Monitoring and Reputation Mechanisms for Service Level Agreements[⋆]

Omer Rana[1], Martijn Warnier[2], Thomas B. Quillinan[2], Frances Brazier[2]

[1]School of Computer Science/Welsh eScience Centre, Cardiff University, UK
[2]Department of Computer Science, VU University, Amsterdam, The Netherlands
`o.f.rana@cs.cardiff.ac.uk, warnier@cs.vu.nl,`
`tb.quillinan@few.vu.nl,frances@cs.vu.nl`

Abstract. A Service Level Agreement (SLA) is an electronic contract between a service user and a provider, and specifies the service to be provided, Quality of Service (QoS) properties that must be maintained by a provider during service provision (generally defined as a set of Service Level Objectives (SLOs)), and a set of penalty clauses specifying what happens when service providers fail to deliver the QoS agreed. Although significant work exists on how SLOs may be specified and monitored, not much work has focused on actually identifying how SLOs may be impacted by the choice of specific penalty clauses. A trusted mediator may be used to resolve conflicts between the parties involved. The objectives of this work are to: (i) identify classes of penalty clauses that can be associated with an SLA; (ii) define how to specify penalties in an extension of WS-Agreement; and (iii) specify to what extent penalty clauses can be enforced based on monitoring of an SLA.

1 Introduction

A Service Level Agreement (SLA) is an agreement between a client and a provider in the context of a particular service provision. SLAs may be between two parties, for instance, a single client and a single provider, or between multiple parties, for example, a single client and multiple providers. SLAs specify Quality of Service (QoS) properties that must be maintained by a provider during service provision – generally defined as a set of Service Level Objectives (SLOs). Often an SLA is only relevant when a client *directly* invokes a service (rather than through an intermediary – such as a broker). Such direct interaction also implies that the SLOs need to be measurable, and must be monitored during the provision of the service.

From an economics perspective, one may associate a cost with an SLA – which is the amount of money a client needs to pay the provider if the agreement has been adhered to (i.e. the requested quality has been met). The cost needs to be agreed between a client and a provider – and may be based on a posted

[⋆] This paper extends preliminary work reported at the Usage of Service Level Agreements in Grids Workshop [17] price (provider publishes), or negotiated through single/multi-round auctions (English, Dutch, Double, etc). How this price is set has been considered elsewhere [3], although the mechanism for doing this can also be determined through equilibrium pricing (based on supply-demand) or through auctions (based on client need). An SLA must also contain a set of penalty clauses specifying the implications of failing to deliver the pre-agreed quality. This penalty may also be defined as a cost – implying that the total revenue made by a provider would be the difference between the cost paid by the client and the discount (penalty) imposed on the provider. This type of analysis assumes that failure to meet an SLA is a non-binary decision – i.e. an SLA may be "partially" violated, and that some mechanism is in place to determine how this can be measured.

Although significant work exists on how SLOs may be specified and monitored [14], not much work has focused on actually identifying how SLOs may be impacted by the choice of specific penalty clauses. A trusted mediator may be necessary to resolve conflicts between involved parties. The outcome of conflict resolution depends on the situation: penalties, impact on potential future agreements between the parties and the mandatory re-running of the agreed service, are examples. While it may seem reasonable to penalize SLA non-compliance, there are a number of concerns when issuing such penalties. For example, determining whether the service provider is the only party that should be penalized, or determining the type of penalty that is applied to each party. Enforcement in the various legal systems of different countries can be tackled through stipulating a 'choice of law clause', that is a clause indicating expressly which countries' laws will be applied in case a conflict between the provider and the client would occur. Automating conflict resolution process could provide substantial benefits. Broadly speaking there are two main approaches for contractual penalties in SLAs: reputation based mechanisms [13, 18] and monetary fines. It is useful to note that often obligations within an SLA are primarily centered on the provider towards the client. An SLA is therefore an agreement between the provider to offer particular QoS to a client for some monetary return. We do not consider scenarios where there is also an obligation on the client towards the provider. An example of such a scenario could be where a provider requires the client to make input data available by a certain time frame to ensure that a particular execution time target is met. If the client is unable to meet the deadline for making such data available, the penalty incurred by the provider would no longer apply.

The use of reputation-based mechanisms to promote data integrity in distributed architectures has been explored by [9]. Knowing the reputation of a client can provide insight into what access may be granted to that client by a provider. Maintaining a measure of each client's reputation allows clients to make decisions regarding the best service provider for a specific task. In this case, reputation is a numerical value quantifying compliance to one or more SLAs. This value represents the previous behaviour of the provider in the system, and can be used by other clients to determine whether or not to interact with that provider. The higher this value, the more likelihood that the provider will act correctly in the future. Applying a numerical weight to users allows a more informed decision to be made when negotiating SLAs in the future. As users (clients and providers) interact with one another in the system, their reputation changes to reflect how they perform. For example, if a service provider consistently provides poor service (that is, violating its SLAs), its reputation will decline.

While reputation based mechanisms work relatively well in community based environments – where each participant monitors and judges other participants – in commercial environments reputation based mechanisms are rarely used. This can partly be attributed to the unbalanced nature of the relationship between clients and service providers. Monetary fines give a higher degree of expected QoS for service providers and (especially) clients. Monetary fines are also used in this paper. Such approaches are not new, other works in this area, such as [7, 8], provide only a partial solution to this problem. For example, they do not have a mechanism for conflict resolution.

The remainder of the paper is organized as follows: Section 2 starts with some background on Service Level Agreements, violations for SLAs and WS-Agreement. Section 4 discusses issues associated with penalties and Section 3 explains how monitoring of SLAs can be performed. The paper ends with a discussion.

2 Background

This section provides background on SLAs, violations for SLAs and WS-Agreement.

2.1 SLAs

An SLA can go through various stages once it has been specified. Assuming that the SLA is initiated by a client application, these stages include: discovering providers; defining the SLA; agreeing on the terms of the SLA; monitoring SLA violations; terminating an SLA; enforcing penalties for SLA violation.

The discovery of suitable providers phase involves choosing possible partners to interact with. This involves searching a known registry (or a distributed number of registries) for providers that match some profile – generally using predefined meta-data. The outcome of this stage is a single (or list of) providers that offer the capability a client needs.

Once a service provider(s) has been identified, the next stage involves defining the SLA between the client and the provider. The SLA may be between a single client and provider, or it may be between one client and multiple providers. In the subsequent analysis, we assume a two party SLA (i.e., one involving a single client and a single provider).

The definition of the SLA impacts the other stages in the SLA lifecycle – as the mechanisms used to identify particular Service Level Objectives (SLOs) will determine how violations will be identified in the future. Hence, an SLA may be defined using `(name, value)` pairs – where `name` refers to a particular SLO and `value` represents the requested quality/service level. An alternative is to use constraints that are more loosely defined – such as the use of `(name,` `relationship, value)` triples. In this context, provided the `relationship` between the `name` and `value` holds, the provider would have fulfilled the SLA requirements. Examples of relationships include `less_than`, `greater_than` or a user defined relationship function that needs to be executed by both the client and the provider.

Other schemes have included the use of server-side functions—an SLA being defined as a function $f(x_1, x_2, ..., x_n)$, where each (x_i) corresponds to a metric that is managed by the service provider. Using this approach, a client requests some capability from the service provider that is a function of what is available at the service provider. For instance, if the service provider has 512MB of available memory at a particular point in time, the client requests 50% of this. In this context, $f(x)$ is evaluated based on currently available capacity at the service provider [20]. An SLA must also be valid within some time period – a parameter that also needs to be agreed upon by the client and the provider.

Agreeing on SLA terms takes place once a description scheme has been identified. The next step is to identify the particular SLOs and their associated constraints. There needs to be some shared agreement on term semantics between the client and the provider. There is, however, no way to guarantee this, unless both the client and provider use a common namespace (or term ontology), and therefore rely on the semantic definitions provided within this namespace.

Agreeing on SLO terms may be a multi-shot process between the two parties. This process can therefore be expressed through a 'negotiation' protocol (a process requiring a provider to make an 'offer' to the client, and the client then making a 'counter offer'). The intention is to either reach convergence/agreement on SLOs – generally within some time bounds (or number of messages) – or indicate that the SLOs cannot be met. Also associated with an SLA must be the 'penalty' terms that specifies the compensation for the client if the SLA was not observed by the service provider. These penalty terms may also be negotiated between a client and a provider – or a fixed set of penalty terms may be used.

Monitoring SLA violation begins once an SLA has been defined. A copy of the SLA must be maintained by both the client and the provider. It is necessary to distinguish between an 'agreement date' (agreeing of an SLA) and an 'effective date' (subsequently providing a service based on the SLOs that have been agreed). A request to invoke a service based on the SLOs, for instance, may be undertaken at a time much later than when the SLOs were agreed.

As outlined in Section 3, during provision it is necessary to determine whether the terms agreed in the SLA have been compiled with during provision. In this context, the monitoring infrastructure is used to identify the difference between the agreed upon SLO and the value that was actually delivered during service provisioning – which is 'trusted' by both the client and the provider. It is also necessary to define what constitutes a 'violation'. Depending on the importance of the violated SLO and/or the consequences of the violation, the provider in breach may avoid dispatch or obtain a diminished monetary sanction from the client. In some instances, a client may be willing to avoid penalizing the provider if some of the SLOs are not fully adhered to compared to others. Any violations detected in the SLA may result in a monetary fine.

An SLA may be terminated in three situations: (i) when the service being defined in the SLA has completed; and (ii) when the time period over which the SLA has been agreed upon has expired; (iii) when the provider is no-longer available after an SLA has been agreed (for instance, the provider is under liquidation). In all three cases, it is necessary for the SLA to be removed from both the client and the provider. Where an SLA was actually used to provision a service, it is necessary to determine whether any violations had occurred during provisioning. As indicated above, penalty clauses are also part of the SLA, and need to be agreed between the client and the provider.

These stages demonstrate one cycle through the creation, use and deletion of an SLA.

2.2 Violations

One of the main issues that the provider and the consumer will have to agree during the SLA negotiation is the penalty scheme or the sanctionatory policy in use. Both the service provider and the client are free to decide what kinds of sanctions they will associate with the various types of SLA breaches, in accordance with the weight of the quality attribute that was not fulfilled. According to the Principles of European Contract Law [4], the term 'unfulfillment' is to be interpreted as comprising: (1) defective performance (parameter monitored at lower level) (2) late performance (service provided at the appropriate level but with unjustified delays) (3) no performance (service not provided at all). Based on these descriptions we define the following broad categories:

- 'All-or-nothing' provisioning: provisioning of a service meets all the SLOs – i.e., all of the SLO constraints must be satisfied for a successful delivery of a service;
- 'Partial' provisioning: provisioning of a service meets some of the SLOs – i.e., some of the SLO constraints must be satisfied for a successful delivery of a service;
- 'Weighted Partial' provisioning: provision of a service meets SLOs that have a weighting greater than a threshold (identified by the client).

Monitoring can be used to detect whether an SLA has been violated. Typically such violations result in a complete failure – making SLA violations an 'all-or-nothing' process. In such an event a completely new SLA needs to be negotiated, possibly with another service provider, which requires additional effort on both the client and the service provider. Based on this all-or-nothing approach, it is necessary for the provider to satisfy all of the SLOs. This equates to a conjunction of SLO terms. An SLA may contain several SLOs, where some SLOs (e.g. at least two CPUs) may be more important than others (e.g. more than 100 MB hard disk space). During the SLA negotiation phase, the importance of the different SLOs may be established. Clients (and service providers) can then react differently according to the importance of the violated SLO. In the WS-Agreement specification [1], the importance of particular terms is captured through the use of a 'Business Value'.

Weighted metrics can also be used to ensure a flexible and fair sanctionatory policy in case an SLA violation occurs. Thus, instead of terminating the SLA altogether it might be possible to re-negotiate, i.e., with the same service provider, the part of the SLA that is violated. Again, the more important the violated SLO, the more difficult (if not impossible) it will be to re-negotiate (part of) the SLA. The WS-Agreement specification supports the definition of a "Business Value" for particular SLOs (see section 4.2). These values reflect the relative importance placed on a particular term by a user, and may be used to support such a sanctioning policy.

2.3 WS-Agreement

WS-Agreement [1] provides a specification for defining SLAs, and comes from the Open Grid Forum (OGF). WS-Agreement is an XML document standard, that is, interactions between clients and providers are performed using an XML standardized format. There are two types of XML documents in WS-Agreement: *templates* and *agreements*. One basic element is that agreements need to be confirmed by both parties. Including penalties in a WS-Agreement, for example, cannot be one-sided. The WS-Agreements needs to be confirmed by the client. The existing WS-agreement specification, however, will need to be extended to include this step. Mobach *et. al.* [15] proposed this extension in the context of the WS-Agreement specification.

Figure 2.3 shows the extended interactions between a service provider (SP) and a consumer (C) described by [15]. The advertisement phase uses WS-Agreement template documents; the request and offer phase use WS-Agreement agreement documents. Templates describe the different services that the provider supports. When a negotiation takes place, the service provider sends these templates to the consumer. The consumer then makes an offer to the provider and, if acceptable, the agreement is created by the provider based on the offer. In figure 2.3 the initial template is generate by the provider, in accordance with the WS-Agreement specification.

1. SP $\rightarrow$ C : Advertisement
2. C $\rightarrow$ SP : Request
3. SP $\rightarrow$ C : Offer
4. C $\rightarrow$ SP : Acceptance/Rejection

Fig. 1. Negotiation using WS-Agreement

Templates and agreements both use the concept of negotiation *terms*. Terms define the service description and guarantees about the service. Guarantees are made relating to the service, such as the quality of service and/or the resource availability during service provision.

Agreements have a name defined by the provider and a context that contains meta-information about the agreement. This meta-information can include identifiers for the service provider and the agreement initiator; the name of the template that the agreement is based on; references to other agreements, and the duration of the agreement [15], as agreements have a fixed period when they are valid. Functional and non-functional requirements are specified in the *Terms* section. This is divided into the *Service Description Terms* (SDT) and *Guarantee Terms* (GT). A SDT holds the functional requirements for the delivery of services, and may refer to one or more components of functionality within one or more services. There may be any number of SDTs in a single agreement. GTs hold a list of services that the guarantee applies to, with the conditions that this guarantee applies, and any potential pre-conditions that must exist. Templates have a similar structure to agreements, with an additional *Creation Constraints* section. These constraints could include, for example, the maximum or minimum value for a service request. Creation constraints are an indication of the valid values for agreement requests. Creating an agreement that complies with these values does not guarantee the acceptance of the agreement by the service provider.

3 Monitoring

Monitoring plays an important role in determining whether an SLA has been violated, and thereby determine which penalty clause should be invoked as a consequence. From a legal point of view, monitoring appears as a pre-requisite for contract enforcement. In the present context, what needs to be put into effect are the consequences of breaching the agreed SLOs. In addition, service clients base their trust in service providers largely on the provided monitoring infrastructure. Traditionally, in the context of SLAs three monitoring modules can be distinguished [19, 14]: A trusted third party (TTP); a trusted module at the service provider; a model on the client site.

The trusted third party provides an independent module that can monitor (and log) all communication between clients and service providers. Both the service provider and client commit for each SLA. It is important for the TTP to be trusted by *both* the client and the provider. It is therefore necessary to establish the choice of a TTP before monitoring commences. It is also possible for the TTP to be defined in the SLA, requiring both the client and the provider to confirm this. After successfully completing the SLA both parties receive a signed ticket from the TTP that can be used for non-repudiation and reputation building of the service provider. Notice that a TTP cannot monitor the internal state of either client or service provider.

Using a trusted module at the service provider's site as an outside observer is functionally the same as using a TTP with the extension that trusted modules *may* also have the ability to observe the internal state of a service provider.

A module can monitor communication between client and service provider and can similarly provide (signed) tickets after successful completion of an SLA. Thus, the main difference between these approaches is that the trusted module is integrated into the service provider. This has as advantage that the internal state of the service provider can also be observed. The use of such a module provides weaker verification of SLOs than the use of an external TTP. There are two restrictions to using this approach:

– The service provider may not reveal it's internal state to the monitor module, and only allow a set of pre-defined variables to be monitored.
– The service provider may willingly or by error report incorrect information to the monitor module.

However, the service provider has the incentive to correctly report data to the monitoring module, to avoid incurring penalties for any SLO violations that are not caused by it. Consider the following example: if the SLO is "execution time", a network latency may result in an extra delay in the client's experience of this SLO. However, as the provider is not responsible for managing the network, the additional latency should not lead to a penalty for the provider. A co-located monitor at the provider would enable the provider to confirm that it was not at fault.

The third option – using a model on the client side – requires a client to determine if SLOs diverge from the predicted behavior, i.e., predicted by the model, of the service provider. Notice that in this case it is almost impossible to prove to third parties that the service provider is misbehaving. The applicability of this method is limited, it can possibly be used as a means for individual clients to establish their trust level in specific service providers, provided that a model that predicts the service provider's behavior can be successfully constructed.

Monitoring facilitates a direct and automatic SLA enforcement at run-time and without undue delay (that is, once a SLA violation is recorded, the agreed sanction can be automatically triggered), it also facilitates a more traditional enforcement. In either case, if the provider or the client contests the automatic sanction imposed, it can use monitoring data to argue its case. It is therefore vital to monitor all those metrics that have legal relevance and to give the parties the possibility to retrieve such data in a format that is admissible as evidence.

In situations that require a high level of assurance, the monitoring modules discussed above (especially the first two) can be combined. In the next section a monitoring architecture is provided that combines a trusted third party together with the use of a trusted module at the client side.

3.1 A Monitoring Architecture

In essence, monitoring only makes sense at the service provider's side. It is here that the resources specified in the SLAs (number of CPUs, disk space etc.) are actually hosted/provided. The problem is that a TTP can only monitor SLAs if the service provider allows this. Moreover, the TTP can never be sure if the monitoring information (provided by the service provider) is correct. For this a trusted module [16] at the service provider is mandatory.

The module should be able to monitor all resources that the service provider offers, for example, number of CPU's, type of CPU or upload limit of the network connection etc. For obvious reasons it is important that only a TTP that is explicitly trusted by the service provider can actually view this kind of information. This can be ensured by supplying 'trusted' TTP with a secret key that can be used for communicating with the module.

To ensure that a client can trust the TTP it is required that (1) the client can chose (during SLA negotiation) which TTP to use and (2) to ensure that there is enough choice for a client, each service provider needs to offer. Of course, this does not guarantee that the client will always find a TTP that it wants to use. Clients remain free to chose another service provider altogether, or ask a service provider to use an additional TTP the client does trust. Once a client and a service provider have created an SLA, the service itself is monitored by a TTP, using the trusted module at the service provider. Messages are exchanged between the Client (C), Service Provider (SP) and Trusted Third Party (TTP). Figure 2 gives a representation of the message exchange in the system.

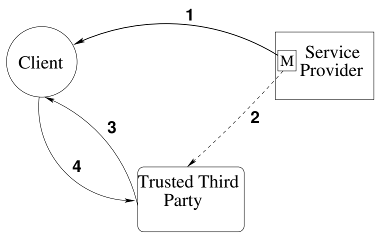

Fig. 2. Message Exchange

The messages are detailed in Figure 3. SLA$_1$ ... SLA$_n$ are SLAs, T_P is the timestamp of principal P. K_{TTP-M} is a shared key between the TTP and the trusted module M. The other messages are encrypted by the public key of the principal sending the message (K_P).

Whenever a service provider provides a service to a client (1), it is monitored by the module and sent to the TTP ($2_1 \dots 2_n$). The TTP stores a log of the information monitored for each SLA. Messages 3 and 4 are optional. They are only used if the client suspects a violation and requests the log (log$_k$ in Figure 3) from the TTP. The protocol ensures that only trusted parties gain access to the monitoring information.

The monitoring "interval" is also an important consideration when verifying violations of an agreement. Associated with this is the requirement for understanding whether *spot* or *aggregate* data should be considered for an SLO – or whether both need to be considered. For instance, short peaks in load (a common occurrence in many systems) may not always signify *real* exceptional situations

$$\begin{array}{llll}
1. & \text{SP} & \rightarrow \text{C} & : \text{Service} \\
2_1. & \text{SP} & \rightarrow \text{TTP} & : \{\text{SLA}_1, \log_1, T_{\text{M}_1}\}K_{\text{TTP-M}} \\
..... & & & \\
2_n. & \text{SP} & \rightarrow \text{TTP} & : \{\text{SLA}_n, \log_n, T_{\text{M}_n}\}K_{\text{TTP-M}} \\
[\ 3. & \text{C} & \rightarrow \text{TTP} & : \{\text{SLA}_k, \log_k, T_{\text{C}_k}\}K_{\text{TTP}} \quad] \\
[\ 4. & \text{TTP} & \rightarrow \text{C} & : \{\text{SLA}_k, \log_k, T_{\text{TTP}_k}\}K_C \quad]
\end{array}$$

Where $k, n \in \mathbb{N}$ and $k < n$.

Fig. 3. Message exchange between the Service Provide (SP) with trusted module (M), Client (C) and Trusted Third Party (TTP)

– and any adaptive behaviour at the service provider to these short peaks could lead to unstable behaviour [10]. For long running services (i.e. where the execution time of a service exceeds significantly the monitoring interval), it is therefore necessary to also determine what constitutes as a violation of an SLO.

4 Types of Penalties

Using penalty clauses in SLAs leads to two questions that need to be answered: what types of penalty clauses can be used; and how (if at all) can these be included in SLAs. The focus is on penalty clauses for service providers, since the 'burden of proof' and the interest in demonstrating that the agreed SLOs have been violated lies on the main beneficiary of the service, that is in the service client. One point should be kept in mind when designing 'penalty schemes'. Behind the imposition of any contractual sanctions lies the idea that faulty behavior of a provider should be deterred. As such, it is always possible for the service provider to contest its liability in the unwanted result (SLA breach) and claim that a 'force majeure' situation occurred. Although the situation is impossible to be dealt with through automatic enforcement, monitoring the message exchanges among the provider and the client can give an indication whether the SLA violation was the consequence of a 'misconduct' from the provider (either intentional or negligent). The parties are advised to stipulate either in the SLA or in the associated Collaboration Agreement how they choose to deal with the situation where the provider's faulty behavior cannot be documented, and a 'force majeure' situation did occur. Assuming only monetary fines are used, a penalty clause in an SLA may consist of the following:

- a decrease in the agreed payment for using the service, i.e., a direct financial sanction;
- a reduction in price along with additional compensation for subsequent interaction;

During the negotiation phase, client and provider can agree on a direct financial sanction. Usually, the amount to be paid depends on the value of the loss suffered by the client through the violation (which should cover entirely) and if agreed, a fixed sum of money that has to be paid as 'fine' for the unwanted behavior. Due to the difficulties in proving and documenting the financial value of the loss, during the SLA formation phase the parties may choose an 'agreed payment for non performance' that is a fixed sum of money that will have to be paid upon nonperformance, regardless of the fact that no financial loss was suffered by the client. During the formation phase client and provider can agree on a direct financial sanction (referred to as a 'fine' below) if an SLA is not (completely) fulfilled. The service provider can deposit the fine at a TTP, that acts as a mediator, before the service provision commences. On successful completion of the service provision (based on the SLA) the TTP returns the deposit to the service provider, otherwise the client receives the deposit as compensation for the SLA violation. Similarly, a fine can be combined with a discount for future services with the same provider. Notice that a trusted monitor is required for this, as a client can never prove by itself that an SLA was (partially) violated. However, for this to work properly –and especially automatically– some kind of micro payment [11] system is required – such as Paypal.

4.1 Negotiating Penalties

In Section 2.3, the messages that are exchanged within the system are described. Supporting penalties within this framework can be easily achieved using the *terms* section of WS-Agreement templates and agreements. This allows the use of the extended negotiation protocol defined by [15].

While negotiations can be managed in the existing framework, this does not adequately reflect the complexity of penalty negotiation. For example, if a mutually trusted third party cannot be agreed upon by both consumer and provider, there is little point in proceeding with the SLA negotiation. Similarly, if an SLA cannot be agreed upon, there is no need to negotiate the penalty clause. Therefore it is instead proposed to separate these three stages into distinct negotiation steps. Each of these steps follows the same steps as shown in Figure 2.3: Advertisement; Request; Offer, and Acceptance/Rejection. These steps can be considered negotiations for three separate services.

For example, negotiations to select a TTP proceeds as follows: In the *Creation Constraint* section of the WS-Agreement template, the TTPs trusted by the service provider are listed. When the consumer receives this template, they create an agreement offer specifying the TTP that they have selected. The offer is then processed by the provider. If it is acceptable, the provider produces the agreement document. This is passed to the consumer for acceptance/rejection. Negotiations for the SLAs and penalties are handled using the same process.

One concern with this approach is the verification that a SLA template refers to the TTP agreement previously negotiated and, similarly, the penalty template to the SLA and TTP agreements. This is achieved by the use of the references to the prior agreements within the *context* section of proceeding templates and agreements. Each penalty agreement then contains references to the TTP and SLA agreements. This ensures that a verifiable link is maintained throughout the service negotiation and provision.

Another approach to the multi-step process could be to specify the template and agreement documents as a single document, with separate services for each of the three stages. This would eliminate the need for three separate negotiations. However, this approach would make the templates more complicated. Furthermore, there is the possibility that service providers would develop standardized TTP templates and each of their SLA templates would refer to a standard TTP template.

4.2 Mapping to WS-Agreement

The WS-Agreement specification provides an XML schema [1] to represent the top-level structure of an agreement. This includes concepts such as an agreement identifier, guarantee terms in an agreement etc. Key to the discussion here is the use of a 'Business Value' (BV) and 'Preference' specification made available in WS-Agreement. A BV allows a provider to assess the importance of a given SLO to a client. Similarly, a provider may indicate to a client the confidence that a provider has in meeting a particular SLO. Based on the specification, a BV may be expressed using a `penalty` or `reward` type. The penalty is used to indicate the likely compensation that will be required of a provider if the SLO with which the penalty is associated is not met. A BV list is specified as:

```
<wsag:BusinessValueList>
<wsag:Importance>xs:integer</wsag:Importance>
<wsag:Penalty> </wsag:Penalty>
<wsag:Reward> </wsag:Reward>
<wsag:Preference> </wsag:Preference>
<wsag:CustomBusinessValue>
  </wsag:CustomBusinessValue>
</wsag:BusinessValueList>
```

Notice that a BV list consists of both a penalty <u>and</u> a reward – to enable a provider to assess the risk/benefit of violating a particular SLO. `Preference` is used in the BV list is used to provide a more detailed sub-division of a business values for different alternatives that may exist. Essentially, `Preference` allows a service provider to consider different possible alternatives for reaching the same overall SLO requirement. For instance, if a client requests access to a particular number of CPUs, it is possible to fulfill this requirement based on CPUs from one or more resource owners. `Preference` allows the provider to chose between the available options to improve its own revenue or meet other constraints that it has (provided this is not prohibited by the service provision agreement or other agreements between the parties involved).

A `Penalty` in WS-Agreement may be associated with one or more SLOs, and occurs when these SLO(s) are violated. According to the WS-Agreement specification, assessment of a violation needs to be monitored over an `AssessmentInterval` – which is defined either as a time interval or some integer count. Essentially, this means that a penalty can only be imposed if an SLO is violated within a particular time window, or if a certain number of service requests/accesses fail. `ValueUnit` identify the type of penalty – in this case a monetary value – that must be incurred by the service provider if the penalty occurs. In the current WS-Agreement specification, the concept of a `ValueExpr` is vague – as being either an integer, float or a 'user defined expression'. This implies that a user and provider may determine a dynamic formula that dictates the penalty amount depending on the particular context in which the WS-Agreement is being used. In WS-Agreement the ability to also specify a `Reward`, in addition to a penalty provides an incentive mechanism for a provider to meet the SLO.

```
<wsag:Penalty>
   <wsag:AssesmentInterval>
       <wsag:TimeInterval>xs:duration
        </wsag:TimeInterval> |
       <wsag:Count>xs:positiveInteger</wsag:Count>
   </wsag:AssesmentInterval>
   <wsag:ValueUnit>xs:string</wsag:ValueUnit>
   <wsag:ValueExpr>xs:any</wsag:ValueExpr>
</wsag:Penalty>

<wsag:AssesmentInterval>
    <wsag:Count>4</wsag:Count>
</wsag:AssesmentInterval>
<wsag:ValueUnit>US Dollar</wsag:ValueUnit>
<wsag:ValueExpr>500</wsag:ValueExpr>
```

Although useful, the description of penalty and rewards in WS-Agreement is still very simple and cannot account for the varying types of penalties that can be defined in real agreements [17]. For instance, it is not currently possible to define variations in penalties at different quality levels. In addition, the extent to which terms and conditions specified in WS-Agreements are legally binding is currently subject of research [5].

5 Discussion and Conclusions

The use of penalties in SLAs has obvious benefits for both clients and service providers. Monetary sanctions (and optionally reputation based mechanisms) can be used as, pre-agreed, penalties. It has been shown how the WS-Agreement specification can be used to specify penalties and rewards, in the context of a particular resource sharing scenario. Both of these approaches require the participation of a trusted mediator and monitoring module in the form of a trusted third party. We identify the types of monitoring infrastructure that can be used to validate SLOs during service provisioning. As monetary sanctions are the *de facto* standard in industry for penalty clauses, these are preferred over reputation based solutions, though the latter can be used if so required.

A particular focus has been discussion of the types of violations that can occur in SLOs during provisioning. Based on European legal contract law, we identify three types of violations that may lead to penalties – an 'all or nothing', 'a partial' or a 'weighted partial' violation of a contract. An observation in this work is that flagging a violations incurs a cost for the client (as well as the provider). It is therefore in the interest of the client to continue with service provision, even if some of the SLOs are not being observed fully – a trade off discussed in this paper. A key contribution of this work is a model that demonstrates how a client may provide weighting to certain SLOs over others, the legal basis on which this model is based and subsequently how this approach can be used alongside WS-Agreement.

Acknowledgments: This research has arisen as a result of collaboration between the EU "SORMA" project (IST 034286), support provided by the NLnet Foundation (`http://www.nlnet.nl`) and in part by the NWO TOKEN programme. Part of this work was also supported by the EU IST-2006-027004 "S3MS" project. We are grateful to Dana Cojocarasu (Research Center for Computers and Law, University of Oslo, Norway) for contribution and discussion regarding European law for electronic contracts.

References

1. A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. Web Services Agreement Specification (WS-Agreement). *GRAAP Working Group at the Open Grid Forum*, September 2006.
2. ARAD Automatic Real-time Decision-making, 2002. available at:`http://www.haifa.il.ibm.com/projects/software/arad/papers/ARAD-May-2002.pdf`.
3. M. Becker, N. Borrisov, V. Deora, O. F. Rana, D. Neumann, "Using k-Pricing for Penalty Calculation in Grid Market", *Proceedings of IEEE HICSS 2008 Conference, Hawaii, January 2008.*
4. M. J. Bonell. The UNIDROIT Principles of International Commercial Contracts and the Principles of European Contract Law: Similar Rules for the Same Purposes? , 1996.
5. M. Boonk, F. Brazier, D. de Groot, M. van Stekelenburg, A. Oskamp, and M. Warnier. Conditions for Access and Use of Legal Document Retrieval Web Services. In *Prooceedings of the Eleventh International Conference on Artificial Intelligence and Law (ICAIL'07)*. ACM Press, 2007.
6. Y. Chen, S. Iyer, X. Liu, D. Milojicic, and A. Sahai. SLA Decomposition: Translating Service Level Objectives to System Level Thresholds. *HPL-2007-17*, 2007.
7. B. C. Clayton, T. B. Quillinan, and S. N. Foley. Automating security configuration for the grid. *Journal of Scientific Programming*, 13(2):113–125, 2005.
8. S. N. Foley. Using trust management to support transferable hash-based micropayments. In *Proceedings of the 7th International Financial Cryptography Conference*, Gosier, Guadeloupe, FWI, January 2003.
9. A. Gilbert, A. Abraham, and M. Paprzycki. A System for Ensuring Data Integrity in Grid Environments. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, pages 435–439, Las Vegas, Nevada, USA, April 5–7 2004.

10. D. Gmach, S. Krompass, A. Scholz, M. Wimmer, and A. Kemper, “Adaptive Quality of Service Management for Enterprise Services”, ACM Transactions on the Web, Vol. 2, No. 1, February 2008.
11. R. Hauser, M. Steiner, and M. Waidner. *Micro-payments Based on IKP*. IBM TJ Watson Research Center, 1996.
12. L. Joita, O. F. Rana, P. Chacin, I. Chao, F. Freitag, L. Navarro, and O. Ardaiz. Application Deployment on Catallactic Grid Middleware. *IEEE DS-Online*, 7(12), 2006.
13. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proc. of the 12th Int. World Wide Web Conference*, Budapest, Hungary, May 20-24 2003. ACM Press.
14. A. Keller and H. Ludwig. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *Journal of Network and Systems Management*, 11(1):57–81, 2003.
15. D. G. A. Mobach, B. J. Overeinder, and F. M. T. Brazier. A WS-Agreement Based Resource Negotiation Framework for Mobile Agents. *Scalable Computing: Practice and Experience*, 7(1):23–36, 2006.
16. S. Pearson and B. Balacheff. *Trusted computing platforms: TCPA Technology in Context*. Prentice Hall PTR, 2002.
17. O. Rana, M. Warnier, T. B. Quillinan, F. M. T. Brazier, and D. Cojocarasu, Managing Violations in Service Level Agreements, In *the Proceedings of the Usage of Service Level Agreements in Grids Workshop*, at IEEE/ACM Grid Conference, Austin, Texas, September 2007.
18. J. Sabater and C. Sierra. Social regret, a reputation model based on social relations. *SIGecom Exch.*, 3(1):44–56, 2002.
19. A. Sahai, V. Machiraju, M. Sayal, A. van Moorsel, and F. Casati. Automated SLA Monitoring for Web Services. In *Management Technologies for E-commerce and E-business Applications: 13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, (DSOM 2002)*. Springer, 2002.
20. R. S. V. Yarmolenko. An Evaluation of Heuristics for SLA Based Parallel Job Scheduling. *3rd High Performance Grid Computing Workshop (in conjunction with IPDPS 2006), Rhodes, Greece*, 2006.
21. E. Wustenhoff. Service Level Agreement in the Data Center. *Sun Microsystems Professional Series*, April 2002.