Comparison of Surveillance Strategies to Identify Undesirable Behaviour in Multi-Agent Systems

Sarah Edenhofer[1], Christopher Stifter[1], Sven Tomforde[1], Jan Kantert[2], Christian Müller-Schloer[2] and Jörg Hähner[1]

[1]*Organic Computing Group, University of Augsburg, Augsburg, Germany*

[2]*Institute of Systems Engineering, Leibniz University Hannover, Hannover, Germany*

Keywords: Multi-Agent Systems, Trust, Cooperation, Coordination, Self-organising Systems, Organic Computing.

Abstract: Open, distributed systems face the challenge to maintain an appropriate operation performance even in the presence of bad behaving or malicious agents. A promising mechanism to counter the resulting negative impact of such agents is to establish technical trust. In this paper, we investigate strategies to improve the efficiency of trust mechanisms regarding the isolation of undesired participants by means of reputation and accusation techniques. We demonstrate the potential benefit of the developed techniques within simulations of a Trusted Desktop Computing Grid.

1 INTRODUCTION

Open distributed systems are characterised by the possibility to join and leave at any time. This entails the problem that the participating elements (i.e. the agents) are not under control of a centralised entity – consequently, their behaviour has to be considered as black-box. These black-boxes can contribute to the overall system's goal, but they can also disturb the efficient operation by means of unintentionally wrong or even intentionally malicious behaviour.

A basic mechanism to counter the resulting negative impact of such disturbances is the utilisation of technical trust. Establishing trust relationships among distributed agents that model the reliability and trustworthiness of interaction partners allows for isolating untrusted agents, see (Steghöfer and Reif, 2012). In this paper, we investigate techniques to improve the efficiency of isolation mechanisms on the basis of technical trust. Therefore, we introduce and compare two novel strategies for distributed surveillance: a reputation-based and an accusation-based strategy. Within simulations of our application scenario, we demonstrate the potential benefit of these strategies and highlight that the duration until an effective isolation of malicious elements takes place can be reduced significantly.

The remainder of this paper is structured as follows: In Section 2, we introduce our application scenario – the Trusted Desktop Grid (TDG) – and define agents categories and their goals. Section 3 presents the novel strategies to isolate malicious agents using reputation and accusation techniques. Afterwards, Section 4 evaluates the approach using simulations of the TDG. Section 5 discusses the achieved results and derives a statement of which technique should be used in which cases, we present related work in Section 6. Finally, Section 7 summarises the paper and gives an outlook to future work.

2 TRUSTED DESKTOP GRID

We use an open, distributed Trusted Desktop Grid (TDG) as application scenario to show and prove the effective application of distributed algorithms as well as Organic Computing (Müller-Schloer et al., 2011) methods. In this scenario, we use an open and heterogeneous Multi-Agent System (MAS) and we do not assume benevolence. The agents in the system cooperate to gain an advantage. The mechanism determining this cooperation is *Trust*. Because of the openness of the system, different agents may try to exploit it. They may be uncooperative, malfunctioning or even malicious.

An agent, which acts on behalf of the user, is submitting *jobs* it wants to have calculated (Klejnowski,

Edenhofer, S., Stifter, C., Tomforde, S., Kantert, J., Müller-Schloer, C. and Hähner, J.
Comparison of Surveillance Strategies to Identify Undesirable Behaviour in Multi-Agent Systems.
DOI: 10.5220/0005679201320140
In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016) - Volume 1*, pages 132-140
ISBN: 978-989-758-172-4

2014). Each job is assumed to be composed of several independently processable *work units* (WU). The agents are expected to volunteer their machines as workers for other agents' WU as well as to share their resources.

2.1 Agent Goal

The benefit of an agent can be measured by its *speedup* σ, informally speaking its benefit of having its work processed distributively over having to process all work on its own (in accordance with (Klejnowski, 2014)). A job J is a set of WUs, which is released in time step t_J^{rel} and completed in t_J^{compl}, when the last WU is finished.

The speedup σ in Equation 1 is a metric known from multi-core systems. It is based on the assumption that parallelisation helps to process a task (i.e. a job) faster than processing it on a single core. σ is the ratio of the time the agent would have needed to process all WUs on its own to the real time it took to calculate J distributedly in the system. This is why the speedup can only be determined after the last result has been returned to the submitter.

$$\sigma = \frac{\sum_J (t_{self}^{compl} - t_{self}^{rel})}{\sum_J (t_{dist}^{compl} - t_{dist}^{rel})} \quad (1)$$

In short, we can write $\sigma := \frac{t_{self}}{t_{dist}}$ with t_{self} being the time it would require an agent to process all WUs of a job without cooperation, i.e. sequentially. t_{dist} is the time it takes until all WUs are computed distributedly and the last result is returned to the submitting agent. If no cooperation partners can be found, agents need to calculate their own WUs. This results in a speedup value equal to one. In general, we assume that agents behave selfishly and only cooperate if they can expect an advantage, i.e. $\sigma > 1$.

2.2 Worker and Submitter Component

Each agent is free do decide which agent it wants to give its WUs to and for which agents it wants to work for. Therefore, every agent has a *submitter* and a *worker* component.

The *submitter component* is the scheduler of the agent and responsible for distributing WUs. If an agent receives a job J from the user consisting of multiple WUs, it creates a list of suited workers, i.e. workers it trusts. It then asks workers from this list to cooperate and calculate WUs, until either no more WU or no more workers are left. If all workers were asked and still unprocessed WUs remain, the agent calculates them on its own.

The *worker component* decides whether an agent wants to work for a certain submitter. When the agent receives a request to process a WU, it calculates its expected reward for accepting and rejecting the WU. If the reward of accepting the WU prevails, the agent takes the WU, puts it in its own working queue, where the WU remains until the agent starts to process it, i.e. until the other WU in the queue were processed. Afterwards, it transfers the result back to the submitter where the result is validated (Klejnowski, 2014). A job is completed, if all WUs were returned to the submitter.

2.3 Global Goal

The global goal—also referred to as the system goal—is to enable and encourage agents to cooperate and thereby achieve the best possible average σ. The systems' focus is *coordination*, i.e. shaping the environment in a way that allows for cooperation and, thereby, leads to optimising the global goal.

2.4 Agent Types

In the context of our TDG, such disturbances are, for example, agents that return wrong results, or agents that refuse to work for other agents while submitting WU to them. Behaviour like this can lead to a lower system-speedup. In the following, we discuss different types of agents, each behaving differently.

Adaptive Agents are cooperative. They work for other agents which have good reputation in the system. If, for example, the WU-queue of this agent is saturated to capacity, the agent may reject another WU.

Altruistic Agents accept every job, regardless of the circumstances and cooperation-partners.

Freeriders do not work for other agents and reject all work requests. However, they ask other agents to work for them. This increases the overall system load and can decrease the benefit for well-behaving agents.

Egoists only pretend to work for other agents. They accept most work requests but return fake results. This wastes the time of other agents. If results have to be validated, the average σ is decreased.

Sloppy Agents are cooperative but do only accept a certain percentage of WU offered to them (Edenhofer et al., 2015), expressed by the acceptance rate α.

2.5 Trust Metric

To overcome the problems of open, distributed MAS, that we just described and to optimise the global goal, we introduce a trust metric (Klejnowski et al., 2010),

justifying the name *Trusted* Desktop Grid. Agents get *ratings* for their actions from their particular interaction partners, representing the amount of trust earned through this interaction.

An agent a has multiple ratings with values between -1 and 1 that it gets from other agents. The amount of ratings k is limited to implement oblivion. The global average of all ratings for one single agent is called *reputation*. For further details see (Kantert et al., 2015). These ratings allow to make estimations about the future behaviour of an agent, based on its previous actions. In our system, agents get a positive rating, if they work for other agents and a negative rating, if they reject or cancel work requests (Klejnowski, 2014).

As a result, we can isolate malevolent agents and minimise the negative impact of malicious agents (cf. robustness in Section 2.3).

2.6 Trust Communities

To further increase the robustness and performance of the system, we introduced an agent organisation called Trust Community (TC). A TC consists of various agents which greatly trust each other. The agents are aware of their membership. A so-called Trusted Community Manager (TCM) is elected by the agents. It maintains the TC, preserves and optimises the composition and stability of the system. It is capable of inviting members to the TC, excluding members from the TC or assigning roles to the members for administrative reasons (for further details see (Klejnowski, 2014)). The election of the TCM can either depend on the ID of the agent (the agent with the highest or lowest ID is elected for TCM), or the reputation (members with a high trust rating are more likely to be elected as TCM).

The advantage of a TC organisation is that agents can reduce security measures such as replication of WU and are able to gain a better speedup. Members of a TC can better resist attacks because they can always decide to just cooperate inside the community and ignore the environment. Attacks within the TC are managed by the TCM. It is capable of monitoring a subset of the agents and, eventually, excludes them in case of misbehaviour.

3 STRATEGIES

As described in Section 2.6, the TCM has the ability to monitor the TC members. Total surveillance is neither desirable nor possible in an open, distributed system with autonomous agents. Therefore, the TCM

Figure 1: The Round-Robin Strategy with $step\text{-}width = 2$ and $\mathcal{S} = 30\,\%$, i.e. 3 of 9 agents are chosen (represented by dark grey) and these agents are shifted by 2 each time step.

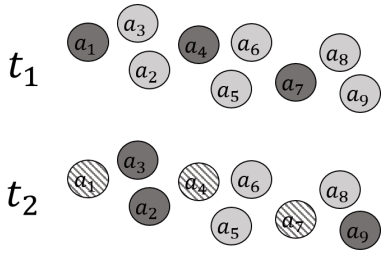


Figure 2: The Lottery-Based Strategy with $\mathcal{S} = 30\,\%$. At each timestep, 3 out of 9 agents are chosen (dark grey) as a disjoint set to the agents chosen before (striped) until every agent has been chosen once.

can only monitor a certain percentage of all TC members at each time, expressed by the *surveillance rate* $\mathcal{S}$. It can use different *surveillance strategies* presented in this section to choose these $\mathcal{S}\,\%$ of agents, which we refer to as the *chosen* agents.

Accusation-based Strategy - An agent can be accused by another agent in case of an incident, for example if it returns a fake result. The TCM is more likely to observe an agent with more accusations than an agent with less (or zero) accusations. For further details see (Edenhofer et al., 2015).

Reputation-based Strategy - Here, an agent is more likely to be monitored if its reputation is low in comparison to the other TC members' reputation. This concept can be realised by using, e.g. a roulette-wheel approach that considers the available reputation values.

Random-based Strategy - The agents that are observed are chosen randomly from all agents available in each time step. Each agent has the same probability to be selected.

Round-Robin-based Strategy - For this strategy, we consider all TC-members as elements in a sorted list. The chosen agents are the first $\mathcal{S}\,\%$ elements in the first time step. In the second time step, the chosen agents are the next $\mathcal{S}\,\%$ elements, shifted by *step-width*. This is illustrated in Figure 1: the chosen agents (dark grey) in t_1 are a_1, a_2 and a_3, in t_2 these are a_3, a_4 and a_5, because the selection is shifted by the $step\text{-}width = 2$.

Lottery-based Strategy - The agents that should be observed are randomly chosen in each time frame but

no repetition is allowed until all agents have been observed. In Figure 2, we can see that in t_1 agents a_1, a_4 and a_7 are selected (dark grey). In t_2, these agents cannot be chosen again (striped), a_2, a_3 and a_9 are chosen. In this example, the remaining three agents a_5, a_6 and a_8 would be the chosen ones in t_3, in t_4 any agent can be selected again.

The accusation-based and reputation-based strategies use additional knowledge about the agents (i.e. their previous incidents, respectively their reputation-value). We call these strategies *quality-based*. In contrast to that, we call the other three strategies *quantity-based*, since their success highly depends on the parameter assignment of $\mathcal{S}$(cf. Section 4). It should further be noted that the knowledge acquisition equals computational costs.

4 EVALUATION

We evaluated the performance of the surveillance strategies using two performance metrics: the *number of exclusions* (from a TC) and the *average residence time*. The *residence time* of an agent a (t_a^{res}) is the time the agent spends inside the TC (see Equation 2), from joining the TC (t_{join}) to being excluded ($t_{exclusion}$).

$$t_a^{res} = t_{exclusion} - t_{join} \tag{2}$$

During our experiments, we iterated over the parameters listed in Table 1.

Table 1: Parameter for the evaluation.

parameter	abbreviation
incidents before exclusion	*ibe*
forgiveness	F
acceptance (rate)	α
surveillance (rate)	$\mathcal{S}$

ibe determines the number of incidents one agent can commit before being excluded from the TC. In most cases we use a value of 2 for this, because this "three strikes" approach is a compromise between giving autonomous agents the chance to improve their behaviour on the one hand and having a healthy TC on the other hand. All incidents have a timeout (in ticks) called *forgiveness* F. Our experiments compare $F = 250, 1000, 5000, 10000$. Lower values do not result in a significant number of exclusions, because the older incidents are forgiven too fast. Higher values result in some sort of saturation and add no value to the system – resulting in a decreased efficiency. The acceptance rate α of the Sloppy Agents (SLA) ranges from 50% to 90%, increased by 10. With lower values, the probability that an agent is invited into a TC is too low. A value of 100% would equal the strategy performed by Altruistic Agents (ALT).

As discussed in Section 3, we are interested in rather low surveillance rates. Most comparisons are using a $\mathcal{S}$ of 10% and 30%, though we will also have a look at a low $\mathcal{S}$ of 2%, 5% and 8%. All simulations lasted 200000 ticks with 100 agents each (30 SLA and 70 ALT). If an agent was excluded from a TC, it was blacklisted for the remaining runtime and therefore could not get excluded twice from the same TC. In most upcoming sections we only allowed for one TC during our experiments. Thereby we wanted to ensure a controlled environment. Nevertheless, we will discuss some results with multiple TC in Section 4.3.

For the evaluation, we used the *average of 40 runs* (respectively the average of 10 runs with multiple TC) of one setup, as well as the *standard-deviation*.

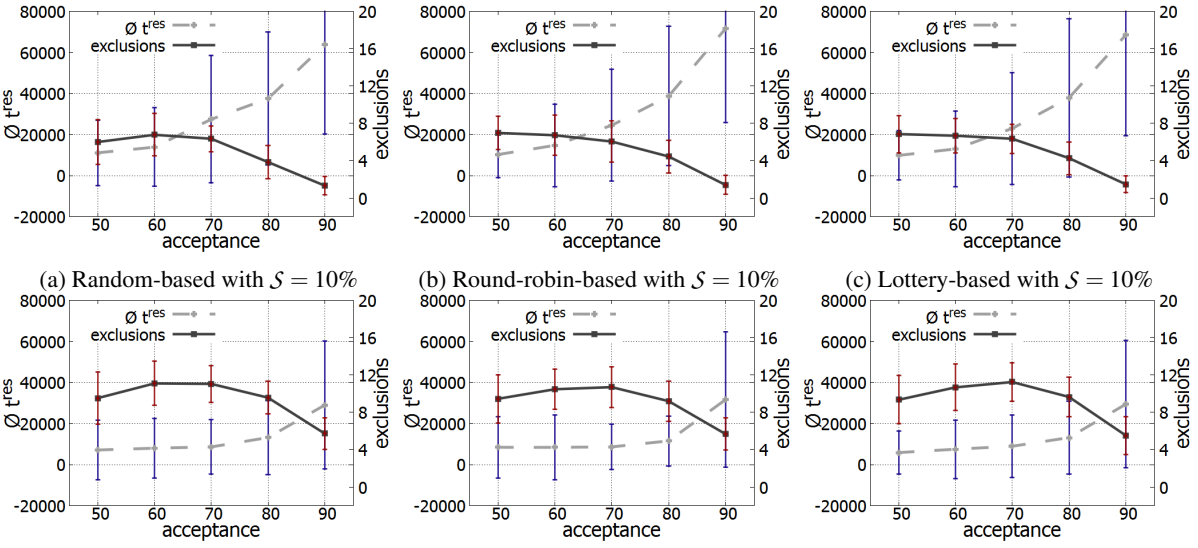
In Section 4.1, we show the results for the quantity-based strategies, the results for quality-based strategies are introduced in Section 4.2, followed by the results for multiple TC in Section 4.3.

4.1 Quantity-based Strategies

In this section, we compare the three quantity-based strategies, i.e. random-based, round-robin-based, and lottery-based, with each other. For the round-robin-based strategy, we use a *step-width* of 2. Using different *step-widths* does not yield a better performance. The three strategies (cf. Figure 3) have pretty much the same performance. Both the number of exclusions and $\varnothing t^{res}$ show the same trend and very similar values. If $\mathcal{S}$ is increased to 30%, more agents are excluded. Furthermore, $\varnothing t^{res}$ is lower than with $\mathcal{S} = 10\%$ (with smaller standard deviation). The similarity between the strategies exists with $\mathcal{S} = 10\%$ as well as with $\mathcal{S} = 30\%$. Given that these strategies perform that similar is due to their simplicity, we will only use the random-based strategy for comparison with the quality-based strategies in the following sections. Therefore, Figure 4 shows the performance of the random-based strategy at different F levels. At $F = 250$, there are nearly no exclusions and if there are some, $\varnothing t^{res}$ is pretty bad. The performance gets better the higher F.

4.2 Quality-based Strategies

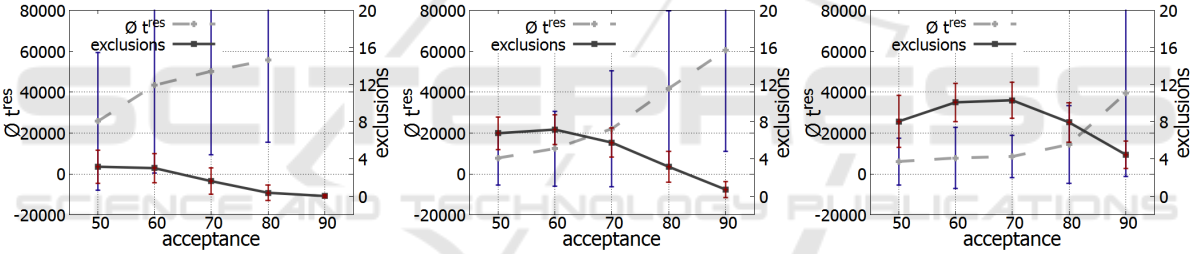
In this section, we show the evaluation of the accusation-based and the reputation-based strategy. Figures 6, 7 and 8 show the results for the reputation-based strategy. Figure 6 shows the performance with different values for F at different $\mathcal{S}$ levels: 10% in (a) - (d) and 30% in (e) - (h). Similar to the random-based



(a) Random-based with $\mathcal{S} = 10\%$ (b) Round-robin-based with $\mathcal{S} = 10\%$ (c) Lottery-based with $\mathcal{S} = 10\%$

(d) Random-based with $\mathcal{S} = 30\%$ (e) Round-robin-based with $\mathcal{S} = 30\%$ (f) Lottery-based with $\mathcal{S} = 30\%$

Figure 3: Comparison of the quantity-based strategies at $\mathcal{F} = 10000$. From left to right: results of random-, round-robin-, lottery-based strategy. The influence of $\mathcal{S}$ can be seen with $\mathcal{S} = 10\%$ (top) and $\mathcal{S} = 30\%$ (bottom). The x-axis shows the acceptance rate α, the left y-axis and the dashed line show $\varnothing t^{res}$, the right y-axis and the thicker solid line show the number of exclusions.



(a) Random-based, $\mathcal{F} = 250$, $\mathcal{S} = 30$. (b) Random-based, $\mathcal{F} = 1000$, $\mathcal{S} = 30$. (c) Random-based, $\mathcal{F} = 5000$, $\mathcal{S} = 30$.

Figure 4: Random-based strategy with $\mathcal{S} = 30$, $ibe = 2$ and $\mathcal{F} = 250, 1000, 5000$. The x-axis shows the acceptance rate α, the left y-axis and the dashed line show $\varnothing t^{res}$, the right y-axis and the thicker solid line show the number of exclusions.

strategy the performance increases with increasing $\mathcal{F}$. Be aware that $\varnothing t^{res}$ is not meaningful, if the number of exclusions is almost zero (cf. for example with Figure 6 (a)). If $\mathcal{F}$ is further increased to 10000, the outcome stagnates. Changing $\mathcal{S}$ from 10% to 30% does not yield a huge improvement in performance. If we lower $\mathcal{S}$ (cf. Figure 7), we eventually see a drop in performance at $\mathcal{S} = 2$. An increase of the *ibe* value creates a lower outcome (cf. Figure 8).

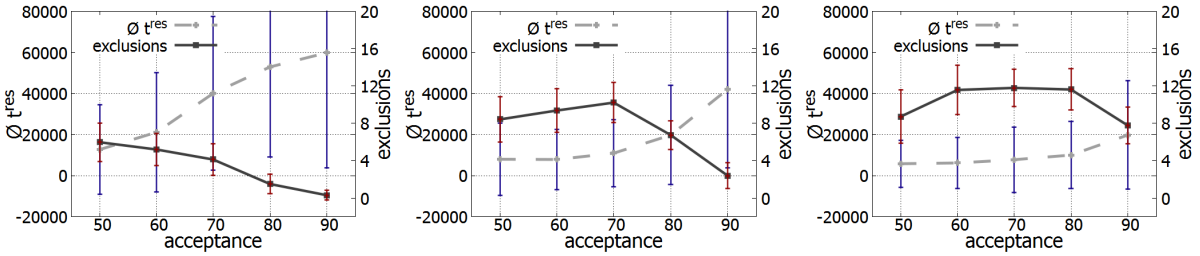
For the accusation-based strategy we show the results for different values for $\mathcal{F}$ at $\mathcal{S} = 10$. The strategy shows very similar values to the reputation-based strategy, while showing the same trends if varying the parameters.

4.3 Evaluation With Multiple TC

During the above mentioned evaluations we only allowed for one TC. In Figure 9, we show the results of the accusation- and reputation-based strategy, if multiple TC are allowed. The trends – with different parameter assignments – already described remain valid. If we compare the values of the accusation-based strategy to those of the reputation-based one, we can see that the results of the reputation-based strategy are significantly better.

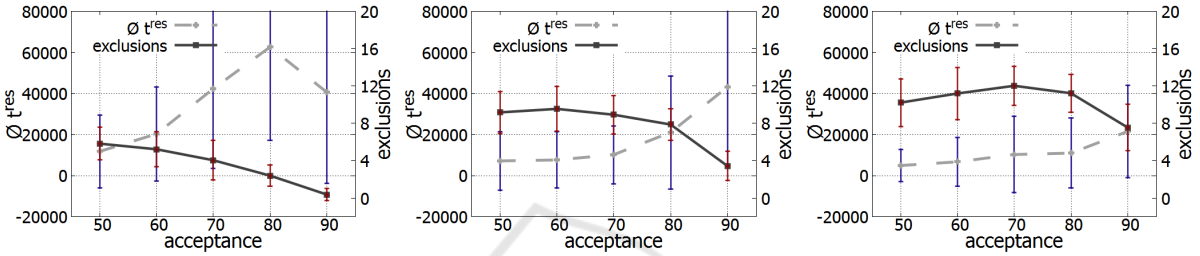
5 DISCUSSION

As mentioned in Section 4, our experiments iterated over the parameters listed in Table 1. As we can see

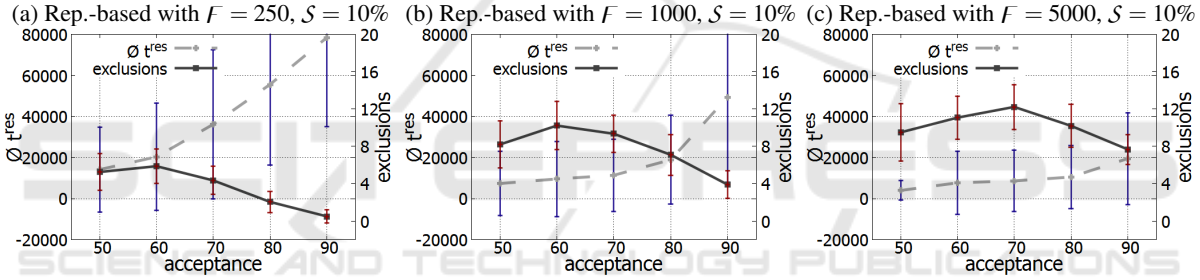


(a) Accusation-based with $F = 250$ (b) Accusation-based with $F = 1000$ (c) Accusation-based with $F = 5000$

Figure 5: Comparison of the accusation-based strategy's performance with different values for F (left to right) at $S = 10$. The x-axis shows the acceptance rate α, the left y-axis and the dashed line show $\varnothing t^{res}$, the right y-axis and the thicker solid line show the number of exclusions.

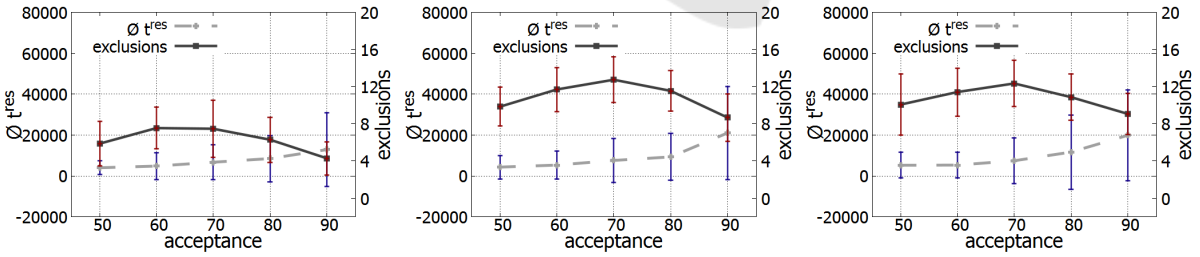


(a) Rep.-based with $F = 250$, $S = 10\%$ (b) Rep.-based with $F = 1000$, $S = 10\%$ (c) Rep.-based with $F = 5000$, $S = 10\%$



(d) Rep.-based with $F = 250$, $S = 30\%$ (e) Rep.-based with $F = 1000$, $S = 30\%$ (f) Rep.-based with $F = 5000$, $S = 30\%$

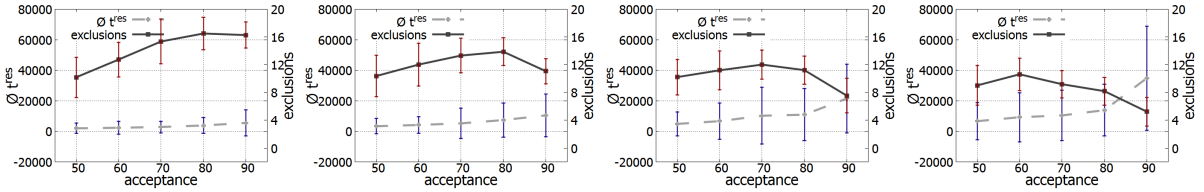
Figure 6: Comparison of the reputation-based strategy's performance with different values for F (left to right) at different S levels: (a) - (d) 10%, (e) - (h) 30%. The x-axis shows the acceptance rate α, the left y-axis and the dashed line show $\varnothing t^{res}$, the right y-axis and the thicker solid line show the number of exclusions.



(a) Reputation-based with $S = 2\%$ (b) Reputation-based with $S = 5\%$ (c) Reputation-based with $S = 8\%$

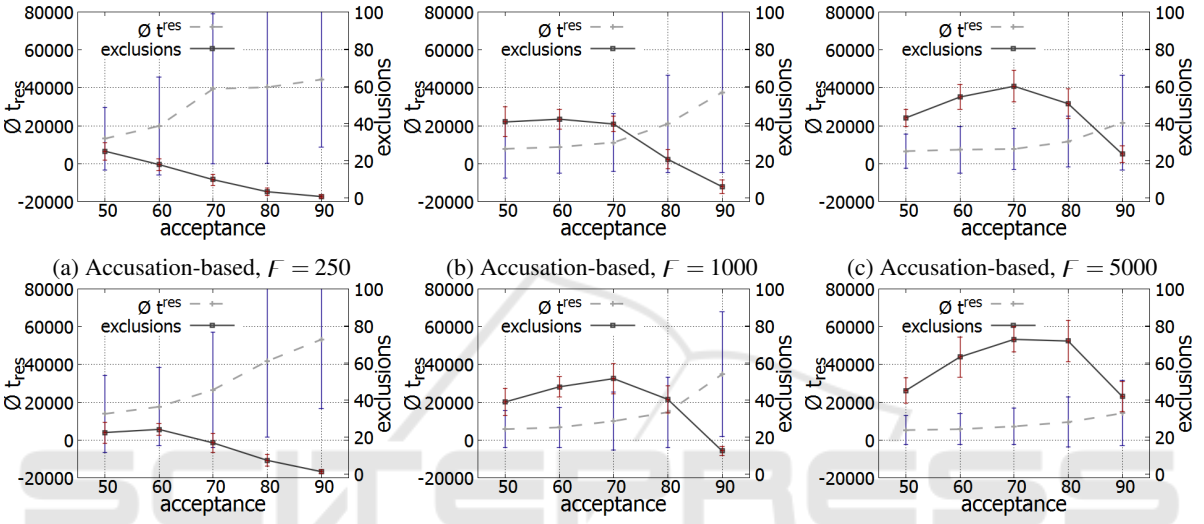
Figure 7: Reputation-based strategy with $F = 10000$ at different S levels: $2\%, 5\%, 8\%$ (left to right). The x-axis shows the acceptance rate α, the left y-axis and the dashed line show $\varnothing t^{res}$, the right y-axis and the thicker solid line show the number of exclusions.

in all figures, the performance of the strategies depends on the value for α. An increasing α has two major effects: First, the number of SLA in the TC gets higher. This simply leads to more exclusions. Second, the higher α, the less incidents. This means, the time between two incidents of the same agent increases. Eventually, this time is too long in relation to the current F. Therefore, the older incidents are



(a) Rep.-based with $ibe = 0$ (b) Rep.-based with $ibe = 1$ (c) Rep.-based with $ibe = 2$ (d) Rep.-based with $ibe = 3$

Figure 8: Comparison of the reputation-based strategy's performance with $F = 5000$, $\mathcal{S} = 10$ and different *ibe* values: $0, 1, 2, 3$ (left to right). The x-axis shows the acceptance rate α, the left y-axis and the dashed line show $\varnothing t^{res}$, the right y-axis and the thicker solid line show the number of exclusions.



(a) Accusation-based, $F = 250$ (b) Accusation-based, $F = 1000$ (c) Accusation-based, $F = 5000$

(d) Reputation-based, $F = 250$ (e) Reputation-based $F = 1000$ (f) Reputation-based $F = 5000$

Figure 9: Comparison of the accusation-based ((a) - (c)) and the reputation-based strategy ((d) - (f)) with different values for F (left to right: $250, 1000, 5000$) at $\mathcal{S} = 10$, if multiple TC are allowed. The x-axis shows the acceptance rate α, the left y-axis and the dashed line show $\varnothing t^{res}$, the right y-axis and the thicker solid line show the number of exclusions.

already forgiven when the agent commits another violation. This second effect seems to dominate whenever the performance drops although α is increasing (for example at around 70% in Figure 4 (c)).

In Figures 4, 5, 6, and 9, we compare the performance for varying F. The results have in common that at $F = 250$ the performance is pretty bad, but progressively getting better with increasing F. This, again, is due to F being too small: when the last incident, which is necessary to exclude an agent, is observed, the first incident is already forgiven. For the same reasons the performance gets worse with higher *ibe* (cf. Figure 8): the higher the value of *ibe*, the longer F must be. In case of increasing F, the outcome stagnates at around $F \approx 5000$. A higher F has no additional benefit, because all agents are already excluded at lower F values.

Additionally, we showed the impact of different $\mathcal{S}$ levels in Figures 3, 6, and especially 7. It is plausible that the performance gets better with higher $\mathcal{S}$, particularly resulting in a higher number of exclusions. If $\mathcal{S}$ is too low, it takes too long to repeatedly observe every SLA and consequently, again, F is too low. The outcome stagnates with increasing $\mathcal{S}$, because there are no more agents to exclude. This stagnation arises at different $\mathcal{S}$ for the quantity-based and quality-based strategies. This is due to the way these groups of strategies choose the agents they observe: the quantity-based strategies choose independently from the agent's identity. This means, the $\mathcal{S}\%$ agents are chosen from the total number of agents of the TC. In contrast, if using a quality-based strategy, the $\mathcal{S}\%$ agents are primarily chosen from the SLA subset of agents.

Comparing the accusation- with the reputation-based strategy is especially interesting if we allow for multiple TC (cf. Section 4.3 and Figure 9). While the results are pretty similar if only one TC is allowed (cf. Figures 5 and 6 (a) - (c)), we can see a significant advantage of the reputation-based strategy over the accusation-based one in regards to both the number of exclusions and $\varnothing t^{res}$. This is particularly

true for higher α. If multiple TC are allowed, an excluded agent cannot join the same TC again, but is allowed to join any other (this is also the reason why there are more exclusions, considering the changed value range in Figure 9). To explain why reputation-based performs better than accusation-based, we have to consider the peculiarities of these two strategies. Both use a form of history when judging which agents should be observed: the accusation-based strategy is more likely to observe agents which accumulated more accusations in the past than agents that did not do so. The reputation-based strategy is more likely to observe agents with a low reputation (relative to the other TC members). If an agent is excluded from a TC, its accusations are reset: the TCM of the new TC, where this agent becomes a member, does not know about former accusations in other TC. In contrast to that, when the TCM uses the reputation-based strategy, there is already some information available for the TCM in form of the agent's reputation. Therefore, from the moment the agent enters the new TC, it is already more likely to be observed. Thus, the results focussing on the number of exclusions as well as $\varnothing t^{res}$ of the reputation-based strategy are superior to those of the accusation-based strategy.

6 RELATED WORK

Our application scenario described in Section 2 is a Desktop Grid System. We model our grid nodes as agents, which can be seen as black boxes. Thereby, we cannot observe the internal state. Thus, their actions and behaviour can only be predicted with uncertainty (Hewitt, 1991). Our TDG supports Bag-of-Tasks application (Anglano et al., 2006). A classification and taxonomy of Desktop Grid Systems can be found in (Choi et al., 2007), respectively (Choi et al., 2008).

Desktop Grid Systems are used to share resources between multiple administrative authorities. One example for a peer-to-peer based system is the ShareGrid Project (Anglano et al., 2008). A second approach is the Organic Grid, a peer-to-peer based approach with decentralised scheduling (Chakravarti et al., 2004). Compared to our system, these approaches assume benevolence (Wang and Vassileva, 2004), i.e. that there are no malicious agents participating and misbehaving. Another approach is the open source Berkeley Open Infrastructure for Network Computing Project (BOINC) (Anderson and Fedak, 2006) or XtremWeb (Fedak et al., 2001), which aims at setting up a Global Computing application and "harvest[ing] the idle time of Internet connected computers which may be widely distributed across the world, to run a very large and distributed application" with an ad-hoc verification process for participating computers. We introduce a trust metric (see Section 2.5) with a reputation system to cope with the problem of misbehaving agents. A panoramic view on computational trust in Multi Agent Systems can be found in (Ramchurn et al., 2004), (Castelfranchi and Falcone, 2010), or (Sabater and Sierra, 2005). Sabotage-tolerance and distributed trust management in Desktop Grid Systems was evaluated in (Domingues et al., 2007). Here, mechanisms for sabotage detection are presented, but proposed for a paradigm of volunteer-based computing. Trust is also used in other disciplines such as philosophy (Karlins and Abelson, 1959), psychology (Hume, 1739), or sociology (Buskens, 1998).

7 CONCLUSION

In this paper, we presented several strategies to find and exclude malicious or bad behaving agents from open, distributed multi-agent systems. We have to differentiate between quantity-based strategies (such as the random-based, the round-robin-based, and the Lottery-based strategy) and quality-based strategies (like the accusation-based and the reputation-based strategy). To show the advantages and disadvantages of these strategies, we implemented them in our application scenario, the Trusted Desktop Grid (TDG). In several evaluations we showed the influence of the parameters *acceptance rate* (how many work units does one agent accept?), *surveillance rate* (how much budget do we have for surveillance, how many agents can we observe within a certain period of time?), the *forgiveness* (when will a former incident be forgotten?), as well as the *incidents before exclusion*. We found out that in most cases (especially at low surveillance levels) the quality-based strategies perform better than the quantity-based strategies. For future work we plan to further improve the TDG and the surveillance in multi-agent systems, e.g. by establishing a dynamic surveillance. This means, the choice of the current strategy used for surveillance will depend on a dynamically changing surveillance budget to make the system more realistic and life-like.

REFERENCES

Anderson, D. and Fedak, G. (2006). The Computational and Storage Potential of Volunteer Computing. In *Proc. of CCGRID 2006*, pages 73–80. IEEE.

Anglano, C., Brevik, J., Canonico, M., Nurmi, D., and Wolski, R. (2006). Fault-aware Scheduling for Bag-of-Tasks Applications on Desktop Grids. In *Proc. of GRID 2006*, pages 56–63. IEEE.

Anglano, C., Canonico, M., Guazzone, M., Botta, M., Rabellino, S., Arena, S., and Girardi, G. (2008). Peer-to-Peer Desktop Grids in the Real World: The ShareGrid Project. *Proc. of CCGrid 2008*, pages 609–614.

Buskens, V. (1998). The Social Structure of Trust. *Social Networks*, 20(3):265 – 289.

Castelfranchi, C. and Falcone, R. (2010). *Trust Theory: A Socio-Cognitive and Computational Model*, volume 18. John Wiley & Sons.

Chakravarti, A. J., Baumgartner, G., and Lauria, M. (2004). Application-Specific Scheduling for the Organic Grid. In *Proc. of GRID 2004 Workshops*, pages 146–155, Washington, DC, USA. IEEE.

Choi, S., Buyya, R., Kim, H., and Byun, E. (2008). A Taxonomy of Desktop Grids and its Mapping to State of the Art Systems. Technical report, Grid Computing and Dist. Sys. Laboratory, The University of Melbourne.

Choi, S., Kim, H., Byun, E., Baik, M., Kim, S., Park, C., and Hwang, C. (2007). Characterizing and Classifying Desktop Grid. In *Proc. of CCGRID 2007*, pages 743–748.

Domingues, P., Sousa, B., and Moura Silva, L. (2007). Sabotage-Tolerance and Trustmanagement in Desktop Grid Computing. In *Future Gener. Comput. Syst. 23, 7*, pages 904–912.

Edenhofer, S., Stifter, C., Jänen, U., Kantert, J., Tomforde, S., Hähner, J., and Müller-Schloer, C. (2015). An Accusation-Based Strategy to Handle Undesirable Behaviour in Multi-Agent Systems. In *2015 IEEE Eighth International Conference on Autonomic Computing Workshops (ICACW)*.

Fedak, G., Germain, C., Neri, V., and Cappello, F. (2001). Xtremweb: A Generic Global Computing System. In *Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on*, pages 582–587.

Hewitt, C. (1991). Open Information Systems Semantics for Distributed Artificial Intelligence. *Artificial intelligence*, 47(1):79–106.

Hume, D. (1739). *A Treatise of Human Nature*. (Oxford Clarendon Press.

Kantert, J., Edenhofer, S., Tomforde, S., and Müller-Schloer, C. (2015). Representation of Trust and Reputation in Self-Managed Computing Systems. In *IEEE 13th International Conference on Dependable, Autonomic and Secure Computing, DASC 2015*, pages 1827–1834, Liverpool, UK. IEEE.

Karlins, M. and Abelson, H. (1959). *Persuasion: How Opinions and Attitudes are Changed*. Springer Pub. Co.

Klejnowski, L. (2014). *Trusted Community : A Novel Multiagent Organisation for Open Distributed Systems*. PhD thesis, Leibniz Universität Hannover.

Klejnowski, L., Bernard, Y., Hähner, J., and Müller-Schloer, C. (2010). An Architecture for Trust-Adaptive Agents. In *Proc. of SASO Workshops*, pages 178–183.

Müller-Schloer, C., Schmeck, H., and Ungerer, T. (2011). *Organic Computing - A Paradigm Shift for Complex Systems*. Springer.

Ramchurn, S. D., Huynh, D., and Jennings, N. R. (2004). Trust in Multi-agent Systems. *Knowl. Eng. Rev.*, 19(1):1–25.

Sabater, J. and Sierra, C. (2005). Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60.

Steghöfer, J.-P. and Reif, W. (2012). Die Guten, die Bösen und die Vertrauenswürdigen Vertrauen im Organic Computing. *Informatik-Spektrum*, 35(2):119–131.

Wang, Y. and Vassileva, J. (2004). Trust-Based Community Formation in Peer-to-Peer File Sharing Networks. In *Proc. on Web Intelligence*, pages 341–348.