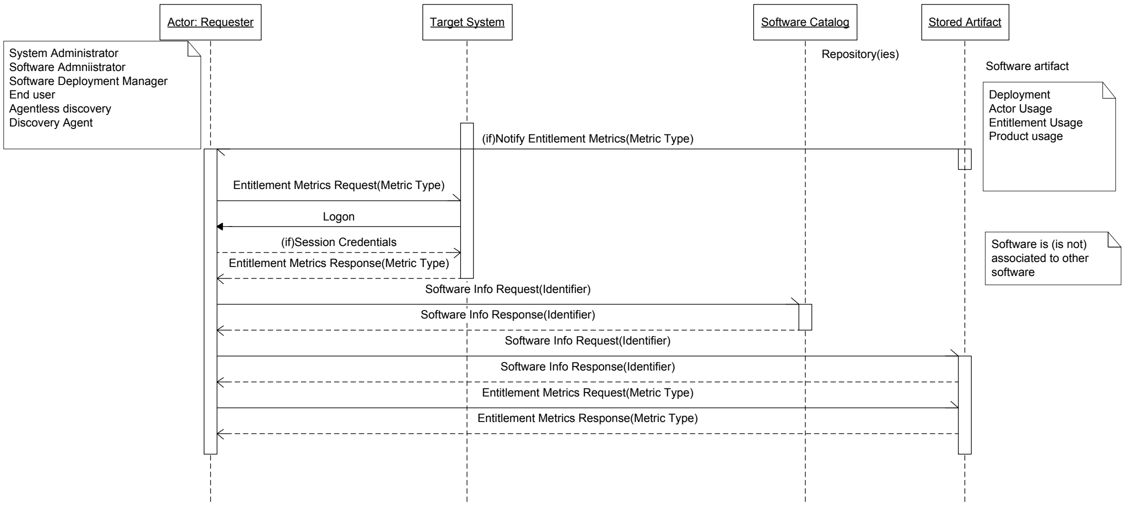

Actor: Requester
Target System
Stored Artifact: Actor Usage
Actor: Administrator
Stored Artifact: Entitlement Usage
Stored Artifact: Product Usage
End user
Software Artifact
Software Administrator
Software Deployment Administrator
Software
Artifact
Software Artifact
Utilization metrics
Use Software(Identifier)
Notify Software Usage(Identifier)
Create Artifact(Identifier)
Verify Software License Type
(if)Notify Software Event(Identifier,Usage)
Get Software Usage Request(Identifier, Environment)
Get Software Usage Response(Identifier, Environment)
(if)Notify Software Usage(Identifier)
Create Artifact(Identifier)
Verify Software License Type
Verify Software Authorization(Usage Type(s))
(if)Notify Software Event(Identifier,Usage)
Get Software Usage Request(Identifier, Environment)
Get Software Usage Response(Identifier, Environment)
(if)Notify Software Event(Identifier,Usage)
Get Software Usage Request(Identifier, Environment)
Get Software Usage Response(Identifier, Environment)
(if)Notify Software Usage(Identifier)
Verify Software Authorization(Usage Type(s))
(if)Notify Software Event(Identifier,Usage)
Create Artifact(Identifier)
Get Software Usage Request(Identifier, Environment)
Verify Software License Type
Get Software Usage Response(Identifier, Environment)
Verify Software Authorization(Usage Type)

Change Software

Requests to change a software instance may occur during its lifecycle and metrics captured on licensable events. Software may be added (i.e. for upgrade or new) or software removed (i.e. for replacement). A software upgrade may require a new software request and deployment, and removal of existing software. Such a request may result in a change in entitlement.

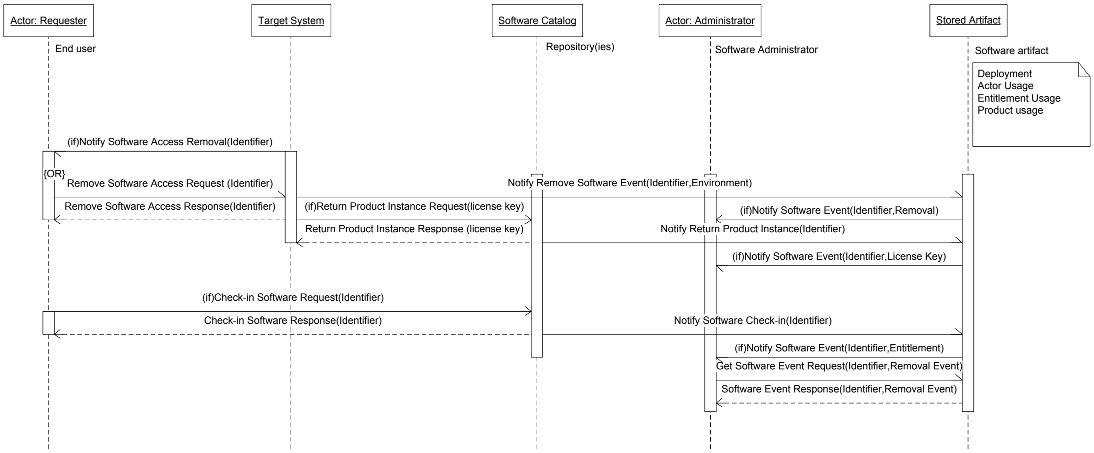
Discovery Reporting

The Entitlement metrics are available for discovery and reporting. The metrics are created at key points in the software instance lifecycle - at deployment, by actor usage, when an entitlement is used, and if the software instance is used or accessed (utilization).


Actor: Requester
Target System
Software Catalog
Stored Artifact
System Administrator
Software Admniistrator
Software Deployment Manager
End user
Agentless discovery
Discovery Agent
Repository(ies)
Software artifact
Deployment
Actor Usage
Entitlement Usage
Product usage
(if)Notify Entitlement Metrics(Metric Type)
Entitlement Metrics Request(Metric Type)
Logon
(if)Session Credentials
Entitlement Metrics Response(Metric Type)
Software is (is not) associated to other software
Software Info Request(Identifier)
Software Info Response(Identifier)
Software Info Request(Identifier)
Software Info Response(Identifier)
Entitlement Metrics Request(Metric Type)
Entitlement Metrics Response(Metric Type)

Remove Software

The software instance may be retired or returned to the software offering catalog when a request is made to remove that instance. The software may be returned to the software catalog and made available for re-deployment. If required, licensing keys may also be returned. The entitlement metrics are created to reflect the removal of the software instance and the associated entitlement.


Actor: Requester
End user
Target System
Software Catalog
Repository(ies)
Actor: Administrator
Software Administrator
Stored Artifact
Software artifact
Deployment
Actor Usage
Entitlement Usage
Product usage
(if)Notify Software Access Removal(Identifier)
{OR}
Remove Software Access Request (Identifier)
Remove Software Access Response(Identifier)
(if)Return Product Instance Request(license key)
Return Product Instance Response (license key)
Notify Remove Software Event(Identifier,Environment)
(if)Notify Software Event(Identifier,Removal)
Notify Return Product Instance(Identifier)
(if)Notify Software Event(Identifier,License Key)
(if)Check-in Software Request(Identifier)
Check-in Software Response(Identifier)
Notify Software Check-in(Identifier)
(if)Notify Software Event(Identifier,Entitlement)
Get Software Event Request(Identifier,Removal Event)
Software Event Response(Identifier,Removal Event)