Towards a Flexible and Privacy-Preserving Reputation System for Markets of Composed Services

Sonja Brangewitz
Department of Economics
University of Paderborn
Paderborn, Germany
Email: sonja.brangewitz@wiwi.upb.de

Ronald Petrlic
Network Security Group
University of Paderborn
Paderborn, Germany
Email: ronald.petrlic@upb.de

Alexander Jungmann
C-LAB
University of Paderborn
Paderborn, Germany
Email: alexander.jungmann@c-lab.de

Marie C. Platenius
Heinz Nixdorf Institute
University of Paderborn
Paderborn, Germany
Email: m.platenius@upb.de

***Abstract*—One future goal of service-oriented computing is to realize global markets of composed services. On such markets, service providers offer services that can be flexibly combined with each other. However, most often, market participants are not able to individually estimate the quality of traded services in advance. As a consequence, even potentially profitable transactions between customers and providers might not take place. In the worst case, this can induce a market failure. To overcome this problem, we propose the incorporation of reputation information as an indicator for expected service quality. We address *On-The-Fly Computing* as a representative environment of markets of composed services. In this environment, customers provide feedback on transactions. We present a conceptual design of a reputation system which collects and processes user feedback, and provides it to participants in the market. Our contribution includes the identification of requirements for such a reputation system from a technical and an economic perspective. Based on these requirements, we propose a flexible solution that facilitates the incorporation of reputation information into markets of composed services while simultaneously preserving privacy of customers who provide feedback. The requirements we formulate in this paper have just been partially met in literature. An integrated approach, however, has not been addressed yet.**

Keywords–Reputation; Service Market; Service Composition; Privacy Protection; On-The-Fly Computing.

I. Introduction

A major goal of On-The-Fly (OTF) Computing [1][2][3] is the automated composition of software services that are traded on dynamic markets and that can be flexibly combined with each other. A user formulates a request for an individual software solution, receives an answer in terms of a composed service, and finally executes the composed service.

As an illustrative example, let us assume that someone wants to post-process a holiday video. However, it does not pay off to use a monolithic software solution because such software provides a lot of dispensable functionality, and is therefore too expensive to buy for just this purpose. What this person needs is an individually customized software composed of only those services, which together are able to satisfy his needs. A famous web-based platform for individual post-processing tasks is Instagram [4], which provides different image processing services that can be applied to an uploaded photo or video. However, the variety of available services is restricted and the selection of appropriate services has still to be done manually.

Now, let us consider a market of image processing services. A person, who wants to post-process his video, becomes a user within this market by formulating a request describing what he expects from the composed service (e.g., the functionality to create videos with reduced image noise and an increased brilliance homogeneously distributed throughout the entire video). Subsequently, a post-processing solution that satisfies the user's request is automatically composed based on image processing services that are supplied by different market participants. In this scenario, the user only has to pay for the actually utilized functionality.

However, for market participants it is difficult to estimate the quality of services before the service is actually used. For example, an image processing service's response time can be predicted to a certain extent, but it is very dependent on the specific context, e.g., its execution environment and its current load. Other markets such as eBay or Amazon solve this problem by using a reputation system. Within such a system, the experiences other users made in previous transactions are collected. Thereby, the reputation information provides new users an indicator for the service quality they can expect. As an example, let us consider that many users were entirely satisfied with a specific image processing service and rated it with five stars, for example. As a consequence, this service gained a high reputation, which makes it more attractive for future users. Not only the requesters, but also the whole market benefits from considering reputation, because the providers of high-quality products are rewarded with a high reputation, thereby increasing their chances for future sales. On the other hand, low-quality or even deceptive service providers will vanish from the market after some time, which again pays off for all customers. Existing reputation systems used by eBay or

Amazon, for example, do not explicitly consider ratings for composed services. Other reputation systems, such as those to rate trips or hotels, often ask the user to evaluate different aspects. However, single services cannot be combined with each other as flexibly as needed on the OTF market. Thus, a reputation system for composed services is still an open challenge.

The contribution of this paper covers the identification of requirements for a reputation system for markets of composed services such as OTF Computing. Furthermore, it covers the conceptual design of our proposed solution in terms of a flexible reputation system. Technical details and intermediate results of a prototypical implementation are not part of the contribution and are consequently beyond the scope of this paper. We are, however, currently working on an exemplary realization in order to analyze the influence and demonstrate the benefit of the incorporation of reputation information into the OTF Computing process. The contribution of this paper is not necessarily restricted to OTF Computing alone. Results of our work can also be adopted to other areas in which reputation of combinable products play a role.

To the best of our knowledge, there are currently no existing reputation system approaches that can be directly applied in OTF Computing. There are indeed reputation systems which cover the requirement of privacy protection. However, either those systems entail a high overhead and are thus impractical (as covered in related work) and too inflexible to be used in such a complex scenario as in OTF Computing, or privacy is only a "property" which is said to be achieved—but not enforced cryptographically. We rather pursue a *privacy-by-design* approach for our proposed reputation system in OTF Computing. Related to our idea of flexibility, reputation provided and requested depending on specific circumstances has been studied in multi-agent systems [5][6]. Furthermore, reputation has already been considered in the area of service composition: A survey is presented by Mármol et al. [7]. However, privacy protection is not considered by already existing approaches. Each of the existing approaches only deals with a subset of the requirements we identified.

This paper is organized as follows. Section II introduces OTF Computing while mainly focusing on those aspects that are relevant for the work at hand. Furthermore, it motivates the significance of reputation in OTF Computing. Section III gives a detailed problem description by subsequently introducing crucial requirements for a reputation system in OTF Computing. Section IV presents our conceptional solution in terms of a flexible reputation system that covers all identified requirements. Existing approaches that only partially cover these requirements are discussed in Section V. Section VI points out remaining research challenges. Finally, the paper concludes with Section VII.

II. ON-THE-FLY COMPUTING

A major goal of OTF Computing is automated composition of flexibly combinable services that are traded on markets. A user's request for an individual software solution should be resolved by automatically composing a solution on demand. OTF Computing addresses the entire process, starting with fundamental concepts for organizing large-scale service markets up to the final execution of a composed service. Embedding automatic service composition into service markets is one key challenge for realizing OTF Computing.

A. Automatic Service Composition

In general, we interpret automatic service composition as the sequential application of composition steps. A composition step may, for example, correspond to selecting a service in order to realize a placeholder within a workflow [8]. Regarding our initial example in terms of image processing services, a placeholder could correspond to a class of services which provide similar functionality (such as smoothing filters). For execution, a specific service (e.g., Gaussian smoothing) must then to be selected. A composition step, however, may also correspond to a single step within a composition algorithm based on Artificial Intelligence (AI) planning approaches [9][10].

For simplicity, let us assume that a workflow is available and that a service composition step corresponds to selecting a service. We divide a single composition step into two separate processes which subsequently reduce the amount of qualified service candidates. First of all, a *Service Matching* process determines to what extent a particular service fulfills a placeholder's functional (e.g., signatures and behavior) as well as non-functional requirements (e.g., quality properties such as response time or reliability) [11][12]. Based on the matching result, services that provide significantly different functionality or that violate important non-functional restrictions can be discarded directly. Subsequent to the matching process, a *Service Recommendation* process identifies (and ranks) the best service candidate(s) out of the set of remaining services. During the recommendation process, explicitly given non-functional objectives regarding the final composed service (e.g., maximizing the performance while simultaneously minimizing the costs) as well as implicit knowledge from previous composition processes (e.g., a certain service is more qualified in a particular context than others) are incorporated. The incorporation of knowledge from previous composition processes is realized by means of Reinforcement Learning [13] and requires feedback about the quality of the execution result [14].

B. Market Infrastructure Perspective

Figure 1 shows the transactional view on the entire OTF Computing process, reduced to those processes that are relevant for the work at hand. *OTF Provider Selection* and *Service Provider Selection* are decision-making processes regarding transactions within the market. Three different classes of market participants are involved in the overall process: users,

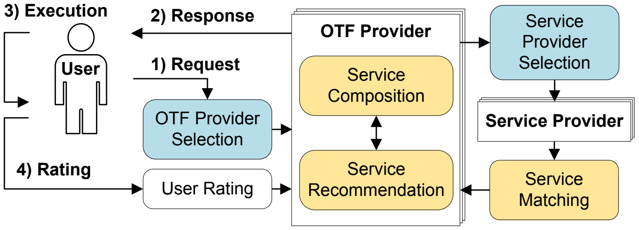

Figure 1: Overall On-The-Fly Computing process.

OTF providers, and service providers. A user formulates a request for an individual software solution and sends it to an OTF provider of his choice (*Step 1*). The selected OTF provider processes the request and automatically composes a solution based on elementary services that are supplied by independent service providers.

For each composition step, an OTF provider asks a selected subset of service providers for elementary services. The previously mentioned matching process is part of the OTF architecture and takes place before an OTF provider receives answers about appropriate elementary services. The matching process operates as a filter ensuring that only services that fulfill the desired requirements to a certain extent are returned. The recommendation process, in turn, is part of the OTF provider-specific composition process and highly depends on the context of the request.

As soon as a composed service is created, it is passed on to the user (*Step 2*), who subsequently executes it (*Step 3*). After execution, the user rates his degree of satisfaction regarding the quality of the execution result (*Step 4*). In the current setting, the value of the user rating is immediately returned to the associated OTF provider. By transforming the value into a reward and incorporating it into the Reinforcement Learning process within the recommendation system, the OTF provider improves his internal composition strategy (recommendation process) for future user requests [15].

C. Reputation as Signal for Quality

In a dynamic market of software services, information about quality (e.g., service quality or the quality of OTF providers) is essential. A user may resort only to OTF providers of a certain quality (e.g., with respect to customer support), while simultaneously accepting only composed services of a certain quality level (e.g., composed services with high reliability and trustworthiness). OTF providers, in turn, have to build composed services consisting of elementary services with a quality level according to a user's request. Information about quality, however, is either difficult to estimate before a transaction actually took place, or cannot be simply trusted if the quality information is provided by the associated market participant itself (e.g., when a service provider specifies the quality of his own services). Our solution to overcome these issues is to replace the previously mentioned and fairly simple user rating procedure (cf. Figure 1) with a flexible reputation system, which aggregates user ratings into single reputation values and provides them to market participants. Reputation can then be incorporated as an estimation of quality into the different decision-making processes.

III. PROBLEM DESCRIPTION AND REQUIREMENTS

Our goal is to explicitly incorporate reputation information as an estimation of quality into the OTF Computing process. Using goal-oriented requirements engineering [16], we systematize our reputation information requirements by investigating the role of reputation from different perspectives.

A. Reputation Information Within the On-The-Fly Process

As shown in Figure 1, the OTF Computing process is initiated by a user's request. To enable users to choose an OTF provider they want to establish a business relationship with, i.e., to buy a composed service from, reputation information about OTF providers must be available.

(*R1*) *OTF Provider Reputation:* The reputation system must provide reputation information about OTF providers.

The selected OTF provider has to ensure that the requested composed service satisfies the user's requirements regarding reputation. For this purpose, the reputation of service providers and the reputation of their supplied elementary services has to be considered during the composition process. In order to enable OTF providers to select service providers they want to retrieve elementary services from, reputation information about service providers must be available.

(*R2*) *Service Provider Reputation:* The reputation system must provide reputation information about service providers.

Reputation of elementary services influences the reputation of composed services. For example, if a composed image processing service uses a well-known, reputable implementation of a specific image filter, it can be assumed, that the composed service's reputation will be higher, than the reputation of a composed service made of unknown elementary services. Thus, the service matching processes (cf. Figure 1) as well as the service recommendation process have to consider the reputation of elementary services. While the matching process has to determine to what extent an elementary service fulfills certain requirements considering reputation, the recommendation process has to determine the best composition steps including reputation. Reputation information, however, cannot be simply extrapolated from service providers to elementary services, since a service provider may supply services of varying quality. Therefore, reputation information about elementary services must be available, too.

(*R3*) *Service Reputation:* The reputation system must provide reputation information about elementary services that have been consumed as a part of a composed service.

The recommendation process additionally rates alternative composition steps based on experience gained from previous composition processes. Reputation information about previously composed services is needed as feedback for the recommendation process in order to adapt its recommendation strategy by means of Reinforcement Learning. An OTF provider's experience, however, can be considered a business secret that must not be revealed to other market participants.

(*R4*) *Composed Service Reputation:* The reputation system must provide reputation information about composed services without revealing business secrets of OTF providers.

Users only interact with OTF providers and not with service providers directly (cf. Figure 1). As a consequence, a user's feedback mainly contains information about OTF providers and their composed services. Only once in a while may a user be able to additionally rate elementary services. For example, when using a composed service for an image processing task, users may not be aware of all elementary services, e.g., of the filter service that reduces image noise. However, they may be able to rate an elementary service that implements an image compression algorithm, since the way the algorithm effects the execution result can be directly observed in terms of the size and quality of the generated image or video.

(*R5*) *Incomplete User Rating:* The reputation system has to consider that a user is most often only able to rate OTF providers and their composed services, while a user is only sometimes able to rate elementary services and never able to rate service providers.

B. Technical Requirements

The reputation system needs to provide access to the different reputation values mentioned in the previous section for the different parties illustrated in Figure 1. Those parties have diverse and variable needs for reputation value computations and access as well as interaction preferences. For the service recommendation process, recent ratings are more important to accelerate the learning process and therefore reputation values that put a higher weight on those ratings are desired (e.g., rather a geometric mean than an average with equal weights). In contrast, for a user, it might be preferable that a certain composed service has a very low failure rate and thus, during the provider selection process, reputation values that include historic values to a sufficient extent and put a higher weight on negative ratings have to be considered. The reputation system's functionality to process user feedback and to provide it as reputation information has to satisfy the diverse needs of the requesting parties.

(*R6*) *Flexible Feedback Processing:* The reputation system must support flexible processing of user feedback.

Certain restrictions may be applied: Concerning requirement (*R4*), reputation information about composed services shall be retrievable only by the OTF provider that originally accomplished the service composition process.

(*R7*) *Access Control:* The reputation system must implement access control to reputation values.

Furthermore, the reputation system shall support different interaction models. Parties, such as the OTF provider's service recommendation component, need new reputation information as soon as it is available. New reputation information has to be automatically forwarded by the reputation system without explicitly asking for it. Other processes that rarely need to retrieve reputation information, such as users or the service matching component, shall be able to access those data actively on demand to reduce the data traffic.

(*R8*) *Interaction:* The reputation system must support alternative interaction concepts. Reputation information must either be provided on demand triggered by a request event, or actively sent to a party as soon as new reputation information is available.

Furthermore, security and privacy protection are crucial issues—as we have already investigated more generally for the OTF Computing as well [2]. If users could arbitrarily rate any services (without having used them), the reputation system would not constitute any benefit. If any party would be able to manipulate the reputation values, users could not trust the provided values and thus the reputation system's benefit would be lost as well.

(*R9*) *Rating Authorization:* Only authorized users, i.e., users that performed a transaction with an OTF provider, are allowed to rate that transaction.

(*R10*) *Correctness:* The computed reputation value provided by the reputation system must be correct, i.e., it must not be possible for any party to manipulate the reputation value (computation).

Depending on the traded services on the market, users might only be willing to rate transactions if they can stay anonymous. They do not want to (publicly) reveal which services were consumed by them. It has been shown in the past that designing a reputation system that provides user anonymity is a challenging task [17].

(*R11*) *Anonymity of Rating User:* No party shall be able to relate (individual) ratings to users.

(*R12*) *Unlinkability of User Rating to Transaction:* The OTF provider must not be able to relate a rating to a transaction (previously executed with a certain user)—in order to achieve user anonymity.

IV. A Flexible Reputation System

This section introduces the conceptual design of our proposed solution in terms of a flexible reputation system. First, the system's internal processes as well as its interaction capabilities are described. Afterwards, we illustrate in particular how the system meets each requirement listed in Section III. An overview of our proposed solution is given in Figure 2. It shows the internal structure of our flexible reputation system as well as the interactions with the OTF Computing process. Both is further explained in the following.

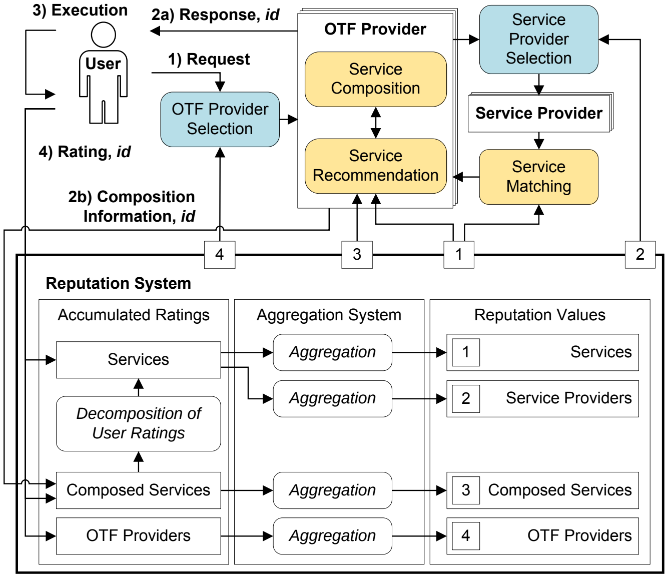

Figure 2: Proposed OTF Computing Reputation System. Internal structure and interactions with the OTF Computing process are depicted.

A. Basic Internal Structure

The reputation system is modeled as a stand-alone and independent component within the OTF Computing environment. The reputation values are derived by processing user ratings of services, composed services, as well as OTF providers. The internal structure can be divided into three main sections.

The *Accumulated Ratings* section provides functionality for accumulating raw values of incoming user ratings over time. To increase robustness, these values can be stored by means of a distributed storage system. The number of values to be stored is not necessarily restricted. However, depending on the available storage space and the amount of incoming values, outdated values may either be discarded or at least consolidated into a lower amount of values in the long run.

The *Aggregation System* provides functionality for processing a set of raw values in order to generate an aggregated representation. However, one can flexibly choose the set of raw values to be incorporated into the process, the actual aggregation function to be applied (e.g., arithmetic/geometric averaging, identifying the maximum or approximating the future trend by time series analysis) and the final representation (e.g., single scalars such as mean or median, or density functions in terms of their statistical parameters).

The *Reputation Values* section finally provides the interfaces for accessing the different reputation values of services, service providers, composed services, and OTF providers. When accessing reputation values, the set of raw user ratings to be considered, the actual aggregation function as well as the final representation can be flexibly specified. Reputation values are not stored within the system, but always computed on demand dependent on the previously mentioned specifications. This flexibility allows requests for reputation information to adapt to more complex reputation requirements imposed by users. For example, a user may want an image processing service with a reputation value higher than 4 based on at least 20 user ratings that are not older than 6 months. Another user may want an image processing service which has an average reputation value of 4, while no elementary service should have a reputation value less than 2.

B. Integration into the On-The-Fly Computing Process

Reputation values are consumed by the *Service Matching*, the *Service Recommendation*, the *Service Provider Selection*, and the *OTF Provider Selection* processes within the overall OTF Computing process. Beside flexibility regarding how a reputation value is internally computed, our proposed reputation system also provides flexible interaction capabilities. On the one hand, reputation values can be accessed by a *pull* approach whenever they are needed. Following this approach, the requester inherits the active role by asking for reputation data if and only if it is necessary. This solution is efficient when reputation information is needed less frequently (e.g., when a user wants to choose an OTF provider). On the other hand, a *push* approach shifts the active role to the reputation system. Reputation information is sent to a party as soon as new data is available. This approach also allows for creating a local cache of the latest reputation values without flooding the reputation system with redundant requests for possibly new information.

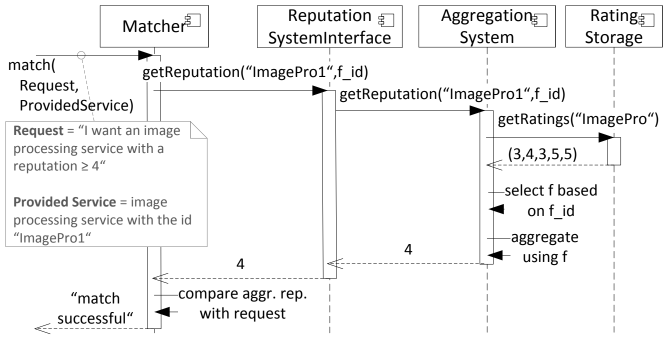
Matcher
Reputation SystemInterface
Aggregation System
Rating Storage
match(Request, ProvidedService)
getReputation("ImagePro1",f_id)
getReputation("ImagePro1",f_id)
getRatings("ImagePro")
Request = "I want an image processing service with a reputation ≥ 4"
Provided Service = image processing service with the id "ImagePro1"
(3,4,3,5,5)
select f based on f_id
aggregate using f
4
4
compare aggr. rep. with request
"match successful"

Figure 3: Simplified example interaction with the reputation system.

Figure 3 shows the interaction with the proposed reputation system using the example of the service matching process (matcher). During the OTF Computing process, the matcher is called for each elementary service that possibly satisfies an OTF provider's request (cf. Section II-B). In this context, Figure 3 illustrates the access of reputation information for exactly one elementary service by a *pull* approach.

The reputation matching process is initiated by providing the request information and the description of an elementary service and by calling the *match* operation. For the sake of simplicity, the request in the depicted example only shows an extract: An image processing service should have a minimum reputation value of 4. This request shall now be matched against an elementary service with id `ImagePro1`. The matcher asks the reputation system for a reputation value of service `ImagePro1` aggregated by means of an aggregation function with id `f_id`. Hence, the aggregation system fetches the relevant user rating values (3,4,3,5,5) from the storage, selects the corresponding aggregation function (here, arithmetic averaging), and computes an aggregated reputation value of 4. Based on this result, the matcher decides that the service matches to the request.

After a composed service was executed (*Step 3* in Figure 2), users are encouraged to provide feedback on their transactions. They are asked to rate composed services, OTF providers, and single services. The feedback in terms of user ratings is the foundation for generating reputation information within the reputation system. To be able to identify which composed service a rating belongs to, OTF providers attach an *id* to their response (*Step 2a* in Figure 2). This *id* corresponds to the particular structure of a composed service, meaning that identical composed services have identical *id*s. During the rating process for a composed service, this *id* is forwarded to the reputation system (*Step 4* in Figure 2).

Elementary services that are consumed as a part of a composed service cannot always be rated separately by the user. In fact, due to complex user requests, we expect that this is rarely possible. Thus, in order to still be able to provide reputation values for elementary services and to benefit from all information available, our reputation system decomposes user ratings of composed services. To enable this decomposition, the *id* the OTF provider sends with his response (cf. Figure 2) is reused: Simultaneously with his response to the user (*Step 2a* in Figure 2), the OTF provider sends the same *id* together with *composition information* to the reputation system (*Step*

2b in Figure 2).

As pointed out above, our reputation system for the OTF Computing shall provide flexibility, which also means that different implementations for the components are supported. We have already shown that such an implementation of a reputation system for the OTF Computing can be done in a *secure* and *privacy-preserving* way—respecting the requirements stated in Section III [18]. In contrast to related work, as covered in Section V, this approach only requires a single *reputation provider*, which is in line with the requirements of OTF Computing, and does not need any other components (such as a bulletin board). The approach is based on the Paillier cryptosystem [19] to provide a reputation value as an aggregation of individual user ratings without revealing anything about the individual ratings to any party.

C. Satisfying On-The-Fly Computing Requirements

Our proposed solution in terms of a flexible reputation system fulfills all requirements listed in Section III. This section points out how the reputation system fulfills each of these requirements in particular.

The proposed reputation system enables users to rate OTF providers, composed services, and—if possible—elementary services. Assured by the transfered *id*, in this context, only users that are involved in a particular transaction taking place on the OTF market, i.e., users that have requested, received and executed a particular composed service, are allowed to participate in the rating process. This ensures ratings by authorized users (*R9*). How to realize the rating process in particular (i.e., what kind of questions have to be asked and how a user rating value is represented) is beyond the scope of this paper.

Correctness of the provided reputation values is ensured by design. Reputation values are computed on demand by the system itself based on a pre-defined set of aggregation functions. Furthermore, the entire system is an independent component within the OTF Computing environment. As a consequence, manipulations of the computation process by other participants are eliminated (*R10*).

Anonymity of users (*R11*) as well as unlinkability of user ratings to transactions (*R12*) is ensured by the accumulation and aggregation functionality. For reasons of privacy protection, i.e., in order to not reveal individual user ratings, the reputation system always collects individual ratings and aggregates them. Although the single user ratings are stored within the reputation system, they are not accessible to market participants so that individual ratings are not traceable. In this context, it is important that the amount of accumulated user ratings is high enough and that the aggregation operation sufficiently condenses the user ratings such that it can be guaranteed that no information on individual ratings can be recovered. If not enough user ratings are included in the aggregation process (e.g., when not enough user ratings are available yet, or if a request explicitly specifies to only consider just a few user ratings), the reputation system will not provide a value but will raise an exception.

All processes that need reputation information within the entire OTF Computing process have access to the reputation system. The flexibility of our proposed solution enables each market participant to freely choose an interaction approach (*push* or *pull*) that is most appropriate with respect to the market participant's internal processes (*R8*). Furthermore, the process of generating reputation values can be adjusted by each market participant individually by specifying the set of user ratings to be considered, the actual aggregation function to be applied, and the final representation of the aggregated value (*R6*).

Reputation information about OTF providers (*R1*) is provided by the reputation system in a straight-forward manner. Users rate their satisfaction regarding the transaction with an OTF provider. These ratings are accumulated and aggregated by the reputation system and can be accessed by other users. The process of generating reputation information about composed services (*R4*) is similar. Users rate their satisfaction regarding the execution process and the execution result of a composed service. These ratings, again, are accumulated and aggregated by the reputation system. In comparison to the reputation of OTF providers, however, reputation information about composed services is OTF provider-related. In order to preserve business secrets, OTF providers can only access anonymized user ratings of composed services they originally sold (*R7*).

Besides being directly rated by users, ratings of elementary services also have to be derived from ratings for composed services (*R3*). For this purpose, OTF providers send information about their composed service to the reputation system. In order to not reveal their business secrets, this composition information, however, only consists of abstract, structural information. Only the set of elementary services included in a composed service is exposed, but not, for example, when and how often a particular service is called. This way, the provider's business secrets are protected, while it also allows for a mapping of the rating for a composed service to single services (*R5*).

Since users only interact with OTF providers, user ratings for service providers cannot be provided to the reputation system (*R5*). To overcome this problem, the aggregation system extrapolates from reputation information about elementary services to information about the associated service providers during the aggregation process (*R2*).

While composing services, reputation information about elementary services have most likely to be aggregated in order to choose composed services not only based on their (aggregated) non-functional properties, but also based on their overall reputation. How to determine this overall reputation, however, depends on the user requirements and the composition strategy of the respective OTF provider. If a user requires, e.g., all elementary services to satisfy a minimal reputation value, an OTF provider has to check the reputation value of each services individually. Another user might be satisfied with an average reputation value above a specific threshold. In this case, an OTF provider has to determine the average reputation value by aggregating all single values. Subsequently, the aggregated value and the threshold value have to be compared. In either case, aggregation of reputation values is not part of the reputation system itself. For that reason, a further investigation of how to appropriately integrate reputation information into service composition in addition to common non-functional properties is beyond the scope of this paper.

V. Related Work

There is a lot of literature on reputation, both in economics and computer science. Our interpretation of reputation is used for instance by Shapiro [20] or as well by Bar-Isaac and Tadelis [21], who summarize the economic literature on reputation. Design aspects related to mathematically modeling a reputation system and challenges that arise with online transactions, are explicitly discussed by Friedman et al. [22] and Dellarocas [23], for example. More closely related, we identify three involved fields, *Reputation Systems*, *Privacy-Preserving Systems* and *Service Composition*, and their overlappings with each other as shown in Figure 4. In the following, we present related work which has been done within these overlappings in more detail.

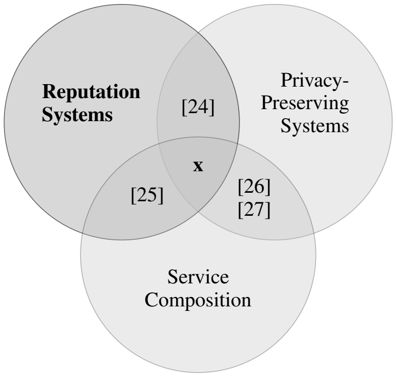

Figure 4: Overview of related work.

A. Reputation Systems and Privacy-Preserving Systems

Researchers have come up with *privacy-preserving* reputation systems in the past. Kerschbaum et al. [24] present a system which requires two centralized mutually mistrusting reputation providers in order to achieve anonymous user ratings. Users encrypt their ratings and send them to the first reputation provider which collects a number of ratings and then publishes them to a bulletin board. The second reputation provider retrieves the ratings from the bulletin board to decrypt and aggregate them before providing a (computed) reputation value. The approach is based on the Paillier cryptosystem [19]. However, the approach is too inflexible and complex to be used in our OTF Computing setting. We want to keep a lean OTF infrastructure with only one reputation provider and no other additional components, such as a bulletin board, used only by the reputation system.

B. Reputation Systems and Service Composition

Motallebi et al. [25] integrate *Component Reputation* and *Component Trust* in order to derive the reputation of a composed service from trust values for single services. They do this by taking into account the frequency of invocations of these services. However, this approach covers only some of our requirements for a reputation system in On-The-Fly Computing. For example, neither service providers are considered, nor is privacy or security a topic within their publication.

C. Service Composition and Privacy-Preserving Systems

Tbahriti et al. [26] identify privacy preservation as one of the most challenging problems in *Data-as-a-Service (DaaS)* services composition. DaaS is about combining web services for data publishing and sharing. In their proposed approach, *privacy policies* specify how collected data is treated and *privacy requirements* specify how the service-consuming services are expected to treat the provided data. Similarly, Costante et al. [27] come up with a solution for web service selection and composition that takes privacy into account. Users are able to specify their privacy preferences which are checked against the service providers' privacy policies. Only in the case of a successful match are the service providers' services selected and used for composition. Both approaches do not take into account reputation of elementary or composed services.

In contrast to related work, we pursue a *privacy-by-design* approach that builds privacy protection into the reputation system for OTF Computing. This allows us to prove that privacy is achieved rather than to rely on guarantees made by the participants.

D. Conclusion: Related Work

It is noteworthy that no work—to the best of our knowledge— that includes all the different fields mentioned above (the overlapping marked "x" in Figure 4) has been done. This is where we contribute with this paper: We are the first to present the requirements for such a system, describe a flexible solution, and point out further interesting research challenges that still need to be solved in the future.

VI. Research Challenges

The introduction of a reputation system in the OTF Computing in Section IV is conceptual and provides flexibility for further specifications. As research challenges, we highlight some of the trade-offs that result from the requirements imposed in Section III. A more detailed investigation of each of the research challenges is beyond the scope of this conceptual contribution and is planned to be considered in future work.

Efficiency in Learning versus Privacy Protection: The reinforcement learning approach, which is used to improve the service composition process, needs direct feedback after each composition. If the feedback is absent, the learning process is hampered. However, for reasons of privacy protection, no direct feedback is given to any party. Only an aggregated value of the accumulation of several individual ratings (feedback) is provided, as described in Section IV. Thus, a research challenge is to investigate the trade-off between privacy protection and learning efficiency. It has to be investigated how a delayed feedback after several service composition processes (accumulation) in an aggregated form affects the convergence behavior of the learning process.

Benefit of Privacy Protection: As discussed in this paper, the design of a privacy-preserving solution entails a multitude of trade-offs that need to be taken into account, e.g., the trade-off between privacy and learning mechanism efficiency. Thus, it needs to be investigated whether market participants are interested in implementing a privacy-preserving solution at all. We need to prove that privacy protection is a benefit of OTF Computing and that users rather use such a market than any other which does not provide such strong privacy guarantees. Concerning the introduced reputation system, we want to examine whether users are more willingly providing ratings when their privacy is protected—which is not the case in any other state-of-the-art reputation system in use today.

Manipulation Resistance versus Privacy Protection: An important further issue is to obtain truthful user feedback. Ratings may be dishonest or randomly chosen [22]. So far we assumed that users have no incentives to strategically manipulate their feedback and moreover we supposed that feedback on a transaction is always provided. Truthful rating behavior is induced by *incentive compatible reputation mechanisms* [28] (and the references mentioned therein). To ensure privacy protection, several ratings need to be accumulated and aggregated. It has already been analyzed how the aggregation of ratings impacts the efficiency of a reputation mechanism [29] and how it influences incentives for truthful rating behavior [30]. An important next step now is to further understand the interplay of incentive compatibility and privacy protection. Therefore, a challenging question is whether and how it is possible to design reputation systems that induce truthful feedback and respect privacy protection.

Fuzzy Matching of Reputation Values: Another open issue is how reputation should be matched. Since the reputation of a service is not an objective measure, such as signatures or protocols, uncertainty might be introduced into the matching process. For example, as noted in our fuzzy matching survey [12], the user stating the request might tolerate variations (e.g., "I want a service with *approximately* five stars"), or the request might include requirements for which the corresponding information on the provider side do not exist yet (e.g., there has not been much feedback yet because the service is new on the market and thus the reputation is unclear). We are going to analyze how a fuzzy reputation matching can cope with such challenges.

Context-Specific Reputation: In our current system, we focus on the overall reputation of (composed) services and providers. However, in reality, reputation is rather context-specific [31]. For example, an image processing service could have a good reputation regarding the response time but a bad reputation regarding security. Thus, the reputation system should maintain vectors instead of single values for ratings and reputation. However, this also increases the complexity of the different components that access the reputation system. For example, matching could become much more detailed, but also less efficient in terms of performance.

Trust and Reputation: Another open issue is to distinguish between trust and reputation. Trust can be understood as a private reputation value in contrast to the public reputation value [32]. It needs to be analyzed whether reputation systems should distinguish between these concepts and how the whole scenario could benefit from it.

VII. Conclusion

In the context of OTF Computing, we use a reputation system to collect information about experiences users make with composed services in transactions. From an economic perspective, the buying decision of a user and the future sale opportunity of an OTF provider crucially depend on the current reputation value. Our contribution in this paper comprises the collection of requirements and the proposal of a conceptual solution for a flexible reputation system in OTF Computing. To fulfill the posed requirements, we identified necessary operations as well as additional properties and described their interaction. We analyzed the influence of reputation information on the processes and proposed the integration of a reputation system in the OTF Computing infrastructure. In our work, we put a special focus on composed services as well as on privacy. As part of our contribution, we combined approaches from the literature on reputation systems, service composition, and privacy protection. Finally, we presented research challenges that arise from conflicting objectives and deserve further investigations.

Acknowledgment

This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" (SFB 901).

References

[1] M. Happe, F. M. auf der Heide, P. Kling, M. Platzner, and C. Plessl, "On-the-fly computing: A novel paradigm for individualized it services," in Proceedings of the Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS). IEEE, 2013.

[2] R. Petrlic, A. Jungmann, M. C. Platenius, W. Schäfer, and C. Sorge, "Security and Privacy Challenges in On-The-Fly Computing," in Tagungsband der 4. Konferenz Software-Technologien und -Prozesse (STeP 2014), 2014, to appear.

[3] "Collaborative Research Center 901 - On-The-Fly Computing," 2014, URL: http://sfb901.uni-paderborn.de [accessed: 2014-03-15].

[4] "Instagram," 2014, URL: http://www.instagram.com [accessed: 2014-03-15].

[5] V. T. Silva, R. Hermoso, and R. Centeno, "A Hybrid Reputation Model Based on the Use of Organizations," in Coordination, Organizations, Institutions and Norms in Agent Systems IV. Springer, 2009, pp. 111–125.

[6] J. S.P. Guedes, V. Torres da Silva, and C. Lucena, "A Reputation Model Based on Testimonies," in Agent-Oriented Information Systems IV. Springer, 2008, pp. 37–52.

[7] F. G. Mármol and M. Q. Kuhnen, "Reputation-based Web service orchestration in cloud computing: A survey," Concurrency and Computation: Practice and Experience, 2013.

[8] N. Hiratsuka, F. Ishikawa, and S. Honiden, "Service Selection with Combinational Use of Functionally-Equivalent Services," in Proceedings of the 18th IEEE International Conference on Web Services (ICWS), 2011, pp. 97–104.

[9] P. Bartalos and M. Bielikova, "Semantic Web Service Composition Framework Based on Parallel Processing," in Proceedings of the 11th IEEE Conference on Commerce and Enterprise Computing (CEC), 2009, pp. 495–498.

[10] M. Aiello, E. el Khoury, A. Lazovik, and P. Ratelband, "Optimal QoS-Aware Web Service Composition," in Proceedings of the 11th IEEE Conference on Commerce and Enterprise Computing (CEC), 2009, pp. 491–494.

[11] M. C. Platenius, "Fuzzy Service Matching in On-The-Fly Computing," in Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering (ESEC/FSE). ACM, 2013, pp. 715–718.

[12] M. C. Platenius, M. von Detten, S. Becker, W. Schäfer, and G. Engels, "A Survey of Fuzzy Service Matching Approaches in the Context of On-the-fly Computing," in Proceedings of the 16th International ACM Sigsoft Symposium on Component-based Software Engineering (CBSE). ACM, 2013, pp. 143–152.

[13] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. MIT Press, 1998.

[14] A. Jungmann and B. Kleinjohann, “Learning Recommendation System for Automated Service Composition,” in Proceedings of the 10th IEEE International Conference on Services Computing (SCC), 2013, pp. 97–104.

[15] A. Jungmann, B. Kleinjohann, and L. Kleinjohann, “Learning service recommendations,” Int. J. Business Process Integration and Management, vol. 6, no. 4, 2013, pp. 284–297.

[16] A. Van Lamsweerde, “Goal-oriented requirements engineering: A guided tour,” in Proceedings of the Fifth IEEE International Symposium on Requirements Engineering (RE), 2001, pp. 249–262.

[17] A. Narayanan and V. Shmatikov, “Robust De-anonymization of Large Sparse Datasets,” in Proceedings of the IEEE Symposium on Security and Privacy (SP), 2008, pp. 111–125.

[18] R. Petrlic, S. Lutters, and C. Sorge, “Privacy-Preserving Reputation Management,” in Proceedings of the 29th Symposium On Applied Computing. ACM, 2014.

[19] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in Proceedings of the 17th international conference on Theory and application of cryptographic techniques. Springer, 1999, pp. 223–238.

[20] C. Shapiro, “Premiums for High Quality Products as Returns to Reputations,” The Quarterly Journal of Economics, vol. 98, no. 4, 1983, pp. 659–680.

[21] H. Bar-Isaac and S. Tadelis, “Seller Reputation,” Foundations and Trends in Microeconomics, vol. 4, no. 4, 2008, pp. 273–351.

[22] E. Friedman, P. Resnick, and R. Sami, “Manipulation-Resistant Reputation Systems,” in Algorithmic Game Theory, Chapter 27. Cambridge University Press, 2007.

[23] C. Dellarocas, “Reputation Mechanism Design in Online Trading Environments with Pure Moral Hazard.” Information Systems Research, vol. 16, no. 2, 2005, pp. 209–230.

[24] F. Kerschbaum, “A verifiable, centralized, coercion-free reputation system,” in Proceedings of the 8th ACM workshop on Privacy in the electronic society (WPES), 2009, pp. 61–70.

[25] M. R. Motallebi, F. Ishikawa, and S. Honiden, “Component Trust for Web Service Compositions,” in AAAI Spring Symposium Series, 2012.

[26] S.-E. Tbahriti, M. Mrissa, B. Medjahed, C. Ghedira, M. Barhamgi, and J. Fayn, “Privacy-Aware DaaS Services Composition,” in Database and Expert Systems Applications, 2011, pp. 202–216.

[27] E. Costante, F. Paci, and N. Zannone, “Privacy-Aware Web Service Composition and Ranking,” in Proceedings of the 20th IEEE International Conference on Web Services (ICWS), 2013, pp. 131–138.

[28] S. Phoomvuthisarn, “A Survey Study on Reputation-based Trust Mechanisms in Service-Oriented Computing,” Journal of Information Science and Technology, vol. 2, no. 2, 2011, pp. 1–12.

[29] C. Dellarocas, “How Often Should Reputation Mechanisms Update a Trader’s Reputation Profile?” Information Systems Research, vol. 17, no. 3, 2006, pp. 271–285.

[30] C. Aperjis and R. Johari, “Optimal Windows for Aggregating Ratings in Electronic Marketplaces,” Management Science, vol. 56, no. 5, 2010, pp. 864–880.

[31] Y. Wang and J. Vassileva, “A Review on Trust and Reputation for Web Service Selection,” in 27th International Conference on Distributed Computing Systems Workshops (ICDCSW), 2007, pp. 25–25.

[32] R. Kiefhaber, G. Anders, F. Siefert, T. Ungerer, and W. Reif, “Confidence as a Means to Assess the Accuracy of Trust Values,” in 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2012, pp. 690–697.