Project no. FP7- 216556

Instrument: Integrated Project (IP)

Objective ICT-2007.1.2: Service and Software Architectures, Infrastructures and Engineering

Deliverable D.A2a
Business SLA Management
(full lifecycle)

Keywords:
Service Level Agreement, Service-Oriented Infrastructure, Business SLA

Due date of deliverable: 31th July 2010

Actual submission to EC date: 10th September 2010

Start date of project: 1st June 2008

Duration: 38 months

Lead contractor for this deliverable: Telefónica Investigación y Desarrollo

Revision: V.1.1 (9th September 2010)

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)

<table>
<thead>
<tr>
<th>Dissemination level</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
</tr>
<tr>
<td>Public</td>
</tr>
<tr>
<td>Yes</td>
</tr>
</tbody>
</table>
Document Status

Deliverable Lead: TID
Reviewer 1: John M. Kennedy
Reviewer 2: Sylvia Scheu
PMT Reviewer: Wolfgang Theilmann
Complete version submitted to reviewers: 7th July 2010
Comments of reviewer 1 received: 21st July 2010
Comments of reviewer 2 received: 3rd August 2010
Revised deliverable submitted to PMT: 6th August 2010
PMT Approval: 10th September 2010

Contributors

<table>
<thead>
<tr>
<th>Partner</th>
<th>Contributors</th>
</tr>
</thead>
<tbody>
<tr>
<td>TID</td>
<td>Juan Lambea, Beatriz Fuentes, Sergio García</td>
</tr>
<tr>
<td>PMI</td>
<td>Valentina Mazza</td>
</tr>
<tr>
<td>UDO</td>
<td>Edwin Yaqub</td>
</tr>
<tr>
<td>CITY</td>
<td>George Spanoudakis</td>
</tr>
</tbody>
</table>

Notices

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. Copyright 2009 by the SLA@SOI consortium.

* Other names and brands may be claimed as the property of others.

This work is licensed under a Creative Commons Attribution 3.0 License.
## Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>18/03/2010</td>
<td>Juan Lambea</td>
<td>New structure with common chapters</td>
</tr>
<tr>
<td>0.2</td>
<td>21/05/2010</td>
<td>Juan Lambea</td>
<td>Updated with the new chapter</td>
</tr>
<tr>
<td>0.31</td>
<td>10/06/2010</td>
<td>Sergio García, Beatriz Fuentes, Juan Lambea</td>
<td>Updated sections 2.2, 3, 4.2, 6 and 7.</td>
</tr>
<tr>
<td>0.32</td>
<td>21/06/2010</td>
<td>Sergio García, Juan Lambea</td>
<td>Updated sections 2.1, 3, 4, 5</td>
</tr>
<tr>
<td>0.4</td>
<td>22/06/2010</td>
<td>Valentina Mazza</td>
<td>Updated section 6.2.2</td>
</tr>
<tr>
<td>0.5</td>
<td>22/06/2010</td>
<td>Edwin Yaqub</td>
<td>Updated sections 2.1, 6, 7.1</td>
</tr>
<tr>
<td>0.61</td>
<td>24/06/2010</td>
<td>Beatriz Fuentes</td>
<td>Section 4.3</td>
</tr>
<tr>
<td>0.62</td>
<td>25/06/2010</td>
<td>Beatriz Fuentes</td>
<td>Sections 8.2, 8.3.2, Adjustment subsection in 4.4</td>
</tr>
<tr>
<td>0.7</td>
<td>28/06/2010</td>
<td>Juan Lambea</td>
<td>Sections 5, 7 updated</td>
</tr>
<tr>
<td>0.8</td>
<td>05/07/2010</td>
<td>Juan Lambea in behalf of George Spanoudakis</td>
<td>Updated chapter 7 and section 2.1, and 8.1</td>
</tr>
<tr>
<td>0.91</td>
<td>05/07/2010</td>
<td>Sergio Garcia</td>
<td>Update section 5.3, appendix A and appendix B.</td>
</tr>
<tr>
<td>0.92</td>
<td>06/07/2010</td>
<td>Juan Lambea</td>
<td>Updated section 1, 2.3, 4.2.1 and 9</td>
</tr>
<tr>
<td>0.93</td>
<td>06/07/2010</td>
<td>Beatriz Fuentes</td>
<td>Updated sections 8.2,8.4</td>
</tr>
<tr>
<td>1.0</td>
<td>05/08/2010</td>
<td>Sergio Garcia</td>
<td>Updated document with internal reviewers recommendations</td>
</tr>
<tr>
<td>1.1</td>
<td>09/09/2010</td>
<td>Juan Lambea</td>
<td>Updated document with reviewers recommendations from PMT</td>
</tr>
</tbody>
</table>
Executive Summary

This deliverable is the summary of the work done in second year of the European Project SLA@SOI in the Business SLA Management work package. The objective of this work package is to enrich SLAs, from the business level down to the physical infrastructure, with the business terms and constraints needed in a real eContracting framework. The project envisages developing an automated e-contracting lifecycle that covers the whole interaction between service providers and customers. This spans from the commercial product definition (based on offers and prices) to the negotiation and sales process. The management of the post-sales relationship including SLA monitoring and customer behaviour analysis is also addressed.

It is necessary for service providers to define service level agreement templates (SLAT), i.e., template contracts including business terms and constraints. This deliverable presents the architecture used to provide this functionality, and the SLA model customization used to define and describe the Business Terms and Parameters. The Business Terms are based on the work done in the first year, in feedback provided by the use cases of the project and other external sources about contracting[9].

The document also explains product management questions related to business management, and how answers to these questions have been implemented. The business negotiation approach for selling products as well as details on how this can be customized by business people is explained. The business assessment flow used to modify the prices and parameters stored by the framework in the selling process is detailed, as is the translation of violations into business actions like penalties and service terminations.

Regarding the customers, this deliverable also describes how their relationships can be managed. This includes the different ways to communicate with customers so that they are able to consume products and services and to receive reports and information about the history of the consumption of their SLAs. It also explains the way they can provide their feedback through the framework, so that the business can study it and take decisions to enhance customer perception and satisfaction.
# Table of Contents

1 Introduction .............................................................................................................. 9

2 Contribution Overview .............................................................................................. 10
  2.1 Key Innovations.................................................................................................. 10
  2.2 Framework Contributions ................................................................................... 12
  2.3 Task-level activities ........................................................................................... 14

3 Architecture ............................................................................................................. 16
  3.1 The refined architecture....................................................................................... 16
  3.2 Business Manager .............................................................................................. 17
  3.3 Business SLA Manager ...................................................................................... 17
  3.4 Internal Components ......................................................................................... 18
  3.5 External Components ......................................................................................... 21

4 Business SLA Definition Framework ....................................................................... 24
  4.1 Business SLA Model Specification ..................................................................... 24
    4.1.1 The Generic SLA Model ............................................................................. 24
    4.1.2 Business Terms Model ............................................................................. 26
  4.2 Product Management Specification ..................................................................... 32
    4.2.1 Product Definition ...................................................................................... 32
    4.2.2 Product Management ................................................................................ 33
  4.3 Business Adjustment Specification .................................................................... 35
  4.4 Implementation Specification ............................................................................ 36

5 Other Business SLA Interfaces ................................................................................ 43
  5.1 Product Discovery Specification ......................................................................... 43
  5.2 Control Track Specification ............................................................................... 43
  5.3 Business Web Tool Specification ...................................................................... 44
  5.4 Implementation .................................................................................................. 45

6 Business SLA Negotiation ......................................................................................... 47
  6.1 Business SLA Negotiation Specification ............................................................ 47
    6.1.1 MonoProvider Negotiation ........................................................................ 48
    6.1.2 MultiProvider Negotiation ........................................................................ 48
    6.1.3 Summary of actors profiles ....................................................................... 49
  6.2 Customization of generic SLA negotiation protocol Specification ................... 51
  6.3 Implementation .................................................................................................. 55

7 Assessment Engine .................................................................................................. 60
  7.1 Assessment and BSLAT Customizing Specification ............................................ 60
    7.1.1 Offer values checking .................................................................................. 60
    7.1.2 Policies application .................................................................................... 61
    7.1.3 Promotion and Customization application ............................................... 61
    7.1.4 Validation of the offer ................................................................................ 61
  7.2 Customer Criteria Interface Specification ......................................................... 62
    7.2.1 Criteria definition ....................................................................................... 62
    7.3 Implementation ................................................................................................ 64

8 Post-sale Management: Violations, Penalties and Reporting .................................... 67
  8.1 Reporting and Representation of Business Violation Evidence Specification ...... 67
    8.1.1 Overview .................................................................................................... 67
    8.1.2 Business SLA reporting architecture ....................................................... 68
    8.1.3 Business level SLA reporting policies ..................................................... 71
    8.1.4 Business SLAM reporting policies ......................................................... 72
8.1.5 Structure of BSLAM reports ................................................................. 74
8.2 Violations and Penalties management Specification ............................... 76
8.3 Customer Relations Specification .......................................................... 77
8.3.1 Customer Relationships Management .............................................. 77
8.3.2 Push/Pull Reporting ........................................................................... 78
8.3.3 Customer rating ................................................................................ 78
8.4 Implementation ..................................................................................... 78

9 Conclusions .............................................................................................. 82
  9.1 Summary of the contributions .............................................................. 82
  9.2 Outlook to next steps .......................................................................... 83

10 References .............................................................................................. 84
Appendix A: Glossary .................................................................................. 85
Appendix B: Abbreviations .......................................................................... 88
Appendix C: Business Terms ................................................................. 89
Appendix D: API ......................................................................................... 93
# Table of Figures

- Figure 1: Business components and interactions ........................................ 17
- Figure 2: Component Diagram and the interactions among them .................. 19
- Figure 3: BSS and SLA@SOI framework relationship .................................. 22
- Figure 4: SLA Model architecture .............................................................. 24
- Figure 5: SLA(T) Model .............................................................................. 25
- Figure 6: Agreement Terms .......................................................................... 25
- Figure 7: Business Level Guaranteed States .................................................. 26
- Figure 8: Access Control terms ...................................................................... 27
- Figure 9: Business Level Guaranteed Actions .................................................. 28
- Figure 10: Pricing Business Terms ................................................................. 29
- Figure 11: Party Annotation ........................................................................... 31
- Figure 12: Product Diagram ........................................................................... 33
- Figure 13: States diagram of the product lifecycle ............................................ 34
- Figure 14: Annotations on Party inside the SLA Model ................................. 36
- Figure 15: Penalty as a Guarantee Action inside the SLA Model .................... 37
- Figure 16: Business component price inside the SLA Model ......................... 38
- Figure 17: Termination Clause inside the SLA Model ..................................... 39
- Figure 18: Rule Template example for a Promotion based on dates .............. 40
- Figure 19: Example of Parameters used in the Rule Template ....................... 41
- Figure 20: Business departments relationships .............................................. 45
- Figure 21: A generic negotiation process ....................................................... 52
- Figure 22: Bilateral protocol from customer (buyer) side ............................... 54
- Figure 23: Bilateral protocol from provider (seller) side ................................ 54
- Figure 24: Different tiers of the Protocol Engine ............................................. 58
- Figure 25: Customers and provider interact using the PE and SC .................. 58
- Figure 26: A general overview of the negotiation interaction ....................... 59
- Figure 27: Assessment process ..................................................................... 60
- Figure 28: Validation of the offer ................................................................. 61
- Figure 29: Detailed Assessment Process Implemented ................................... 65
- Figure 30: Business monitoring reporting architecture .................................. 68
- Figure 31: BSLAM Reporting Scheme Entity Relationship Model .................. 71
- Figure 32: Example of BSLAM Reporting Policy .......................................... 73
- Figure 33: Example of BSLAM Report .......................................................... 75
1 Introduction

This deliverable describes the architecture, design and implementation of the business SLA management framework which has been created within the second year of the European research project SLA@SOI. The document includes details of the Business Service Level Agreements (SLAs) model and implementation characteristics of the features supported by the development in the second project year.

The main motivation of the work package is to provide a set of business elements needed for a full eContracting suite for SOI.

This document starts with a review of the contribution of the work package to the overall framework. This includes the key innovations and the high level tasks completed. In the third chapter the architecture is explained from the general framework down to the particular components developed. Chapter 4 explains in detail the Business SLA Framework, Business SLA Model and terms introduced, product definition and management, business adjustment, and their implementations, for this phase of the project. Chapter 5 describes the different interfaces and their implementation. Chapter 6 explains and describes the business SLA negotiation details and its implementation in the second year, from the general business requirements to the specific negotiation protocol customization. In chapter 7 the assessment process and its implementation is explained. Finally, Chapter 8 describes the post-sale management studies including customer relationships and reporting facilities and evidence of business violation.

This document serves as formal deliverable D.2a for reporting the work progress of work package A2 at project month 26.
2 Contribution Overview

2.1 Key Innovations

Work package A2 focuses on the business interactions between the service providers and the customers, in all the aspects in which SLAs take part. During the second year of the project, there have been innovative achievements in several aspects: the commercial offering management, the business modeling of SLAs, the specification and customization of SLAs, the customization of the business negotiation of SLAs, and the representation of business violations and their reporting. Some of these achievements represent scientific contributions, while the others are innovative solutions with practical application.

Comprehensive Business Management suit for e-contracting

The basic contribution of this work package is the definition of a business model on top of the SLA model defined in the project. This model allows SLA@SOI to define products and services, and all the business oriented information that is commonly included in SLA best practices. This model enables the global framework to provide a real business environment for trading and e-contracting of e-products and e-services between customers and providers. It also provides the possibility to define business rules, negotiation templates, etc. to customize the business strategy of the services providers (discounts and promotions, different SLA offers, etc.).

The definition of such a model, together with the SLA model and framework, enables the leveraging of automated analysis, verification and negotiation of SLAs to be provided at the business layer: supporting the negotiation of prices, automatic charging of economic fees for the breaching of SLAs, etc.

The first version of this business SLA model has been already published as scientific work [7] and further dissemination will be carried out with the updated versions. Business SLA model is based on some classes from SID model from TMForums [12] and other concepts from A5 work package [3] from SLA@SOI.

Another one of the innovative solutions provided by A2 during the second year is the creation of a tool for the back office management of the commercial offer. This tool supports the definition and characterization of new products and the services on which they rely. The SLAs of the atomic services that compose a customer-facing product are merged within the tool’s holistic SLA framework, based on business rules that can be defined and managed by an administrator. Moreover, the business SLAs can be graphically defined by the user, in order to fit the commercial requirements of the product and the marketplace. This characterization includes:

- The management of the catalogue of atomic services with their business SLAs.
- The management of the commercial products catalogue and their SLATs.
- The definition of the commercial offer: prices, discounts, promotions.
• The specification of policies that must be taken into account when a customer modifies some guarantee terms.

• The generation of the business reports with data and graphs (SLAs consumption, violations, etc.).

• The customer feedback collecting and sharing with new customers

The Business SLA tool is completely integrated with the SLA@SOI architecture and information model, and interacts with the two main business components, as it will be shown in this deliverable: the Business Manager and the Business SLA Manager.

Customization of Business SLA definitions taking into account the requirement of customer and service provider

One of the innovative aspects that this work package pursues is the customization of the SLA’s definitions taking into account the requirements and preferences of both the customers and the providers. In this sense, SLA@SOI aims at developing mechanisms for calculating the best business SLAs for both parties during the negotiation and establishment phases. The information required to carry out this business SLA assessment also includes gathering data about the customer and the profiles from the Business Support Systems (BSS).

The preferences and profiles of the customer and the provider are currently defined as promotions modelled with business rules associated with the offered products of the marketplace. This approach enables SLA@SOI to define special conditions depending on factors such as the socioeconomic situation of the customer, the country of origin, or any other criteria. These rules can define additional discounts on the price, special SLATs options, etc.

Dynamic negotiation of agreements

One of the topics addressed in SLA@SOI is the customization of the business negotiation. Negotiation is the activity leading to the stipulation of an agreement among the participants in the interactions. In particular, a customer and a provider would reach an agreement about a particular good or service. Negotiation interaction follows the steps defined in the negotiation protocols - these specify the roles and the activities each party can perform. It would be interesting to have a means to customize the negotiation protocol depending on the parties involved during the negotiation interaction. We should be able to customize some of the parameters of the negotiation protocol according to the customer profile. In particular we could foresee that depending on the customer profile some decision could be taken. We could imagine that a well known customer could have a different interaction than an unknown customer; or even, that this interaction could be guided by the rating of the customer. Similarly, we could imagine that the behaviour of the customer could differ depending on the provider. These kind of things are studied and detailed in this document.

Business SLA post-sale management

With respect to representation and reporting of business SLA violations, the main innovation of SLA@SOI is the adoption of an architecture that separates SLA monitoring from SLA reporting and evidence representation. The main reason underpinning the need for this separation is that SLA guarantee terms might need
to be monitored for different purposes that have different reporting requirements. To address the diversity of reporting requirements, SLA@SOI has developed a layered architecture where a business SLA reporting module exists on top of the monitoring infrastructure of the SLA@SOI framework, taking care of the reporting requirements. The primary function of this component is to receive low level monitoring information from the SLA@SOI monitoring infrastructure and to transform it into business SLA monitoring reports (BSLAM reports) by taking into account the requirements of the business layer components of the SLA@SOI platform. The operation of this module is driven by business reporting policies that determine the monitored SLA terms that should be included in a BSLAM report, the types of monitoring results about these terms that should be included in the report, and the required frequency of the generation of the reports. The specification of business reporting policies is enabled by the introduction of an XML schema for specifying reporting policies and an XML schema defining the general representation structure of business level reports.

2.2 Framework Contributions

This section describes the various features that have been implemented by this work package and contributed to the SLA@SOI Framework.

Customer registration

This feature allows a customer to be registered in the framework and to be authenticated to contract products. A customer can be an individual person or an organization. When the customers provide the information, it includes the user and password they will use with the framework.

Product definition

With this feature, the administrator (or business manager) can create a product in the framework. A product is seen as a service that can be sold to a customer and is characterized by some attributes that are defined in the Business SLA Template (BSLAT). One or more templates can be attached to a product, each of them with different prices and conditions inside the guarantee terms offered. The administrator can manage other attributes like promotions and policies. The promotions and policies are discounts or recharges associated to the final price of the product.

Product management

The Administrator and other user roles can modify the status of a product in its lifecycle. The lifecycle of a product denotes the current status of it, and the status determines the departments that can act on the product at each stage.

The different statuses are: Defined, Promoted, Pending, Rejected, Ready, Expired.

Product discovery

Another product feature is the possibility for end customers and providers to query the system to recover the available products that fulfil their requirements.
in terms of searching. This search is based on all the definition and attributes of the product and its categorization.

**BSLAT customization**

It is possible to do a customization of the prices included in a BSLAT using the end customer profile and the promotions. This feature is a type of promotion oriented to specific customer attributes that can be taken into account.

**Business Adjustment**

Although different adjustment components exist in the different SLAMs of the framework, adjustment functionality must be available on the business level to take into account information not accessible from the lower layers. Some decisions may depend on business criteria, for instance: the service provider could decide if some users must be prioritized when solving SLA violations. In the case of the malfunctioning of the service, the Business Adjustment component is in charge of calculating and applying the rewards or penalties incurred by any of the parties.

Furthermore, the different enforcement actions should be properly coordinated. Being at the highest level possible, the Business Manager is the best place to effectively implement this coordination.

**Business Negotiation**

This feature provides a way to sell the products to the customers. The way of selling the products reuses the negotiation protocol defined in the A5 Work Package. But this negotiation can be customized from the business layer by using other features (e.g. the Control feature described below). In the business negotiation it will take into account other parameters (see the Product Definition feature) and customer profile personalization (see BSLAT Customization feature).

**SLA Management**

This feature makes available all the administrative information related to the lifecycle of a given SLA. In particular, the detailed information of the content of the Agreement, any extension and revision to it, and the history of its performance should be persistently stored and accessible to both the service provider and the customer.

**Push/Pull Business SLA Reporting**

This feature allows the service provider to generate reports about the administrative information of an SLA, including the description of the purchased service, validity dates and the conditions of the agreement, as well as information about the performance of the service, in terms of violations and penalties occurred. The generated report will be periodically sent to the customer, and also an interface will be offered so the customer can request the business information on demand.

Another part of this feature is related with the post sale of the framework products and the possibility to the customers to rate. Customers can rate the
services and the products and this information will be shared with the rest of the customers.

**Control and tracking the business across the layers**

The Control feature allows the framework to be driven by a Business Manager in terms of definition of policies (based on rules) that can be applied to the different layers (business, software and infrastructure) for negotiation and adjustment.

The Tracking feature allows each level to share the SLA violations with the Business Manager, sending them or not (depending on the defined control policy).

These features make sense in a framework provider that has to control all the layers for the delivery of products to the customers, such as in a big enterprise where they have their own three layers.

### 2.3 Task-level activities

**Business SLA Models**

During Y2, the work done by this task has been based on the definition of the business model that has to support the features and the scientific advances made in the other activities and tasks in the A2 work package. The outcome is the business model to support the development and the Business Terms integration in the generic SLA model of the framework.

The important work done in the Business Terms model has been done taking into account the use cases requirements as well as other sources of SLAs and contracts [8].

**Business SLA commercial creation**

In this task we have defined the flow and the way in which different Departments can interact with the platform at the business level. We have studied the different roles of each department and the elements affected by the decisions that are made. The different actions to be taken by the different departments can affect anything from the loading of data to the simple approval of it, including the lifecycle and management of other additional elements. In project year 1, we have developed a first release of a web tool to manage/show SLAs, but we used it as proof of concept that has been evolved in year 2.

**Business SLA negotiation**

In this task we have studied the different requirements (from the business customer to the provider) that can be addressed in a business negotiation. Importantly, the negotiation protocol has also been studied to see how it can be customized using other framework features, and how it is affected by the business assessment process (policies). In project year 1 we have defined/implemented some interfaces that have been redefined/recoded in project year 2.
**Business SLA assessment**

The assessment task has been based on the definition of the different tools that allow manipulation of the prices using different approaches (policies, promotions, customer profile promotions, etc). This activity entails the modelling of different kind of elements that can modify the prices using an engine that processes rules. In project year 1 we have defined policies to drive the Business SLA assessment, and we have reused this approach as a little part of the assessment. In year 2 the business SLA assessment is composed by several components (promotions, policies, etc).

**Business SLA post-sale management**

In this task we have defined how we can retrieve information about the SLAs and the monitoring information stored across the platform. The Business component is the collector of the information (in order to deliver it to the business customer and providers). Other relevant information collected includes historical violations and penalties generation.

Also, additional ways to communicate with the customers have been explored, in order to retrieve their feedback. Customers can rate the services (SLA) included in a product and this information can be used by other customers to assess the services they want to contract. Business violation evidence and reporting have also been analyzed and modelled.

Basic reporting was implemented in project year 1, however the new complexity of the SLA@SOI architecture has obliged to redefine fully the collecting and reporting process in project year 2.
3 Architecture

In this section, the components implemented as part of the A2 work package are introduced. First, a high-level description of those components placed within the overall framework architecture is presented, together with the interactions that they expose. Then, a subsection is devoted to detail the internal subcomponents and their relationships. Finally, some business elements not included in the framework are described in the External Components subsection.

3.1 The refined architecture

The SLA@SOI project has designed an architectural framework for SLA Management across the IT stacks. The proposed architecture is defined in [2], and it encapsulates the business activities into two main components: Business Manager and Business SLA Manager. The diagram in Figure 1 has been extracted from the general framework diagram, and focuses on the high-level components implementing the business logic. These components expose interactions for the relationship with customers and 3rd parties, as well as interactions for the communication with other components of the framework:

- **Business Manager** implements <<query_product_catalog>> interaction that allows the final customer to search products and services. It also implements <<customer_relations>> interaction that permits the customer to interact with the framework and lets them register and share information with the business platform. <<provider_relations>> interaction is an additional interface in the Business Manager that allows the providers to interact with the framework (this interface has been defined in collaboration with Task 3.6 of work package A3). There is another additional interaction called <<control/track/query>>, which connects the business layer with the SLA Managers.

- **Business SLA Manager** implements the <<negotiate/coordinate>> interaction with the end customer. This interaction enables the customer to contract a product and to order the provision of SLA resources (in some cases). Business SLA Manager, Software and Infrastructure SLA Managers implement <<control/track/query>> interactions. This interaction allows the Business Manager to communicate with the SLA Managers of the different layers. The aim of the communication is to query the SLA and SLAT registries, to set and retrieve policies and to receive SLA violations.
One important question that is necessary to remark here with reference to current SLA@SOI architecture is the way to communicate the framework with other providers. The interactions between the framework and providers are done mainly using business components; Business Manager and Business SLA Manager.

### 3.2 Business Manager

Business Manager (BM) is the module responsible for the overall contracting and sales process. The need for this component comes from the fact that business information is needed to take proper business decisions. Part of this information is private (in some countries there are specific laws for the protection of data) and therefore must not be shared among components.

The responsibilities of a Business Manager are:

- Take overall business decisions that affect all levels based on business information
- Decisions have to be taken in one single point based on all available the information. There is no other component in the architecture that would be able to link with all the layers and to act as collection point.

In order to do this, business information is needed to indicate how decisions will be taken, such as:

- Global prices application, rewards, promotions and discounts
- Service Provider selection based on price and customer requirements
- Business addressed rules based on profits and costs

### 3.3 Business SLA Manager

Business SLA Manager is the component in charge of managing the negotiation process in order to obtain the different agreements. This component contains the registries where SLAs and SLATs are stored and it extends the Generic SLA Manager. The responsibilities of a Business SLA Manager are:

- negotiate/operate SLAs with customers & 3rd parties
- oversee the complete set of SLAs in its domain
- provide domain-wide SLA planning & optimization

---

**Figure 1: Business components and interactions**

![Diagram of business components and interactions]
Depending on the specific context/requirements of the use case a separate Business SLA Manager may be set up, for example for:

- a complete organization
- per department
- each individual service

This component is connected with the BM using the POC (Planning and Optimization Component of the Generic SLA Manager, [3]) and PAC (Provisioning and Adjustment Component of the Generic SLA Manager, [3]) components. The real functionality behind those components is inside the BM.

### 3.4 Internal Components

Figure 2 shows the internal components of the Business Manager and the Business SLA Manager and the relations among them described in the general architecture.

The general architecture diagram specifies some internal modules of the Business Manager. Those represent a high-level encapsulation of the functionality, and they have been refined into some more specialized components.

The internal components are the following:

<table>
<thead>
<tr>
<th>High Level Component Group</th>
<th>Internal Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Relations</td>
<td>Customer Management</td>
</tr>
<tr>
<td>Provider Relations</td>
<td>Provider Management</td>
</tr>
<tr>
<td>Product Management</td>
<td>Product Management</td>
</tr>
<tr>
<td></td>
<td>Product Creation</td>
</tr>
<tr>
<td></td>
<td>Product Discovery</td>
</tr>
<tr>
<td></td>
<td>SLAT Composition</td>
</tr>
<tr>
<td>Business Intelligence</td>
<td>Business Engine</td>
</tr>
<tr>
<td></td>
<td>Assessment Engine</td>
</tr>
<tr>
<td></td>
<td>SLA Evaluator</td>
</tr>
<tr>
<td></td>
<td>Billing Engine</td>
</tr>
<tr>
<td></td>
<td>Settlement</td>
</tr>
<tr>
<td></td>
<td>Post-Sales Management</td>
</tr>
<tr>
<td></td>
<td>Violation Management</td>
</tr>
<tr>
<td></td>
<td>Penalties Engine</td>
</tr>
</tbody>
</table>

Control & Tracking Manager is a transversal component that is used by the Business Engine (to set negotiation and adjustment policies in the SLAMs) and Violation Management (to receive violations from the different layers).
Figure 2: Component Diagram and the interactions among them

The following paragraph explains the basic interactions and modules depicted in the above figure:

1. \text{<<ICustomerRelations>>} interaction. This interaction covers the registration and reporting operations.
   
   - The registration operation allows the customers to be registered into the framework and modify their information in order to able to interact with it. This operation is managed by the Customer Management module through the \text{<<register>>} internal interaction. The customer information will be stored inside the data store of the Business Manager that contains all the private data of customers.
   
   - The reporting operations allow the customers to request their reporting information from the Post-sale module that has access to all the reporting data and implied registries. In the same way, the Post-sale module could provide the customer periodic reports about the behaviour of their SLAs. These operations are managed by the \text{<<reporting>>} internal interaction.
2. Post-Sale Management interactions. This module is in charge of managing all the data related to the relationship between the framework and the customer.

- It receives and requires information from the Violation Management module and Billing Engine module through the <<violationComunication>> and <<billingComunication>> internal interactions.
- Through the <<IQueriable/modifiable>> interaction, the postSale component is able to access to the registries that contain all the SLA and SLAT inside the Business SLA Manager in order to know about the signed conditions.
- Sometimes, Reporting information can be stored over different SLA Managers. A way is needed to the Business Manager to be able to access this monitoring information. The post-sale management module will implement the <<IRepor>> interaction to do it.
- Control and Tracking Manager Relations. This module will receive and send all the messages to the interface <<IBusinessControlAndTrack>>. This will decide the corresponding Module to process the message (Violation Management <<tracking>> or Business Engine <<control>>).

3. Violation Management. This module is in charge of processing the violation events received through <<tracking>> internal interaction. It contains a PenaltiesEngine to evaluate the economical consequences of the violation. Then, the Violation Management can contact to either the Billing Engine through <<penaltyBillingEvaluation>> internal interaction in order to charge the corresponding penalty, the Provider Management through <<violationCommunication>> internal interaction, the Post-Sale module through <<violationComunication>> internal interaction to communicate the violation or termination of the SLA to the corresponding parts, or the Protocol Engine through <<trigger-(re)negotiation>> interaction to launch a renegotiation.

4. Billing Engine. This module is in charge of the Billing management. It receives messages of Violation Management module through <<penaltyBillingEvaluation >> interaction in order to apply the corresponding charges. This module could contact the Provider Management through <<billingCommunication>> interaction in order to describe the corresponding part of the charging to each service provider. This module is contacted by the post-sale module through <<billingCommunication>> interaction in order to extract the corresponding billing information for each customer.

5. IQueriable interface. This interface gives access to the products and templates catalogue, and so it is needed to give access to different modules such as Product Management, Post-Sale management or Violation Management module. The Product Discovery module will access the Business SLA Manager data stores in order to obtain information on the requested products and templates. This module could ask the module Assessment Engine through <<assessmentSLAT>> internal interaction for a customization of the templates for a specific customer.

6. Product Creation Module. This module is in charge of creating and modifying products. It will be accessed by the <<Product creation>> interface. When needed, this module will access the module SLAT
Composition through \texttt{composeSLATs} to merge the different templates included in a unique composite template. This module will need access to the product and template catalogue, so it will use the \texttt{IQueriable/Modificable} interaction to obtain the product and SLATs required and to save the new products and Templates. This kind of operation also depends on the software composition engine.

7. \texttt{IProductSales} interface. This interface is in charge of managing the purchasing process between the end customer and the framework. It will access the Protocol Engine inside the Business SLA Manager in charge of guiding all the negotiation process. This module will contact the Business Engine module in order to check the credentials of the customer through \texttt{assess/customize/Auth} interaction (customer’s authentication). Then if the customer is authorized to operate, this module can access the Business Engine module through the \texttt{assess/customize/Auth} internal interaction to evaluate the offer of the customer or to others SLA Managers by the \texttt{negotiation/report} interface of the different SLA Managers with the aim of negotiating the final SLA. Finally, if an agreement is achieved, the Business Manager will configure the Monitoring Manager through \texttt{configure_monitoring} interaction.

3.5 \textbf{External Components}

Inside the Business Engine module there is another module called Assessment Engine and part of this module is SLA customization. The Assessment Engine customizes the SLA Templates calculating the best option for both, a specific customer and the service provider.

For the customization of the SLA templates this component takes into account:

- The SLA templates retrieved from the template registry.
- Customer requirements and limitations set as part of the negotiation process.
- The negotiation template including mark-up, ranges, and benefits ratios defined by the Business department. This information defines the limits allowed in the negotiation process including the different business rationales for each kind of user group.
- Customer offer customizing data based on the customer specific profile. This data will be retrieved from the integration with the Business Support System (BSS). This will be a private module developed by Telefónica.

\textit{Business Support System (BSS)}

As explained above, BSS is an information system dealing with customers and service providers. This system will take care of the confidential use of customer information. The correlation of this information will allow the system to generate the different profiles of customers.

\textit{BSS and SLA@SOI framework Relationship}

This section describes the components of the SLA@SOI framework that have a relationship with BSS. Then the different interfaces needed to communicate with the BSS will be explained. They are not detailed because they are company-dependant.
In order to customize the SLA Templates, the Assessment module will take into account the profile data of the customer obtained from Business Support System (BSS) that will require the templates of a specific product.

Figure 3 shows the relationships among the different components of the architecture. Only those components involved in the assessment process are depicted.

**Figure 3: BSS and SLA@SOI framework relationship**

The above diagram shows how a customer requests the SLATs corresponding to a specific product/service. The flow indicated by the arrows in the figure is now explained in some detail:

- The customer asks for the SLATs of a service/product about which they are interested. The customer has to contact with the Business Manager. Inside the Business Manager, the component in charge of managing the access to the products and their associated templates is the Product Discovery sub module inside the Product Management module. So, it is the one which receives the request.

- This module will query the SLAT registry inside the Business SLA Manager in order to obtain the corresponding SLATs.

- From this point, Product Discovery module will request Assessment module inside Business Engine to customize the SLATs according to the characteristics of the customer who is requesting for. Regarding the request to the Assessment module, the customer ID and the SLA Templates to be customized are sent to it.

- Once the request arrives to the Assessment module, it retrieves the customer information from the customer database in order to know if it is possible to customize the SLA Templates.
• If it is possible, the Assessment module asks the Business SLAT Customization sub module for the customization of the SLA Templates for a given customer.

• In order to customize the SLA Templates, Business SLAT customization module needs to know the customer profile. So, this component asks for it from the Business Support System (BSS).

• With all the data, the Business SLAT customization component will execute all the processes needed for the customization of the SLA Templates. At the end, the customized SLA Templates are returned to the Product Discovery module.

• Product Discovery module will return to the customer all the SLA Templates requested, the customized as well as the non-customized ones.
4 Business SLA Definition Framework

4.1 Business SLA Model Specification

As part of the Work Package A5 of SLA@SOI a generic SLA Model has been defined. This model offers several extension mechanisms that can be exploited to model domain specific SLA information.

The objective of this section is to define how the Business SLA specific information is modelled with the Generic SLA information model.

4.1.1 The Generic SLA Model

This section briefly describes the Generic SLA Model, paying attention to the elements that are later used to extend the Business layer. For deep details of the SLA Model, refer to the DA1a [2] and DA5a [3] deliverables.

Figure 4 shows the general information hierarchy of the SLA Model defined in SLA@SOI. It is a generic information framework for modelling SLAs and SLATs and serializing them to be used by the SLA Managers.

At the lowest level of the hierarchy, there are the Primitive (Ground) Terms on which everything else is built. Building on these Primitives, there are a handful of Ground Expressions, supporting annotations, and the generic expression of constraints, events, and functions. These expressions support extension/customization through standard vocabularies, in which specific terms, and semantics can be defined. Above this is a generic Service Description model, providing a means to describe the functional properties (interfaces, operations, and endpoints) of services.
The main body of the present specification defines the SLA(T) Model (shown in Figure 5), which builds on the Service Description level to allow for the expression of non-functional service properties and quality of service (QoS) guarantees. The SLA(T) Model provides the basic structure of an SLA(T), but leaves the specification of particular QoS terms open, supporting extension through standard vocabularies.

Basically, an SLA is a set of agreements between two (or more) parties. These agreements (see Figure 6) are expressed with terms, each of which denotes guarantees made by, or obligations on, the various parties. Each agreement term comprises an optional constraint expression specifying the conditions under which the agreement term holds (i.e. a precondition on the term). If no preconditions are specified then it is assumed that the term holds for the entire effective duration of the SLA. Guarantees defined in the agreement are either guaranteed states or guaranteed actions.
A guaranteed state is a guarantee made by one of the parties to the agreement, that a certain state of affairs will hold, e.g. Service Level Objectives (SLOs) or targets for Key Performance Indicators (KPIs). The state of affairs is defined by a constraint expression. A guaranteed action is an action that one of the parties to the SLA is obligated to perform (or may perform, or is forbidden from performing) under certain, specified circumstances.

At the highest level is a Business SLA(T) Model, which builds on the SLA(T) Model and Common Metrics to model business specific information. The main two mechanisms that have been used to extend the generic model are the extension of classes of the model, annotations and the definition of new Standard Terms.

### 4.1.2 Business Terms integration

In order to represent all the information related to the SLA business layer, this section describes how the different terms have been integrated in the model. The parameters described in this section are those identified in the Year 1 deliverable, together with the analysis of [8] and the requirements of B-Line use cases [10].

#### Guarantee Terms

##### a) Guaranteed States

![Business Level Guaranteed States Diagram]

The following Guaranteed States have been identified in the business layer:
- **Consumer Commitment.** It represents the commitments that the customer agrees in the contract in terms of service usage (for instance, maximum usage, peak times, or capacity usage).

- **Compliance.** Specification of the standards and recommendations that will be supported during the delivery of the services, including names, versions and dates.

- **Support.** Specification of how the support of the service will be provided. On one hand, it includes the availability of the contracted support. On the other hand, it may include a set of support procedures for different issues, including when the first feedback has to be offered, the feedback frequency, and the expected resolution time depending on the severity.

- **Archiving.** This term specifies the agreements in relation to the archiving of data of the service (service and historic information). It specifies the maximum capacity or the lifespan of the data, and the restrictions in the location of the data.

- **ArbitrationAndMediation.** Specification of the places where the legal disputes and judgments by arbitration should be arranged.

- **Reliability.** It represents the maximum time of service unavailability and the maximum time between two consecutive outages, without causing a penalty.

---

**Figure 8: Access Control terms**

Finally, Figure 8 describes how the **AccessControl** guaranteed state models the different roles of that the parties might have to access the various operations of the interfaces described in the SLA(T).
b) Guaranteed Actions

- **UpdateProcess.** When offering services, it is usual to bring down the service to update or manage it for short periods of time. This parameter specifies if this process is periodic and its frequency, how to distribute software and install software, if required, or the scheduled downtimes.

- **BackupRecoveryMechanism.** It includes the mechanisms required to backup the essential information of the service and the policy to do so, and the agreement on the time to recover the service from a backup in case of service outage.

- **Reporting.** This agreement term refers to the commitment to send reports about the service usage and SLAs status over time. It includes information about the report itself, the format and frequency and how to deliver it to the customer.

Figure 9: Business Level Guaranteed Actions
• **Monitoring.** This agreement term specifies which SLA parameters have to be monitored on a continuous basis in order to inform the parties. It includes information about the parameters themselves, their units and the frequency of check.

![Diagram](image)

**Figure 10: Pricing Business Terms**

• **Termination and TerminationClause.** These terms refer to the conditions under which any of the parties may terminate the SLA. Every cause may include a notification method and a fee to be paid in case of cancellation.

• **Penalty.** A penalty term is an amount of money to be paid in case of breach of another SLA term. These penalties can also be used to trigger SLA termination causes.

• **ProductOfferingPrice.** This term specifies the required commitment of a user to pay an amount of money for the product being provided. It includes the billing frequency and the time during which the offer is valid. The product can be divided into several components, each of them with a price type, price and amount of components that aggregate the service. These components can specify price modifications (e.g., discounts).

• **Billing.** This term details the structure, content and frequency of the bill that is sent to the customer. It includes information about the payment or the delivery information. It is related to the products, the penalties and the termination fees.
**Standard Terms**

The SLA@SOI standard terms are the basic vocabulary to be used by the expressions in the classes described above. As for business terms, the following Standard Terms have been defined:

**Standard Terms for Price Types & Billing Frequencies:**

- **bus:one_time_charge (price-type)** This represents a unique charge to a customer for the consumption of a product. In this case, this component price is charged when the product is contracted.

- **bus:tariff_usage (price-type)** This represents the way of charging to a customer for the consumption of a product. In this case, this component price is charged each time the product is consumed following a tariff. A tariff can be defined with different prices based on the usage interval.

- **bus:flat_rate(price-type)** This represents what is charged to a customer for the consumption of a product. In this case, this component price is charged once for a period of time, no matter what the use of services will be.

- **bus:per_request (billing-freq)** The bill is charged once the service is consumed.

- **bus:per_day (billing-freq)** The bill for the use of services will be charged once a day.

- **bus:per_week (price-type)** The bill for the use of services will be charged once a week.

- **bus:per_month (price-type & billing-freq)** The bill for the use of services will be charged once a month.

- **bus:per_year (billing-freq)** The bill for the use of services will be charged once a year.

**Standard Terms for Price Modification Types:**

- **bus:discount** During the negotiation process, the economic conditions of a SLA template could change. These changes could be reflected on a discount in the price of the final agreement. If the product to purchase is modified in some way (promotions or customizations) the final result of the price will be decreased.

- **bus:increment** If the product to purchase is modified in some way (price policies or customizations) the final result of the price will be increased.

**Standard Terms for type inside of price modification type:**

- **bus:amount** The discount or surcharge over a component price is a quantity of money.

- **bus:percentage** The discount or surcharge over a component price is a percentage over the original quantity

**Standard Terms for Business metrics:**
bus:penalty_count
expression: penalty_count( s : SERVICE): QUANTITY
definition: The number of penalizations applied to a service due to the reception of a violation of the signed conditions of a service.

bus:violation_count
expression: violation_count( t : AGREEMENT_TERM): QUANTITY
definition: The number of notifications received of violations to an agreement-term, guaranteed state or guaranteed action of a service.

Non-modelled parameters

This section describes a set of parameters that have not been directly modelled as business terms, either because they have been already defined in other parts of the model, or because it has not been considered useful for SLA@SOI

Parameters described by SLA Model:

- **ServiceDescription.** The description of the service itself and its interfaces and parameters are now covered by the Abstract SLA Model.
- **ContactPoint.** In order to represent the important Business Contact Point information (Name, E-mail, phone number, etc.), the Party class instances are annotated, as shown in Figure 11.

![Figure 11: Party Annotation](image)

Parameters not covered by the model, but that will be analyzed again for their inclusion in Year 3, include:

- **Exclusions.** The modelling of the parameters that have to be considered explicitly as not agreed has not been included in the year 2 model.
- **Renegotiation and ChangeProcedures.** This term includes the specification of parameters that are allowed to be negotiated during the runtime.
- **SensitiveInformationAccess.** As long as the specification of the information used by the service is not described, the access control to it is not modelled.
- **Warranty.** Not applicable for services.
4.2 Product Management Specification

A Product needs to go through different stages during its lifecycle (such as creation, negotiation, provisioning, retirement, etc.). The functionality of the role of business manager is managing the entire task related to a product and its transitions between these stages indicating to the appropriate modules when they have to perform specific operations. Inside this functionality there is another specific subject related to the product management that is in charge of product creation.

4.2.1 Product Definition

A Product is the Service Offering that a Service Provider presents to its end Customers. A Product may be different in terms of his nature:
- an atomic service or
- a combination of several services (either from the same or several Service Providers) or
- an aggregation of them.
In the former case it is necessary to have tools that allow characterizing the atomic services while in the latter case a creation process to arrive at a final holistic business SLA is needed.

As a result of the complete process the Service Offer (Product) is created and the SLA Template(s) provisioned into the appropriate modules within the framework (together with any Business Rules that must be followed during the negotiation process for that particular Service Offer).

Product Manager defines and creates new Products obtaining as a result a Business SLA Template for the Product (can be based on an existing template or not). The Products may be based on atomic services, in which case it is just necessary to define the characteristics of the SLA template associated to it. It also may be possible to create Products based on a composition of atomic services (coming from one or several Service Providers) in which case it is necessary to provide information about the atomic Services as well as any other parameters needed to arrive at the final Product.

This operation entails the browsing of the different SLAT registries in order to find services to be built as products.

A product has a unique business SLA template associated and other elements that can be used to characterize it and to modify its price. The product could have different offers and each offer has different component prices. Each offer can be focused to a specific Geographical Area. Each component price can have a discount or a recharge.

A product has two main information blocks associated with it:
- Promotions: Provides a way to define a discount applicable with some constraints like dates. An example of a promotion could be: "give 5% of discount in subscription price in the launching week of a product (1/1/2011 and 8/1/2011)".
- Policies: Defined as the variation based on a percentage or an amount of a guarantee term used in a SLA. It is used to define the way how the
customer change can affect an element in the SLA Offer. An example could be: "Increase the subscription price in 2% if the original guarantee term X is decreased in 10%".

Figure 12: Product Diagram

4.2.2 Product Management

A Product will go through different stages during its lifecycle. Each state shows about what the interactions expected against it are. In the next paragraph, the different states are described. Then the relations among them will be depicted by a state diagram. The different states are:

- Defined: This is the first step. In this step, the information about the product such as product description or pricing information is defined.
- Promoted: After the definition of the product, it could be necessary to associate a promotion to the product. This action could be interesting
in order to create an attractive opportunity for the product to the customers.

- Pending: After the definition of the promotions associated to a product, it is necessary that these promotions are approved by the business department.

- Rejected: After the evaluation of the products and the associated promotions, the business department decides that the offer does not fit with the establish margins or strategy of the company. From this state new details of the product could be defined or redefined in order to be evaluated again.

- Ready: The business department decides that the product and its promotions are in accordance with the plans/strategy of the company. From this moment the product is available to be purchased.

- Expired: The conditions under which the promotion was defined are expired or changed. So the promotion is no longer available.

- Withdraw: At any time, the business department could decide that a product is no longer available for purchase any more. It could be because the current environmental circumstances dictate that the product/promotion no longer fits with the plans/strategy of the company.

Bellow the relationships among the different states are depicted:

![States diagram of the product lifecycle.](image)

**Figure 13: States diagram of the product lifecycle.**

The product lifecycle will guide in part the SLA status. That is, before a SLA related to a product could be negotiated, the product must be in ready state in order to be purchased. In the same way, in the case of a renegotiation of an
agreement, if the product is no longer available to be purchased, it is possible that the SLA may not be renegotiated.

4.3 Business Adjustment Specification

SLAs can be seen as containers for the functional and non-functional properties that both parties, customer and provider, agree specifying its obligations and rights during the product lifetime. However, this also represents a responsibility for the service provider, since it motivates the need of the implementation of an SLA enforcement process. Furthermore, the better the product fulfils its associated SLA, the better the user perception will be.

The quality of the final product delivered to the customer, depends on the quality of the atomic services it is composed of. The quality of each service must be guaranteed at all the levels of the IT stack, and for this reason SLA@SOI framework has included a specific Adjustment component in all the layers. At software and infrastructure levels, Provisioning and Adjustment Components (PAC) have been placed as part of the corresponding SLA Managers, and at the business level a specific component for Business Adjustment is deployed inside the Business Manager.

Different adjustment components receive different inputs in the different layers. Software and Infrastructure Adjustments react on available monitoring or prediction information of the corresponding layer (e.g. by forcing dynamic resources reallocation on the infrastructure, substituting a service or increasing the service level). Nevertheless, not all the exceptions can be solved locally, and therefore decisions may need to be escalated to the highest SLA management level, that is, the Business Manager.

Business Adjustment Component receives notifications of the violations produced at software and infrastructure layers, so it is aware of the behaviour of each of the SLAs. In case more detailed information is needed, it can make use of the <<control/track>> interaction to query the monitoring and adjustment registries of the underlying layers. Furthermore, Business Adjustment can access all the business data, mainly economical information, which is not accessible from the lower layers. Based on the collected information, Business Adjustment makes an analysis of the situation and decides plans and executes the necessary actions.

At the business layer, Adjustment takes care of the economical impact of the bad performance of the service, and triggers actions devoted to minimize the penalties paid to the customer, trying to maximizing the benefit and the revenue. Examples of decisions and actions that can be taken at business level are:

- Penalty calculation. An SLA specifies a penalty to be paid in case one of the Agreement Terms is breached. Business Adjustment must evaluate if a given SLA Violation has a penalty attached, and, if it has, then it is in charge of calculating the exact penalty and of applying it (through communication to the Billing module or for instance indicating the Lifecycle management module that the service has to be adjusted or interrupted).

- Analysis of profits and costs. Business Analysis will detect those services that are not profitable enough to be maintained. That is, when the difference between the price the customer is paying and the amount of penalties paid is lower than a given threshold.

- Guaranteed Actions. An SLA can specify an action to be applied in case a Guarantee Term is violated. Business Adjustment will take care of this kind of actions. A particular case is when the SLA specifies that, when a violation of a given Guarantee Term occurs, then the SLA must not try to
be recovered, but the agreement must come to an end. In this case, the Business Adjustment will trigger the termination of the agreement, including the commands to stop the services and the release of resources.

- Renegotiation. In some cases, Business Adjustment can determine that a modification in some of the conditions of the SLA can better be fulfilled with the current resources. Therefore, it will trigger renegotiation with the customer, to try to sign an SLA that will have fewer violations than the current one.

As explained above, the reaction may be specific to the particular SLA or to a particular profile, thus Business Adjustment provides hooks and rules to support individual SLA violations analysis and action decisions.

### 4.4 Implementation Specification

#### Business SLA Model Implementation

This section provides some examples of how the business terms have been implemented using the SLA Model.

As described in previous section, the contact point information is modelled as annotations of the Party SLA Model class, as shown in the code below:

```xml
<slasoi:Party>
  <slasoi:Text>
    <slasoi:Properties>
      <slasoi:Entry>
        <slasoi:Key>http://www.slaeso.org/slaesoomodel#gslm:org</slasoi:Key>
        <slasoi:Value>http://localhost:8080/services/GSLAEConverter/UGNegotiation</slasoi:Value>
      </slasoi:Entry>
      <slasoi:Entry>
        <slasoi:Key>http://www.slaeso.org/business/ContactPointName</slasoi:Key>
        <slasoi:Value>ServiceProvider Name</slasoi:Value>
      </slasoi:Entry>
      <slasoi:Entry>
        <slasoi:Key>http://www.slaeso.org/business/ContactPointPhone</slasoi:Key>
        <slasoi:Value>43460662107</slasoi:Value>
      </slasoi:Entry>
      <slasoi:Entry>
        <slasoi:Key>http://www.slaeso.org/business/ContactPointFax</slasoi:Key>
        <slasoi:Value>43460662107</slasoi:Value>
      </slasoi:Entry>
      <slasoi:Entry>
        <slasoi:Key>http://www.slaeso.org/business/ContactPointAddress</slasoi:Key>
        <slasoi:Value>Contact Point Address</slasoi:Value>
      </slasoi:Entry>
      <slasoi:Entry>
        <slasoi:Key>http://www.slaeso.org/business/ContactPointEmail</slasoi:Key>
        <slasoi:Value>email@email.com</slasoi:Value>
      </slasoi:Entry>
    </slasoi:Properties>
    <slasoi:ID>4</slasoi:ID>
    <slasoi:Role>http://www.slaeso.org/slaesosupplier</slasoi:Role>
  </slasoi:Text>
</slasoi:Party>
```

Figure 14: Annotations on Party inside the SLA Model

The following piece of code represents the XML serialization of a Guaranteed Action that represents the payment of a penalty (1€) in case of breaching of a Guaranteed State (ORCThroughputConstraint).
Figure 15: Penalty as a Guarantee Action inside the SLA Model

In the following example, the business terms that describe the payments related actions are described. Following the model shown in Figure 10, the XML below describes the price for a product, composed of several pricing components and information about the billing/charging.
Another important business term is the termination clauses specifications, that is, the specification of the termination of an SLA based on some specific conditions. In the example, the case of a termination triggered by the overcoming of a threshold number of violations of an SLA agreement term is shown.
Figure 17: Termination Clause inside the SLA Model

Product Management Implementation

The way in which can be done the product management is with the using of some APIs (more details about APIs at Appendix D: ). It has been defined an API to different objectives:

- manage the products
- change the status of a product (product lifecycle)
- create and register a Business SLAT
- promotion management
- penalty management
- business adjustment and negotiation policy management

A lot of functionality built in A2 has been done using Drools (for instance promotions and business adjustment). Drools is a set of products oriented to business rules engine that allows to build business logic based on rules (detailed information at [11]). This way to build the logic skips code changes in the applications, offering a flexible and fast way to modify the processes and criteria in software applications (it is more appropriate when you have a lot of different criteria and also they are complex). It has been defined different ways to manage
rules, modelled in the Business Management. It is possible to define and store rules and template rules.

- A promotion can be defined/implemented as the customization of a template rule with some parameters fulfilled or customized. For instance, we can have the promotion template rule, “promotion between dates”.

```java
package org.slaosi.businessManager.common.drools;

import org.slaosi.businessManager.common.util.SimpleDate;
import org.slaosi.businessManager.common.model.EmCustomersProducts;
import org.slaosi.businessManager.common.model.EmComponentPrice;
import org.slaosi.businessManager.common.model.EmPromotions;
import org.slaosi.businessManager.common.model.EmPriceVariationType;
import org.slaosi.businessManager.common.model.EmPriceVariation;
import java.math.BigDecimal;

template "Date Range Promotion Template"

rule "Rule @({RULE_ID} - @{RULE_NAME})"
when
  cp : EmComponentPrice()
  customer_product : EmCustomersProducts()
  eval (new SimpleDate(cp.getProductValidFrom()), isnBetween("@{DATE_BEGIN}","@{DATE_END}"))
  eval (cp.getPriceType().getUnitPriceType())==@{PRICE_TYPE})
then
  EmPriceVariationType pv = new EmPriceVariationType(new Long(@{VARIATION_TYPE}));
  EmPromotions promo = new EmPromotions(new Long(@{RULE_ID}));
  EmPriceVariation pv = new EmPriceVariation(promo, pv, cp, customer_product, "@{RULE_NAME}",
    new BigDecimal(), new BigDecimal(@{DISCOUNT}));
  customer_product.getPriceVariations().add(pv);
end
end template
```

**Figure 18: Rule Template example for a Promotion based on dates**

And when the promotion is associated to a product, it’s necessary to fulfil the parameters (for instance with the validity dates and the discount), as you can see in the following figure.
Business Adjustment Implementation

Business Adjustment extends the generic Provisioning and Adjustment Component (PAC), and takes advantage of the generic implementation for autonomic management provided as part of task A3.5 [1]. The same approach has been followed to implement the Adjustment components of the different levels [1][3][4].

They provide the implementation of a generic agent, able to execute one or more tasks or activities. Each task can rely on a rule engine mechanism to take decisions and trigger actions. Some common tasks are provided: EventBusHandlingTask, receiving events from a messaging bus; TranslationTask, translating or parsing events; AnalysisTask that relies on a reasoning engine to analyze the incoming events. A persistency mechanism is also available, based on hibernate and SQL.

In the autonomic architecture, configuration of agents and tasks are encapsulated in an xml configuration file. New tasks can be defined just extending the abstract ITask class, and assigned to an agent through the configuration file.

This architecture is very convenient for the implementation of the Business Adjustment, since it can be extended and customized without special difficulty. In this case, an agent is deployed inside the Business Manager, which is configured to execute an AnalysisTask:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Type</th>
<th>Rule Type</th>
<th>View Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE_BEGIN</td>
<td>DATE</td>
<td>WHEN</td>
<td>Init Date</td>
</tr>
<tr>
<td>DATE_END</td>
<td>DATE</td>
<td>WHEN</td>
<td>End Date</td>
</tr>
<tr>
<td>PRICE_TYPE</td>
<td>SELECT</td>
<td>WHEN</td>
<td>Price Type</td>
</tr>
<tr>
<td>VARIATION_TYPE</td>
<td>SELECT</td>
<td>THEN</td>
<td>Price Variation Type</td>
</tr>
<tr>
<td>DISCOUNT</td>
<td>NUMBER</td>
<td>THEN</td>
<td>Discount</td>
</tr>
</tbody>
</table>

Figure 19: Example of Parameters used in the Rule Template

Generic policies can be defined as promotions.
The Business Adjustment implements the `trackEvent` interface, through which it receives the SLA violations produced at software and infrastructure layers. These events are then in charge of the `BusinessAnalysisTask`, that feeds them into the rule engine which has previously been loaded with the appropriate definition of rules (following the specification in the configuration file).

Drools has been chosen as the rule engine, since it presents characteristics that fulfills the requirements of the adjustment, such as correlation of events, possibility to directly invoke Java code, rule packages, possibility of defining priority for each rule, complex event processing or temporal reasoning. Furthermore, rules can be changed on the fly, and it provides auditing functionality, which is an interested feature to keep a record of how decisions were made.

```xml
<AgentConfiguration>
  <agentId>Manager</agentId>
  <tasks>
    <TaskConfiguration>
      <taskId>Adjustment</taskId>
      <className>
        org.slasoi.businessManager.violationPenalty.BusinessAnalysisTask
      </className>
      <parameters>
        <entry>
          <string>RULES_FILE</string>
          <string>/rules/org/slasoi/businessManager/violationPenalty/business_adjustment.drl</string>
        </entry>
      </parameters>
    </TaskConfiguration>
  </tasks>
</AgentConfiguration>
```
5 Other Business SLA Interfaces

The aim of this chapter is to specify in a summary the functionality: product discovery interaction, control and track interaction, and business web tool. And also the implementation done in project year 2.

5.1 Product Discovery Specification

In today's environments, in which more business relationships with customers based on the Internet appear each day, the way in which products and services are exposed to the customers for consumption is very important, because it is the first channel of communication with them. The objective of product discovery is to provide an easy way to access Products/Services in order for customers to find the products that they wish to purchase.

The functionality needed by a Customer (or in some cases a Service Provider) is to access information about the available Products, and their templates, for purchasing. Basically this entails two different tasks:
1) Maintaining and managing the Service Registry for each Service Provider in which the details about the different Services/Products are stored and
2) Exposing this information to end Customers as a result of their queries.

There are two searching techniques used to find a product. One is a keyword-based search and the other uses category searching techniques to improve the online searching, in order to generate more relevant results. This type of search is based on a previous classification of the products according to some of their characteristics, sometimes made in a manual way. According to that, a service can have different categories and then products can be assigned to different categories. Category searching will allow the customer to find products according, as commented above, to some of their specific characteristics in which they were classified, such as the service type (travel agency or data storing) or the service provider that offers them. According to the explanation above, it will be developed which categories are interesting for the final customers who purchase the products. The objective of this kind of search is to allow customers to find and select products for purchase according to some specific characteristics or the customer's preferences. For instance, the categories associated with an email service could be 'internet' and 'messaging'. Once the customer finds the product they want, they can retrieve the different Business SLA templates available (these operations are included in the interaction <<query_product_catalog>>). In the next step, the customer will start the negotiation process with the framework as shown in Chapter 6 (negotiate interaction).

5.2 Control Track Specification

The interaction <<control/track/query>> is the link between the Business Manager and the different SLA Managers. This communication is bidirectional in the sense of some operations arrive as notifications or events to the BM component and other operations are raised to the SLAMs from the Business
Manager. The main aim of the control interaction is to be able to define and retrieve different kinds of policies from each SLAM. This can be useful in an environment in which a large enterprise has customers and multiple SLAM layers (software, infrastructure). Also in some cases it can be useful to have a way in a SLAM to change on the fly some of the rules and policies that drive the behaviour of some of the SLAM components.

These characteristics allow the SLAM to be able to change the negotiation, adjustment and provider control policies using the control interface.

Regarding the main aim of the track interface, it allows communicating important information to the Business Manager, including violations. The Business layer has to be aware of all the business relevant details of the consumption of the services in order for it to take actions to solve anomalous situations etc. Depending on the received event type, it will decide which internal component is in charge of managing it. For instance, if a violation is received, the component in charge of it is violation management, but if it is a matter of something related to business, the component will be the business engine.

More details about the API can be found at Appendix D:

### 5.3 Business Web Tool Specification

The Business Web Tool represents the user interface for the different departments in charge of the business management of products and services and supports the various tasks that must be carried out by them during the overall product lifecycle. Although the department's configuration depends very much on each organization, three major roles for these departments have been identified: the business department, the sales department and the customer relations department.

The business department defines the strategy to offer the products in the market. This department defines the products themselves, the product bundling, the aggregation of the services that comprises a product, and the management of the SLAs that must be enforced for them.

The sales department is in charge of driving the negotiation with the customers during the purchasing phase and the agreement of all the business terms of the required SLAs, as well as the definition and management of offers and promotions. The sales department is also in charge of managing and storing the customer profile information.

The customer department is in charge of the post-sales management of relationships with the customers. That includes reporting on the usage of the products and SLA breaches, detailing penalties and rewards. The historic information is then used by the sales department to define new promotions and offers.

The relationships and exchanges of information between all the departments are described in Figure 20.
To fulfil these needs, the business tool supports the functionality to publish new services and SLA templates, as well as the management of new and existing products and their lifecycle status. This functionality, which is basic for the business department, can be used by the sales department to look for the details of the commercial offers.

The sales department can define the promotions over product prices by using a predefined set of rules with a number of configuration parameters. For instance, it is possible to define and select a price discount on the basis of the country of the customer.

It also offers an interface to manage the business policies used to drive all the dynamic processes described. For instance, it is possible to add and configure special policies for setting discounts and charges under different circumstances.

In all cases, the business web tool offers an interface for defining and generating reports on the use of products, generation of revenues, and the consumption of SLAs.

Additionally, the tool includes a section to administer the basics of the business platform. This supports definition of rules and parameters, definition of policies, specification of business terms (pricing, billing, etc.), configuration of service and services types, configuration of local information and management of users and parties.

### 5.4 Implementation

#### Product Discovery Implementation

This functionality has been developed using web services. There are two operations to interact with the Product Catalogue to find a product. The first

![Figure 20: Business departments relationships.](image-url)
operation, called getProducts, is to recover the list of products that fulfil a criteria defined by the customer. The information used to select the products includes:

- Product information
- Offers information
- Categories

The information returned by the operation is a product list. This list allows the customer retrieves all the offers for the products they are searching for. Each product offer is associated with a different Business SLA template linked with the product. The second operation, called getTemplates, allows the customer to request the Business SLA Templates available for a product. The next customer step is to negotiate with the Business SLA Manager in a direct way using the negotiation interface.

**Control and Track Implementation**

The interaction <<control/track/query>> is defined using different interfaces. The control server is stored in the Generic SLAM, and offers its interface to the Business Manager, its client. Control is defined to allow reuse of the elements in the different SLAMs and also has a central point in which it is possible to manage the information. Adjustment and negotiation data are changed in the same way, indicating the type of sets of rules that require changing in the SLAM.

All of the different data has been modelled using rule modelling so that the design remains uniform.

Control interface offered by SLAMs has an API described at Appendix D:

Regarding the track interface, the server it is implemented in is the Business Manager, in order for the SLAMs to be able to communicate with it.

Track interface offered by Business Manager has an API described at Appendix D:

**Business Web Tool Implementation**

The Business Web Tools is a web portal for the management of the business aspects of the SLA-aware marketplace. It provides all the identified functionality by interacting with the Business Manager, the Business SLA Manager and their database.

The Business Web Tool is implemented with Java Server Pages (JSP) over the Spring framework. It can be deployed both over a Tomcat container or an OSGi platform and in both cases it makes use of the BM and BSLAM interfaces and the MySql database. The business rules are stored and executed on a JBoss Drools Engine.
6 Business SLA Negotiation

In this chapter is described the functionality about business negotiation like protocol customization, actor profiles, etc. Also business SLA negotiation is studied from the point of view of customer and providers.

6.1 Business SLA Negotiation Specification

One of the objectives of the SLA@SOI project is the development of a negotiation process that enables consumers and providers to agree on certain business SLAs. SLA Negotiation is the process leading to the stipulation of an SLA between a customer and a provider. Even if the steps in a negotiation process follow the rules of the corresponding protocol, there exist some configurable parameters making the process customisable. Configuration parameters could be derived exploiting the knowledge of the actors participating to the negotiation.

The negotiation always has to be driven by Business requirements.

Customers and providers are the actors of the negotiation process; from the provider side view, a negotiation could be mono or multi provider. A mono provider negotiation is the general negotiation case, in which all the resources belong to one single provider; the alternative case is that in which the resources belong to more than one provider. In such a case, before the stipulation of the final SLA between the customer and the main provider, the provider has to stipulate the SLAs with the third party providers. This situation will influence the protocol to follow during the negotiation. In fact, the principal provider could act as an agreement broker (deciding to accept/reject/customize SLA offers from the Customer without consulting the Third Party Providers), or as a gateway (forwarding the Customer’s offers to the Third Party Provider permitting them to accept/reject/customize the offers) or as a mix of them.

Other than the kind of provider, the negotiation protocol could be influenced by the profiles of the actors.

An SLA negotiation is defined by specifying the requirements defining the variables to negotiate, the structure of it, the process, the protocol and the strategy. While the process defines the phases of the negotiation interaction, the negotiation protocol defines the set of activities that each actor can perform.

In this deliverable a survey of the existing single negotiation protocols is provided.

Information deriving from the customer and provider profiles can be used to customize some of the protocol. The knowledge about a customer or a provider should impact on the settings of these parameters influencing the negotiation process, and a mapping between the negotiation protocol and the negotiation profiles can be defined.

All this information expresses some business requirements that will help to understand why the adaptation of the protocol is needed.

Two scenarios are presented: one general scenario where all the resources belong to one provider, and another where the resources are shared among different providers (introducing the third party role).
6.1.1 MonoProvider Negotiation

This is the general case where all the resources belong to one provider. Some of these requirements can be introduced in the multi-provider scenario too.

- During the product definition process, a negotiation template is defined with some restrictions such as negotiation margins. This can constrain the way in which the negotiation process is performed against the customer (protocol).
- One of the requirements is the possibility to customize the negotiation protocol, asking what sort of strategies could be adopted taking into account the profile of the customer.
- We can take into account the historic use of the services (the usage profile): This will give an idea of the preferences of the customers in new SLAs. This will help identify which of all the characteristics of an SLA a customer could be most interested in. This way, the appropriate importance can be given to each characteristic during the negotiation process.
- Also, it is desirable to take into account the quality of services selected previously. This is another characteristic of the usage profile. This will help to know what kind of quality for specific services a customer is interested in. In this way, the level of services offered could be adjusted to at least the quality selected before.
- For the provider side, the margin of benefits obtained from the purchase process is usually maximized. Sometimes it is possible to optimize these margins, by minimizing the quality of services being offered if the customer does not use them.
- Another desire of the provider can be to obtain new customers over all, as a priority. From this point of view it could be required that the negotiation process could be flexible towards the requirements of the customers, for instance offering promotions, in order to achieve the final agreement.

6.1.2 MultiProvider Negotiation

In this scenario the resources belong to several providers. So, before the final SLA between the principal provider and the customer can be agreed, it is first required to agree the corresponding SLAs between the principal and the rest of third party providers.

This will influence the protocol of negotiation to follow because there will be different interests for each of the providers. Because of this, a different strategy / protocol may be needed for dealing with each one of these providers.

- The principal provider (service aggregator) could act in different roles requiring it that to follow a different protocol. The service aggregator could act as an agreement broker for, or just as a gateway to, the 3rd Party Providers but the main preference is as a mix of the two former:
  1. (as a broker) the Service Aggregator can decide to accept/reject/customize SLA offers from the Customer without consulting the 3rd Party Providers – i.e. the Service Aggregator negotiates on behalf of the 3rd Parties
2. (as a gateway) the Service Aggregator just sends the Customer’s offer to the 3rd Party Provider – and the 3rd Party decides whether or not to accept/reject/customize the offer.

3. (as a mix) the service aggregator has first to negotiate with the third Party provider. In the end, a negotiation template is agreed where the 3rd Party provider can decide their option (1 or 2). But, in some cases it could be useful to define the negotiation policy as default (1) and in case an agreement is not reached, the system uses the second option (2).

- The historic use of the services (usage profile) can be taken into account: this can assign the initial preferences among the services of the different third party providers.
- Maximized benefit. A specific third party could be selected because the prices of its services are less than the other providers, or have different promotions that in the end make the price less.
- Maximized reliability. As commented above, a provider could be selected before others because its reliability in the past was better than the others.

### 6.1.3 Summary of actors profiles

In this section is explained the different groups of study profiles. From the more generic point of view to specific profiles such as: customer, provider and also business product profiles.

**Generic negotiation profile**

This is a way to define different strategies to execute business negotiation. Each negotiation profile can be defined based on the negotiation profile of customers and providers, and other additional items that can be considered as generic. The details about negotiation profiles for customers and providers are described in the next sections.

**Generic negotiation aspects to take into account:**

- Blacklist: List of parties banned for some reason.
- Blacklist Reason: Reason to be inside the black list.
  - E.g. Party never achieved a successful negotiation in the past despite numerous attempts
- Negotiation Rank: A measure of whether previous negotiations have been Successful or not (an automated value calculated on runtime negotiations).
  - Examples:
    - 0 -> Unsuccessful
    - ...
    - 10 -> Successful
- Average Rounds: the average number of negotiation rounds needed in the past to reach an agreement (automated value calculated on runtime negotiations)
Customer negotiation profile

- Customer rank. There can be different kinds of customers such as personal customers or company customers, or different levels (platinum, gold). Different parameters could be applied over each customer. Their rank could be related to:
  - Annual bill
  - Number of services contracted and used
- Behaviour of the customer. Describes whether the customer in recent times is contracting more services, leaving services, maintaining the same number of services or is a new customer.
- Customer segment. In the case of the customer being a company, it could be interesting to know the segment in which the company sells its products/services in order to know the state of the segment. This could affect the customer behaviour.
- Location aspects. Negotiation results are better if the geographical location/market is taken into account in the process.

Service Provider negotiation profile

- New Businesses Interest Rank: Interest of the provider in achieving new agreements.
- Provider rank. There can be different kinds of providers. Some factors that can help differentiate them are:
  - Type of company: national, multinational, regional.
  - Number of employees.
  - Last year profits.
- State of the company: stable, increasing, decreasing.
- Location aspects. Negotiations can be targeted to the specific geographical region of the provider.
- Maximum total number of penalties for product retirement: A provider can avoid a high amount of penalties for their product. The way to solve this problem is to retire the product if this number is achieved.
- Maximum total value of penalties for product retirement: The same as the above, but taking cost into account.

Business product negotiation profile

For each product that will be sold to the customer, each provider can define some business parameters that can be used to drive the different negotiations between customers and the framework. Some of these parameters can be:

- Forecasted sales per month: We can take into account the value and the number of new contracts that can be expected per month for a specific product.
• Evaluation period for the forecast: The period in which the forecasted sales are calculated.
• Minimum mark-up ratio/Benefits: The minimum sales for the product to be profitable.
• Maximum number of customer penalties for termination: A provider can avoid a high amount of penalties for each customer. The way to solve this problem is to terminate the SLA if this number is achieved.
• Maximum value of customer penalties for termination: The same as the above case but taking cost into account.
• Penalty driven renegotiation: Whether or not the agreement can be renegotiated with the customer because it has penalties.
• Online negotiation with provider: In some cases the negotiation can be done by the business level alone and we can avoid the online negotiation with the lower-level providers.

### 6.2 Customization of generic SLA negotiation protocol Specification

The A2 contribution allows to use a full customizing of the protocol engine of the Generic SLAM. The interaction <<control/track/query>>, and precisely, control interface allows to the Business Manager to change the behaviour of SLAMs components, changing the rules that drives them (it is based on Drools engine).

Next paragraphs describes the way to obtain this customization adding negotiation concepts etc.

The SLA Negotiation is the process leading to the stipulation of a SLA agreement between a customer and a provider. Such agreements can be reached by using different interaction mechanisms amongst the involved parties, but the focus of the project during this year was on single negotiations (one-to-one, one-to-many and many-to-one). In this section we give an overview of the main topics related to the SLA negotiation Customization, with reference to the work done in the work package during project year 2.

Each negotiation mechanism requires that each negotiable variable has to be defined as ascending or descending, specifying in such a way if for each offer exchange, the value for that variable increases or decreases.

Negotiation activity requires, moreover, the definition of some aspects that are strictly related: the requirements, the process, the protocol and the strategy. Let’s examine such parameters in detail.

Concerning the **negotiation requirements**, they define the variables that will be negotiated and the results expected from the negotiation. Negotiation requirements could be related to the process structure – e.g. defining the role for an actor - or variables that have to be negotiated. Each actor has a role in the negotiation; moreover each actor can have more than one role: in particular an actor could be a provider in a certain interaction and a customer in another one. Other requirements are related to the value of specific variables: maximum number of customers/providers, value of negotiable variables, and objective function.
Such requirements are not fixed during the negotiation, and as a consequence the constraints on some of the values can be relaxed or modified during the negotiation process.

In SLA@SOI, the objective function is based on the QoS values exchanged for the SLA definition.

A particular importance is gained by the negotiation process during a negotiation interaction because it specifies all the sequence of activities needed for the interaction among the parties.

A generic negotiation process contains the following phases:

- Admission: all the participants (customer and provider, often referred to in the rest of the document as buyer and seller, respectively) are admitted in the process
- Setup: in this state one of the participants inserts the parameters of the process.
- Pre-negotiation: the object and the parameters of the negotiation are defined
- Change proposals: participants exchange proposals
- Agreement Formation: definition of the agreement among the partners

The following figure reports the conceptual model for the negotiation process:

![Diagram of negotiation process]

**Figure 21: A generic negotiation process**

The admission phase can be executed in parallel with the proposal exchange; the process ends when a certain number of agreements are reached or when the termination conditions occur.
Each negotiation process has its own characteristics formalized at the protocol level; as a consequence each process level classification has a corresponding protocol level classification. Strictly related with the negotiation process is the **negotiation protocol**. The protocol defines the set of condition on the activities that each actor can perform. It defines the conditions and the rules according to which messages can be exchanged and activities can be performed. For each activity, inputs and output are defined. This can be considered as the formalization of the negotiation process. As we can see later in this section, a negotiation protocol is represented by means of its state machine. Each protocol is defined on the basis of the roles (the functionalities of the involved actors), the activities each role can perform, the messages each role can exchange, and finally the rules determining the messages that can be exchanged during the process.

The protocol defines how the information has to be structured during the process in order to go from the initial phase towards the final phase where agreements are possibly finalised. Regarding SLA@SOI, the protocol is provided in the form a .drl file containing the set of the rules that the interaction has to obey. In particular, at the moment, the protocol we decided to use in SLA@SOI is a one-to-one protocol (multi-attribute bilateral protocol). A negotiation protocol is executed by the ProtocolEngine component; major details could be found later in this section, where implementation details are shown.

Another important aspect is the **negotiation strategy**; it defines the decisional models that the parties involved use to establish how to proceed during the interactions leading to the definition of the agreement. In the SLA@SOI framework, the “Planning and Optimisation” Component (POC) is the component dealing with the negotiation strategies and the evaluation and generation of the offer and counter-offer.

One-to-one protocols are characterized by the presence of one customer and one provider during the negotiation. The message exchange between the two involved actors will lead, hopefully, to the agreement among the parties.

A relevant protocol considering the presence of two actors exchanging messages is the multi attribute symmetric bilateral negotiation.

In such a negotiation the proposals are related not only to the price of the product, but are also influenced by more than one attribute referring to the QoS. As a consequence, the maximisation of the objective function of each participant will take into account the combination of such factors. Obviously each actor, having a different role in the negotiation, has its own objective function. The agreement will be reached only at the intersection point of the two functions representing the optimum for both the participants. Another important topic in such a negotiation is the definition of what has to be negotiated. During the setup phase, the negotiable variables can be specified. The negotiable parameters are included in the contract template representing the model of the contract. This model constitutes a sort of dictionary containing the characteristics of the negotiation object. Each proposal exchanged during the business transaction is an instance of this model.

Figure 21 and Figure 23 show the protocol of the bilateral negotiation from the Provider (Seller) and the Customer (Buyer) sides respectively.
Customization of the negotiation requires not only the definition of the specific negotiation protocol to use; in fact, for each negotiation protocol, there are several parameters that could be configured before the interaction starts. During
the Setup phase (see Figure 21) it’s possible to define the parameters according to the preferences. Such preferences could be derived from analysing the negotiation profiles.

The negotiation profiles can impact on the choice of the negotiation protocol and on some of the configurable parameters of the negotiation protocol, as listed in the following.

- The time slot in which proposals can be exchanged.
- The inactivity period constituting the timeout for the process.
- The maximum number of participants for each role in the process.
- The maximum number of counter-offers that can be made during the negotiation process.
- The kind and the amount of information revealed during the negotiation process.
- The exit condition (the agreement reached or the timeout) and what happens if no agreement is reached.
- The negotiation strategy for the participants.

Negotiation profiles can depend on the knowledge of the customers and the providers and this is another aspect that will be taken into account by the work package. In fact, on the basis of the customer and provider profiles we can have a clear understanding of the parties involved in the negotiation interaction and try to propose the most suitable configuration for them. Other than common information about a customer and provider that could characterise their profiles (for example, the presence in a blacklist of one of the party and indeed the reason for its presence, the rank for past negotiations, and the average rounds), and make some negotiation parameters more or less suitable for an interaction, some other information could depend on the role of the party.

In particular, from the customer side, the negotiation profile could be influenced by the past behaviour of the customer, and its rank (in terms of annual bill, number of services contracted and used). Other important information could be related to the customers business segment: if we consider that a customer could be a company, we could be interested in the business segment in which it operates.

Similarly, some information could be more specific to the provider side: other than past penalties for the provisioning of the negotiated goods, it could be useful to know its rank (in terms of its popularity, or the number of its employees) or its location.

### 6.3 Implementation

#### Customization of generic SLA negotiation protocol

As is described in previous section, Business Manager can change the rules that drives protocol engine in SLAMs. There are two operation to do this interaction: setPolicies and getPolicies (one parameter it used to specify if the rules are from protocol engine or for adjustment). To model negotiation, real world business models for negotiation have been considered. These include a fixed price model - where a user either takes what is offered or leaves it, a sealed bid model – where
the users may submit sealed bids to make offers to the provider, a British styled auction – where offers made by users are not sealed from each other, and so on and so forth. This led to the adoption of a protocol description approach whereby a protocol is basically modeled as a state machine representing a negotiation model. Further constraints are then added to this description and provide the means to ensure that the integrity of the protocol is not violated at any stage during negotiation. This protocol description approach is termed as the Generic Negotiation Protocol. The Generic Protocol does not enforce any particular negotiation model; rather it allows the protocol author to tailor it as per their custom requirements.

Another important requirement considered for the Generic Protocol is the flexibility to extend and modify it according to any domain specific needs. It is therefore kept loosely coupled with the Protocol Engine – the software machinery developed to execute it.

Design
The actors of the system, i.e. a customer or a provider, use the same negotiation protocol to communicate. A negotiation session is established on both sides. Negotiation proceeds as per the negotiation protocol. On a higher level, it can be stated that the state machine defined by the protocol stays synchronized between the negotiating participants at all times. The protocol itself is encoded using the Rule Oriented Approach. All protocol constraints are expressed using rules. Drools is the Rule Engine adopted by SLA@SOI. A light weight object model is developed to model the state machine. A taxonomy of events has been developed. Events are passed from the Protocol Engine to the Rule Engine and result in the execution of rules.

The decision to use rules for the protocol description is based on the following reasons:

Expressiveness: The rules allow rich expression semantics in an easy to use “if-then” style. Support for logical operators, querying the knowledge base, and time series is also available out of the box.

Declarative Style: Expressing business logic in a declarative way, as in the Rule Oriented Approach, helps maintain the code over time. On the contrary, if imperative style is adopted, it becomes hard to add, modify or remove the logic and requires recompilation.

Readability: The protocol description is human readable but at the same time is machine interpretable. Experience shows that it is relatively easy to traverse to the right section of the protocol and quickly grasp the logic although it may involve different protocol authors.

Once encoded, the protocol is made available to the SLA Manager, which loads it and executes it as per its INegotiation interface. Before negotiations begin, the administrative users may even change the whole behaviour of the protocol at runtime, by using the IControl interaction.

Rules get fired in an automated fashion, primarily as events from the Protocol Engine arrive into the Rule Engine. Rules may also get fired when their pre-conditions are qualified due to changes in working memory over time. The rules control various aspects of the protocol, some of which are explained below.

Rules may control a state machine’s transitioning logic, and may safeguard a certain number of hops within a state. Coupled together with incoming events, different kinds of validation may be performed, e.g., checking if the invoked
operation was allowed or not based on the current state, or whether the incoming offer obeys certain business specific checks like values of QoS being non null or lying in a certain range. Rules may even implement timeouts controlling the window within which responses are expected to arrive. If a rule results in a negative evaluation, i.e., if a condition violates a protocol constraint, the rule may interrupt the processing of the operation by throwing an exception, which would then be delegated to the caller. In SLA@SOI, the actual negotiation strategies or intelligence to process and generate offers or counter offers is kept in a separate component called the “Planning and Optimization” component. Therefore, the rules are relieved from performing such activities.

An example rule that maintains the number of hops in the NEGOTIATE state of the state machine is shown below.

```
rule "NegotiateState_Evaluate_ProposalArrivedEvent_Rule"
when
    negotiateState : State(name == StateName.NEGOTIATE, status == StateStatus.RUNNING, currentHop < numberOfHopsAllowed );
    event : Event(eventName == EventName.ProposalArrivedEvent);
the
    negotiateState.setCurrentHop(negotiateState.getCurrentHop()+1);
    update(negotiateState);
    event.setProcessedSuccessfully(true);
    retract(event);
end
```

Protocol Engine

The Protocol Engine is a framework component that executes a negotiation protocol. While the latter always tends to get domain specific, the former is designed to remain domain agnostic. The Protocol Engine is part of the Generic SLA Manager, and being so, is inherited by all SLA Managers. To the end user, it basically provides two interactions exposed by the INegotiation and IControl interfaces.

The Protocol Engine allows negotiating parties to successfully connect and negotiate with each other as per the negotiation protocol. For Year 2, it assumes a Bilateral Protocol with multiple rounds of negotiation and also provides for overwriting of a Protocol’s rules before a negotiation session is established between the participants. In Y3, the Protocol Engine may be extended to deal with multilateral negotiations having different visibility rules among negotiating participants. The Protocol Engine is developed as a multi-tier application, where three tiers work together to perform various operations. These are briefly captured by the Figure 24.
Figure 24: Different tiers of the Protocol Engine

In order to maintain compliance with WSAG based clients, the Protocol Engine (PE) works with another integral component of the SLAM, called the Syntax Converter (SC). The Syntax Converter acts as a proxy to the Protocol Engine and for WSAG based clients, converts the WSAG model to the SLA model and vice versa. The Syntax Converter and the Protocol Engine share a 1-1 relationship in terms of their operations. Figure 25 illustrates how a Protocol Engine from the customer’s side interacts over the internet via its Syntax Converter with a provider.

Figure 25: Customers and provider interact using the PE and SC

Interaction

Expanding on the above interaction slightly, it becomes worthwhile to take a closer look at the operations provided by the INegotiation interface (implemented by the Protocol Engine component of the SLA@SOI framework). As shown in Figure 26, negotiation is initiated using the initiateNegotiation operation. This generates a negotiation identifier that is used in the subsequent negotiate operation to submit an offer. As a result, counter offers may be returned back. Exchanging offers and counter offers may continue for several rounds (as per the
protocol). At some point, the initiator requests to create an agreement and invokes the createAgreement operation. This results in the creation of an SLA which is then provisioned at a later stage through the invocation of the provision operation.

![Diagram](image.png)

**Figure 26: A general overview of the negotiation interaction**
7 Assessment Engine

In this chapter is explained the way and process in which the assessment is done, taking into account the different blocks that can affect to the business in terms of finance. Also is defined the customer criteria about his requirements that can be used in the assessment process.

7.1 Assessment and BSLAT Customizing Specification

The objective of the relationship between customer and service provider is to obtain an agreement between them that offers the services required to the customer and returns to the service provider the desired benefits. A negotiation process will be necessary that allows both parties to agree on the characteristics of services required (it is necessary to know if the service provider is able to give the required resources) and the corresponding price.

7.1.1 Offer values checking

The first step during the negotiation process is to check the offer received from the customer against the templates given to them in a previous step of the negotiation process. This step will obtain the differences related to the original template. These differences have to be taken into account in the later negotiation with the different layers/service providers that offer the services described inside the template. These differences will also have to be taken into account in the assessment of the price that it is explained below.

![Assessment process](image)
7.1.2 Policies application

The characteristics offered by a service provider to a customer inside an SLA template are not always the ones desired by them. So, during the negotiation process, these characteristics can change, and will be adjusted to meet the needs of the customer. From the service provider side, there must be a way to assess the price of the offer for the characteristics required by the customer. This will be done by applying different policies that will denote different discounts or additional charges on the price of the final offer.

7.1.3 Promotion and Customization application

Apart from the modification of the characteristics of an offer that was explained above, there are other factors that have to be taken into account in order to assess the final price of the offer. These factors are the promotions and the profile of the customer.

Promotions are used to modify the conditions of the same offer taking into account different factors such as date of contract or location conditions. This is done in order to encourage the introduction of a new product into the market (make it well known) or to increase the sales of it. In a real company it has to be driven by marketing people.

The other factor is the profile of the customer. This profile gives information, obtained from personal information and previous relationships with the customer, about the kind of customer the framework is dealing with. This profile is obtained from the Business Support System (BSS). From this information, it could be possible for the service provider to offer different discounts to the customer in order to introduce the product to them.

7.1.4 Validation of the offer

The final objective of a service provider in the relationship with a customer is the obtaining of benefits. From this point of view, the service provider is not allowed to offer certain services without obtaining a minimum of profit.

The way to specify this will be by indicating the minimum margin that the provider expects from the negotiation process. This will be done by indicating the maximum difference between the prices of the original offer (the original one sent to the customer that contains the desired benefits the service provider wants to earn) and the new one received from the customer in a first step, or the one obtained from the negotiation process with the lower layers of the framework in a following step.

Minimum offer Price

Original offer Price

Maximum difference Parameter

Figure 28: Validation of the offer
This margin will be specified as a parameter during the creation of the product and its associated different offers (i.e. the original offers).

In this step, the difference between the price of the final offer (the one obtained from the customer or from the negotiated process with lower layers) and the original one (the one created during product creation process) will be checked to be not over the maximum difference specified inside the parameter explained above.

This validation will be made in two different parts of the assessment. These are:

- **After the reception of the offer from the customer.** Before starting the negotiation with the other actor of the negotiation process (i.e. other SLA Managers), this offer is checked to be a valid one, that is, that it will provide a minimum of benefits to the service provider. This is made in order to avoid the expense of the negotiation process if the offer is not valid from the first moment. The offer is rejected, avoiding, in this way, the negotiation process with the other actors.

- **After the negotiation with all SLA Managers implied.** During the negotiation process, the Business Manager will negotiate with different SLA Managers in order to know if it is possible to give to the customer the characteristics of the different services desired. It is possible that these SLA Managers are not able to supply the required conditions at that moment requested, and so they could return different offers with the characteristics they are able to fulfil. After the reception of these the Business Manager has to:
  - Build the different composed offers (from the individual offers received from each SLA Manager) that will be returned to the customer taking into account the values received.
  - Check the newly built offers. This includes discarding the ones that do not deliver the minimum benefits and finally returning the remaining ones to the final customer.

### 7.2 Customer Criteria Interface Specification

The negotiation process inside SLA@SOI covers different aspects. One important aspect is the way in which the customer is able to offer their requirements and limitations so that the negotiation process will take them into account in order to assess the best SLA. In this section, some of the different aspects a customer would be interested to have taken into account will be explained, as will how this process will be realized.

#### 7.2.1 Criteria definition

Inside the negotiation process, the customer is able to specify requirements and limitations. These conditions will be taken into account by the assessment module in order to give a customized offer to the customer who requires it. Below are specified some of the requirements a customer is able to specify:

Parameters:
1. Location: Sometimes it is required by law that certain products should be up and running inside the country where they are purchased. For example, if a customer purchases an online storage product, the personal data must not be stored outside the country the customer belongs to.

2. Delivery time. The customer can express that the time between the request of the product and the time when the product is up and running must be less than a specific time.

3. Service provider preference: The customer expresses their preference for a service provider against another, or expresses that they do not want products from a specific service provider.

4. Price limitations: The customer can express that the agreed price must be less than a specific amount.

5. Quality limitations: The customer can express that the quality of the product must be above a specific level.

6. Advertising: The customer can express that their personal data must not be used by the service provider for advertising purposes.

7. Customer Personal Data Protection: The customer can express that their personal data will not be given to a third party in order to be used as data for advertising or other means.

It must be possible to specify other criteria given by the framework with information about the customer:

- Data and Service usage analysis for obtaining anonymous customer behaviour. It is possible to analyse the customer usage information in order to predict the behaviour of the customer. This can be taken into account as another specific characteristic of a customer.

**Evaluation Matrix**

A way to improve the customer experience is the classification of the customer based on their interest. One approach to choose among the different products is to give different values to the different requirements in order to obtain a relative value of each product that will help in the selection of the one that fits the customer requirements the best.

Determining the relative value of each feature can be handled in several ways. One common approach is to rate each major feature from one to five, where five is the most important. This is described for customers in section 8.3.3. But here we are explaining how we can classify products to customers based on the customer behaviour and preferences.

Typically, some providers will perform better in some categories and poorer in others.

The process to follow could be:

- Determine the criteria to evaluate the products.
- Determine the importance of each criterion, and assign a weight to each.
- Score each alternative on all of the selection criteria (using a consistent grading scale, such as one to five, with five being the highest score).
- Multiply each product's scores by the importance ratings to calculate weighted scores.
- Sum each product's weighted scores to calculate a total score.
The table below depicts an example of the matrix resulting from that process.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Importance</th>
<th>Product No. 1</th>
<th>Product No. 2</th>
<th>Product No. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Raw Score</td>
<td>Weighted Score</td>
<td>Raw Score</td>
<td>Weighted Score</td>
</tr>
<tr>
<td>Ease of installation</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>Ease of integration with present system</td>
<td>4</td>
<td>4</td>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>Minimal additional technical staff necessary</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>Intuitive Web interface</td>
<td>3</td>
<td>4</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>Support and training</td>
<td>3</td>
<td>5</td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td>Cost</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Total Score</td>
<td>19</td>
<td>62</td>
<td>19</td>
<td>64</td>
</tr>
</tbody>
</table>

Finally, after the process, the product selected would be product number 3.

### 7.3 Implementation

#### Assessment Implementation

The assessment process is applied when it is needed in the negotiation process. The following diagram depicts the steps to follow during the assessment process in the negotiation of the offer that will be used by the customer to achieve the final agreement.
An offer could be composed of different types of prices such as subscription or monthly fee, so these modifications will apply individually on each one.

The implementation method used to apply the policies is the creation of different rule templates that will be customized (using different parameter values) to each type of modification that will be applied by a rules engine (Drools in our case) in an automatic way.

In the same way as explained above, the implementation method used to define the promotions, will be, again, the creation of different rule templates. They will be adapted to each type of customization to take into account the different conditions over which the rules will be applied, and finally they are applied by the rules engine.

For template rule descriptions see section 4.4.
**Customer Criteria Interface Implementation**

With reference to the parameters defined in this section it is possible to map each one with the implementation.

1. **Delivery time.** In the negotiation interface it is possible to define the provision in another step. It can be automatic or it can be customer ordered using the coordination interface.

2. **Price limitations:** When the customer finds products they manually select the products that fulfil their requirements.

3. **Quality limitations:** When the customer finds the templates of a product, they receive all the templates with the different levels of quality, then they are able to select the quality they wants

The offers evaluation matrix based on different criteria (Service provider preference, Location, Customer Personal Data Protection and Advertising) has not been implemented in year 2.
8 Post-sale Management: Violations, Penalties and Reporting

From the business point of view, the post-sale management is one of the most critical concerns.

8.1 Reporting and Representation of Business Violation Evidence Specification

8.1.1 Overview

The main reason underpinning the need for separating business level SLA reporting from SLA monitoring in the SLA@SOI framework is that SLA guarantee terms might need to be monitored for different purposes that have different reporting requirements.

The average response time of services in a service oriented architecture, for example, might need to be monitored continuously in order to enable automatic replacements of services that fail to satisfy the thresholds set for this guarantee term (adjustment) as well as to provide evidence about the satisfaction or otherwise of corresponding business SLA guarantee terms for the same services. The reporting requirements in this case can be very different depending on the purpose of monitoring. Adjustment, for instance, will typically be based on low level monitoring results that could be used to make automatic decisions about service replacement and is likely to require frequent periodic notifications of the intended consumer of the relevant information (adjustment module). Periodic updates about the satisfaction or otherwise of the same guarantee term however, at the business layer, are likely to be less frequent (e.g., weekly or monthly updates) and require less information than the adjustment layer. At the business layer it might be necessary to report only the weekly or monthly average response time of a service as well as the response time in specific days within the relevant period where the average response time exceeded the threshold value significantly.

To address the diversity of reporting requirements, SLA@SOI has developed a layered architecture where a business SLA reporting module exists on the top of the monitoring infrastructure of the SLA@SOI framework, taking care of the reporting requirements. The primary function of this component is to receive low level monitoring information from the SLA@SOI monitoring infrastructure and to transform it into business SLA monitoring reports (BSLAM reports) by taking into account requirements of business layer components of the SLA@SOI platform. The operation of this module is driven by business reporting policies that determine the monitored SLA terms that should be included in a BSLAM report, the types of monitoring results about these terms that should be included in the
report, and the required frequency of the generation of the reports. The specification of business reporting policies is enabled by the introduction of an XML schema for specifying reporting policies, and an XML schema defining the general representation structure of business level reports.

The architecture for business level SLA reporting and the schemas used to drive the generation of the relevant reports are summarised in the following sections, whilst a detailed specification of them is available Deliverable D.A5a [6].

### 8.1.2 Business SLA reporting architecture

To satisfy the reporting requirements for the generation of BSLAM reports, we have designed a solution that is based on separating the BSLAM reporting functionality from the basic monitoring functionality in the SLA@SOI platform. This separation is related to both the assignment of the basic monitoring and reporting functionalities to different components in the SLA@SOI platform and the use of separate specifications in order to drive these two functionalities.

The responsibility for the generation of BSLAM reports in the SLA@SOI platform is assigned to a component called *BSLAM reporter*. The main functionality of *BSLAM reporter* is to get basic monitoring results from the SLA@SOI monitoring subsystem, and use them in order to generate and provide BSLAM reports to the *BusinessIntelligence* component of *BusinessManager* in the SLA@SOI platform. These reports will be assembled and structured according to BSLAM reporting policies. These policies are specified through the *BusinessIntelligence* component of *BusinessManager* in order to express BSLAM reporting requirements.

The overall architecture of the SLA@SOI platform that relates to BSLAM reporting is shown in the next figure.

![Figure 30: Business monitoring reporting architecture](image)
According to this architecture, a business component of the SLA@SOI platform can specify one or more monitoring reporting policies (referred to as a BSLAM reporting policy in the figure above) and pass them to BSLAM reporter. Following this, BSLAM reporter undertakes the responsibility of generating the relevant business monitoring reports and making them available to the BusinessIntelligence component. To realise this responsibility, BSLAM reporter identifies the low level monitoring information that will be necessary in order to generate the corresponding BSLAM Report, and provides this information to SoftwareLandscape (i.e., the component that keeps a record of the SLA(s) which are being monitored and the exact configuration of monitors and event captors that enable the monitoring of particular SLAs; aka monitoring configuration).

A monitoring policy refers to guarantee terms and business monitoring parameters in particular SLAs. Hence the related SLA guarantee terms and any required events related to these terms can be extracted from the policy by BSLAM Reporter and passed to SoftwareLandscape. Being aware of the exact monitor in the SLA@SOI monitoring infrastructure that will have responsibility for monitoring the particular SLA items required for a BSLAM report, the SoftwareLandscape is able to subscribe BSLAM reporter as a listener to monitoring result events. These events carry monitoring information about the particular SLA items and are generated by reasoning components which have been assigned responsibility for checking the relevant items. The latter components are indirectly accessible through the component called ServiceManager in the architecture (it should be noted that the SLA@SOI monitoring subsystem may deploy different reasoning components, such as the toolkits EVEREST or ASTRO, in order to monitor different SLAs or even different items within the same SLA).

In cases where the SLA terms that need to be reported in a BSLAM report have been assigned to different reasoning components, BSLAM Reporter is registered as a listener of all the reasoning components that produce monitoring results relevant to the SLA items that are included in the BSLAM reports that it will have to generate. Thus, it is ensured that all the required information will be communicated by the reasoning components to BSLAM reporter.

Furthermore, if at runtime, the responsibility for monitoring an SLA item, which is part of a BSLAM report, is moved across different reasoning components of the SLA@SOI monitoring subsystem, the SoftwareLandscape component will be aware of this decision and will itself modify the subscriptions of BSLAM reporter to notifications of monitoring results produced by the new reasoning components accordingly. The multiple subscription of BSLAM reporter does not complicate the communication of monitoring results to it, since all the reasoning components which produce monitoring results publish them to a single common communication infrastructure (see MonitoringEventChannel in the previous figure) which then forwards them to all the listeners that have a registered interest in them. Furthermore, all the monitoring results, which are published by different reasoning components to this communication infrastructure, have the same event structure.

According to the architecture of the previous figure, the SoftwareLandscape component notifies BSLAM reporter of the exact reasoning component that has been assigned responsibility for each SLA item in a BSLAM reporting policy. Then,

---

1 The assignment of different SLAs or items of a single SLA to different reasoning components might be dictated by the need to make use of the specialised capabilities of individual reasoning components or other requirements (e.g. balancing the load across these components).
it sends to BSLAM reporter updated notifications for new reasoning components that may be deployed for the SLA items required by BSLAM reporter in cases where changes in the monitoring assignments of the relevant SLA items become necessary (e.g. if some reasoning component becomes unavailable). Such notifications of new reasoning components are necessary for BSLAM reporter in cases where it needs to obtain primitive monitoring information and include it in a BSLAM report in order to demonstrate the validity of derived monitoring results.

The above business monitoring reporting architecture assumes that such primitive information is not sent to BSLAM reporter on a routine basis as this would be too expensive due to the potentially vast volume of primitive monitoring data required. If, nevertheless, a business layer component envisages the need to request such primitive information for an SLA item at some point, it should include a relevant specification in the BSLAM reporting policy that it provides to BSLAM reporter. The latter component will then have responsibility to request the SoftwareLandscape component to configure the reasoning component that has responsibility for monitoring the relevant SLA item so as to maintain the required primitive monitoring information and provide it to BSLAM reporter if requested. It should also be noted that, in general, the availability of such primitive monitoring evidence may be needed for different time periods. Hence, to avoid the maintenance of vast amounts of unnecessary monitoring information in the reasoning components, it is assumed that the period over which primitive monitoring information might be required needs to be specified as part of reporting policies.

The elements of monitoring evidence which are used internally by the monitoring subsystem (e.g. primitive monitoring events), or sent by this infrastructure to BSLAM reporter, might be required to be digitally signed and the components involved in the production of these elements might be required to be certified in order to reduce the scope of repudiation. The ability to digitally sign elements of monitoring evidence provides a means for confirming that these elements have indeed been produced by the component that claims to be their generator.

It should be noted, however, that digital signatures provide no guarantee that all the runtime events which have occurred during the operation of the computational entities governed by an SLA (e.g. services, infrastructure elements) will be captured and taken into account by the components of the monitoring subsystem that have responsibility for capturing and performing checks upon them. This is a much broader issue related to the trustworthiness of the components involved. To address this issue of trustworthiness, we adopt the use of certificates. Certificates in general are used to guarantee different types of properties of software components (e.g. their identity, correctness etc) and can be produced on the basis of different component verification methods including, for example, formal proofs and testing. In the context of the SLA@SOI monitoring subsystem, a certificate should at least guarantee the identity of a given external component (e.g., an event captor or reasoning component) as well as of the infrastructure as a whole.

During the operation of the monitoring infrastructure, the MonitoringManager has responsibility for ensuring that the event captors and monitors that are deployed for monitoring a particular SLA have the required certificates, and the BSLAM reporter has responsibility to check that the monitoring evidence produced by reasoning components is digitally signed if the latter condition is required by a business level SLA reporting policy.

---

2 Further properties might need certification beyond this minimal level. A discussion of such properties is available in [6].
8.1.3 Business level SLA reporting policies

BSLAM reporting policies are related to SLAs, (business) SLA@SOI components and BSLAM reports as shown in Figure 30. More specifically, a BSLAM reporting policy refers to a single SLA and should always be associated with at least one SLA (as otherwise the guarantee terms and monitoring parameters in it would not be well defined). A single SLA, however, may have zero or more BSLAM reporting policies. This might happen either because different SLA@SOI components have their own reporting requirements for the same SLA or because a single component might have different reporting frequency requirements for the same item within the same SLA. It should be noted that a BSLAM reporting policy is allowed to specify only a single reporting interval for all the SLA items which are included in it. Thus if a component has different requirements regarding the reporting frequency for different items within the same SLA, it should specify different BSLAM reporting policies for these items.

![Figure 31: BSLAM Reporting Scheme Entity Relationship Model](image)

BSLAM reporting policies are restricted to one SLA only since a reporting policy is expected to be applicable during the lifetime of an SLA and determine a regular interval for producing consecutive policy reports. Since different SLAs may have different life times and reporting interval requirements, associating a single BSLAM policy with more than one of them could lead to either altering the reporting requirements of one of the SLAs or ending up with reports that include replicated information. If an SLA1, for instance, has a reporting interval of 2 days and an SLA2 has a reporting interval of 3 days, on the second day after the activation of the policy a BSLAM report would include monitored terms from SLA1 but it would not be clear whether it should also include terms of the SLA2 that have been monitored up to that day (since there is no such requirement for SLA2). Similarly on the third day, the BSLAM report should include terms of SLA2 and the terms of SLA1 up to the second day but it would not be clear whether it should also include terms of SLA 1 for the third date. To avoid such ambiguities BSLAM reports are restricted to single SLAs with a single reporting period.

A single BSLAM reporting policy will in general drive the generation of zero or more BSLAM monitoring reports. A BSLAM reporting policy may lead to the generation of zero BSLAM reports because there is always – at least in theory – the possibility of being unable to monitor services in ways that would enable the generation of even one BSLAM report. Normally, however, a BSLAM policy will
lead to the generation of more than one BSLAM reports – as many as the frequency of the report generation requires in the policy.

8.1.4 Business SLAM reporting policies

The specification of a BSLAM reporting policy includes the following basic parts:

- A (unique) identifier of the policy.
- A (unique) identifier of the component of the SLA@SOI platform that requires the reports to be produced according to the policy.
- A reference to the SLA that the policy applies to. The reference includes the identifier of the SLA and a unique resource identifier indicating where the specification of this SLA can be retrieved from.
- Temporal reporting preferences of the policy which are determined by the policy’s start date and time as well as the periodic time interval that reports should be generated.
- One or more items in the SLA that the policy applies to that should be included in the reports generated by the policy (reported items).

To illustrate the basic features of this schema, in this section we discuss an example of a BSLAM reporting policy specified according to the schema. The XML specification of this example is shown in Figure 32 and refers to a policy identified as BSLAMRP_01.

The policy (BSLAMRP_01) specifies business level monitoring reporting requirements for the following guaranteed term specified as part of an SLA SLA_01:

**SLA guarantee term GT_01**:  
Qualifying condition (QC): Invocation_Rate(S_01, 1 hour) < 60 invocations/hour  
Service Level Objective (SLO): mean(Completion_Time(S_01) 1 hour)) < 0.2 sec

BSLAMRP_01 applies to SLA_01 as indicated by the content of the element <slaId></slaId> and the requestor of the policy is the component Business_Control_and_Tracking. BSLAMRP_01 specifies one item to be reported, namely RI_1. The type of this item is GTassessmentResult indicating that the item refers to assessments of SLA guarantee terms (as opposed to business monitoring parameters). RI_1 is specified to include assessment results for GT_01, as indicated by the contents of the elements <GTId></GTId> in its specifications. The types of the assessments of interest for the reported item is “violations” as indicated by the sub-element <SLOAssessmentResult assessmentresult="violation"/> of RI_1.

BSLAMRP_01 specifies that reports should be generated and, therefore, take into consideration monitoring information occurring from 9.00am on January 14, 2010 onwards as indicated by the contents of the element <reportingStartDateAndTime/></> in the policy. Furthermore, it specifies that reports should be generated every 5 hours (according to the element <reportingTimeInterval unit="hour" value="5"/> in the policy). The period and

---

3 The definition of the guarantee term in this example is based on the Abstract Syntax Model developed for specifying SLA guarantee terms in work package A5.
frequency of the BSLAM report defined by the policy applies to both RI_1 and RI_2.

```xml
<?xml version="1.0" encoding="utf-8"?>
<!-- Created with Liquid XML Studio 1.0.8.0 -->
<BSLAMReportingPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="C:\spanoudakis\Projects\SLAatSOI\Deliverables\D.A2_5b\BSLAMReportingPolicy.xsd" policyId="BSLAMRP_01" requestorId="Business_Control_and_Tracking">
  <monitoredSLA>
    <slaId>SLA_01</slaId>
    <slaURI>http://sla_repository.eu</slaURI>
  </monitoredSLA>
  <reportingStartDateAndTime>
    <startDate>2010-01-14</startDate>
    <startTime>09:00:00</startTime>
  </reportingStartDateAndTime>
  <requiredEvidenceReportConfiguration configurationId="conf_01" evidenceType="derived">
    <evidenceAvailabilityTimeWindow unit="day" value="3" />
    <evidenceDispatchMethod dispatchMethod="direct" security="signed" />
  </requiredEvidenceReportConfiguration>
  <requiredEvidenceReportConfiguration configurationId="conf_02" evidenceType="primitive">
    <evidenceAvailabilityTimeWindow unit="day" value="5" />
    <evidenceDispatchMethod dispatchMethod="onDemand" security="signed" />
  </requiredEvidenceReportConfiguration>
</BSLAMReportingPolicy>
```

**Figure 32: Example of BSLAM Reporting Policy**

Furthermore, `BSLAMRP_01` specifies two evidence report configurations for RI_1. The first configuration (`conf_01`) refers to the evidence that should be included in the generated reports as indicated by the value of the attribute `dispatchMethod` of the evidence dispatch element of the configuration that is "direct". The evidence indicated by `conf_01` is derived events (as specified by the value of the attribute `evidenceType` of `conf_01`). This means that all the derived events that are generated by the SLA@SOI infrastructure for the SLO of GT_01 should be included in the report for RI_01. In this case, the derived evidence is the hourly average completion times of S_01. This evidence is derived since it refers to the average completion times that are derived through some computation over a series of individual completion times.
The second configuration (conf_02) specifies that the evidence required for RI_01 in a report should also include the primitive events which are captured and processed by the SLA@SOI monitoring infrastructure in order to assess GT_01. This is indicated by the value of the attribute evidenceType of the configuration that has been set to “primitive”. Thus, conf_02 requires that all the service operation invocation and response events should be included in a report since events of these types are required in order to assess the guarantee term. These primitive events, however, should not be routinely notified to BSLAM reporter by the monitoring infrastructure but should be available for retrieval by BSLAM reporter within a specified time window after their processing, as indicated by the value of the attribute evidenceDispatchMethod of conf_02 that has been set to "onDemand". The time window for maintaining these primitive events is indicated by the value of the element <evidenceAvailabilityTimeWindow unit="day" value="5" /> of conf_02. Finally, the primitive events required by conf_02 should be digitally signed as indicated by the value of the attribute security of the evidence dispatch element of the configuration.

An extended version of the example in the previous figure is given in [6].

8.1.5 Structure of BSLAM reports

The reports generated according to BSLAM policies within the SLA@SOI infrastructure need also to be represented in XML. The schema that defines the structure of such reports is overviewed in this rest of this section along with an example of a BSLAM report. A BSLAM report includes:

- a unique identifier of the report;
- the identifier of the BSLAM policy according to which the report has been produced;
- the identifier of the component for which the report has been produced;
- a reference to the SLA that the report is concerned with;
- a time stamp indicating the time of the generation of the report; and
- one or more elements representing the monitoring results for the SLA items that should be covered by the report (i.e., the reported item instance elements).

The specification of the reported item instance elements includes:

- An identifier for item instance;
- The identifier of the SLA item that the reported item refers to;
- The identifier of the monitoring entity (i.e., the component of the SLA@SOI monitoring infrastructure) that has produced the monitoring result included in the report (this is included in order to be able to authenticate and check the authorisation of this entity to produce these results);
- The time when the last primitive monitoring event that has been taken into account by the SLA@SOI monitoring infrastructure for the assessment of the relevant item occurred, known as occurrence time;
- The time when the reasoning component of the SLA@SOI monitoring infrastructure generated the assessment of the relevant SLA item, that is known as detection time; and
- The time when the reasoning component of the SLA@SOI monitoring infrastructure notified the assessment of the relevant SLA item to the BSLAM reporter, known as reporting time.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<BSLAMReport policyId="BSLAMREP_01" requestorId="Business_Control_and_Tracking"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="file:/foo/BSLAMReportSchema.xsd">
    <reportTimestamp>
        <date>2010-01-14</date> <time>14:00:00</time>
    </reportTimestamp>
    <monitoredSLA>
        <slaId>SLA_01</slaId> <slaURI>http://foo.eu/slaURI</slaURI>
    </monitoredSLA>
    <reportedItemInstance reportedItemId="RI_1"
        reportedItemInstanceId="RI_1_1">
        <GTAssessmentResult GTId="GT_01" GTAssessmentResult="violation">
            <SLOAssessmentResult result="violation">
                <QCTermValue>58</QCTermValue>
                <KPIValue>0.23</KPIValue>
            </SLOAssessmentResult>
        </GTAssessmentResult>
        <detectionMonitoringEntityId>RCG1</detectionMonitoringEntityId>
        <occurrenceTime><date>2010-01-14</date> <time>12:48:00</time></occurrenceTime>
        <detectionTime><date>2010-01-14</date> <time>13:01:00</time></detectionTime>
        <reportingTime><date>2010-01-14</date> <time>13:05:00</time></reportingTime>
        <evidence evidenceReportConfigurationId="conf_01"
            type="derived" dispatchMethod="reportsInclusive">
            <involvedEvents>
                <involvedEvent>
                    <EventID>RCG1_I_R_150</EventID>
                    <EventContext>...</EventContext>
                    <EventPayload>...</EventPayload>
                </involvedEvent>
                <involvedEvent>
                    <EventID>RCG1_AVG_RT_155</EventID>
                    <EventContext>...</EventContext>
                    <EventPayload>...</EventPayload>
                </involvedEvent>
            </involvedEvents>
        </evidence>
    </reportedItemInstance>
</BSLAMReport>
```

**Figure 33: Example of BSLAM Report**
The previous figure shows an example of a BSLAM report generated according to the BSLAM reporting policy BSLAMRP_01 that is shown in the figure in section 8.1.4.

This report is identified as BSLAMR_01 and is time stamped at 2pm on the start date of the policy underpinning its production. This report includes one reported items for the SLA term RI_1, namely the reported item RI_1_1. This item contains information regarding a violation of the service level objective GT_01. The actual values of the invocation rate and the average completion time that violated GT_01 are reported as the values of the QCTermValue and KPIValue elements respectively and are equal to 58 invocations per hour and 0.23 sec respectively.

The occurrence time of RI_1_1 is equal to the timestamp of the last S_01 response event that was taken into account during the computation of the hourly average S_01 completion time between 12 pm and 1 pm. The detection time of RI_1_1 is the time point when the reasoning component which had responsibility for the monitoring of the relevant item in the SLA@SOI infrastructure (i.e., the component with id “RCG1” as indicated by the contents of the element detectionMonitoringEntityId) computed the hourly average S_01 completion time between 12 pm and 1 pm. Finally, the reporting time of RI_1_1 contains the time when RCG1 generated and dispatched a monitoring event regarding the computed result of the aforementioned average S_01 completion time to the SLA@SOI communication infrastructure.

Derived evidence is also included in RI_1_1. This evidence has been included due to the evidence report configuration conf_01 (see the value of the attribute evidenceReportConfigurationId) as specified in the reporting policy BSLAMRP_01. The reported evidence consists of two derived events, namely RCG1_I_R_150 and RCG1_AVG_RT_155. The former event contains the computed invocation rate of S_01 (i.e., 58 invocations per hour) by considering all the S_01 invocation events that occurred between 12 pm and 1 pm on the BSLAMRP_01 policy start date. The second event contains the computed S_01 average completion time (i.e., 0.23 sec). This time has been computed by considering all the S_01 response events that occurred between 12 pm and 1 pm on the BSLAMRP_01 policy start date.

It should be noted that in the policy BSLAMRP_01, RI_1 specifies another evidence report configuration (i.e., conf_02). According to this configuration, the Business_Control_and_Tracking requestor component is eligible to request the primitive evidence regarding the reported values of RI_1_1, within a 3-day time window after the time point indicated by the timestamp of the example report (see the element reportTimestamp). As the latter information is only available upon request, it is not included in a regularly produced report for the policy such as the one shown in last figure of this section.

### 8.2 Violations and Penalties management Specification

From a Business perspective, the importance of violation management is clear. Most Guarantee Terms set in a Service Level Agreement are subject to direct financial penalties or indirect financial repercussions if not met. Hence the relevance of detecting all the violations that occur during the service lifetime, the analysis of these violations against the corresponding SLAs, and the calculation of the penalties is determinant in the business.
Inside the SLA@SOI framework, the business layer is the one that takes care of the economical impact derived from the performance of the service. The Violation and Penalty Management Component of the Business Manager must analyze and correlate information from two different sources: from the signed Business SLAs (stored in the SLA Registry), and the runtime information about the performance of the service.

Regarding the first source of information, it has already been explained in section 4.1.2 that the agreement terms may have a penalty attached when a given guarantee term is breached. This penalty requires a financial amount to be paid, either by the Service Provider, when the bad performance is caused by a failure at the software or infrastructure resources, or by the customer, in those cases where they are responsible for the situation (e.g. when the number of requests exceeds the maximum threshold indicated in the SLA).

On the other hand, the Violation and Penalty Management Component must implement a mechanism to receive the runtime information about each SLA from the software and infrastructure levels. The incoming data, namely SLA Violations, must then be checked against the conditions signed in the SLA. Once one of these notifications is received, the Violation and Penalty Management Component must evaluate if the corresponding SLA establishes a penalty for this specific violation. If it does, then it is in charge of calculating the exact penalty and applying it, through communication with the Billing Engine.

Furthermore, a mechanism must be implemented to allow the persistent storage of the information generated: the violations of each SLA, the duration of the malfunction and the penalties attached. This information will be used by the Reporting Component to generate the corresponding reports, making the data available both internally to the Service Provider, and externally to the customer.

### 8.3 Customer Relations Specification

This is the interaction (ICustomerRelation) between the Customer and the Business SLA Framework.

We can split the different customer interactions as follows:

- **Customer Relationships Management:** This functionality entails the registration, authorization, authentication and management activities.
- **Customer reporting:** All the information about consumed services interchanging with customers.
- **Customer feedback collecting:** Capturing the customers impression about the consumed services.

#### 8.3.1 Customer Relationships Management

The customers have to be registered in the framework and internal components of the framework have to check if the customers are valid to interact with them. This functionality can be summarized as follows:

- **Customer registration:** Allows Customers to register themselves within the framework in order to interact later with it. A customer has to have a registered identity to interact in each business interaction. The value of the customers operations is very important to track and adapt the business to them. The customer has to load data and detailed information
about them. They can be an individual person or an organization or company.

- Customer authentication: Allows the framework to authenticate the customer in order to allow them to interact with it. It must include a way to verify that the customer is who they say they are.
- Customer authorization: For the administrators or business managers of the framework, it must be possible to reject certain customers and prevent them from interacting with the framework for a sale process, should they be on a black list for example.

**8.3.2 Push/Pull Reporting**

This functionality generates reports with information about a given product, its attached SLA and the performance of the service. Some requirements for this component are the following:

- Reports must be generated in a pdf file.
- Reports must contain the terms of the Business SLA, that is, the conditions under which the service must be provided. This will support the revision of the SLA definitions and components if required.
- Reports must contain information about the performance of the service, namely violations incurred and the penalties attached.
- The Reporting component must support requests about the status of a given SLA and of a given GuaranteeTerm of an SLA.
- The information must be available on demand, through a web service interface, to be used both from the customer side and from the Service Provider GUI.
- Specific reports must be automatically sent to the customer: periodically, to inform them about the behaviour of the service, as well as per the specification of reporting terms inside the SLA (e.g. when a violation occurs). These automatic reports will be sent by email to the customer address.

**8.3.3 Customer rating**

The Products and Services can be rated by a Customer. The final aim of a post sale framework is to collect and capture Customers feedback about the consumed products and services. Detecting the weakness in the customer relations or problems in the products consumption, the framework can help enhance the products and solve the problems found by customers. Furthermore, making customer ratings visible to prospective customers will help them gauge the robustness of the SLAs of the products and services in which they are interested, and make an informed purchasing decision.

**8.4 Implementation**

**Business Violation Evidences Representation Implementation**

The implementation of business violation evidences representation and the business SLA reporting layer in the SLA@SOI infrastructure will take place in the
third year of the project. However some reporting facilities have been developed in the framework. The implementation details are included in this chapter under Customer Relationships Management and Push/Pull reporting.

Violations and Penalties management Implementation

At software and infrastructure levels, the Low Level Monitoring System performs a continuous supervision of the parameters that characterize the service. The collected information is then fed into the SLA Level Monitoring System, which compares the instrumentation data with the conditions specified in the SLA as GuaranteeTerms. In case a violation in one of those terms is detected, a specific SLAViolation event is triggered and captured by the adjustment component of the corresponding layer. The information flows to the Business layer through the <<control/track>> interaction, for the evaluation of the economical impact of the underlying problem, realized by the Violation and Penalty Management System. The output of this analysis will be used at the business layer in the Post-sale sub-component, and it should permanently be stored in the database.

The violations and penalties management system has been embedded in the Business Adjustment component. The reason for that is the close relationship between both functionalities. On the one hand, the input data is in both cases the violation events produced at software and infrastructure layers. On the other, both modules aim to analyze those events, and take the corresponding actions: a recovery action in the adjustment functionality, the application of a penalty in the other case.

Therefore, the implementation of the violations and penalty management module extends the approach already described in section 4.4. Specific rules have been added to the rule engine, that, once a violation arrives:

1. Retrieve the corresponding SLA from the SLA Registry.
2. Check the SLA establishes a penalty charge for the Guarantee Term that has been violated.
3. If it does, then the information corresponding to the penalty is extracted, and its value calculated.
4. The information regarding the application of the penalty is stored in the database. This information will be included in the generated reports to the customer, and it will be taken into account in the billing procedures.
5. Whether the violation has a penalty attached or not, it is stored in the database, for reporting purposes.

The SLA model has been extended to include business-related terms. It establishes two business terms associated with penalties: penalty count and violation count. Penalty count is defined as the number of penalties applied to a service due to the reception of a violation of the signed conditions of a service, while violation count is the number of notifications received of violations to an agreement-term, guaranteed state or guaranteed action of a service. More detailed information about how the Guarantee Terms and penalties are described in the SLA has already been described in section 4.1.2.
Customer Relationships Management

A set of web services have been defined in order for the customer to register themselves in the framework. The Customer may provide detailed information about them in the register. If the customer is not registered and authorized to interact with the framework, they cannot buy any product or service.

Another operation defined as a web service is authentication. It allows the checking of the customer.

Internally other operation are accessible that allows the status of the customer to be changed. It is used to restrict the access of a customer if necessary, as in a black list.

Push/Pull Reporting

A specific component has been implemented for report generation, both for the customer and the service provider. The component offers a webservice interface through which reports can be requested on demand. Reports can be retrieved by product and client, by SLA id or by a specific GuaranteeTerm of a given SLA. Aggregated reports for a specific product can also be requested. The output includes the information of the associated SLA and the performance of the service, in terms of violations and penalties.

While the methods described above can be considered as a pull interface, the component also implements a push one, providing an automatic mechanism to send reports to the customer. These reports are sent with the periodicity indicated in the SLA (see section 4.1.1 for the syntax of the Reporting information in the SLA model), but also when a violation occurs. To this end, an email is sent to the contact email address of the customer.

The content of the report is retrieved from the business database, where all the business information is stored: product description, business SLA (from the SLA registry), prices, violations and penalties. If more detailed monitoring information is needed, the Reporting Component can use the <<control/track>> interaction to retrieve this information from the registries in the underlying SLAMs and from the Low Level Monitoring System Database.

All this information is structured into reports in pdf format. To this end, the Reporting component uses the iText library [5], which allows dynamic pdf document generation and manipulation. In the case of the on-demand reports, the pdf file can be viewed using a web browser. In the case of the push interface, the file is attached to the email sent to the customer.

Customer rating

In the Business Web Tool GUI, a way has been provided to collect the feedback of the customers about a product. The customer has to rank the services (SLAs) included in the product, and then in an indirect way they rank the product. The system aggregates the rated information in order for it to be queryable. A product
can use different SLAs from different providers and each service can have
different rating intervals. Part of the aggregation functionality is to normalize the
rates to be able to give a summarized value for the product.
9 Conclusions

9.1 Summary of the contributions

In this document we have studied many aspects around business level management of SLAs and report the results from the second year of the European research project SLA@SOI. This includes business processes, business data models, and business terms. We have also defined and described business components and modules which fit inside the overall framework architecture, oriented to negotiate automatic agreements between customers and providers.

The design of the business framework has been based on the SLA model and generic components defined in A5 and some derived classes from SID model from TMForum [12].

The basic business framework is consistent and well defined, and was successfully implemented in the Business Manager and Business SLA Manager modules. This is an enhancement of the eContracting module of project year 1, being enriched in project year 2 with some novel features such as business SLA customization, business SLA model terms enhancement, ....

Business SLA Model and Terms were refined and extended based on B-line requirements and other information sources, as described in Section 4.1. Other important developed areas were product management and discovery and business adjustment features. The department relationships were defined to help construct business elements in the graphical user interface used for business management (Business Web Tools).

The document also addressed the customization of the negotiation protocol which specifies the roles and set of activities that the parties have to perform in order to achieve an agreement. During the interactions some of the parameters can be customized depending on the characteristics of the parties: the time slot of the interactions, the maximum number of counter offers, the termination conditions and so on. Some of the parameters can be automatically defined on the basis of the profiles of the customers and the providers.

To address the needs for business level reporting of SLA monitoring information, we have introduced a new component in the architecture of the SLA@SOI platform, the BSLAM reporter. The primary function of this component is to receive low level monitoring information from the SLA@SOI monitoring infrastructure and transform it into business SLA monitoring reports (BSLAM reports) by taking into account the requirements of the business layer components of the SLA@SOI platform.

The business assessment was extended with new business elements such as promotions and SLAT customization based on customer profiles.

To support the generation of BSLAM reports we have introduced two XML schemas. The first of these schemas enables the specification of policies regarding the generation and structure of BSLAM reports. Such policies can be defined by components of the SLA@SOI platform in order to indicate the monitored SLA terms that should be covered in a BSLAM report in order to address their needs, the types of monitoring results about these terms that should be included in the report, and the required frequency of the generation of
the reports. The second schema defines the general representation structure of
the business SLA monitoring reports. This structure includes variation points
where each report is customized depending on the policy that underpins it.

Previous work on violations and penalties was extended, taking into account
historical information stored previously.

Finally, the post-sale management component was developed with customer
relationship features like customer registration, automatic generation of rich
reports, and customer rating of services.

9.2 Outlook to next steps

The objectives for the business framework in the last project year will be to finish
the eContracting suite. There will also be further in depth study of some aspects
related to business layer including:

- commercial creation process,
  - holistic business SLA composition; use of algorithms and rules for
    merging terms of the SLATs in a smart way
  - business adjustment graphical interfaces
  - service usage data mining to be reused in commercial creation
    process
- negotiation,
  - business negotiation driven by customer and provider profiles and
    other business information
  - renegotiation
  - full business negotiation framework
- assessment,
  - new constraints, limits and specific conditions to be taken into
    account
- post-sale management.
  - arbitration & penalty management mechanism implementation
  - violations & penalties escalated to human decisions
  - customization of reports for customers
  - customer rating external interface implementation
  - billing features

With these new features we will cover all the functionality required by a full
eContracting suite, suitable for companies wanting to trade and sell services on
the SLA@SOI framework.
10 References


Appendix A: Glossary

The following list shows the most important entries of the SLA@SOI glossary. Note that terms that are specific for the current document and not part of the overall project wide glossary are marked with an asterix *.

**Agreement Initiator**
An agreement initiator is a party to a service level agreement. The initiator creates and manages an agreement on the availability of a service on behalf of either the service customer or service provider, depending on the domain-specific signalling requirements.

**Agreement Offer**
An offer is the description of the agreement relationship that is sent from agreement initiator to agreement responder during agreement creation, indicating the relationship which the initiator would like to form.

**Agreement Responder**
The agreement responder is a party to a service level agreement. The responder implements and exposes an agreement on behalf of either the service provider or service customer, depending on the domain-specific signalling requirements.

**Agreement Template**
An agreement template is an XML document used by the agreement responder to advertise the types of offers it is willing to accept.

**Agreement Term**
Agreement terms define the content of a service level agreement.

**Business Service**
A business service is exposed/invoked via at least some non IT elements.

**Business Manager**
A specialization of service provider: person that defines the SLATs of products and joins available services in a product.

**External Service**
External services are exposed across the boundaries of an organization, i.e. across at least two administrative domains.

**Framework Administrator**
A specialization of service provider: person that configures/adapts the SLA@SOI framework for a specific application.

**Guarantee Term**
Guarantee terms define the assurance on service quality associated with the service described by the service definition terms. They refer to the service description that is the subject of the agreement and define service level objectives, qualifying conditions and business value expressing the importance of the service level objectives.

**Hybrid Service**
A hybrid service is a set or bundle of other services where all these services are exposed to the customer but have different service interface types (e.g. an IT service and a business service).
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure Manager</td>
<td>A specialization of <em>infrastructure provider</em>: person/system that is interested to measure and control infrastructure properties.</td>
</tr>
<tr>
<td>Infrastructure Provider</td>
<td>A specific kind of service provider that focuses on the provisioning of <em>infrastructure services</em>.</td>
</tr>
<tr>
<td>Infrastructure Service</td>
<td>An infrastructure service is a specific <em>IT service</em> which exposes resource/hardware-centric capabilities.</td>
</tr>
<tr>
<td>Internal Service</td>
<td>Internal services are exposed within the boundaries of an organization, i.e. within one administrative domain.</td>
</tr>
<tr>
<td>IT Service</td>
<td>An IT service is exposed/invoked by means of information technology. Specific classes of IT services may be software services, infrastructure services or media services.</td>
</tr>
<tr>
<td>Offered Service</td>
<td>An abstract service (more precisely: service type) which is offered by a specific <em>Service Provider</em> to its <em>Service Customers</em>.</td>
</tr>
<tr>
<td>Operation Level Agreements</td>
<td>A specification of the conditions under which an internal service or a component is to be used by its “customer”.</td>
</tr>
<tr>
<td>Service</td>
<td>A means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks. See also <em>service interface type</em>, <em>service concreteness</em>, <em>service exposure</em></td>
</tr>
<tr>
<td>Service Concreteness</td>
<td>The stage a service reaches over time from a fully abstract type to actually instantiated. See also <em>service type</em>, <em>offered service</em>, <em>service implementation</em>, <em>service instance</em></td>
</tr>
<tr>
<td>Service Consumer</td>
<td>Person(s) who actually consume/use the provided services. Typically they belong to the <em>service customer</em>.</td>
</tr>
<tr>
<td>Service Customer</td>
<td>Someone (person or group) who orders/buys services and defines and agrees the service level targets.</td>
</tr>
<tr>
<td>Service Description Term</td>
<td>Service Description Terms describe the functionality that will be delivered under the <em>service level agreement</em>. The agreement description may include also other non-functional items referring to the service description terms.</td>
</tr>
<tr>
<td>Service Exposure</td>
<td>Services can be exposed either internally (within the same administrative domain) or externally. See also <em>internal service</em>, <em>external service</em></td>
</tr>
<tr>
<td>Service Implementation</td>
<td>A service implementation is a possible concrete realization of a given <em>service type</em>.</td>
</tr>
<tr>
<td>Service Instance</td>
<td>A concrete realization of an <em>offered service</em> which is ready for consumption by service users. It relies on the instantiations of all the resources required for a given <em>service implementation</em>.</td>
</tr>
<tr>
<td>Service Interface Type</td>
<td>Describes the nature of an actually exposed service, i.e. about the nature of his invocation interface. See also <em>business service</em>, <em>IT service</em>, <em>hybrid service</em></td>
</tr>
<tr>
<td>Service Level Consequence</td>
<td>An action that takes place in the event that a service level objective is not met.</td>
</tr>
<tr>
<td>Service Level Agreement</td>
<td>An agreement defines a dynamically-established and dynamically managed relationship between parties. The object of this relationship is the delivery of a service by one of the parties within the context of the agreement. The management of this delivery is</td>
</tr>
</tbody>
</table>
achieved by agreeing on the respective roles, rights and obligations of the parties. The agreement may specify not only functional properties for identification or creation of the service, but also non-functional properties of the service such as performance or availability. Entities can dynamically establish and manage agreements via Web service interfaces.

Service Level Objective
Service Level Objective represents the quality of service aspect of the agreement. Syntactically, it is an assertion over the agreement terms of the agreement as well as such qualities as date and time.

Service Provider
An organization supplying services to one or more internal customers or external customers.

SLA Manager
A specialization of service provider: person/system that is responsible for managing SLATs and SLA relationships.

Software Designer
A specialization of software provider: person that designs/develops the architecture and components of a specific SLA based application.

Software Manager
A specialization of service provider: person that defines software-based services, takes care of their management and supports the SLA manager in creating appropriate SLA templates.

Software Provider
An organization producing software components which might be used by a service provider to assemble actual services.

Software Service
A software service is a specific IT service which is exposed/invoked by means of software entities such as Web services, user interfaces, or software-based business processes.

Software Component
Software components are the entities produced at design-time by a software provider.

Service Type
A service type (or abstract service) specifies the external interface of a service possibly including non-functional aspects. It does not specify any means (components, resources) which are needed for the actual provisioning of that service.
## Appendix B: Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOP</td>
<td>Aspect Oriented Programming</td>
</tr>
<tr>
<td>BM</td>
<td>Business Manager</td>
</tr>
<tr>
<td>B-SLAM</td>
<td>Business SLA Manager</td>
</tr>
<tr>
<td>EMF</td>
<td>Eclipse Modelling Framework</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>IE</td>
<td>Interaction Event</td>
</tr>
<tr>
<td>FCR</td>
<td>Finite capacity regions</td>
</tr>
<tr>
<td>Infr-SLAM</td>
<td>Infrastructure SLA Manager</td>
</tr>
<tr>
<td>Infr-SM</td>
<td>Infrastructure Service Manager</td>
</tr>
<tr>
<td>IoC</td>
<td>Inversion of Control</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicator</td>
</tr>
<tr>
<td>LLMS</td>
<td>Low Level Monitoring System</td>
</tr>
<tr>
<td>LQN</td>
<td>Layered Queueing Networks</td>
</tr>
<tr>
<td>MA</td>
<td>Manageability Agent</td>
</tr>
<tr>
<td>MRE</td>
<td>Monitoring Result Event</td>
</tr>
<tr>
<td>MVC</td>
<td>Model View Controller</td>
</tr>
<tr>
<td>NFP</td>
<td>Non-functional property</td>
</tr>
<tr>
<td>ORC</td>
<td>Open Reference Case</td>
</tr>
<tr>
<td>OVF</td>
<td>Open Virtualization Format</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>QPN</td>
<td>Queueing Petri Nets</td>
</tr>
<tr>
<td>PAC</td>
<td>Provisioning and Adjustment Component</td>
</tr>
<tr>
<td>POC</td>
<td>Planning and Optimization Component</td>
</tr>
<tr>
<td>POJO</td>
<td>Plain Old Java Objects</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a Service</td>
</tr>
<tr>
<td>SE</td>
<td>Service Evaluation</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SLAM</td>
<td>SLA Manager</td>
</tr>
<tr>
<td>SLAT</td>
<td>Service Level Agreement Template</td>
</tr>
<tr>
<td>SM</td>
<td>Service Manager</td>
</tr>
<tr>
<td>SME</td>
<td>Small and Medium-sized Enterprise</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SW-SLAM</td>
<td>Software SLA Manager</td>
</tr>
<tr>
<td>SW-SM</td>
<td>Software Service Manager</td>
</tr>
<tr>
<td>TCO</td>
<td>Total Cost of Ownership</td>
</tr>
<tr>
<td>TOGAF</td>
<td>The Open Group Architecture Framework</td>
</tr>
</tbody>
</table>
## Appendix C: Business Terms

The modeling task of Y2 builds on the work done during the first year, together with inputs from other work packages and an analysis of the bibliography. The following table reflects the changes in the model. The first column represents the name of the business term, while its explanation is shown in the second column. The third column (SLA@SOI model) specifies if the business term has been modelled as such (as described in section 4.1), is in any other part of the model, or has not been modeled. Finally, the last column (Y2) indicates the changes that have been made in the model with regards to that business term (typically, if it has been modeled again, or if it is new in the model).

<table>
<thead>
<tr>
<th>Types</th>
<th>Description</th>
<th>SLA@SOI Model</th>
<th>Y2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Availability dates</td>
<td>Ready for service date and contract duration.</td>
<td>Business Term</td>
<td>New modelling, as reliability</td>
</tr>
<tr>
<td>Design Information</td>
<td>Necessary when the Customer is also a SP who needs technical details in order to assess if the service offered is valid to build his own service. Information about redundancy level, routing alternatives, transport restrictions and others, can be included. In any case, if the Customer can modify some technical details, SLA shall include what type of changes he can make and under what conditions, as well as the procedure to access to the technical information he needs.</td>
<td>Service Description</td>
<td>New modelling</td>
</tr>
<tr>
<td>Change Procedures</td>
<td>Changes are required by one party and must be agreed by both parties. A change can affect the service provided to a customer, a group of costumers or all of them. Changes can be distributed within an upgrade or separately. SLA shall content the method to specify changes in service's requirements and the procedure to carry out them.</td>
<td>Not included</td>
<td>Not included</td>
</tr>
<tr>
<td><strong>Update Procedures</strong></td>
<td>Specification of the upgrade expectations (once a year, 6 months...), the distribution media (email, CD...), the procedures for releasing and installing patches, scheduled downtimes, and what to do if upgrade fails.</td>
<td>Business Term</td>
<td>New modelling, including scheduled downtime</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td><strong>Backup &amp; Recovery Mechanisms</strong></td>
<td>It may be advisable to include in the SLA a specification of the mechanisms the SP will use to insure service continuity during and following the occurrence of a disaster and the Customer support required to assist with disaster recovery. This kind of information could be more interesting for customers who are intermediate SPs and want to ensure a good availability level of the service to their end customers. It also includes the time to recover a useable service after a reliability failure.</td>
<td>Business Term</td>
<td>Included Time To Recover in Y2</td>
</tr>
<tr>
<td><strong>Priorities</strong></td>
<td>Sometimes, it is needed to specify the priorities of tasks or priorities in the execution of functionality of the service. In the other hand, when a problem happens, may be It's needed to explain how the support is applied, under which priorities or criterions.</td>
<td>Business Term</td>
<td>Included as part of Support Procedures</td>
</tr>
<tr>
<td><strong>Archiving</strong></td>
<td>Specification of how the historic information of the service and SLAs is stored: quantity, time and location.</td>
<td>Business Term</td>
<td>New in Y2 model</td>
</tr>
<tr>
<td><strong>Access Control</strong></td>
<td>Definitions of the roles and their permissions that entitle users to access the service functionality and support and to access sensitive information</td>
<td>To be included as part of the Service Description</td>
<td>New in Y2 model</td>
</tr>
<tr>
<td><strong>Contact Points</strong></td>
<td>It is useful to specify a single point of contact for technical aspects and a different one for administrative issues. Points of contact should be specified for both parties and include a complete description: names, phone and facsimile numbers, postal and email addresses, etc.</td>
<td>As Party Annotation</td>
<td>New modelling</td>
</tr>
<tr>
<td>Equipment</td>
<td>If necessary, SLA must include a description of the Service Provider equipment to be located on customer premises and its requirements of space, power and environment. As well as how SP staff can access it.</td>
<td>Not included in business layer or SLA Model</td>
<td>Not included</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Resources assigned</td>
<td>Description of the support capabilities including the scope of services offered, the periods when supporting service is accessible, the means of contact, etc.</td>
<td>Not included in business layer or SLA Model</td>
<td>Not included</td>
</tr>
<tr>
<td>Support process</td>
<td>Information about the support process is included. This can include the deadline for the first notification of an issue, the feedback frequencies, or the resolution time depending on the severity.</td>
<td>Business Term</td>
<td>New modelling</td>
</tr>
<tr>
<td>Arbitration and Mediation</td>
<td>Specification of the places where the legal disputes and judgments by arbitration should be arranged.</td>
<td>Business Term</td>
<td>New in Y2</td>
</tr>
</tbody>
</table>

### Monitoring and Reporting

<table>
<thead>
<tr>
<th>Monitoring</th>
<th>Description of parameters to be collected and measures to be made, data collection and measurement intervals, and aggregation intervals.</th>
<th>Business Term</th>
<th>New modelling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reporting</td>
<td>Specification of the content of performance reports, their format, the frequency of report production and delivery, the delivery mechanism and associated distribution list. Specification of ad hoc reporting support should be included.</td>
<td>Business Term</td>
<td>New modelling</td>
</tr>
</tbody>
</table>

### Changes in Reports

| Changes in Reports | SLA shall include how the customer can perform changes in the report content, format, frequency, means of delivery, etc, if proceed. A description of the notification procedures for changes made by SP should also be included. | Not modeled | Not included |

### Tariffs and Billing

<p>| Billing | This term details the structure, content and frequency of the bill that is sent to the customer. It includes information about the payment or the delivery information. It is related to the products, the penalties and the termination fees. | Business Term | New modelling |</p>
<table>
<thead>
<tr>
<th>Products and prices</th>
<th>Specification of the products offered with their price and the price modification (e.g. discounts). A product may be composed of several pricing components</th>
<th>Business Term</th>
<th>New modelling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalties</td>
<td>Description of the penalties in case the service doesn’t reach the level agreed: discounts, bonuses, ... Description of claiming procedures. Specification of reliability: definition of the frequency with which the service functions and facilities fail to be available as scheduled and the period between, which service availability failures occur, without committing penalty</td>
<td>Business Term</td>
<td>New modelling</td>
</tr>
<tr>
<td>Termination</td>
<td>Specification of the circumstances and causes under which a SLA might be terminated and the fees that should be applied, if any.</td>
<td>Business Term</td>
<td>New in Y2</td>
</tr>
<tr>
<td>Payments</td>
<td>In the event of termination of an Agreement for any cause, it is possible to specify any fees or expenses due for services delivered up to the effective date of termination.</td>
<td>Included in TerminationClause</td>
<td>New modelling</td>
</tr>
<tr>
<td>Termination for convenience clauses</td>
<td>Including reasons, notification method, periods, data transfer.</td>
<td>Included in TerminationClause</td>
<td>New modelling</td>
</tr>
<tr>
<td>Process for unilateral termination</td>
<td>Including notification method, periods, data transfer.</td>
<td>Included in TerminationClause</td>
<td>New modelling</td>
</tr>
<tr>
<td>Consumer Commitments</td>
<td>It defines the limits of usage of the customer: Maximum throughput (maximum usage of the service by the customer), peak period (period of time during which the customer uses the service most), critical period (period of time during which the service must be available), capacity estimates (maximum usage of service resources)</td>
<td>Business Term</td>
<td>New in Y2</td>
</tr>
</tbody>
</table>
Appendix D: API

Policies
These elements are related with way to control the difference values of the customer offers and the internal Business SLATs.

```java
package org.slasoi.businessManager.common.service

Class: PromotionManager

public abstract List<EmPolicies> getPolicies();
Description: This method retrieves all the policies.

description: This method retrieves a policy by its id.

description: This method retrieves the values of the parameters associated to a policy.

description: This method deletes a policy, given its id.

description: This method saves a policy into the database.

description: This method save the status of a policy into the database.

Promotions
The promotions are used to define the discounts that can be applied using different criteria (based on rules) to the products.

```
public abstract List<EmPromotions> getPromotions();
Description: This method retrieves all the promotions.

public abstract EmPromotions getPromotionById(Long id);
Description: This method retrieves a promotion given its id.

public abstract List<EmPromotionValues> getTemplateParams(Long promotionTypeId);
Description: This method retrieves the values of the parameters associated to a promotion.

public abstract List<EmPromotionValues> getPromotionValues(Long id);
Description: This method retrieves the values of the parameters associated to a promotion.

public abstract void savePromotion(EmPromotions promotion, Long nuProductId, Long nuPromotionTypeId, List<Long> paramIds, List<String> paramValues);
Description: This method saves a promotion into the database.

public abstract void deletePromotion(EmPromotions promotion);
Description: This method deletes a promotion into the database.

**Products**
These elements allow interacting with the system to create and find products.

package org.slasoi.businessManager.common.service
Class: ProductManager

public abstract EmSpProducts getProductById(Long id);
Description: This method retrieves a product by a given id.

public abstract List<EmSpProducts> findProducts(String filter);
Description: This method retrieves a list of products that match with the filter given.

public abstract List<EmSpProducts> findProductsActive(String filter);
Description: This method retrieves a list of the products in Active state.
public abstract void saveOrUpdate(EmSpProducts product);  
Description: This method saves a product into the database.

public abstract List<EmSpProducts> getAllInit();  
Description: This method retrieves all the products.

public abstract List<EmSpProducts> findProductForContract(String filter, Long nuCountryCustomer);  
Description: This method retrieves all the products associated to a customer.

public abstract void saveProduct(EmSpProducts product, ProductOffersAttr productOffers) throws Exception;  
Description: This method saves a product into the database.

public abstract void saveProductsStatus(List<Long> productsIds, List<String> productsStatus);  
Description: This method saves the state of a list of a product into the database.

**BSLAT Creation**
The creation of a BSLA template, stores it in the SLAT registry provided by A5 WP in the Generic SLAM.

package org.slasoi.businessManager.view.controller;  
Class: BusinessSlaController

public ModelAndView saveTemplate(HttpServletRequest request, HttpServletResponse response, PublishBslatForm form);  
Description: This method publishes a template into the register, creating it before using the data obtaining from the corresponding web page.

**SLA Management**
These operations store the start and the stop in the SLA.

package org.slasoi.businessManager.productSLA.management;  
Class: SLAManagement
public int startSLA(SLA bSLA,List<LowSLAInfo>lowSLAInfo);
Description: This method initiates an SLA inside SLAManager.

public int stopSLA(String bSLAID,String reason,String source);
Description: This method terminate an SLA inside the SLAManager.

Change the status of a product (product lifecycle)
package org.slasoi.businessManager.productSLA.management;
Class: ProductManagement

public int ckeckState(ProductState currentState, ProductState nextStage );
Description: This method checks if the transition between states of a product is valid.

public boolean cchangeProductState(long productId,ProductState nextStage);
Descriptio: This method updates the state of a product.

**Track Interface**
This interface provides some operations to receive information and interact with the SLAMs.

package org.slasoi.businessManager.common.api;
Class: TrackIf

public int trackEvent (List <AdjustmentNotificationType> violationList);
Description: Business Manager is informed about a violation Event.

public boolean getCustomerPurchaseAuth(long productId, long customerID)
throws CustomerNotFoundException,ProductNotFoundException;
Description: This method checks if a customer it is authorized to purchase a product.

**Control Interface**
This interface enables the Business Manager to communicate and send operations to the SLAMs.
package org.slasoi.businessManager.common.api;
Class: ControlIf

public int setPolicies(PolicyClassType policyClass,List<Policy> policies);
Description: This method sets the policies of a specific class.

```java
public List<Policy> getPolicies(PolicyClassType policyClass);
```

Description: This method retrieves the policies of a specific type.

```java
public List<Parameter> getMonitoringData(ParamMonitoringQueryType monitoringQuery);
```

Description: This method returns a list of the violations associated to the query given.

```java
public List<Parameter> getMonitoringData(String monitoringQuery);
```

Description: This method returns a list of the violations associated to the query given.

```java
public List<SLAViolation> getMonitoringData(SLAViolationQuery slaViolationQuery);
```

Description: This method returns a list of the violations associated to the query given.

**Penalty Management**

These operations provide the support for the penalties.

```java
package org.slasoi.businessManager.violationPenalty.utils;
Class: GuaranteeTermParser

static public Penalty getPenaltiesFromSLA(SLA sla, String guaranteeTerm);
Description: This method retrieves the penalty associated to a GuaranteeTerm given its identification.

static public boolean checkTermination(SLA sla, String guaranteeTerm);
Description: This method checks if a GuaranteeTerm has a termination actions associated given its identification.

**Negotiation policy management**

The operation allows the Business Manager to assess the offer.

```java
package org.slasoi.businessManager.businessEngine.assessment.impl;
Class: Assessment
```
public boolean assessOffer(long customerId, EmSpProductsOffer offer);

Description: This method decides if an offer is valid to be sent to a customer

**Business adjustment policy management**

The configuration of the Business adjustment is done by Drools file rule definition