

# A Compendium on Service Oriented Architecture and Service Level Agreements

Ravishankar. V

SAP ERP Consultant

Research Scholar – Bharathiar University  
Coimbatore, India

R. Baskaran

Department of Computer Science, CEG Guindy,  
Anna University Chennai  
Chennai-600 025, India

## ABSTRACT

Service Oriented Architecture provides an effective application integration and on-demand services provision. It supports loose coupling structure of the services which provides effective service management and also effective usage of the existing services. SOA gives strong support to business Innovation if implemented properly with proper understanding of the Principles of SOA. In order to incorporate the effective SOA, there should be a thorough deployment of the SOA, the SLA metrics needs to be standardized and focused. Since, the reliability of any service depends on the successful functionality. This article studies and reviews on a broad spectrum the Service Oriented Architecture and its Service Level Agreement (SLA) and its basic metrics.

## Keywords

SOA, SLAs of SOA, Metrics of SLA in SOA, SLA Mon Tools

## 1. INTRODUCTION

The SOA is an application architectural concept that defines the use of services to support business requirements. In SOA, the resources are made available as independent services, which can be accessed in a standardized manner. The Services are loosely coupled in order to facilitate the usage of the existing services and the components are reusable.

In SOA environment, the Service Level Agreement (SLA) refers to mutually agreed understandings and expectation focused as the evaluation criteria for Quality of Service, between the Service Provider and Service Consumers. As the Services become more flexible, the Quality of Services becomes more critical and important. Every user is concerned about the Quality of Service. Thus there is a huge necessity of defining the QoS Metrics during the contract between the Service Providers and the Service Consumers, which would prove useful in making the Service transaction execute smoothly and efficiently [1]. More and more Services are done in order to create larger business process and services with the SOA developments [2], [3]. This initiates a closer examination of the quality of the services. The traditional mechanism makes it difficult to adapt to the complexity in open networked environment.

The SLA is the contractual basis between the customer and the Service Provider about the Product. The Service of SOA is managed by SLA hence the Service Management can be achieved by the effective SLA management. In a Complex network environment requirements of end-to-end Quality of Service are different. The use of SLA guarantees in product

can make the network more customer-focused. Further SLA based services are becoming a requirement to the provisioning of IP based network services to ensure Quality of Service.

The Technical component of SLA is made up of a number of negotiated Service Level Objectives (SLO) which is based on Key Performance Indicators (KPI) metrics and Key Quality Indicators (KQI).

## 2. EXTENDED SERVICE ORIENTED ARCHITECTURE

SLA is a negotiated basic agreement between two parties (Service Provide and Consumer) which is designed to create a common understanding about the services, the priorities, responsibilities, etc Service Provider uses SLA to provide Service with a certain level of Quality and to charge customer – [4].

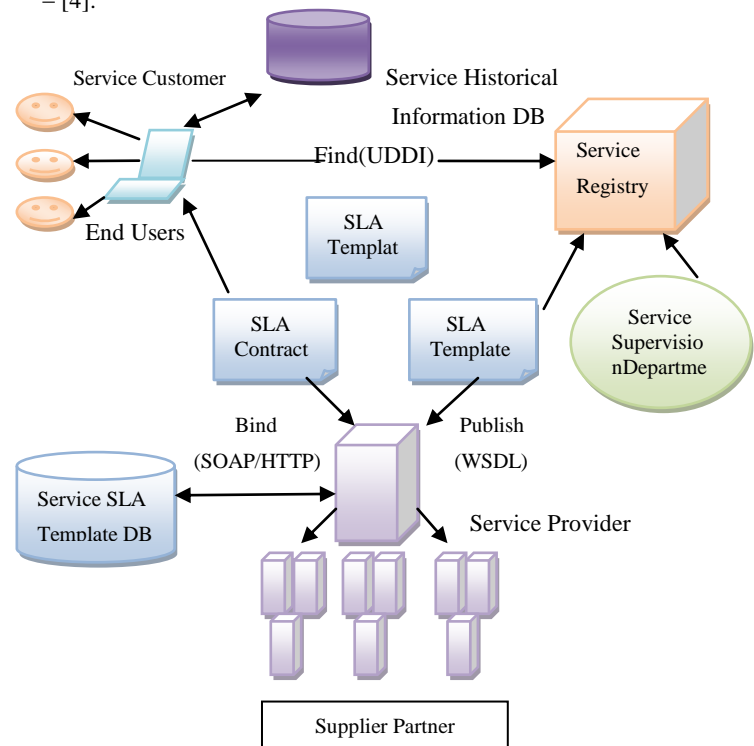


Fig 1: Extended Service Oriented Architecture.

The extended SOA, has an extension to the basic SOA architecture based on the SLA by adding service supervision department, services historical information database, SLA template database, SLA management module, and end – users/suppliers/partners roles. Its aim is to assure service

requester of discovering and binding services to act more fast and thoroughly, and help service provider to compose and offer services more effectively in service providing a chain [5]. The architecture is shown above in Figure 1.

### 3. SERVICE MANAGEMENT PROCESS



Fig 2: Service Management Process

Figure 2 shows the whole process of transforming the actual requirements into policies.

The first stage is to analyze the Customer requirements, based on which the Service Level Requirements are identified. After which the SLA metrics are derived which are expected to be delivered. Based on the SLA derived the SLO (Service Level Objectives) are derived, which specifies the Availability, Response time, Security, etc. The Objectives are then interpreted into metrics which focuses on the right metrics based on the management task on hand. The final stage includes the derivation of the Policy based management to implement/enforce the SLAs. The following content provides an overview of Service Process management architecture and Service Management process based on the Extended Service oriented architecture [5].

The SLA based SOA, service management process is introduced in terms of service process management architecture. This process reveals the dynamic subscription and change of SLA contract by leveraging closed-loop mechanism [6]. It thus reveals the service degradation and the violation of service via the result by the following:

Service result charges the customer/consumer to adjust the service to republish. The Service Consumer subscribes the SLA with Service Provider. SC and SP comes to an agreement with the content and the price of service. The SLA contract uses the best practices provided by the Service Provider or customized by the requirements of Service Consumer. During the Service Implementation, the Service Provider monitors the service parameters to evaluate the service degradation and consumer violation and then consider self healing, service re-providing and SLA re-subscription.

The Service Management Process can be defined into three phases: Service Provisioning Phase, Service Implementation Phase and Service Summary Phase.

#### 3.1 Provisioning Phase:

During this phase, the Service Publisher and Service Composer subscribes the SLA contract to confirm the responsibility after undergoing various process like matching the Service kinds, Service Quality, SLA template, SLA level

basic, Services adopted and also after evaluating the historical information base. And during this phase, the judging is done based on the negotiation of the Service Functionality, Service Features, Service Quality, Cost etc. During the provisioning phase, the following are also considered: Content of Service providing, Scope, Service Quality level, Customer Violation Judgment, Service degradation judgment and penalty for service failure or Service cancel.

#### 3.2 Implementation Phase:

The Service Implementation phase direct in three results: Service Failure, Service Degradation and Service Complete. If there is Service Failure/Service degradation, then there is a resume service which is introduced in order to recover from the above failure. During the Service Failure, there are three basic things to be verified: Service recover, SLA contract re-subscribed with the SP and search of another SP to regain the same service. Once the Service implementation is completed, the following are done: Comparison of SLA contract with Service Reports, Identify the service degradation, Customer violation, etc.

#### 3.3 Summary Phase:

The SC is charged according to the result of the judgment of the SLA and the negotiation is done with the suppliers and the partners. After the completion of the Service process, the Published Service and the SLA templates are adjusted in terms of Problems emerging from the Service process and the SR is updated.

### 4. SLA ENFORCEMENT

SLA proposals provide wide way to specify the set of Functional and Non-functional Properties that must be guaranteed during the Service provision [7].

#### 4.1 Non-Functional Properties

The Non-functional properties can be classified into two. They are Controllable Properties and Non-controllable properties.

##### 4.1.1 Controllable Properties:

The Service Providers can control the values that the properties present by performing actions to change it. The value of this property should be continuous and dynamic in nature based on the context of the transaction with a valid Service Level Agreement [7].

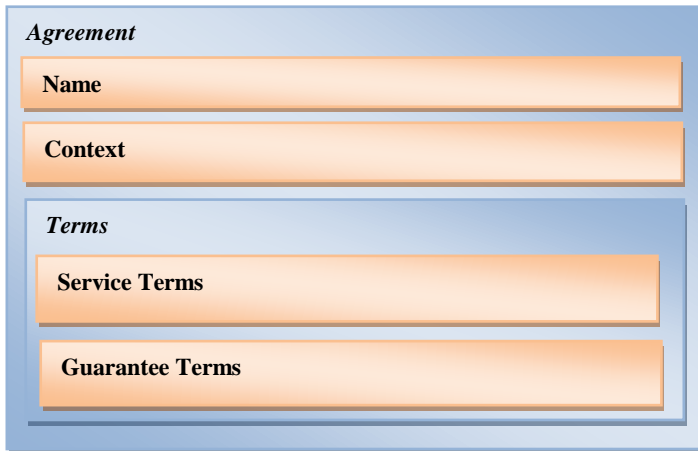
##### 4.1.2 Non-Controllable Properties:

The Service Providers cannot control the values that the properties present. These are the attributes of the provider which are default and are expressed in the SLA. It can also be defined as non-measurable properties where there is no control over its values i.e. they cannot be quantified [7].

#### 4.2 Functional Properties:

The Functional Properties comprises of Service that launches the process of the main frame of job execution system. These would be primary properties of each service working as a backbone of the Services.

## 5. FRAMEWORK FOR THE DYNAMIC NEGOTIATION OF SLAS



**Fig 3: Structure of SLA agreement:**

An agreement between a Service Provider and Consumer specifies one or more Service Level Objectives focusing on the requirements of the Service Consumers and the provisions from the Service Provider based on the availability of the resources and the Service qualities [8-10]. The structure of the agreement is similar to the Figure 3 given above. The above structure gives a picture of the agreement parties and the set of terms. The terms represent the contractual obligations and include a description of the service as well as the specific guarantees given. The description can be reference to the existing service or a domain specific description of a service or the properties of a service. The term specifies the non-functional characteristics in the service level objectives as an expression over properties of service. It is like an optional qualifying condition under which Objectives are to be met. This would also add up the business importance in meeting these objectives.

## 6. METRICS

The Metrics are used in order to monitor the performance of services which are specified in the SLA. It works as a means of defining the Service Levels, tracking adherence to those Service levels and support the improvement of the Services consistently. The terms used for measurements, values derived from the measurements and service thresholds vary from SLAs. The terms are to be defined clearly in the SLA. The terms vaguely clarified here are Measure, Metric, Threshold, Objectives [11].

Measure, is a value from the physical measurement, like a single instance of a response time, the utilization of system over a specific period of time, etc.

Metric, is a quantitative measure of the degree to which a system, component, or process possesses a given attribute. This could be composed based on the measures. This should include the clear definition of the underlying measurement, the formula and the method for deriving the metric including the timeframe, the units used for the calculation.

Threshold is a limit of the acceptable value range of a metric.

Objective, is the desired value of a metric, this value depends on the threshold limit specified.

The metrics are classified to be well defined, if

- The metric is measurable
- The metric is calculated with clarity on where and how the measurements are collected, its algorithms, the format of it and the frequency of collection.
- The metric should include the expected end result which would satisfy the expected level of performance.
- The metric should support the customer requirements including compliance issues or continuous improvement of the service.
- The metric should focus on the efficiency and effectiveness of the process
- The metric should support the statistical analysis
- The metric applies the appropriate industry or other standards
- The metric should include the assumptions and definitions used in gathering, computing and evaluation the service.

In order to consider the SLA factors, the complete understanding on the flow pattern of the system is required. This would enable the limiting of the conditions. In order to understand the flow pattern of the system, the clear understanding of the System architecture is also required, which would give clarity on how the information is exchanged between components and from where the data can be collected.

In order to frame the SLA factors as discussed above, it is always better to define SLA with well defined smaller number of metrics would be best, but keeping in mind that for the sake of simplicity, the important metrics are not omitted.

## 7. KEY PERFORMANCE INDICATORS

The Services needs to satisfy the three common Key Performance Indicators [12].

### 7.1 Availability:

The availability indicator of the service specifies the service availability, during the period of when the service was invoked and the total duration of the time interval (Until it is finished). Without proper availability of service at the required time, the Service does not become efficient or useful. Hence, this proves to be one of the most KPI for a Service.

### 7.2 Response Time:

The response time is the duration of when the Service was requested and when the request was responded [13]. This also acts as one of the most important KPI for a Service.

### 7.3 Throughput:

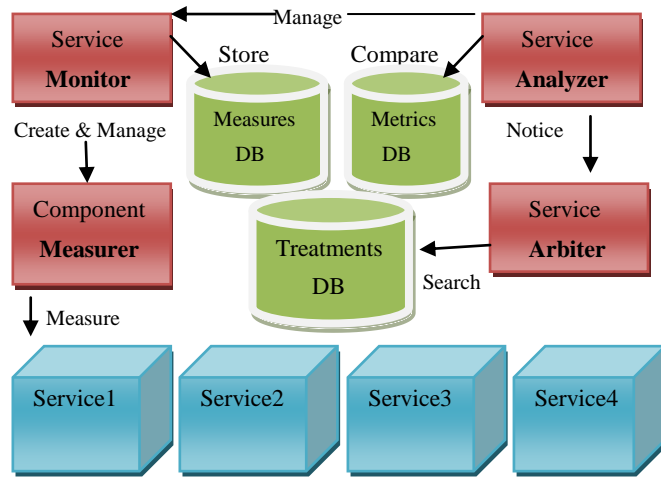
This specifies the service invocations that can be processed by the system within a given interval of time. For example, the throughput means the number of requests that can be processed within a specific duration. Throughput also is important KPI which goes hand in hand with the Availability and response time for better Service reliability.

## 8. SLA MONITORING

SLA monitoring is required in order to monitor the SLA metrics which are defined or derived.

### 8.1 SLA Monitoring Process 1

SLA Monitoring is required in order to monitor the SLA metrics which are defined or derived [12].



**Fig 4: SLA Monitoring Process 1**

SLA monitoring in Figure 4 could comprise of three phases: Monitor, Analyzer and Arbiter.

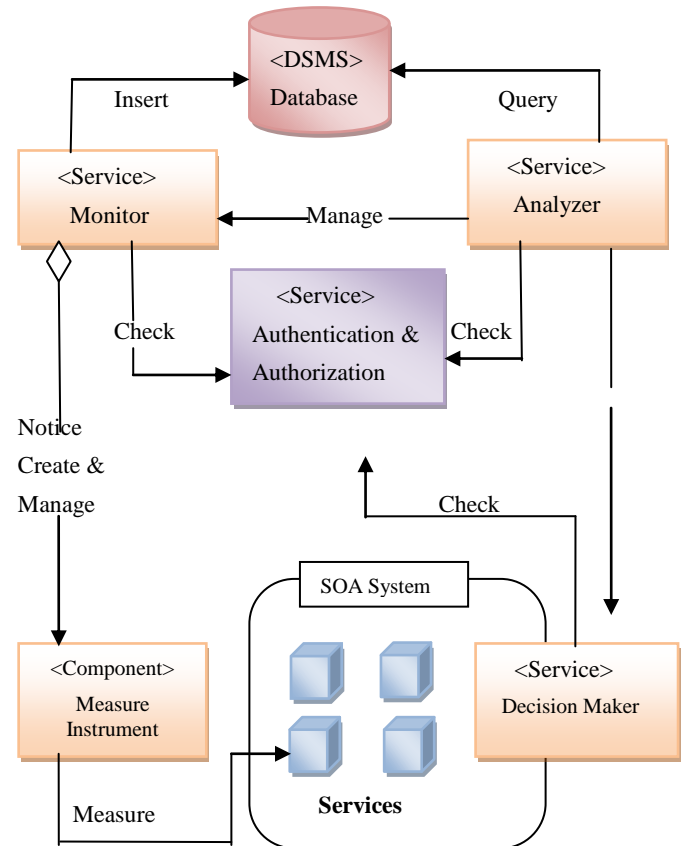
The Monitor service has the measuring component which has the responsibility to maintain the information updated. These are components used to get all performance data of the quality attributes of SLA.

The Analyzer checks for SLA violations by managing the Monitors. *This* will also receive the attributes collected by the Monitor which is compared with those metrics and indicators stored in a database. Any discrepancies or violation found will be notified to the Arbiter.

The Arbiter selects and find the treatment in order to solve the Violations detected by the Analyzer. It has predefined treatments in order to solve the violations detected by the Analyzer.

### 8.2 SLA Monitoring Process 2

This architecture of SLA Monitoring comprises of phases: Monitor, Decision Maker and Analyzer. The below architecture differs a bit compared to the previous one, where the third phase is Decision Maker Phase, instead of the Arbiter Phase that does a slightly different task / process [14].



**Fig 5: SLA Monitoring Tool SALMon Architecture**

Figure 5 describes the second model of SLA monitoring tool. Here the Decision maker is a similar tool like the Arbiter in the previous model [14]. Here the Decision maker goes a step further in providing alternative service which can be provided in addition to solving the violations detected by the analyzer.

## 9. CONCLUSION

In this article, the review of the SOA-SLA metrics and the SLA Monitoring tool were presented. The SLA metrics focusing on the Functional and Non-functional aspects of the Architecture and the basic KPIs of the Service were also presented. The objective and the future research should focus on the SLAs drilling down a bit, which focuses on the Architecture / Process as well in more detail so as to accomplish a more successful Service Oriented Architecture paradigm. Since, the future is building up on more of Service Orientation with the utmost throughput and efficiency expected, with a reliable architecture for a stable environment.

## 10. REFERENCES

- [1] Ke Xu, Xiaoqi Zhang, Meina Song, Junde Song. 2009. "Research on SLA management model in Service Operation Support System", IEEE.
- [2] Faramarz Safi Esfahan, Masrah Azrifah Azmi Murad. 2009. "SLA-Driven Business Process Distribution", IEEE.
- [3] Wenhui Sun, Jinyu Zhang, Feng Liu. 2006. "WS-SLA: A Framework for Web Services Oriented Service Level Agreements", IEEE.

- [4] Lu Liu, Wen-an Zhou. 2009. ICT&SSME Center, Beijing University of Posts and Telecommunications, Beijing, P. R. China - “A Novel SOA-Oriented Federate SLA Management Architecture”, International Symposium on Information Engineering and Electronic Commerce.
- [5] Zhang Nan, Xue-song Qiu, Luo-ming Meng. 2006. “A SLA-Based Service Process Management Approach for SOA”, IEEE.
- [6] Philip Bianco, Grace A. Lewis, Paulo Merson. 2008. “Service Level Agreements in. Service-Oriented Architecture. Environments.” TECHNICAL NOTE.
- [7] Jose Antonio Parejo, Pablo Fernandez, Antonio Ruiz-Cortés, Jos’e Mar’ia Garc’ia. 2008. “SLAWs: Towards a Conceptual Architecture for SLA enforcement”, IEEE.13
- [8] Giuseppe Di Modica, Valerio Regalbuto, Orazio Tomarchio, Lorenzo Vita. 2007. “Dynamic re-negotiations of SLA in service composition scenarios”, IEEE.
- [9] Farhana Zulkernine and Patrick Martin, Chris Craddock, SVP, Kirk Wilson, 2009. A Policy-based Middleware for Web Services SLA Negotiation, IEEE International Conference on Web Services.
- [10] Giuseppe Di Modica, Valerio Regalbuto, Orazio Tomarchio, Lorenzo Vita. 2007. “Enabling Re-negotiations of SLA by Extending the WS-Agreement Specification”, IEEE.
- [11] Susan McPherson, Jon Curtiss, Mukesh Rohatgi, Michael Ballotti, Suzanne Beers, Burt Liebowitz, Patrick Van Metre. 2010. “Recommendations for Enterprise Service SLA Guidance in the DoD, The 2010 Military Communications Conference.
- [12] Frank Schulz. 2010. “Towards Measuring the Degree of Fulfillment of Service Level Agreements”, Third International Conference on Information and Computing, IEEE.
- [13] Nicolas Repp, Andr’e Miede, Michael Niemann, Ralf Steinmetz. 2008. “WS-Re2Policy: A policy language for distributed SLA monitoring and enforcement”, IEEE.
- [14] David Ameller, Xavier Franch. 2008. “Service Level Agreement Monitor (SALMon)”, Seventh International Conference on Composition-Based Software Systems, IEEE.