INTERNATIONAL STANDARD

ISO/IEC 19086-2

First edition
2018-12

Cloud computing — Service level agreement (SLA) framework —

Part 2: **Metric model**

Informatique en nuage — Cadre de travail de l'accord du niveau de service —

Partie 2: Modèle métrique

Reference number
ISO/IEC 19086-2:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/IEC JTC1, *Information technology*, Subcommittee SC 38, *Cloud Computing and Distributed Platforms*.

A list of all parts in the ISO 19086 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

The measurement of properties of cloud services, especially for the purpose of cloud service level agreements (SLAs), presents many challenges, which inhibit the uptake of cloud services and inhibit the overall effectiveness of the cloud services marketplace. Metrics in practice are usually described using natural languages, typically in 'plain English', which is often difficult to understand, compare, and implement. Such definitions of metrics lead to many problems. Typical concerns include:

— **Clarity:** The metric definition may be incomplete, ambiguous, illogical, self-contradictory, or not defined at all. For example, cases exist where 'availability' is defined in ways which have little to do with generally accepted definitions of 'availability'; where the definition is such that the service can be unavailable for the majority of the time yet the metric will show 100 % availability; where the metric requires continuous monitoring, which is actually not possible; or where the provider is able to determine at its sole discretion what the result is.

— **Comparability:** It may be impractical or effectively impossible to compare different services in terms of their promised service levels because of the significant inconsistency in how their respective metrics and SLOs/SQOs are defined.

— **Implementation:** It may be impractical or even impossible to measure the metric in practice, and to determine whether promised service levels have been met or not.

This document has been developed to help address these and similar concerns. It includes technical content, but the high-level concepts are expected to be understandable by non-technical individuals who understand the business context for metrics. It provides a metric model that defines the conditions and rules for performing a measurement and understanding the result.

A metric complying with the model defined by this document addresses the concerns above:

— **Clarity:** A definition of a metric eliminates the ambiguities which currently exist in natural language descriptions.

— **Comparability:** The structured nature of the metric facilitates the comparison of different metrics and SLOs/SQOs based on a metric.

— **Implementation:** The structured representation of the information needed to measure a characteristic facilitates the process of developing measurement tools. Likewise, if the metric is found not to be implementable, then the metric will need to be revised so that it can be implemented, and the structure of the technical specification is expected to facilitate this revision process.

The focus of this document is on metrics for cloud SLAs, but it is also usable for cloud service metrics (CSMs) that are not included in cloud SLAs [such as ones used by cloud service providers (CSPs) for their internal performance monitoring], and may also be usable for non-CSMs.

Cloud computing — Service level agreement (SLA) framework —

Part 2: Metric model

1 Scope

This document establishes common terminology, defines a model for specifying metrics for cloud SLAs, and includes applications of the model with examples. This document establishes a common terminology and approach for specifying metrics.

This document is for the benefit of and use for both cloud service providers (CSPs) and cloud service customers (CSCs). This document is intended to complement ISO/IEC 19086-1, ISO/IEC 19086-3 and ISO/IEC 19086-4.

This document does not mandate the use of a specific set of metrics for cloud SLAs.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 17788, | ITU-T Y.3500, *Information technology — Cloud computing — Overview and vocabulary*

ISO/IEC 19086-1, *Information technology — Cloud computing — Service level agreement (SLA) framework — Part 1: Overview and concepts*

W3C Recommendation 28 October 2004. *XML Schema Part 1: Structures Second Edition*. http://www.w3.org/TR/xmlschema-1/

W3C Recommendation 28 October 2004. *XML Schema Part 2: Datatypes Second Edition*. http://www.w3.org/TR/xmlschema-2/

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 17788 and the following definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

3.1
cloud service characteristic
qualitative or quantitative property of a cloud service

3.2
cloud service metric
metric (3.6) used to assess a *cloud service characteristic* (3.1)

3.3
cloud service objective
SO
commitment a cloud service provider (CSP) makes for a specific characteristic of a cloud service

Note 1 to entry: The set of SOs is the union of the set of cloud service level objectives (SLOs) and the set of cloud service qualitative objectives (SQOs).

3.4
measurement
set of operations having the objective of determining a *measurement result* (3.5)

Note 1 to entry: Based on the definition of measurement in ISO/IEC/IEEE 15939:2017. Also used here to describe an actual instance of execution of these operations leading to the production of a measurement result instance.

3.5
measurement result
value that expresses a qualitative or quantitative assessment of a *cloud service characteristic* (3.1)

3.6
metric
standard of measurement that defines the conditions and the rules for performing the *measurement* (3.4) and for understanding the *measurement result* (3.5)

Note 1 to entry: The metric describes what the result of the measurement means, but not how the measurement implements the metric.

Note 2 to entry: A metric is to be applied in practice within a given context that requires specific properties to be measured, at a given time(s) for a specific objective.

Note 3 to entry: The metrics model proposed in this document supports the definition of composite metrics, which can be defined in terms of one or more underlying (reusable) metrics.

[SOURCE: ISO/IEC 19086-1:2016, 3.10, modified — Notes to entry have been modified.]

3.7
unit
real scalar quantity, defined and adopted by convention, with which any other quantity of the same kind can be compared to express the ratio of the second quantity to the first one as a number

[SOURCE: ISO/IEC 80000-1:2009, modified — NOTES 1 to 5 have not been included.]

4 Symbols and abbreviated terms

CSA	Cloud Service Agreement
CSC	Cloud Service Customer
CSM	Cloud Service Metric
CSP	Cloud Service Provider
PII	Personally Identifiable Information
SLA	Service Level Agreement
SLO	Cloud Service Level Objective
SO	Cloud Service Objective
SQO	Cloud Service Qualitative Objective

5 Conformance

A metric specification is in conformance with this document when the specification of the metric uses the data-types and relationships described in Clause 8. If the XML namespace at http://standards.iso.org/iso-iec/19086/-2/ed-1/en is used for an XML document representing a metric, that document shall be valid per the schema in Annex D.

6 Metrics overview

6.1 General

This document describes metrics, metrics usage, and defines a model for the consistent specification of metrics for cloud SLAs. The underlying model and template is important for those defining metrics and implementing measurement systems based on specific metrics.

6.2 Background

Cloud computing is an integral part of IT, yet it is still difficult to define the properties and performance of a cloud service. A CSC purchases cloud services for one overarching reason – to achieve organization goals. To this end, it is important for a CSC to understand the properties of a cloud service, to understand the service capabilities for these properties, and to understand if it will meet the CSC's requirements necessary to achieve the CSC's organization goals. Likewise it is important to a CSP to be able to communicate the properties and performance of a cloud service so CSCs will be able to determine whether any given service meets their requirements.

Typical categories of cloud service characteristics that the CSC might be interested in include: performance, availability, information security, accessibility, cloud service support, termination of service, governance, service changes, service reliability, attestations/certifications, data management, and PII protection. A description of each along with associated SLOs/SQOs is included in ISO/IEC 19086-1 and ISO/IEC 19086-4.

The cloud procurement process may be split into three basic aspects:

a) selecting a cloud service that meets the CSC's requirements;

b) agreement (CSA, which includes the cloud SLA) between the CSP and the CSC on the properties and performance of the cloud service;

c) operational management, where the operation of the cloud service is monitored to ensure the service is operating within the constraints specified within the cloud SLA/CSA.

6.2.1 Choosing a cloud service

Currently, CSPs create their own methods to define the representation of a cloud service characteristic, and therefore influence the understanding of the characteristic itself. This makes comparing cloud service characteristics across CSPs (and sometimes within a CSP) difficult or impossible. The cloud service characteristics are often described using ambiguous text descriptions and are not only difficult to compare, but difficult to understand. This makes transferring a set of requirements into an agreement between the CSC and CSP difficult; resulting in an agreement that might not meet the needs of either party.

As described in ISO/IEC 19086-1, a commitment written into a cloud SLA takes the form of either an SLO or an SQO. SLOs are quantitative commitments for representations of cloud service characteristics, while SQOs are qualitative commitments for representations of cloud service characteristics.

A text description of availability might look something like:

— **Commitment**

The cloud service will be available 99,9 % of the time in a given billing cycle. If we fail to meet this commitment, you will be eligible to receive a credit to your account.

— **Unavailability**

Unavailability means: a) The cloud service provides no response, or b) the cloud service returns a server error response to a valid user request during two or more consecutive one minute intervals, or c) the cloud service fails to deliver an average download time for a reference document of one second or less as measured by a third party. Unavailability due to ***scheduled maintenance*** is excluded from these conditions and does not contribute towards unavailability calculations.

This is a good example to consider since it is fundamental to all cloud services (CSCs want the service to be available for their use). Although a number (percentage) is given for availability it is not clear how this number is calculated. A definition of unavailability is provided, but it is not clear how the availability percentage is calculated from unavailability. Another provider could use the concept of availability and calculate it differently, making comparison of an SO or measured service levels impossible.

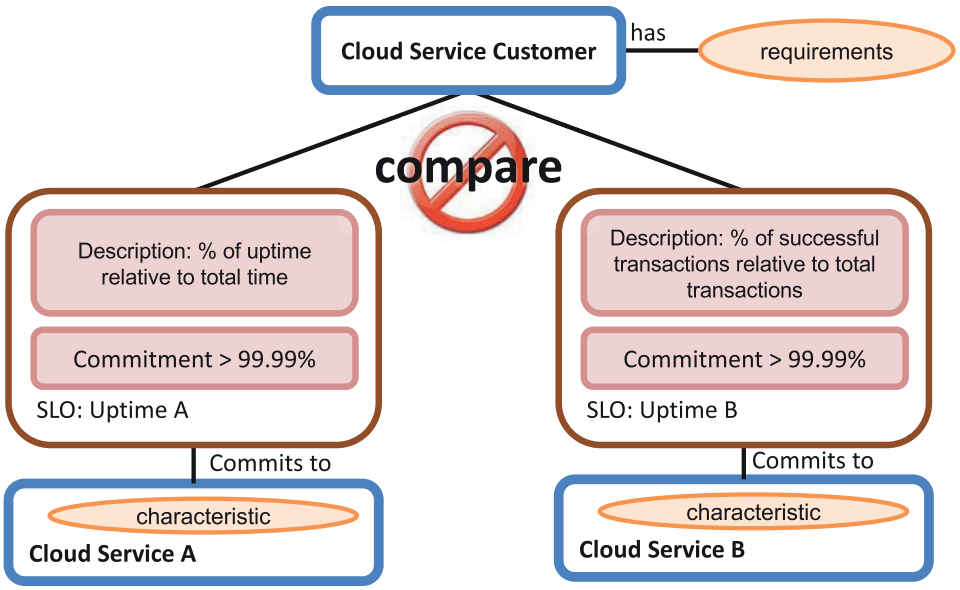

Figure 1 — Side by side comparison of availability commitment from two different providers

Figure 1 shows a CSC comparing availability for two services. While the characteristic (availability) and commitment level (99,99 %) seem identical; the text defining uptime for the two services is fundamentally different. Cloud service A uses a time based description for availability while cloud service B uses a transaction based description for availability. The CSC is therefore unable to evaluate how these two services differ.

6.2.2 Convert requirements to agreement

Once a CSC has chosen a service that meets their requirements, the CSP will work with the CSC to come to an agreement on what exactly will be provided and at what level. The requirements will be codified into a set of SLOs and SQOs written in a cloud SLA. As with the service capability descriptions used during the decision making process, the SLOs/SQOs are currently described using natural language and take different forms depending on the CSP. This not only adds ambiguity to the process, it adds additional time and complexity to the process as each SO must be thoroughly reviewed and assessed in each case.

6.2.3 Ensure the agreement is being met

Once an agreement is in place and the cloud service is provisioned for the CSC, the CSC users will start using the service or services. During operation, both the CSP and the CSC will monitor the services to ensure that it is operating as expected. The CSP will provide tools to measure the cloud service characteristics of the service and/or provide the data to the CSC. The CSC will compare these service level measurement results to the SO specified in the agreement. Due to the ambiguous and inconsistent nature of how SLOs/SQOs and metrics are currently described, it is difficult for the CSC to have confidence that these measurement results are calculated in the same manner as defined in the cloud SLA.

6.3 Metrics

A metric, or standard of measurement, helps to clarify cloud service characteristics by providing details about how a cloud service characteristic is measured. A metric provides a definition (e.g. expression, unit, rules, parameters) of the measurement of the characteristic, and therefore, provides knowledge about the characteristic itself. It provides the necessary information for repeatability, reproducibility, and comparability of measurements and measurement results.

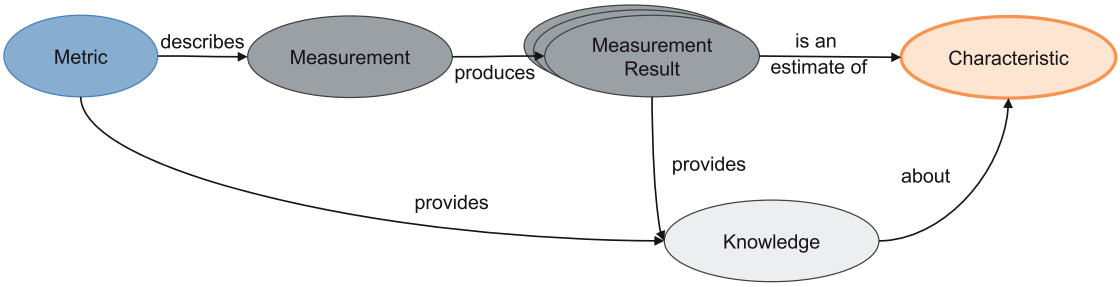

Figure 2 — Measurement process

The metric describes a cloud service characteristic and the details (parameters, data, rules, expressions, additional details) necessary to use it. For example, an "availability" metric will define the practical aspects of how to perform the measurement necessary to calculate availability, how to measure downtime, exclusion rules, etc. The measurement result is a value that results from making a measurement that follows a given metric. The measurement result is an estimate of a characteristic that is being observed. As shown in Figure 2, metric defines the rules so a measurement can be made in a repeatable, comparable manner. A measurement produces a measurement result that, combined with the information in the metric, provides knowledge about the characteristic. Characteristics of cloud services are almost never exactly known. Instead, an approximation of the characteristic is estimated (based on one or more measurements) with some understanding of the uncertainty between the approximation and the actual value of the characteristic. This uncertainty is directly tied to the measurement process used. In other words, a metric is a standard set of rules that allow a measurement process that follows the rules to generate repeatable, comparable estimates of a characteristic.

Since the metric provides both understanding of the characteristic and the information necessary to make repeatable and comparable measurements, it can be used not only for the measurement process, but in the conceptual understanding of a given characteristic.

What level of detail is included in a metric is up to the stakeholders. A metric may be only the basic equations needed to calculate a measurement result or it may include the detailed measurement process itself.

6.4 Cloud service metrics (CSMs)

Example scenarios for cloud services include the application of an availability metric for a SLO commitment of 99 % in a cloud SLA or the application of an accessibility metric for a service quality objective of "high" in a decision process scenario. In this manner CSMs help CSPs communicate the

properties of their cloud services, help CSCs and CSPs agree on what will be provided, and allow cloud service features to be measured to ensure the agreement is met.

Cloud services can use metrics in many different ways. Metrics can be used at different layers of a cloud service (e.g. hardware layers, logic layers, governance layers or service layers). They can also be used at different stages of the cloud service life cycle (e.g. selection, procurement, operation, audit and termination).

Although the metric model described in Clause 8 is designed for general use, in this document the focus is on metrics used in cloud SLAs.

The definition and usage of appropriate metrics with their underlying measures are essential components of the cloud SLA and the SLOs/SQOs, which are constituents of the cloud SLA. Within a cloud SLA, the metrics are used to set the boundaries and margins of errors the provider of the service commits to deliver, and their limitations. Standardized metrics and metric templates for cloud SLAs makes it easier and quicker to develop cloud SLAs and the included SLOs/SQOs. Once the cloud SLA is in place, metrics could be used at runtime to measure the services levels and determine if the service is meeting the commitments in the cloud SLA.

Metrics for cloud services have several uses for CSCs and CSPs including but not limited to:

— Metrics can be used by CSPs to describe the performance of a cloud service.

— Metrics help CSCs to compare offerings from different CSPs (when CSPs uses the same metrics to describe the performance of a cloud service).

— Metrics can be used to describe SLOs and SQOs within a cloud SLA.

— Metrics can be used by CSCs to determine if the CSP is meeting their commitments as described in the cloud SLA and to claim remedies if the commitments are not being met.

6.4.1 Major stakeholders

This subclause describes stakeholder types that have an interest in cloud SLA metrics. Each stakeholder type represents a different set of interests and related concerns. Each stakeholder type uses metrics in a different way to address those interests.

Attorney

Negotiates agreements for cloud services between CSP and CSC. Uses this document as reference for specifying metrics in cloud SLAs.

Auditors

Evaluate cloud services for performance against a set of requirements and commitments. Use ISO/IEC 19086-2 based metrics to define and measure cloud service characteristics that relate to the requirements. A specific type of auditor may also use this document to evaluate the measurement system itself. Needs to understand the metric used to measure a cloud service characteristic.

Certification authority

Develop certification criteria for cloud services. Use metrics based on this document to develop repeatable certification processes. Use ISO/IEC 19086-2 based metrics to describe auditing results to show compliance or non-compliance against the certification criteria.

CSC procurement

Works on the contract negotiation between the CSC and the CSP. Uses metrics based on this document to help review the service capabilities and compare one service to another. It is worth noting that the CSC procurement officers do not need a full understanding of the metric, rather an understanding of the basic description and the identifier for the metric is all that is required.

CSP operations

Monitors the covered services using metrics based on this document and ensures the service is operating according to SLO or SQO commitments in the cloud SLA. Needs to know the metric used for a given SO and needs monitoring to be done using the same metric.

CSP sales team

Assist CSC procurement experts in evaluating and acquiring cloud services. Use CSMs to help review the service capabilities and compare one service to another.

CSP service risk manager

Ensure the CSP service will meet certification and auditing (both internal and external) requirements and other controls. Use metrics based on this document to measure and compare cloud service characteristics.

CSP support and care representative

Evaluate claims that a given SO has not been met.

Development teams

Implement measurement systems based on metrics to monitor performance of the covered services and determine if the commitments in the CSA are being met.

Service integrator

Combines services to create new services. Determine performance of hybrid service by combining performance metrics based on this document from two or more services.

Third party monitors

Monitors the covered services using metrics based on this document. Needs to know the metric used for a given SLO or SQO and needs monitoring to be done using the same metric. Can be leveraged by the CSP, the CSC or other entities to perform measurement of metrics based on this document.

6.4.2 CSM usage categories

This subclause describes general usage categories for metrics based on this document.

6.4.2.1 Describing cloud services

Metrics can be used by the CSP to describe the characteristics of a cloud service in a standardized way. By using a standardized way (based on a metric) of representing the characteristics, CSCs can compare the characteristics of one service offering to another service offering.

CSPs should use the standard description model provided in this document in order to specify the metrics they offer in their cloud SLAs, abiding to the various necessary fields and the format of the information within each field and associate these metrics to SLOs or SQOs.

6.4.2.2 Comparing cloud services

While metric descriptions for a similar cloud service characteristic may vary between CSPs, metrics based on this document will help CSCs compare services from different CSPs. This document provides a model for expressing metrics consistently that will enable CSCs to understand the service characteristics for a given service and compare it to other services that use the same metrics to describe the service characteristics. In the case where the metrics used are different, using a consistent model to define metrics will help the CSC have a better understanding of how the defined characteristics differ.

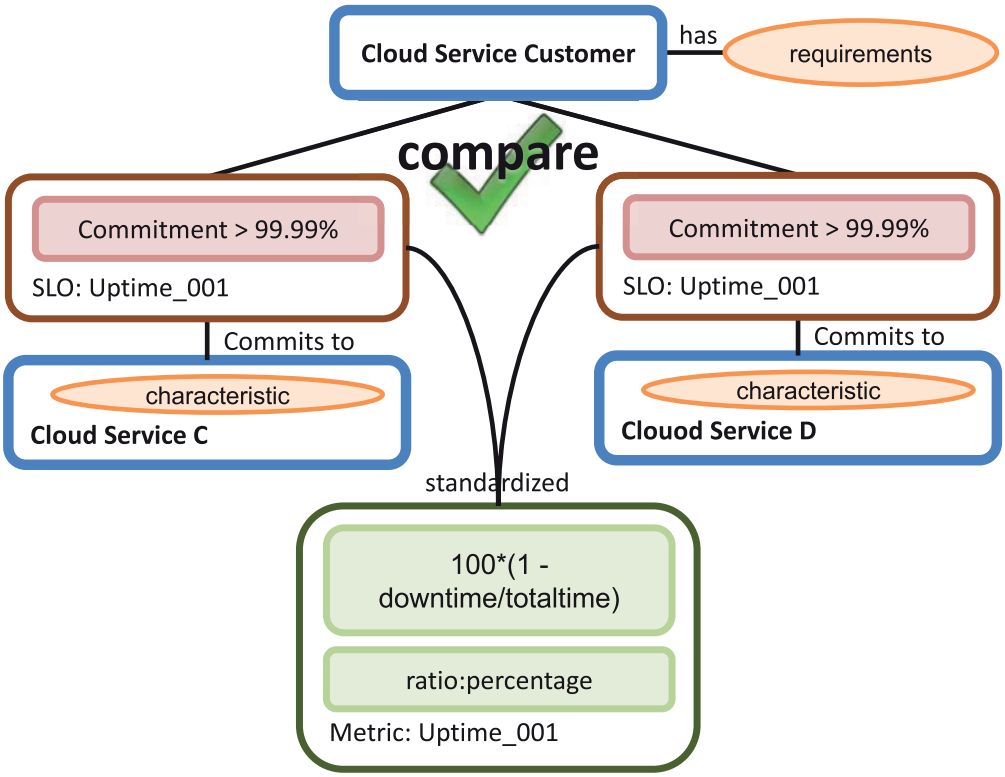

Figure 3 — CSC compares two services using the same metric

Since a CSM describes not only the requirements for measuring a characteristic but it also describes the characteristic itself, as shown in Figure 3, CSCs may use standardized metrics to understand and compare service offerings for different cloud services (note that not all the information for the metric is shown in the figure due to space constraints). As in Figure 1, Figure 3 shows a CSC comparing two services, but in this case the CSC is able to compare the two services because they use a standardized model to describe the same metric for uptime SLO and therefore, the CSC can determine the service that best meets its requirements.

CSCs or third parties (e.g. service brokers, consultants, service selectors on behalf of the CSCs) may use the standardized metric descriptions in order to compare different cloud SLAs and potentially rate them. Comparison between cloud SLAs could also be performed in an automated, tool-supported fashion.

The CSC can even define their requirements using the same metrics making it more likely that the chosen service will fit their requirements.

6.4.2.3 Understanding cloud service agreements (CSAs)

Once a CSC has chosen a cloud service based on analysing the service characteristics and their own requirements, this information may be included in a cloud SLA for the service. A cloud SLA has SOs for the characteristics of concern.

An SO provides a value understood to have a specific meaning, which relates a characteristic of a service to a specific level that should be met by the provider of the service. A metric for a characteristic provides a way of understanding what a given SO value means and gives guidance in how to make the measurement. A metric does not prescribe a specific method for making a measurement; rather it provides some rules on how to make the measurement.

By using SOs to describe the performance level for each characteristic based on metrics defined using this document both the CSC and the CSP will have a clear understanding on the performance of the cloud service and how to monitor the set of service properties.

6.4.2.4 Developing performance monitoring tools

This document defines a model for specifying metrics in a consistent way that will help development teams understand how to create tools for obtaining and interpreting measurement results based on the metric.

Metric templates and machine-readable description instances are key for tool developers to provide tools for automating some or all of the aforementioned processes (especially the cloud SLA instantiation, cloud SLA auditing and comparison usages).

CSCs (or third parties such as cloud SLA auditors) may use metric descriptions to develop runtime service monitoring mechanisms to detect violations of the agreement (in most cases CSCs monitor the covered services themselves and claim remedies when the CSP fails to meet their commitments).

6.4.2.5 Monitoring service performance

This document provides a model for expressing metrics in a consistent way that helps operations centres develop monitoring tools necessary to measure a cloud service characteristic and interpret measurement results (which represent the service level) against the SO commitments made in CSAs.

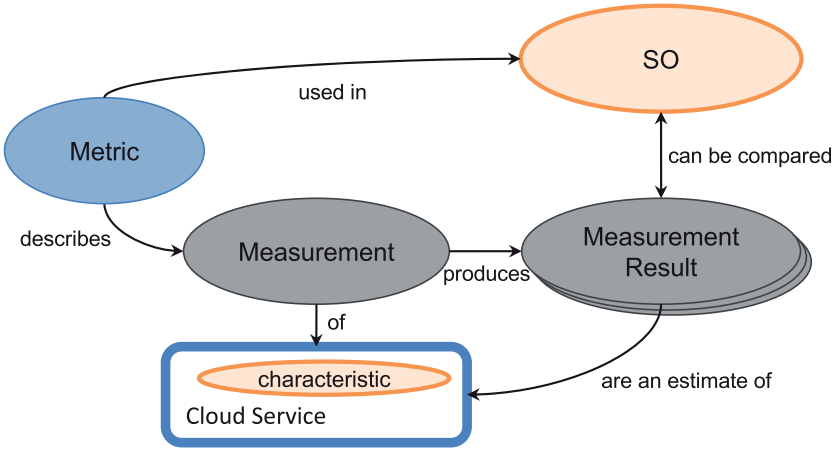

Figure 4 — Using metrics to verify the cloud SLA

6.4.2.6 Verifying the cloud SLA

A measurement produces a measurement result – a value estimating the characteristic. Since the measurement is based on the same metric that is used to describe the characteristic in the cloud SLA (via an SO included in the SLA) it is possible to compare the measurement result (representing the service level) to the SO commitment (shown in Figure 4). If the service level does not meet the SO, then the commitment is not met and a remedy, as described in the cloud SLA or CSA, will likely be sought as shown in Figure 5 below.

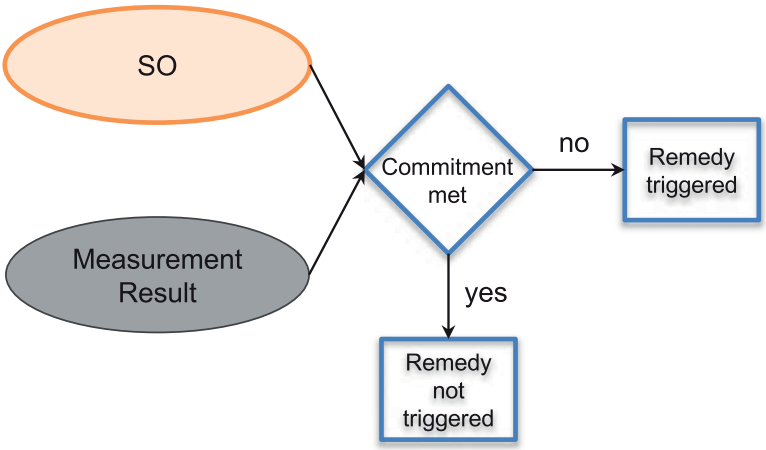

Figure 5 — Comparing a measurement result to a SLO to determine whether the commitment was met

6.4.2.7 Closing the loop

Ultimately, a CSC wants the cloud service to meet its requirements and therefore help to achieve the business goals. By tying all phases of this procurement process together through the use of a set of standardized metrics, it is easy for a CSC to verify the service is meeting those requirements and to obtain compensation in the case that it does not.

7 Metric model overview

7.1 General

The metric model provides a basis for metrics expression and communication to the potential stakeholders.

As identified in ISO/IEC 19086-1, “the definition of SLOs and SQOs is intended to be technology and business model neutral, so not all of these SLOs or SQOs will apply to every cloud service, and those that do apply may be structured and applied in different ways to specific cloud services”.

Thus, each SO should be mapped to one (or more) specific metric description that is instantiated based on the respective CSP’s definition. Even for SOs with the same label (e.g. Availability), not all CSPs follow the same definition or computation formula. Furthermore, based on the service type, availability may be defined with relation to performance aspects (e.g. response time to a set of requests), network connectivity aspects and/or with a given ratio of errors. As a conclusion, a defined metric should be capable of describing the way this SO is calculated, e.g. how to calculate the downtime, what intervals may be considered as eligible to be included, what are the limits of the errors.

A defined metric or formula based on this document should reference the SLO/SQO in ISO/IEC 19086-1 or ISO/IEC 19086-4 it refers to where applicable. A complex SO may be described using more than one metric.

Not all stakeholders need or want to access the same amount of information related to such metrics. Some will only need to understand measurement results and trust that others chose correct metrics. Some will need to understand how metrics are composed and specified to figure out which ones to choose. Some will need a way to describe metrics to develop new metrics based on measurable cloud service characteristics.

It is useful to note that although this metric model is designed to support cloud SLA metrics, it can be used to define any metrics.

7.2 Basic concepts

7.2.1 Introduction

A metric definition is supported by several attributes including several basic concepts. It needs information to identify the metric and a basic description of the metric. It needs information about what type of measurement scale is used (e.g. nominal, ordinal, interval, and ratio). It needs the expressions used in calculating a measurement result along with the definition for the variables and constants used in the expressions plus any rules or constraints that must be taken into account.

7.2.2 Cloud service level objectives and cloud service qualitative objectives

A cloud SLA includes a collection of SLOs and SQOs for which the CSP makes a commitment. SLOs are quantitative commitments for cloud service characteristics, while SQOs are qualitative commitments for cloud service characteristics.

The metric model in this document can be used to describe metrics used to describe both SLOs and SQOs. In the case of an SQO, however, there may be no expression or parameters as the SQO may be expressed as a set of rules describing a set of categories for the metric.

SLO example — Availability (see C.1 for this metric in ISO/IEC 19086-2 form)

Commitment

The cloud service will be available 99,9 % of the time in a given billing cycle. If we fail to meet this commitment, you will be eligible to receive a credit to your account.

Unavailability

Unavailability means: a) The cloud service provides no response, or b) the cloud service returns a server error response to a valid user request during two or more consecutive one minute intervals, or c) the cloud service fails to deliver an average download time for a reference document of one second or less as measured by a third party. Unavailability due to ***scheduled maintenance*** is excluded from these conditions and does not contribute towards unavailability calculations.

Definitions

Scheduled maintenance: maintenance scheduled at least five business days in advance, and that does not exceed one hour in any seven-day period.

SQO example — PII consent (see C.2 for this metric in ISO/IEC 19086-2 form)

Description

This SQO relates to a qualitative metric describing the type of consent obtained for collecting, using and sharing PII data. The type of consent can be ranked in levels according to its preference.

Formulation and output

— Level 0 – No Consent: Consent is not obtained at or before collection of private data.

— Level 1 – Implied Consent: The consent is inferred from the behaviour of the data subject, or even from failing to explicitly object. No opt-out or opt-in mechanisms are offered.

— Level 2 – Opt-out Consent: Data subjects can take measures to prevent the collection of private data, but no opt-in mechanisms are offered.

— Level 3 – Opt-in Consent: Data subjects explicitly grant permission for collecting or using private data.

7.2.3 Metric data format

The model described in Clause 8 is the conceptual basis for specifying metrics in a way which is reasonably consistent for all users, to reduce the ambiguity which can exist in metric definitions, to facilitate comparability between metrics, and potentially to facilitate the automation of metric specifications.

A detailed metric specification, i.e. the specification of a metric, could be expressed in XML, JSON or other serialization formats, as shown in the examples in Annex B. Two approaches to specifying metrics based on the model are described in Clause 8. These are:

— **Table-based approach.** Specifying metrics using the table-driven approach is described in Annex B. The examples provided in Annex C demonstrate how this can be done. Composite metrics could be specified this way, although the resulting tables will likewise be complex. Collections of supporting tables for underlying metrics and sub-expressions can be developed to facilitate the use of the table-driven approach.

— **Automated approach.** It should be possible to develop automated applications which can be used to generate detailed metric specifications directly in XML or JSON based on reasonably simple user interfaces and utilizing libraries of standard underlying metrics and sub-expressions. In the case of common metrics, user input could be limited to specifying key parameters such as required availability percentages. No examples are shown for this approach, since they would be highly dependent on how the applications would be written. As one of the benefits of cloud is the ability to quickly provision services, the process of choosing/agreeing/monitoring should be as quick as possible. Metrics based on a common and standardized model will assist in this goal.

8 Metric model

8.1 Metric model development

The core goals considered when developing the metric model include the following:

— **Consistent representation of information**

Information related to metrics should be represented in a consistent, repeatable way in order to efficiently organize it, share it and use it.

— **Explicit relationships**

Metrics should be represented in such a way that the relationships among different metrics, if any, are explicit. This clarifies the effects these concepts have on one another and their importance.

— **Repository of definitions**

There should be a way to organize metrics so they are reusable, searchable and derivable.

— **Comparability**

Metrics should contain enough information to allow users to efficiently compare them to find and understand the similarities and differences.

— **Adaptability**

The model should be sufficiently adaptable to allow for easy integration with other metric models. These models could be complementary to the model (e.g. represent measurement methods and process).

— **Composability**

Metrics should allow metric definitions to be reusable. Thus one should be able to use one or more metrics to build a composite metric. Metrics that are composed of underlying metrics build upon

the information the underlying metric contains. This results in a metric that could be composed of underlying metrics of different types (e.g. qualitative and quantitative). A key aspect of the metric model is to allow metric definitions to be composed using other metric definitions. This is an effort to limit the duplication of information without too much of an increase in complexity. The metric model allows metrics of different kind - qualitative or quantitative - to be defined, which in turns means that metrics of different types can potentially be combined with one another. This can affect the measurement of a particular characteristic in several ways including uncertainty, precision, accuracy etc.

8.1.1 Metric model specification

The metric model supporting ISO/IEC 19086-1 and ISO/IEC 19086-4 SLOs/SQOs is shown in Figure 6.

The model shown in Figure 7 represents concepts that are useful to describe and understand a metric. The model uses UML class diagram notation. Classes (shown as rectangles) represent the types of elements that may be found in an instance of a metric. Classes contain attributes that define the data attached to an element. Relationships between classes are called associations, and are shown as lines between rectangles. Associations with an empty diamond are aggregations that represent composition of reusable elements. Arrows at the end of associations show the direction of the association (e.g. from Metric to Rules). The label next to the association represents a particular role for the class.

8.1.2 Use of UML class diagrams and textual descriptions

This document uses both text and UML class diagrams to describe the metric model. Both are normative, and are intended to be complementary. However, if a conflict exists between what is specified in UML and what is specified in text, the text takes precedence. All associations shown in the diagrams are implied as having multiplicity which are defined in the tables.

NOTE The REGEX expression "^[a-zA-Z_][\w.-]*$" can be used to constrain ID values so they conform to the XML Schema defined in Annex D.

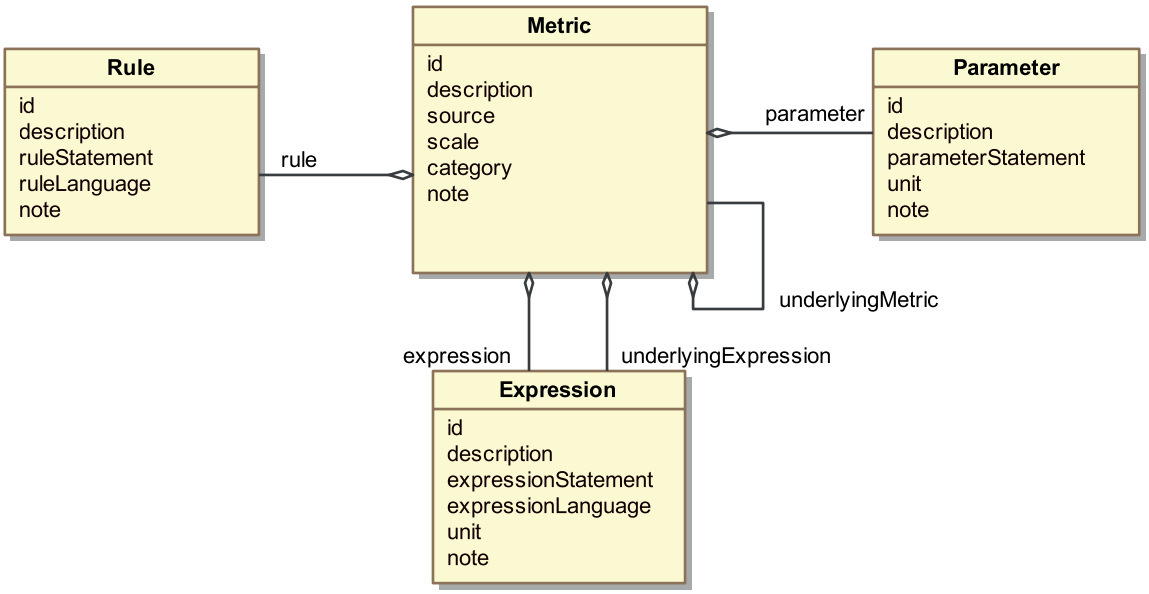

Figure 6 — UML representation of the metric model

8.1.3 Metric model description

The metric element contains the basic information related to the metric. This information provides context for the metric. A metric element may be composed of underlying metric elements that are written in the same form as the metric element.

The expression element contains the equations or textual description (e.g. expressed in natural language) needed to quantitatively or qualitatively calculate a measurement result based upon the metric.

Parameter elements contain values used in the expression. A parameter is evaluated before a measurement process and stays constant during the measurement process but can change between two measurements.

Rule elements provide constraints necessary for understanding of the characteristic and to ensure repeatability of measurements made using the metric.

The model shows composition relationships and is not explicit about the relationship between expression, rules, parameters, and underlying metrics. These are expressed through the rule, parameter, or expressions themselves. For instance, expressions can reference expression ids, metric ids, and parameter ids. Therefore within the context of an expression, each id shall be unique. All associations shown in the diagrams are implied as having multiplicity which are defined in the tables.

8.1.4 Extending the metric

The model can be extended as needed to support additional business requirements. Extensibility is supported in order to enhance the flexibility of the metric model, its descriptive ability across different use cases as well as to minimize the need for changes and updates in the main standard in order to address new advancements in CSMs.

Extensibility refers specifically to either the creation of new elements in the metric model or the extension of existing elements with new fields (a combination of both is also allowed). This is necessary in order to have a base model that covers most generic cases but is able to adapt to different types of metrics and give specific descriptions of features per case. The baseline model could not cater to all potential categories and types of metrics with the same set of generic fields. Examples are provided in C.3.

8.1.5 Metric model details

Each class from the model is described in detail in the following tables. These tables form the normative description of the model and provide more detail than shown in Figure 6.

This document uses both text and UML class diagrams to describe the metric model. Both are normative, and are intended to be complementary. However, if a conflict exists between what is specified in UML and what is specified in text, the text takes precedence.

Table 1 — Metric details

Class Name	Metric		
Description	Information about the metric		
attribute -or- Element	**Type**	**Multiplicity**	**Definition**
id	string	1	a unique identifier for the metric within a context
description	string	0..*	a description of the metric that may be provided for multiple languages
source	string	1	the individual or organization who created the metric
scale	enumeratedList	1	classification of the type of measurement result when using the metric. The value of scale shall be “nominal, ordinal, interval, or ratio”. SLOs shall use either the “interval” or “ratio” scale. SQOs shall use the “nominal” or “ordinal” scales.

Table 1 *(continued)*

Class Name	Metric		
Description	Information about the metric		
attribute -or- Element	**Type**	**Multiplicity**	**Definition**
note	string	0..*	additional information about the metric and how to use it
category	string	0..1	a grouping of metrics with similar expressions, rules, and parameters
expression	Expression	0..1	The expression of the calculation of the metric. An SLO metric shall have an expression while an SQO may or may not have an expression (e.g. specified using natural language). It shall be written using the appropriate id to represent an underlyingMetric, parameter, or rule. Expressions can reference other expression ids, metric ids, and parameter ids. Therefore within the context of a specific expression, Ids used within that expression shall be unique.
parameter	Parameter	0..*	A parameter is used to define a constant (at runtime) needed in a metric expression or rule. A parameter's value is determined before the metric is used in a measurement. A parameter may be used by more than one metric if it is defined using a unique id within the set of metrics it is used in.
rule	Rule	0..*	A rule is used to constrain a metric and indicate possible method(s) for measurement. Rules can reference other expression id, rule ids and parameter ids. Therefore within the context of a specific rule, Ids used within that rule shall be unique.
underlyingMetric	Metric	0..*	A metric element that is used within an expression to define a variable. The expression shall use the underlyingMetric id to reference the underlyingMetric within the expression.
underlyingExpression	Expression	0..*	A supporting expression that is used within an expression, parameter, or rule.

Table 2 — Expression details

Class Name	Expression		
Description	The expression of the calculation of the metric and supporting information		
attribute -or- Element	**Type**	**Multiplicity**	**Definition**
id	string	1	a unique identifier (within the context of the metric) for the expression
description	string	0..*	a description of the expression
expressionStatement	string	1	the expression statement written using the ids to represent underlyingMetrics, parameters, and rules Expressions can reference other expression ids, metric ids, and parameter ids. Therefore within the context of a specific expression, Ids used within that expression shall be unique.
expressionLanguage	string	1	the language used to express the metric (i.e. ISO 80000)
note	string	0..*	additional information about the expression
unit	string	0..1 required when scale is ratio or interval	real scalar quantity, defined and adopted by convention, with which any other quantity of the same kind can be compared to express the ratio of the two quantities as a number

Table 3 — Parameter details

Class Name	Parameter		
Description	A parameter is used to define a constant (at runtime) needed in a metric expression or rule. A parameter may be used by more than one metric if it is defined using a unique id within the set of metrics it is used in.		
attribute -or- Element	**Type**	**Multiplicity**	**Definition**
id	string	1	the unique identifier of the parameter
description	string	0..*	a description of the parameter
parameterStatement	string	1	the statement or value of the parameter
unit	string	1	the unit of the parameter
note	string	0..*	additional information about the parameter

Table 4 — Rule details

Class Name	Rule		
Description	A rule is used to constrain a metric and indicate possible method(s) for measurement. For instance an “AvailabilityDuringBusinessHour” metric could be defined with a scope that constrains some piece of a generic “availability” metric element that limits the measurement period to defined business hours. A rule describes constraints on the metric expression. A constraint can be expressed in many different formats (e.g. plain English, ISO 80000, SBVR).		
attribute -or- Element	**Type**	**Multiplicity**	**Definition**
id	string	1	the unique identifier for the rule
description	string	0..*	a description of the rule
ruleStatement	string	1	a constraint on the metric
ruleLanguage	string	1	the language used to express the rule in the ruleStatement
note	string	0..*	additional information about the rule

Annex A
(informative)

SO and SO evaluation

A.1 Model specification

The model supporting ISO/IEC 19086-1 and ISO/IEC 19086-4 SOs and SO evaluations is shown in Figure A.1. The full metric model is described in Clause 8.

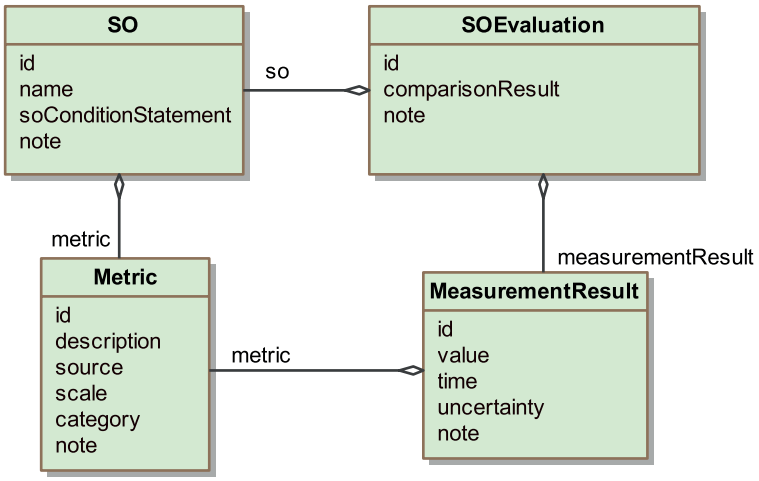

Figure A.1 — SO and SO Evaluation model

A.2 Model details

Each class is described in detail in the following tables.

Table A.1 — SO details

Class Name	SO		
Description	The SO associated with a specific metric		
attribute -or- Element	Type	Multiplicity/ Default	Definition
id	string	1	the unique identifier for the SO
name	string	0..1	The name of the SO as described in ISO/ IEC 19086-1
soConditionStatement	string	1	The condition statement that expresses the commitment for that SO using the associated metric
note	string	0..*	additional information about the SO
metric	Metric	1	the metric associated with the SO. The metric used in the SO

Table A.2 — MeasurementResult details

Class Name	MeasurementResult		
Description	A measurement result of a cloud service characteristic based on a metric		
attribute -or- Element	**Type**	**Multiplicity/ Default**	**Definition**
id	string	1	the unique identifier for the measurementResult
value	string	1	the value for the measurementResult
uncertainty	string	1	a statement describing the uncertainty of the measurementResult based on the specific measurement process used
time	string	1	the date and time the measurement took place
note	string	0..1	additional information about the measurementResult
metric	Metric	1	The metric used to generate measurementResultValue and measurementResultUncertainty. It should point to the same SO metric

Table A.3 — SO Evaluation details

Class Name	SOEvaluation		
Description	The evaluation of an SO against a measurement result when both the SO and the measurement result (quantitative or qualitative) are based on the same metric		
attribute -or- Element	**Type**	**Multiplicity/ Default**	**Definition**
id	string	1	the unique identifier for the SO evaluation
comparisonResult	boolean	1	the result of the comparison of the measurementResult to the SO commitment
note	string	0..1	additional information about the SO evaluation
so	SO	1	the SO associated with the SO evaluation. The SO evaluated against
measurementResult	MeasurementResult	1	the measurementResult associated with the SO evaluation.

Annex B
(informative)

Metric — Table form

B.1 General

While the model described in A.1 is useful for understanding the types information and relationships within a metric, it is difficult to use without a format for displaying the data. One way to format the data is using a set of tables that can be put into a document or stored in a relational database. This template (expressed using three tables) is based on the UML class diagram found in Figure 6 of 8.1. To use the template a metric table is first created, along with an expression table and any rule and parameter tables necessary. The template described provides the format necessary for creating a metric that can be used for describing an SO value. The details of each attribute are described above in the tables found in 8.1.5. If a document-based format is used for a metric, these tables should be used.

For instances where the attribute/relationship cardinality is >1, the multiple attribute/relationship values occur in adjacent rows with the same name in the first column.

B.1.1 Metric in table form

The metric table contains the basic details of the metric including the description, unique reference id and the units of the metric. There are pointers to tables containing the expression (required), parameters (optional), rules (optional), and any underlying metrics (optional). Optional rows may be left out of the instantiated tables.

The expression table contains the expression (equation or text) for quantitatively or qualitatively calculating the measurement described by the metric. The expression language and a note about the expression are also provided in this table.

The parameter table contains a parameter statement, unit and note that is used in the expression. There may be zero to many parameters associated with an expression or rule.

The rule table contains a rule or constraint to which a measurement conforms. There may be zero to many rules associated with a given metric.

B.2 Metric table

Metric (id)	
description	
source	
scale	
note	
category	
expressionStatement	
parameter	
rule	
underlyingMetric	
underlyingExpression	

B.2.1 Expression table

Expression (id)	
description	
expressionStatement	
expressionLanguage	
unit	
note	

B.2.2 Rule table

Rule (id)	
description	
ruleStatement	
ruleLanguage	
note	

B.2.3 Parameter table

Parameter (id)	
description	
parameterStatement	
unit	
note	

B.3 SO table form

The following table provides the information necessary to describe an SO.

SO (id)	
name	
soConditionStatement	
note	
metric	

B.4 SO evaluation table

The following table provides the information necessary to evaluate an SO against a measurement result when both the SO and the measurement result (quantitative or qualitative) are based on the same metric.

SOEvaluation (id)	
comparisonResult (true/false)	
note	
so	
measurementResult	

B.5 Measurement result table

MeasurementResult (id)	
value	
uncertainty	
time	
note	
metric	

Annex C
(informative)

CSM examples

C.1 Cloud service availability example: SLO using tables and XML

The following is an example of a metric for measuring cloud service availability (an SLO from ISO/IEC 19086-1). It is a simulated SLO based on real world examples. This example shows how SLOs are currently described.

Example availability SLO

— **Commitment**

The cloud service will be available 99,95 % of the time in a given billing cycle. If we fail to meet this commitment, you will be eligible to receive a credit to your account.

— **Unavailability**

Unavailability means: a) The cloud service provides no response, or b) the cloud service returns a server error response to a valid user request during two or more consecutive one minute intervals, or c) the cloud service fails to deliver an average download time for a reference document of one second or less as measured by a third party. Unavailability due to scheduled maintenance is excluded from these conditions and does not contribute towards unavailability calculations.

— **Definitions**

Scheduled maintenance: maintenance scheduled at least five business days in advance, and that does not exceed one hour in any seven-day period.

Analysis of SLO text

The SLO value is provided in the commitment section and given as available 99,95 % of the time. The time over which this "available" characteristic should be measured is also referenced in the commitment section as the "billing cycle", but the value for "billing cycle" is not provided. For the purpose of this example it is assumed that the "billing cycle" is thirty days. The description of unavailable (it is assumed that unavailable means the service is "not available") is described in the unavailability section. As described, there are three types of "unavailable" a) no response, b) server error to valid request, c) service response slow. The unavailability section also states an exception to the unavailability, if one of the three types of unavailability occurs, but maintenance has been scheduled, it does not count as unavailable.

This gives enough information to build a metric for availability. In the following subclauses the metric is described using the metric template tables from Annex 0 and using XML based on the XML schema from Annex A.

C.1.1 Cloud service availability SLO example

SO (SO_001)	
name	Cloud service availability
soConditionStatement	>99,95 %
metric	M_AVL_002

C.1.2 Description of availability example using tables

A cloud SLO consists of a description of the characteristic being committed to, the value for the SLO (the value the service provider has committed to), and a metric used to understand the meaning of that value. The metric is also used to make measurements of the characteristic.

Metric (M_AVL_002)	
description	CloudServiceAvailability
source	example
scale	RATIO
note	Based on real world example
expressionStatement	E_ 001
parameter	P_ 001
underlyingMetric	M_ TQD_001

Expression (E_ 001)	
expressionStatement	100 x (P_001 - M_TQD_001)/P_001
expressionLanguage	ISO 80000
unit	percentage

Parameter (P_001)	
parameterStatement	2,592 × 10^6
unit	second
note	Parameter represents a 30 day billing cycle in seconds

Metric (M_TQD_001)	
description	TotalQualifiedDownTime
source	example
scale	RATIO
underlyingMetric	M_QDT_001
expression	E_001

Expression (E_001)	
expressionStatement	Σ(M_QDT_001)
expressionLanguage	ISO 80000
unit	second

Metric (M_QDT_001)	
description	QualifiedDownTime
source	example
scale	RATIO
rule	R_001, R_002, R_003
expression	E_001

Expression (E_Q001)	
expressionStatement	Δ(t)
expressionLanguage	ISO 80000
unit	second

Rule (R_001)	
ruleStatement	the time duration starts when it is observed that the cloud service is unavailable and ends when the cloud service is observed to be not unavailable according to rule R_002, R_003, R_004
ruleLanguage	English

Rule (R_002)	
ruleStatement	the cloud service provides no response
ruleLanguage	English

Rule (R_003)	
ruleStatement	the cloud service returns a server error response to a valid user request during two or more consecutive one minute intervals
ruleLanguage	English

Rule (R_004)	
ruleStatement	the cloud service fails to deliver an average download time for a reference document of one second or less. Unavailability due to scheduled maintenance is excluded from these conditions and does not contribute towards unavailability calculations
ruleLanguage	English

C.1.3 SLO metric availability example expressed in XML

The following code represents the availability example described in C.1.1 using XML notation instead of tables to represent the metric.

```
<?xml version="1.0" encoding="UTF-8"?>
<Metrics xmlns="http://standards.iso.org/iso-iec/19086/-2/ed-1/en"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Metric description="CloudServiceAvailability" source="example" id="M_AVL_002"
```

```
scale="RATIO">
        <Parameter id="P_001" parameterStatement="2.592 x10^6" unit="second"
            note="Parameter represents a 30 day billing cycle in seconds"/>
        <Expression id = "E_001" expressionStatement=" 100 x (P_001- M_TQD_001)/P_001"
expressionLanguage="ISO80000" unit="percentage"/>
        <UnderlyingMetricRef refid="M_TQD_001"/>
    </Metric>
    <Metric description="TotalQualifiedDowntime" source="example" id="M_TQD_001"
scale="RATIO" >
        <Expression id="E_001" expressionStatement="&#x3A3;(M_QDT_001)"
expressionLanguage="ISO80000" unit="second"/>
        <UnderlyingMetricRef refid="M_QDT_001"/>
    </Metric>
    <Metric description="QualifiedDownTime" source="example" id="M_QDT_001" scale="RATIO" >
        <Rule id="R_001" ruleLanguage="English" xml:lang="en"
            ruleStatement="The time duration starts when it is observed that the cloud
service is unavailable and ends when the cloud service is observed to be not unavailable
according to rule R_002, R_003, R_004"/>
        <Rule id="R_002" ruleLanguage="English" xml:lang="en" ruleStatement="The cloud
service provides no response"/>
        <Rule id="R_003" ruleLanguage="English" xml:lang="en"
            ruleStatement="the cloud service returns a server error response to a valid
user request during two or more consecutive one minute intervals"/>
        <Rule id="R_004" ruleLanguage="English" xml:lang="en"
            ruleStatement="the cloud service fails to deliver an average download time for
a reference document of one second or less. Unavailability due to scheduled maintenance
is excluded from these conditions and does not contribute towards unavailability
calculations."/>
        <Expression id="E_001" expressionStatement="&#x394;(t)"
expressionLanguage="ISO80000" unit="second"/>
    </Metric>
</Metrics>
```

C.1.4 SLO evaluation of availability example

Once a CSA has been reached between the CSC and the CSP and the CSP has set up the appropriate accounts for the CSC, the CSC users may begin to use the service. For the example above the following are simulated results. Assume the qualified downtime for a billing period was 2 h. Converting hours to seconds the qualified downtime is 7 200 sec.

— For the underlying metric, M_QDT_001 = [4 000 s, 3 200 s] (there were two qualified downtime events)

— So the underlying metric is calculated to be M_TQD_001 = 7 200 s

— For the parameter P_001 = 2,592 × 10^6 (the billing period)

— The Metric expression is M_AVL_002 = 100*(P_001 - M_TQD_001)/P_001

— So for measurement result M_AVL_002 = 100*(2,592 × 10^6 - 7200)/2,592 × 10^6

— M_AVL_002 = 99,722 %

One way in which cloud SLA metrics will be used is in the evaluation of an SLO written into a cloud SLA. Figure 4 shows how the measurement result for M_AVL_002 is linked to the SLO value through a metric. Since both the measurement result and the SLO value are based on the same metric the values can be compared to determine if the SLO was met. Using the example above the example SO evaluation would be:

MeasurementResult (MR_001)	
value	99,722
uncertainty	Not stated
time	May 23rd, 2016 9:00pm edt
metric	M_AVL_002

SOEvaluation (SOE_001)	
comparisonResult (true/false)	FALSE
so	SO_001
measurementResult	MR_001

In the above example (SOE_001), the commitment was not met during this billing period.

C.2 PII consent SQO example

Example SQO

The following is an example of a qualitative privacy metric describing the type of consent obtained for collecting, using and sharing PII data (based on ISO/IEC ISO/IEC 19086-4, NIST 800-53v4, and GAPP). The type of PII consent can be ranked in levels according to its preference.

"**Commitment** - The CSC (i.e. data subject) must provide unambiguous consent to the CSP for collecting or using his PII data. Such consent is to be expressed by a statement or by a clear affirmative action. If the CSP fails to meet this commitment, then the CSC will be eligible to receive a credit to his account."

Analysis of SQO text

The PII consent is defined by the CSP in a qualitative manner, so that data subjects can provide their consent based on any of the following levels:

— Level 0 – No Consent: Consent is not obtained at or before collection of PII data.

— Level 1 – Implied Consent: The consent is inferred from the behaviour of the data subject, or even from failing to explicitly object. No opt-out or opt-in mechanisms are offered.

— Level 2 – Opt-out Consent: Data subjects can take measures for prevent the collection of private data, but no opt-in mechanisms are offered

— Level 3 – Opt-in Consent: Data subjects explicitly grant permission for collecting or using private data.

This gives enough information to build a metric for PII Consent. In the following subclauses the metric above is described using the metric template tables from Annex 0 and using XML based on the XML schema from Annex A.

C.2.1 PII consent SQO example

SO (SO_002)	
name	PII Consent
soConditionStatement	True if "3"
metric	M_TPC_001

C.2.2 PII consent SQO metric example expressed in tables

A cloud SQO consists of a description of the characteristic being committed to, the qualitative value for the SQO (the value the service provider has committed to), and a metric used to understand the meaning of that value. The metric is also used to make measurements of the characteristic.

Metric (M_TPC_001)	
description	TypePIIConsent
source	example
scale	ORDINAL

Metric (M_TPC_001)	
note	The type of consent obtained for collecting, using and sharing private data is ranked in levels.
expression	E_001
underlyingMetric	M_TCL_001, M_TCL_002, M_TCL_003, M_TCL_004

Expression (E_001)	
expressionStatement	the value of either M_TCL_001, M_TCL_002, M_TCL_003, or M_TCL_004, depending on the fulfilment of their respective rules
expressionLanguage	ENGLISH

Metric (M_TCL_001)	
description	TypePIIConsent_Level0
source	Example source
scale	ORDINAL
rule	R_001
expression	E_001

Expression (E_001)	
expressionStatement	0
expressionLanguage	ISO 80000

Rule (R_001)	
ruleStatement	No Consent: Consent is not obtained at or before collection of private data
ruleLanguage	English

Metric (M_TCL_002)	
description	TypePIIConsent_Level1
source	**example**
scale	ORDINAL
rule	R_002
expression	E_001

Expression (E_001)	
expressionStatement	1
expressionLanguage	ISO 80000

Rule (R_002)	
ruleStatement	Implied Consent: The consent is inferred from the behaviour of the data subject, or even from failing to explicitly object. No opt-out or opt-in mechanisms are offered
ruleLanguage	English

Metric (M_TCL_003)	
description	TypePIIConsent_Level2
source	example
scale	ORDINAL
rule	R_003
expression	E_003

Expression (E_003)	
expressionStatement	2
expressionLanguage	ISO 80000

Rule (R_003)	
ruleStatement	Opt-out Consent: Data subjects can take measures for prevent the collection of private data, but no opt-in mechanisms are offered.
ruleLanguage	English

Metric (M_TCL_004)	
description	TypePIIConsent_Level3
source	example
scale	ORDINAL
rule	R_004
expression	E_004

Expression (E_004)	
expressionStatement	3
expressionLanguage	ISO 80000

Rule (R_004)	
ruleStatement	Opt-in Consent: Data subjects explicitly grant permission for collecting or using private data
ruleLanguage	English

C.2.3 PII consent SQO metric example expressed in XML

The following code represents the PII Consent example described using XML notation instead of tables to represent the metric.

```
<?xml version="1.0" encoding="UTF-8"?>
<Metrics xmlns="http://standards.iso.org/iso-iec/19086/-2/ed-1/en">
    <Metric id="M_TPC_001" description="TypeOfPIIConsent" source="example" scale="ORDINAL"
note="The type of consent obtained for collecting, using and sharing private data is
ranked in levels">
        <Expression  expressionStatement="M_TPC_001 is equal to the value of either
M_TCL_001, M_TCL_002, M_TCL_003, or M_TCL_004, depending on the fulfillment of their
respective rules" expressionLanguage="English"/>
        <UnderlyingMetricRef refid="M_TCL_001"/>
        <UnderlyingMetricRef refid="M_TCL_002"/>
        <UnderlyingMetricRef refid="M_TCL_003"/>
        <UnderlyingMetricRef refid="M_TCL_004"/>
    </Metric>
    <Metric id="M_TCL_001" description="TypeOfPIIConsent_Level0" source="example"
scale="ORDINAL">
        <Expression id="E_TCL_001" expressionStatement="0" expressionLanguage="ISO80000"/>
        <Rule id="R_001" ruleLanguage="English" xml:lang="en"
            ruleStatement="No Consent: Consent is not obtained at or before collection of
private data"/>
    </Metric>
    <Metric id="M_TCL_002" description="TypeOfPIIConsent_Level1" source="example"
scale="ORDINAL">
        <Expression id="E_TCL_002" expressionStatement="1" expressionLanguage="ISO80000"/>
        <Rule id="R_002" ruleLanguage="English" xml:lang="en"
            ruleStatement="Implied Consent: The consent is infered from the behaviour
of the data subject, or even from failing to explicitly object. No opt-out or opt-in
mechanisms are offered"/>
    </Metric>
    <Metric id="M_TCL_003" description="TypeOfPIIConsent_Level2" source="example"
scale="ORDINAL">
        <Expression id="E_003" expression="2" expressionLanguage="ISO80000"/>
        <Rule id="R_003" ruleLanguage="English" xml:lang="en"
            ruleStatement="Opt-out Consent: Data subjects can take measures for prevent
the collection of private data, but no opt-in mechanisms are offered."/>
    </Metric>
    <Metric id="M_TCL_004" description="TypeOfPIIConsent_Level3" source="example"
scale="ORDINAL">
        <Expression id="E_004" expressionStatement="3" expressionLanguage="ISO80000"/>
        <Rule id="R_004" ruleLanguage="English" xml:lang="en"
            ruleStatement="Opt-in Consent: Data subjects explicitly grant permission for
collecting or using private data"/>
    </Metric>
</Metrics>
```

C.2.4 Measurement Result table for PII consent example

MeasurementResult (MR_002)	
value	3
uncertainty	Third Party Audit-based
time	May 25th, 2016 9:00pm edt
metric	M_TPC_001

C.2.5 SQO evaluation example

Once a CSA has been reached between the CSC and the CSP containing the SQO referenced in C.2, it is possible to evaluate the fulfilment of the agreed qualitative level, e.g. by assessing the CSP's privacy policy and practices.

SOEvaluation (SOE_002)	
comparisonResult (true/false)	TRUE
so	SO_002
measurementResult	MR_002

In the above example (SOE_002), the SLO commitment was met during this billing period.

C.3 Extensibility examples in JSON — "Sample" element definition

C.3.1 General

In many cases, when defining a metric (or underlying metric), it is desirable to also define additional details about how a measurement is performed.

C.3.2 Example 1

For instance, information regarding a potential sampling process is necessary, which may be applied in the model by creating an extension element Sample (used inside a metric definition). The following example is indicative of an actual Blob storage service available online, in which different commitments are given in the cloud SLA per type of operation. We initialize the Sample element for only one such operation but it can be replicated for the other cases as well. The following snippet from a metric definition provides additional detail for underlyingMetric M_HER_001 using a user-defined "Sample" element:

```
"expression": {
    "id": "E_001",
    "expressionStatement": "E_003/E_002",
    "expressionLanguage": "ISO80000",
    "underlyingExpression": [
        {
            "id": "E_002",
            "expressionStatement": "&#3A3;(SAMPLE_001
             belonging to P_001)",
            "expressionLanguage": "ISO80000",
            "note": "Number of samples within the
             boundary period"
        },
        {
            "id": "E_003",
            "expressionStatement":
            "&#3A3;(SAMPLE_001.value>P_002
                      belongingtoP_001)",
            "expressionLanguage": "ISO80000",
            "note": "Number of error samples within
             the boundary period"
        }
    ]
},
```

The new "samples" element looks like this:

```
"samples": [
 {
     "name":
     "STORAGEGETBLOCKLISTAPICALLresponsetime",
     "id": "SAMPLE_001",
     "scale": "interval",
```

```
                                    "value_limit": "P_002",
                                    "unit": "second",
                                    "protocol": "REST",
                                    "operation": "GetBlockList",
                                    "note":
                                    "examplesampletomeasuretheresponsetimeof
                                    theservice"
                                }
```

Here is the full JSON for the example:

```
{
    "description": "MAS Availaiblity",
    "id": "M_MAS_001",
    "source": "example",
    "scale": "NOMINAL",
    "expression": {
        "expressionStatement": "M_MUP_002 < P_002",
        "expressionLanguage": "ISO80000"
    },
    "parameter": [
        {
            "id": "P_002",
            "description": "availability_limit",
            "unit": "%",
            "parameterStatement": "99.9"
        }
    ],
    "underlyingMetric": [
        {
            "description": "Monthly Uptime Percentage",
            "id": "M_002",
            "source": "example",
            "scale": "RATIO",
            "expression": {
                "id": "E_002",
                "expressionStatement": "100 - M_AER_001",
                "expressionLanguage": "ISO80000",
                "unit": "%"
            },
            "underlyingMetric": [
                {
                    "description": "Average Error Rate",
                    "id": "M_AER_001",
                    "source": "example",
                    "unit": "%",
                    "scale": "RATIO",
                    "expression": {
                        "expressionStatement": "M_AER_001 = AVG(M_HER_001) AND M_HER_001
belonging to P_001",
                        "expressionLanguage": "ISO80000"
                    },
                    "parameter": [
                        {
                            "id": "P_001",
                            "description": "billing period",
                            "unit": "month",
                            "parameterStatement": "1"
                        }
                    ],
                    "underlyingMetric": [
                        {
                            "description": "Hourly Error Rate",
                            "id": "M_HER_001",
                            "source": "example",
                            "unit": "%",
                            "scale": "RATIO",
                            "expression": {
                                "id": "E_001",
                                "expressionStatement": "E_003/E_002",
                                "expressionLanguage": "ISO80000",
                                "underlyingExpression": [
```

```
                                    {
                                        "id": "E_002",
                                        "expressionStatement": "&#3A3;
                                        (SAMPLE_001 belonging to
P_001)",
                                        "expressionLanguage": "ISO80000",
                                        "note": "Number of samples within the
                                        boundary period"
                                    },
                                    {
                                        "id": "E_003",
                                        "expressionStatement":
                                        "&#3A3;(SAMPLE_001.value>P_002
                                                  belongingtoP_001)",
                                        "expressionLanguage": "ISO80000",
                                        "note": "Number of error samples within
                                        the boundary period"
                                    }
                                ]
                            },
                            "parameter": [
                                {
                                    "parameterStatement": "3600",
                                    "description": "boundary period",
                                    "unit": "second",
                                    "id": "P_001"
                                },
                                {
                                    "parameterStatement": "60",
                                    "unit": "second",
                                    "id": "P_002"
                                },
                                {
                                    "id": "P_003",
                                    "description": "billing period",
                                    "unit": "month",
                                    "parameterStatement": "1"
                                }
                            ],
                            "samples": [
                                {
                                    "name": "STORAGEGETBLOCKLISTAPICALLresponsetime",
                                    "id": "SAMPLE_001",
                                    "scale": "interval",
                                    "value_limit": "P_002",
                                    "unit": "second",
                                    "protocol": "REST",
                                    "operation": "GetBlockList",
                                    "note": "examplesampletomeasuretheresponsetimeof
                                    theservice"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "note": "example"
```


C.3.3 Example 2

In some cases, a metric value is determined as a specific outcome of a specific measurement process (e.g. latency for uploading of a specific file size or a more futuristic vCPU performance commitment on a given benchmark). However in this case, a more detailed definition should be given that indicates more features such as for example what benchmark was used (or e.g. what is the size of the file for testing etc.).

Since the Sample element is not defined in the model, the Sample class can be altered, thanks to the flexibility of the proposed model, to obtain the following form:

```
                "parameter": [
                      {
                          "id": "P_001",
                          "unit": "month",
                          "parameterStatement": "1"
                      },
                      {
                          "id": "P_004",
                          "parameterStatement": "small, default, large",
                          "scale": "ordinal"
                      },
                      {
                          "id": "P_005",
                          "unit": "perday",
                          "parameterStatement": "3"
                      }
                  ],
                  "samples": [
                      {
                          "description": "DaCapo Benchmark",
                          "id": "SAMPLE_001",
                          "scale": "interval",
                          "value": "Throughput",
                          "unit": "operations/sec",
                          "operation": "Avrora",
                          "workload_type": "P_004",
                          "workload_value": "default",
                          "frequency": "P_005",
                          "note": "example definition of a benchmark test"
                       ]
```

Through this extended case we are able to more accurately depict the specific test needed to validate a specific offering, including aspects of frequency, workload, type etc. thus having a fully defined measurement process.

Here is the full JSON for the example:

```
{
    "description": "Provider X vCore commitment for Micro VM Size Offering",
    "id": "M_MAS_001",
    "source": "example",
    "scale": "NOMINAL",
    "expression": {
        "id": "E_001",
        "expressionStatement": "M_STD_001 < P_002 & M_AVG_001 <> P_003",
        "expressionLanguage": "ISO80000"
    },
    "parameter": [
        {
            "id": "P_002",
            "unit": "%",
            "parameterStatement": "10"
        },
        {
            "id": "P_003",
            "unit": "operations per second",
            "parameterStatement": "X"
        }
    ],
    "underlyingMetric": [
        {
            "name": "Average Standard Deviation of Benchmarked Values as % of mean value",
            "id": "M_STD_001",
            "source": "example",
            "unit": "%",
            "scale": "RATIO",
            "expression": {
                "expressionStatement": "100*average
                [(abs(SAMPLE_001-M_AVB_001)/M_AVB_001]",
```

```
            "expressionLanguage": "ISO80000"
        },
        "parameter": [],
        "underlyingMetric": [
            {
                "description": "Average Value of Benchmark Execution",
                "id": "M_AVB_001",
                "unit": "",
                "scale": "interval",
                "expression": {
                    "id": "E_001",
                    "expressionStatement": "average(SAMPLE_001) belonging in P_001",
                    "expressionLanguage": "ISO80000"
                },
                "parameter": [
                    {
                        "id": "P_001",
                        "unit": "month",
                        "parameterStatement": "1"
                    },
                    {
                        "id": "P_004",
                        "parameterStatement": "small, default, large",
                        "scale": "ordinal"
                    },
                    {
                        "id": "P_005",
                        "unit": "perday",
                        "parameterStatement": "3"
                    }
                ],
                "samples": [
                    {
                        "description": "DaCapo Benchmark",
                        "id": "SAMPLE_001",
                        "scale": "interval",
                        "value": "Throughput",
                        "unit": "operations/sec",
                        "operation": "Avrora",
                        "workload_type": "P_004",
                        "workload_value": "default",
                        "frequency": "P_005",
                        "note": "example definition of a benchmark test"
                    }
                ]
            }
        ]
    },
    "note": "this is an example metric in JSON form"
}
```

Annex D
(normative)

XML schema

XML representations of a metric shall conform to both Clause 8 and the following XML Schema document associated with the http://standards.iso.org/iso-iec/19086/-2/ed-1/en" name-space. This XML Schema is defined as follows (based on WC3 XML schema Parts 1 and 2):

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://standards.iso.org/iso-iec/19086/-2/ed-1/en"
    xmlns="http://standards.iso.org/iso-iec/19086/-2/ed-1/en"
    xmlns:tns="http://standards.iso.org/iso-iec/19086/-2/ed-1/en"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="https://www.w3.org/2001/xml.xsd"/>
    <!--reads local copy of xml.xsd -->
    <!-- In anticipation of possible defects which will be identified after
        publication, an alternative download location always with the latest
        defect-corrected version is:
        http://standards.iso.org/iso-iec/19086/-2/ed-1/en
    -->
    <!-- ISO/IEC 19086-2 contains further explanations about the Metrics, how they
        should be used, and recommended values for use with it.  References in this
        XSD are to clauses from that standard. -->
    <xs:annotation>
        <xs:documentation>
            Schema for ISO-IEC 19086-2 Service Level Agreement (SLA) Framework — Metric
Model
            http://standards.iso.org/iso-iec/19086/-2/ed-1/en

        </xs:documentation>
        <xs:appinfo>
            <version>1.0</version>
        </xs:appinfo>
    </xs:annotation>

    <!-- elements -->
    <xs:complexType name="tBase">
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="tReference">
        <xs:complexContent>
            <xs:extension base="tBase">
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                    <xs:any namespace="##other" processContents="lax"/>
                </xs:choice>
                <xs:attribute name="refid" type="xs:anyURI" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tMetricWithID">
        <xs:complexContent>
            <xs:extension base="tBase">
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                    <xs:element name="Expression" type="tExpression"/>
                    <xs:element name="ExpressionRef" type="tReference"/>
                    <xs:element name="Parameter" type="tParameter"/>
                    <xs:element name="ParameterRef" type="tReference"/>
                    <xs:element name="Rule" type="tRule"/>
                    <xs:element name="RuleRef" type="tReference"/>
                    <xs:element name="UnderlyingMetric" type="tMetricWithID"/>
```

```
                <xs:element name="UnderlyingMetricRef" type="tReference"/>
                <xs:element name="UnderlyingExpression" type="tExpression"/>
                <xs:element name="UnderlyingExpressionRef" type="tReference"/>
                <xs:any namespace="##other" processContents="lax"/>
            </xs:choice>
            <xs:attribute ref="xml:id" use="required"/>
            <xs:attribute name="description" type="xs:string"/>
            <xs:attribute name="source" type="xs:string" use="required"/>
            <xs:attribute name="scale" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:sring">
                        <xs:enumeration value="NOMINAL"/>
                        <xs:enumeration value="ORDINAL"/>
                        <xs:enumeration value="INTERVAL"/>
                        <xs:enumeration value="RATIO"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="category" type="xs:string"/>
            <xs:attribute name="note" type="xs:string"/>
            <xs:attribute ref="xml:lang"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tExpression">
    <xs:complexContent>
        <xs:extension base="tBase">
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="SubExpression" type="tExpression"/>
                <xs:element name="SubExpressionRef" type="tReference"/>
                <xs:any namespace="##other" processContents="lax"/>
            </xs:choice>
            <xs:attribute ref="xml:id" use="optional"/>
            <xs:attribute name="description" type="xs:string"/>
            <xs:attribute name="expressionStatement" type="xs:string" use="required"/>
            <xs:attribute name="expressionLanguage" type="xs:string" use="required"/>
            <xs:attribute name="unit" type="xs:string"/>
            <xs:attribute name="note" type="xs:string"/>
            <xs:attribute ref="xml:lang"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tExpressionWithID">
    <xs:complexContent>
        <xs:restriction base="tExpression">
            <xs:attribute ref="xml:id" use="required"/>
        </xs:restriction>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tParameter">
    <xs:complexContent>
        <xs:extension base="tBase">
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:any namespace="##other" processContents="lax"/>
            </xs:choice>
            <xs:attribute ref="xml:id" use="optional"/>
            <xs:attribute name="description" type="xs:string"/>
            <xs:attribute name="parameterStatement" type="xs:string" use="required"/>
            <xs:attribute name="unit" type="xs:string" use="required"/>
            <xs:attribute name="note" type="xs:string"/>
            <xs:attribute ref="xml:lang"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tParameterWithID">
    <xs:complexContent>
        <xs:restriction base="tParameter">
            <xs:attribute ref="xml:id" use="required"/>
        </xs:restriction>
    </xs:complexContent>
</xs:complexType>
```

```
    <xs:complexType name="tRule">
        <xs:complexContent>
            <xs:extension base="tBase">
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                    <xs:any namespace="##other" processContents="lax"/>
                </xs:choice>
                <xs:attribute ref="xml:id" use="optional"/>
                <xs:attribute name="description" type="xs:string"/>
                <xs:attribute name="ruleStatement" type="xs:string" use="required"/>
                <xs:attribute name="ruleLanguage" type="xs:string" use="required"/>
                <xs:attribute name="note" type="xs:string"/>
                <xs:attribute ref="xml:lang"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tRuleWithID">
        <xs:complexContent>
            <xs:restriction base="tRule">
                <xs:attribute ref="xml:id" use="required"/>
            </xs:restriction>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="Metrics">
        <xs:complexType>
            <xs:complexContent>
                <xs:extension base="tBase">
                    <xs:choice minOccurs="0" maxOccurs="unbounded">
                        <xs:element name="Metric" type="tMetricWithID"/>
                        <xs:element name="Expression" type="tExpressionWithID"/>
                        <xs:element name="Parameter" type="tParameterWithID"/>
                        <xs:element name="Rule" type="tRuleWithID"/>
                        <xs:any namespace="##other" processContents="lax"/>
                    </xs:choice>
                    <xs:attribute ref="xml:lang"/>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <!-- Uniqueness constraints not necessary when using xs:ID xs:IDREF hence removed
the elements xs:unique-->
    </xs:element>
</xs:schema>
```

Bibliography

[1] ISO/IEC/IEEE 15939:2017, *Systems and software engineering — Measurement process*

[2] ISO/IEC 17789:2014 | ITU-T Y.3502, *Information technology — Cloud computing — Reference architecture*

[3] ISO/IEC 19086-3, *Information technology — Cloud computing — Service level agreement (SLA) framework — Part 3: Core conformance requirements*

[4] ISO/IEC 19086-4, *CD Information Technology — Cloud Computing — Service Level Agreement (SLA) Framework – Part 4: Security and Privacy*

[5] ISO/IEC 19501, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

[6] ISO 80000-1, *Quantities and units*

[7] NIST 800-53v4, *Force, J. T., & Initiative, T. (2013). Security and privacy controls for federal information systems and organizations. NIST Special Publication, 800(53), 8-13*

ICS 35.210
Price based on 39 pages