CloudAnchor: Agent-Based Brokerage of Federated Cloud Resources

Bruno Veloso[1,2], Benedita Malheiro[2,3](✉), and Juan Carlos Burguillo[1]

[1] EET/UVigo – School of Telecommunication Engineering,
University of Vigo, Vigo, Spain
[2] INESC TEC, Porto, Portugal
[3] ISEP/IPP – School of Engineering,
Polytechnic Institute of Porto, Porto, Portugal
mbm@isep.ipp.pt

Abstract. This paper presents CloudAnchor, a brokerage platform conceived to help Small and Medium Sized Enterprises (SME) embrace Infrastructure as a Service (IaaS) cloud computing both as providers and consumers. The platform, which transacts automatically single and federated IaaS cloud resources, is a multi-layered Multi-Agent System (MAS) where providers, consumers and virtual providers, representing provider coalitions, are modelled by dedicated agents. Federated resources are detained and negotiated by virtual providers on behalf of the corresponding coalition of providers. CloudAnchor negotiates and establishes Service Level Agreements (SLA) on behalf of SME businesses regarding the provision of brokerage services as well as the provision of single and federated IaaS resources. The discovery, invitation, acceptance and negotiation processes rely on a distributed trust model designed to select the best business partners for consumers and providers and improve runtime.

Keywords: Federated resources · Multi-Agent System (MAS) · Infrastructure as a Service (IaaS) · Service Level Agreement (SLA) · Trust-based brokerage

1 Introduction

The IaaS cloud computing market, while still maturing, is already populated by major players, making it difficult for SME providers to thrive. In this scenario, brokers, which are entities that manage the use, performance and delivery of cloud services, and negotiate relationships between cloud providers and consumers [7], are emerging as the preferential middle-ware to match demand and offer. SME are still in the early stages of adopting the cloud paradigm or providing cloud services and, consequently, require support services and platforms to migrate to the cloud. For SME providers, brokers offer additional business opportunities and simplify the management and integration of disparate cloud services — potentially across different providers —fostering the creation of provider coalitions. In the case of SME consumers, brokers provide seamless provider lookup

Y. Demazeau et al. (Eds.): PAAMS 2016, LNAI 9662, pp. 207–218, 2016.
DOI: 10.1007/978-3-319-39324-7_18

and invitation as well as SLA negotiation services, increasing the chances of obtaining the desired resources at the best price and within the deadline. The ultimate goal of this research is to support the adoption and provision of IaaS by SME both as consumers and as providers.

This paper describes CloudAnchor, an SME business-to-business (B2B) brokerage platform regarding single and federated infrastructure resources. The federated resources must be decomposable in standard Virtual Machine (VM) packages. This problem is by nature distributed, decentralised, dynamic and involves multiple stakeholders (consumers and providers) continuously entering and leaving the system (open system). The stakeholders are loosely coupled, but, depending on the situation, can either compete (consumers and providers compete for getting and leasing resources) or cooperate (coalition of providers). Furthermore, businesses wish to retain autonomy, privacy and the control of their strategic knowledge, leading to the adoption of the agent-based paradigm.

In terms of contributions, the proposed brokerage platform provides: (*i*) decentralised trust models of partners (providers, consumers and platform) based on the outcome of partner invitation, partner acceptance, SLA negotiation and SLA enforcement; (*ii*) trust-based invitation/acceptance of potential partners; (*iii*) trust-based negotiation and establishment of brokerage, coalition and resource SLA instances; and (*iv*) federated resources through the creation of virtual providers, representing provider coalitions. In particular, the creation and application of such partner trust models is, as far as we know, a novel approach.

In terms of organisation, this document contains five sections. Section 2 is dedicated to cloud brokerage. It introduces the most relevant concepts of cloud brokerage as well as describes related agent-based cloud resource brokerage platforms. Section 3 describes our approach, including the platform architecture and services. Section 4 describes the tests and discusses the results obtained. Finally, Sect. 5 draws the conclusions and suggests future developments.

2 Cloud Brokerage

According to the NIST, a cloud broker may include service intermediation, aggregation and arbitrage [7]. Such tasks require the modelling of resources (single and federated), businesses and their relationships (SLA) as well as methodologies for partner discovery, resource negotiation and resource provision. Frequently, brokers build trust and/or reputation models to support SLA negotiation. In recent years, several agent-based cloud brokers were proposed by An *et al.* (2010) [3], Venticinque *et al.* (2011) [13], Ferrer *et al.* (2012) [6], Al Falasi *et al.* (2013) [1] and Pawar *et al.* (2014) [9]. While An *et al.* propose a platform for single resource SLA negotiation, the remaining platforms offer single and federated resource SLA negotiation services and adopt trust and/or reputation models of business partners. In terms of services, 80 % include partner discovery, 100 % include SLA negotiation and 40 % SLA enforcement.

Service Level Agreements. The establishment of Service Level Agreements (SLA) is the ultimate goal of a broker. The adoption of specifications and standards to represent SLA instances are essential to ensure interoperability. There are two major concurrent specifications: the Web Service Agreement (WS-Agreement)[1] created by the Open Grid Forum (OGF); and Web Service Level Agreement (WSLA)[2] created by International Business Machines (IBM). However, in the case of cloud federated resources, these specifications do not cover all the needs. Whereas Alhamad *et al.* (2010) rely on performance and business metrics to establish SLA contracts between consumers and providers [2], we adopt a trust-based performance metric. Moreover, we define brokerage, resource and coalition SLA to represent the different business relationships and their dependencies [11].

Peer Trust and Reputation. Trust is, by default, a subjective property of direct (one-to-one) relationships attributed by a trustor to a trustee and, according to Castelfranchi, implies a decision to rely on someone [5]. Built from the outcomes of past interactions, it is typically intended to be used in future interactions between trustor and trustee. Reputation is obtained from third parties and can be used to characterise new business partners or complement trust built from first hand information. Pinyol and Sabater (2013) classify reputation and trust models according to: (*i*) the paradigm, *i.e.* if the model is cognitive or numerical; (*ii*) the information sources; (*iii*) the visibility, *i.e.* if the trust can be observed by other agents; (*iv*) the granularity, *i.e.* the context of the trust information; (*v*) the source behaviour, *e.g.* credible or deceptive; and (*vi*) the type of information exchanged [10]. In the specific case of analysed agent-based cloud brokers, trust and/or reputation models have been used in the partner discovery stage by [6,13] or [9] and in the SLA negotiation stage by [1] (to specify rewards and penalties). In CloudAnchor, we apply the trust models in both stages: consumers use trust for partner lookup and invitation and for SLA negotiation, whereas providers for invitation assessment and SLA negotiation.

SLA Negotiation. For Bichler *et al.*, negotiation can be unstructured, semi-structured or structured and includes six main attributes: (*i*) the number of participants (bilateral, multilateral, multi-bilateral or arbitrary); (*ii*) the number of issues (single or multidimensional); (*iii*) the scope (restricted or open); (*iv*) the domain of the offers (private or public); (*v*) the scenario (competitive and/or cooperative); and (*vi*) the protocol [4]. In the case of SLA negotiation, An *et al.* (2010) adopt a cloud resources negotiation protocol composed of alternating offers, commonly used for bilateral bargaining, which includes the negotiation of SLA violation fines [3]. Venticinque *et al.* (2011) use the Iterated Contract Net Interaction Protocol (ICNIP) for SLA negotiation and take into account multiple constraints and parameters, including provider reputation [13]. CloudAnchor adopts two negotiation protocols – the one shot bilateral protocol for bSLA and the ICNIP for rSLA and cSLA – and implements a multidimensional negotiation, including partner trust as well as resource price and uptime.

[1] https://www.ogf.org/documents/GFD.107.pdf.

[2] http://www.research.ibm.com/people/a/akeller/Data/WSLASpecV1-20030128.pdf.

3 CloudAnchor Broker

CloudAnchor, when compared with the analysed cloud brokerage platforms, implements two distinctive features: (*i*) a business model contemplating the negotiation, establishment and enforcement of brokerage, coalition and resource agreements; and (*ii*) a decentralised trust model of the peers, taking into account all past interactions, including provider invitation/acceptance, SLA negotiation and SLA enforcement outcomes, and not just the usual SLA enforcement results. This novel approach allows consumers to invite providers based on the outcome of past provider/consumer invitation/acceptance ratios as well as provider and consumer SLA establishment/negotiation and enforcement. The SLA enforcement results are provided by the SLA monitoring and enforcement modules, which, in our case, are external to the platform.

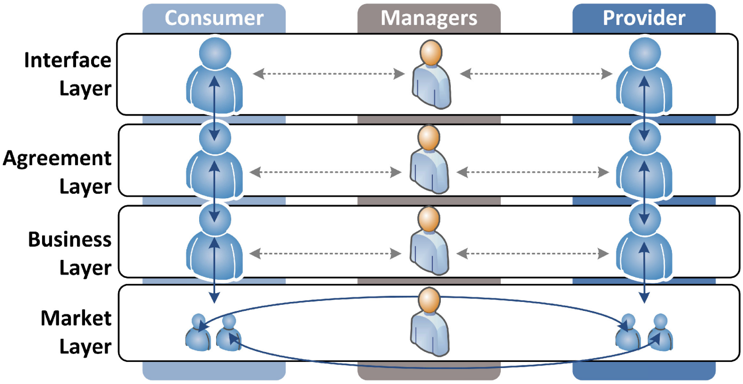

Fig. 1. CloudAnchor broker architecture

The broker architecture, which is detailed in [12], is organized in interface, agreement, business and market layers (Fig. 1) and comprises of five types of specialised dedicated agents: (*i*) interface agents to interact with consumer and provider SME businesses; (*ii*) agreement agents to manage SLA instances; (*iii*) business agents to model consumer and provider SME businesses; (*iv*) market delegate agents to negotiate specific resources on behalf of consumer and provider SME businesses; and (*v*) layer manager agents responsible for the management of platform layers (creation/removal of agents in the layer). Each business (consumer or provider SME) is represented in the platform by the corresponding: (*i*) interface agent located in interface layer; (*ii*) agreement agent located in agreement layer; (*iii*) business agent in the business layer; and (*iv*) an undetermined number of delegate agents involved in specific resource negotiations in the market layer. These agents are identified by a trading code, preventing third parties from intruding in undergoing negotiations [11]. To overcome the existing interoperability issues between different IaaS platforms, CloudAnchor interacts with the provider platforms through the Deltacloud abstraction framework [8].

The CloudAnchor brokerage platform implements an open event-driven multi-layered agent-based architecture. Businesses are represented by dedicated autonomous agents and are, thus, able to specify their self-models, by uploading their strategic knowledge (lookup, invitation, acceptance and negotiation strategies), resource offers (providers) or resource requests (consumers), as well as to build peer models of their business partners based on the outcomes of their previous interactions (local peer trust). The layered approach allows the distribution and delegation of the interface, agreement, business (knowledge and processes) and negotiation related tasks to corresponding dedicated agents, where each business is represented by a set of dedicated specialized agents rather than by a single agent to increase the overall responsiveness. In terms of external events, there are business registration/de-registration, resource request/offer and SLA fulfilled/violated events. These events drive the execution of the business registration service, business de-registration service, provider lookup and invitation service, provider resource publication service and SLA termination service. In particular, whenever a consumer requests a new resource via its interface agent (resource request event), it triggers the resource finding process. The consumer business agent automatically looks up and invites providers for negotiation. If the invited provider business agents accept the invitation, dedicated delegate market agents are created by both consumer and provider business agents to negotiate and establish the resource SLA (rSLA). If the providers are unable to provide the resource single-handedly, the platform attempts to create a virtual provider. Virtual providers are temporary coalitions of providers established on the fly to provide federated resources, *i.e.*, resources which were not offered by any single provider. When an rSLA terminates, the parties involved (consumer and provider) receive an agreement fulfilled or agreement violation event.

SLA Templates, Classes and Hierarchy. The platform contemplates the negotiation, establishment and termination of: (*i*) brokerage agreements – bSLA – which define the platform service provision terms for each business; (*ii*) resource agreements – rSLA — that specify the resource provisioning terms between businesses; and (*iii*) coalition agreements – cSLA — which details the platform coalition service provision terms between the businesses that form the coalition. A bSLA establishes a one-to-one relationship between a business and the brokerage platform; an rSLA establishes a one-to-one relationship between one consumer and one provider businesses; and a cSLA establishes a one-to-many binding relationship between a virtual provider and a collection of providers. The different agreements are described according to the WS-Agreement specification. There is a hierarchical dependency between the different types of SLA. A bSLA defines the general conditions which apply to any service performed on behalf of the business by the platform, including resource (rSLA) and coalition agreements (cSLA). As a result, when two businesses establish a resource provision contract (rSLA), they must fulfil the agreed brokerage, coalition (in the case of a federated resource) and service provision terms. Figure 2 describes the SLA life-cycle state: SLA template (Temp), SLA partially instanced (Inst), SLA under negotiation (Nego), SLA under enforcement SLA (Enfo) and SLA terminated (Term).

The state transitions are event-driven: (*i*) when a service becomes available, the provider partially instances an SLA (Temp to Inst); (*ii*) when a provider accepts a consumer invitation, they engage in the negotiation of the SLA terms (Inst to Nego); (*iii*) if they reach an agreement, the SLA is established and applied (Nego to Enfo), otherwise the SLA terminates (Nego to Term); (*iv*) when the enforcement succeeds or fails, the SLA terminates (Enfo to Term) and the service becomes available (Term to Temp).

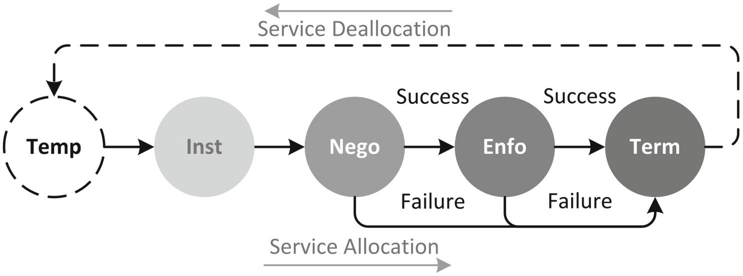

Fig. 2. SLA life-cycle

The provision of a resource results in several payments: (*i*) the consumer pays the established resource provisioning fee to the provider; (*ii*) the provider pays the accorded brokerage fee to the platform; and (*iii*) the consumer pays the negotiated brokerage fee to the platform. In the case of a federated resource, there is an additional brokerage fee that the federated providers pay to the platform. These fees are typically distinct. According to the default SLA terms, if a business fails to fulfil an established SLA, it reimburses the partner.

Distributed Trust Model. The platform creates and maintains a distributed decentralised self and acquaintances models of the business partners with the trust model. Each entity (platform, consumer, provider and virtual provider) builds these local models based on their past common interactions, *i.e.*, the ratio of successful invitations, negotiations and fulfilled SLA. While consumers, providers and virtual providers hold a partial incomplete trust model of their counterparts, the platform is in a unique position as it keeps a global and complete trust model of all businesses. The CloudAnchor distributed trust model implements, according to Pinyol and Sabater (2013) classification, a local (granularity), private (visibility) numerical approach (paradigm) based solely on direct interactions (sources). The trust model supports all brokerage stages: provider invitation (I), SLA negotiation (N) and SLA enforcement (E). Each business agent builds corresponding self and partner models. By default, at start up, businesses are fully trusted. For a given brokerage stage S, the local dynamic trustworthiness attributed by a trustor business a to a trustee partner b is given by Eq. 1:

$$T_S(a,b)_n = \frac{n-1}{n} \times T_S(a,b)_{n-1} + \frac{1}{n} \times Out_{S,n} \qquad (1)$$

where n is the number of stage S interactions accomplished between a and b, and $Out_{S,n}$ is the boolean outcome of the last stage S interaction – success (1) or failure (0). As a result, each business agent builds local partner invitation (I), SLA negotiation (N) and SLA enforcement (E) models: (i) $T_I(a,b)_n$ corresponds to the ratio of acceptances versus invitations to negotiate; (ii) $T_N(a,b)_n$ is the ratio of established versus negotiated SLA; and (iii) $T_E(a,b)_n$ is the ratio of fulfilled versus established SLA. Additionally, businesses maintain their own I, N and E self models. Equation 2 represents, for a given brokerage stage S, the self trustworthiness of business a:

$$T_S(a)_n = \frac{\sum_{i=1}^{n} Out_{S,i}}{n} \tag{2}$$

where n is the total number of stage S interactions accomplished with all business partners and $Out_{S,i}$ is the boolean outcome of the last stage S interaction. In the case of a virtual provider vp created on behalf of a consumer c, the initial self model values $T_S(vp)_0$ are obtained through Eq. 3, where $T_S(c,p_i)_n$ corresponds to the trustworthiness of provider p_i according to consumer c.

$$T_S(vp)_0 = \min_{i=1,\ldots m} T_S(c,p_i)_n \tag{3}$$

Provider Lookup and Invitation. In this stage consumers discover and invite providers for negotiation and providers decide whether to accept or reject the invitations. Consumers select and invite the best provider candidates for negotiation by applying a cascade filter based on the enforcement, negotiation and invitation trustworthiness of the potential providers. Providers accept consumer invitations based on the consumer enforcement trustworthiness. The provider invitation algorithm (Algorithm 1) looks for providers (line 1) in the service registry, calculates the acceptance threshold for each filter (line 2, 6 and 12) and implements the T_E based filter (lines 3–5), the T_N based filter (lines 7–11) and the T_I based filter (lines 11–13). Algorithm 2 determines, on the provider side, when to accept a consumer invitation by calculating the mean enforcement trustworthiness – $\overline{T_E}$ (line 2) and $\overline{T_N}$ (line 3) – of all provider clients and verifying that $T_E(p,c) \geq \overline{T_E}$, where p represents the provider and c the consumer.

Algorithm 1. Lookup and invitation

```
1:  $L[n] \leftarrow LookUp$
2:  $\overline{T_E} \leftarrow CalcT_E Mean(L[n])$
3:  for $p = 0; p < n; p{+}{+}$ do
4:      if $L[p].T_E(c,p) \geq \overline{T_E}$ then
5:          $L_E[p] \leftarrow L[p]$
6:  $\overline{T_N} \leftarrow CalcT_N Mean(L_E[m])$
7:  for $p = 0; p < m; p{+}{+}$ do
8:      if $L_E[p].T_N(c,p) \geq \overline{T_N}$ then
9:          $L_N[p] \leftarrow L_E[p]$
10: $\overline{T_I} \leftarrow CalcT_I Mean(L_N[l])$
11: for $p = 0; p < l; p + +$ do
12:     if $L_N[p].T_I(c,p) \geq \overline{T_I}$ then
13:         $L_I[p] \leftarrow L_E[p]$
14: return $L_I[p]$
```

Algorithm 2. Acceptance

```
1: $L[c] \leftarrow ClientConsumerList$
2: $\overline{T_E} \leftarrow CalcT_E Mean(L[c])$
3: $\overline{T_N} \leftarrow CalcT_N Mean(L[c])$
4: if $T_E(p,c) \geq \overline{T_E}$ then
5:     return accept
6: return reject
```

SLA Negotiation. CloudAnchor implements a competitive, private, restricted (by invitation), multidimensional (uptime, price and trust) bilateral structured protocol – the ICNIP – for rSLA and cSLA negotiations and a private, restricted, multidimensional (time, price and trust) bilateral protocol - the one shot protocol – for bSLA negotiations. The negotiation of coalition and resource agreements takes place in the market layer between provider and consumer delegate agents. Provider delegates implement a dynamic price adaptation strategy according to Eq. 4c, where r represents the resource, p the provider, c the consumer and n the current negotiation round. First, they determine the maximum resource price to propose to the consumer $Pr(p,c,r)_{max,n}$ using Eq. 4a, which takes into account $T_E(c,p)_n$, *i.e.* the provider's enforcement trustworthiness as perceived by the consumer, the maximum resource price range $\Delta Pr(p,r)$ and the minimum resource price $Pr_{min}(p,r)$. Then, they determine the consumer's loyalty discount through Eq. 4b. The discount depends on the ratio between the uptime of the resources delivered to the consumer and the total number of provider's resource hours, the consumer's enforcement trustworthiness according to the provider $T_E(p,c)_n$ and the current resource price range $Pr(p,c,r)_{max,n} - Pr_{min}(p,r)$. Finally, Eq. 4c establishes the current price proposal $Pr(p,c,r)_n$ as the difference between the consumer's maximum price $Pr(p,c,r)_{max,n}$ and discount $Di(p,c,r)_n$.

$$Pr(p,c,r)_{max,n} = Pr_{min}(p,r) + T_E(c,p)_n \times \Delta Pr(p,r) \tag{4a}$$

$$Di(p,c,r)_n = \frac{\sum_{i=1}^{d} \Delta t_{up,i}}{n_r \times \Delta t} \times T_E(p,c)_n \times (Pr(p,c,r)_{max,n} - Pr_{min}(p,r)) \tag{4b}$$

$$Pr(p,c,r)_n = Pr(p,c,r)_{max,n} - Di(p,c,r)_n \tag{4c}$$

Consumer delegates determine the utility of provider proposals through Eq. 5, where $U(c,p,r)_n$ represents the utility of the provider's proposal according to the consumer, $T_E(c,p)_n$ is the current provider's enforcement trustworthiness as perceived by the consumer, $Pr(p,c,r)_n$ is the proposed resource price, $\Delta t_{up}(r)$ is the proposed resource uptime and, finally, the α_c parameter allows the consumer to define the relative importance between the price and uptime dimensions.

$$U(c,p,r)_n = T_E(c,p)_n \times (\alpha_c(1 - Pr(p,c,r)_n) + (1 - \alpha_c)(\Delta t_{up}(r)). \tag{5}$$

4 Tests and Results

We performed several tests to verify the correct operation of the platform and the impact of the trust-based invitation/acceptance and negotiation in terms of the negotiated resource price and negotiation time involving one consumer and multiple providers. The tests contemplated three provision – undersupply, equilibrium and oversupply – and three resource consumption scenarios: (*i*) single resources, *i.e.*, when the demand can be met by any single provider; (*ii*) federated resources, *i.e.*, when the demand can only be met by coalitions of providers; and (*iii*) mixed (single and federated) resource provision, *i.e.*, when the demand includes both single and federated resources (Table 1). The resulting nine experiments were performed with the base algorithm and with the application of the

Trust-based Invitation/acceptance and Negotiation (TIN) model. In the base algorithm the peer trustworthiness remains 100 %. During these experiments, the consumer remains fully trusted and requests 1000 standard virtual machines (VM) in different combinations: (*i*) 1000 VM in the single resource provision scenario; (*ii*) 500 VM and 20 packages of 25 VM in the mixed resource provision scenario; and (*iii*) 40 packages of 25 VM in the federated resource provision scenario. In the case of the providers there are: (*i*) 25 providers holding 20 standard VM each in the under-supply market scenario; (*ii*) 50 providers holding 20 standard VM each in the supply and demand equilibrium market scenario; and (*iii*) 100 providers holding 20 standard VM each in the oversupply market scenario. During these tests, all providers violate 25 % of the established rSLA, ending up with 75 % trustworthiness. Providers implement identical price adaptation policies, *i.e.*, they apply Eq. 4c with an initial maximum value of 47 € and a minimum value of 27 € per standard VM. In terms of hardware, the tests were executed on a platform with one quad-core i7-2600 3.40 GHz Central Processing Unit (CPU) with 2 threads per core, 16 GiB Random Access Memory (RAM) and a 1.8 TiB of storage capacity.

Table 1. Resource consumption and provision scenarios

Resource	Scenario	Businesses	VM/business	$\overline{T_E}$ (%)
Demand	Single	1 consumer	1000×1	100
	Mixed		$500 \times 1 + 20 \times 25$	
	Federated		40×25	
Offer	Undersupply	25 providers	20	75
	Equilibrium	50 providers		
	Oversupply	100 providers		

Figure 3 presents the base test results (without TIN). We can observe that the price depends solely on the resource type ($\overline{Pr_{fed}} > \overline{Pr_{sin}}$) and is independent of the number of providers. In terms of runtime, the negotiation time per VM decreases from single to federated resources ($\overline{\Delta t_{sin}} > \overline{\Delta t_{mix}} > \overline{\Delta t_{fed}}$) because federated resources are negotiated in packages.

Figure 4 groups the TIN results. We can observe that the average VM price increases with the number of providers ($\overline{Pr_{ove}} > \overline{Pr_{equ}} > \overline{Pr_{und}}$) and with the resource type ($\overline{Pr_{fed}} > \overline{Pr_{sin}}$). In terms of runtime, the negotiation time per VM decreases from single to federated resources ($\overline{\Delta t_{sin}} > \overline{\Delta t_{mix}} > \overline{\Delta t_{fed}}$) as expected. Table 2 presents the comparison between the base and TIN results. The TIN average VM price is lower in all cases and the TIN negotiation time decreases considerably when compared with the base values. These results show that the trust-based invitation/acceptance and trust-based negotiation grants faster negotiations, better proposals (lower prices) and selects the best providers in each negotiation.

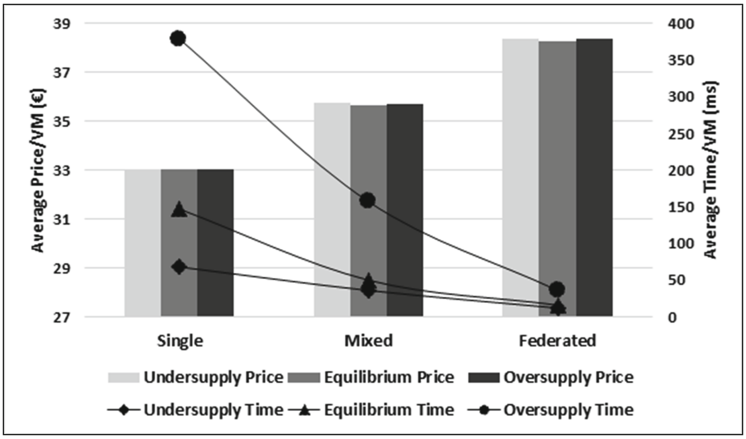

Fig. 3. Base results

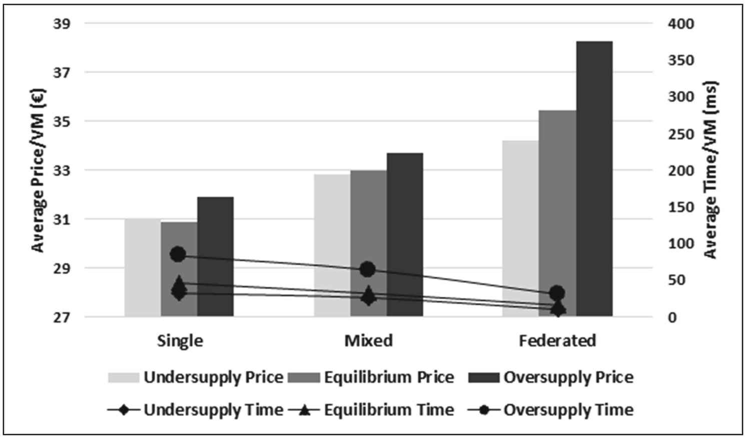

Fig. 4. TIN results

Table 2. TIN *versus* base results

	Single			Mixed			Federated		
	Under	Equi	Over	Under	Equi	Over	Under	Equi	Over
$\Delta Price$	−06.3 %	−06.5 %	−03.4 %	−08.1 %	−07.5 %	−05.6 %	−10.8 %	−07.4 %	−00.3 %
$\Delta Time$	−52.9 %	−68.7 %	−78.1 %	−27.7 %	−36.0 %	−59.2 %	−16.6 %	−06.3 %	−13.9 %

5 Conclusions

CloudAnchor offers cloud infrastructure brokerage services for SME vendors and consumers regarding single and federated resources, which must be decomposable in standard VM packages. The brokerage services include provider lookup and discovery, trust-based provider selection, invitation and acceptance as well as trust-based SLA negotiation.

The platform contemplates the negotiation and establishment of three different types of SLA – brokerage, resource and coalition – supported by the decentralised peer trust model. This novel model takes all past interactions into account, including provider invitation, resource negotiation and resource provision, and not just the SLA enforcement outcomes. Our approach allows: (*i*) consumers to invite providers based on the outcome of past invitations, SLA negotiations and SLA enforcements; and (*ii*) providers to decide whether or not to accept an invitation for negotiation based on the outcome of past SLA negotiation and SLA enforcement. The trust-based results demonstrate the usefulness of the model showing a considerable reduction of the average resource negotiation time and a slighter decrease of the average resource price paid by the consumers.

In terms of future developments, we plan to: (*i*) implement the renegotiation of bSLA and cSLA based on the trust model and on the success of the businesses within the platform; (*ii*) enrich SLA templates with new parameters to meet increasing business demands; and (*iii*) create virtual providers not only at the request of consumers, but also at the request of providers. Concerning the validation of the platform, we are performing experiments to assess the impact of the distributed trust model from the provider perspective, *i.e.*, when consumers violate agreements.

Acknowledgements. This work was partially financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE Programme within project ≪FCOMP-01-0202-FEDER-023151≫ and project ≪POCI-01-0145-FEDER-006961≫, and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

References

1. Al Falasi, A., Serhani, M.A., Elnaffar, S.: The sky: a social approach to clouds federation. Procedia Comput. Sci. **19**, 131–138 (2013)
2. Alhamad, M., Dillon, T., Chang, E.: Conceptual SLA framework for cloud computing. In: 2010 4th IEEE International Conference on Digital Ecosystems and Technologies (DEST), pp. 606–610. IEEE (2010)
3. An, B., Lesser, V., Irwin, D., Zink, M.: Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 981–988. International Foundation for Autonomous Agents and Multiagent Systems (2010)
4. Bichler, M., Kersten, G., Strecker, S.: Towards a structured design of electronic negotiations. Group Decis. Negot. **12**(4), 311–335 (2003)
5. Castelfranchi, C., Falcone, R.: Principles of trust for MAS: cognitive anatomy, social importance, and quantification. In: Proceedings of the International Conference on Multi Agent Systems, pp. 72–79. IEEE (1998)

6. Ferrer, A.J., Hernández, F., Tordsson, J., Elmroth, E., Ali-Eldin, A., Zsigri, C., Sirvent, R., Guitart, J., Badia, R.M., Djemame, K., et al.: Optimis: a holistic approach to cloud service provisioning. Future Gener. Comput. Syst. **28**(1), 66–77 (2012)
7. Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., Leaf, D.: NIST Cloud Computing Reference Architecture: Recommendations of the National Institute of Standards and Technology (Special Publication 500–292). CreateSpace Independent Publishing Platform, USA (2012)
8. Meireles, F., Malheiro, B.: Integrated management of IaaS resources. In: Lopes, L., et al. (eds.) Euro-Par 2014, Part II. LNCS, vol. 8806, pp. 73–84. Springer, Heidelberg (2014)
9. Pawar, P.S., Rajarajan, M., Dimitrakos, T., Zisman, A.: Trust assessment using cloud broker. In: Zhou, J., Gal-Oz, N., Zhang, J., Gudes, E. (eds.) Trust Management VIII. IFIP AICT, vol. 430, pp. 237–244. Springer, Heidelberg (2014)
10. Pinyol, I., Sabater-Mir, J.: Computational trust and reputation models for open multi-agent systems: a review. Artif. Intell. Rev. **40**(1), 1–25 (2013)
11. Veloso, B., Malheiro, B., Burguillo, J.C.: Media brokerage: agent-based SLA negotiation. In: Rocha, A., Correia, A.M., Costanzo, S., Reis, L.P. (eds.) New Contributions in Information Systems and Technologies. AISC, vol. 353, pp. 575–584. Springer, Heidelberg (2015)
12. Veloso, B., Meireles, F., Malheiro, B., Burguillo, J.C.: Federated IaaS resource brokerage. In: Kecskemeti, G., Kertes, A., Nemeth, Z. (eds.) Developing Interoperable and Federated Cloud Architectures, Chap. 9. IGI Global, Pennsylvania (2016)
13. Venticinque, S., Aversa, R., Di Martino, B., Rak, M., Petcu, D.: A cloud agency for SLA negotiation and management. In: Guarracino, M.R., et al. (eds.) Euro-Par-Workshop 2010. LNCS, vol. 6586, pp. 587–594. Springer, Heidelberg (2011)