

Untangling the Cloud

A Taxonomy for SLAs in
Federated Service Ecosystems



Version 1.0 (November 24th, 2025)

ISBN: 978-3-9828074-1-6

Published by

eco – Association of the Internet Industry (eco – Verband der Internetwirtschaft e.V.) Lichtstrasse 43h, 50825 Cologne, Germany

Copyright © eco Association on behalf of FACIS - funded by the German Federal Ministry for Economic Affairs and Energy (IPCEI-CIS)

Image Source: Adobe Stock

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA



**Comissioned author:
Technical University of Munich**

Arcisstraße 21,
80333 Munich
Germany

Point of contact: Yannick Heß & Prof. Dr. Sebastian Lins

E-mail: yannick.hess@tum.de

Website: <https://www.tum.de/>

Peer reviewed by the following persons in October 2025:

- Andreas Weiss, eco – Association of the Internet Industry
- Thomas Niessen
- Oliver Sümer, Fieldfisher Partnerschaft von Rechtsanwälten mbB
- Melanie Ludolph, Fieldfisher Partnerschaft von Rechtsanwälten mbB
- Dr. Bahne Sievers, Fieldfisher Partnerschaft von Rechtsanwälten mbB



Table of Contents



1. Introduction: Project FACIS - Shaping SLA Governance for Multi-Provider Clouds	1
2. Context and Prevalent Challenges	2
3. SLA Taxonomy	3
3.1 Overview of SLA Taxonomy	4
3.2 Foundation: Setting the Stage for the SLA	9
3.2.1 Definition of Terms	10
3.2.2 Covered Services and Their Interdependencies	10
3.2.3 Roles of Involved Parties and Their Responsibilities	11
3.2.4 General SLA Restrictions	14
3.2.5 SLA Validity Period and Change Management	15
3.2.6 Contractual and Statutory Law Context of an SLA	15
Concluding FACIS' Recommendations Related to SLA Foundations	16
3.3 Service Level Objectives: Determining Commitments about Service Levels	17
3.3.1 Core Classes of Service Level Objectives	20
3.3.2 Service Level Indicators	23
3.3.3 Commitment Framing	25
3.3.4 Classifying the Applicability of Service Level Objectives and Indicators: Commitment Levels, Operation Context, and Business Impact	26
Concluding FACIS' Recommendations Related to SLA Foundations	27
3.4 Monitoring: Measuring Compliance with Commitments	27
3.4.1 Metric to Measure Adherence to Commitments	28
3.4.2 Properties of Metrics	32
Concluding FACIS' Recommendations Related to Monitoring	34
3.5 Enforcement: Claiming Compensation in Case of (Non-)Adherence to Commitments	36
3.5.1 Service Level Objective Compliance Reporting	37
3.5.2 Remedy Claims	39
3.5.3 Decision to Award Remedies	41
Concluding FACIS' Recommendations Related to Enforcement	41
3.6 Automation: Fostering Machine-Readability of SLA	42
3.6.1 SLA Languages	43
3.6.2 Assessing SLA Languages: Formalization, Automation, Maturity, Generalizability, and Tool Support	44
Concluding FACIS' Recommendations Related to Automation	45
4. Conclusion	46
Appendix: Example Availability Commitments	47
Glossary of Terms	48
References	49

List of Figures

Figure 1 – Illustration of (1) entangled supply chains and (2) federated cloud ecosystems.....	2
Figure 2 – Illustration of the relationship between SLO, SLI, and Metric.....	17
Figure 3 – Side-by-side comparison of the availability of SLO from two different providers that use different metrics to measure uptime.	18
Figure 4 – Illustration of the measurement process.....	28
Figure 5 – Additional information for metrics.....	32
Figure 6 – Summary of enforcement process and related dimensions.....	36

List of Tables

Table 1 – Summary of dimensions for SLA foundation.....	9
Table 2 – Overview of characteristics assigned to the Interdependency of Services dimension.....	10
Table 3 – Common roles in cloud computing (adapted from Liu et al., 2011).....	12
Table 4 – Example responsibilities of providers.....	12
Table 5 – Example responsibilities of customers.....	13
Table 6 – Example restrictions contained in SLAs.....	14
Table 7 – Summary of dimensions for SLO.....	18
Table 8 – Overview of characteristics assigned to the SLO dimension: Core classes of SLO.....	20
Table 9 – Overview of characteristics assigned to the Commitment Framing dimension.....	25
Table 10 – Summary of dimensions for monitoring.....	28
Table 11 – Characteristics of the metric dimension, adapted from ISO/IEC 19086-2 (2016b).....	30
Table 12 – Example for measurements classified according to boundaries, timing, and invasiveness.....	33
Table 13 – Summary of dimensions for enforcement.....	37
Table 14 – Summary of dimensions for automation.....	42

Introduction: Project FACIS - Shaping SLA Governance for Multi-Provider Clouds



FACIS (Federation Architecture for Composed Infrastructure Services) is designed to enhance the establishment of cooperative data ecosystems, particularly for infrastructure-related service offerings operating in networked cloud-edge environments. FACIS deals with the development of an SLA Governance Framework for multi-provider environments, which provides the rules and mechanisms for seamless cooperation among service providers. Cloud SLA Governance covers issues related to cloud SLA design, evaluation, negotiation and acceptance, implementation and execution, and changes to the cloud SLA.

This document focuses on one of the key deliverables of the FACIS project: **A taxonomy of cloud SLA**. In general, a taxonomy is a structured classification system developed through an iterative and rigorous process (Nickerson et al., 2013). Taxonomies organize and structure knowledge by identifying dimensions that capture the shared and abstract characteristics of the objects being classified - in this case, SLAs. Dimensions capture the building blocks, key elements and contents, and important properties of SLAs in a standardized manner. Each dimension can include one or more characteristics to describe specific aspects of an object. For example, since SLAs typically define service level commitments as building block, these can form a dimension of the taxonomy. Such commitments might relate to availability, IT security, or other commitments that reflect the specific and unique characteristics of this dimension.

Taxonomies support the classification of existing SLAs from different providers: A specific cloud SLA can be described and classified by selecting fitting dimensions and corresponding characteristics. This structured approach enables and eases the comparison of SLAs because their specific features can be directly compared along the same dimensions and characteristics. Taxonomies also aggregate and synthesize knowledge in the field and thereby provide a starting point for future development and research initiatives. For instance, a taxonomy can inform the development of SLA ontologies, which are formal representations of a domain's concepts, relationships, properties, and rules, and are particularly relevant for the automation of SLA management. Among other things, a taxonomy can help to identify and detail concepts (e.g., covered cloud services, service level commitments), their relationships (e.g., service level indicators clarify how to measure those commitments), and properties (e.g., each commitment is bound to a specific time period). Nevertheless, taxonomies have limitations because they are mostly descriptive and written in natural language. With this taxonomy, we do not focus on the complex relationships between dimensions but rather discuss SLA-related challenges and specifics of multi-provider contexts.



Context and Prevalent Challenges

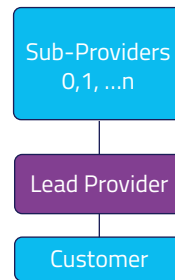


We consider SLAs as essential elements of every IT and cloud service contract that define quality levels, metrics, limits, roles, and responsibilities between providers and customers, among other things. SLAs vary between cloud providers, and in some cases, different customers can negotiate varying contract terms with the same cloud provider for the same cloud service. For the FACIS SLA governance framework in multi-provider contexts, we particularly consider two service provisioning scenarios: (1) an entangled supply chain involving multiple sub-providers, and (2) federated cloud environments and ecosystems (Figure 1). FACIS focuses on federated ecosystem scenarios since entangled supply chains represent the current status quo of most cloud services.

Today, cloud services typically build on an entangled supply chain, comprising software, platform, and infrastructure services (Floerecke et al., 2021). In this three-layered model of cloud computing, each service provider has specific responsibilities (F. Liu et al., 2011). In practice, cloud service providers will rely on multiple sub-providers to operate their services, often leading to opaque and lengthy supply chains. For example, a provider may offer a cloud-based software service that, in turn, harnesses specific operating platforms from another sub-provider, which then again run on virtual machines that are managed by an infrastructure provider. Multiple cloud providers are thus tightly integrated to enable service provisioning, and their services are deeply intertwined. However, in most cases, the customer will interact with the leading cloud service provider that will be then responsible for managing all involved sub-providers. An entangled supply chain can lead to many problems, such as limiting customers' oversight over sub-providers, unclear or changing data processing locations, and the risk of blame chains (e.g., cloud providers pointing to sub-providers in case of failure and refusing to take any liability).

In contrast, in a federated cloud ecosystem, multiple cloud providers collaborate and interoperate with each other to provide a seamless service experience for customers. Each provider maintains its own infrastructure and services, but they work together to share resources, data, and services with each other based on the customer's requests. Thus, providers often offer joint services. The customers may decide which service providers will be involved and take over certain service tasks. Federated cloud ecosystems thus allow customers to access a broader range of services and resources across multiple providers. In most cases, the customer will have individual agreements with each provider involved.

(1) Lead Service Provider



(2) Federated Cloud Service

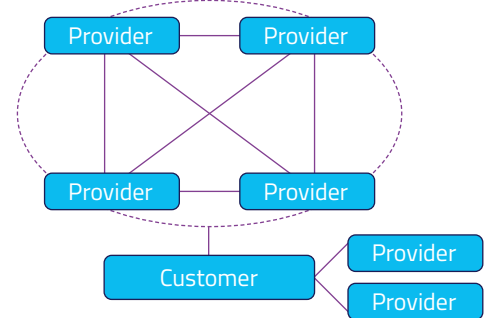


Figure 1 – Illustration of (1) entangled supply chains and (2) federated cloud ecosystems.

We seek to address the complexities of SLA management in those scenarios and consider them in the taxonomy, such as:

- **Cascading SLAs and end-to-end management of SLAs.** Defining and allocating responsibility between the different providers involved in service delivery is challenging, such as dealing with the risk of blame chains due to shared responsibilities across providers.
- **Addressing interdependencies and differences between SLA definitions, commitments, and metrics across different providers.** Each cloud provider makes different commitments about fulfilling specific service levels. Today, ambiguities exist in SLA because natural language is used to describe service levels and corresponding commitments. In addition, providers typically use different metrics to assess compliance with their commitments, hampering the comparison of different SLAs.
- **End-to-end SLA monitoring and compliance reporting.** Most providers already have sophisticated monitoring systems and corresponding organizational processes in place. However, multi-provider contexts require end-to-end monitoring; therefore, monitoring practices should be aligned and integrated across the cloud service provisioning stack.
- **Automation of SLA management.** To ensure economic feasibility, automation of SLA management practices is required, in addition to standardization and mutual agreement.

SLA Taxonomy



We iteratively developed a taxonomy to mitigate those issues and foster multi-provider service provisioning and the emergence of cloud ecosystems. The taxonomy captures important dimensions and corresponding characteristics that are grouped into four key building blocks of SLAs:

1. **Foundation:** Setting the stage for the SLA.
2. **Service Level Objectives:** Determining commitments about service levels.
3. **Monitoring:** Measuring compliance with commitments.
4. **Enforcement:** Claiming compensation in case of (non-) adherence to commitments.

Next, the taxonomy will be summarized and then each building block and its dimensions will be discussed in more detail.

Besides those SLA building blocks that jointly create an SLA blueprint, cloud providers can think about the **automation of SLA management through fostering machine-readability of SLA**.

Notably, each SLA may focus on the four core building blocks but should remain modular and may contain additional appendices needed to consider the unique specifics of a service and its operating context. For example, the base SLA sets out common definitions in the foundation and provides information on service level commitments, measurement rules, reporting, and remedies that apply to every service. Service annexes then add parameters, methods, and thresholds for specific layers (IaaS, PaaS, SaaS). Where sector rules or local law require more, a Sector Annex or Jurisdictional Annex may extend the Base SLA without rewriting it. This structure keeps day-to-day operations consistent while allowing precise tailoring where risk or regulation differ.



3.1 Overview of SLA Taxonomy

Dimension	Description of Dimension	Key Characteristics	Considerations in Multi-Provider Contexts
(1) Foundation: Setting the Stage for the SLA			
Definition of Terms	A section that determines each term and provides definition for clarity.	Term definitions, e.g., unavailability, maintenance window, force majeure	<ul style="list-style-type: none"> Reduce the risk of misinterpretation. Bridge technical and legal perspectives on SLAs.
Covered Services	Detailed definition and description of the services (NOT) covered by the SLA.	Name, functionality, non-functional attributes, pricing model	<ul style="list-style-type: none"> Depict the interplay of multiple services involved.
Interdependency of Services	Definition of the service's dependencies on upstream or downstream services and their sub-providers.	Individual service SLA, cross-service SLA (intra-provider), cross-provider SLA (inter-provider)	<ul style="list-style-type: none"> SLA must address how the performance of one service and its provider affects the overall service offering. Describe the connection between services involved. Position each involved service within the cloud service stack.
Roles of Involved Parties	Description of the roles of the parties involved.	(Lead) service provider, sub-provider, customer, end user, cloud broker, cloud carrier, auditor	<ul style="list-style-type: none"> Depict all involved parties and determine their role. Determine a lead service provider that acts as SLA integrator.
Provider	Detailed description of the provider and their interfaces involved in the SLA.	Provider details, action interfaces, etc.	<ul style="list-style-type: none"> There may be more than one provider that needs to be described (e.g., for federation service provided by multiple providers alongside each other). Action interfaces may vary.
Customer	Detailed description of the (expected) customer.	Details of (expected) clients	<ul style="list-style-type: none"> n/a
Responsibilities	Description of the responsibilities that each party has according to the agreement.	Providers' responsibilities, customers' responsibilities, shared responsibilities	<ul style="list-style-type: none"> Highlight responsibilities that are shared among involved providers. Consider inherited customer responsibilities.
SLA General Restrictions	Statements that may restrict and limit the application or validity of the SLA in its entirety.	Validity restrictions, usage restrictions, limitations of liability, ...	<ul style="list-style-type: none"> SLAs may contain different restrictions, which hampers the comparison of SLAs across providers.
SLA Validity Period	Definition of the start and end date of the SLA.	Start date, end date, unlimited	<ul style="list-style-type: none"> Ensure validity of SLAs with sub-providers.

Dimension	Description of Dimension	Key Characteristics	Considerations in Multi-Provider Contexts
SLA Change Management	Description about how to handle changes to the SLA.	Version number, change log, change notification requirements	<ul style="list-style-type: none"> n/a
Contractual Law Context: Jurisdiction and Governing Law	Terms and obligations related to the jurisdiction and governing law agreed upon by the parties in the SLA.	List of countries / locations and their relevant regulations	<ul style="list-style-type: none"> Determine the country's or state's laws that apply.
Statutory Law Context	Laws imposed by governments or regulators that apply regardless of what is written in the contract.	List of mandatory statutory rules	<ul style="list-style-type: none"> Governing law needs to be considered when it comes to cross-jurisdictional, international use cases.

(2) Service Level Objectives: Determining Commitments

Service Level Objective (SLO)	Commitments that a cloud service provider makes for a specific, quantitative or qualitative characteristic of a cloud service.	Examples: availability, performance, capacity, elasticity, data protection & management, information security, reliability, accessibility, customer support, governance, change management, termination of service, sustainability, interoperability, customizability, useability, maintainability	<ul style="list-style-type: none"> Each service may comprise a different set of SLOs. Compatibility of commitments of all parties involved is important. Harmonization of SLOs is useful to ease comparison of SLAs. SLO core classes are not mutually exclusive but rather interdependent and may even overlap.
Commitment Time Period	Defines the time frame for a given commitment.	Example: one month	<ul style="list-style-type: none"> n/a
Service Level Indicator (SLI)	An attribute, parameter, or scale associated with a service that is used to specify or determine a certain quality of the service.	Quantitative: interval scale, ratio scale; qualitative: nominal scale, ordinal scales	<ul style="list-style-type: none"> An agreed-upon list or standard of SLI fosters comparison of SLAs.
Commitment Framing	Defines the tone of SLO/SLI and how the providers' efforts to comply with them are framed.	Positive tone (achievement-focused), negative tone (prevention-focused), best-effort framing	<ul style="list-style-type: none"> How commitments are framed may differ between involved parties.
Commitment Level	An SLO/SLI might be differentiated according to specific service levels, such as gold, silver and bronze commitments.	Specific levels such as starter, essential, premium, max; bronze, silver, gold	<ul style="list-style-type: none"> n/a
Operation Context	Categorizes SLO/SLI based on when they are applicable.	Normal operations, incident response, disaster recovery	<ul style="list-style-type: none"> n/a
Business Impact	Categorizes SLO/SLI based on their potential business impact for customers.	No business impact; minimal business impact; significant business impact; critical business impact (ISO/IEC, 2016a)	<ul style="list-style-type: none"> n/a
Underlying Standard	An SLO/SLI can be based on a specific document, best practice, or recommendation.	Standards, business best practice, industry norm	<ul style="list-style-type: none"> Determining and explicitly listing standards used for SLO/SLI fosters harmonization and increases transparency.

Dimension	Description of Dimension	Key Characteristics	Considerations in Multi-Provider Contexts
(3) Monitoring: Measuring Compliance with Commitments			
Measurement Metric	Standard of measurement that defines the conditions and the rules for performing the measurement of an SLI and for understanding the results of a measurement.	Information, expressions, parameters, measurement rules, results, uncertainty	<ul style="list-style-type: none"> Use a standard set of metrics to ease comparison.
Measurement Scope	The level of granularity at which a metric is observed.	Single resource metric, composite metric, aggregated metric	<ul style="list-style-type: none"> Aggregated metrics are useful for tracking SLOs and ensuring the overall performance of cloud services that depend on resources from multiple providers.
Measurement Responsibility	The distinction by which party is responsible for monitoring and measuring the metrics.	Customer, provider, single third-party observer, multiple third-party observers	<ul style="list-style-type: none"> Clarify who is responsible for measuring service interfaces in case multiple providers are involved.
Measurement Boundaries	Distinction between measuring the internal resources of a cloud service versus measuring how the service is perceived by external users.	Internal monitoring, external monitoring	<ul style="list-style-type: none"> External monitoring may be performed by involved sub-providers.
Measurement Timing	Distinction between measuring before or after an event.	Proactive, reactive	<ul style="list-style-type: none"> Involved providers may align whether they engage in proactive or reactive measurement.
Measurement Invasiveness	Distinction between the general technique used to obtain data.	Active, passive	<ul style="list-style-type: none"> n/a
Measurement Authenticity	Distinction between measuring real-life behavior of the cloud service or relying on testing systems.	Simulation / testing, in operation	<ul style="list-style-type: none"> n/a
(4) Enforcement: Claiming Compensation in case of (Non-)Adherence to Commitments			
Reporting Audience	Who will be informed about SLA compliance.	Providers' internal employees, customers, end users, sub-providers, the general public, regulatory bodies, and related institutions, ...	<ul style="list-style-type: none"> Sub-providers should report to the lead cloud provider to increase transparency.
Reporting Method	How the audience will be informed about SLA compliance.	Ticket, report, website, email, text messages, telephone, social media	<ul style="list-style-type: none"> In multi-provider settings, interoperable dashboards that integrate different reporting systems of providers are valuable.
Reporting Period	The timeframe over which the service quality will be measured, with corresponding results then reported to the audience.	Yearly, quarterly, monthly, daily, hourly, on incident	<ul style="list-style-type: none"> Reporting times should be standardized or at least compatible with all providers involved. Validity of remedy claims may depend on reporting times.

Dimension	Description of Dimension	Key Characteristics	Considerations in Multi-Provider Contexts
Remedies	Compensation for the customer if the cloud provider fails to meet a specified SLO.	Service credits, refunds on charges, service termination, other forms of compensation	<ul style="list-style-type: none"> It is challenging to determine who is responsible and should provide compensation.
Claim Responsibility	Determining who is responsible for identifying cases of non-adherence and initiating a remedy process.	Customer, cloud provider, single third-party observer, multiple third-party observers	<ul style="list-style-type: none"> In most cases, the customer is responsible and has therefore to manage the complex process of claiming remedies from multiple providers. Customers will face challenges to claim remedies from sub-providers in entangled supply chains.
Claim Process	Defining how an actor can initiate a claim for compensation.	Email, support ticket, contact manager, website, compensation forms, etc.	<ul style="list-style-type: none"> n/a
Responsibilities of Customers When Claiming	The duties and tasks that are required by the customer in case of claiming remedies.	Paid all invoices, cooperate in good faith, support resolution of remedy process, providing information, engaging in verification	<ul style="list-style-type: none"> n/a
Claim Content	The information that should be included in the remedy claim to be valid.	Information on the covered and impacted services, information on incident, evidence (e.g., log files), timing of incident, duration of incident, information on applied resolution means by the customer	<ul style="list-style-type: none"> The content may vary between providers, while each of them may impose unique requirements.
Claim Deadline	This defines a time period during which a remedy must be claimed.	Years, months, days	<ul style="list-style-type: none"> n/a
Remedy Decision Style	Information on how the provider decides whether a remedy is valid or not.	Good faith, best-effort, binding, use all information reasonably available, examine and dedicate each claim with the utmost care	<ul style="list-style-type: none"> n/a
Exclusion of Remedies	Some events or incidents may be excluded from liability and declared exceptions. In those cases, even if an SLO or SLI is not met, the related remedy will not be triggered.	Scheduled outages, force majeure, other factors beyond the control of the provider, improper use of service, specific functionality, features designated pregeneral availability, being attacked	<ul style="list-style-type: none"> Exceptions need to be harmonized and consistent across involved parties.
Remedy Compensation Restrictions and Conditions	SLA may impose certain requirements and restrictions for issuing remedies, like limiting the total amount of compensation granted by the cloud provider.	Limiting the amount of compensation, forbidding cumulative compensation, not considering snowballing effects, binding remedies to specific accounts or users, or limiting the number of claims for a specific time period.	<ul style="list-style-type: none"> Entangled supply chains and federated cloud systems bear risks of snowballing effects, which are however not covered in SLAs.

Dimension	Description of Dimension	Key Characteristics	Considerations in Multi-Provider Contexts
(5) Automation: Fostering Machine-Readability of SLA			
Machine-readable Language	A language, ontology, or related mechanism for transferring SLAs into machine-readable representations.	WSLA, WS-Agreement, SLAng, SLAC, RBSLA, SLA*, CSLA, SLALOM, SSLA, knowledge graphs, rSLA, blockchain smart contracts, WSDL, SLAaaS ...	<ul style="list-style-type: none"> Each language comes with benefits and drawbacks, while some languages are considering multi-provider settings.
Focus	Phases of the SLA management lifecycle (not) supported by a language.	Definition, negotiation, establishment, monitoring, compliance checking, enforcement, termination, renewal	<ul style="list-style-type: none"> A combination of SLA languages may be needed to support the entire SLA management lifecycle.
Formalization	The degree of how precisely and unambiguously an SLA language defines semantics, syntax, and logic.	Unstructured, semi-structured, formal	<ul style="list-style-type: none"> Formal or at least semi-structured formalization is required to foster harmonization and standardization that is required in multi-provider settings.
Automation	The degree to which the SLA language enables the automation of SLA management.	Partly, fully	<ul style="list-style-type: none"> n/a
Maturity	An indication about how well a language is applicable and already used in practice.	Not mature at all, somewhat mature, mature, inconclusive	<ul style="list-style-type: none"> Maturity is key to ensuring practical applicability and diffusion.
Generalizability	The extent to which the language is restricted to a specific application domain / context or not.	Web services, generally applicable within the area of cloud services, tailored to a specific application domain, fog computing	<ul style="list-style-type: none"> Applicability to cloud service contexts need to be ensured.
Tool Support for Language	Determine whether and what type of tool is available to use the language for SLA modelling.	None, freeware tool, open-source tool, proprietary tool	<ul style="list-style-type: none"> n/a

3.2 Foundation: Setting the Stage for the SLA

SLAs typically have a standardized structure and contain information that is required as a minimum. Multi-provider SLA management demands that the foundation of each SLA is compared and aligned with all parties involved. This also includes even seemingly minor building blocks like defining

terms in the beginning of each SLA because it reduces risks of false interpretation of SLA information, legal loopholes due to vague content, and mismatches in commitments across multiple providers. Table 1 summarizes key dimensions that need to be considered as foundation in each SLA.

Dimension	Description
Definition of Terms	A section that determines each term and provides definition for clarity.
Covered Services	A detailed definition and description of the services (NOT) covered by the SLA.
Interdependency of Services	Definition of the service’s dependencies on upstream or downstream services and their sub-providers.
Roles of Involved Parties	A description of the roles of the parties involved.
Provider	Detailed description of the provider and their interfaces involved in the SLA.
Customer	Detailed description of the (expected) customer.
Responsibilities	A description of the responsibilities that each party has according to the agreement.
SLA General Restrictions	Statements that may restrict and limit the application or validity of the SLA in its entirety.
SLA Validity Period	Definition of the start and end date of the SLA.
SLA Change Management	Description about how to handle changes to the SLA.
Contractual Law Context: Jurisdiction and Governing Law	Terms and obligations related to the jurisdiction and governing law agreed upon by the parties in the SLA itself.
Statutory Law Context	Laws imposed by governments or regulators that apply regardless of what is written in the contract.

Table 1 – Summary of dimensions for SLA foundation

3.2.1 Definition of Terms

SLAs should start by defining key terms to reduce the risk of misinterpretation. SLAs typically define and explain terms like *Unavailability / Downtime, Service Interruption, Maintenance / Scheduled Service Window and Force Majeure*. While this building block may seem intuitive and less important, it lays the groundwork for the entire SLA and gains more importance when considering that SLAs are often written by legal experts but implemented by technicians. Creating a common ground helps to ease communication between those two expert groups and ensures that service level commitments are actually monitored and implemented by socio-technical means.

It is recommended that the SLA uses definitions from industry standards when possible (ISO/IEC, 2016a). Cloud providers should also critically reflect whether certain terms are domain- or context-specific and therefore require additional clarification. For example, services operated in the healthcare sector may refer to different terms than those used in the automotive industry.

3.2.2 Covered Services and Their Interdependencies

Each SLA comprises names, detailed definitions, and descriptions of the services that are or are not covered by the SLA (ISO/IEC, 2016a). The service scope should be clearly defined, including the functional and non-functional attributes of the service. Most SLAs only apply to paid services and therefore exclude any services offered for free. Information on the service's pricing model and any service exclusions may thus also be included.

Services from one or several providers may be interdependent and rely on each other, particularly in the case of entangled supply chains. SLAs must clearly state their scope and interdependence on other services. SLAs can either cover a single service from one provider, multiple services from one provider, or, theoretically, even span services provided by different providers (Table 2).

Characteristic	Description	Example / Reference
Individual Service SLA	SLA covers one service from one provider. No interdependency.	<ul style="list-style-type: none"> Google Workspace Service Level Agreement Google Cloud Datastore Service Level Agreement Amazon S3 Service Level Agreement OVH Public Cloud Service Level Agreement VMware Cloud™ on AWS Service Level Agreement
Cross-Service SLA (Intra-Provider)	SLA covers multiple services from one service provider. Services may depend on each other.	<ul style="list-style-type: none"> IONOS Service Level Agreement Adobe On-Demand and Managed Service – Unified Service Level Agreement Service Level Agreement for Microsoft Online Services Oracle PaaS and IaaS Public Cloud Services Pillar Document EQS Cloud Services: Service Level Agreement
Cross-Provider SLA (Inter-Provider)	SLA spans services provided by different providers. Services may depend on each other.	<ul style="list-style-type: none"> n/a

Table 2 – Overview of characteristics assigned to the Interdependency of Services dimension

FACIS' Recommendations:

Already proposed initiatives have unified and standardized service description languages and templates that may be harnessed. For instance:

- Modeling Service Level Agreements with Linked USDL Agreement (Garcia et al., 2017): A semantic model to specify, manage, and share SLA descriptions. This model is part of the Linked USDL family of ontologies that can describe not only technical but also business-related aspects of services.
- Unified Semantic Cloud Service Description Model (CSDM) (Sun et al., 2018): The model extends the basic structure of USDL by defining cloud-service-specific attributes. An additional transaction module models the rating system of cloud services from several aspects, such as risk, trust, and reputation. An OWL-based annotation system is proposed to enrich the semantic expressivity of this model.
- A Service Level Agreement Language for Cloud Services (rSLA) (Tata et al., 2016): Language for specifying and enforcing SLAs for cloud services, allowing for dynamic instrumentation of heterogeneous cloud services and instantaneous deployment of SLA monitoring.

Although mentioned briefly in some SLAs (e.g., in the introduction of the IBM Cloud SLA), clear statements on interdependencies between services from providers remain scarce. However, such interdependency, if not made transparent and accounted for, can become problematic in entangled supply chain scenarios covering the involvement of multiple sub-providers. SLAs of services that depend on each other must state dependencies and be overall compatible with one another. It is thus crucial to extend the service description about how each provider's service is involved in the service provisioning.

It is recommended to

(1) describe the connection between involved services (e.g., summarizing data flows), and

(2) to position each service along the cloud service stack (refer to Singh et al. (2016) and the NIST Cloud Computing Reference Architecture (F. Liu et al., 2011)).

Interdependent services' SLAs should be referenced (Falasi et al., 2013). This will increase the transparency for cloud customers and open the black box of entangled supply chains.

3.2.3 Roles of Involved Parties and Their Responsibilities

Each SLA comprises a description of roles and responsibilities for both the provider and the customer (ISO/IEC, 2016a). Each party that participates in a transaction or process and/or performs tasks in cloud computing is an entity (a person or an organization) and most commonly takes over one of the following roles: customer, end user, (lead) provider, sub-provider, broker, carrier, or auditor (Table 3). Supplementary roles may cover aggregators, integrators, platform operators, or consultants. Refer to Floercke et al. (2021) for a detailed description of common actors and roles in cloud service ecosystems.

SLAs should also detail information of the service provider(s) and the customer, whereas the description of the provider may include a description of the action interfaces exposed to other parties (e.g., service consumer, third party observers) (Binu & Gangadhar, 2014). Some SLAs also include information on potential customers to outline what the provider expects from customers and assign customer responsibilities.

Role	Description
Customer	A person or organization that maintains a business relationship with and uses services from cloud providers.
End users	A person that uses the cloud services' functionality.
(Lead) Provider	A person, organization, or entity responsible for making a service available to interested parties.
Sub-provider	A third-party service provider engaged by a lead cloud provider to deliver specific components, functionalities, or support services that are part of the overall cloud service. These can include infrastructure, platform components, software services, or operational support, depending on the scope of the primary service.
Broker	An entity that manages the use, performance, and delivery of cloud services, and negotiates relationships between cloud providers and cloud consumers.
Carrier	An intermediary that provides connectivity and transport of cloud services from cloud providers to cloud consumers or operates the underlying data center infrastructure.
Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance, and security of the cloud implementation.

Table 3 – Common roles in cloud computing (adapted from Liu et al., 2011)

Cloud services typically involve single and shared responsibilities among all parties, often characterized by blurry lines of who is responsible for what. In essence, a provider is responsible for making a service available to interested parties (F. Liu et al., 2011).

The provider thus acquires and manages the computing infrastructure required, runs the cloud software that provides the services, and makes arrangement to deliver the cloud services to the consumers through network access. Table 4 lists example responsibilities of providers observed in SLAs.

Responsibility	Example / Reference
Service monitoring	<ul style="list-style-type: none"> “Each Cloud One Solution systems, networks, and capacity are continually monitored by Trend Micro and its IaaS third party service provider to provide optimal availability and efficiency to Cloud One Solution customers.” (TrendMicro SLA)
Service maintenance	<ul style="list-style-type: none"> “Scheduled maintenance shall not exceed more than eight (8) hours per calendar month except as may be imposed by any IaaS third party service provider over which Trend Micro has no control.” (TrendMicro SLA)
Customer support	<ul style="list-style-type: none"> oneclick also provides additional support for third-party hardware and software products purchased and licensed via oneclick if the customer has booked a corresponding support package (oneclick SLA)

Table 4 – Example responsibilities of providers

SLAs can include customer responsibilities that describe obligations on the customer-side. These include refraining from illegal activities or those considered unwanted by the service provider, but also technical prerequisites for using the service on the customer side (Table 5). For instance, the hardware on which the service should be operated can be indicated in the SLA, to guarantee correct hardware for the service (Binu & Gangadhar, 2014). The inclusion of customer responsibilities can be crucial to clearly stating what customers are expected (not) to do when using a service.

Since the cloud provider and customer often share the control of resources in a cloud service, they also share responsibilities (F. Liu et al., 2011). For example, security is a shared responsibility. Ensuring a high level of IT security requires the operation of technical measures and safeguards by the provider (e.g., strong encryption, multi-tenancy measures) but also requires customers to use the cloud service securely (e.g., following password policies).

Responsibility	Description	Example / Reference
Refrain from illegal activities	Responsibility of the service user to refrain from performing illegal activities using the service	<ul style="list-style-type: none"> “You can’t do illegal things on that” (Participant_2, provider workshop¹) “The customer must take all necessary measures to ensure that it and all of its employees are aware of and in compliance with any requirements, responsibilities and limitations set forth in the Terms of Use, including, without limitation, any applicable data privacy and data protection laws, rules, and regulations, as well as Trend Micro’s Acceptable Use Policy as published in the Terms of Use.” (TrendMicro SLA)
Refrain from unwanted activities	Responsibility of the service user to refrain from performing unwanted activities using the service	<ul style="list-style-type: none"> Spam messages (Participant_2, provider workshop) Non-Adherence to licenses granted (Participant_2, provider workshop)
Fulfill technical prerequisites	Responsibility to meet the given technical prerequisites	<ul style="list-style-type: none"> “Technical prerequisites have to be provided by the customer to be able to rely on these functional availabilities” (ProjectPartner_1, provider workshop) “Customer’s network must be properly configured pursuant to the Documentation” (Lookout SLA)

Table 5 – Example responsibilities of customers

FACIS’ Recommendations:

In multi-provider environments, clear risk allocation (e.g., for data loss, outages, or breaches) is essential to reduce blame chain issues. In such cases, service providers may blame either customers for misbehavior (e.g., not configuring the services correctly has led to a service downtime), or blame sub-providers

(e.g., infrastructure services not functioning correctly, which are not in the control of the provider). Customers will then struggle to enforce SLA commitments and claim any compensation for non-adherence.

¹The term “provider workshop” refers to the FACIS “Workshop on SLA Governance for Service Providers,” held in Cologne on March 5th 2025. For details on the workshop, please refer to the FACIS workshop website and the workshop results overview.

Statements on customer or shared responsibilities become critical in multi-provider settings because several providers can have different perspectives on acceptable use and technical prerequisites. This can become particularly complex in entangled supply chain scenarios: statements made by providers using other sub-providers' services need to be consistent with those responsibilities they agreed to take over from their sub-providers. Certain customer responsibilities may thus be inherited from agreements made with sub-providers. Furthermore, each provider's customer responsibility statements must be made transparent to the end user, so that they understand what they must adhere to and for which of the services they are using.

3.2.4 General SLA Restrictions

On a general level, different statements may be included that restrict and limit the SLA in its entirety (Table 6). Among other things, restrictions cancel out the entire validity of the SLA in special cases where the provider has no control, such as in cases of force majeure like war, natural disasters, or union strikes. Restrictions can also relate to customers' service usage (e.g., forbidding downloading or copying any part of the service), or preserving intellectual property of the cloud provider (e.g., restricting the use to build or support services competitive to the provider). Restriction can also relate to specific services – for instance, an SLA is not applicable for public or free services. SLAs often require customers to pay all invoices, otherwise customers cannot enforce any commitments.

The cloud provider may further add additional exclusion criteria for dealing with remedies, as outlined in Section 3.5.3.

Restriction	Example / Reference
Validity restrictions like force majeure	<ul style="list-style-type: none"> “Neither party shall be responsible for failure or delay of performance if caused by: an act of war, hostility, or sabotage; act of God; pandemic; electrical, internet, or telecommunication outage that is not caused by the obligated party; government restrictions; or other event outside the reasonable control of the obligated party.” (Oracle Cloud SLA)
Usage restrictions	<ul style="list-style-type: none"> “You may not, and may not cause or permit others to: (a) modify, make derivative works of, disassemble, decompile, reverse engineer, reproduce, republish, download, or copy any part of the Services (including data structures or similar materials produced by programs).” (Oracle Cloud SLA)
Limitation of liability	<ul style="list-style-type: none"> “Neither party nor its affiliates will be held responsible for any indirect, consequential, incidental, special, punitive, or exemplary damages. This includes any loss of revenue, profits (excluding fees specified in this agreement), sales, data, use of data, goodwill, or reputation.” (Oracle Cloud SLA)

Table 6 – Example restrictions contained in SLAs

FACIS' Recommendations:

We observed that those restrictions and corresponding conditions were described in varying detail across SLAs. Some SLAs contained specific obligations for SLA validity, while others were superficial, and it is not entirely clear when they apply. At the moment, it remains open how these conditions

can be standardized for usage and combination in a multi-provider scenario. Consequently, we recommend providing sufficient information in an easy-to-understand way to increase transparency and reduce risks of misinterpretation.

3.2.5 SLA Validity Period and Change Management

The SLA covers further meta-data that is relevant for the agreement. The importance of the validity period was highlighted by several SLA standards and frameworks. The time for which a SLA is valid is mentioned with date of starting and date of ending of agreement (M. Singh et al., 2021), or by stating that the SLA is valid for an indefinite period until the service usage is terminated. Besides, SLAs evolve, especially in (federated) cloud environments. Tracking changes, version control, and update propagation are thus crucial. SLAs often described how an SLA is updated and whether the customer will be informed or has the opportunity to reject.

For example, the SLA of Lookout states that: "Lookout reserves the right to update this SLA from time to time after providing thirty (30) days advance notice. Notices will be sufficient if provided to the administrator of your Cloud Security Service account either: (a) as a note on the screen presented immediately after completion of the log-in authentication credentials at the log-in screen, or (b) by email to the registered email address provided for the administrator(s) for Customer's account." Aruba's SLA even offers the chance to opt out of the contract: "However, in this case the Customer shall be given the opportunity to withdraw from the contract according to the rules laid down in contract within thirty days of the date of publication of the change and/or the replacement of the SLA."

On the contrary, we also observed that SLAs may contain harsh statements about how the provider deals with updates to the SLA. For instance, TrendMicro stated that "Trend Micro reserves the right to modify this Service Level Agreement at any time without prior notice," similar to Alibaba's cloud SLA that stated: "We reserve the right to change the terms of this SLA anytime by posting an amended and restated version of this SLA on the Alibaba Cloud International Website. Your continued use of the service after the publication of the amended SLA shall be deemed as your acceptance of the amended SLA." Under such circumstances, customers will be unable to deal with SLA changes but rather accept them in general.

FACIS' Recommendations:

Entangled cloud service chains put an additional burden on validity periods and change management. The lead cloud provider must ensure that all SLAs with their sub-providers remain valid at least until specified in the SLA with the customer. Otherwise, notification mechanisms that inform the cloud customers about changes in the SLA must be implemented.

3.2.6 Contractual and Statutory Law Context of an SLA

SLAs also include legal "boilerplate clauses," meaning standard provisions commonly found in commercial contracts, such as severability or written form clauses. They also determine contractual and statutory law contexts. As with any contractual agreement, also SLAs are governed by the laws of one specific country. The governing law is normally explicitly defined within the contract by the parties. If not, the governing law will be defined by the international civil law rules.

Besides, each country has its unique set of mandatory statutory rules that contractual agreements cannot overrule. Such mandatory rules frequently affect warranty and liability provisions. Statutory law imposes non-negotiable rules that can override contracts, especially in areas like data protection, cybersecurity, and consumer rights.

FACIS' Recommendations:

Multi-provider settings may involve providers that are located in different countries. This can lead to inconsistencies and conflicts in mandatory statutory rules. For example, a warranty clause of an SLA that is in line with UK law may be regarded as in breach of German statutory rules and, therefore, be legally void if the SLA is governed by German law. Consequently, the topic of governing law will have to be considered when it comes to cross-jurisdictional, international use cases.

Concluding FACIS' Recommendations Related to SLA Foundations

Each SLA has a foundation that comprises important information about the service and agreement. In multi-provider contexts it is of high importance to ensure that all information is described in a transparent manner to cloud customers and that relevant service interdependencies and responsibilities of each party are determined, despite the involvement of multiple parties. In addition, consistency must be ensured across all SLAs because a customer may inherit foundations that are specified by the SLA between the provider and its sub-providers.

To tackle responsibility challenges, we propose three major *coordination models* of sharing responsibility and engaging in multi-provider SLA management:

1. Determine a *lead service provider* (sometimes called hub-and-spoke), most often the provider who directly engages with the customer. The lead service provider has then the responsibility to ensure that the SLA they offer is backed up by and consistent with the SLAs they have agreed with their supplying sub-providers (TM Forum, 2014). The lead service provider then takes over an "integrator role" that always holds the responsibility for ensuring that all commitments made for the overall service offering can be fulfilled by each involved party.
2. Determine a *dedicated SLA broker*. This can be a cloud provider involved in service provisioning or a third party. The SLA broker takes care of harmonizing SLAs across the cloud ecosystem. The SLA broker then acts as a type of SLA integrator or orchestrator and thereby facilitates the SLA management process. The SLA broker may take over different roles and tasks, depending on the ecosystem, service complexity, and number of customers, among others.
3. *Decentralized (multi-party) SLA management*. Following the principles of ecosystems and decentralization, cloud customers and providers can also engage in *n-to-n* SLA management, where each provider interacts with the cloud customer and supposedly also with each other provider. No centralized or dedicated SLA broker role exists but rather customers (and supposedly providers) have set up mechanisms that enable direct communication (e.g., performance reports) and SLA management between all parties involved.



3.3 Service Level Objectives: Determining Commitments about Service Levels

An SLA covers multiple service level objectives (SLOs). They are commitments that a cloud service provider makes for a specific, quantitative or qualitative characteristic of a cloud service (ISO/IEC, 2016a). The SLO thus determines what is promised to the customer. SLO commitments are typically operationalized by referring to specific service level indicators (SLIs). Similar to key performance indicators, SLIs determine how to assess whether a specific SLO commitment is fulfilled or not. For example, a provider may commit to ensure high availability of the service. A potential SLI can then be “uptime” defined as the percentage of time in a given period that the cloud service is accessible and usable, like 99.999%. One or multiple metrics can be applied to measure an SLI. For example, MTBF (Mean Time Between Failure), MTTR (Mean Time To Repair), or MTTF (Mean Time To Failure) are common metrics used to measure SLIs related to availability and reliability. Figure 2 illustrates the relationship between the concepts of SLO, SLI, and metric. The Appendix provides an example of this 3-layered hierarchy applied to “service availability.”

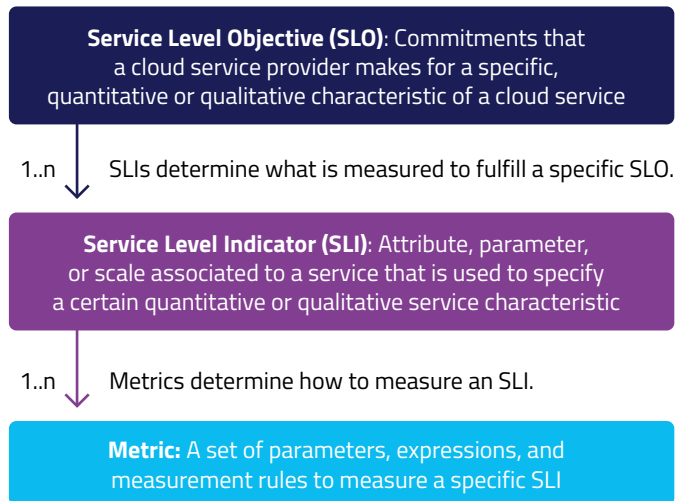


Figure 2 – Illustration of the relationship between SLO, SLI, and Metric.

Not all SLOs will apply to every cloud service, and those that do apply may be framed and applied in different ways to specific cloud services (ISO/IEC, 2016a). For example, service availability can be measured in different ways for IaaS and SaaS offerings. A service availability for compute services might be reported against an SLO of uptime, where uptime would be described in the context of compute services by using SLI like “instances being accessible and usable upon demand by an authorized entity” (ISO/IEC, 2016a, p. 9). In contrast, for storage services, potential SLI can be “requests for the stored object returning errors”

(ISO/IEC, 2016a, p. 9). Differences in SLOs and assigned SLIs challenge the comparability of SLAs in multi-provider contexts, as illustrated in Figure 3.

To support comparability of SLOs and SLIs, an SLA can also provide reference to underlying standards, including (ISO) norms, business best practices, or industry norms.

Table 7 summarizes key dimensions related to SLO that need to be considered.

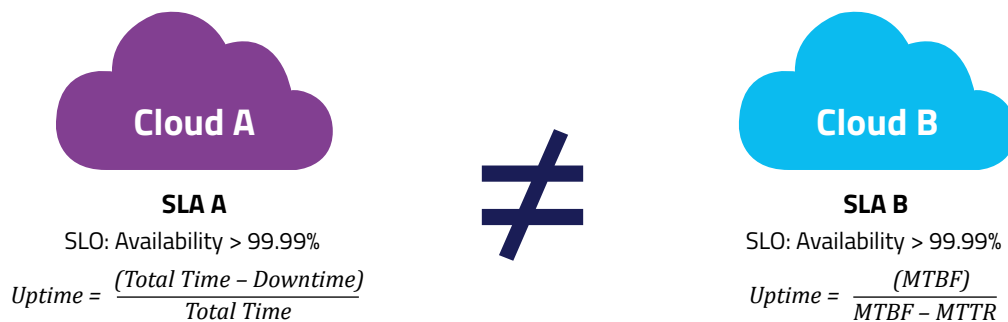


Figure 3 – Side-by-side comparison of the availability of SLO from two different providers that use different metrics to measure uptime.

Dimension	Description
Service Level Objective (SLO)	Commitments that a cloud service provider makes for a specific, quantitative or qualitative characteristic of a cloud service.
Commitment Time Period	Defines the time frame for a given commitment.
Service Level Indicator (SLI)	An attribute, parameter, or scale associated with a service that is used to specify or determine a certain quality of the service.
Commitment Framing	Defines the tone of SLO/SLI and how providers' efforts to comply with them are framed.
Commitment Level	An SLO/SLI might be differentiated according to specific service levels, such as gold, silver and bronze commitments.
Operation Context	Categorizes SLO/SLI based on when they are applicable.
Business Impact	Categorizes SLO/SLI based on their potential business impact for customers.
Underlying Standard	An SLO/SLI can be based on a specific document, best practice or recommendation.

Table 7 – Summary of dimensions for SLO



3.3.1 Core Classes of Service Level Objectives

SLO refer to commitments that determine what is promised to the customer about a specific, quantitative or qualitative characteristic of a cloud service (ISO/IEC, 2016a). SLAs can define the time frame for a given commitment. For instance, most SLAs promise to meet availability rates for one month.

While diverse SLOs exist, they can be aggregated into several classes that form a core set of SLOs typically covered in SLAs, as summarized in Table 8. These SLO core classes are not mutually exclusive but rather interdependent and may even overlap. For example, commitments related to backing up data are important for data management and protection, as well as in regard to ensuring service reliability. Analyzed SLAs mostly cover SLOs related to availability and reliability.

Service Level Objective	Description	Example of potential SLI
Availability	Availability is the property of being accessible and usable upon demand by a customer (ISO/IEC, 2016a).	<ul style="list-style-type: none"> Uptime –The amount or percentage of time in a given period that the cloud service is accessible and usable (ISO/IEC, 2016a) Availability and accessibility of functionality (e.g., being able to perform operations) or data Rejection Probability (Xia et al., 2013)
Performance	Commitments that can be used to express the performance of service components (ISO/IEC, 2016a).	<ul style="list-style-type: none"> Maximum/mean response time or its variance (ISO/IEC, 2016a) Latency (e.g., the time it takes for data to travel from one point to another) Jitter (i.e., variation in latency over time) Packet Delivery/Packet Loss Service Response Time Function Veracity Transaction Response Time (Zhao et al., 2015)
Capacity	Capacity of the cloud resources (such as storage space, processing power) and the capacity of the network used to access the resources (ISO/IEC, 2016a).	<ul style="list-style-type: none"> Maximum capacity of available resources like disk space or GPU (ISO/IEC, 2016a) Maximum number of simultaneous cloud users (ISO/IEC, 2016a) Service throughput (ISO/IEC, 2016a) (e.g., volume of data processed within a specific timeframe) Bandwidth capacity of network services Network speed (Zhao et al., 2015)
Elasticity	The ability of a cloud service to dynamically adjust the amount of resources that are allocated to an instance of the service (ISO/IEC, 2016a).	<ul style="list-style-type: none"> Elasticity speed quantity: describes how fast a cloud service is able to react to a resource request (ISO/IEC, 2016a) Time-to-set-up, time-to-ramp-up Boot Time (Xiaoyong et al., 2015) Elasticity precision quantity describes how precise the resource allocation meets the actual resource requirements at a given point in time (ISO/IEC, 2016a) Scalability focusing on vertical or horizontal scaling
Data Protection & Management	Protection of personal identifiable information, intellectual property, and related data management practices.	<ul style="list-style-type: none"> Data location (ISO/IEC, 2016a) Data deletion time (ISO/IEC, 2016a) Refer to ISO/IEC 19086-4 for details
Information Security	Commitments covering security measures for ensuring availability, confidentiality, and integrity.	<ul style="list-style-type: none"> (Network) Slicing Security Patch Time Physical Security of Facilities Refer to ISO/IEC 19086-4 for details

Service Level Objective	Description	Example of potential SLI
Reliability	Commitments covering fault tolerance and resilience, backups, and disaster recovery (ISO/IEC, 2016a).	<ul style="list-style-type: none"> Time to Service Recovery (ISO/IEC, 2016a) Backup Interval and Retention Period for Back-up Data (ISO/IEC, 2016a) Recovery Time Objective (ISO/IEC, 2016a) Redundancy (e.g., what type of redundancy) Error rates (e.g., acceptable levels of errors, such as failed API calls)
Accessibility	Commitments about assistive technologies that the provider implements (ISO/IEC, 2016a).	<ul style="list-style-type: none"> Supported Accessibility Standards (ISO/IEC, 2016a)
Customer Support	Support commitments for the covered services that are available to customers (ISO/IEC, 2016a).	<ul style="list-style-type: none"> Support hours and methods (ISO/IEC, 2016a) Incident Notification time (ISO/IEC, 2016a) (e.g., how quickly the provider responds to incidents or requests) Maximum First Support Response Time (ISO/IEC, 2016a) Maximum Incident Resolution Time (ISO/IEC, 2016a) Incident remedial actions of providers
Governance	Commitments to support and/or comply with regulations, standards, and related requirements (ISO/IEC, 2016a).	<ul style="list-style-type: none"> A list of regulations including the name, clause, and certification number (if applicable), that the provider attests to or has been certified to comply with (ISO/IEC, 2016a) Internal / external audit schedule that the provider undertakes (ISO/IEC, 2016a)
Change Management	Commitments related to changes to features and functionality.	<ul style="list-style-type: none"> The minimum period of time between the issuance of a service change notification and the implementation of the change (ISO/IEC, 2016a) The minimum time period between the initial availability of a feature or function and deprecation of that feature or function (ISO/IEC, 2016a) Management of changes in services (e.g., through novel releases) on which the provided service depends on
Termination of Service	Commitments related to the exit process, where the use of a cloud service is terminated and there is an orderly process by which the customer stops using the cloud service (ISO/IEC, 2016a).	<ul style="list-style-type: none"> Data Retention Period (ISO/IEC, 2016a) Return of Assets (ISO/IEC, 2016a)
Sustainability	Commitments to measurable environmental performance indicators when delivering cloud services.	<ul style="list-style-type: none"> Energy Efficiency Power Usage Effectiveness Carbon Footprint Use of Renewable Energy
Interoperability	Commitments about the extent to which cloud services support standardized interfaces, formats, and APIs that allow (seamless) exchange and use of data and services among various cloud services.	<ul style="list-style-type: none"> Data portability formats OpenAPI 3.0-compliant interfaces

Service Level Objective	Description	Example of potential SLI
Customizability	Commitments about the degree to which the cloud service can be tailored to meet the specific (non-) functional needs of the customer (ISO/IEC, 2017b).	<ul style="list-style-type: none"> Customer-defined Roles and Access Policies Customization Deployment Time
Usability	Commitments about the degree to which a cloud service can be used by customers to achieve specified goals with effectiveness, efficiency, and satisfaction (ISO/IEC, 2017b).	<ul style="list-style-type: none"> Learnability (ISO/IEC, 2017b) User error protection (ISO/IEC, 2017b) IT service interface appearance (ISO/IEC, 2017b)
Maintainability	Commitments related to the degree of effectiveness and efficiency with which a service can be modified by the intended maintainers (ISO/IEC, 2016c).	<ul style="list-style-type: none"> n/a
Data Quality	Commitments related to the characteristics of data when used under specified conditions (ISO/IEC, 2015).	<ul style="list-style-type: none"> Accuracy, completeness, consistency, currentness (ISO/IEC, 2015) Data Freshness (Zhao et al., 2015) Fairness

Table 8 – Overview of characteristics assigned to the SLO dimension: Core classes of SLO



3.3.2 Service Level Indicators

Service Level Indicator (SLI) is an attribute, parameter, or scale associated to a service that is used to specify or determine a certain quality of the service (Girs et al., 2020). SLIs are either of quantitative or qualitative nature and are often expressed in the form of scales. An SLI determines what is measured to fulfill a specific SLO. For each SLO, multiple SLIs may exist, and they may be individually or jointly monitored. Metrics define how an SLI is measured.

SLIs are most often compared to a target value (e.g., the minimum acceptable availability rate is 99.5%), by using a benchmark for a similar service (e.g., the mean response time of the new service in no more than the mean response time of the old service), or compared over time (e.g., time series, how the uptime rate changed over the month) (ISO/IEC, 2016c). Cloud providers should also consider whether SLIs and corresponding metrics need to be adapted because their cloud service evolved over time.

Following the ISO/IEC 19086-1 (2016a), we distinguish SLIs between:

(A) Quantitative SLIs, with

- **interval scale** that refers to numeric scales with equal intervals between values but no true zero (e.g., response time guarantees with time windows like within 4 hours, 2 hours, or 1 hour depending on SLA tier)
- **ratio scale** that refers to numeric scales with equal intervals and a true zero point (e.g., uptime percentage: e.g., 99.0%; number of incidents per month: e.g., 0, 2, 5; latency in ms: e.g., 0 ms, 100 ms, 200 ms)

(B) Qualitative SLIs, with

- **nominal scale** that does not have a natural order or ranking (e.g., service region: US-East, EU-Central, AP-South; security certificate ISO/IEC 27001: yes/no)
- **ordinal scale** that categorizes data with a meaningful order, but the intervals between categories are not uniform or known (e.g., priority level of support: Basic, Standard, Premium)

Lists and examples of SLIs can be found here:

- ISO/IEC 19086-3: Core conformance requirements (ISO/IEC, 2017a)
- ISO/IEC 19086-4: IT security and data protection (ISO/IEC, 2019)
- NIST Cloud Computing Service Metrics Description (De Vault et al., 2017)
- A Taxonomy of Quality Metrics for Cloud Services (Guerron et al., 2020)

Further research articles develop, evaluate, or review SLIs as well (e.g., Becker et al., 2015; Şener et al., 2024; Zheng et al., 2014; P. Zhou et al., 2015).

In addition, the ISO/IEC 250XX-family of standards provides information on quality models (incl. definitions of quality characteristics) and measurements. Particularly relevant are:

- ISO/IEC 25010: Product quality model; ISO/IEC 25011: IT service quality models; ISO/IEC 25012: Data quality model; ISO/IEC 25019: Quality-in-use model.
- ISO/IEC 25022: Measurement of quality in use; ISO/IEC 25023: Measurement of system and software product quality; ISO/IEC 25024: Measurement of data quality; ISO/IEC 25025: Measurement of IS Service quality; ISO/IEC 25059: Quality model for AI systems.



3.3.3 Commitment Framing

Commitments made to the customer within the agreement can be framed differently, which also impacts on what exactly is promised to the customer. We particularly observed that SLO and

SLI can differ in their commitment tone (positive vs. negative) and providers' efforts to fulfill these commitments (see Table 9).

Characteristic	Description	Example
Positive Framing (achievement-focused)	Definition of what constitutes a "good" (quality of) service and framing of assurance thereof.	<ul style="list-style-type: none"> ▪ "Aruba will make every reasonable effort to ensure maximum availability of the virtual infrastructure created and allocated by the Customer" (Aruba)
Negative Framing (prevention-focused)	Definition of what constitutes a "bad" quality of service and framing of avoidance thereof. Commitment of preventing undesirable, "bad" quality of service.	<ul style="list-style-type: none"> ▪ "oneclick is obliged to process or rectify any errors that occur to the best of its knowledge and in accordance with the technically applicable standard" (oneclick™)
Best Effort Framing	Assurance of undertaking "commercially reasonable efforts" to adhere to commitments made.	<ul style="list-style-type: none"> ▪ "Commercially reasonable efforts" (OHVcloud) ▪ "As possible at all times" (Elastx) ▪ "Best effort" (Vultr) ▪ "Reasonable efforts" (Tencent)

Table 9 – Overview of characteristics assigned to the Commitment Framing dimension

Providers can choose to either define "good" quality of service and positive outcomes they promise to their customers, relying on a positive tone and taking a more achievement-focused framing. Aruba's SLA states, for instance, that "Aruba will make every reasonable effort to ensure maximum availability of the virtual infrastructure created and allocated by the Customer."

Alternatively, providers can define "bad" quality of service and negative outcomes they promise to avoid, using a negative framing and resembling a prevention-focused framing. For example, oneclick™ states in its SLA that: "Disruptions can be caused by oneclick™ components, third-party components, infrastructure components or disruptions in the general network stability on the Internet, among other things. oneclick™ always endeavors to guarantee the operation of the platform despite these influences." And "oneclick is obliged to process or rectify any errors that occur to the best of its knowledge and

in accordance with the technically applicable standard.² The Vultr company similarly describes what they are not committed to do: "Vultr does not proactively monitor the packet loss or transmission latency of specific customers. [...] In the event that Vultr discovers (either from its own efforts or after being notified by You) that You are experiencing packet loss in excess of one percent (1%) [...], Vultr will take all actions necessary to determine the source of the Excess Packet Loss/Latency." They also report exclusions of their commitments in a transparent manner: "Vultr Cloud GPU product has limited availability."

² Statements from oneclick's SLA were translated into English by the FACIS team.

Independent of positive or negative framing, providers tend to characterize their efforts to keep their commitments by including clauses like using *commercially reasonable efforts* in service provisioning, pushing the boundary of non-adherence, and impacting when penalties apply. Such *best-effort approaches* are popular among the reviewed SLAs. VMware, for instance, states that they “will use commercially reasonable efforts to ensure that, during any given billing month of the Subscription Term, Availability of each component of the Service Offering (‘service component’) meets the ‘Availability Commitment’ specified in the table below.”

Those effort-related framing may also consider the business value that can be achieved or is at risk with commitments (Open Grid Forum, 2007). For example, in an SLA representing resource reservation for job submission, the customer may express the “importance” of meeting a commitment, while a provider may specify “confidence” or “likelihood” of meeting that commitment. In an untrusted or cross-organizational scenario, the business value may be expressed as a joint assertion using “penalty” or “reward” value type. A penalty expresses indirectly both the importance to a consumer, where a higher penalty is more likely to motivate the provider to meet this objective, and also specifies compensation to be assessed for failing to meet the objective.

FACIS’ Recommendations:

In a multi-provider environment, the general Commitment Framing should be consistent or at least compatible across all providers involved. Otherwise, it becomes difficult to assess what exactly is promised to the service customer in the end, and when the quality of service drops below the levels assured by the providers. However, concepts like “best effort” become blurry in cloud settings characterized by shared responsibility and accountability, and therefore require more clarification to achieve transparency.

3.3.4 Classifying the Applicability of Service Level Objectives and Indicators: Commitment Levels, Operation Context, and Business Impact

The SLA may contain further meta data to contextualize and classify SLOs and SLIs, ultimately increasing transparency for customers. First, an SLO/SLI might be differentiated according to specific service levels, such as gold, silver, and bronze commitments. For instance, Cloud-IAM states that “The determination of uptime is contingent upon the support level chosen by the Client,” differentiating between starter, essential, premium, and max support levels.

Second, SLOs and SLIs can be categorized based on when they are applicable, like normal operations, incident response, or disaster recovery. For instance, Vultr states that “the uptime guarantee ONLY applies to network and instance availability during normal operation.”

Finally, SLOs and SLIs can be dependent on the business impact in case of non-adherence (e.g., no business impact, minimal, significant, or critical business impact (ISO/IEC, 2016a)). For instance, Elastic offers different maximum response times depending on the business-critical impact of an incident that was reported by a customer.



Concluding FACIS' Recommendations Related to SLA Foundations

Finding a shared understanding of predominant SLOs and SLIs is challenging due to semantic ambiguity and different perspectives on each component. We need to consider the specifics of each service and the service provisioning scenario/use case to identify a minimum set of SLA objectives, indicators, and metrics. Depending on the service, for example, not only do SLO definitions and SLI metrics differ, but also their importance.

We recommend

- (1) agreeing on a set of core SLO classes (cf. Table 8),
- (2) defining a general set of SLIs that fit to those core SLO classes, and finally
- (3) adjusting the specific manifestation of SLIs to fit the service's unique use case and the federated environment by selecting appropriate metrics.

In multi-provider and federated service ecosystems, it can be reasonable that a multi-party SLA defines overall SLOs that consider the entire cloud stack. An SLA may thus contain individual service level statements for the specific service, as well as intra-service statements within a single-provider (intra-provider SLO/SLI) or even inter-service statements that span services offered by multiple providers (inter-provider SLO/SLI). Hence, owners and responsible roles for the service levels statements must be settled, and accountability needs to be defined (e.g., shared accountability, single accountability, customer responsibility).

3.4 Monitoring: Measuring Compliance with Commitments

It is essential to verify that the cloud provider complies with its SLOs and their corresponding SLI. Cloud customers are interested in monitoring SLA compliance and may request compensation for SLA non-adherence. The SLA should therefore comprise specific information on how SLIs are monitored by defining explicit metrics. In addition, providers may agree to offer reporting capabilities or monitoring tools for customers to enable the monitoring of services' performance and ensure that they are operating as promised (ISO/IEC, 2016a). Table 10 summarizes relevant dimensions for monitoring SLAs.

Dimension	Description
Measurement Metric	Standard of measurement that defines the conditions and the rules for performing the measurement of an SLI and for understanding the results of a measurement.
Measurement Scope	The level of granularity at which a metric is observed.
Measurement Responsibility	The distinction by which party is responsible for monitoring and measuring the metrics.
Measurement Boundaries	Distinction between measuring the internal resources of a cloud service versus measuring how the service is perceived by external users.
Measurement Timing	Distinction between measuring before or after an event.
Measurement Invasiveness	Distinction between the general technique used to obtain data.
Measurement Authenticity	Distinction between measuring real-life behavior of the cloud service or relying on testing systems.

Table 10 – Summary of dimensions for monitoring

3.4.1 Metric to Measure Adherence to Commitments

A metric is a standard of measurement that defines the conditions and the rules for performing the measurement of an SLI and for understanding the results of a measurement. Metrics thus determine how to measure an SLI. A metric can be characterized in terms of its relevant parameters, expressions (e.g., calculating formulas), and measurement rules to measure a specific SLI (ISO/IEC, 2016a).

The metrics can be used to determine whether a measurement of a cloud service characteristic at a specific point is within stated boundaries of the SLI. Figure 4 illustrates the measurement process. Metrics in practice are usually described using natural languages, typically in “plain English,” which is often difficult to understand, compare, and implement.

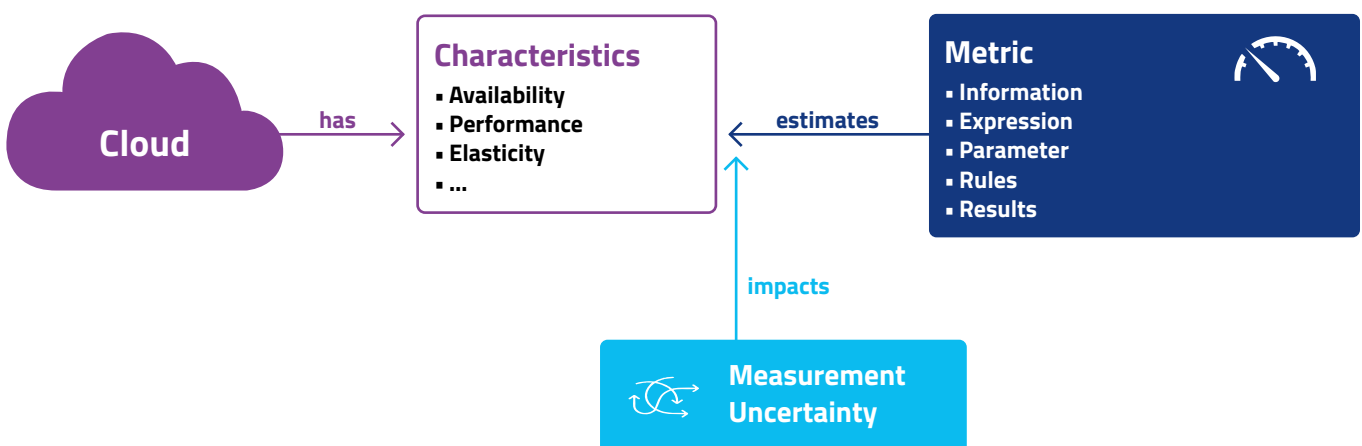


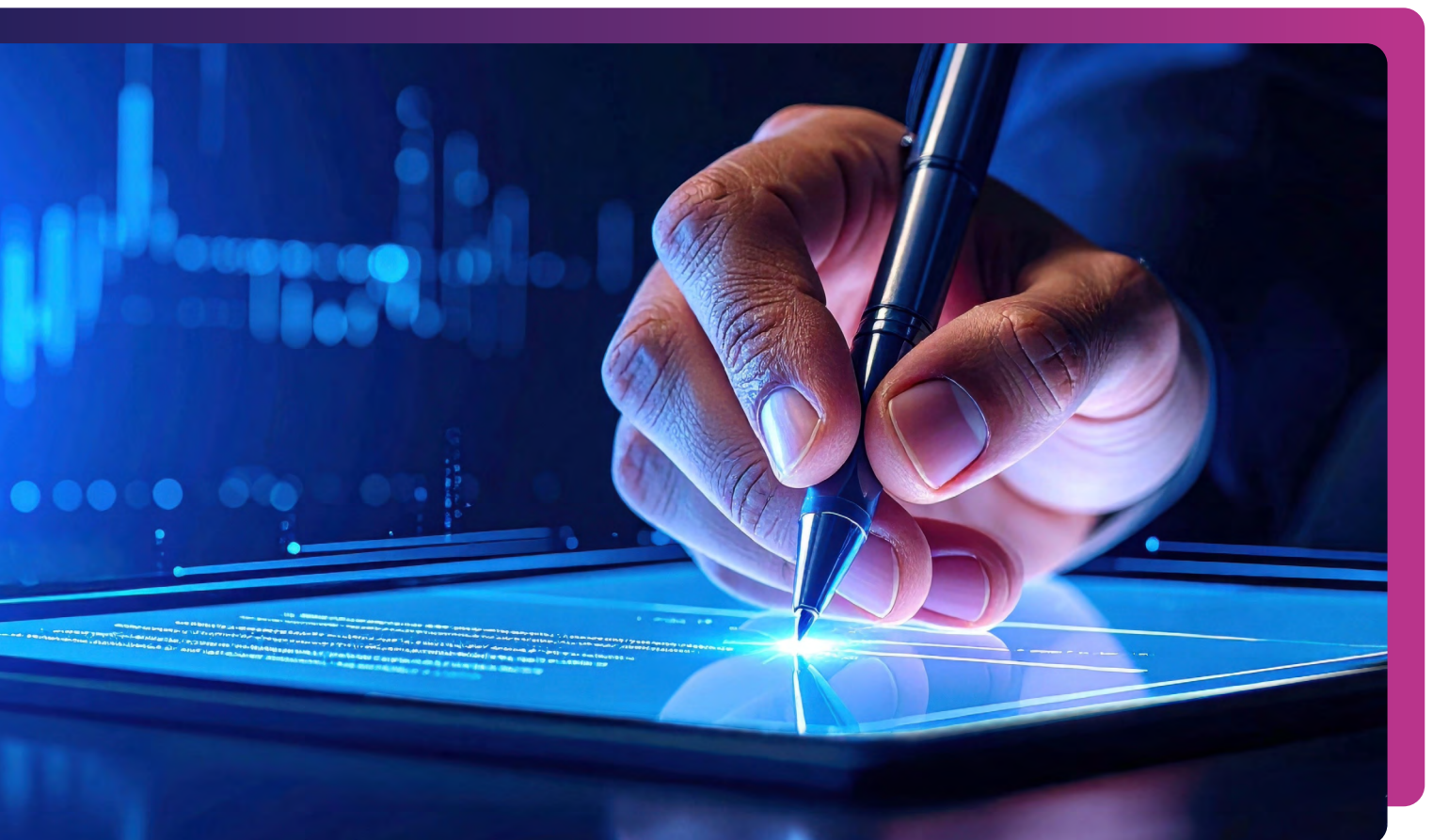
Figure 4 – Illustration of the measurement process.

A metric can generally be described by the characteristics summarized in Table 11. We recommend that cloud providers adhere to this metric blueprint to increase SLA transparency.

Each metric comprises meta-data, including its name and corresponding SLIs and SLOs (ISO/IEC, 2016b). Most importantly, a metric determines the concrete measurement in form of an expression, equation, or method of the calculation. Further supporting information to ensure consistent measurement may be included. A metric covers quantitative or qualitative parameters, attributes, data, or units that relate to the cloud service characteristic and are used to perform the calculations based on the expression. A metric should also define measurement results that need to be applied to ensure reliable results. Various guiding principles for such rules can be considered, including repeatability, reproducibility, and comparability of measurements and measurement results (refer to ISO/IEC, 2016b).

Each measurement leads to a result that, however, is only an estimate of a characteristic that is being observed.

Hence, a certain degree of uncertainty between the approximation and the actual value of the characteristic remains (ISO/IEC, 2016b; TM Forum, 2014). Providers may consider measurement uncertainty in their SLAs in several ways. Generally speaking, measurement uncertainty characterizes the dispersion of the quantity values assigned to a metric (de Vault et al., 2018). It thus resembles the degree of doubt about the accuracy or precision of a metric. The NIST argues that “the result of a measurement is not meaningful if a statement of the uncertainty of the measurement is not specified” (de Vault et al., 2018, p. 22). Measuring metrics may involve sampling errors (e.g., missing short downtimes), clock synchronization issues across distributed systems, or delayed logging or data loss; ultimately reflecting limitations in measurement tools, methods, or conditions. Providing information on measurement uncertainty enables customers to assess the quality of the measurement results and to build confidence to compare results and use them within the range of the measurement uncertainty (de Vault et al., 2018). Information on measurement uncertainty may cover standard deviations or confidence intervals (e.g., MTBF = 300 hours \pm 10 hours with 95% confidence).



Characteristic	Description	Example
Information	Information and meta-data about the metric.	<ul style="list-style-type: none"> Name of the metric, e.g., Mean Time Between Failures (MTBF) Assigned SLI, e.g., operation reliability percentage
Expression	The expression, equation, or method of calculation and supporting information.	<ul style="list-style-type: none"> $MTBF = \frac{Total\ Operational\ Time}{Number\ of\ Failures}$
Parameter	A set of specific parameters that relates to the cloud service characteristic and is used to perform the calculations based on the expression.	<ul style="list-style-type: none"> Total operational time Number of failures
Rules	Constraints to which a measurement conforms and to ensure reliable results.	<ul style="list-style-type: none"> Define what counts as a failure Exclude planned downtimes Define the observation period (e.g., hours, days)
Measurement Results	An approximate measurement result of a cloud service characteristic based on the metric.	<ul style="list-style-type: none"> If a system runs for 1,000 hours and experiences 4 failures during that period: MTBF = 250 hours
Measurement Uncertainty	Information that characterizes the dispersion of the quantity values assigned to a metric and reflects the degree of doubt about the accuracy or precision of a metric.	<ul style="list-style-type: none"> If availability is checked every 5 minutes, a 2-minute failure may be missed. Missing failure events due to log corruption or dropped events reduces reliability of MTBF calculation. If components have clock drift, the start/end of failure intervals may be wrongly recorded.

Table 11 – Characteristics of the metric dimension, adapted from ISO/IEC 19086-2 (2016b)

Example metrics (including expressions and parameters):

Availability:

- **Uptime Percentage.** Percentage of the total time a service is available.

$$\text{Uptime percentage} = \frac{\text{total service time} - \text{total time not available}}{\text{total service time}}$$
- **Maximum allowed downtime.** The total permissible downtime within a given period. For instance, *Maximum allowed downtime 43.2 minutes per month < total service downtime per month*
- **Rejection Probability:**

$$RJ = 1 - (1 - RJC) * (1 - \sum_{0 < x < |Q|} \pi_x (RJQ(x) \wedge RJH(x))^{\bar{g}}$$
 with RJC being the request rejection probability at the CMU handling phase, \bar{g} the expected number of jobs that one request generates, and RJQ(x) and RJH(x) binary indicators for rejection based on insufficient queue space or hot machines (Xia et al., 2013)

Performance:

- **Response Time.** Time taken for the system to respond to a user request.

$$\text{Response Time} = \text{Timestamp when the client receives the full response} - \text{Timestamp when the client sends the request}$$
- **Transaction Throughput.** Number of successful transactions processed per second.

$$\text{Transaction Throughput} = \frac{\text{Number of Transactions}}{\text{Time period in s}}$$

Capacity:

- **Maximum concurrent users supported.** The maximum number of users or sessions the system can support simultaneously without performance degradation. For instance, *5,000 concurrent users during peak hours with no more than 5% performance degradation*
- **Peak Bandwidth Capacity.** The maximum data transfer rate the service can support during peak usage without degradation. For instance, *1 Gbps of bandwidth during peak usage periods with no packet loss exceeding 0.1%*

Elasticity:

- **Time-to-Set-Up (TTS).** The total time taken to install, configure, and make a system or service ready for use.

$$TTS = \text{Time the system is fully operational and available} - \text{Time setup begins (e.g., installation starts)}$$
- **Time-to-Ramp-Up (TTRU).** The time it takes (after setting up) for the system to reach stable or optimal performance under expected load

$$TTRU = \text{time stable (When the system consistently meets performance targets (e.g., response time, throughput))} - \text{time start (When the system begins being used or tested under load)}$$
- **Configuration Change Management Capacity Change Time.** System Overhead Rate (Xia et al., 2013)

Data Protection & Management:

- **Data Retention Duration.** How long data backups or logs are retained.

$$\text{Retention Period} = \text{time data deletion} - \text{time data creation}$$
- **Data Loss Incidents.** The number of incidents where data loss occurs due to a failure or breach.

The ISO/IEC 2502X-family provides additional information on specific metrics that can be used to measure service quality.

FACIS' Recommendations:

Using a standard set of metrics in cloud SLAs makes it easier and faster to define SLOs and SLIs and compare SLAs from multiple providers.

3.4.2 Properties of Metrics

While the suggested metric blueprint (Table 11) already covers rich information on how to measure SLOs/SLIs, we observed that cloud providers frequently provided additional information on metrics and corresponding measurements in

their SLAs: Information related to the measurement (1) scope, (2) responsibility, (3) boundaries, (4) timing, (5) invasiveness, and (6) authenticity. Figure 5 summarizes these dimensions.

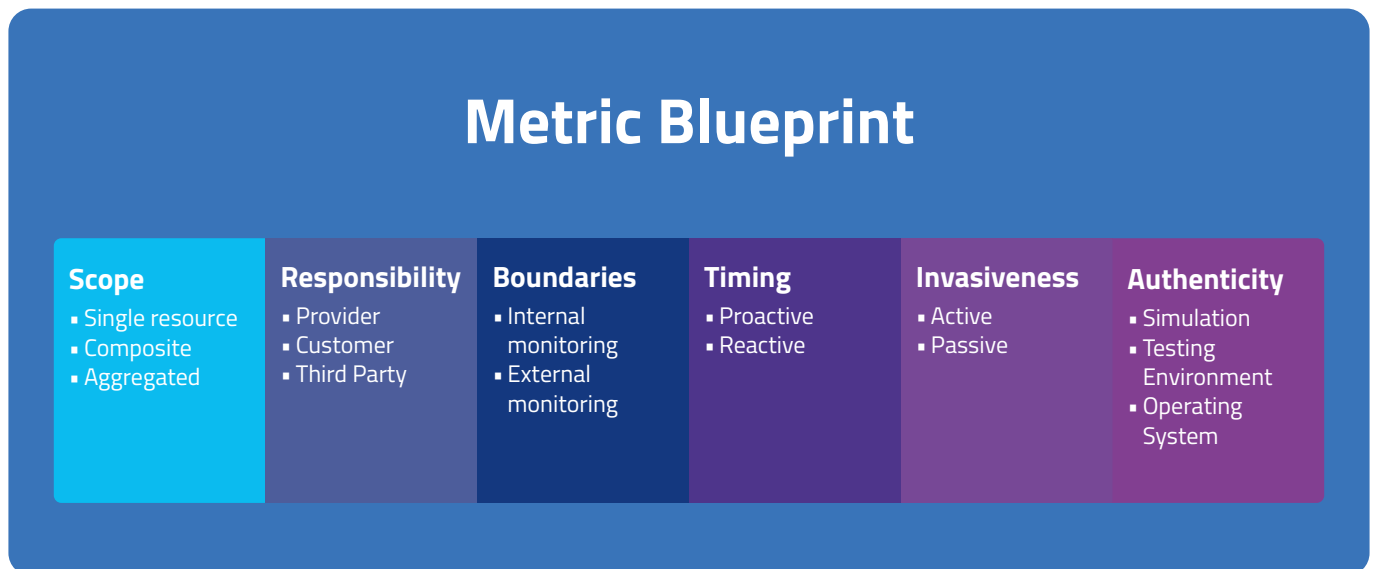


Figure 5 – Additional information for metrics.

First, metrics can be categorized by considering their scope (Aljournah et al., 2015). Measurement scope describes the level of granularity at which a metric is observed; or in other words, the extent of the resources and services being monitored within a cloud environment. It determines whether the metric is focused on individual resources or combinations of them:

→ **Single resource metrics** measure individual resources controlled by the cloud provider, such as the performance or characteristics of servers or applications. For example, CPU usage, memory consumption, or application response time. Single resource metrics offer insights into the behavior and health of individual components like servers, databases, or virtual machines.

→ **Composite metrics** combine multiple single resource metrics controlled by the cloud provider, typically using a specific algorithm or measurement method. This allows for the measurement of more abstract cloud characteristics that result from the interaction of multiple resources. For example, average application response time, which combines the CPU, memory, and network metrics. Or overall system throughput, calculated from individual network and server performance metrics.

→ **Aggregated (end-to-end) metrics** combine single resource or composite metrics from multiple cloud providers. The goal is to provide an end-to-end measurement of service characteristics that are relevant to the cloud customer. For example, end-to-end latency across services hosted by different cloud providers.

How the measurement is conducted for a specific metric can also differ based on further dimensions. We identified five key dimensions that may be considered:

- **Measurement Responsibility.** The responsibility for monitoring and thus measurement of the metrics may reside with either the customer side (Xiaoyong et al., 2015; H. Zhou et al., 2019), the provider side (Hammadi & Hussain, 2012; H. Zhou et al., 2019), or be performed by an independent third party. In the case of third-party observation, one or several observers can be used (e.g., Battula et al., 2022; Binu & Gangadhar, 2014; Hammadi & Hussain, 2012; Taghavi et al., 2019; H. Zhou et al., 2019). Depending on the party that is responsible for the monitoring, the responsibility for remedy claim triggering can be different (see Section 3.5.2).
- **Measurement Boundaries.** A metric can either measure internal resources of a cloud services (internal monitoring) or measure how the service is perceived by external users (external monitoring). Internal monitoring provides reliable results but cannot compensate for any impact outside of the cloud providers’ control. In contrast, external monitoring results are often a better reflection of how a customer actually perceives service quality.

- **Measurement Timing.** Distinction between measuring before or after an event. Proactive measurements aim to anticipate and prevent issues before they occur. Reactive measurements aim to diagnose and fix problems after they have already happened.
- **Measurement Invasiveness.** Distinction between the general techniques used to obtain data, particularly focusing on active vs. passive monitoring. Active monitoring involves actively collecting data by interacting with the system and performing checks on resources. Passive monitoring refers to (neutral and non-intrusive) observation and recording data from resources without active intervention (e.g., through log files or observing user traffic).
- **Measurement Authenticity.** A metric can be further characterized by clarifying whether it measures real-life behavior of the cloud service and thus the operating systems are monitored. Or whether data is gathered on test systems or through simulations.

Table 12 summarizes and classifies measurement examples.

FACIS’ Recommendations:

Aggregated metrics are useful for tracking SLOs and ensuring the overall performance of cloud-based services that depend on resources from multiple providers.

Example	Boundaries	Timing	Invasiveness	Authenticity
Running load tests on test servers.	Internal	Proactive	Active	Test
Simulating user traffic or API calls to anticipate performance bottlenecks.	External	Proactive	Active	Simulation
Monitoring internal resources over time to predict issues.	Internal	Proactive	Passive	Operation
Actively diagnosing internal system issues after a failure.	Internal	Reactive	Active	Operation
Actively analyzing external metrics after a service failure (e.g., network connection issues).	External	Reactive	Active	Operation
Investigating internal logs and metrics after a failure or anomaly.	Internal	Reactive	Passive	Operation

Table 12 – Example for measurements classified according to boundaries, timing, and invasiveness

Concluding FACIS' Recommendations Related to Monitoring

Multi-provider contexts require end-to-end monitoring; therefore, monitoring practices and specified metrics should be aligned and integrated across the cloud service provisioning stack. Using harmonized and standardized SLOs, SLIs, and metrics can ease the comparison and integration of multiple services. We recommend that SLAs should apply the blueprint for metrics shown in Table 11 and illustrated in Figure 4.

Due to the ambiguous and inconsistent nature of how SLOs, SLIs, and metrics are currently described, it is difficult for the customers to have confidence that measurement results are calculated in the same manner as defined in the cloud SLA (ISO/IEC, 2016b). Following ISO/IEC 19086-2 (2016b), we suggest that metrics should fulfill the following quality criteria to foster multi-provider SLA management:

- **Clarity:** A definition of a metric eliminates the ambiguities which currently exist in natural language descriptions. Clarity also comprises a consistent representation of information and that relationships between metrics are made explicit.
- **Composability:** Metrics should be reusable to build composite and aggregated metrics.
- **Comparability:** Following a strict structure for metrics (i.e., the suggested blueprint, Table 11) facilitates their comparison.
- **Applicability:** Metrics need to be usable and implementable, thus bridging the legal SLA document and technical implementations.
- **Precision:** Measurement results are only estimations of a service characteristic. A metric, together with corresponding rules and information that define measurement conditions are needed to ensure a high precision of the metric. Measurement uncertainty should be reduced to a minimum.





In case a multi-party agreement is set, metrics in SLAs may be classified based on their scope. Considering uptime, for instance, an aggregated and thus cross-provider metric for cloud software should communicate that a software function can be used if needed. On a resource level for a specific provider, a metric could be the availability of a database system (in %), as well as metrics related to functional availability. Multi-party SLAs in a federated environment must consider how the performance of one provider affects the overall service. The service's dependence on upstream or downstream services must be clarified. Due to multi-party agreements' dynamic nature (e.g., relationships between providers and their services may change), monitoring becomes especially challenging (Falasi et al., 2013). Therefore, monitoring in federated environments needs to involve not only the actual monitoring activities, but also acquiring all service relationships' runtime information and performing periodic SLA inspections to account for possible changes in any SLAs agreed upon (Falasi et al., 2013).

Research proposes the usage of SLA Management Services involving dedicated Monitoring Agents for each service (i.e., to periodically detect changes in the service's SLA relationships and for monitoring of the service performance regarding agreed upon SLI's) and Monitoring Coordinators (i.e., responsible for collecting dependent SLA evaluation reports, performing SLI analysis to ensure each parent service's SLI are met, and broadcasting changes in established SLAs to all relevant parties) (Falasi et al., 2013).

Applying mathematical approaches is viable for combining and aggregating metrics across services (and their providers), but these require knowledge of the impact of metrics on the end service and which mathematical operations are valid, such as correlation or sum of means (TM Forum, 2014). Alternatively, we recommend that the cloud providers involved can interact with each other. For instance, dashboards can be used for easy access to measurement information in cloud ecosystems. Alerts or warnings can then be generated when SLA metrics move outside the expected boundaries. This approach avoids the need to (mathematically) combine metrics and build composite or aggregated metrics, and is simple in cases when the understanding of performance impacts of one service on another is not well established (TM Forum, 2014).



3.5 Enforcement: Claiming Compensation in Case of (Non-)Adherence to Commitments

The SLA should determine what happens in case a provider does not adhere to its commitments. SLA compliance is often directly linked to remedies or penalties – such as service credits, compensations, or contract adjustments – in the event of violations. Comparing prevalent SLAs in the cloud market reveals

that the enforcement of SLA typically covers three key building blocks: (1) gathering information on non-adherence, (2) issuing a claim, and (3) deciding to award a remedy. Figure 6 summarizes this enforcement process and Table 13 summarizes related dimensions.

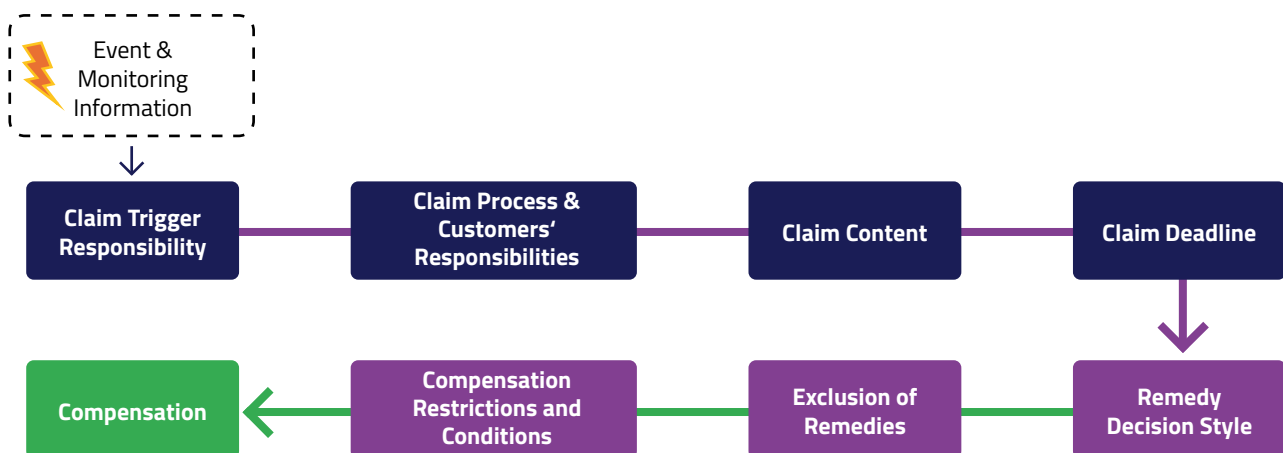


Figure 6 – Summary of enforcement process and related dimensions.

Dimension	Description
Reporting Audience	Who will be informed about SLA compliance.
Reporting Method	How the audience will be informed about SLA compliance.
Reporting Period	The timeframe over which the service quality will be measured and corresponding results then reported to the audience.
Remedies	Compensation for the customer if the cloud provider fails to meet a specified SLO.
Claim Responsibility	Determining who is responsible for identifying cases of non-adherence and initiating a remedy process.
Claim Process	Defining how an actor can initiate a claim for compensation.
Responsibilities of Customers when Claiming	The duties and tasks that are required by the customer in case of claiming remedies.
Claim Content	The information that should be included in the remedy claim to be valid.
Claim Deadline	This defines a time period during which a remedy must be claimed.
Remedy Decision Style	Information on how the provider decides whether a remedy is valid or not.
Exclusion of Remedies	Some events or incidents may be excluded from liability and declared exceptions. In those cases, even if an SLO or SLI is not met, the related remedy will not be triggered.
Remedy Compensation Restrictions and Conditions	SLA may impose certain requirements and restrictions for issuing remedies, like limiting the total amount of compensation granted by the cloud provider.

Table 13 – Summary of dimensions for enforcement

3.5.1 Service Level Objective Compliance Reporting

Reporting plays a crucial role in SLA enforcement by providing transparent, (objective) evidence of whether agreed service levels are being met or not. Accurate and timely reporting not only builds trust between providers and customers but also serves as the factual basis for enforcing SLA terms. Without effective reporting mechanisms, it becomes difficult to validate breaches or trigger contractual remedies, weakening the enforceability of SLAs.

Since multiple parties may be involved in the service provisioning (see Section 3.2.3), the SLA should first settle who will be informed about SLA compliance. Reports may address different actors as audience, including providers’ internal employees (e.g., internal auditors, SLA officers), customers, end users, sub-providers, the general public, regulatory bodies, and related institutions.

Service providers can use different forms of reporting (non-) adherence to SLAs. Reporting formats include tickets, reports to customers, or public websites that, for example, show current service status (e.g., OVHcloud status website), email, text messages, telephone, and social media posts. For instance, Rackspace posits that “Rackspace will notify you via text message, email, or the Rackspace ticketing system of Rackspace Monitoring Alerts on your DB Server(s) within five (5) minutes of the Rackspace Monitoring Alert being generated.”

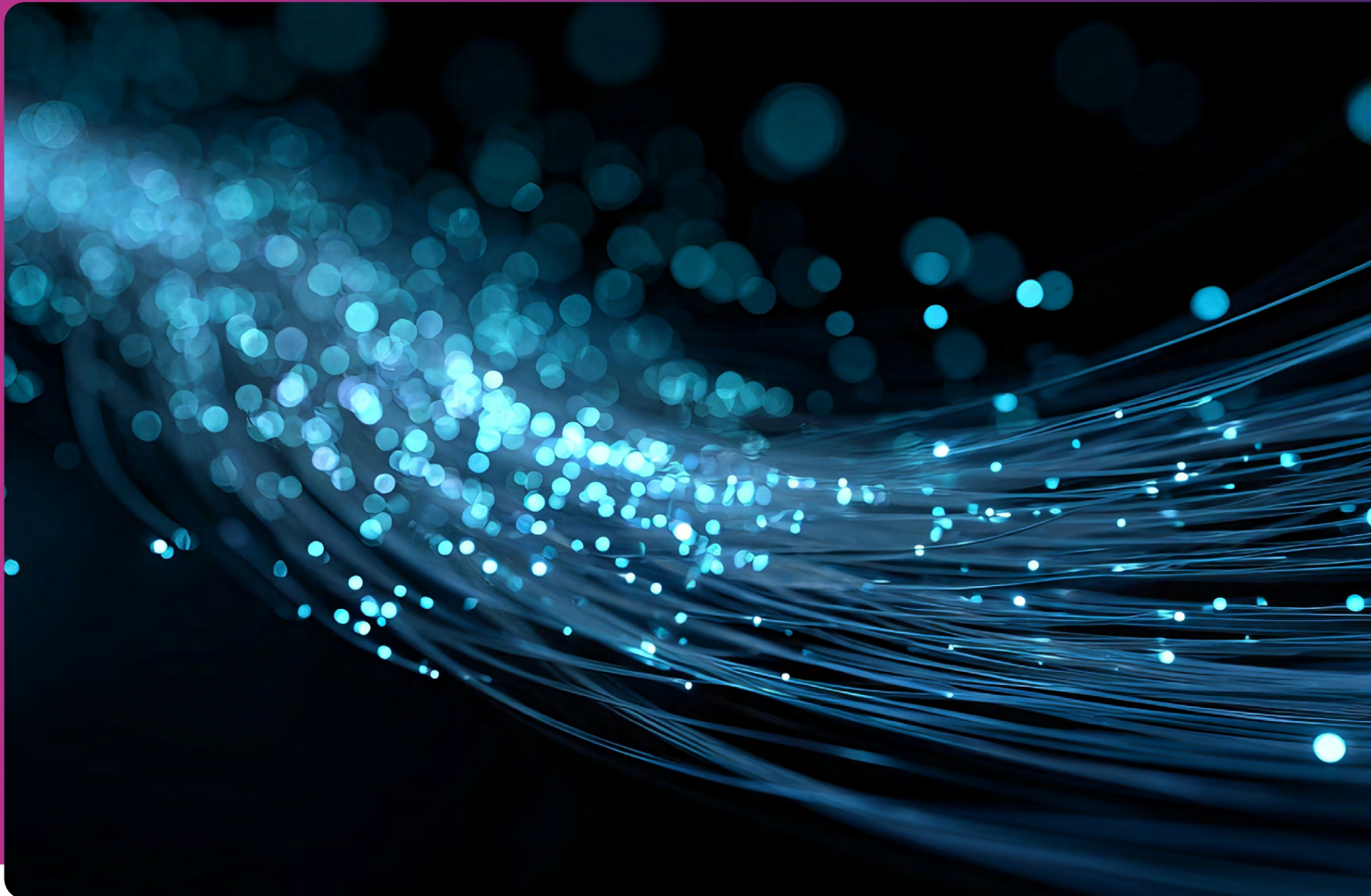
Essential to deciding on (non-)adherence to SLO commitments is the reporting period, which defines the time frame over which service quality will be measured, evaluated, and the results then reported.

Reporting periods include hourly, daily, monthly, quarterly, and yearly, or in case an event or incident happened. Most commitments made are subject to a specific reporting period, which will be applied to decide on non-adherence and, thus, for the validity of remedy claims.

FACIS' Recommendations:

If customers provision multiple services from several providers, it becomes challenging for them to keep track and manage the reporting. As some reports are scheduled, but others are generated automatically when any unprecedented event occurs, reporting time intervals have to be aligned (M. Singh et al., 2021). Thus, in multi-provider environments, reporting times can be standardized or at least compatible with all providers involved.

If standardized reporting periods are not available, reporting periods may be aligned by transforming shorter reporting periods into longer periods via calculations. In entangled supply chain scenarios, lead cloud providers may demand the establishment of a reporting chain from involved sub-providers to ensure that the lead provider gets informed about SLO compliance across the entire cloud stack.



3.5.2 Remedy Claims

Remedies may be provided by the provider to the customer in the event the cloud service fails to meet the SLOs or associated SLIs for their respective reporting period (ISO/IEC, 2016a). Remedies for failure to achieve SLOs stated in the cloud SLA may take different forms, such as re-funds on charges, service credits (e.g., financial compensation for downtime or poor performance), or other forms of compensation. They can include termination rights (e.g., conditions under which a customer can terminate the agreement due to repeated SLA breaches). For instance, Scaleway agrees in their SLA: "The compensation shall take the form of a voucher which shall be applied by Scaleway on the invoice issued for the Public Cloud Resources impacted, following the month during which the Client's request was issued." Similarly, Google issues service credits: "If Google does not meet the Google Workspace SLA, and if Customer complies with the requirements under this Google Workspace SLA, Customer will be eligible to receive the Service Credits described below." Remedies may also include service termination rights on the side of the provider, for example, in case service customers do not pay license fees (Tan et al., 2022).

Importantly, the SLA needs to determine a remedy claim process in case of non-adherence (ISO/IEC, 2016a). In most SLAs, the customer has to determine when the cloud service has failed to meet its SLOs and then report claims to the provider (Xiaoyong et al., 2015; H. Zhou et al., 2019). Thus, although cloud providers do actively monitor their services and even inform their customers about outages, most of them demand that customers initiate a claim process by themselves to receive remedies like service credits. For instance, Cloudflare states: "To submit a Claim, Customer must contact Customer Support and provide notice of its intention to submit a Claim." Similarly, Kamatera states: "Submit KAMATERA's customer support with a written claim regarding any incident of which it is believed that compensation is applicable (the 'Claim')."

The claiming processes differ in SLAs and are characterized by different levels of automation. Some automated processes only need to be initiated by the customer, while others require contacting the account manager, writing an email or support ticket, or filling out compensation forms. For instance, Tencent states: "To receive Compensation for an Incident, you must submit a claim to Tencent (a „Compensation Claim“). A Compensation Claim can be submitted via your Account (the

„Compensation Claim Form“):

To be able to initiate a claim, some SLAs require further duties and tasks for customers. Among other things, we observed that SLAs require customers to pay all (previous) invoices, cooperate in good faith, support resolution of remedy process, provide additional information, and engage in verification of the incident. For instance, Scaleway requests that “(i) the Client shall have paid all invoices issued by and due to Scaleway in connection with the Services, [...] and (iii) the Client shall cooperate in good faith with Scaleway to resolve the issue, and in particular: by remaining available during the entire period necessary for resolving the issue encountered; by providing all the information reasonably available to the Client regarding the issue encountered; by carrying out any verification that may reasonably be necessary to resolve the issue encountered.” Similarly, Oracle states that: “You must continue to be in compliance with the Oracle Cloud Services agreement referenced in Your order for You to be eligible to receive Service Credits.” Wasabi requests: “You must be current on all payments to Wasabi in order to be eligible for Service Credits.”

Notably, SLAs frequently determine the exact content of customers’ claim; otherwise, a claim will be regarded as invalid. Our synthesis of SLAs reveals that a claim should typically contain information about:

1. the covered and impacted services,
2. the incident leading to non-adherence,
3. evidence for a failure to meet the commitments (e.g., log files),
4. timing and duration of the incident, and
5. applied resolution means by the customer.

For instance, Alibaba requests that the “claim must include at least the following information: (a) A detailed description of the incident, including the logs or messages for request failure documenting the errors and claimed outage; (b) The date, time and duration of the Downtime; (c) Information relating the affected instances, including the affected instance IDs; and (d) Any other information that we reasonably ask you to provide to support your claim.”

If customers themselves must prove non-adherence to SLOs, concrete evidence such as log files and similar documentation is demanded. For some SLOs that can be affected by forces outside of the provider’s control (e.g., latency), the customer needs to provide evidence for the provider’s responsibility for non-adherence (e.g., customers must provide a sufficient Internet connection on the customer’s side).

For instance, Cloudflare states:

“In order for Company to consider a Claim, Customer must submit the Claim, including sufficient evidence to support the Claim, by the end of the billing month following the billing month in which the Incident which is the subject of the Claim occurs.”

Finally, there might be a limited timespan when customers can claim remedies. While those claim deadlines varied greatly, we observe that customers have on average about 1 month to submit a claim. The shortest deadline in our SLA sample was 5 days (Cloudflare SLA), and the longest timespan was 90 days (DigitalOcean Kubernetes SLA).

In other, more rare SLA cases, the provider or an independent third party is able to monitor the service levels and automatically initiate claims (Hammadi & Hussain, 2012; H. Zhou et al., 2019). For instance, Vultr also informs customers in case of non-adherence: “In the event that Vultr discovers (either from its own efforts or after being notified by You) that You are experiencing packet loss [...] and You notify Vultr via a support ticket (or Vultr has notified You of an event), Vultr will take all actions necessary to determine the source of the Excess Packet Loss/Latency.”

In general, the customer should trust the provider with monitoring and remedy claiming (H. Zhou et al., 2019). A possibility to solve trust issues and remove the burden of monitoring and remedy claiming from the customer is to delegate monitoring and claim triggering by one or several third-party observers (e.g., Battula et al., 2022; Binu & Gangadhar, 2014; Hammadi & Hussain, 2012; Taghavi et al., 2019; H. Zhou et al., 2019). While a single third-party observer is already helpful with real-time assessment of service quality (Hammadi & Hussain, 2012), using multiple observers aids in distributing the trust from a single party. The number of observers and their payment can be up to negotiation between the service customer and provider, depending on the trustworthiness wished for monitoring and claim triggering, and the willingness to pay for more observers (H. Zhou et al., 2019). Observers need to be chosen carefully and at random to avoid collusion and observer auditing processes are necessary (H. Zhou et al., 2019). Blockchain solutions and smart contracts prove useful in monitoring and remedy claims triggering by multiple observers.

Besides remedies for non-adherence on the provider’s side, there may also be remedial actions from the providerside if customers do not adhere to commitments made (e.g., customer responsibilities). While non-adherence to SLOs is mostly on a financial basis, remedial actions for non-adherence to customer responsibilities may be harsher, including temporarily disabling service functionality (e.g., (non-)torrenting of illegal activities) or even service termination.

3.5.3 Decision to Award Remedies

Besides detailing the claim process, the SLA may also cover information about how a provider decides on them. Most providers argue that they will decide based on good faith or best-effort approach. For instance, Cloudflare states: “Company will use all information reasonably available to it to validate Claims and make a good faith judgment on whether the SLA and Service Levels apply to the Claim.” Additional information on the decision process may be included, like the deadline until which a decision is made. OVHcloud states, for instance: “OVHcloud will issue the Service Credit to Customer within forty-five (45) calendar days in which the request is confirmed by OVHcloud.” Similarly, Oracle promises: “Oracle will use commercially reasonable efforts to process a claim within sixty (60) days of Oracle’s receipt of such claim.”

Sometimes SLAs explicitly state that their decision to reward or not reward any remedies is binding and cannot be disputed by customers. For instance, Alibaba states: “You agree that any decision or determination made by us relating to your claim for any Service Credit shall be final and binding on you.” Awarding a remedy like service credit also often excludes any further liabilities of providers. For instance, oneclick™ states: “All other warranty, liability and compensation claims of the Customer against oneclick in connection with the operation of the Platform, regardless of the title on which they are based, are excluded, subject to mandatory applicable statutory provisions. [...] All claims of the customer – of any kind whatsoever – arising from the restrictions of the guaranteed availability are settled with the credit note.”³

Most importantly, SLAs list exclusions from remedies by describing the circumstances under which the SLOs and their associated remedies do not apply (ISO/IEC, 2016a). These may vary between agreements and be subject to the laws of a particular jurisdiction. Examples of exceptions include scheduled outages (e.g., for maintenance), force majeure (e.g., natural disasters), improper use from the customer, features designated pre-general availability, being attacked, and other factors beyond the control of the service provider. Interestingly, Kamatera even argued to reject claims for any other reason: “To remove any doubt AND NOTWITHSTANDING ANY OF THE ABOVE, KAMATERA reserves the right to reject any CLAIM FOR compensation OR CREDIT to the Customer under this SLA, at KAMATERA’s sole discretion and for any reason.”

Once providers have decided to award a remedy, they can impose additional restrictions and conditions that are agreed on in the SLA. We observed that restrictions often relate to limiting the amount of compensation (e.g., 50% of the monthly service fee), forbidding cumulative compensation (e.g., credits are only awarded per incident, even if multiple services were affected), not considering snowballing effects (e.g., if a database server was unavailable and thus a software service as well, only the database unavailability will be rewarded), binding remedies to specific accounts or users, or limiting the amount of claims for a specific time period (e.g., only one claim per month). For instance, Tencent states: “The Compensation provided to you [...] for any particular Service in any given calendar month will not, under any circumstance, exceed the Fees paid and payable by you for that particular Service in that calendar month. [...] If more than one Service Level is not met because of an Incident, you may choose only one Service Level under which to make a claim based on that Incident.”

Remedies are most often only to be offset against future payments and to be made by the customer. Sometimes, those remedies are automatically applied to the next invoice or must be applied for by the customer at a given time before they are withdrawn. For instance, OVHcloud states that “Service Credits expire on the earlier to occur of: (i) the expiration or termination of Customer’s Account, or (ii) twelve (12) months from the date the Service Credit is issued by OVHcloud to Customer.”

Concluding FACIS’ Recommendations Related to Enforcement

In entangled supply chains, the risk of blame chains may complicate or even undermine customers’ ability to claim rewards. SLAs that we analyzed mostly excluded any remedies that are based on incidents caused by sub-providers involved in the service provisioning.

In contrast, cloud ecosystems may counteract the risk of blame chains because customers have single agreements with providers. Still, issuing claims remain challenging in multi-provider set-tings. Remedies, such as service credits, are usually not credited automatically; they must be actively claimed by the customer. We therefore suggest introducing standardized processes for claiming service credits and other forms of compensation and providing a clearly responsible party for the customer. Furthermore, remedies should be harmonized across service providers.

³ Statements from oneclick’s SLA were translated into English by the FACIS team.

3.6 Automation: Fostering Machine-Readability of SLA

Automating SLA management can increase efficiency, but requires that SLAs are represented in a machine-readable manner. Over the last decades, several initiatives have developed SLA languages, ontologies, and related frameworks to achieve automation and machine-readability. Recent advancements

in distributed ledger technology also offer advanced concepts like smart contracts that may be used for SLA automation (e.g., Battula et al., 2022; Tan et al., 2022; H. Zhou et al., 2019). From a legal perspective, the automation of SLA is in line with regulatory requirements and therefore allowed. Nevertheless, automation requires additional mechanisms, like approved techniques for digital signatures. Regarding automation, we identified several languages that can be further characterized by their degree of formalization, maturity, generalization, and tool support (Table 14).

Dimension	Description
Machine-readable Language	A language, ontology, or related mechanism for transferring SLAs into machine-readable representations.
Focus	The SLA management lifecycle phases a language (not) supports.
Formalization	The degree of how precisely and unambiguously an SLA language defines semantics, syntax, and logic.
Automation	The degree to which the SLA language enables the automation of SLA management.
Maturity	An indication about how well a language is applicable and already used in practice.
Generalizability	The extent to which the language is restricted to a specific application domain/context or not.
Tool Support for Language	Determine whether and what type of tool is available to use the language for SLA modelling.

Table 14 – Summary of dimensions for automation

3.6.1 SLA Languages

Several SLA languages and ontologies exist, either stemming from initiatives on SLAs for web services or cloud services in particular (Alqahtani et al., 2019; Maarouf et al., 2015). Those languages differ in their focus, such as supporting agreement negotiation, monitoring and enforcement, or identifying SLA violations. Prominent SLA languages include:

- **Web Service Level Agreement** (WSLA; (Keller & Ludwig, 2003)): A language introduced by IBM research as a framework for the definition and monitoring of SLAs for web services. The WSLA framework provides a formal language, an XML schema, and a run-time architecture that interprets the language and handles SLA management tasks.
- **Web Service Agreement Specification** (WS-Agreement; (Open Grid Forum, 2007)): WS-Agreement is an SLA specification presented by the Grid Resource Allocation and Agreement Protocol Working Group of the Compute Area of the Open Grid Forum. It is an XML-based language and web service protocol.
- **Service Level Agreement Language** (SLAng; (Lamanna et al., 2003)): SLAng is under development in the Department of Computer Science at the University College London. It is a language for concrete service level agreements currently providing support for ASP SLAs.
- **Service-Level-Agreement Language for Cloud Computing** (SLAC; (Uriarte et al., 2014)): A language to define a tailored SLA for the cloud domain. Including a syntax of the language as well as the semantics to check the conformance of SLAs.
- **Rule-based Service Level Agreement Language** (RBSLA; (Paschke, 2005)): Based on RuleML. With this language, SLAs can be implemented in a machine-readable syntax which can be fed into a rule engine to monitor the contract performance at run-time and automatically execute the contractual rules.
- **SLA-Star Ontology** (SLA*; (Kearney et al., 2010)): An abstract and domain-independent syntax for machine-readable SLAs and SLA templates, which can be represented by any syntactic format, such as XML or OWL.
- **Cloud SLA** (CSLA; (Kouki et al., 2014)): The CSLA language consists of three sections: the validation period of the agreement, the parties of the agreement, and a reference to the template used to create the agreement.
- **Language for SLA Specification and Monitoring** (SLALOM; (Correia et al., 2011)): SLALOM synthesizes common SLA concepts by composing the BPMN metamodel with that of the SLA life cycle as described in ITIL. The derived metamodel expresses the SLALOM abstract syntax model.
- **Ontology of Secure Service Level Agreement** (SSLA; (Lee et al., 2015)): Ontologies for security SLAs can be used to understand the security agreements of a provider, to negotiate desired security levels, and to audit the compliance of a provider with respect to federal regulations (such as HIPAA).
- **Semantically Rich Framework to Automate Cloud Service Level Agreements** (Natolana Ganapathy & Joshi, 2022): A framework to automate the process of extracting knowledge embedded in cloud SLAs and representing it in a semantically rich knowledge graph helping the user to make a calculated decision in choosing a provider. The framework captures the key terms, measures, and deontic rules, in the form of obligations and permissions present in the cloud SLAs.
- **A Service Level Agreement Language for Cloud Services** (rSLA; (Tata et al., 2016)): Language for specifying and enforcing SLAs for cloud services, allowing for dynamic instrumentation of heterogeneous cloud services and instantaneous deployment of SLA monitoring.

FACIS' Recommendations:

It must be determined whether any of these languages is already used in practice and how well they are suited to consider the specifics of multi-provider settings.

3.6.2 Assessing SLA Languages: Formalization, Automation, Maturity, Generalizability, and Tool Support

Each of those languages comes with opportunities and limitations. We identified six key dimensions that support in assessing SLA languages:

1. **Automation Focus.** The SLA management lifecycle consists of several phases that are closely linked to the lifecycle of cloud workflows (Battula et al., 2022; X. Liu et al., 2011). The phases are (1) SLA definition, (2) SLA establishment, (3) SLA Monitoring, (4) SLA execution, (5) SLA Reporting (Battula et al., 2022). In the SLA definition phase, the SLA negotiation takes place, and commitments are defined – consisting of,

for example, customer and provider ID, resource type and configuration (e.g., CPU, GPU, RAM, storage), penalties, and contract duration – which are then transformed into templates (e.g., in the form of JSON files) in the SLA establishment phase. SLA monitoring should be performed continuously to identify violations of the commitments. In case of violations, in the SLA execution phase, the agreed-upon actions take place to enforce the SLA. In the reporting phase, SLA evaluation, penalty calculation, and billing take place. Some frameworks and languages only support specific phases of the SLA management lifecycle. For example, the blockchain smart contract-based automation frameworks by Battula et al. (2022), Zhou et al. (2019), or Tan et al. (2022) only cover phases (2)-(5).



2. **Degree of Formalization.** The degree of formalization refers to how precisely and unambiguously an SLA language defines semantics, syntax, and logic. The degree of formalization may range from completely unstructured documents written in natural language to formal specification (Girs et al., 2020). The degree of formalization can be classified in three different categories: (i) unstructured, (ii) semi-structured, and (iii) formal.

- Unstructured: No structure is provided to write the template. The SLA document is written in natural language in ad-hoc manner.
- Semi-structured: A limited structure, in the form of a template for example, is used to describe the SLA. Key elements that the SLA should contain are listed. However, limited information (if any) is given on how those elements should be specified. Generally, the key elements are described in a natural language.
- Formal: A clear structure is provided, and the elements are formally described. Several methods might be used to formally describe an SLA, such as using an ontology, or a mathematical notation (e.g., statecharts, Petri Nets). In addition to syntactical specification, semantics and/or behavioral specification might also be provided.

3. **Degree of Automation.** The degree to which the SLA language enables the automation of SLA management phases, typically distinguishing partial vs. full automation (cf. Battula et al., 2022; H. Zhou et al., 2019).

4. **Level of Maturity** (Girs et al., 2020). Maturity is an indication about how well a language is applicable and already used in practice. Since some of those languages stem from research projects or may be outdated, their level of maturity should be critically assessed. For example, SLAng is under development and should not yet be used for real SLAs.⁴ The level of maturity of the language can be assessed, for instance, by using the maturity classification in the Redwine-Riddle model (Redwine & Riddle, 1985). According to this classification, a language can be considered:

- Not mature at all: If basic ideas of the language are presented but there is no proof-of-concept.
- Somewhat mature: If the language is thoroughly discussed together with a proof-of-concept and the usability of the language is demonstrated in use cases.

- Mature: If the language is thoroughly discussed together with a proof-of-concept, the usability of the language is demonstrated in use cases, and the language is used or adapted by the research community or organizations.
- Inconclusive: If the language cannot be categorized with respect to the previous three characteristics.

(Girs et al., 2020) analyze a set of languages and ontologies. Their results indicate that most of them were not mature enough to be used in practice.

5. **Generalizability** (Girs et al., 2020). The language can also be analyzed with respect to the application domain. A language could be generally applicable within the area of web services, contextualized to consider specifics of cloud services, or even tailored to a specific application domain within this area (e.g., e-commerce, transportation, and logistics services).
6. **Tool Support** (Girs et al., 2020). To ensure applicability of the language, it might be valuable to assess whether the language is supported by a tool or not. If yes, it should be distinguished between whether the supported tool is a freeware, open-source, and/or proprietary tool.

Concluding FACIS' Recommendations Related to Automation

A wide variety of initiatives, projects, and parties have already developed and tested various SLA languages, ontologies, and related techniques to automate SLA management. A combination of SLA languages may be needed to support the entire SLA management lifecycle. However, their value and maturity must be carefully assessed before they should be applied or taken into consideration. Maturity is key to ensuring practical applicability and diffusion. In addition, languages that are not applicable to cloud contexts or do not consider the specifics of multi-provider settings are not applicable in the context of FACIS. We recommend using formal or at least semi-structured languages to foster harmonization and standardization required in multi-provider settings.

⁴ Source: <https://uclslang.sourceforge.net/faq.php#q11>

Conclusion

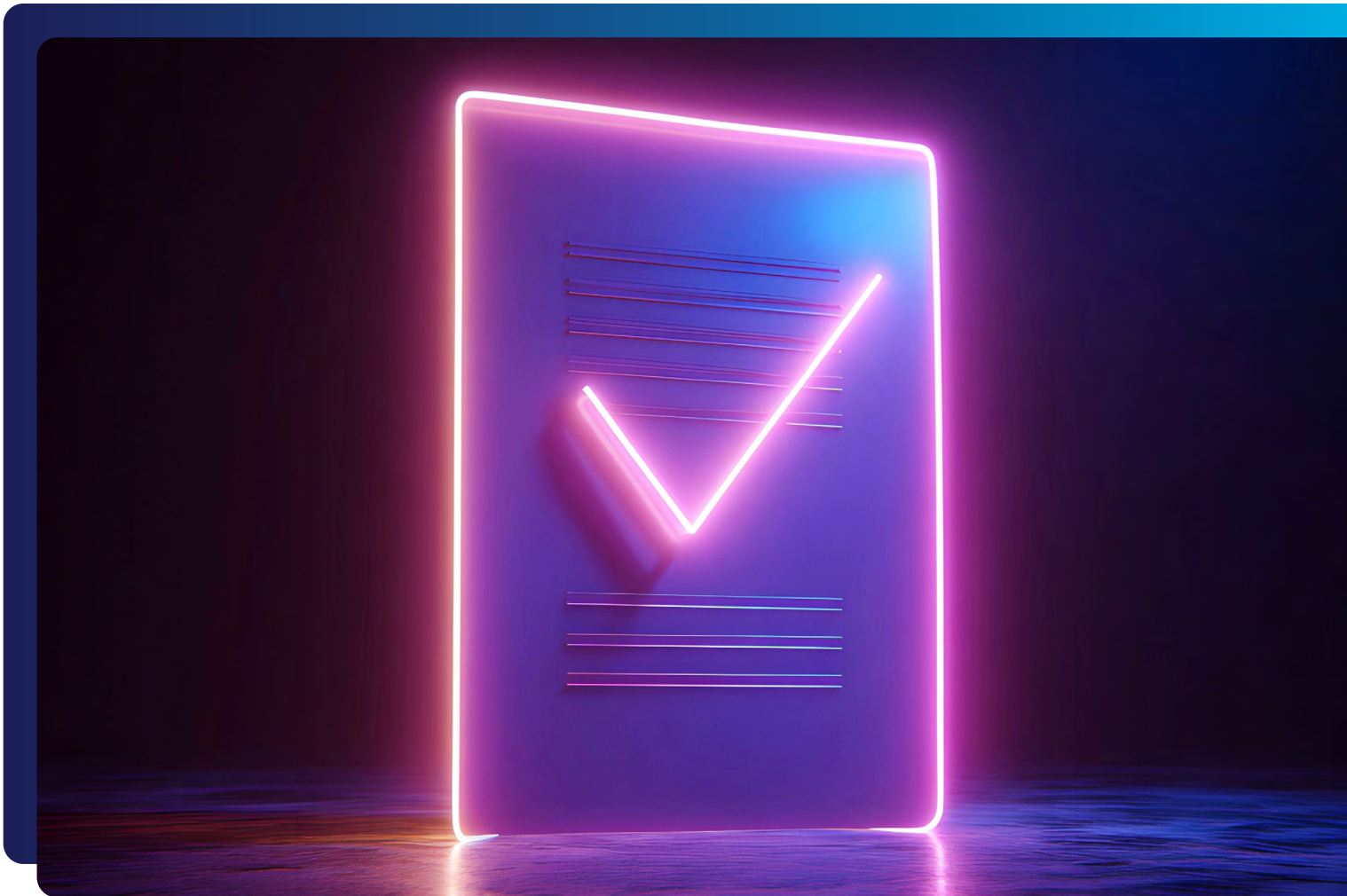


Managing SLAs in multi-provider cloud environments presents significant challenges due to the heterogeneity of services, differing contractual standards, and the lack of unified governance mechanisms. Fragmentation in SLA contents, providers' promises, and enforcement models makes it difficult for customers to ensure consistent service quality and accountability across providers.

Our taxonomy helps address these challenges by identifying the essential building blocks of SLAs. First, it establishes SLA foundations, encompassing the structural and legal frameworks that define roles, responsibilities, and general agreements. Second, it offers a 3-layered hierarchy of SLOs, SLIs, and metrics to support the harmonization of SLA commitments. Third, it

highlights the importance of measurable metrics and monitoring mechanisms to track provider performance. Finally, the taxonomy emphasizes the need for transparent and enforceable remedy and claim processes, which define process steps and compensation in case of SLA violations. By systematically capturing these dimensions and corresponding characteristics, our taxonomy provides a robust basis for SLA design, comparison, and automation in federated and multi-provider cloud settings.

Future work could, for instance, consider dynamic SLAs that automatically adapt to changing conditions or services. Another fruitful avenue for future research could be the integration of SLA into service orchestration, for instance, balancing SLAs among multiple providers to increase effectiveness and save costs.



Appendix: Example Availability Commitments



Service Level Objective (SLO):

The cloud provider commits to ensure that service is **accessible and usable upon demand** by a customer (ISO/IEC, 2016a).

Example Service Level Indicators (SLI):

- Uptime: The amount or percentage of time in a given period that the cloud service is accessible and usable (ISO/IEC, 2016a).
- Availability and accessibility of functionality (e.g., being actually able to perform operations) or data
- MTBF: Mean Time Between Failure. The average time between system failures.
- MTTR: Mean Time To Repair (or recovery/response/resolution time). The average time it takes to re-cover from failure.
- MTTF: Mean Time To Failure. Average time until the first or next failure of a service.
- MTTA: Mean Time To Acknowledge. Average time taken to detect and acknowledge an incident.
- Maximum Outage Duration. The longest single period of service unavailability during a measurement window.
- Packet Loss Frequency (Li et al., 2012): Packet Loss Frequency is defined as the rate between `loss_time_slot` and `total_time_slot`.
- Error Rate: The proportion of failed requests compared to total requests (e.g., HTTP 5xx errors).
- ...

Example Metric Expressions:

- Uptime: may be calculated as the total time over a set of defined intervals less the total downtime during each interval, and may exclude allowable downtime (ISO/IEC, 2016a).
$$\text{Availability (\%)} = \left(\frac{\text{Total Time} - \text{Downtime}}{\text{Total Time}} \right) \times 100$$
- MTBF:
$$\text{MTBF} = \frac{\text{Total Operational Time}}{\text{Number of Failures}}$$
- MTTR:
$$\text{MTTR} = \frac{\text{Total Downtime}}{\text{Number of Failures}}$$
- MTTF: Total operational time before failure.
- MTTA: Time of acknowledgment - Time of failure detection
- Error Rate:
$$\text{Error Rate} = \frac{\text{Failed Requests}}{\text{Total Requests}}$$
- Maximum allowed downtime. The total permissible downtime within a given period. For instance, *Maximum allowed downtime 43.2 minutes per month < total service downtime per month*
- ...

Glossary of Terms



Term	Description	Reference / Literature
Cloud Service Agreement	A documented agreement between the cloud service provider and cloud service customer that governs the covered service(s).	ISO/IEC 19086-1, Nr. 3.3
Cloud Service Level Agreement (SLA)	Part of the cloud service agreement that includes cloud service level objectives for the covered cloud service(s).	ISO/IEC 19086-1, Nr. 3.4
Cloud Service Level Objective (SLO)	Commitments that a cloud service provider makes for a specific, quantitative or qualitative characteristic of a cloud service. Those commitments then become a specific target that needs to be achieved.	Merged from ISO/IEC 19086-1, Nr. 3.5 and 3.6
Service Level Indicator (SLI)	An attribute, parameter, or scale associated with a service that is used to specify or determine a certain quality of the service. SLIs are either of quantitative or qualitative nature and are often expressed in the form of scales. An SLI determines what is measured to fulfill a specific SLO.	(Girs et al., 2020)
Metric	A standard of measurement that defines the conditions and the rules for performing the measurement of an SLI and for understanding the results of a measurement. A metric can be characterized in terms of its relevant parameters, expressions (e.g., calculating formulas), and measurement rules to measure a specific SLI.	ISO/IEC 19086-1, Nr. 3.10
Remedy	Compensation available to the cloud service customer in the event the cloud service provider fails to meet a specified cloud SLO.	ISO/IEC 19086-1, Nr. 3.18

References



- Aljournah, E., Al-Mousawi, F., Ahmad, I., Al-Shammri, M., & Al-Jady, Z. (2015). SLA in Cloud Computing Architectures: A Comprehensive Study. *International Journal of Grid and Distributed Computing*, 8(5), 7–32. <https://doi.org/10.14257/ijgdc.2015.8.5.02>
- Alqahtani, A., Solaiman, E., Patel, P., Dustdar, S., & Ranjan, R. (2019). Service level agreement specification for end-to-end IoT application ecosystems. *Software: Practice and Experience*, 49(12), 1689–1711. <https://doi.org/10.1002/spe.2747>
- Battula, S. K., Garg, S., Naha, R., Amin, M. B., Kang, B., & Aghasian, E. (2022). A blockchain-based framework for automatic SLA management in fog computing environments. *Journal of Supercomputing*, 78(15), 16647–16677. <https://doi.org/10.1007/s11227-022-04545-w>
- Becker, M., Lehrig, S., & Becker, S. (2015). Systematically Deriving Quality Metrics for Cloud Computing Systems. *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, 169–174. <https://doi.org/10.1145/2668930.2688043>
- Binu, V., & Gangadhar, N. D. (2014). A cloud computing service level agreement framework with negotiation and secure monitoring. *2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 1–8. <https://doi.org/10.1109/CCEM.2014.7015474>
- Correia, A., Abreu, F. B. e., & Amaral, V. (2011). SLALOM: A Language for SLA specification and monitoring. <https://doi.org/10.48550/ARXIV.1109.6740>
- De Vault, F., Simmon, E., & Bohn, R. (2017). Cloud computing service metrics description (No. NIST SP 500-307; p. NIST SP 500-307). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.500-307>
- De Vault, F., Simmon, E., & Bohn, R. (2018). Cloud Computing Service Metrics Description (No. NIST Special Publication 500-307). <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-307.pdf>
- Falasi, A. A., Serhani, M. A., & Dssouli, R. (2013). A model for multi-levels SLA monitoring in federated cloud environment. *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, 363–370. <https://doi.org/10.1109/UIC-ATC.2013.14>
- Floerecke, S., Lehner, F., & Schweikl, S. (2021). Cloud computing ecosystem model: Evaluation and role clusters. *Electronic Markets*, 31(4), 923–943. <https://doi.org/10.1007/s12525-020-00419-2>
- Garcia, J. M., Fernandez, P., Pedrinaci, C., Resinas, M., Cardoso, J., & Ruiz-Cortes, A. (2017). Modeling Service Level Agreements with Linked USDL Agreement. *IEEE Transactions on Services Computing*, 10(1), 52–65. <https://doi.org/10.1109/TSC.2016.2593925>
- Girs, S., Sentilles, S., Asadollah, S. A., Ashjaei, M., & Mubeen, S. (2020). A Systematic Literature Study on Definition and Modeling of Service Level Agreements for Cloud Services in IoT. *IEEE Access*, 8, 134498–134513. <https://doi.org/10.1109/ACCESS.2020.3011483>
- Guerron, X., Abrahao, S., Insfran, E., Fernandez-Diego, M., & Gonzalez-Ladron-De-Guevara, F. (2020). A Taxonomy of Quality Metrics for Cloud Services. *IEEE Access*, 8, 131461–131498. <https://doi.org/10.1109/ACCESS.2020.3009079>
- Hammadi, A. M., & Hussain, O. (2012). A framework for SLA assurance in cloud computing. *2012 26th International Conference on Advanced Information Networking and Applications Workshops*, 393–398. <https://doi.org/10.1109/WAINA.2012.280>
- ISO/IEC. (2015). ISO/IEC 25024: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of data quality.
- ISO/IEC. (2016a). ISO/IEC 19086-1: Information technology – Cloud computing – Service level agreement (SLA) framework – Part 1: Overview and concepts (Nos. 19086–1).
- ISO/IEC. (2016b). ISO/IEC 19086-2: Cloud computing – Service level agreement (SLA) framework – Part 2: Metric model (Nos. 19086–2).
- ISO/IEC. (2016c). ISO/IEC 25023: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of system and software product quality.
- ISO/IEC. (2017a). ISO/IEC 19086-3: Cloud computing – Service level agreement (SLA) framework – Part 3: Core conformance requirements.
- ISO/IEC. (2017b). ISO/IEC TS 25011: Information technology – Systems and software Quality Requirements and Evaluation (SQuaRE) – Service quality models.
- ISO/IEC. (2019). ISO/IEC 19086-4: Cloud computing – Service level agreement (SLA) framework – Part 4: Components of security and of protection of PII.
- Kearney, K. T., Torelli, F., & Kotsokalis, C. (2010). SLA*: An abstract syntax for Service Level Agreements. *2010 11th IEEE/ACM International Conference on Grid Computing*, 217–224. <https://doi.org/10.1109/GRID.2010.5697973>
- Keller, A., & Ludwig, H. (2003). The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *Journal of Network and Systems Management*, 11(1), 57–81. <https://doi.org/10.1023/A:1022445108617>
- Kouki, Y., Oliveira, F. A. D., Dupont, S., & Ledoux, T. (2014). A Language Support for Cloud Elasticity Management. *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 206–215. <https://doi.org/10.1109/CCGrid.2014.17>
- Lamanna, D. D., Skene, J., & Emmerich, W. (2003). SLAng: A Language for Defining Service Level Agreements. <https://discovery.ucl.ac.uk/id/eprint/721/1/9.9.6slang.pdf>
- Lee, C.-Y., Kavi, K. M., Paul, R. A., & Gomathisankaran, M. (2015). Ontology of Secure Service Level Agreement. *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*, 166–172. <https://doi.org/10.1109/HASE.2015.33>
- Li, Z., O'Brien, L., Zhang, H., & Cai, R. (2012). On a Catalogue of Metrics for Evaluating Commercial Cloud Services. *2012 ACM/IEEE 13th International Conference on Grid Computing*, 164–173. <https://doi.org/10.1109/grid.2012.15>

- Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). NIST cloud computing reference architecture (No. NIST SP 500-292; 0 ed., p. NIST SP 500-292). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.500-292>
- Liu, X., Yang, Y., Yuan, D., Zhang, G., Li, W., & Cao, D. (2011). A generic QoS framework for cloud workflow systems. 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, 713–720. <https://doi.org/10.1109/DASC.2011.124>
- Maarouf, A., Marzouk, A., & Haqiq, A. (2015). A review of SLA specification languages in the cloud computing. 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA), 1–6. <https://doi.org/10.1109/SITA.2015.7358406>
- Natolana Ganapathy, D., & Joshi, K. P. (2022). A Semantically Rich Framework to Automate Cloud Service Level Agreements. IEEE Transactions on Services Computing, 1–1. <https://doi.org/10.1109/TSC.2022.3140585>
- Nickerson, R. C., Varshney, U., & Muntermann, J. (2013). A method for taxonomy development and its application in information systems. European Journal of Information Systems, 22(3), 336–359. <https://doi.org/10.1057/ejis.2012.26>
- Open Grid Forum. (2007). Web Services Agreement Specification (WS-Agreement).
- Paschke, A. (2005). RBSLA A declarative Rule-based Service Level Agreement Language based on RuleML. International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), 2, 308–314. <https://doi.org/10.1109/CIMCA.2005.1631486>
- Redwine, S. T., & Riddle, W. E. (1985). Software technology maturation. ICSE '85: Proceedings of the 8th International Conference on Software Engineering. 8th international conference on Software engineering, Washington, DC, USA. <https://doi.org/10.5555/319568.319624>
- Şener, U., Gökalp, E., & Eren, P. E. (2024). CLOUD-QM: A quality model for benchmarking cloud-based enterprise information systems. Software Quality Journal, 32(3), 881–920. <https://doi.org/10.1007/s11219-024-09669-1>
- Singh, M., Bhushan, S., & Rani, S. (2021). Investigation of SLA Management in Cloud Computing and Future Directions. 2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST), 78–83. <https://doi.org/10.1109/ICCMST54943.2021.00027>
- Singh, S., Jeong, Y.-S., & Park, J. H. (2016). A survey on cloud computing security: Issues, threats, and solutions. Journal of Network and Computer Applications, 75, 200–222. <https://doi.org/10.1016/j.jnca.2016.09.002>
- Sun, L., Ma, J., Wang, H., Zhang, Y., & Yong, J. (2018). Cloud Service Description Model: An Extension of USDL for Cloud Services. IEEE Transactions on Services Computing, 11(2), 354–368. <https://doi.org/10.1109/TSC.2015.2474386>
- Taghavi, M., Bentahar, J., Otrok, H., & Bakhtiyari, K. (2019). A blockchain-based model for cloud service quality monitoring. IEEE Transactions on Services Computing, 1–1. <https://doi.org/10.1109/TSC.2019.2948010>
- Tan, W., Zhu, H., Tan, J., Zhao, Y., Xu, L. D., & Guo, K. (2022). A novel service level agreement model using blockchain and smart contract for cloud manufacturing in industry 4.0. Enterprise Information Systems, 16(12), 1939426. <https://doi.org/10.1080/17517575.2021.1939426>
- Tata, S., Mohamed, M., Sakairi, T., Mandagere, N., Anya, O., & Ludwig, H. (2016). rSLA: A Service Level Agreement Language for Cloud Services. 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), 415–422. <https://doi.org/10.1109/CLOUD.2016.0062>
- TM Forum. (2014). TR178 Enabling End-to-End Cloud SLA Management V2.0.2.
- Uriarte, R. B., Tiezzi, F., & Nicola, R. D. (2014). SLAC: A Formal Service-Level-Agreement Language for Cloud Computing. 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, 419–426. <https://doi.org/10.1109/UCC.2014.53>
- Xia, Y., Zhou, M., Luo, X., & Zhu, Q. (2013). A comprehensive QoS determination model for infrastructure-as-a-service clouds. 2013 IEEE International Conference on Automation Science and Engineering (CASE), 122–127. <https://doi.org/10.1109/CoASE.2013.6654070>
- Xiaoyong, Y., Hongyan, T., Ying, L., Tong, J., Tiancheng, L., & Zhonghai, W. (2015). A competitive penalty model for availability based cloud SLA. 2015 IEEE 8th International Conference on Cloud Computing, 964–970. <https://doi.org/10.1109/CLOUD.2015.142>
- Zhao, L., Sakr, S., & Liu, A. (2015). A framework for consumer-centric SLA management of cloud-hosted databases. IEEE Transactions on Services Computing, 8(4), 534–549. <https://doi.org/10.1109/TSC.2013.5>
- Zheng, X., Martin, P., Brohman, K., & Xu, L. D. (2014). CLOUDQUAL: A Quality Model for Cloud Services. IEEE Transactions on Industrial Informatics, 10(2), 1527–1536. <https://doi.org/10.1109/TII.2014.2306329>
- Zhou, H., Ouyang, X., Ren, Z., Su, J., De Laat, C., & Zhao, Z. (2019). A blockchain based witness model for trustworthy cloud service level agreement enforcement. IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 1567–1575. <https://doi.org/10.1109/INFOCOM.2019.8737580>
- Zhou, P., Wang, Z., Li, W., & Jiang, N. (2015). Quality Model of Cloud Service. 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, 1418–1423. <https://doi.org/10.1109/HPCC-CSS-ICESS.2015.134>

