Enabling End-to-End Cloud SLA Management

A Collaboration of Standard Bodies

TR178

Version 0.4

September, 2012

Notice

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,
East Tower – 10th Floor,
Morristown, NJ 07960 USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

Table of Contents

List of Figures

List of Tables

Executive Summary

The business and operational model for Cloud adoption and deployment is still an emerging topic both for the Cloud services providers and buyers. To take hold and provide value, the Cloud usually consists of a collection of various services from various providers who form the Cloud Ecosystem, which presents further challenges. Some of the drivers for standards development to realize the Cloud Ecosystem include:

- Cloud consumers desire to have a single point of accountability for Service Level Agreements (SLAs).
- To manage SLAs end-to-end (E2E), Cloud Service Providers (CSPs) need to integrate various vendor service models and service dependencies.
- Vendors in a multi-cloud ecosystem (or "inter-cloud") may only provide a portion of the complete solution. Such vendors need to have consistent methods to report management information, which contributes to the overall SLA calculation, as well as the ability to differentiate themselves in a Cloud ecosystem.
- The various SLA related standards and best practices existing today need to be integrated and augmented to support the Cloud Ecosystem operation. A distinction is drawn between SLA, implying a legal contract with penalties for missed targets, and Service Level Objectives (SLOs), implying service level goals which carry no penalties if missed and no legally binding contract.

The major challenge for managing Cloud SLAs end-to-end is how to aggregate SLAs across different services and their related metrics and KPIs both at a single service level as well as at composite and aggregated level. Note that unlike traditional static models, the service composition can be highly dynamic in a Cloud Ecosystem. These challenges are intensified when most of the Enterprise cloud services deployment happens in a multi-cloud environment.

The SLA definition, SLA policy and SLA negotiation management must be flexible enough to support the Cloud Ecosystem operation; that is static and dynamic SLA management, SLA negotiation and renegotiation must be considered.

This Technical Report (TR178), while organized by the TM Forum, takes an outside-in look by reviewing existing relevant industry work (DMTF, OGF, NIST, ITU-T, ISMA, OASIS and other), as well as the TM Forum Frameworx, SLA management Handbook and Cloud SLA Application Notes. It then recommends a set of business considerations and architecture design principles that are required to support end-to-end Cloud SLA Management with the aim to facilitate discussion regarding SLA consistency across Cloud Deployment Models and Services Models.

The goal of this Technical Report is to enable people in different communities to develop Cloud SLAs and SLA metrics such that associated systems can be joined to facilitate end-to-end SLA management of a single cloud or multi-cloud environment.

1. Introduction

1.1 Document Structure

1.1.1 Audience

The first few sections of the document are targeted for general readers who are interested in the subject of Cloud SLA and for business people who have the responsibility of defining SLA contracts. Target audiences in this group including:

- Cloud buyers and providers who are responsible for the SLA contract development. CIO, Business and Market analysts to understand the business considerations in a Cloud Ecosystem and methods to help analyze and define a meaningful SLA that address business and service objectives.
- Service analysts and policy engineers who are responsible for the SLA metrics.

Primary readings for the above target audiences are: "**Key Concepts and Terms**", "**Business Considerations for Cloud SLA**" and "**Cloud SLA Metrics**" sections.

In addition to the general sections,

- System Architects who are responsible for the design of enterprise architectures/systems support end-to-end Cloud SLAs management should read the "**Architecture Considerations and Enablement**" and "**Use Case**" sections.
- Cloud Service Developers who want to participate in the Cloud ecosystem and monetize their applications should read the "**Architecture Considerations and Enablement**" section especially for the consideration of "**Use Consistent Method for Metric Collection and Reporting**".
- SDO participants who are responsible for the Cloud roadmap and SLA development should read the "**Architecture Considerations and Enablement**" and "**Issues and Future Work Recommendations**" sections.

1.1.2 Introduction

This document provides a set of Business and Architecture considerations and design principles to enable end-to-end Cloud SLA management by examining and leveraging existing industry work and recommending future work.

Section 1 **Introduction**: Provides overview of this document and outlines its structure..

Section 2 **Key Concepts and Terms**: Defines terms and concepts in SLA Management sources in order to understand the document.

Section 3 **Business Considerations for Cloud SLA**: Points out some of the business considerations on negotiation based on TMF GB917 SLA Management Handbook to establish a well-defined SLA agreement

Section 4	**Architecture Considerations and Enablement:** Emphasizes the essential considerations and design principles that will facilitate service providers to achieve the set of SLA management objectives
Section 5	**Use Case:** Depicts a scenario wherein a developer will create a multi-platform application for playing videos and will utilize cloud as a consumer, developer and service provider. The case explores and utilizes the key concepts in prioritizing, defining, understanding and using metrics in a cloud environment.
Section 6	**Issues and Future Work Recommendations:** Identify the standards gaps based on the studies of the TR178 effort and proposed work recommendations both for the TM Forum and for collaborating SDOs.

1.1.3 Appendices

Appendix A	**Terminology, Acronyms and Abbreviations**
Appendix B	**References**
Appendix C	**TM Forum Information Framework Patterns**

(1) Administrative Appendix provides document revision history, acknowledgements for work completed and information about the TM Forum.

1.1.4 Convention

A large number of external resources are used to produce this document; [ORG Doc#] citation style is used with a summary in the "**Error! Reference source not found."** section.

This text box style is used to highlight key considerations or recommendations.

1.2 Landscape of Industry Cloud SLA Standards and Best Practices

There are many Cloud SLA management related works in the industry, this Technical Report is not trying to reconcile them but to look at the commonality and complementary areas and provide an architecture roadmap to prevent standards divergence that will hinder the Cloud adoption.

The type of Cloud SLA related activities can be categorized as followings:

- Groups who are focused on end user requirements, such as the TM Forum Enterprise Cloud Leadership Council (ECLC), US Government National Institute of Standards and Technology (NIST).
- Groups who develop best practices to help guide the Cloud consumers and providers to define Service Level Agreements that meet their mutual business and service objectives.
- Groups who develop architecture artifacts or system specifications for SLA management in general.
- Groups who are looking at SLA metrics that are specific to satisfy the characteristics of Cloud service models and development models, such as elasticity, portability, pay-per-use etc.

The following diagram depicts the key documents that have been consulted for the production of this Technical Report.

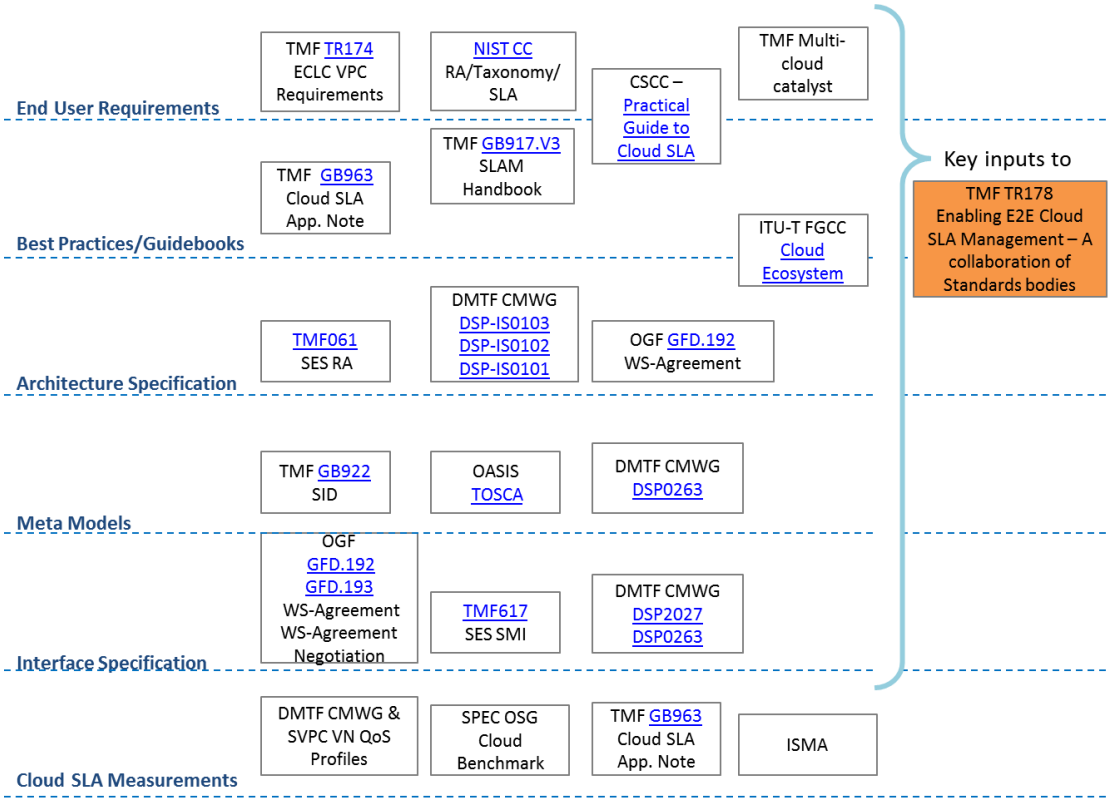

Figure 1 - Landscape of Cloud SLA Related Industry Activities

2. Key Concepts and Terms

The TM Forum GB917 Version 3 published in July 2012 provides a full set of harmonized definitions of terms used in the field of SLA management.

This document incorporates definitions from other Standards Developing Organizations (SDOs) with the attempt to clarify the inconsistencies and use non-TM Forum terminologies when appropriate. This section introduces a few essential terms and fundamental concepts used in this document.

Appendices

2.1 Appendix A

provides a list of abbreviations and acronyms used in the document.

2.2 Cloud Actors

There are several sets of Cloud Actor definitions existing in the industry, each being defined from the perspective of their target audiences. For example: NIST defines Cloud Actors from the perspective of Government Agencies procuring Cloud services, ITU-T defines the terms from the perspective of Telecommunications service providers, and DMTF defines the terms with a focus from Cloud technology implementers.

Table 1. Taxonomy of Cloud Actors

Cloud Actors	Definition
[DMTF DSP-IS0101]	Cloud Service Provider, Cloud Service Consumer, and Cloud Service Developer
[ITU-T FGCC TR]	Cloud Service User, Cloud Service Provider, Cloud Service Partner
[NIST CC RA]	Cloud Service Consumer, Cloud Service Provider, Cloud Broker, Cloud Carrier, Cloud Auditor
[TMF GB922]	Party and Party Role
[TMF GB917]	Customer Role, SP Role, User Role, Integrator Role

Within the TM Forum, a pattern based approach is used to enable different viewpoints to be described and related using a consistent method. This is an important concept to support various Cloud/digital Ecosystems that comprise of many vertical industry sectors. The following sections of this document "**Determine Roles & Responsibilities**" and "**Use Design Patterns for Repeatable Best Practices**" provide more detailed analysis of this subject.

2.3 Master Service Agreement (MSA)

US government National Institution of Standards and Technology suggest that the SLA is part of a "Master Service Agreement" [NIST Contract]. This reflects commercial practice of a Contract between two or more parties being made of a series of documents following standard templates.

The Master Service Agreement (MSA) is a contract between buyers and sellers. It contains considerations of:

- Stakeholders involved in the ecosystem
- Regulatory Compliance, Legal
- Remedy and Compensation
- SLAs
- Other Elements

One of the advantages of a MSA is that contractual Terms and Conditions related to the general business relationship can be separated from Service specific details and conditions, and avoids having unintended differences in the way services are contracted.

Typically, Master Service Agreement will reference documents containing definition of the Services being offered, and separately the SLAs for each of those services and possibly the operational aspects. A subtlety that is often missed is that some of the operational aspects of the MSA have SLAs that are common across all services e.g. operational metrics, such as time to repair. These are often captured in a separate business operational SLA distinct from the individual services.

2.4 Service Level Agreement (SLA)

The TM Forum SLA Handbook [TMF GB917] defines that the Service Level Agreement serves as a means of formally documenting the service(s), performance expectations, responsibilities and limits between cloud service providers and their users. Typically, the service definitions are a reference to the supplier product definition and this means that each SLA has to be written to accommodate the specific needs of each type of service that is being procured or, as mentioned earlier, operational aspects.

A typical SLA describes levels of service using various attributes such as: availability, serviceability, performance, operations, billing, and penalties associated with violations of such attributes.

Service Level Agreements: are referenced by the master service level agreement. They can contain:

Business level objectives (BLO):

- Business Level Objectives relate to the measurements that are not service specific but impacting Cloud Buyers' business objectives such as disaster recovery, data privacy etc.
- They may also determine the deployment model and associated parameters in a Cloud environment, such as Public Cloud, Private Cloud or a hybrid of both.

- Business level objectives often form the Business Policy that determines the value of the service metrics parameters.

Service level objectives (SLO):

- **SLOs** are specific measurable characteristics of the service being monitored. They are usually specified in a **Service Level Specifications (SLS)** template**.**
- SLOs are composed of one or more quality of service (QoS) objects or Service Metrics

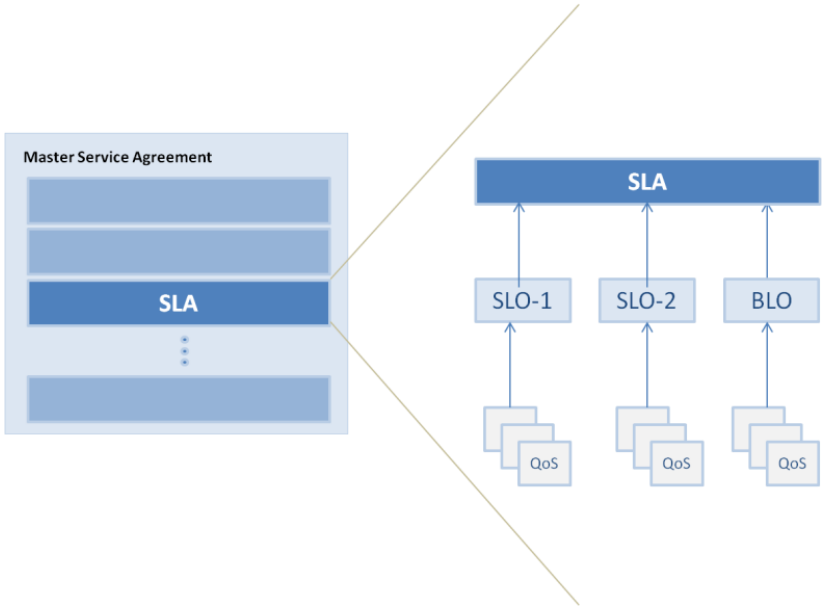

Figure 2 - Master Service Agreement and Service Level Agreement

2.5 SLA Management

SLA Management, as defined by the TM Forum, is about managing service quality throughout the **customer experience lifecycle**. This means managing service quality beyond the in-use phase of the lifecycle to include point of sales, provisioning, in-use phase and service termination aspects. It should also be noted that the in-use phase includes service components such as customer services and billing.

The TM Forum SLA Management Handbook [TMF GB917] contains the full details of this customer centric, service lifecycle focused SLA management.

2.6 End-to-end Cloud SLA Management

The major challenge for managing Cloud SLA end-to-end is how to aggregate SLAs across different services and their related metrics and KPIs both at a single service level as well as at composite and aggregated level. Unlike traditional static models, the service composition can be very dynamic in a Cloud Ecosystem. These challenges are intensified when most of the Enterprise cloud services deployment happens in a multi-cloud environment.

The SLA definition, SLA policy and SLA negotiation management must be flexible enough to support the Cloud Ecosystem operation; static and dynamic SLA management, SLA negotiation and re-negotiation must be considered.

The heart of this Technical Report is to provide a set of common approaches for two parties to determine their Cloud Service Level Agreement, define what to measure, the threshold and indicators as well as some architecture design principles for the service providers to "connect the dots" so that end-to-end Cloud SLA management can be achieved with process automation and architecture flexibility to support different business scenarios and customer needs.

2.7 Service Model

There is a need to clarify the term: "Service Model"

- Within the Cloud industry, the classifications of Services/Service Models are into categories of: IaaS, PaaS, and SaaS. Although the term "service model" is used, they are a set of taxonomies not a relationship model as is often referred to in the software context.
- In the Service Oriented Architecture (SoA) context, a service model is a classification used to indicate that a service belongs to one of several predefined types based on the nature of the logic it encapsulates, the reuse potential of this logic, and how the service may relate to domains within its enterprise e.g. entity centric, task centric, and utility centric.
- In the SLA context, Service Model refers to the resources associated with the service and the relationship of these resources to each other, as well as the Cloud business/service level objectives and Key Performance Indicator (KPI) and Key Quality Indicator (KQI) calculations used in the SLA.

To avoid confusion, this document will use the following prefix to clearly delineate the context when the term is being used:

- **Service Model**: A hierarchy of resources that make up the service
    - **Cloud Service Model**: Classification of Cloud services as defined by NIST [NIST CC RA] i.e. IaaS, PaaS, SaaS etc.
- **SLA Service Model**: The dependency and calculation of KPIs and KQIs into SLA Business Level Objectives and Service Level Objectives
- **SoA Service Model**: Software encapsulation and dependencies

3. Business Considerations for Cloud SLA

TMF GB917 SLA Management Handbook defines that any SLA management strategy should consider two well-differentiated phases:

- **The negotiation of the contract**, and
- The monitoring of its fulfillment in real-time.

Thus, SLA Management encompasses:

- **SLA contract definition**
- SLA enforcement—according to defined policies
- Basic schema with QoS parameters
- SLA negotiation
- SLA monitoring;

This section discusses some of the business considerations of negotiation of the contract for the purpose of deriving a well-defined SLA contract between two parties.

3.1 Cloud Ecosystem

Cloud deployment rarely involves only one single Service Provider (SP).

The diagram below (taken from TMF GB917) depicts a number of cooperating service providers that have various roles with respect to the business relationships between them, notably customer and provider roles. Shown on the left are end Customer Organizations (C1 and C2) which one lead service provider (S3) is supporting directly, and the others support indirectly through the lead SP.

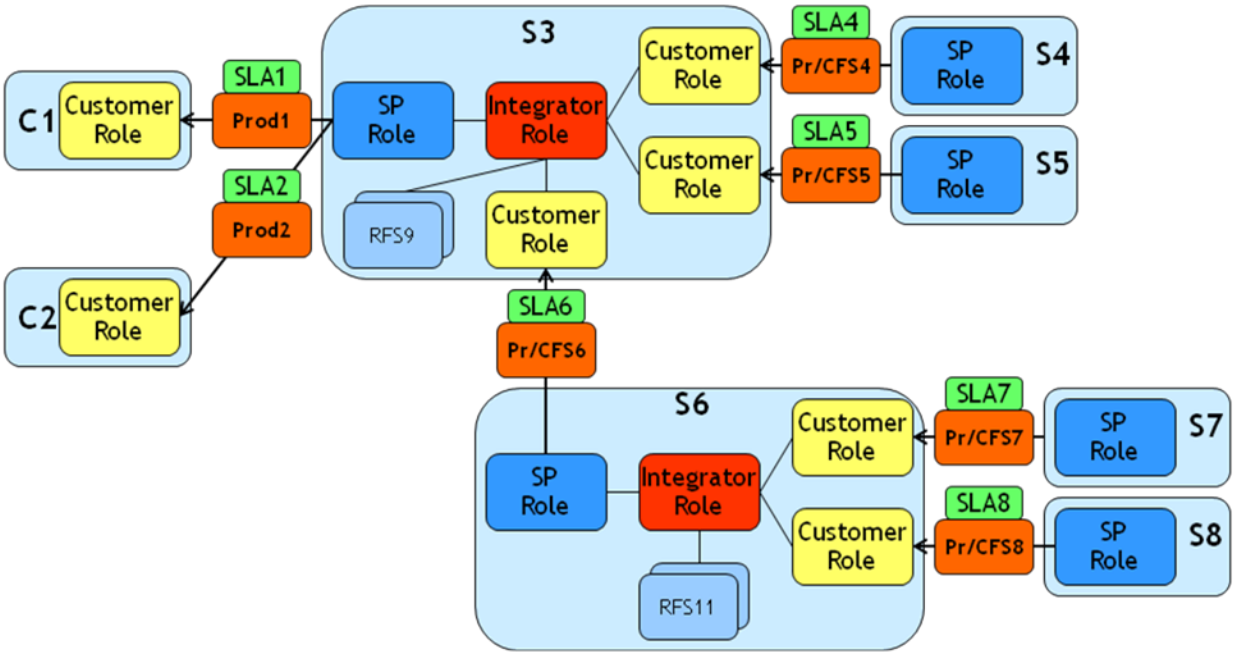

Figure 3 - Complex Service Delivery Relationships

In this diagram (from GB917]), a number of cooperating service providers are shown, each having various roles with respect to the business relationships between them, notably, customer and provider roles. Shown on the left are end Customer Organizations which one lead service provider is supporting directly and the others support indirectly through the Lead SP.

Each service provider defines and offers a **Product** - the formal TM Forum term of a Marketed Service. Associated with that each delivered Products are SLAs that have been agreed to between the SP and their customer. Any SLA failures or remedies will be restricted to what has been agreed to in the SLAs for the Service and they can cover both operational business aspects and the functional service aspects.

The question in a multi cloud provider environment is:

Who is responsible for the end-to-end SLA and how is the overall SLA managed?

A few anchor points can be established:

- The overall SLA is usually offered by the Lead Service Provider (S3) to the end customer organization (e.g. SLA1 and SLA2).
- It is the responsibility of the Lead Service provider to ensure that the SLA they offer is backed up by the SLAs they have agreed with their supplying service providers - S4, S5 and S6 in the figure above.
- An integrator role (S3 and S6 above) always holds the responsibility for ensuring that the SLA presented for their overall product offering (e.g. S3: SLA1, SLA2) is backed up by the supplier SLAs to the lead Service Provider(e.g. SLA4, 5, and 6).

- Where the lead SP (S3) does not have direct business relationships with all the services providers, other integration roles must be present in the value chain. i.e. S6 fronts or leads for S7 and S8 and therefore also has an integrator role. (Note that it is possible for a SP to use a third party for the management of the integration role see below).
- KPI to KQI mapping/translation is most commonly performed within one Service Provider.

The second open question is: "How is the e2e SLA deduced from the individual SLA KQIs exposed by each Service Provider?" There are several basic approaches which go from the pragmatic to the theoretical:

- The most pragmatic approach is to agree on SLA KQIs that are to be exposed by a provided service, and to set upper and lower bounds on the metrics based on a view of the end-to-end requirement. This can work for things such as latency, jitter, and error rates. Alerts or warnings are generated when SLA metrics move outside the expected boundaries. This approach avoids the need to mathematically manipulate and combine metrics, and is simple where the understanding of performance impacts of Impairment is not well established.
- Mathematical approaches are possible for combining metrics but these require knowledge of:
  - The impact of metrics on the end Service
  - Perceived quality of service for any specific level or combination of SLA KQI(s).
  - The mathematical form of the distributions
  - Which mathematical operations are valid such as correlation, sum of means and squares etc.? In practice, these are difficult questions to answer.
- Especially in the early stage of a service deployment, it is difficult to justify more complex methods without evidence that they are effective, and the main constraint is to minimize the number and complexity of monitoring points /probes. They also need to be somewhat independent of the end services, otherwise service scalability and agility is impeded.

3.1.1 Use of third parties to execute Integrator role

In value chains, it is possible to have intermediaries that perform various functions on behalf of others; the most commonly known ones are for security and billing. It is possible for an integrator role to delegate the actual collection and analysis of SLA metrics to a third party SLA Auditor role as illustrated below:

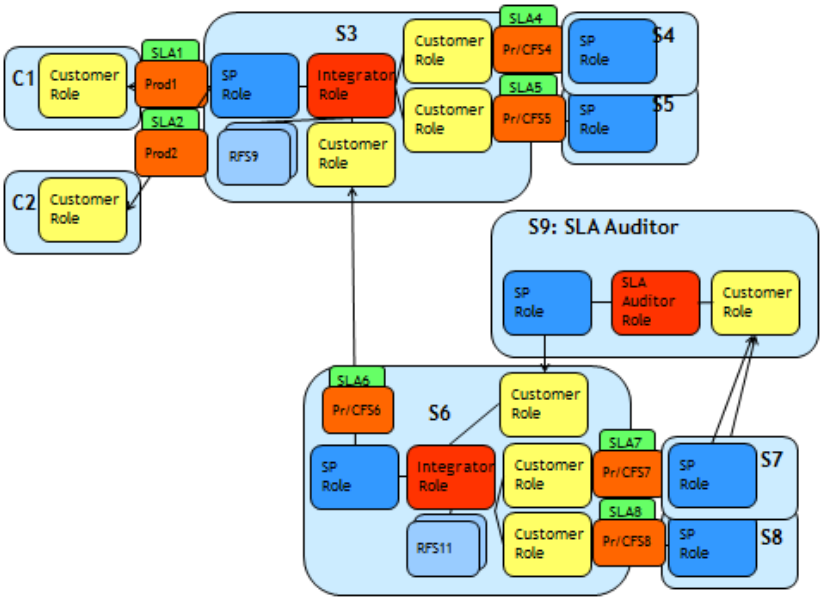

Figure 4 – Delegation to Third Party SLA Auditor

In this example SP S6 has decided to delegate the measurement of SLA Metrics/ KQI to a third party S9, and for S9 to notify S6 when the combination of S7 and S8 SLA performance moves outside defined limits.

Several variations on this theme are possible:

- S6 may itself forward KQI/KPI metrics for its RFS 11 to the SLA Auditor so that the auditor only indicates issues where the SLA6 is in jeopardy of being out of bounds. Which approach is used is determined by S6, but in all cases it remains accountable and responsible for meeting SLA 6.
- S3 may also choose to delegate its Integration responsibility for e2e SLA to a third party SLA Auditor for integrating the SLAs 4, 5 and 6. This may be different from the one chosen by S6. However, there are advantages to all parties in a value chain delegating to a single SLA Auditor which arise from scale advantages, and ease of resolving SLA related disputes with a single neutral third party.

3.2 Determine Roles & Responsibilities

The Practical Guide to Cloud Service Level Agreements [CSCC SLA] produced by the Cloud Standards Customer Council (CSCC) provided a set of prescriptive steps that should be taken by cloud consumers to evaluate cloud SLAs from their providers, these steps include:

1. Understand roles and responsibilities

2. Evaluate business level policies
3. Understand service and deployment model differences
4. Identify critical performance objectives
5. Evaluate security and privacy requirements
6. Identify service management requirements
7. Prepare for service failure management
8. Understand the disaster recovery plan
9. Define an effective management process
10. Understand the exit process

As one can see, SLA accountability, between Customer – Provider and among partners of providers, is often based on business factors rather than technical factors. One of the goals of this document is to provide some pointers and frameworks that can help facilitate stakeholder discussions to better derive the business decisions that need to be made for defining what goes in to the Cloud Service Level Agreement.

First, the business arrangement and decision will be largely based on the stakeholder business model which defines the business logic of a company at the strategic level.

The picture below [Alexander Osterwalder, PhD Thesis] shows that the relationship between Strategy and Business Processes (here exemplified for Enterprises adopting IT and Internet technologies) can be better understood, and consequently the implementation of a strategy better controlled, if business models are formulated.

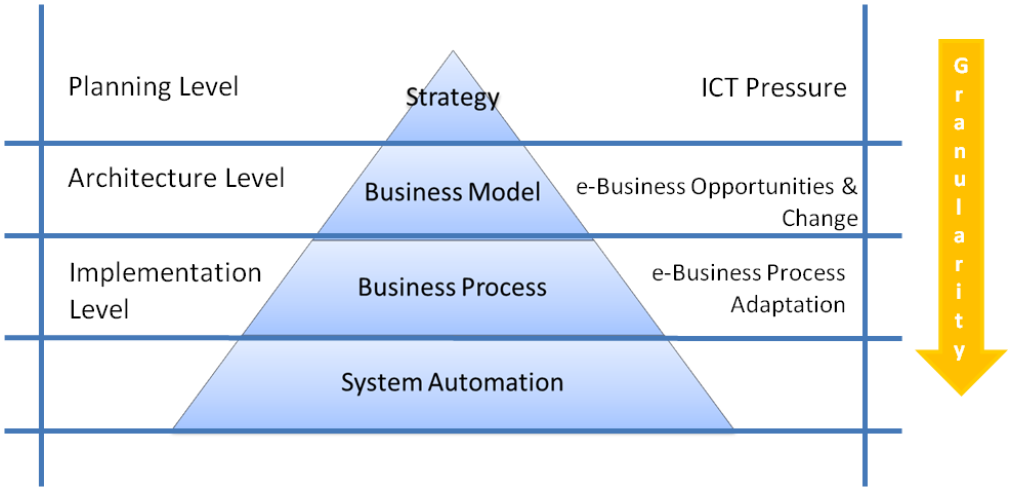

Figure 5 - Relationship between Strategy and Business Processes

Cloud is a fairly new paradigm for both the buyers and suppliers. From Cloud buyers' perspective, especially for Enterprise customers, the Cloud business decision and pricing point are usually dependent on their overall IT strategy.

3.3 Manage Multi-stakeholder SLA Lifecycle and Service Lifecycle

Ultimately, SLA management is to manage the Customer experience and Customer expectations. There are several dimensions to this; it is clearly beyond just monitoring and managing a set of metrics.

The following diagram combines the Customer Experience lifecycle (TMF TR149) and SLA Management lifecycle (GB917) views.

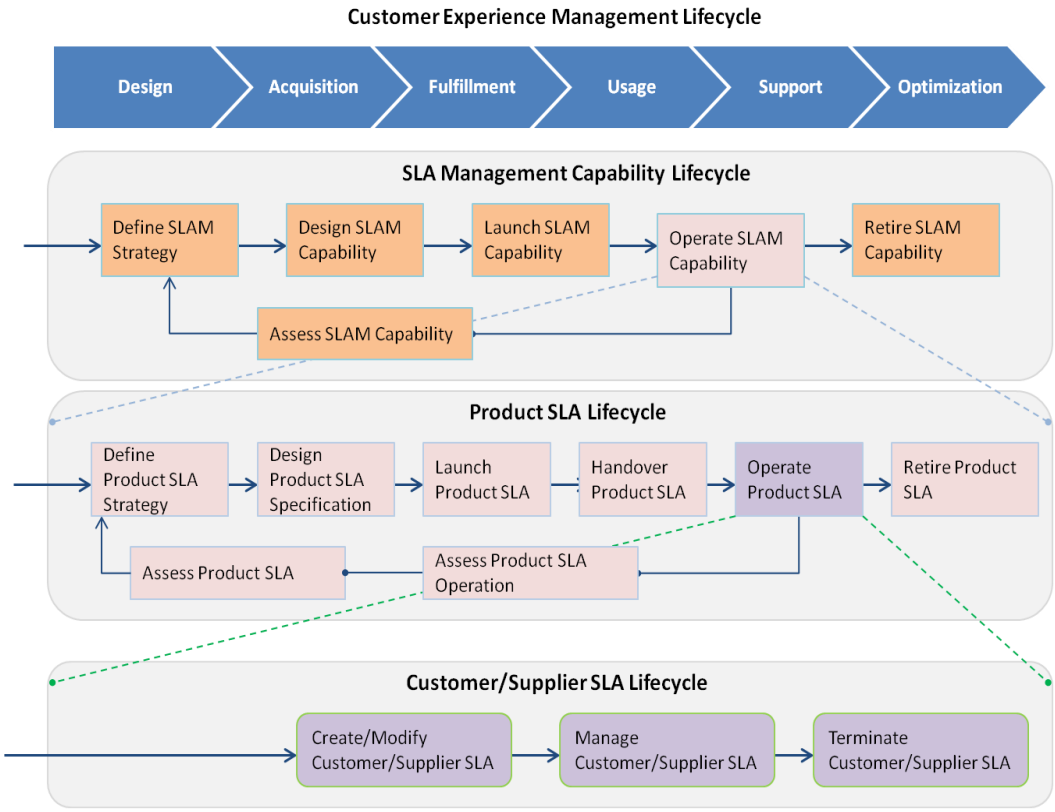

Figure 6 - Multiple Dimensions of SLA Lifecycles

For example, in the use case of this document, Chris is a developer going through two phases of the Customer Experience lifecycle: 1) as a developer who uses the infrastructure (IaaS) and Platform capabilities (PaaS) from the provider 2) as an application provider (SaaS) who leverages the providers' marketplace platform.

In the use case, we also demonstrate the Customer/Supplier SLA lifecycle of switching suppliers, due to one of the suppliers not meeting SLA requirements.

In addition to the consideration of Customer Experience, SLA management, Product SLA and Supplier/Partner SLA lifecycle, for any large scale Cloud service such as storage as a service, there are many line-of-business stakeholders from the design and implementation to the deployment and support lifecycle of the service. These stakeholders can be within one organization or cross multiple business/organization boundaries.

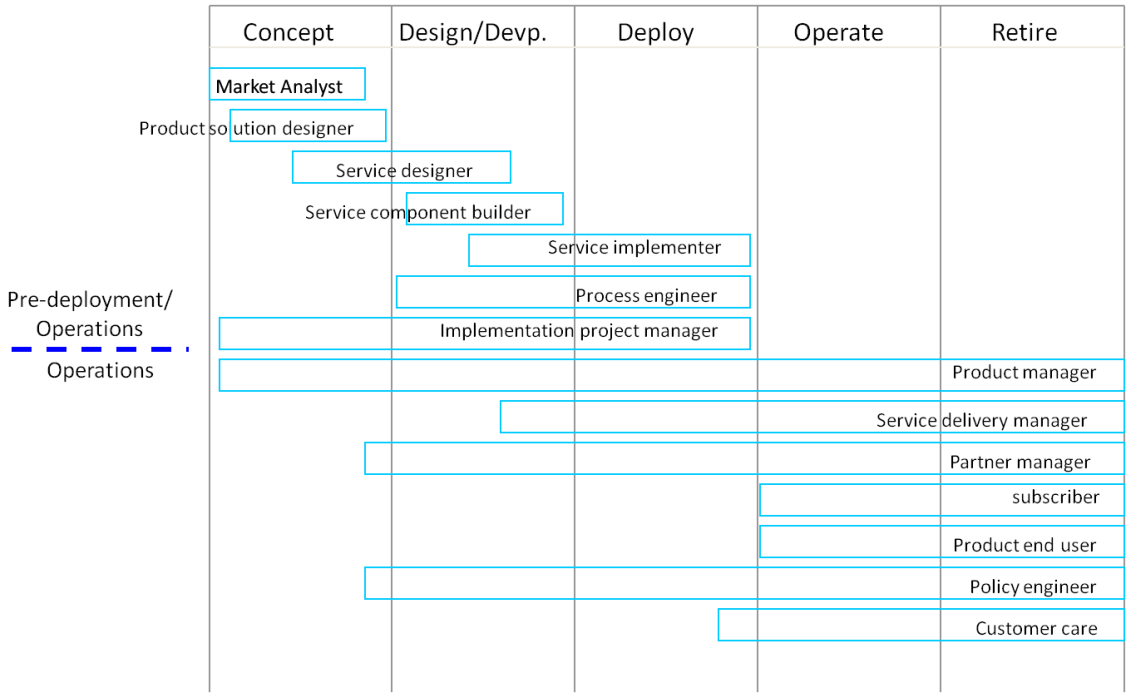

Figure 7 - Service Lifecycle and Multi-stakeholder Involvement

For example,

- Product solution designer may enter some information about general product configuration: e.g. Gold email package: 99.9% availability, 10GB storage
- Implementation manager specifies additional parameters based on customer requirements: e.g. HIPPA, PCI compliance transmission
- Policy engineer may add additional constraints based on business requirements: using datacenter 1, 2, 4 only, etc.

It is important during the SLA and master agreement definition that this multi-dimensional and multi-stakeholder lifecycle complexity is taken into consideration. The "**Architecture Considerations and Enablement**" section of this document makes recommendations of how to "**Use Software Factory to Automate Lifecycle Management**" to manage and to reduce this complexity.

3.4 Cloud SLA Metrics

The term "metric" is not consistently defined in the ICT industry. In this document, the term "metric" is used to describe individual "measures" such as the number of users and "metrics" such as "GB per Second".

Although TM Forum has specific definitions for KPI and KQI, for the purposes of this paper, we don't distinguish, suggesting that most organizations need some simple examples and explanations, before further categorization.

Cloud metrics largely fall into two major categories: Business metrics (often defined within the SLA), and Technical metrics (monitoring metrics) that enable the business SLA to be met.

For example, "response time" may be specified in the SLA, meanwhile other technical measures such as "hops" and "bandwidth" may be used to dynamically allocate resources, enabling "response time" SLAs to be met. Usage based costing metrics are generally a sub-category of the business metrics and will be a major component of a Service Agreement or Service Level Agreement. Some examples of usage based metrics are: Number of Users, Instance Minutes, Storage Resource Capacity Used Bytes, CPU Minutes and RAM in Megabytes. Costs metrics are established based on dollars per unit ("$/ Instance Minute" for example).

Another perspective is recognizing that you will have different metrics objectives if you are looking externally or internally, and if you are a consumer or a provider in some context.

The Use Case of this document generates early discussions and begins foundational discussions regarding: goals, questions and metrics.

SLA metrics require appropriate categorization and clarification to align with SLA objectives and specify consequences when SLAs are not met.

Table 2. Example of "Response Time" Metric Analysis

Metrics	Response Time
Context	Response time for requesting/ obtaining additional storage
Constraints	During business hours 00:00 GMT to 12:00 GMT (no guarantee outside of business hours)
Measures (1)	Date/ Time of Request to Cloud Provider from Cloud Consumer (triggered by storage request)
Measures (2)	Date/ Time of Successful Completion Response from Cloud Provider
Metrics Calculation	Measure (2) - Measure (1)
Collection Method	Automated, triggered as part of service request
Units	Milliseconds
Used & Consequences	Used: Cloud Provider guarantee is a maximum of 3000 Millisecond response time for IaaS Storage requests 00:00 to 12:00. For every 10 IaaS storage requests exceeding 3000 Millisecond response a 10 %

reduction will be applied to total IaaS storage charges for the month

This is just a simple example of how the metric of "response time" can be defined and assessed in context with SLAs, clarifying response time in a "real-life" scenario.

Metrics considerations are dependent on the supported service models (IaaS, PaaS and SaaS) and the type of services provided within that model, for example, network, storage and computing services for IaaS.

In the NIST Cloud SLA Taxonomy, the "metrics" (response time, availability throughput) identified are examples of three frequently cited metrics in context of SLAs; however they are not an all-inclusive list. Metrics in context of SLAs often have stated expectations related to minimums, maximums, defaults and consequences for deviations from stated objectives.

In summary, when considering metrics in a cloud SLA, it is recommended that consumers and providers:

- Understand the business objectives for the cloud opportunity.
- Understand context and where the stakeholders fit into the cloud ecosystem.
- Understand potential cascading SLAs and associated metrics.
- Understand enabling "technical metrics" vs. more visible "business metrics".
- Identify the set of metrics that align with prioritized objectives.
- Understand the usage cost models that are applied.
- Clarify how the metrics will be used and what decisions will be made.
- Ensure these metrics are defined at the right level of granularity and can be monitored on a continuous basis.
- Determine available standards that help provide a consistent measurement method (some will evolve as cloud computing matures).
- Understand the value and limitations of the metrics collected.
- Analyze and leverage the metrics on an ongoing basis as a tool for influencing business decisions.

Please note that this is not exclusive to the consumer and provider roles, but rather any of the identified roles in the cloud eco-system.

Cloud Computing metrics provide critical information to optimize cloud experiences, perform comparative analysis, and help make informed decisions. Metrics are a cornerstone for transparent Service Level Agreement management and good governance.

4. Architecture Considerations and Enablement

Given the complexity of the Cloud SLA definition and Cloud SLA management business considerations, as discussed in the earlier sections, customer facing service providers must prepare their service delivery and management architecture to address those challenges. Furthermore, this architecture must be flexible and extensible to support different business scenarios and diverse requirements from different industry sectors.

In this section, we highlight the architecture considerations and design principles that will enable service providers to meet the end-to-end SLA management objectives. They are:

- **Pattern-based approach** to define roles, responsibilities and their relationships to Cloud services
- **Software factory-based development** to facilitate and coordinate the lifecycle management of
    - Customer Experience
    - SLA Management
    - Product SLA
    - Supplier and Partner SLA
- **Consistent method for metric collection and reporting** of highly distributed and virtualized services
- **Service catalogue** to facilitate multi-stakeholder SLA operations
- **Standardized service template**
- **Automated SLA negotiation** and renegotiation
- **Policy engine** for business policies and SLA rules enforcement

4.1 Use Design Patterns for Repeatable Best Practices

The Cloud industry is so complex and diverse that there may be no one size fits all solution. Best practices may be in favor of standards whose implementation will need to be adjusted based on specific business, deployment and implementation arrangements between the Cloud buyers and suppliers.

However, knowledge of best practices can be captured in the form of design patterns, and relationships can be formalized with models. It is this level of standardization that will help provide the consistency required for all stakeholders to analyze the situation, regardless of their view points, e.g. SLA for a Government contract, or SLA to an individual cloud service developer. The SLA may be initially defined based on a provider's standardized product. Adjustment can then be made for different industry sectors or customer specific requirements.

For example, Cloud Actors defined by NIST, DMTF, ITU-T, while slightly different on their meaning, can be generalized using the TM Forum SID Party-PartyRole pattern [GB922]:

- Party: an organization, individual or system
- Party plays a PartyRole: Cloud Broker, Cloud provider, Cloud consumer, etc.

To facilitate the analysis of Party and PartyRole, TM Forum SLA Management Handbook (GB917) further sub-classes the PartyRole into:

- Service Provider Role
- Integrator Role
- Consumer Role
- User Role

The following picture provides a mapping of this generalization:

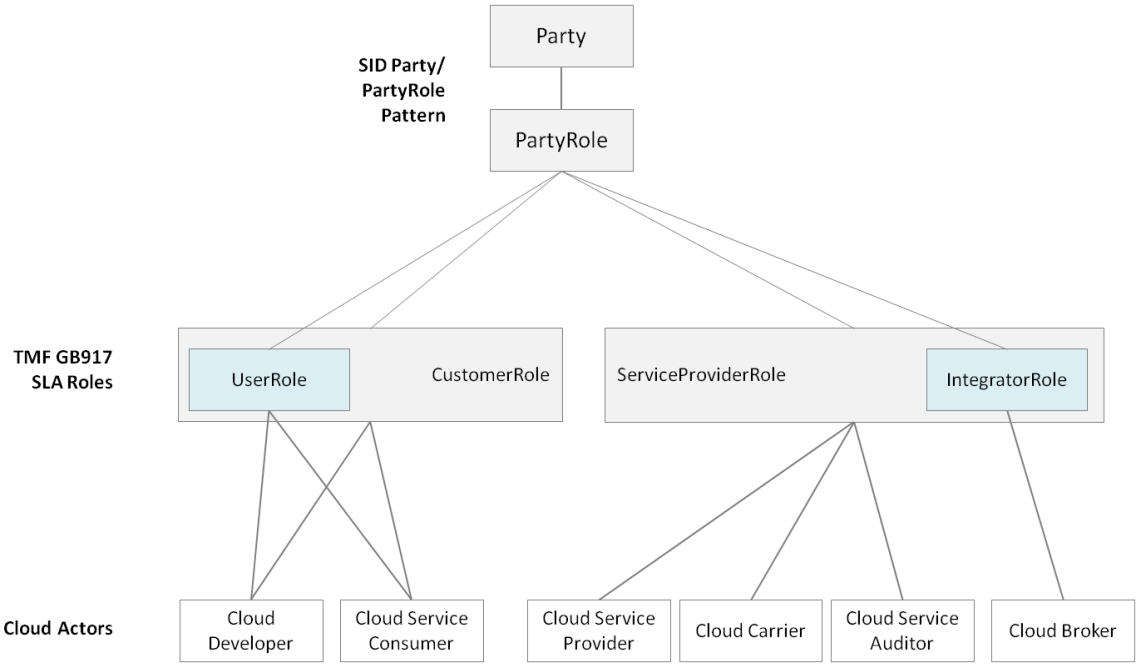

Figure 8 - Mapping of Cloud Actors to Role Pattern

Given that an SLA is the agreement between a consumer and provider of a Product/Cloud Service, this generalization will enable a design pattern like the following one to be specified in terms of the cardinal and relationship between Consumer, Provider, Product, SLA and SLS (service level specification).

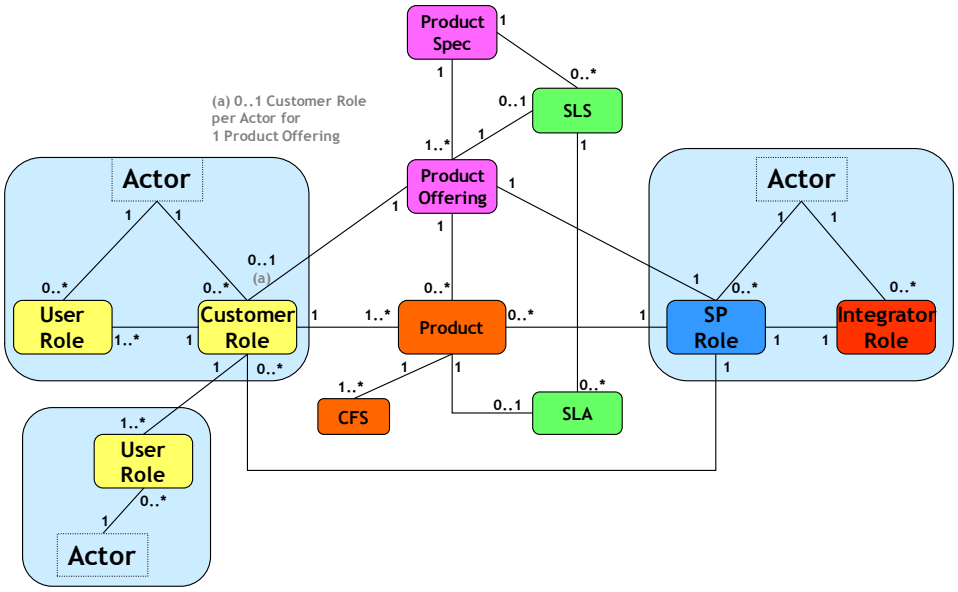

Figure 9 - Pattern of Products, SLAs and Roles

TM Forum Cloud SLA Application Note (GB963) provides extensive analysis of Enterprise cloud requirements and implications of SLA using the Actor-SLA-Product pattern.

Appendix B of this document provides details of additional patterns that support the product, service and resource relationships as well as their relationships to SLO (Service Level Objective).

A modeling and pattern based approach will enable insurmountable Cloud SLA scenarios and SLA configurations to be documented and processed in a much more systematic fashion to support process and system automation.

4.2 Use Software Factory to Automate Lifecycle Management

As mentioned in the "**Business Considerations for Cloud SLA"** section, ultimately, SLA management is to manage the Customer experience and Customer expectations and there are many nested layers of complexity to manage the Customer Experience lifecycle, SLA management lifecycle and product, service lifecycles.

A software factory-based approach is recommended to define the tools and processes that are necessary to effectively and accurately capture the management/SLA information among stakeholders.

While the tools and processes will be different from organization to organization, a small set of standards are necessary to ensure interoperability among systems.

The TM Forum Software Enabled Services (SES) Management Solutions Reference Architecture (SES RA) (TMF061) provides a set of design patterns on how service lifecycle management can be achieved consistently in a multi-stakeholder environment.

Additional work could be done in collaboration with OASIS Topology and Orchestration Specification for Cloud Application (TOSCA) group to harmonize the two designs and for the inclusion of SLA templates.

The SES RA supports the contextual information of a service in relation to the business and operation environment, through the definition of the SES Management Interface (SMI) (which will be explained in the next sub-section) and the SES Lifecycle Management Metadata (or schema) associated with the service, as well as a set of support services that allows unified lifecycle management of services and their decencies that may be managed in different service management frameworks (e.g. ITIL, TM Forum Frameworx or others).

The SES RA patterns such as the SES SMI, SES Lifecycle Metadata and SES Support Services (MSSes or ISSes) enable the management of dependencies that are intrinsic. See next two sub-sections for more details on this concept.

4.3 Use Consistent Method for Metric Collection and Reporting

There are two principal differences with cloud computing that makes the problem of managing resources associated with cloud services more difficult. One difference is the virtualization at the elastic compute and elastic network layers as well as the sheer scale of that virtualization. The other difference is the complexity of the Cloud ecosystem that multiple clouds and multiple enterprise domains are increasingly involved in the delivery of cloud services further complicating resource management.

To enable service providers to manage Cloud SLA end-to-end, standards must be defined to allow management applications (manager) to interact with managed applications and their resources in a consistent manner for data collection and for other management function.

Cloud consumers and providers may mandate that all Cloud Services be managed. A Cloud service (or any digital service) that is manageable is called a “well designed service”. A well designed service, besides exposing its functional capabilities, must be equipped with interface(s) to reveal its manageability to its service provider.

The TM Forum SES Management Solutions solve this problem by defining a management API called SES Management Interface (SMI TMF617). SMI defines for developers a key design pattern for including management capabilities in a service as they design and build it. It also enables service providers to manage each service or composition of services in a consistent and efficient manner.

SMI defines a set of simple operations analogous to the SNMP Get, Set functions. These operations are:

- **getExecutionState** returns the current execution state of a service instance;

- **getManagementReport** returns a report containing information about the service instance health, execution state, eventual failures and metrics (usage, performance for example);
- **getServiceConfiguration** returns data that describe the current set configuration values used by the service instance;
- **setExecutionState** allows a service consumer to activate or suspend service execution;
- **setServiceConfiguration** applies configuration values used by the service instance;
- **registerListener** sets the communication endpoint address to enable emitting notifications to consumers;
- **unregisterListener** de-activates the notification mechanism.

This set of operations will enable service providers to exchange management information with Cloud services that are designed and provided by other providers in a consistent manner, for example an XML document can be exchanged using **getManagementReport** between the Cloud service provider and the "customer facing" Cloud service provider (i.e. Cloud broker, integrator) to report on SLA metrics. The **setServiceConfiguration** operation can be used to set SLA threshold parameters from the service providers' policy engine to the Cloud service component.

4.4 Use Catalogues to Facilitate SLA Configuration and Negotiation

At the most basic level, a service catalogue provides a portfolio of services that are available to the customer. To drive automated product offering/bounding from various service capabilities in near real time, the Active Catalogue concept (TMF867) has been used that models the same approach as Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) from the manufacturing business.

The key concept is to remove the service and product dependencies from the system, and model them in the catalogue via well-defined ProductSpecification, ServiceSpecification and ResourceSpecifications and their dependencies. Appendix B provides an overview of those specifications and relationships.

In addition, federated catalogue management across different stakeholders with well-defined schema and interfaces will enable the integration and automation of SLA configuration and data (SLA service mode) management and synchronization across organization boundaries and span various service lifecycle phases.

Combine this concept with the lifecycle management and SMI described earlier, Figures 10 and 11 depict architecturally how this would work (this view is simplified with only three lifecycle phase considerations: Service Design, Deploy and Operation phases)

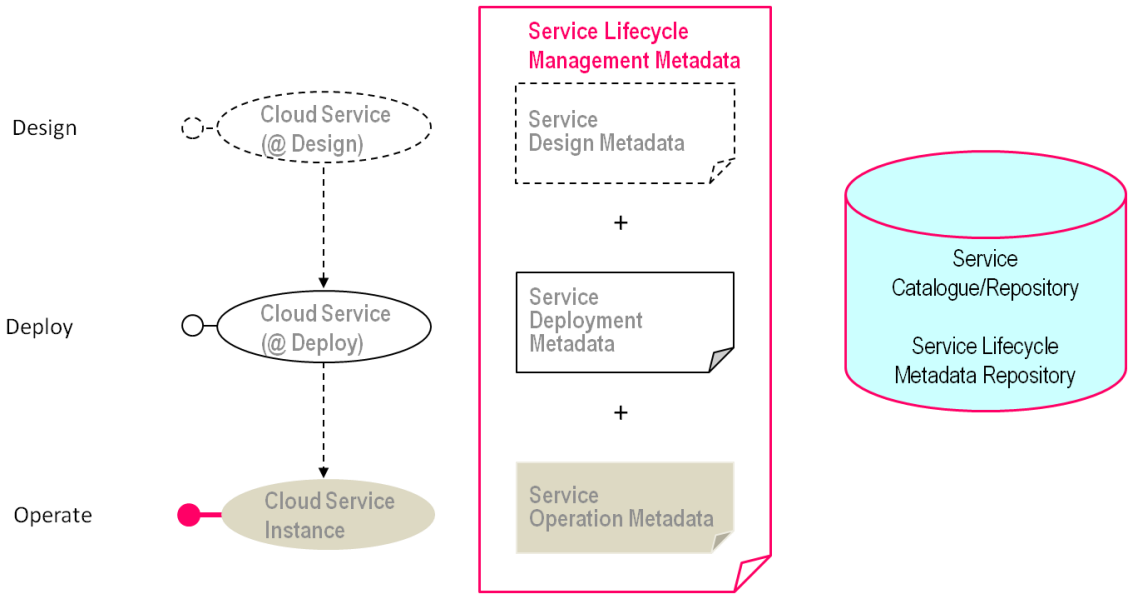

Figure 10 - Three Service Lifecycle Phases

The figure below illustrates the activities during three service lifecycle phases:

- Service Design Phase: Software factory tools and processes can help govern and facilitate the Cloud service design phase. These tools will be implemented according to standards of **Standardized Service Templates** that contain information that will contribute to the SLA calculation at the design phase, for example a typical Response Time of 3-5 seconds.
- Service Deploy Phase: System or policy engineer may provide additional information for the specific deployment, such as response time must less than 4 seconds and using high-security encryption mechanism
- Service Operation Phase: This is where all the monitoring and data retrieval activities happen. The management applications need to have a consistent method to access and/or retrieve and report from each instance of the service. This can be achieved via SMI and the management and monitoring threshold based on the information that is provided in a catalogue/repository from various stakeholders.

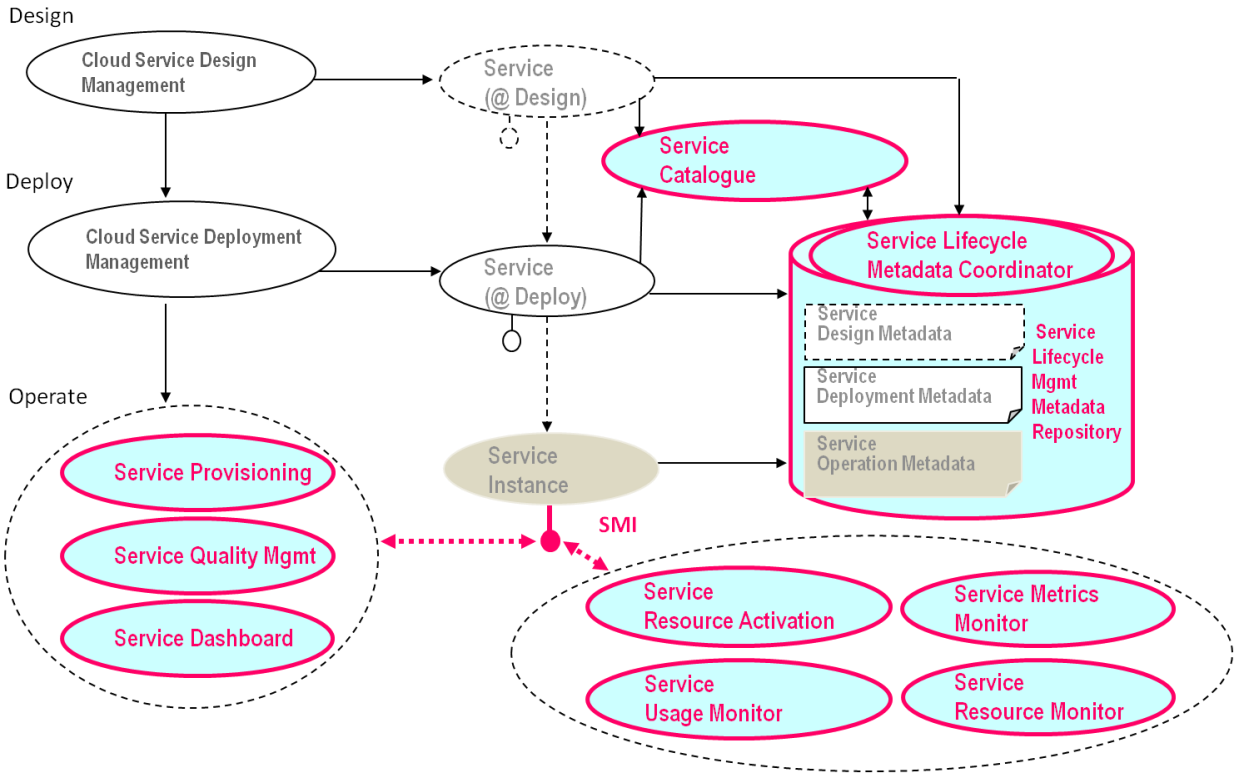

Figure 11 - Service Life Cycle Phases defined with Activities

Service catalogue and federated catalogue management are essential to handle the complexity of SLA management for Cloud. Although there is some work being done in the TM Forum, but more synergy in the industry is required to derive consistent methods for Service Templates, management metadata coordination, and management interfaces to be included for all Cloud services.

4.5 Service Template

In the key concept and terminology section, several definitions of "Service Model" were introduced, each defined with its specific context in mind, such as Cloud Service Model vs. the traditional service model of, say VPN service. There are also some key requirements from the end user perspective and from service designer and provider perspectives around the concept of a "Service Template"

- **Cloud Buyers/Consumers:** want to have a standardized service template so that they can compare service providers' like-to-like. For example, a service template for storage service may contain the following information, we can categorize them as "service characteristics" :
    - Virtual Machine/…/Volume
    - Virtual Machine/…/Volume/Quality/MaximumSize
    - Virtual Machine/…/Volume/Quality/Type [zfs, xfz]
    - Virtual Machine/…/Volume/Quality/Encryption [yes, no]

- Virtual Machine/…/Volume/Constraint/MaximumNumber [1..N]
- Virtual Machine/…/Volume/Cost/Size/perMbyte
- Virtual Machine/…/Volume/Cost/persistent

- **Cloud Service Designer:** in addition to the above information the designer of the service will also need a place to record its
  - Service dependencies
  - Service manageability
- **Cloud Service Provider:** who has the responsibility of SLA and end-to-end service management, the service template will need to hold or to provide a mechanism to express the business or operation context for a given service instance.

Also worth noting are industry standardization efforts in this area including:

- Service Template definition of OASIS TOSCA (Topology, Orchestration Specification of Cloud Application)
- Service Model and associate service specification template from the TM Forum SID (Information Framework)

TOSCA takes a metamodel approach; this metamodel defines both the structure of a service as well as how to manage it. A *Topology Template* (also referred to as the topology model of a service) defines the structure of a service. This includes *Nodes* that describe what the service does and its dependency through the *Relationship Type* templates. *Plans* define the process models that are used to create and terminate a service as well as to manage a service during its lifetime.

The TM Forum SID takes an information model view, the fundmental concept of SID service model is built upon the Product-Service-Resource relationship and the concept of CustomerFacingService (e.g. APIs exposed to external parties) and ResourceFacingService (e.g. interface that is usually internal and supporting the interaction with its depending resources). Appendix B provides detailed explanation of these relationships.

A standardized service template is essential to record service characteristics in a consistent and machine readable form. This template should also help to record information that is intrisic to the service based on the context of business objectives and service objectives such as service dependencies and SLA metrics.

We recommend harmonizing the work among OASIS/TOSCA, TM Forum SID and SES to meet these requirements. Quoting from TOSCA draft spec: The TOSCA Service Templates (the model) has to be linked to deployment artifacts for creating actual Cloud service instances, e.g.

- OVF for virtual machines
- EAR files or SCA assemblies for application components

This is very complementary to the TM Forum SES design. The following diagram suggests an example of TOSCA service template extenstion to support SLA management and possible mapping to SID information framework:

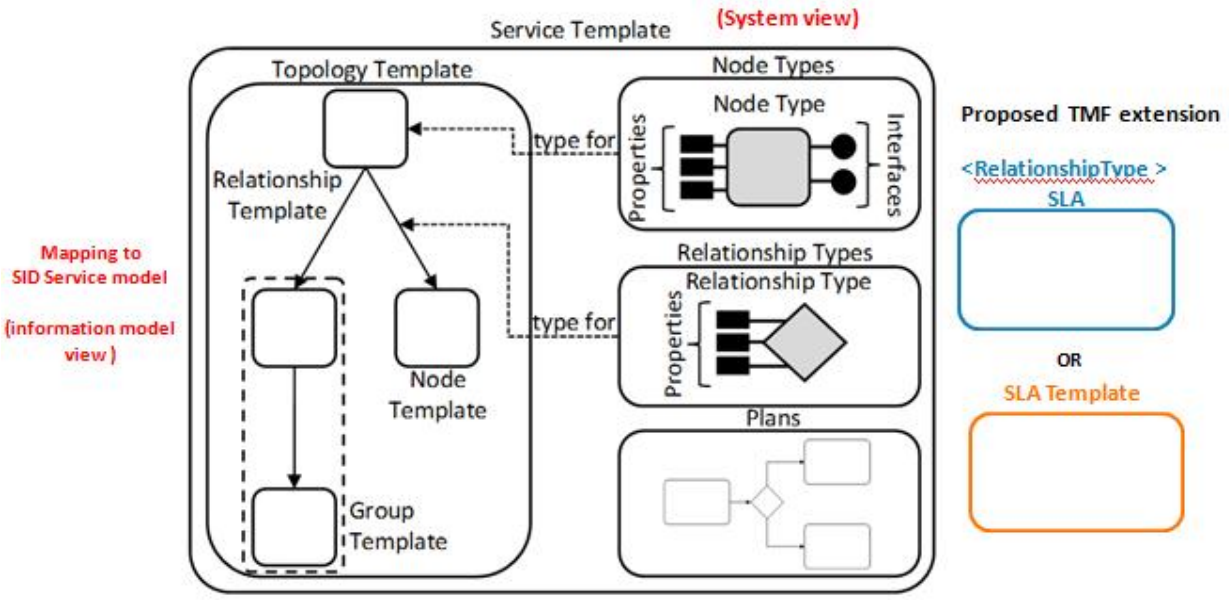

Figure 12 - TOSCA Service Template Extension Example

The service templates should be stored in Catalogues. Additional standardization items are also required to facilitate SLA management and reporting. This may be an integration of various catalogues/knowledge bases which hold the following information in order to effectively interpret the management report/dashboard:

- Metrics ID
- Taxonomy of Services (physical assets)
- Reference data, such as exception ID etc.

4.6 Automated SLA Negotiation

The negotiation of SLAs will be closely tied to the negotiation of information contained in the Service Template as described above.

Since different SLAs may be negotiated for each instance of the same service, the SLA template should be a separate component/class from the service itself as depicted in the above diagram.

The Open Grid Forum Web Service Agreement Specification (WS-Agreement GFD.192) defined a basic agreement structure shown in Figure 13. The WS-Agreement Negotiation (GFD.193) further defines a set of interfaces for the SLA negotiation as depicted in Figure 15 – SLA negotiation and Interfaces

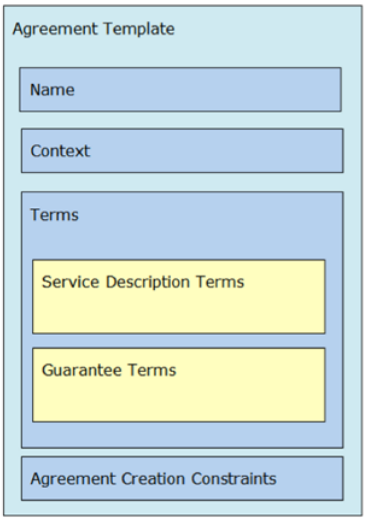

Figure 13 SLA Template

Figure **16** – SLA negotiation and Interfaces

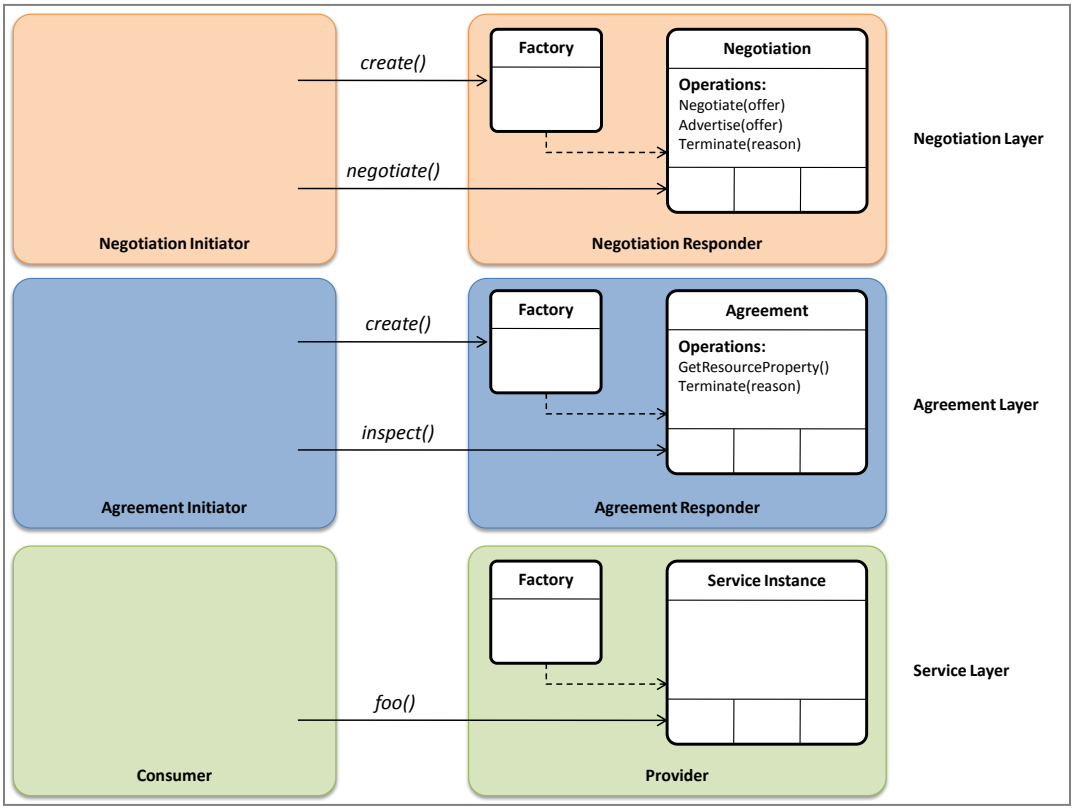

Figure 16 – SLA negotiation and Interfaces

The owner of the SLA provides the service provider with an SLA template document upon request. The service provider then formulates an offer (in the form of an XML document with the same structure as the agreement) and submits it to the SLA owner. As per GFD.192, "An offer item specifies the requirement for the presence in the agreement offer terms of a field and a value for that field. It contains a label, a pointer to the position of the field in the terms of the offer, and also MAY contain a definition of its acceptable values in the form of a restriction of its value space."

The offer is then either accepted or rejected by the SLA owner.

When the offer is accepted, the new SLA becomes immediately effective. Specification GFD.193, section 7.3, describes the guarantee state which "represents a state of fulfillment for each guarantee term of the agreement". By this mechanism, the current state of each metric is reported and noted for compliance or non-compliance.

The SLA can be re-negotiated through the submission of additional offers.

The OGF WS-Agreement and WS-Agreement Negotiation provided a good foundation for SLA negotiation; however, it was designed with a single service in mind (vs. composition of services). While some examples of WS-Agreement in orchestration scenarios exist further studies are required to align work efforts among OGF/WS-agreement, OASIS/TOSCA and TM Forum/SES on SLA Service Template, SLA negotiation and associated interfaces.

4.7 Policy-based SLA Configuration and Enforcement

SLA, Business Objectives, and Business Rules are all forms of policies; the Cloud SLA management architecture must be flexible and adaptable enough to support different policies to meet customer business objectives and operations efficiency for the providers.

The SLA metrics may be defined differently based on the Cloud Service Model (IaaS, PaaS, SaaS, XaaS etc.) and Deploy Models (Public, Private, Hybrid) that the customer subscribes to. For example, the "response time" of a SaaS Application deployed in a public cloud environment will have less stringent requirements than a "response time" of an infrastructure service that is purchased for a Private Cloud environment. Different types of applications will also have different security requirements, for example, the same application subscribed to by a client from the financial industry may have a tighter security or data privacy SLA threshold than a public subscriber.

The DMTF Architecture for Managing Clouds, A White Paper from the Open Cloud Standards Incubator [DMTF DSP-IS0102], provides a good overview of policy-based Cloud management. For the SLA management, we also recommend the consideration and concept of "SLA Continuum." Additional studies are required in this area.

The concept of SLA Continuum is that instead of managing SLA at per service component level, an SLA control point can be established by each provider that contains a Policy Decision Point (PDP) which publishes and negotiates SLAs across different Clouds as well as a Policy Enforcement Point (PEP) which provides enforcement and negotiation functions to ensure the business objectives are met by the underline resources.

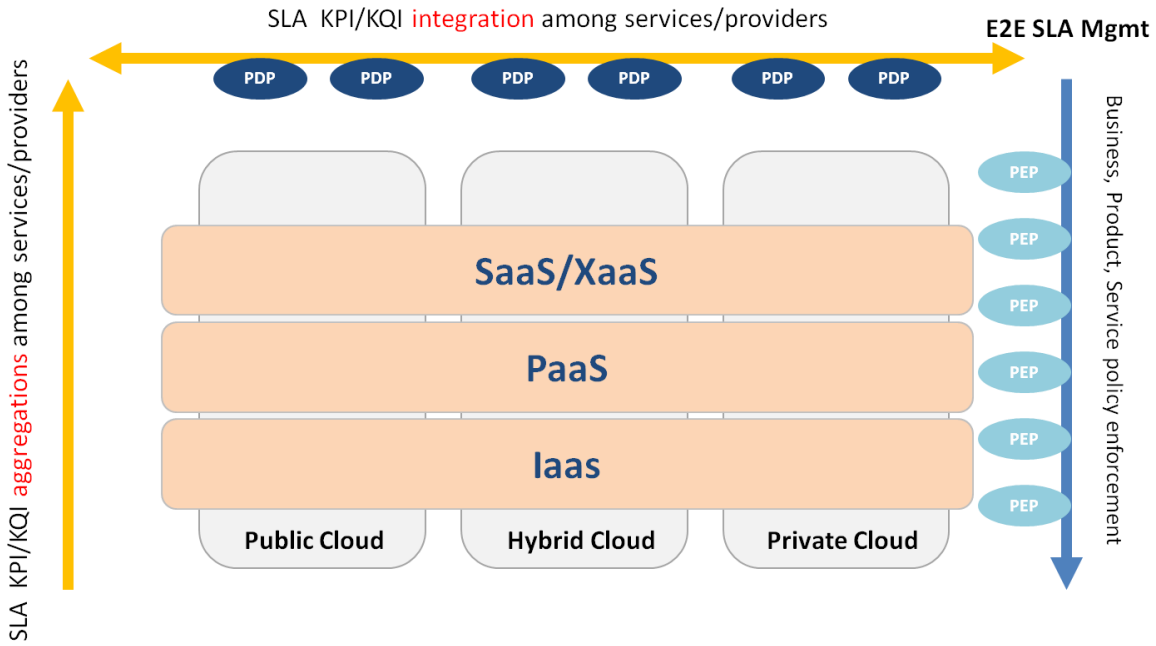

Figure 17 - SLA Continuum for the Cloud Ecosystem

5. Use Case

5.1 Premise

This scenario uses the same premise as the use case for the2012the 2012 TMW America's catalyst project entitled "MWA11-Multi-Cloud Development Experience Catalyst". In this scenario, a developer named Chris has an idea for a training video service.

Chris will create a multi-platform application for playing videos from a selection of training videos that she will produce. She owns no resources and will purchase services in the cloud to allow her to write her video player application. She will host her training videos in the cloud, her customers will download her application from the cloud, and her videos will be consumed via the cloud. This makes Chris a cloud service consumer, a cloud service developer, and a cloud service provider at different points in the product lifecycle.

This scenario assumes the following cloud service providers already exist in the cloud:

- **cWorld** is a cloud service provider and the entry point for consumers using cloud services. cWorld owns the business relationship with all consumers and cloud service providers as well as the Service Level Agreement (SLA), Cloud Service Model and Cloud Service Catalog. IT also provides all billing services.
- **cBroker** is a cloud service broker (API broker). All cloud service providers communicate with other cloud service providers and consumers via the broker. Whereas cloud brokers may provide additional roles in some cloud scenarios, in this scenario cBroker is only an API broker.
- **cAudit** provides auditing services for ensuring reported cloud service metrics meet service quality thresholds as defined by the SLA.
- **cNet** is a cloud carrier and network service provider. It provides variable speed bandwidth services in the cloud.
- **cStorage** is a cloud storage provider providing high volume storage to cloud service providers and consumers.
- **cData** is a direct competitor to cStorage.
- **cDev** provides a complete development environment in the cloud.
- **cSell** is a marketplace provider where cloud service providers can provide applications for consuming cloud services.
- **cEscrow** provides data backup facilities for mission critical cloud content.

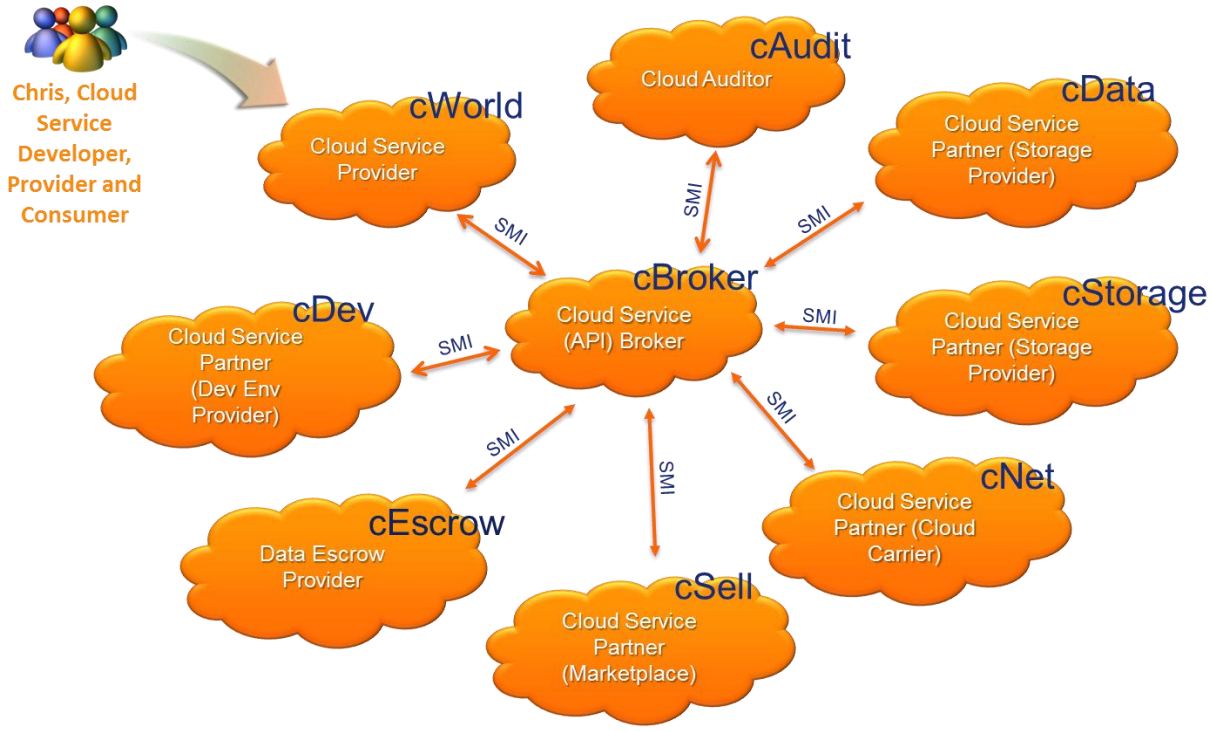

Figure 18 - The InterCloud

5.1.1 Scenario 1 – The Cloud Service Consumer

Chris researches available cloud offerings and decides that cWorld provides access to everything she needs to turn her idea into a reality. She browses the service catalog from the cWorld self-service portal, and purchases a virtual development environment from cDev, storage space from cStorage, and network bandwidth for her consumers from cNet. She also chooses cSell as a marketplace to provide her new multi-platform application. These companies were chosen because she was able to see the Quality of Service (QoS) attained by each of these service providers through a high-level QoS dashboard provided by cAudit.

After agreeing to the terms and conditions of the Master Service Agreement and providing a credit card for payment, Chris is provided with the services she has purchased, and is granted access to a customized dashboard provided by cAudit that allows her to monitor the quality of the services she has purchased, and how well they perform against the targets of the SLA she agreed to in the Master Service Agreement.

Chris proceeds to develop and test her application.

Consumers of Chris' service will have the option to download a video player through cSell to use her video service directly from a tablet, smartphone, or PC desktop. All videos will be hosted by cStorage and transmitted by cNet.

When her application is ready for production use, Chris uses the existing automated process (as described in section 1.3) to add her new service to the existing service in the cloud.

5.1.2 Scenario 2 – The Cloud Service Developer

Chris uses a web browser to request the service catalog template from cWorld. She is provided a service catalog template, a service model template, and an SLA template.

Chris analyzes the provided templates and decides where her service fits. She defines the service description for the service catalog, creates a service model for the services she will provide, and defines where in the global service model it should fit. She also defines the service levels she is willing to commit to in an SLA.

Where her services are dependent upon other service providers in the cloud, Chris notes those dependencies in her service model and SLA.

5.1.3 Scenario 3 – The Cloud Service Provider

Chris deploys her new cloud service by first requesting that her service be added to cWorld's service catalog.

Using a mechanism similar to OGF GFD.192 (as described in section 4.6), Chris submits a template to cWorld which is approved and her offering is added to the service catalog.

After successfully adding her offering to the service catalog she follows a similar process to submit her service model and SLA templates.

With the cloud service ready for use, Chris submits her application[s] to cSell for inclusion in the marketplace.

As consumers begin to use her service, Chris' consumer application and cloud service applications report QoS metrics to cWorld so her performance can be measured against established SLA levels.

After a short while of providing service, Chris notices that cStorage is unable to reliably maintain services up to the levels required by her SLA with them. This poor quality of service is impacting Chris' ability to provide a high quality service to her customers.

With a bit more research into the reports available from cAudit, Chris notes that cData has consistently higher levels of service than cStorage and she decides to switch service providers.

She copies her content from cStorage over to cData, updates her service's service model and SLA templates and renegotiates these new values with cWorld.

She continues providing service at a higher level than before, reevaluating constantly to ensure the highest quality of service possible.

5.1.4 Determine Cloud SLA

For this Use Case and any cloud computing initiative, a key sense of focus and awareness of where the stakeholders fit into the cloud ecosystem is paramount.

Perspectives change depending upon if you are the consumer or provider, and the specific objectives for such metrics collected and analyzed.

This paper summarizes the key concepts when prioritizing, defining, understanding and using metrics in a cloud environment.

Are you consuming or providing services?
Although this sounds straight forward, it's actually dynamic. Temptation is to discuss everything rather than focus on a precise situation and implementation.

For example: "Chris" is the Customer and consumes services from cWorld. "Chris" is also a "Provider" of training video services, where the end-user "student" is the Customer that consumes Chris services.

This Customer – Provider relationship flows through the entire cloud ecosystem.

For example: cBroker is the Customer/ Consumer – considering business objectives for "brokering", sets service expectations for the cNET Carrier Provider. The cBroker is also a Provider in the eyes of vWorld. In each case, this same general model can be applied to generate meaningful discussions.

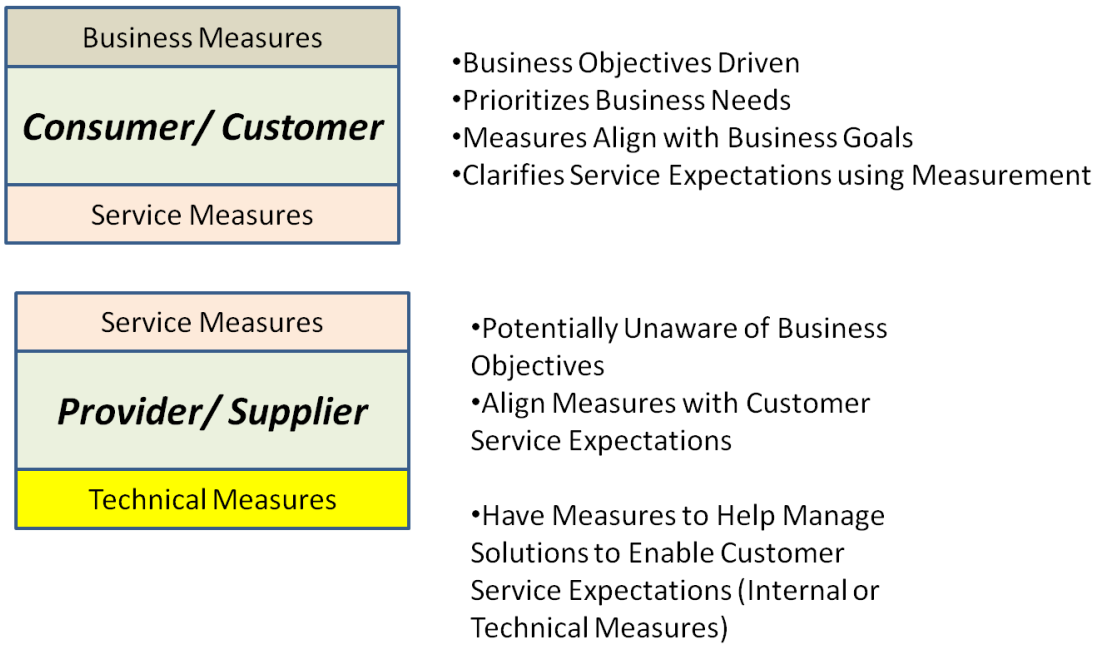

Figure 19 - Model of Customer and Supplier Measures

Consumers consider the business measures that can demonstrate business objectives are realized, they also can identify what service measures are appropriate to help guide their providers.

Providers consider how they are being evaluated by their customers/consumers and also what measures are needed to manage the internal solutions to ensure the service agreements are met.

Actionable Metrics of Value

Cloud Metrics must be valuable, leading to specific decisions and actions. This can be more manual, such as assessing the Total Cost of Ownership for a Cloud Solution, to more automated, such as performing auto-negotiation of services.

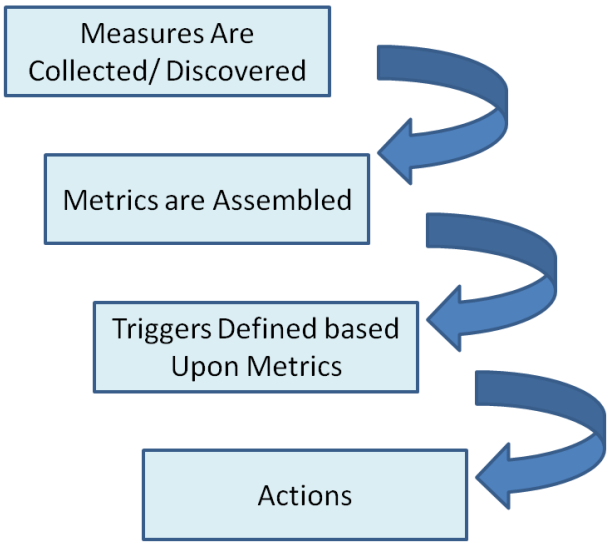

Figure 20 - Machine Driven Auto-Negotiation Scenario

Cloud Measurement General Models

ICT measurement is a complex topic area that is driven based upon goals and objectives. By understanding objectives, the metrics initiatives can help perform trend analysis and benchmarking and optimize performance. One size/measure approach does not "fit all," therefore the measures and metrics need to be structured for a particular organization.

The two models that follow help identify key considerations when establishing cloud metrics.

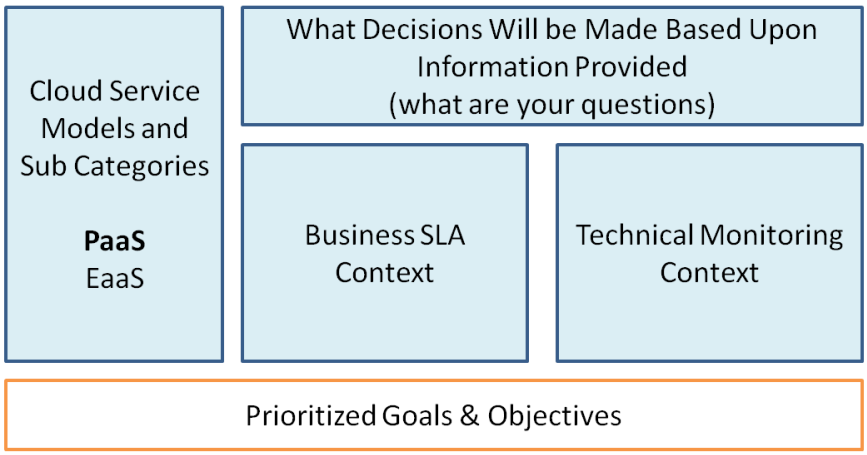

Figure 21 - General Model for Establishing Cloud Metrics

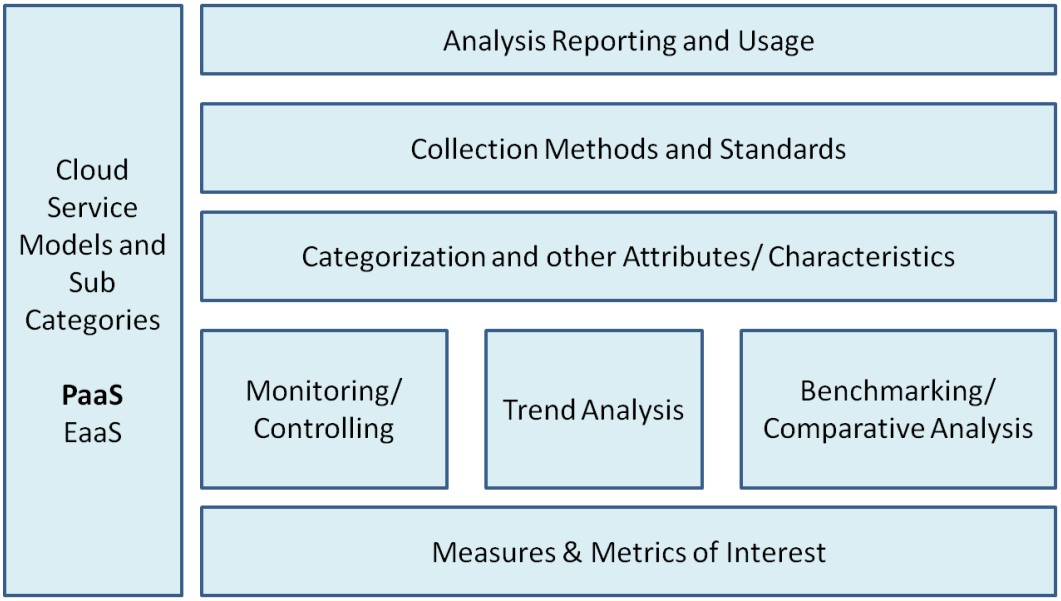

Figure 22 - Model and Sub Categories for Establishing Cloud Metrics

Example Cloud SLA Metrics

Operational Indicators

TM Forum GB963 provides extensive studies based on ECLC Enterprise grade external compute as a service (IaaS) requirements. It provides examples of enterprise KPIs for each in the context of several use cases; it also suggests a systematic approach for KQI identification based on the NIST reference architecture. These include the detailed explanation of the following:

Table 3 – Examples of Cloud Services Operations Indicators

Category	Important Characteristics[1]	Important KQIs[2]
Flexibility	Scalability[3]	Time to Provision New Service Increment Time to Adjust Existing Service Increment
Availability Management	Availability	Time available/total time x 100% per period
Performance Management	Performance Quality Capacity	Performance of defined transaction Conformance to documented requirements Load Testing
Change Management	Conformance of Change to Plan Testing	% rollbacks % of failures
Incident Management	Monitoring	Elapsed time by status and priority Maximum # of incidents by type/period
Problem Management	Ability to provide adequate data & communications about problems	Time to resolve problem, status updates % of jeopardy vs. incident notices by type (Availability, Capacity, Security, Penalties)
Continuity Management	Low Downtime Maintenance Disaster Recovery Data Protection	Number and frequency of backups Time to recover 0 Loss of Data

[1] In some SLAs an 'audit' or 'certification' is specified for one or more of the preceding areas annually or greater, but the team feels this is more accurately represented as a contract term rather than an element of an SLA, and is not changed for Cloud

[2] Implicit for all KQIs is the notion that they will be accurately and timely reported based on a negotiated period

[3] The idea of 'maximum' scalability appears either as a contract term or a Performance SLA term as 'load testing'

Resource Indicators
The Open Systems Group (OSG) Cloud Working group produced a set of Cloud Computing benchmarks.

These benchmarks focus on workload of the "system under study (SUT)" which comprises all components (cloud service, hardware, software, network connections within the SUT), and support services which are being tested by the cloud workload or required by the specific benchmark run rules.

The key metrics identified are:

- Elasticity, which consists of at least the following components
- Provisioning Interval
- Agility
- Scale-up/Down
- Elastic speedup
- Throughput
- Response Time
- Variability

Other relevant metrics include:

- Durability
- Reliability
- Power
- Price
- Density

6. Issues and Future Work Recommendations

Through the development of this Technical Report, we identified several key areas where standards or best practices need to be defined and adopted by the industry in order to enable service providers to provide Cloud SLA management end-to-end.

The following table outlines those recommendations and proposed actions. Most of the work will not be carried out directly by the current E2E Cloud SLA team, instead, Feature Requests/Change Requests will be issued to the appropriate TM Forum teams to tap into domain experts that are required for the standards specification in collaboration with other SDOs.

The E2E Cloud SLA team will continue focus on Cloud SLA metrics definition [R6] and SLA service modeling for the TM Forum release 13.5 (targeted release date Oct, 2013). The team will remain as a focal point to provide guidance and stewardship to ensure requirements highlighted in the **Architecture Considerations and Enablement** section are fulfilled.

Readers should consult the Charters of designated teams, listed on the right most column, for actual schedules and deliverables of the recommendation. In general, we also recommend to use prototype implementations/catalyst projects as way to accelerate the standards development.

Table 4 – Work Recommendation and Organizations Impacted

ID	Description	Priority	Dependency	Recommended Actions
[R1]	The industry can benefit from a single set of definitions and relationships of the followings: • Master Service Agreement • Cloud SLA • Business Level Objectives • Service Level Objectives • SLA metrics Recommend defining a taxonomy standard with mandatory items for Cloud contract that can be extended and customized per industry	H	N/A	Issue TMF Change Request: to: SLAM, SID teams Liaise with: NIST, CSCC
[R2]	Enhance TM Forum GB917 with stronger business considerations and to continue align its work with other related industry Cloud SLA activities.	M	[R1]	Issue TMF Change Request: to: SLAM team Liaise with: NIST, CSCC
[R3]	Enhance TM Forum GB917 to include roles to facilitate Cloud ecosystem	M	[R1]	Issue TMF Change Request:

	operations, including legal and regulatory compliance e.g. Cloud Auditor			to: SLAM team
[R4]	A standardized Service Template is needed in the industry to describe Cloud services consistently. This service template should contain the following minimum set of information • Service characteristic • Container of service metrics • Service dependency • Management dependency Recommend harmonizing TOSCA Service template and SID Service model, with the consideration of [R1]	H	[R1]	Issue TMF Change Request: to: SID and SLAM team Liaise with: OASIS/TOSCA
[R5]	A standardized SLA template that works in conjunction with [R4] Enhance GB917 SLA Template accordingly.	H	[R4]	Issue TMF Change Request: to: SLAM team Liaise with: OASIS/TOSCA DMTF and OGF
[R6]	Continue collect and rationalize Cloud metrics across industry bodies, provide recommendations on how to organize SLA models (per Cloud service models and deployment models)	M	[R5]	TM Forum E2E Cloud SLA Team Liaise with: all Cloud SLA metrics related organizations: CSCC, NIST, ISMA, ITU-T etc.
[R7]	Align and harmonize SMI and WS-Agreement with the consideration of harmonizing OGF/OCCI and WS-Agreement Intended effort is to produce standardized SLA management APIs	M	N/A	Issue TMF Change Request: to: SES team Liaise with: DMTF and OGF

[R8]	Architect and identify standards for Catalogue and federated catalogue management to support software factory based service delivery and Cloud ecosystem operations	M	[R7]	Issue TMF Change Request: to: Catalogue management team Liaise with: OASIS
[R9]	Architect and identify standards for policy based Cloud SLA Management	L	[R4,5,7,8]	Propose to TM Forum Product Architecture & Governance Working Group (PAGWG) for work consideration Liaise with: DMTF and ITU-T

7. Appendices

7.1 Appendix A

7.1.1 Abbreviations and Acronyms

Table 5 – Acronym and its Definition

Acronym	Definition
API	Application Programing Interface
BLO	Business Level Objectives
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CSCC	Cloud Standards Customer Council
CSP	Cloud Service Provider
DMTF	Distributed Management Task Force
ECLC	Enterprise Cloud Leadership Council
E2E	End-to-End
Frameworx	TM Forum frameworks
IaaS	Infrastructure as a Service
ICT	Information, Communications and Technologies
ISMA	International Software Measurement Association
ITIL	Information Technology Infrastructure Library
ITU-T	International Telecommunications Union, Telecommunication standards sector
KPI	Key Performance Indicators
KQI	Key Quality Indicator
MSA	Master Service Agreement
MWA	Management World America
NIST	National Institution of Standards and Technology
OASIS	Organization for the Advancement of Structured Information Standards
OGF	Open Grid Forum
OLA	Operations Level Agreement
OSG	Open Systems Group

PaaS	Platform as a Service
PDP	Policy Decision Point
PEP	Policy Enforcement Point
QoS	Quality of Service
RA	Reference Architecture
RAM	Random Access Memory
SaaS	Software as a Service
SDO	Standards Developing Organizations
SES	Software Enabled Services
SID	Shared Information and Data Model
SLA	Service Level Agreement
SLAM	Service Level Agreement Management
SLO	Service Level Objectives
SLS	Service Level Specifications
SMI	SES Management Interface
SOA	Service Oriented Architecture
SP	Service Provider
SUT	System under Study
TOSCA	Topology and Orchestration Specification for Cloud Application
TR	Technical Report
VPN	Virtual Private Network

7.2 Appendix B

7.2.1 References

Table 6 - References and Description

Reference	Description and Source
[CSCC SLA]	CSCC SLA White Paper http://www.cloud-council.org/04102012.htm
[DMTF DSP0243]	Distributed Management Task Force Inc., *Open Virtualization Format Specification.* January 12, 2010. From http://dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf
[DMTF DSP0263]	Distributed Management Task Force Inc Cloud Infrastructure Management Interface (CIMI) Model and REST Interface over HTTP: An Interface for Managing Cloud Infrastructure. September 14, 2011. From http://dmtf.org/sites/default/files/standards/documents/DSP0263_1.0.0a.pdf
[DMTF DSP0264]	Distributed Management Task Force Inc., *Cloud Infrastructure Management Interface - Common Information Model (CIMI-CIM):A CIM Representation of the CIMI model* September 8, 2011. From http://dmtf.org/sites/default/files/standards/documents/DSP0264_1.0.0a.pdf
[DMTF DSP2027]	Distributed Management Task Force Inc., *Cloud Infrastructure Management Interface (CIMI) Primer.* September 8, 2011. From http://dmtf.org/sites/default/files/standards/documents/DSP2027_1.0.0a.pdf
[DMTF DSP-ISO102]	Distributed Management Task Force Inc., *Architecture for Managing Clouds: A White Paper from the Open Cloud Standards Incubator.* June 18, 2010. From http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0102_1.0.0.pdf
[DMTF DSP-ISO103]	Distributed Management Task Force Inc., *Use Cases and Interactions for Managing Clouds: A White Paper from the Open Cloud Standards Incubator*. June 18, 2010. From http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0103_1.0.0.pdf
[ITU-T FGCC TR]	International Telecommunication Union. *Focus Group on Cloud Reporting Technical Report.* February 2012. From http://www.itu.int/en/ITU-T/focusgroups/cloud/Documents/FG-coud-technical-report.zip
[NIST CC Contract]	National Institute of Standards and Technology. *Draft Master Service*

	Agreement. And *Cloud SLA* From http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/RATax_Jan20_2012/NIST_CC_WG_ContractSLA_Deliverable_Draft_v1_7.pdf
[NIST CC RA]	National Institute of Standards and Technology. *NIST Cloud Computing Reference Architecture and Taxonomy (*NIST SP - 500-292) *Sept*, 2011. From http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909505
[OASIS TOSCA]	Lipton, Paul. *Topology and Orchestration Specification for Cloud Applications Version 1.0.* March 8, 2012. From https://www.oasis-open.org/committees/document.php?document_id=45509&wg_abbrev=tosca
[OGF GFD.192]	Andrieux, A., Czajkowski, K., Dan, A., et al. *Web Services Agreement Specification (WS-Agreement).* October 10, 2011. From http://www.ogf.org/documents/GFD.192.pdf
[OGF GFD.193]	Battré, D., Brazier, F., Clark, K., et al. *WS-Agreement Negotiation Version 1.0.* October 10, 2011. From http://www.ogf.org/documents/GFD.193.pdf
[OGF WSAG4j]	WS-Agreement reference implementation http://packcs-e0.scai.fraunhofer.de/wsag4j/index.html
[TMF GB917]	TM Forum Guidebook GB917 – "SLA Management Guidebook, Release 3.0". From http://www.tmforum.org/DocumentCenter/1722/home.html#TRCDocuments/artf3090
[TMF GB922]	TM Forum Information Framework (SID) - Party/party role, agreement, business interaction, service overview (SID Release 12 Addenda PDF package). From http://www.tmforum.org/components/community_download_file.aspx?dodown=true&did=frs2758&dtype=ContributionAttachment&dpid=artf2871&releaseID=rel2118
[TMF GB963]	TM Forum Cloud SLA App notes GB963. From http://www.tmforum.org/Guidebooks/GB963CloudSLAApplication/48297/article.html
[TMF TMF061]	Software Enabled Services Management Solution Reference Architecture (a.k.a. Service Delivery Framework Reference Architecture). From http://www.tmforum.org/TechnicalSpecifications/TMF061ServiceDelivery/39

	341/article.html
[TMF TMF167]	Software Enabled Services Management Interface (SMI). From http://www.tmforum.org/InformationAgreements/TMF617SoftwareEnabled/48632/article.html
[TMF TR149]	Customer Experience Management. From http://www.tmforum.org/KnowledgeDownloadDetail/9285/home.html?artf=artf2362

7.2.2 IPR Releases and Patent Disclosures

All work referenced and cited in this document, unless otherwise noted, the IPR belong to the respective publication organization.

This document may involve a claim of patent rights by one or more of the contributors to this document, pursuant to the Agreement on Intellectual Rights between the TM Forum and its members. Interested parties should contact the TM Forum office to obtain notice of current patent rights claims subject to this document.

7.3 Appendix C

7.3.1 SID Service Model Overview

Class Service is an abstract base class for defining the Service hierarchy. All Services are characterized as either being possibly visible and usable by a Customer or not. This gives rise to the two subclasses of Service: CustomerFacingService and ResourceFacingService.

The purpose of this entity is twofold. First, it is used to define attributes, methods, and relationships that are common to all Services. Second, it provides a convenient point to define how Services interact with other business entities.

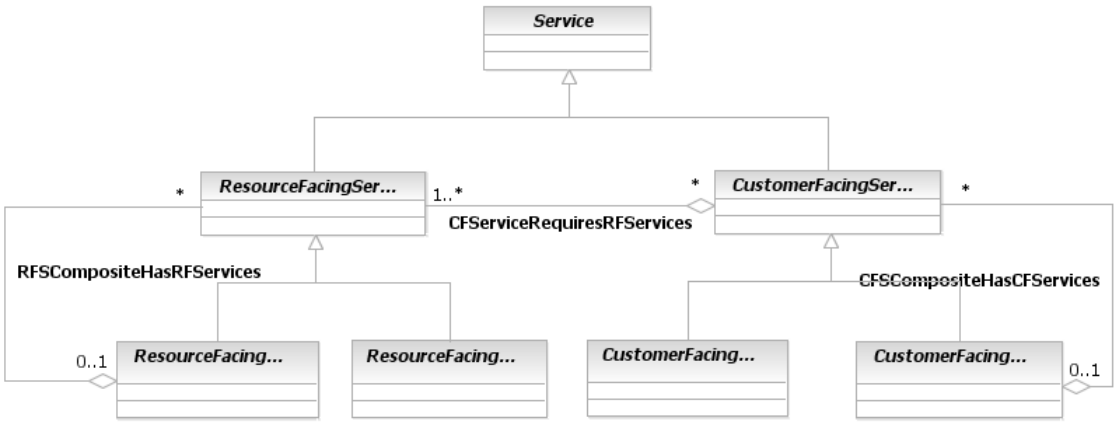

Figure 23 - The Service Model Classes

A Service represents the object that will be instantiated. Each Service instance can be different; therefore, Service is limited to owning just the changeable attributes, methods, relationships, and constraints that can be instantiated. The invariant attributes, methods, relationships, and constraints that can be instantiated are defined by a ServiceSpecification.

Services are defined as being tightly bound to Products. A Product defines the context of the Service. The Service and its related entities (e.g., ServiceSpecification, ServiceRole, and so forth) are related to entities in the Resource, Product, and other domains through a set of relationships.

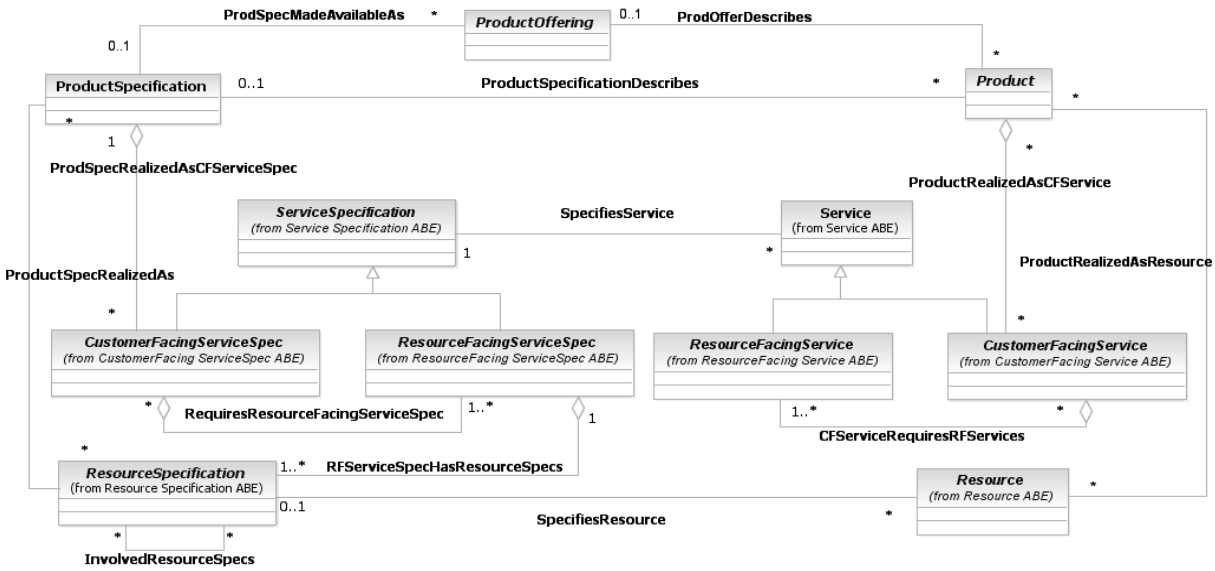

Figure 24 – Relationships between Product, Service and Resource Entities

From this diagram one can also see how different characteristics of the Service and Service Specification are mapped, or related to the Resource parameters.

The next diagram illustrates the relationship between the Service and Location, where the Service is or will be available.

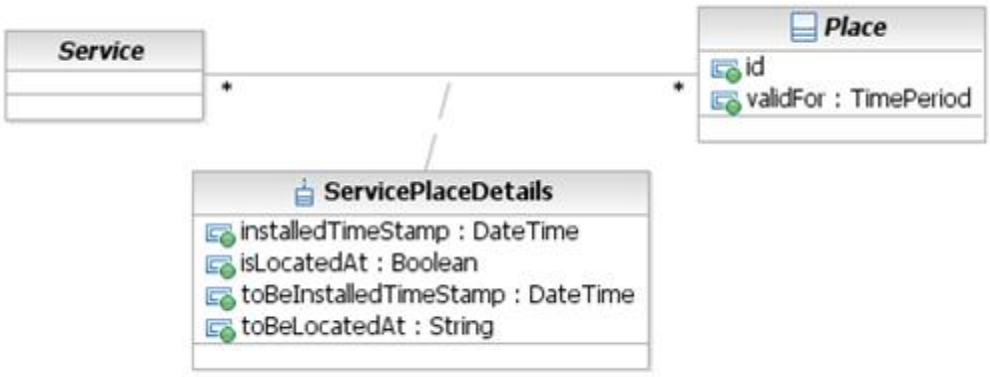

Figure 25 – Service and Location

The same Service can play different roles in the system. The roles are then used to characterize the functionality of the Service, regardless of whether it is a resource- or a customer-facing service.

Representing a Service in terms of ServiceRoles enables the functionality of the Service to be defined independently of BusinessActor, PhysicalResource, LogicalResource, or other Services.

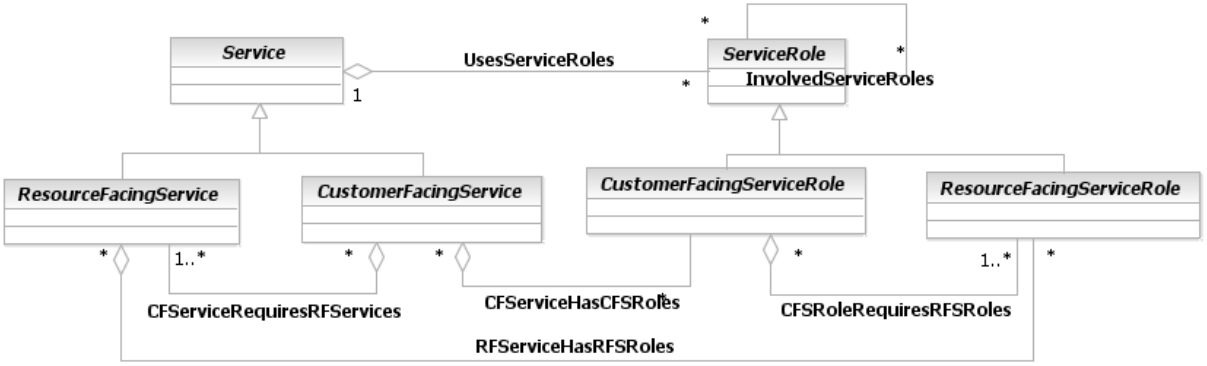

Figure 26 – Service Roles

In the Cloud environment it is extremely important to have the ability to extend the service model dynamically and yet maintain the compatibility of the management interface through the number of services with different characteristics, changing over time. The use of the CharacteristicSpecification/CharacteristicValue pattern allows for such functionality.

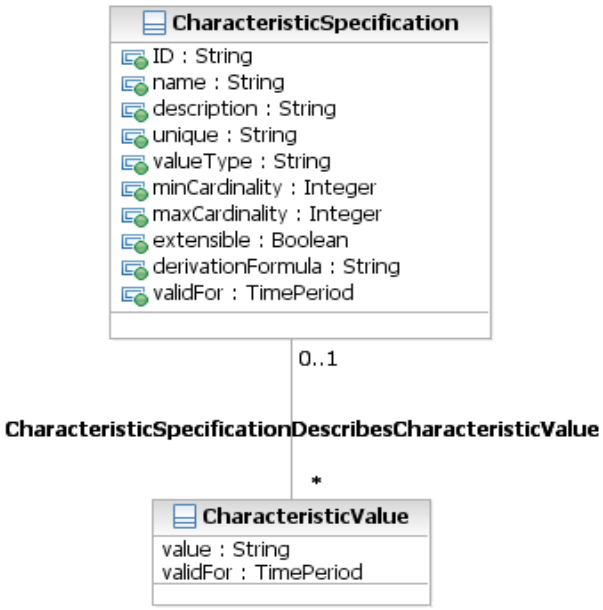

Figure 27 – Characteristic Specification/ Characteristic Pattern

The next diagram illustrates how this pattern can be applied to defining the service parameters. It also illustrates how the dynamically defined service characteristics can be linked with the product characteristics.

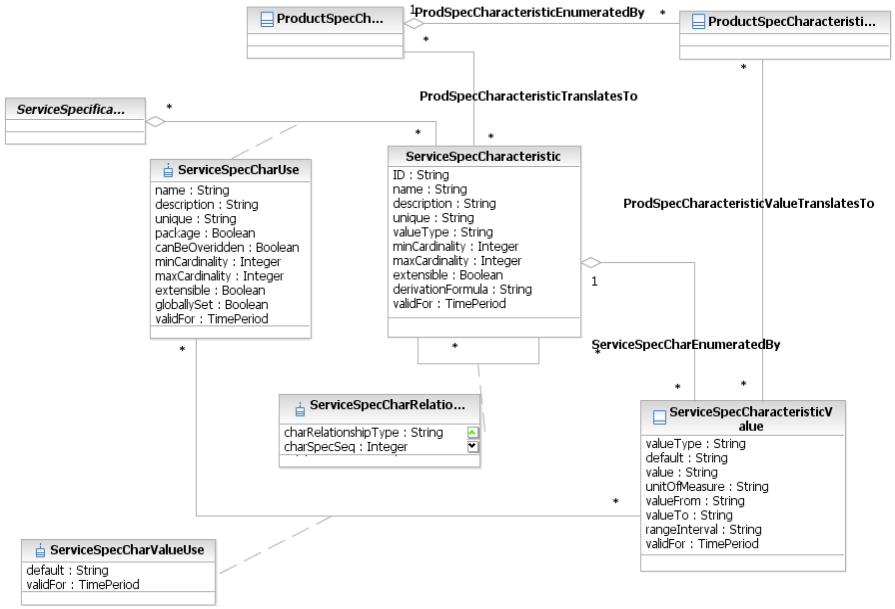

Figure 28 - Relationships between ServiceSpecCharacteristics and Product Specification and its Characteristics

The following diagram depicts the linkage between Service characteristics and the parameters of the resources supporting the service.

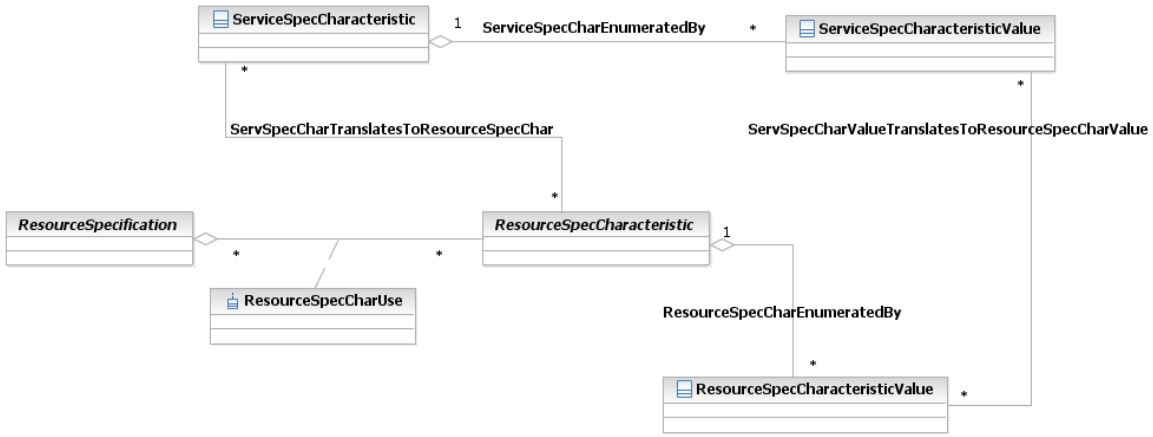

Figure 29 - Relationships between ServiceSpecCharacteristics of Supporting Resources

7.3.2 SID SLA/SLO Model Overview

The SID model also defines the number of classes that can be used to define the basic Service Level Objective model.

Figure 30 - Modeling of the SLO in SID

The ServiceLevelObjective entity defines the quality goal for a ServiceLevelSpecification in terms of parameters and metrics, thresholds, and tolerances associated with the parameters.

The class ServiceLevelSpecConsequence models an action that takes place in the event that a ServiceLevelObjective is not met.

The next diagram illustrates the relationships between ServiceSpecification, Service Level Specification and the pair of ProductOffering and ProductSpecification entities.

Figure 31 - SLS and Product Specification

The ServiceLevelSpecificationParameter on the next diagram is used to specify the SLO parameters and determine compliance with a ServiceLevelObjective.

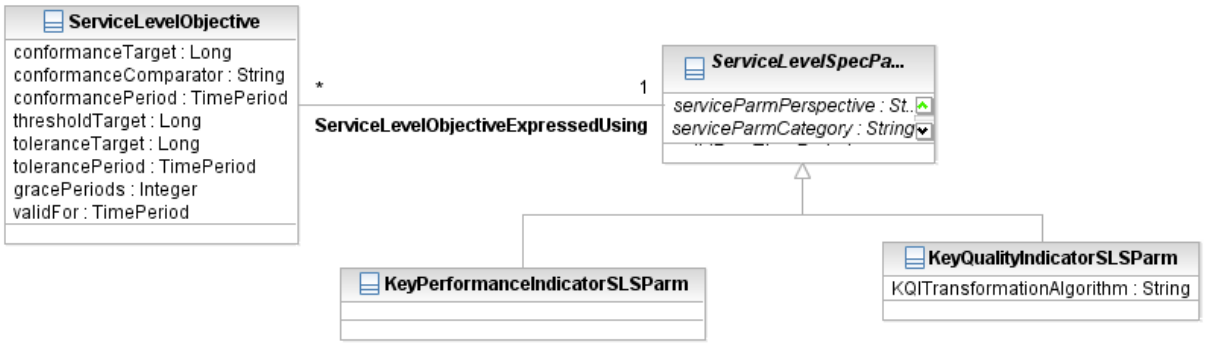

Figure 32 - Modeling of the SLO parameters in SID

The SLP Parameter has two subclasses: the KeyPerformanceIndicatorSLSParam and KeyQualityIndicatorSLSParam.

The KPI SLS Parameter represents a measure of a specific aspect of the performance of a ServiceResource (network or non-network) or a group of ServiceResources of the same type.

The KQI SLS Parameter is a measure of a specific aspect of the performance of a product (ProductSpecification, ProductOffering, or Product) or a service (ServiceSpecification or Service). A KQI draws its data from a number of sources, including KPIs.

On the next diagram the ServiceLevelSpecConsequence class represents an action that takes place in the event that a ServiceLevelObjective is not met.

Figure 33 - Service Level Specification Consequences

And finally, the ServiceLevelSpecApplicability class is used to represent the time of day or days during which a ServiceLevelSpecification, ServiceLevelObjective, or ServiceLevelSpecConsequence is relevant or not.

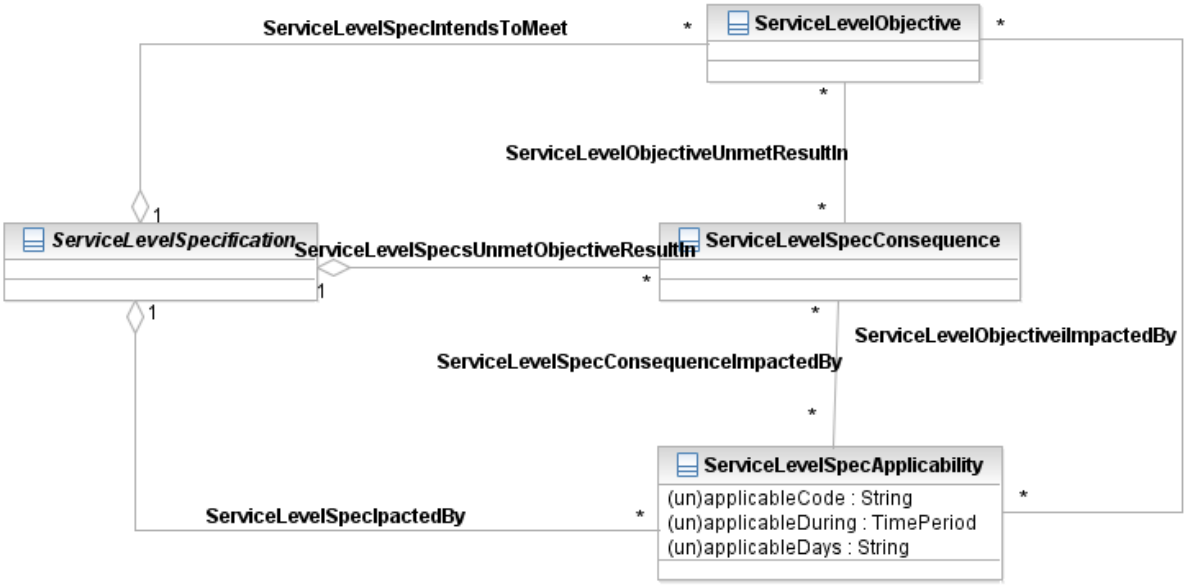

Figure 34 - Modeling of the Service Level Specification Application

7.4 Administrative Appendix

This Appendix provides additional background material about the TM Forum and this document. In general, sections may be included or omitted as desired, however a Document History must always be included.

7.4.1 Document History

Version History

<This section records the changes between this and the previous document version as it is edited by the team concerned. Note: this is an incremental number which does not have to match the release number and used for change control purposes only>

Version Number	Date Modified	Modified by:	Description of changes
0.1	Aug. 30th 2012	Jenny Huang	First completed draft
0.2	Sept. 24 2012	Jenny Huang	Incorporated team review comments and ready for voting for team approval
0.3	Oct 4th 2012	Alicja Kawecki	Minor cosmetic corrections prior to web posting and Member Evaluation
0.4	Dec 6th 2012	Alicja Kawecki	Updated Notice and footer

Release History

< This section records the changes between this and the previous Official document release. The release number is the 'Marketing' number which this version of the document is first being assigned to >

Release Number	Date Modified	Modified by:	Description of changes
<<Release Number >>	DD/MMM/YY	<<name>>	Description e.g. first issue of document

7.4.2 Acknowledgments

This document was prepared by the members of the **TM Forum End-to-End Cloud SLA Management – collaboration across standards organizations team**, including:

- Jenny Huang, AT&T, Co-chair and Editor, CSCC liaison
- Steve Woodward, Cloud Perspectives, ISMA, NIST and ITU-T liaison
- Alex Zhdankin, Cisco, DMTF liaison
- Tayeb Ben Meriem, FT/Orange
- Victor Havard, IBM, co-chair
- Dave Milham, TM Forum

Additional input was provided by the following people throughout the course of the project:

- Evelyne Roch, Ciena
- Elaine Haher, Ericsson
- Charlie Hale, IBM
- Eric Troup, Microsoft
- Ramin Yahyapur, Open Grid Forum liaison, Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen,
- Alan Sill, , Open Grid Forum, Texas Tech University
- Wolfgang Ziegler, Open Grid Forum, Fraunhofer-Institute for Algorithms and Scientific Computing (SCAI)
- Raihan Wadud, Saudi Telecom
- John Wilmes, Syntologica, OASIS liaison