Journal of King Saud University – Computer and Information Sciences (2016) 28, 470–480

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Trust-based Service-Oriented Architecture

Zainab M. Aljazzaf [a,*], Miriam A.M. Capretz [b], Mark Perry [c]

[a] *Department of Information Science, Kuwait University, Kuwait*
[b] *Department of Electrical and Computer Engineering, Western University, Canada*
[c] *School of Law, The University of New England, Armidale, NSW, Australia*

Received 17 May 2015; revised 28 October 2015; accepted 22 December 2015
Available online 22 April 2016

KEYWORDS

Trust;
Service-Oriented Architecture;
Service providers

Abstract Service-Oriented Architecture (SOA) is an architectural style in building Web applications based on services. In SOA, the lack of trust between different parties affects the adoption of such architecture. Because trust is an important factor in successful online interactions, it is a major criterion for service selection. In the context of online services and SOA, the literature shows that the field of trust is not mature. The definitions of trust and its essential aspects do not reflect the true nature of trust online. This paper proposes a comprehensive trust-based SOA solution based on an identified trust definition and its principles for selecting services based on their trustworthiness. In particular, SOA is extended and a new component, the trust framework, which is responsible for the trust process, is added to the architecture. Consequently, its components are identified and built. The trust-based SOA is implemented through experiments and scenarios.

1. Introduction

The development of distributed software requires the interaction of services from different Web service providers. Service-Oriented Computing (SOC) is "a computing paradigm that utilizes services as fundamental elements to support rapid, low-cost development of distributed application in heterogeneous environments" (Papazoglou and Georgakopoulos, 2008). A service is "a discrete unit of business functionality that is made available through a service contract" (Rosen et al., 2008). Specifically, a distributed application may be composed of global services with different properties provided by different organizations. In this environment, the development of trust is challenging.

To realize the potential of SOC, Service-Oriented Architecture (SOA) should be developed. SOA is "a framework for integrating business processes and supporting IT infrastructure as secure, standardized components – services – that can be reused and combined to address changing business priorities" (Bieberstein et al., 2005).

SOA has a significant impact on the way software systems are built. Although there have recently been reports that SOA adoption rates are dropping and that "SOA is dead", Forrester Group reported that SOA adoption is increasing across all of its vertical-industry groups (Lewis, 2013). Gartner Group reports that 50 percent of new vital operational applications and business processes were designed around SOA in 2007

* Corresponding author.
E-mail addresses: dr.zainab@ku.edu.kw (Z.M. Aljazzaf), mcapretz@uwo.ca (M.A.M. Capretz), mperry21@une.edu.au (M. Perry).
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2015.12.003
1319-1578

and that adoption will increase to more than 80 percent by 2010.

Fig. 1 illustrates the relationship between SOA roles and operations. There are three interaction roles in SOA: the *service provider*, which owns, implements, and controls access to the services; a *service requestor*, which is an application, service, or client who is searching and invoking a service; and a *service broker* that groups all of the services together and maintains a registry of available services (Papazoglou and Georgakopoulos, 2008). A service registry is a directory in which the services are published by the providers and searched by the requestors (Papazoglou, 2012).

Moreover, there are three operations within SOA (Papazoglou, 2012). In the *publish operation*, service providers publish their services into the registry. In the *find operation*, requestors search and find services from the service registry. Finally, in the *bind operation*, requestors invoke services at run time using the technical information provided in the WSDL file to bind to the services.

To build a service-oriented application, requestors can select services from different providers on the Internet. Because there are many services with similar functionalities, requestors need to differentiate between them. The only differentiating factor between similar services may be their non-functional properties, which can be considered criteria for service selection. As a non-functional property, trust has been used as a criterion for service selection (Dragoni, 2009; Huhns and Singh, 2005; Kalepu et al., 2003; Azarmi et al., 2012; Kim and Doh, 2013).

Trust is *"the willingness of the trustor to rely on a trustee to do what is promised in a given context, irrespective of the ability to monitor or control the trustee, and even though negative consequences may occur"* (Aljazzaf et al., 2010). A service requestor, or *trustor*, may select a service from a service provider, *trustee*, based on their trustworthiness. Thus, trust can help requestors in their service selection. In addition, some service providers provide poor services or intentionally offer services that are not consistent with their promises (Jin-Dian et al., 2005). Thus, it is necessary to determine the trustworthiness of services and to select a trustworthy service. Moreover, trust is a less expensive approach for service selection than monitoring or Service Level Agreements (SLA) (Wang and Vassileva, 2007).

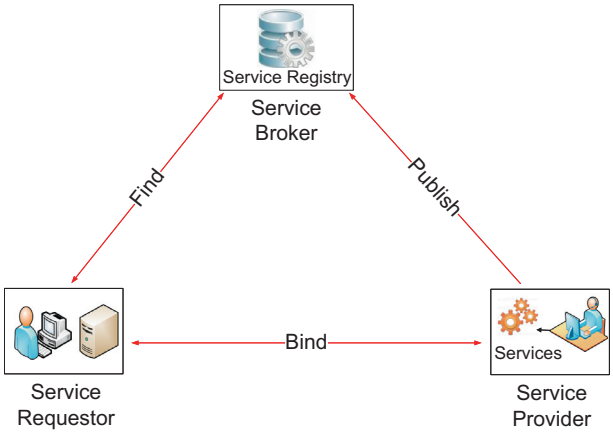

Fig. 1 Service Oriented Architecture (SOA) (Papazoglou, 2012).

There are different principles that reflect the core nature of trust. These principles consider trust aspects and identify requirements for establishing a comprehensive and concrete solution for trust (Aljazzaf et al., 2010; Daignault et al., 2002). Some principles include the following: trust and risk are related, the trust development phases should be considered, trust is dynamic, trust depends on identity, trust is based on information.

Although SOA continues to be broadly adopted, there has been surprisingly little interest in building complete solutions that facilitate trust-based service selection. Such a complete solution is required and should be described in detail. Accordingly, there is a need to extend SOA to support trust, and such extension includes building a unified framework and model of trust considering trust definition and trust principles that incorporate many trust aspects, can be easily extendible, and resolve different trust challenges.

The rest of the paper is presented as follows. Section 2 presented the related work. The proposed trust-based SOA is introduced in Section 3. Section 4 covers the proposed trust framework and discusses its components. The experiment is presented in Section 5, and its evaluation is discussed in Section 6. Finally, Section 7 presents the conclusion and future work.

2. Related work

Research on trust has attracted a great deal of attention in SOC. However, the literature about trust on SOA is still immature. Existing solutions for trust in SOA, including trust frameworks and models, are not built based on a standard definition of trust and do not follow principles that reflect the core nature of trust.

Existing OASIS WS-Trust and WS-Security standards ensure hard security mechanisms of SOA applications. However, trust is not covered as an essential service that reflects the nature of trust as we have defined it.

Moreover, Azarmi et al. (2012) provide a solution for end-to-end security auditing in SOA and maintaining a dynamic trust among services. The trust broker specifies the various levels of trust (Certified, Trusted, or Untrusted) and uses a reputation-based system to preserve the trust levels based on several criteria, including the history of previous interactions. Kim and Doh (2013) build a framework and add a trust mediator as a QoS broker for governing the trust process. The authors propose a trust management model that supports service discovery and selection based on QoS, specifically utilizing security, trust, and reputation. However, the authors define trust as a QoS, and their mechanism uses consumers' feedback, which is highly human dependent and therefore error-prone. Liu et al. (2014) introduce a Web Service evaluation model by leveraging trust as an approach. They incorporate a trust management module into the standard SOA and then transform a Web Service network to a small-world network. However, their framework is built upon only a trust management module and is based on small-world networks. Many researchers have studied security certification, which is aimed at increasing the confidence of the clients by satisfying their security requirements (Anisetti et al., 2012; Anisetti et al., 2013; Katopodis et al., 2014; Kaluvuri et al., 2013; Cimato et al., 2013).

Regarding trust frameworks in the literature, Townend et al. (2012) analyze the concept of provenance and discuss how the formation of personalized provenance recording and retrieval systems can be used to increase the utility of data and produce user trust in service-based systems. A generic framework is developed to enable the creation of provenance for confident decision making. Gan et al. (2010) proposes a service-oriented trust management framework for E-commerce systems. It consists of an authentication center, evaluation, reference, update, and history controllers. Chen et al. (2008) introduce the SCTRUST model to evaluate trust for services. SCTRUST registers, sorts, and queries services as well as querying and updating the trust rates. However, neither of the frameworks provides a comprehensive trust solution.

Regarding the extensions of SOA to support trust, some studies use the regular SOA model where the ranking or trust process is conducted in the service broker by the service registry (Chen et al., 2008) or by an additional component added to the service broker (Kim and Doh, 2007; Liu et al., 2004; Kim and Doh, 2013). Other studies (Cao et al., 2009; Ran, 2003) use the regular SOA model and add an auxiliary component outside of the service broker. The three roles of SOA are connected to the auxiliary component responsible for Web Service evaluation based on QoS and user preference (Cao et al., 2009) or QoS certifying and verifying (Ran, 2003). In Kalepu's extension (Kalepu et al., 2003), there are two verity calculators one on the service broker side and one on the end user side to calculate local and global rankings. In addition, there is an interceptor component between the service broker and the end user to measure the SLA parameter values delivered at the end of each service invocation and to send the values to both the end user and the service broker for verity calculations.

Based on these variations of SOA extensions, this work aims to derive a suitable way of extending SOA to support trust. Chen et al. (2008) provides an extension that modifies the basic roles of SOA, whereas the other extensions (Kim and Doh, 2007; Liu et al., 2004; Cao et al., 2009; Ran, 2003; Kalepu et al., 2003) add new roles to SOA. Between these two extension approaches, the latter method is preferable because it does not require modification of the basic roles of SOA, which facilitates the deployment of the component as a service to the SOA environment.

The literature reflects little interest in building a comprehensive solution that facilitates trust-based service selection, considers trust principles, and resolves different trust challenges.

3. Trust-based SOA

This section presents the SOA extension for supporting trust. Fig. 2 illustrates the proposed trust-based SOA. The service broker is considered a Trusted Third Party (TTP).

The SOA extension supports trust by including a trust framework, interfaces, and additional link interactions. The *trust framework*, called the Total Trust Evaluator Framework (ToTEF), is added into the service broker as a service. ToTEF is responsible for conducting the trust process. This includes rating services and service providers and conducting trust management. ToTEF also maintains its own rating registry, in addition to the service registry in the service broker.

Because trust is dynamic, it can be divided into three development phases: *trust building*, *stabilizing trust*, and *dissolution*, where trust is formed, already exists, and ends, respectively (Kautonen and Karjaluoto, 2008). Most studies assume a system where trust already exists, as in the trust-stabilizing phase, but it is important to consider the "trust bootstrapping," in which trust rates are initialized for new entities, as in the trust building phase (Aljazzaf et al., 2011a). Accordingly, SOA should examine all three trust development phases.

As a general overview, ToTEF establishes trust for services and service providers starting with trust bootstrapping. First, during the Publish operation, ToTEF obtains a publish request from a provider. Subsequently, ToTEF obtains the service description and trust information, identifies the service

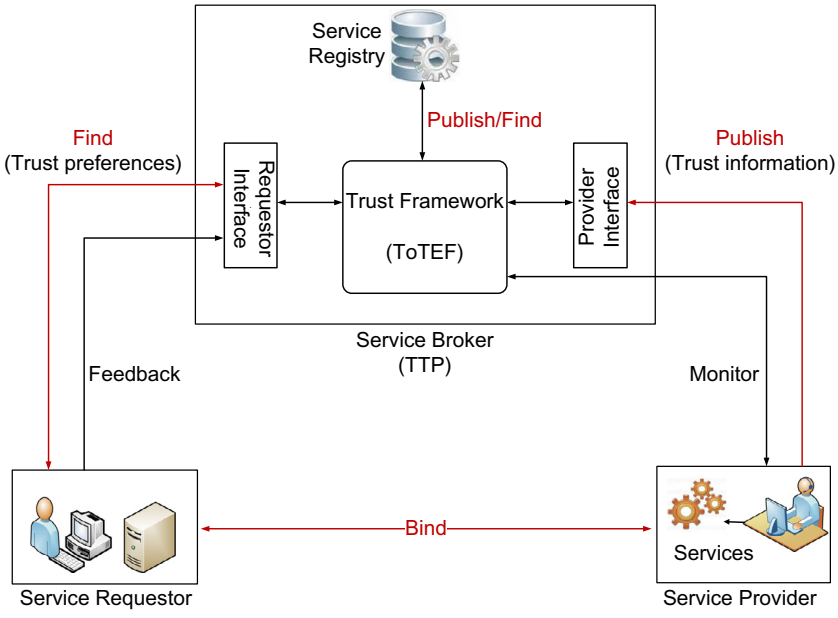

Fig. 2 Trust-based SOA.

and service provider, and then publishes the service in the service registry and stores the trust information in the rating registry.

Then, the trust rate of the trust information is evaluated, which in turn is used to calculate the trust rates for services and service providers. The evaluated rates will be saved in the rating registry to be used by service requestors at the Find time.

To facilitate this, the trust-based SOA needs to have a *provider interface* and a *requestor interface*. The provider interface allows service providers to publish their services, provide their trust information, and view their ratings and service ratings to improve their services and build their Quality of Business (QoBiz) (Moorsel, 2001). The requestor interface allows requestors to search for services and enables service consumers to provide feedback on the services they have used.

Subsequently, the additional *link interactions* include a monitor link and feedback link. Monitoring is used as a technique for trust bootstrapping, thus, the monitor link between ToTEF and service providers monitors the registered services. Alternatively, the feedback link allows requestors to provide their feedback about services and service providers, which helps in the stabilizing trust phase.

Service providers need to publish their services along with the trust information. In our previous work (Aljazzaf et al., 2011b), we identified the services and service providers trust information, known as Trust Metrics (TMs). Table 1 shows a set of TMs, as follows:

- Service Trust Metrics (STM): STMs are services' trust information, which include the trust information about services and their properties. STMs are categorized as follows:
  - Objective Service Trust Metrics (OSTM), which are TMs that have a formula for measurement and the monitoring approach can be used for measuring them such as response time OSTM ($OSTM_r$).
  - Subjective Service Trust Metrics (SSTM), which are the TMs that are difficult to measure and require a different approach for quantifying and measuring them such as security SSTM ($SSTM_{sec}$).
- Provider Trust Metrics (PTM), which are service providers' trust information, such as security PTM (PTM_{sec}), competence PTM (PTM_{comp}), and honesty PTM (PTM_{hons}).

Table 1 The Trust Metrics (TMs).

TM		
STM	OSTM	$OSTM_e$: Execution time OSTM
		$OSTM_r$: Response time OSTM
		$OSTM_l$: Latency OSTM
		$OSTM_{thr}$: Throughput OSTM
	SSTM	$SSTM_{rem}$: Remedies SSTM
		$SSTM_{sec}$: Security SSTM
		$SSTM_{prv}$: Privacy SSTM
		$SSTM_{pym}$: Payment satisfaction SSTM
PTM	STM	PTM_{rem}: Remedies PTM
		PTM_{sec}: Security PTM
		PTM_{prv}: Privacy PTM
	Provider's properties	PTM_{brand}: Brand name PTM
		PTM_{comp}: Competence PTM
		PTM_{hons}: Honesty PTM
	Clues	PTM_{wsite}: Website PTM
		PTM_{loc}: Physical location PTM

TM: Trust metric, STM : Service TM, PTM : provider TM.

With respect to TMs, ToTEF establishes trust for services (T_s) and trust for service providers (T_{pr}) starting with trust bootstrapping. First, in the Publish operation, ToTEF obtains the publish request from the provider interface along with TMs. ToTEF publishes the service in the service registry and stores the TMs in the rating registry.

The trust rates of the TMs (T_{TM}) are then evaluated. Finally, T_{TM} are used to calculate T_s and T_{pr}. The T_{TM}, T_s, and T_{pr} are saved in the rating registry. When ToTEF obtains the Find request from the requestor interface, it searches the service registry for services that match the functional properties. Then, ToTEF selects services that match the requestor's trust preferences from the rating registry. Accordingly, the services that satisfy the requestor's functional and trust preferences are returned to the requestor.

4. ToTEF: Total Trust Evaluator Framework

ToTEF is a unified trust framework that provides a comprehensive trust solution, because it is built according to the trust definition and trust principles to identify its main components. ToTEF contains the necessary components for trust bootstrapping, trust evaluation, trust management, and resolving different trust challenges, such as unfair feedback, trust bias, and culture differences.

ToTEF consists of three stages: the pre-processing stage, processing and evaluation stage, and post-processing stage, as shown in Fig. 3. Each stage consists of several components that perform different functions. In addition, ToTEF has a rat-

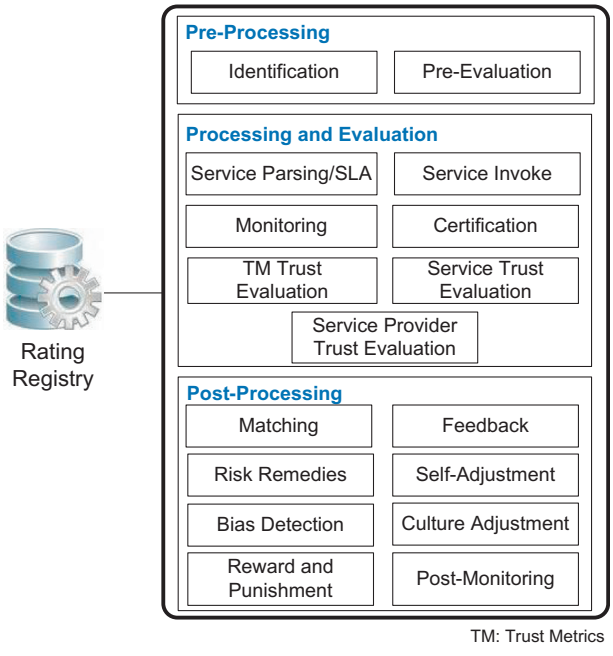

Fig. 3 ToTEF: Total Trust Evaluator Framework.

ing registry that stores the Trust Metrics (TMs) and the TMs rates (T_{TM}). It supports searching, matching, and selecting services based on requestors' trust preferences. The following explains ToTEF stages.

4.1. Pre-processing stage

Service providers publish services and TMs with the service broker. The pre-processing stage identifies and registers new services and service providers in the service registry, stores the TMs into the rating registry, and pre-evaluates services and providers. As depicted in Fig. 3, the pre-processing stage has two components: identification and pre-evaluation.

4.1.1. Identification

Trust depends on identity; each service and service provider needs to have an ID. This component identifies the new services and service providers, assigns IDs to them, and publishes services in the service registry. This component also stores the services and service providers' IDs, along with their TMs, in the rating registry. If a service provider is already registered and identified, this component will assign an ID only to their new services that are registering.

4.1.2. Pre-Evaluation

The next stage, processing and evaluation stage, is responsible for trust bootstrapping and dynamic trust evaluation for TMs, services, and service providers. This may have a high overhead on the service broker, but pre-evaluation component reduces the overall overhead by *rating services based on their providers' rates*. If a provider is trustworthy, its services can be also considered trustworthy. In this case, the pre-evaluation component will assign the trust rates of the provider's new services as equal to providers' trust rates rather than running the next stage, the processing and evaluation stage. ToTEF will determine the extent to which a service provider should be trustworthy when assigning its rate to its newly-registered services as detailed by Aljazzaf et al. (2011a).

4.2. Processing and evaluation stage

This stage is in charge of trust bootstrapping and dynamic trust evaluation of the TMs, services, and service providers. This work proposes a number of dynamic approaches for trust bootstrapping, evaluation, and evolution. These approaches include monitoring, certification, and feedback from service consumers.

This stage parses services descriptions, invokes services, monitors and certifies TMs, and evaluates the trust rates of the TMs, services, and service providers. This stage involves seven components, as shown in Fig. 3: service parsing/SLA, service invocation, monitoring, certification, TM trust evaluation, service trust evaluation, and service provider trust evaluation.

4.2.1. Service parsing/SLA

The trust bootstrapping process necessitates a dynamic evaluation of services, which requires obtaining information necessary for the evaluation. Specifically, this component obtains the information about a service that is necessary for each of the subsequent components, invocation, monitoring, and certification, in performing their processes. The required service information can include the service operation, input and output parameters of the operation and their data types, binding information, and policies. Moreover, TMs could be obtained from either the service description, which requires extension, or from the SLA. In this work, the TMs are obtained from the provider interface.

4.2.2. Service Invocation

With information obtained during Service Parsing/SLA, such as services' operations and binding information, this component will invoke the associated services.

4.2.3. Monitoring

In this work, the monitoring approach (Zhengping et al., 2007; Zhang et al., 2010; Nguyen et al., 2010) is proposed as a method for trust bootstrapping and dynamic trust evaluation. In particular, the monitoring component monitors TMs that can be measured, which are the Objective Service TMs (OSTMs), such as execution time OSTM ($OSTM_e$). The collected information is stored in the rating registry and used by the subsequent components for trust evaluation.

The following presents the evaluation approach for monitoring the OSTM used in this research (Lee et al., 2003; Aljazzaf, 2015):

- Latency ($OSTM_l$): The Latency or network latency time of a service is "the round-trip Delay (RTD) between sending a request and receiving a response" (Lee et al., 2003).
- Execution Time ($OSTM_e$): The execution time of a service is the time taken by the service to execute and process its sequence of activities.
- Response Time ($OSTM_r$): The response time of a service is the time requires to process and complete a service request; the response time include the execution time and the latency. The following is the formula to evaluate the response time:

$$OSTM_r = OSTM_e + OSTM_l \tag{1}$$

- Throughput ($OSTM_{thr}$): The throughput of a service refers to the number of requests a service can process per unit of time. Throughput depends on the power of service machines and it is measured by sending many requests over period of time and count the number of respond. The following is the formula to evaluate the throughput:

$$OSTM_{thr} = \frac{\text{Number of requests}}{\text{time period}} \tag{2}$$

4.2.4. Certification

This component is responsible for certifying some TMs, such as security and privacy Subjective Service TMs (SSTM), based on the services' policies about such TMs. The certification and rating process for security and privacy is beyond the scope of this paper. However, security and privacy can be certified as detailed in (Anisetti et al., 2012; Anisetti et al., 2013; Katopodis et al., 2014; Kaluvuri et al., 2013; Cimato et al., 2013) or their ratings can be obtained from security and pri-

vacy rating systems (Mayer, 1990; El Yamany, 2009; Allison et al., 2009; Diego, 2011).

For example (Anisetti et al., 2012), security certification provides a security-enhanced service discovery and selection approach and enhances requestors' security requirements. In this case, the certification component certifies services' security properties, such as confidentiality and integrity. Subsequently, the matching component selects the service that matches the clients' security preferences.

4.2.5. TM Trust Evaluation

This component evaluates trust rates for TMs (T_{TM}). Ratings TMs are based on the published TMs by a service provider and collected by the monitoring and certification components during the processing and evaluation stage. Different trust approaches are proposed in this work to rate various TMs, which are presented in Table 2, as follows:

- *The trust rates of the Objective TMs, OSTMs (T_{OSTM})* are bootstrapped and evaluated using the monitoring approach. After OSTMs are published, they need to be collected using the monitoring approach. Then, the monitored OSTM and the published OSTM are compared and used to evaluate the T_{OSTM}.
- *The trust rates of the Subjective TMs, SSTMs (T_{SSTM})* are bootstrapped and evaluated using certification and feedback approaches. The bootstrapping approaches for some SSTM, such as payment satisfaction SSTM ($T_{SSTM_{pym}}$), are based totally in feedback approach and will have initial rates after receiving feedback from the service consumer about $SSTM_{pym}$. The trust rates of the remedies SSTM ($T_{SSTM_{rem}}$), security SSTM ($T_{SSTM_{sec}}$), and privacy SSTM ($T_{SSTM_{prv}}$) are based on the certification approach.
- *The trust rates of the Provider TMs, PTMs (T_{PTM})* are bootstrapped and evaluated based on the trust rates of their services' TMs (T_{STM}). In particular, the trust rates of the remedies PTM ($T_{PTM_{rem}}$), security PTM ($T_{PTM_{sec}}$), and privacy PTM ($T_{PTM_{prv}}$) are rated based on the $T_{SSTM_{sec}}$, $T_{SSTM_{prv}}$, and $T_{SSTM_{rem}}$ of their services. The trust rates of the website PTM ($T_{PTM_{wsite}}$) and physical location PTM ($T_{PTM_{loc}}$) are evaluated by service requestors, who can provide their feedback for PTM_{wsite} and PTM_{loc} after they utilize the services.

Table 2 Trust ratings of the TM (T_{TM}) and the evaluation approaches.

Trust rate of the TM (T_{TM})		Trust bootstrapping evaluation approach
T_{STM}	T_{OSTM_e}: Trust rate of Execution time OSTM	Monitoring
	T_{OSTM_r}: Trust rate of Response time OSTM	Monitoring
	T_{OSTM_l}: Trust rate of Latency OSTM	Monitoring
	$T_{OSTM_{thr}}$: Trust rate of Throughput OSTM	Monitoring
	$T_{SSTM_{rem}}$: Trust rate of Remedies SSTM	Certification
	$T_{SSTM_{sec}}$: Trust rate of Security SSTM	Certification
	$T_{SSTM_{prv}}$: Trust rate of Privacy SSTM	Certification
	$T_{SSTM_{pym}}$: Trust rate of Payment satisfaction SSTM	Feedback
T_{PTM}	$T_{PTM_{rem}}$: Trust rate of Remedies PTM	Based on $T_{SSTM_{rem}}$
	$T_{PTM_{sec}}$: Trust rate of Security PTM	Based on $T_{SSTM_{sec}}$
	$T_{PTM_{prv}}$: Trust rate of Privacy PTM	Based on $T_{SSTM_{prv}}$
	$T_{PTM_{brand}}$: Trust rate of Brand name PTM	Based on all T_{TM}
	$T_{PTM_{comp}}$: Trust rate of Competence PTM	Based on all T_{TM}
	$T_{PTM_{hons}}$: Trust rate of Honesty PTM	Based on all T_{TM}
	$T_{PTM_{wsite}}$: Trust rate of Website PTM	Feedback
	$T_{PTM_{loc}}$: Trust rate of Physical location PTM	Feedback

TM: Trust metric, STM : Service TM, PTM : provider TM.

The trust rates of the competence PTM ($T_{PTM_{comp}}$), honesty PTM ($T_{PTM_{hons}}$), and brand name PTM ($T_{PTM_{brand}}$) are evaluated through long-term interaction with the providers' services, as detailed by (Aljazzaf et al., 2011a).

4.2.6. Service Trust Evaluation

The service trust evaluation component evaluates trust rate of services (T_s). T_s are based on the T_{TM} of their published TMs. Fig. 4 presents the UML activity diagram that explains the trust evaluation process for services. The service trust evaluation process starts with the trust bootstrapping process. Within this diagram, the framework evaluates and updates T_s based on the honesty of the service provider, as follows:

- *A provider is not honest*: If the service provider is not honest, the trust framework will start the trust bootstrapping process for the new services. The framework will monitor

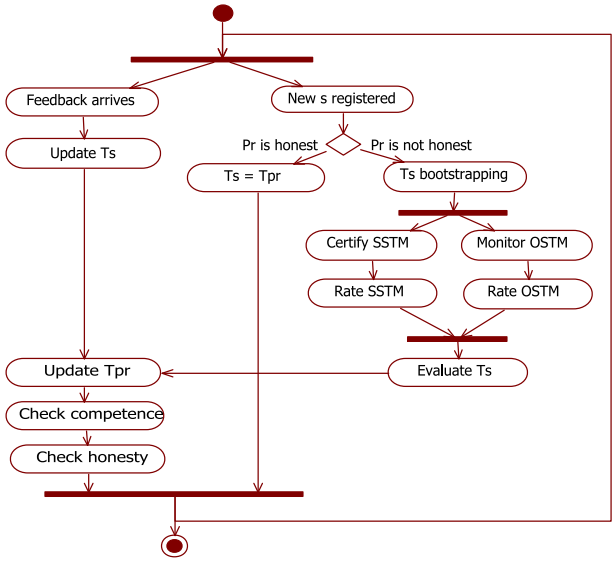

Fig. 4 Trust bootstrapping and rating services: activity diagram.

and rate OSTM, certify and rate SSTM, and then evaluate T_s.

- *A provider is honest*: If the provider is honest, the rates of its new services will equal the rate of their provider.

If feedback is returned for one or more of the Service's Trust Metrics (STMs), then T_{STM}, T_s, and trust rate of the provider (T_{pr}) are updated accordingly. Consequently, the trust mediator will check the competence and honesty of the provider.

4.2.7. Service Provider Trust Evaluation

This component evaluates the trust rate of service providers (T_{pr}) based on the trust rates of their services (T_s), the information from the service trust evaluation component. The UML activity diagram in Fig. 5 presents the trust bootstrapping process for service providers as follows: if a provider is new, T_s will be bootstrapped and the value is then assigned to the provider; i.e., $T_{pr} = T_s$. If the provider is not a new provider and is not honest, the framework will trust bootstrap the new service, evaluate the T_s, and then update T_{pr} accordingly ($T_{pr} \leftarrow Avg(T_{pr}, T_s)$). In addition, this component checks the competence and honesty of the provider and update the T_{PTM} if feedback is provided.

4.3. Post-processing stage

The post-processing stage contains various components that play significant roles in trust-based service discovery and trust management. It addresses different trust challenges in the literature, such as culture differences, unfair feedback, and bias detection. As shown in Fig. 3, the post-processing stage includes eight components: matching, feedback, risk remedies, self-adjustment, bias detection, culture adjustment, reward and punishment, and post-monitoring. This work addresses only the matching component as the other components are beyond the scope of this work, However, the function of each component is presented.

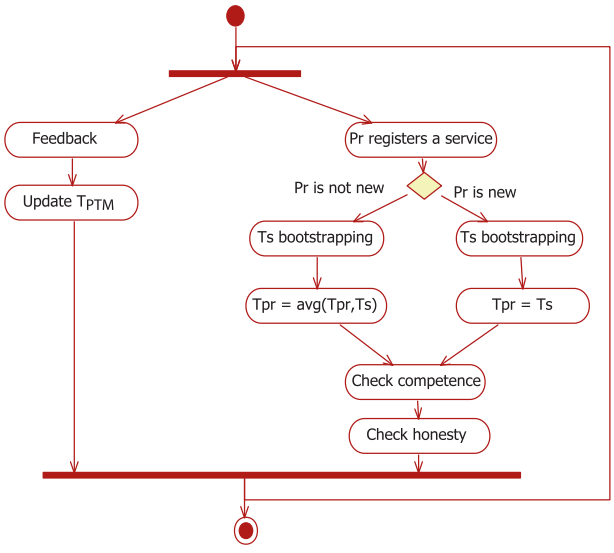

Fig. 5 Trust bootstrapping and rating service providers: activity diagram.

4.3.1. Matching

The matching component supports service selection based on the trust rates of services and service providers on requestors' trust preferences. Trust is subjective and context-specific. The concept of 'subjectivity' indicates that an entity's trust varies among different requestors, and the term 'context-specific' indicates that trust is diverse in various situations.

At the Find time, trust rates are evaluated based on the requestor's trust preferences and weighted as the average T_{TM} of the requestor's selected TMs, supporting the subjective and context-specific properties of trust. Specifically, the requestor selects a list of TMs, and the matching component evaluates the T_s for the services based that list such that:

$$T_s = Avg(T_{STM}) \tag{3}$$

For example, if a requestor requests a service based on $OSTM_e(s), OSTM_l(s), OSTM_{thr}(s)$, and $SSTM_{sec}(s)$ in addition to the service's functional property, then the matching component will evaluate the T_s based on the required TMs. Thus, if $T_{OSTM_e}(s) = 8.1, T_{OSTM_l}(s) = 6.5, T_{OSTM_{thr}}(s) = 7.2$, and $T_{SSTM_{sec}}(s) = 6$, then using Eq. 3:

$$\begin{aligned} T_s &= Avg[T_{OSTM_e}(s) + T_{OSTM_l}(s) + T_{OSTM_{thr}}(s) + T_{SSTM_{sec}}(s)] \\ &= Avg[8.1 + 6.5 + 7.2 + 6] = 6.95 \end{aligned}$$

The trust matching model also allows requestors to specify weights for the selected TMs. For example, a requestor may provide a weight of 60% for the $OSTM_e(s)$, 70% for the $OSTM_l(s)$, 100% for the $OSTM_{thr}(s)$, and 100% for the $SSTM_{sec}(s)$. This results in the following change to T_s:

$$\begin{aligned} T_s &= Avg[0.6 \times T_{OSTM_e}(s) + 0.7 \times T_{OSTM_l}(s) \\ &\qquad +1 \times T_{OSTM_{thr}}(s) + 1 \times T_{SSTM_{sec}}(s)] \\ &= Avg[0.6 \times 8.1 + 0.7 \times 6.5 + 1 \times 7.2 + 1 \times 6] = 5.65 \end{aligned}$$

4.3.2. Feedback

Service consumers may provide their feedback about the services and service providers to represent their satisfaction or dissatisfaction. Since consumers may provide unfair feedback, it is essential to impede such feedback. Hence, the feedback component is responsible for addressing the unfair feedback problem.

4.3.3. Risk remedies

Since trust and risk are related, it is important to provide remedies in case an unexpected event occurs. In addition to trusting services and providers, requestors need to trust service brokers, which are TTPs. Hence, this component supports risk remedies for service brokers.

4.3.4. Self-Adjustment

The self-adjustment component considers the dynamic nature of trust and is responsible for trust degradation, trust declining, and trust re-building. Trust rates should be continuously evaluated to reflect recent interactions. The trust mediator should be able to decline and rebuild trust, and the service providers should be able, through the service broker, to review their trust rates and consumer feedback. This can provide a good opportunity for service providers to improve their services, understand consumer needs, and build their QoBiz.

4.3.5. Bias detection

The bias detection component is responsible for detecting trust biases, which may occur if there is a significant decline from the stored rate to the evaluated one. The trust mediator can monitor services to detect biases.

4.3.6. Culture adjustment

The Web is an open environment that spans different countries, regulations, and cultures. It is important to consider cultural differences when establishing trust and selecting services. Thus, culture adjustment component is responsible for mitigating cultural differences.

4.3.7. Reward and punishment

This component is responsible for punishments and rewards. Service brokers can motivate providers to contribute positively to the network and punish other providers who act negatively and try to disrupt the system.

4.3.8. Post-monitoring

The monitoring component in the processing and evaluation phase is dedicated to the trust bootstrapping process. However, this component plays an important role in trust management, since the self-adjustment and bias detection components need to monitor services for dynamically detecting changes in their behavior that may affect their rates as well as the rates of their providers. The post-monitoring component can periodically monitor services to test their trust rates and their providers' trust rates. In addition, post-monitoring is important for monitoring the interactions between requestors and services to measure the TMs of the consumed services.

5. Experiment

This section presents the implementation and experimentation of the trust bootstrapping solution. Fig. 6 presents an electronic-market, or e-market, case study. The e-market is constructed as a composition of many Web Services, such as 'Search' for items, 'Sort items' based on different criteria, such as price, 'Calculate' the final price, and 'Check credit' for the buyers. Moreover, there are many Web Services that have the same functionality, such as 'Search', provided by different service providers. To build the e-market enterprise application, the developer needs to select Web Services that he/she can trust. The service broker acts as a TTP and supports trust-based service selection. Specifically, the e-market developer communicates with the service broker to select Web Services based on functional properties and trust criteria that meet the functional and trust preferences of the application.

The experiment requires a set of providers, each of whom offers a set of services. Different providers may provide services with the same functionality but with a different set of TMs. For example, a provider may offer a service with a set of TMs, such as $OSTM_r$, $OSTM_l$, and $SSTM_{sec}$, while another provider may offer a service with the same functionality but with a set of different TMs, such as $OSTM_{thr}$, $SSTM_{sec}$, and $SSTM_{rem}$. Alternatively, the services may have the same TMs, but each TM may have different trust rates evaluated by ToTEF. Therefore, services with similar functional properties may have different trust rates, and the service with, for example, the highest rate will be selected by the requestor.

Subsequently, services provided by a number of providers are created. Table 3 illustrates a sample of providers and their services. There are six providers that provide services of the e-market application, including search service, get items service, sort items service, place order service, calculate service, check address service, and check credit service. For example, Provider 5 has published three services, which include the search service, sort items service, and place order service.

Services are deployed on Windows machines (running Windows 8), which features a 2.4 GHz Processor, 16 GB of RAM, and 1 TB Hard Drive. *Java programing language* is used to implement different ToTEF components, such as pre-evaluation, TM trust evaluation, and matching parts.

Services are implemented using *Web Services* technology. Specifically, *WSDL* is used to describe the services and *SOAP* is used as a messaging standard. Using *Java* and *NetBeans IDE*

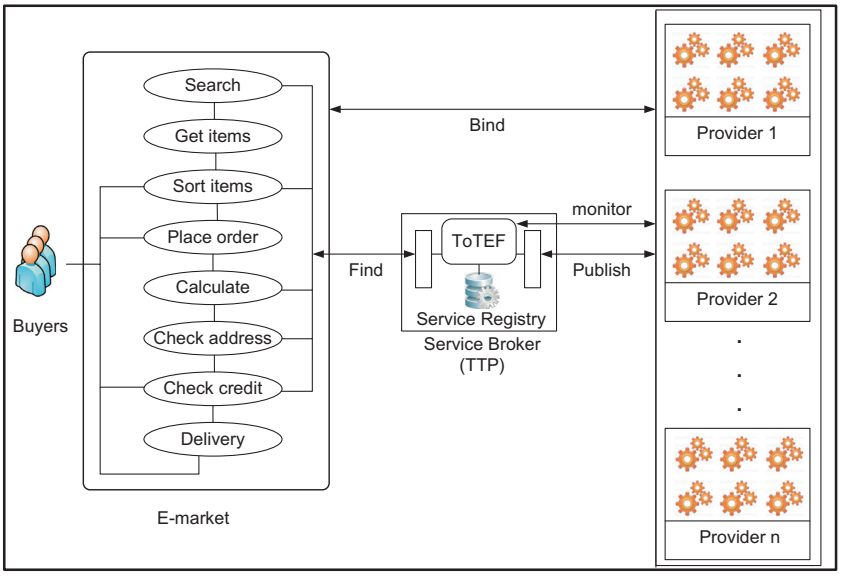

Fig. 6 Trust-based SOA, E-market case study.

Table 3 Service providers and their services used in the experiment.

Pr	Services						
	Search	Get items	Sort items	Place order	Calculate	Check address	Check credit
1	√	√		√	√	√	√
2		√	√	√	√		
3	√			√		√	
4	√		√			√	√
5	√		√	√			
6			√	√	√		√

Pr: Service providers.

6.9.1, the service providers are implemented as *Enterprise Java Beans (EJB)*. The Web Services are deployed into *GlassFish Server 3*.

The rating registry database is implemented as a *Structured Query Language (SQL) database* using *MySQL Server 5.1*. *Java Database Connectivity (JDBC)* API is used to connect the trust SQL database and ToTEF components. ToTEF uses SoapUI, a functional testing tool for testing and monitoring Web Services, to parse WSDL, invoke and monitor Web Services, which are performed by parsing/SLA, invocation, and monitoring components, respectively.

After publishing the services and TMs, ToTEF starts trust bootstrapping the TMs and evaluates the T_{TM} that are stored in the rating registry. Then T_s are calculated based on the service's evaluated T_{STM}. Table 4 shows services published by Provider 2; This table only presents two published TMs: $OSTM_r$ and $SSTM_{sec}$. T_s is evaluated based on all of the published TMs. However, during the Find operation, T_s is re-evaluated based on the requestor's trust preferences for a set of TMs, as the scenario will demonstrate in the next section. The term 'sid' refers to the service's ID number. We assume that the security and privacy SSTM ratings are obtained from security and privacy rating systems (El Yamany et al., 2010; Allison et al., 2009; Anisetti et al., 2012) and, for simplicity, use a two-scale rating of either 1 or 10, i.e, $T_{SSTM_{sec}} = 1$ or $T_{SSTM_{sec}} = 10$.

The T_{pr} are bootstrapped based on the bootstrapped T_s of their services. Table 5 presents PTMs, T_{PTM}, and T_{pr} for some providers. 'pid' represents provider ID and 'pnum' provides the number of times T_{pr} is evaluated.

6. Evaluation

This section presents the evaluation and scenario for the e-market application presented in Fig. 6. Fig. 7 shows trust rates for the services (T_s) that are provided by each service provider. For example, the fourth provider provides four services: Get items, Calculate, Check address, and Check credit. Moreover, if a request needs a "Sort items" service, which is provided by provider 1 and provider 2, then he/she will select the service provided by provider 1, which has a higher trust rate.

Fig. 8 indicates the trust rates of the service providers (T_{pr}). If a provider offers trusted services (Fig. 7), it will also be trusted (Fig. 8), as a higher service provider rate indicates that its services are also highly rated.

These figures demonstrate that T_{pr} are based on the T_s of their services. For example, because Provider 1 has highly trusted services, its trust rate is high, at $T_{pr} = 9.787$. However, Provider 4 offers services with around average trust rates, and thus, its trust rate is near the average, at $T_{pr} = 7.036$.

6.1. Scenario: service selection based on requestors' trust preferences and providers rates

This scenario shows how requestors can select a service based on their trust preferences and service providers' trust rates. In this situation, the developer of the e-market application wants to select a 'Calculate' service to build its composition of services. Because there are many 'Calculate' services, the developer should select a service that he can trust based on his trust preferences, which include execution time ($OSTM_e$), throughput ($OSTM_{thr}$), privacy ($SSTM_{prv}$), competence (PTM_{comp}), and honesty (PTM_{hons}). Subsequently, ToTEF finds different services based on the developer's preferred TMs and displays the results for the developer, as presented in Table 6.

The table shows three 'Calculate' services provided by different providers, and each with different trust rates. For example, $T_{s1} = 10$, which is the highest rate, and $T_{s21} = 6.84$, which is the lowest rate. The developer can select the service with the

Table 4 Part of service table that represents services provided by provider 2.

sid	Service's function	$OSTM_r$published		$OSTM_r$monitored	T_{OSTM_r}	$SSTM_{sec}$ supported?	$T_{SSTM_{sec}}$	...	T_s
		$OSTM_r$min	$OSTM_r$max						
7	Calculate	33	48	34	10	1	10	...	8.71
8	Get items	27	36	44	7.78	1	10	...	7.11
9	Place order	38	47	49	9.57	1	10	...	8.35
10	Sort items	64	72	74	9.72	1	10	...	8.39

Table 5 Service provider TMs and trust ratings.

pid	$T_{PTM_{sec}}$	$T_{PTM_{rem}}$	$T_{PTM_{prv}}$	pnum	T_{pr}	PTM_{brand}	PTM_{wsite}	PTM_{loc}	$T_{PTM_{comp}}$	$T_{PTM_{hons}}$	$T_{PTM_{brand}}$
1	10	10	10	6	9.787	Star	Star.com	Address 1	1	1	1
2	10	10	1	4	8.141	Moon	Moon.com	Address 2	1	0	0
3	10	1	10	3	8.571	Sun	Sun.com	Address 3	1	0	0

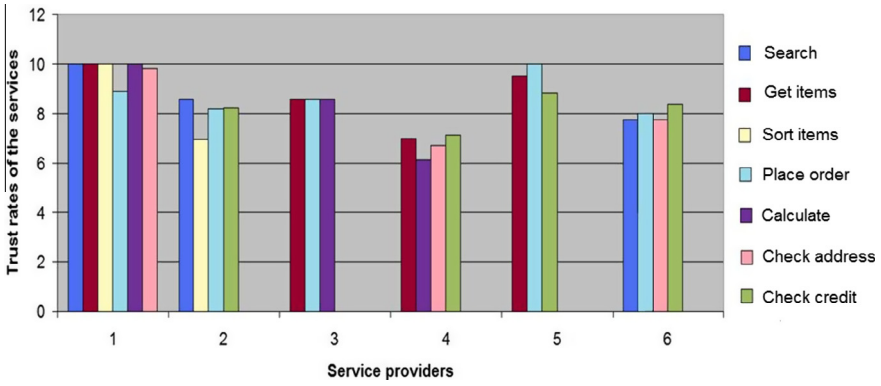

Fig. 7 T_s provided by different providers.

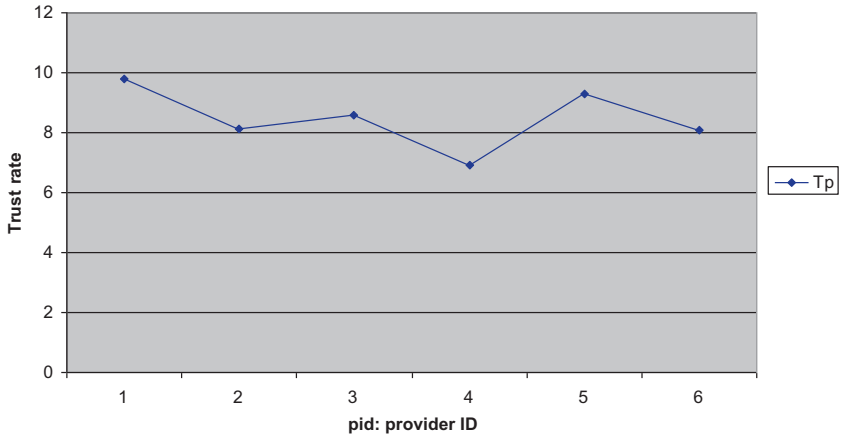

Fig. 8 Trust rates of the service providers.

Table 6 Scenario: selecting a 'Calculate' service based on requestor's trust preferences.

sid	T_{OSTM_e}	$T_{OSTM_{thr}}$	$T_{SSTM_{prv}}$	Preference T_s	$T_{PTM_{comp}}$	$T_{PTM_{hons}}$	pid	pnum
1	10	10	10	10	1	1	1	6
7	10	10	1	7	1	0	2	4
21	10	9.52	1	6.840	0	0	6	4

maximum trust rate, which in this case, is Service 1, with $T_{s1} = 10$. Moreover, the PTM supports the developer's choice; for example, the competence and honesty of a provider will encourage the developer to select its services. Therefore, the developer would select Service 1, which is offered by a competent and honest provider.

7. Conclusion and future work

Trust is an important factor in successful online interactions. It is used as a criterion for selecting services and thus affects the adoption of SOA. This paper presented an extension to SOA to support trust-based service selection. Specifically, the Total Trust Evaluator Framework (ToTEF), which is responsible for trust process, is built and added to the architecture. ToTEF is a comprehensive solution because it is built according to the definition and principles of trust to identify its main components. ToTEF contains the necessary components for trust bootstrapping, trust evaluation, trust management, and resolving trust challenges, such as unfair feedback, trust bias, and culture differences. Moreover, trust rating service providers is considered to help requestors in their selection decision.

For future work, a variety of issues merit exploration, and ToTEF has components that need to be further addressed in order to complete the trust solution. These include addressing the certification component and post-processing phase components, such as unfair feedback and culture differences. In addition, there are other issues need to be addressed such as trustworthiness of the service broker and rating a composition of services.

References

Aljazzaf, Z.M., 2015. Modelling and measuring the quality of online services. Kuwait J. Sci. Eng.

Aljazzaf, Z.M., Capretz, M.A., Perry, M., 2011a. Trust bootstrapping services and service providers. In: PST 2011 the Ninth

Annual International Conference on Privacy, Security and Trust., pp. 7–15.
Aljazzaf, Z.M., Perry, M., Capretz, M.A., 2010. Online trust: definition and principles. In: ICCGI 2010: The fifth International Multi-Conference on Computing in the Global Information Technology, pp. 163–168.
Aljazzaf, Z.M., Perry, M., Capretz, M.A., 2011b. Trust metrics for services and service providers. In: ICIW2011: The Sixth International Conference on Internet and Web Applications and Services, pp. 195–200.
Allison, D., El Yamany, H., Capretz, M., 2009. Privacy and trust policies within SOA. In: ICITST '09: International Conference for Internet Technology and Secured Transactions, 2009, pp. 1–6.
Anisetti, M., Ardagna, C., Damiani, E., Maggesi, J., 2012. Security certification-aware service discovery and selection. In: 5th IEEE International Conference on SOC and Applications, pp. 1–8.
Anisetti, M., Ardagna, C., Damiani, E., Saonara, F., 2013. A test-based security certification scheme for web services. ACM Trans. Web 7 (2), 5:1–5:41.
Azarmi, M., Bhargava, B., Angin, P., Ranchal, R., Ahmed, N., Sinclair, A., Linderman, M., Othmane, L., 2012. An end-to-end security auditing approach for service oriented architectures. In: 2012 IEEE 31st Symposium on Reliable Distributed Systems (SRDS), pp. 279–284.
Bieberstein, N., Bose, S., Fiammante, M., Jones, K., Shah, R., 2005. Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap. Prentice Hall PTR, Upper Saddle River, NJ, USA.
Cao, J., Huang, J., Wang, G., Gu, J., 2009. QoS and preference based web service evaluation approach. In: GCC '09: Eighth International Conference on Grid and Cooperative Computing, pp. 420–426.
Chen, M., He, L., Cai, X., Xia, W., 2008. Trust evaluation model for composite service based on subjective logic. In: International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 1482–1485.
Cimato, S., Damiani, E., Zavatarelli, F., Menicocci, R., 2013. Towards the certification of cloud services. In: IEEE 9th World Congress on Services, pp. 92–97.
Daignault, M., Shepherd, M., Marche, S., Watters, C., 2002. Enabling trust online. In: ISEC '02: Proceedings of the Third International Symposium on Electronic Commerce. IEEE Computer Society, p. 3.
Diego, G., 2011. Semantic Privacy Policies for Service Description and Discovery in Service-Oriented Architecture. University of Western Ontario, London, Canada (Ph.D. thesis).
Dragoni, N., 2009. Toward trustworthy web services – approaches, weaknesses and trust-by-contract framework. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology 3, pp. 599–606.
El Yamany, H.F., 2009. A Fine-Grained Intelligent Security Framework for Service-Oriented Architecture. The University of Western Ontario, Ontario, Canada (Ph.D. thesis).
El Yamany, H.F., Capretz, M.A.M., Allison, D.S., 2010. Intelligent security and access control framework for service-oriented architecture. J. Inf. Software Technol.
Gan, Z., He, J., Li, K., 2010. A service-oriented trust management framework for e-commerce systems. In: 7th Web Information Systems and Applications Conference (WISA), pp. 203–207.
Huhns, M.N., Singh, M.P., 2005. Service-oriented computing: key concepts and principles. IEEE Internet Comput. 9, 75–81.
Jin-Dian, S., He-Qing, G., Yin, G., 2005. An adaptive trust model of web services. J. Wuhan Univ. J. Nat. Sci. 10 (1), 21–25.
Kalepu, S., Krishnaswamy, S., Loke, S., 2003. Verity: a QoS metric for selecting web services and providers. In: 4th International Conference on Web Information Systems Engineering Workshops, pp. 131–139.
Kaluvuri, S., Bezzi, M., Roudier, Y., 2013. Bringing common criteria certification to web services. In: IEEE Ninth World Congress on Services, pp. 98–102.
Katopodis, S., Spanoudakis, G., Mahbub, K., 2014. Towards hybrid cloud service certification models. In: IEEE International Conference on Services Computing, pp. 394–399.
Kautonen, T., Karjaluoto, H. (Eds.), 2008. Trust and New Technologies: Marketing and Management on the Internet and Mobile Media. Edward Elgar.
Kim, Y., Doh, D., 2007. A trust type based model for managing QoS in web services composition. Int. Conf. Convergence Inf. Technol., 438–443
Kim, Y., Doh, K., 2013. Quantitative trust management to support QoS-aware service selection in service-oriented environments. In: International Conference on Parallel and Distributed Systems (ICPADS), pp. 504–509.
Lee, K., Jeon, J., Lee, W., Jeong, S., Park, S., 2003. QoS for web services: Requirements and possible approaches. Tech. rep., W3C, Web Services Architecture Working Group. URL <http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/>, last accessed Jan, 2014.
Lewis, G.A., 2013. Is SOA being pushed beyond its limits? Adv. Comput. Sci. Int. J. 2 (1), 17–23.
Liu, F., Li, H., Gao, L., Zhao, H., Liu, X., Ma, Y., 2014. A web service trust evaluation model based on small-world networks. Knowledge-Based Syst. J. 57, 161–167.
Liu, Y., Ngu, A., Zeng, L., 2004. QoS computation and policing in dynamic web service selection. In: WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters. ACM, New York, NY, USA, pp. 66–73.
Mayer, F., 1990. A brief comparison of two different environmental guidelines for determining 'levels of trust'. In: Proceedings of the Sixth Annual Conference on Computer Security Applications , 1990, pp. 244–250.
Moorsel, A., 2001. Metrics for the internet age: quality of experience and quality of business. In: 5th Performability Workshop.
Nguyen, H.T., Zhao, W., Yang, J., 2010. A trust and reputation model based on bayesian network for web services. In: ICWS '10: IEEE International Conference on Web Services, pp. 251–258.
Papazoglou, M., 2012. Web Services and SOA: Principles and Technology. Pearson Education, Essex, England, New York.
Papazoglou, M.P., Georgakopoulos, D. (Eds.), 2008. Service-Oriented Computing. The MIT Press, Cambridge, MA.
Ran, S., 2003. A model for web services discovery with QoS. ACM SIGecom Exchanges 4 (1), 1–10.
Rosen, M., Lublinsky, B., Smith, K.T., Balcer, M.J., 2008. Applied SOA: Service-Oriented Architecture and Design Strategies. Wiley Publishing.
Townend, P., Venters, C., Lau, L., Djemame, K., Dimitrova, V., Marshall, A., Xu, J., Dibsdale, C., Taylor, N., Austin, J., McAvoy, J., Fletcher, M., Hobson, S., 2012. A framework for improving trust in dynamic service-oriented systems. In: Townend, P. (Ed.), 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, pp. 136–141.
Wang, Y., Vassileva, J., 2007. A review on trust and reputation for web service selection. In: ICDCSW '07. IEEE Computer Society, Washington, DC, USA, p. 25.
Zhang, Y., Zheng, Z., Lyu, M., 2010. Wsexpress: a qos-aware search engine for web services. In: IEEE International Conference on Web Services, pp. 91–98.
Zhengping, L., Xiaoli, L., Guoqing, W., Min, Y., Fan, Z., 2007. A formal framework for trust management of service-oriented systems. In: IEEE International Conference on Service-Oriented Computing and Applications. IEEE Computer Society, Washington, DC, USA, pp. 241–248.