

# Renegotiation of Electronic Brokerage Contracts

Rúben Cunha<sup>1</sup>, Bruno Veloso<sup>2</sup>, and Benedita Malheiro<sup>1,2</sup>

<sup>1</sup> ISEP/IPP – School of Engineering, Polytechnic Institute of Porto, Portugal

<sup>2</sup> INESC TEC, Porto, Portugal

1100478@isep.ipp.pt, bruno.miguel.veloso@gmail.com, mbm@isep.ipp.pt

**Abstract.** CloudAnchor is a multiagent e-commerce platform which offers brokerage and resource trading services to Infrastructure as a Service (IaaS) providers and consumers. The access to these services requires the prior negotiation of Service Level Agreements (SLA) between the parties. In particular, the brokerage SLA (bSLA), which is mandatory for a business to have access to the platform, specifies the brokerage fee the business will pay every time it successfully trades a resource within the platform. However, while the negotiation of the resource SLA (rSLA) includes the uptime of the service, the brokerage SLA was negotiated for an unspecified time span. Since the commercial relationship – defined through the bSLA – between a business and the platform can be long lasting, it is essential for businesses to be able to renegotiate the bSLA terms, *i.e.*, renegotiate the brokerage fee. To address this issue, we designed a bSLA renegotiation mechanism, which takes into account the duration of the bSLA as well as the past behaviour (trust) and success (transactions) of the business in the CloudAnchor platform. The results show that the implemented bSLA renegotiation mechanism privileges, first, the most reliable businesses, and, then, those with higher volume of transactions, ensuring that the most reliable businesses get the best brokerage fees and resource prices. The proposed renegotiation mechanism promotes the fulfilment of SLA by all parties and increases the satisfaction of the trustworthy businesses in the CloudAnchor platform.

**Keywords:** Brokerage, Service Level Agreements, Negotiation, Trust

## 1 Introduction

CloudAnchor is a multiagent e-commerce platform which offers brokerage and resource trading services to Infrastructure as a Service (IaaS) providers and consumers [9]. The access to these services implies the automated negotiation of contracts – Service Level Agreements (SLA) – between the parties, specifically, brokerage SLA (bSLA) and resource SLA (rSLA) [7]. SLA negotiation relies on a decentralised distributed trust model built from historical interaction data between parties, supporting the implemented Trust-based partner Invitation/acceptance & Negotiation (TIN) services [8].

First, any provider or consumer business must establish a bSLA with the platform, which specifies the brokerage fee the business will pay every time it

successfully trades a resource. The rSLA is celebrated between consumer and provider and specifies the terms of the resource provision, *i.e.*, uptime, price, *etc.*

While rSLA are typically time bound – resources are leased for well-defined periods – this is not the case of bSLA. Since the relationship between a business and the platform can be long lasting, it is essential for businesses to be able to renegotiate their bSLA terms, *i.e.*, renegotiate the brokerage fee.

To address this issue, we designed a bSLA renegotiation mechanism, which takes into account the duration of the bSLA as well as the past behaviour (trust) and success (transactions) of the business in the CloudAnchor platform. Several tests were performed involving multiple consumers and providers with different trustworthinesses. The results show that the implemented bSLA renegotiation mechanism privileges, first, the most reliable businesses, and, then, those with higher volume of transactions, ensuring that the most reliable businesses get the lowest brokerage fees and resource prices. The renegotiation mechanism is fair for both businesses and platform, promotes the fulfilment of SLA and increases the satisfaction of the trustworthy businesses in the CloudAnchor platform.

This paper is organised in five sections. Section 2 provides a survey on related work; Section 3 presents the designed bSLA renegotiation mechanism; Section 4 describes the experiments and results; and, finally, Section 5 draws the conclusions and refers future developments.

## 2 SLA Renegotiation

The renegotiation of SLA based on past partner behaviour, *i.e.*, taking into account the trading history between partners, has been addressed by other authors like [1,2,3,5,6].

Di Modica *et al.* [1,2] address the renegotiation of Service-Level-Objective (SLO) at run-time. They propose a WS-Agreement Negotiation protocol extension together with a finer detailed description of the SLO Guarantee Terms, including the specification of the scope of the changes which can be performed at run time. Similarly, Sharaf and Djemane (2015) [6] present another extension of the WS-Agreement specification to support SLA renegotiation. Consumers and providers must define the renegotiable terms or SLO of the SLA that could change during the SLA lifetime, resulting in new SLA negotiation states. Our approach allows the renegotiation of active brokerage agreements based on the trust and success of the business in the platform, *i.e.*, free from any pre-defined change boundaries.

Frankova *et al.* [3] propose an extension to the Open Grid Forum (OGF) WS-Agreement protocol to support run-time renegotiations of SLA. The authors propose the use of semantics to generate warnings related with agreement violations, which can be used to trigger the renegotiation of the Quality of Service (QoS) terms of active agreements. Our negotiation protocol, which is also an extension of the WS-Agreement, can be triggered automatically or on demand by any business registered in the platform.

Hani *et al.* [4] present a review on SLA management systems for cloud-based systems. They identify the following management stages: service negotiation, resource provisioning & allocation, service monitoring & assessment and service termination & penalty. Our platform provides brokerage and negotiation services, including the renegotiation of bSLA, allowing businesses to renegotiate their brokerage fee according to their trading behaviour reactively, *i.e.*, wait till the end of the current agreement to renegotiate, or pro-actively, decide to renegotiate the current agreement at any other time.

Parkin *et al.* [5] propose a re-negotiation protocol which is a WS-Agreement specification extension for the renegotiation of resource agreements between providers and consumers. This renegotiation protocol has a fault tolerance mechanism to deal with lost and duplicated messages. Our trust and success based SLA renegotiation proposal, although it has been applied to brokerage agreements, is a general mechanism applicable to any type of SLA renegotiation.

### 3 Renegotiation of Brokerage Contracts

In order to have access to the platform services a business must, first, negotiate and establish a bSLA with the platform, specifying the value of the brokerage fee to be paid to the platform every time a resource is successfully traded. This brokerage agreement remained unchanged regardless of the success and behaviour of the business within the platform. The newly designed (re)negotiation mechanism, which is based on the success and behaviour of the business so far, allows businesses to (re)negotiate the terms of the brokerage agreement, namely, the duration and brokerage fee.

The SLA agent of each business (provider or consumer), which is the renegotiation initiator, calculates and proposes the new brokerage fee to the platform SLA agent. The proposed brokerage fee depends on: (*i*) the duration of bSLA contract; (*ii*) the number of successfully negotiated resources; (*iii*) the negotiation success rate; and (*iv*) the business trustworthiness. Equation 1 determines the partial discount based on the bSLA duration where  $id$  is the business identifier and  $\Delta t$  is the duration of the contract in days. This exponential function is adjusted so that the discount ranges from 0.0% to 25.0% for periods from less than one day to one year.

$$Disc(id, t) = \frac{1}{0.277 \log_{10}(10 + \sqrt{\Delta t})} + 0.134 \quad (1)$$

Equation 2 calculates the partial discount based on the number of resources obtained or provided, where  $id$  is the business identifier and  $\#rSLA_N$  is the total number of resources negotiated by the business.

$$Disc(id, \#rSLA) = \begin{cases} 0.00, & 0 < \#rSLA_N \leq 100 \\ 0.20, & 100 < \#rSLA_N \leq 500 \\ 0.25, & 500 < \#rSLA_N \leq 1000 \\ 0.27, & 1000 < \#rSLA_N \leq 2500 \\ 0.30, & \#rSLA_N \geq 2500 \end{cases} \quad (2)$$

Each business builds partner and self trust models regarding the partner invitation/acceptance ( $I$ ), rSLA negotiation ( $N$ ) and rSLA enforcement ( $E$ ) stages [8]. Equation 3 represents the general self trust formula applied to each interaction stage  $S$ , where  $id$  is the business identifier,  $n$  is the number of interactions of type  $S$  of the business in the platform and  $Out_{S,i}$  is the boolean outcome (success or failure) of the interaction  $i$  of type  $S$ .

$$T_S(id) = \frac{\sum_{i=1}^n Out_{S,i}}{n} \quad (3)$$

The success rate of a business is provided by Equation 4 where  $id$  represents the business identifier,  $T_E(id)$  the business enforcement self trust given by Equation 3,  $s$  the number of established rSLA,  $\Delta t_i$  the duration each established rSLA,  $r$  the number of resources offered/required by the business and  $\Delta t_j$  the time span of the offer/request of each resource.

$$Suc(id) = T_E(id) \frac{\sum_{i=1}^s \Delta t_i}{\sum_{j=1}^r \Delta t_j} \quad (4)$$

Equation 5 represents the credibility of a business where  $id$  is the business identifier,  $T_E(id)$  is the enforcement self trust and  $Suc(id)$  is the success rate of the business in the platform.

$$Cred(id) = \frac{T_E(id) + Suc(id)}{2} \quad (5)$$

Equation 6 calculates the brokerage fee discount where  $id$  is the business identifier,  $\#rSLA$  is the total number of established rSLA,  $Cred(id)$  is the business credibility in the platform,  $Disc(id, t)$  is the partial discount based on the duration of the bSLA and  $Disc(id, \#rSLA)$  is the partial discount based on the number of established rSLA.

$$Disc(id, t, \#rSLA) = Cred(id) \left( \frac{Disc(id, t) + Disc(id, \#rSLA)}{2} \right) \quad (6)$$

Finally, Equation 7 calculates the brokerage fee to be proposed to the platform, where  $bFee_{default}$  is the default brokerage fee (2.0%) and  $Disc(id, t, \#rSLA)$  is the fee discount provided by Equation 6.

$$bFee(id, t, \#rSLA) = bFee_{default}(1 - Disc(id, t, \#rSLA)) \quad (7)$$

The business proposes the brokerage fee given Equation 7 to the platform SLA agent, which, in turn, applies the conditions presented in Table 1 to decide whether to make a counter offer or to accept it.

## 4 Tests and Results

We performed two sets of experiments in a single resource request/provision scenario. The scenario encompasses five groups of ten consumers and five groups

**Table 1.** Platform conditions for the assessment of brokerage fee proposals.

Providers				Consumers			
Cond.	Trust	# Resources	<i>bFee</i> (%)	Cond.	Trust	# Resources	<i>bFee</i> (%)
1	1.00	0	2.00	1	1.00	0	2.00
2	[0.35; 1.00[	0	3.00	2	[0.35; 1.00[	0	3.00
3	1.00	[1; 1000[	[1.70; 2.0]	3	1.00	< 1000	[1.70; 2.0]
4	1.00	[1000; 3000]	[1.20; 1.40]	4	1.00	[1000; 3000]	[1.20; 1.40]
5	1.00	> 3000	1.10	5	1.00	> 3000	1.10
6	[0.85; 1.00[	[1000; 3000]	[1.30; 1.50]	6	[0.85; 1.00[	< 1000	[1.50; 2.00]
7	[0.85; 1.00[	> 3000	1.15	7	[0.85; 1.00[	[1000; 3000]	[1.30; 1.50]
8	[0.70; 0.85[	[1000; 3000]	[1.50; 1.70]	8	[0.85; 1.00[	> 3000	1.15
9	[0.70; 0.85[	> 3000	1.35	9	[0.70; 0.85[	< 1000	[1.70; 2.00]
10	[0.50; 0.70[	[1000; 3000]	[1.70; 2.00]	10	[0.70; 0.85[	[1000; 3000]	[1.50; 1.70]
11	[0.50; 0.70[	> 3000	1.45	11	[0.70; 0.85[	> 3000	1.35
12	[0.40; 1.00[	< 500	[2.00; 3.00]	12	[0.50; 0.70[	< 1000	[2.00; 2.50]
13	[0.40; 1.00[	[500; 1000[	[1.50; 2.50]	13	[0.50; 0.70[	[1000; 3000]	[1.70; 2.00]
14	[0.35; 0.50[	[1000; 3000]	[2.50; 3.00]	14	[0.50; 0.70[	> 3000	1.45
15	[0.35; 0.50[	> 3000	2.10	15	[0.35; 0.50[	< 1000	[2.50; 3.00]
16	[0; 0.35[	—	3.50	16	[0.35; 0.50[	[1000; 3000]	[2.50; 3.00]
				17	[0.35; 0.50[	> 3000	2.10
				18	[0; 0.35[	—	3.50

of ten providers, each group with different trustworthiness. Each provider holds 1000 standard virtual machines (VM) and each consumer makes 1000 single VM requests, using the Trust-based Invitation/acceptance & Negotiation (TIN) services. If all businesses were 100 % trustworthy, the offer would meet the demand. However, since the average trustworthiness of the providers or consumers is 60 %, it corresponds in fact to an under supply market scenario with an anticipated average failure rate of 40 %. Table 2 presents the groups used in both experiments.

**Table 2.** Groups of consumers and providers used in the experiments

ID	Trust (%)	#VM/Consumer	ID	Trust (%)	#VM/Provider
GC_020	20	1000	GP_020	20	1000
GC_040	40	1000	GP_040	40	1000
GC_060	60	1000	GP_060	60	1000
GC_080	80	1000	GP_080	80	1000
GC_100	100	1000	GP_100	100	1000

At start up, the brokerage fee (*bFee*) is 2.0 % for all businesses (consumers or providers), corresponding to the default brokerage fee applied to any newly

registered business. Without renegotiation, the brokerage fee ( $bFee$ ) remains unchanged regardless of the success and behaviour of the businesses in the platform. The tests include five complete bSLA negotiation cycles. In the first test, the rSLA are negotiated for the duration of one cycle (short term), while, in the second test, rSLA are negotiated till the end of the five cycles (long term).

#### 4.1 Short Term Resource Agreements

Table 3 displays the results of the first test, including both bSLA and rSLA related data. In this experiment the duration of the rSLA corresponds to a single cycle. As a result, when a new cycle starts, the providers regain their 1000 VM and the consumers request again 1000 VM sequentially.

**Table 3.** Platform results with bSLA renegotiation – single cycle rSLA

Trust (%)	C	Consumers								Providers							
		rSLA				$bFee$				rSLA				$bFee$			
		T (#)	TF (#)	SFR (%)	AE (€)	AC (€)	AL (€)	Bus (%)	Plat (%)	T (#)	TF (#)	SFR (%)	AR (€)	AC (€)	AL (€)	Bus (%)	Pla (%)
20	1	80	70	80	34.0	0.7	34.0	<b>2.0</b>	<b>2.0</b>	58	51	79	32.7	0.7	33.4	<b>2.0</b>	<b>2.0</b>
	2	9	8	78	36.3	1.2	36.2	2.0	<b>3.5</b>								
	3	4	3	75	36.5	1.2	36.3	2.0	<b>3.5</b>								
	4	3	3	100	36.5	1.2	36.2	2.0	<b>3.5</b>								
	5	3	2	67	36.5	1.2	36.6	2.0	<b>3.5</b>								
40	1	98	71	59	34.0	0.7	34.0	<b>2.0</b>	<b>2.0</b>	74	57	59	33.0	0.7	33.6	<b>2.0</b>	<b>2.0</b>
	2	16	11	63	35.8	1.1	35.7	1.9	<b>3.2</b>	1	1	100	32.8	0.6	32.9	<b>2.0</b>	<b>2.0</b>
	3	16	11	56	35.7	1.0	35.7	1.9	<b>3.0</b>								
	4	16	11	63	35.9	1.1	35.9	1.9	<b>3.0</b>								
	5	14	10	57	35.9	1.1	35.9	1.9	<b>3.0</b>								
60	1	99	58	39	33.7	0.7	33.7	<b>2.0</b>	<b>2.0</b>	124	76	40	33.1	0.7	33.7	<b>2.0</b>	<b>2.0</b>
	2	27	11	41	35.1	1.0	35.1	1.9	<b>2.8</b>	7	5	43	34.4	0.7	35.1	<b>2.0</b>	<b>2.0</b>
	3	30	14	40	35.0	0.9	35.1	1.9	<b>2.7</b>	1	1	0	35.8	1.1	36.9	<b>1.9</b>	<b>2.9</b>
	4	25	11	40	34.8	0.9	34.8	1.9	<b>2.7</b>								
	5	38	16	39	35.0	0.9	35.0	1.9	<b>2.6</b>								
80	1	182	70	20	33.6	0.7	33.5	<b>2.0</b>	<b>2.0</b>	383	116	20	32.2	0.6	32.9	<b>2.0</b>	<b>2.0</b>
	2	68	19	21	33.9	0.9	33.9	1.8	<b>2.6</b>	295	72	20	31.9	0.6	32.5	<b>1.9</b>	<b>1.9</b>
	3	39	10	21	33.8	0.8	33.8	1.8	<b>2.5</b>	292	70	20	31.8	0.6	32.4	<b>1.8</b>	<b>1.8</b>
	4	50	13	20	33.9	0.8	33.9	1.8	<b>2.4</b>	295	71	20	31.8	0.5	32.4	<b>1.7</b>	<b>1.8</b>
	5	40	10	20	33.7	0.8	33.6	1.8	<b>2.3</b>	287	69	20	31.8	0.5	32.3	<b>1.7</b>	<b>1.7</b>
100	1	1000	79	0	33.1	0.6	0.0	<b>2.0</b>	<b>2.0</b>	821	48	0	32.7	0.7	0.0	<b>2.0</b>	<b>2.0</b>
	2	1000	50	0	32.9	0.5	0.0	<b>1.7</b>	<b>1.7</b>	817	21	0	32.6	0.6	0.0	<b>1.7</b>	<b>1.7</b>
	3	1000	51	0	32.8	0.5	0.0	<b>1.5</b>	<b>1.5</b>	796	17	0	32.5	0.5	0.0	<b>1.5</b>	<b>1.5</b>
	4	1000	51	0	32.6	0.4	0.0	<b>1.2</b>	<b>1.2</b>	798	18	0	32.5	0.4	0.0	<b>1.3</b>	<b>1.3</b>
	5	1000	49	0	32.5	0.5	0.0	1.0	<b>1.2</b>	807	19	0	32.4	0.4	0.0	<b>1.2</b>	<b>1.2</b>

C – Cycle; T – Total number of rSLA; TF – Total number of Failed rSLA; SFR – rSLA Self Failure Rate; AE – Average resource Expenditure; AC – Average resource Cost; AL – Average resource Loss; AR – Average resource Revenue; Bus –  $bFee$  Business proposal; Plat –  $bFee$  Platform counter-proposal.

In terms of rSLA, Table 3 displays the average number of established rSLA, the number of failed rSLA, the rSLA self failure rate (the percentage of self violated rSLA) per business, the average consumer resource expenditure (AE) or

average provider revenue (AR), the average brokerage cost (AC) of the successful rSLA and the average resource loss (the penalties for the self violated rSLA) per resource and business. In the case of established rSLA, when a business (consumer or provider) fails to fulfil the terms of an established rSLA, it must reimburse its partner of the resource price and *bFee* values. In terms of the bSLA, the table lists the brokerage fee proposed by the business and the platform and highlights the resulting *bFee*.

In the five cycles there were, in average, 5142 successfully traded resources out of 5853 negotiated rSLA per business (failure rate of 12.1%). This means that, in average, the five groups of consumers and providers were able to successfully trade 20.6% of the 25 000 resources required/offered per business in the five cycles. Only GC\_100, GP\_100, GC\_080 and GP\_080 traded a significant number of resources. As expected, the most trusted group of providers (GP\_100) leased the majority of its resources at a medium average price (it is providing quantity-based discounts to trusted consumers), while the most trusted group of consumers (GC\_100) obtained 95.8% of its resources at a low average price.

Table 4 summarises the impact of the bSLA renegotiation in the case of short term resource agreements. The overall losses, *i.e.*, the amount paid in penalties by all businesses, has a negligible increase of 0.1% and the overall costs, *i.e.*, the amount paid in brokerage fees by all businesses, decrease 21.1%. The total revenues of the providers as well as the total expenditure of the consumers remains unchanged since the bSLA renegotiation does not have a direct influence on the negotiation of rSLA. The impact of the bSLA renegotiation in terms of average amount spent by consumers per resource, including costs and losses, is -0.4% and in the average amount received by providers per resource, deducing the costs and losses, is +0.5%, *i.e.*, it is advantageous for both parties.

**Table 4.** Impact of the bSLA Renegotiation

$\Delta$	Consumers	Providers	Overall
	(%)	(%)	(%)
<b>Total Losses</b>	+0.3	-0.1	+0.1
<b>Total Costs</b>	-22.1	-20.0	-21.1

These results clearly show the advantages of the *bFee* renegotiation in terms of the average price/revenue, costs and losses per resource successfully traded.

## 4.2 Long Term Resource Agreements

In the second test, each consumer requests just 1000 VM, instead of 1000 VM per cycle, and the rSLA are established till the end of the five negotiation cycles, *i.e.*, providers do not regain their resources at the end of each negotiation cycle. Once a business has established 1000 rSLA, it stops trading. In such a scenario, the most trusted businesses will stop trading earlier, leaving the less trusted to continue trading. Table 5 presents the results.

**Table 5.** Platform results with bSLA renegotiation – multi-cycle rSLA

Trust (%)	C	Consumers								Providers							
		rSLA				bFee				rSLA				bFee			
		T (#)	TF (#)	SFR (%)	AE (€)	AC (€)	AL (€)	Bus (%)	Plat (%)	T (#)	TF (#)	SFR (%)	AR (€)	AC (€)	AL (€)	Bus (%)	Plat (%)
<b>20</b>	1	72	62	79	169.2	3.5	169.2	<b>2.0</b>	<b>2.0</b>	59	52	80	164.0	3.5	167.2	<b>2.0</b>	<b>2.0</b>
	2	24	20	83	147.5	4.8	147.2	2.0	<b>3.5</b>								
	3	15	13	80	110.9	3.9	111.0	1.9	<b>3.5</b>								
	4	11	9	82	74.4	2.6	74.5	1.9	<b>3.5</b>								
	5	12	10	83	37.5	1.3	37.5	1.9	<b>3.5</b>								
<b>40</b>	1	82	60	60	169.0	3.5	169.0	<b>2.0</b>	<b>2.0</b>	71	53	59	164.1	3.5	167.3	<b>2.0</b>	<b>2.0</b>
	2	57	36	60	144.6	4.8	144.3	1.9	<b>3.5</b>								
	3	48	30	60	108.8	3.0	108.5	1.9	<b>2.9</b>								
	4	36	22	61	72.4	1.6	72.2	1.9	<b>2.4</b>								
	5	46	28	61	36.2	0.8	36.2	1.9	<b>2.4</b>								
<b>60</b>	1	89	53	39	168.0	3.5	168.3	<b>2.0</b>	<b>2.0</b>	118	73	40	165.0	3.5	168.1	<b>2.0</b>	<b>2.0</b>
	2	100	43	40	140.7	4.0	140.3	1.9	<b>2.8</b>	2	1	50	141.7	2.8	145.2	<b>2.0</b>	<b>2.0</b>
	3	92	40	40	105.5	2.7	105.5	1.9	<b>2.6</b>								
	4	99	43	39	70.1	1.6	70.1	1.9	<b>2.4</b>								
	5	81	35	41	35.0	0.8	34.9	1.9	<b>2.2</b>								
<b>80</b>	1	130	53	20	167.7	3.5	167.6	<b>2.0</b>	<b>2.0</b>	442	114	20	159.9	3.0	163.4	<b>2.0</b>	<b>2.0</b>
	2	163	45	20	135.4	3.6	135.3	1.9	<b>2.7</b>	143	61	20	132.3	2.4	134.8	1.8	<b>1.9</b>
	3	139	40	20	101.5	2.4	101.2	1.8	<b>2.4</b>	112	50	20	99.3	1.8	101.2	<b>1.8</b>	<b>1.8</b>
	4	135	35	20	67.5	1.4	67.5	1.8	<b>2.1</b>	94	41	20	66.1	1.2	67.4	<b>1.8</b>	<b>1.8</b>
	5	135	34	20	33.7	0.6	33.7	<b>1.8</b>	<b>1.8</b>	100	39	20	33.0	0.6	33.7	<b>1.8</b>	<b>1.8</b>
<b>100</b>	1	1000	104	0	164.8	3.0	0	<b>2.0</b>	<b>2.0</b>	684	40	0	163.6	3.5	0	<b>2.0</b>	<b>2.0</b>
	2									199	81	0	137.2	2.4	0	1.7	<b>1.8</b>
	3									183	71	0	103.3	1.8	0	<b>1.7</b>	<b>1.7</b>
	4									186	68	0	68.6	1.2	0	<b>1.7</b>	<b>1.7</b>
	5									174	68	0	34.4	0.5	0	<b>1.6</b>	<b>1.6</b>

C – Cycle; T – Total number of rSLA; TF – Total number of Failed rSLA; SFR – rSLA Self Failure Rate; AE – Average resource Expenditure; AC – Average resource Cost; AL – Average resource Loss; AR – Average resource Revenue; Bus – bFee Business proposal; Plat – bFee Platform counter-proposal.

The consumers were able, in average, to negotiate 2567 and successfully trade 1753 resources per business in the five cycles (31.7% failure rate). In terms of the average number of resources required/offered per business, the businesses were able to successfully trade 35.1% of the 5000 required/offered. The rSLA were established by descending order of the trustworthiness of consumers and providers. The average price of the traded resources was inversely proportional to the trustworthiness of the group.

In this adverse scenario, where, cycle after cycle, businesses with lower trustworthiness are forced to trade among themselves, the overall losses and costs with brokerage renegotiation increase +1.5% and +0.3%, respectively, when compared to those without bSLA renegotiation. The impact of the bSLA renegotiation is, in terms of average amount spent per resource by the consumers and including costs and losses, +0.2% and, in terms of the average amount received per resource by the providers and deducing costs and losses, +0.1%.



## 5 Conclusion

The CloudAnchor platform was enriched with a brokerage renegotiation mechanism to allow registered businesses to negotiate the brokerage fee with the platform at will or periodically. Our contribution is a brokerage fee renegotiation algorithm contemplating both the business and the platform sides. On the business side, the brokerage fee is determined using the duration of the agreement, the cumulative number of traded resources as well as the success and the trustworthiness of the business in the platform. On the platform side, the brokerage fee is assessed based on the cumulative number of traded resources and the trustworthiness of the business. The algorithm was tested with short and long term rSLA in a demanding under supply resource scenario caused by a pre-defined average rSLA failure rate of 40.0 %.

For the short term rSLA, the implemented bSLA renegotiation mechanism, when compared with the prior open-ended bSLA, allows businesses to reduce brokerage costs by 21.1 %, while trading resources at the same price and experiencing identical losses. The revenues of providers and the expenditure of consumers are identical because they depend exclusively on the rSLA negotiation mechanism. The losses, which are originated by SLA violations, result in the application of the agreed penalties. In this case, the culprit reimburses the victim of the negotiated resource price accorded during the rSLA negotiation and bFee agreed during the bSLA negotiation. Since the resource price, when compared with the bFee, is predominant, the overall losses are identical. The costs, corresponding to the brokerage service fee, depend exclusively on the fee specified during the bSLA renegotiation and, consequently, benefit from the newly implemented mechanism.

In the case of the long term rSLA, the bSLA renegotiation introduces a reduced increase in the overall losses and costs of 1.5 % and 0.3 %, respectively. The resources, not only are negotiated by descending business trustworthiness, but, once negotiated, become unavailable till the end of the experiment, leaving, cycle after cycle, businesses with lower trustworthiness to negotiate together at higher prices.

As a trust-based mechanism, it promotes the fulfilment of agreements. On the one hand, increases the satisfaction of compliant businesses and, on the other hand, penalises faulty businesses, eventually leading to their de-registration from the platform. Finally, it privileges, first, the most reliable businesses, and among these, those who traded more resources, to the detriment of the less reliable ones, ensuring to the first ones trade resources at the best prices.

The renegotiation of bSLA can be refined by providing businesses with the ability to identify when to renegotiate in order to obtain the most favourable brokerage fees.

## Acknowledgements

This work was partially financed by the European Regional Development Fund (ERDF) through the Operational Programme for Competitiveness and Interna-

tionalisation (COMPETE Programme), within project «FCOMP-01-0202-FEDER-023151» and project «POCI-01-0145-FEDER-006961», and by national funds through the Fundação para a Ciência e Tecnologia (FCT) - Portuguese Foundation for Science and Technology - as part of project UID/EEA/50014/2013.

## References

1. G. Di Modica, V. Regalbuto, O. Tomarchio, and L. Vita. Enabling re-negotiations of sla by extending the ws-agreement specification. In *IEEE International Conference on Services Computing (SCC 2007)*, pages 248–251. IEEE, 2007.
2. G. Di Modica, O. Tomarchio, and L. Vita. Dynamic slas management in service oriented environments. *Journal of Systems and Software*, 82(5):759–771, 2009.
3. G. Frankova, D. Malfatti, and M. Aiello. Semantics and extensions of ws-agreement. *Journal of Software*, 1(1):23–31, 2006.
4. A. F. M. Hani, I. V. Paputungan, and M. F. Hassan. Renegotiation in service level agreement management for a cloud-based system. *ACM Computing Surveys (CSUR)*, 47(3):51, 2015.
5. M. Parkin, P. Hasselmeyer, and B. Koller. An sla re-negotiation protocol. In *Proceedings of the 2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC08), CEUR Workshop Proceedings, ISSN 1613-0073, Volume 411*. Citeseer, 2008.
6. S. Sharaf and K. Djemame. Enabling service-level agreement renegotiation through extending ws-agreement specification. *Service Oriented Computing and Applications*, 9(2):177–191, 2015.
7. B. Veloso, B. Malheiro, and J. C. Burguillo. Media brokerage: Agent-based sla negotiation. In A. Rocha, M. A. Correia, S. Costanzo, and P. L. Reis, editors, *New Contributions in Information Systems and Technologies: Volume 1*, pages 575–584. Springer International Publishing, Cham, 2015.
8. B. Veloso, B. Malheiro, and J. C. Burguillo. Cloudanchor: Agent-based brokerage of federated cloud resources. In Y. Demazeau, T. Ito, J. Bajo, and J. M. Escalona, editors, *Advances in Practical Applications of Scalable Multi-agent Systems. The PAAMS Collection: 14th International Conference, PAAMS 2016, Sevilla, Spain, June 1-3, 2016, Proceedings*, pages 207–218. Springer International Publishing, Cham, 2016.
9. B. Veloso, F. Meireles, B. Malheiro, and J. C. Burguillo. Federated iaas resource brokerage. In G. Kecskemeti, A. Kertesz, and Z. Nemeth, editors, *Interoperable and Federated Cloud Architecture*, chapter 9, pages 252–280. IGI Global, Hershey, PA, USA, 2016.