Rated Tags as a Service:
A Cloud-based Social Commerce Service

Daniel Kailer

Munich University of Applied Sciences
Department of Computer Science and Mathematics
Munich, Germany
Email: dkailer@hm.edu

***Abstract*—The rise of social media has influenced many web applications, particularly in the area of e-commerce. Especially small and medium enterprises (SMEs) benefit from user-generated content, because these enterprises often do not have the dedicated resources to generate or categorize content. Moreover, SMEs often do not have the resources to create social media features by themselves. As a solution to these problems, a Cloud-based social commerce service named Rated Tags as a Service is presented in this paper. The intention of the Rated Tags system is to improve the decision making process of customers. This was already successfully evaluated in a user study. This paper discusses client- and server-side challenges for providing such a feature in a service-oriented way and proposes a corresponding Cloud-based architecture.**

Keywords–SaaS; E-Commerce; Architecture; SME.

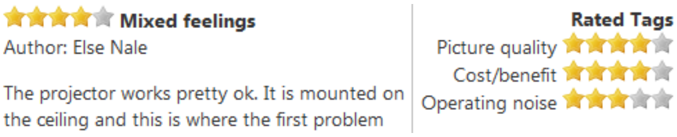
I. Introduction

The rise of Web 2.0 and social media has influenced many web applications in recent years. Especially e-commerce companies make use of social media to transform their business into a more customer-centered environment [1]. The term social media stands for interactive, web-based applications that allow the creation and exchange of user-generated content [2]. The use of social media in e-commerce is often referred to as social commerce [3].

Social media, particularly user-generated content, is especially interesting for small and medium enterprises (SMEs), because it is an easy and cheap way to generate content and SMEs typically do not have dedicated resources to create or categorize content. SMEs usually also do not have the resources to implement and maintain their own infrastructure for these social interaction features. A possible solution to this is the service-oriented paradigm, i.e., the ability to use existing services instead of implementing and maintaining them by themselves.

This paper proposes a client- and server-side architecture for a novel social commerce service named Rated Tags as a Service. The Rated Tags system itself was designed to support the decision making of e-commerce customers through social tagging. The next step is the service-oriented provisioning of the system for online retailers, especially small and medium ones. This paper discusses some client- and server-side challenges and proposes an Cloud-based architecture for such a service.

The remainder of this paper is organized as follows. In Section II, the concepts behind the Rated Tags system and its related work will be briefly explained. After that, the proposed architecture and its implications are discussed in Section III. Finally, Section IV concludes this article and presents areas for future research.



★★★★☆ **Mixed feelings**
Author: Else Nale

The projector works pretty ok. It is mounted on the ceiling and this is where the first problem

Rated Tags
Picture quality ★★★★☆
Cost/benefit ★★★★☆
Operating noise ★★★☆☆

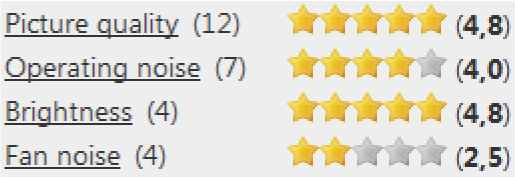
Figure 1: Exemplary excerpt of a customer review with attached Rated Tags

II. Background and Related Work

Many online retailers make use of user-generated product ratings and reviews. Especially consumer reviews are a helpful resource for information seeking customers. However, it is a time-consuming task to analyze the vast amount of reviews, especially because of the unstructured nature of textual reviews. A social commerce feature named Rated Tags was developed by the author of this paper to reduce this decision making effort of customers.

A Rated Tag is the combination of a user-generated tag and a 5-star rating [4]. Similar to traditional social tagging, reviewers can create a Rated Tag and assign it to their review. For example, if a reviewer writes about the picture quality and operating noise of a projector, he or she can assign the Rated Tags “picture quality” and “operating noise” to their review and rate them on a 5-star scale. An exemplary excerpt of a customer review that contains Rated Tags is shown in Figure 1. Because of the user-generated nature of Rated Tags, other reviewers can reuse the created Rated Tags and assign them in their reviews. As a consequence, other information seeking customers are able to display only reviews that discuss a specific aspect of a product, for example the picture quality of a projector. Additionally, the ratings of the Rated Tags can be aggregated, which gives customers a good overview about the discussed aspects in the reviews, as shown in Figure 2. Thus, Rated Tags can be classified as an interactive decision aid (IDA).

To determine the helpfulness of Rated Tags as an IDA, the system was evaluated in a case-control study with 34 participants. The participants were provided with 5 products, whereas each product had 20 customer reviews. The participants of the



Picture quality (12) ★★★★★ (4,8)
Operating noise (7) ★★★★☆ (4,0)
Brightness (4) ★★★★★ (4,8)
Fan noise (4) ★★☆☆☆ (2,5)

Figure 2: Exemplary summary of aggregated Rated Tags

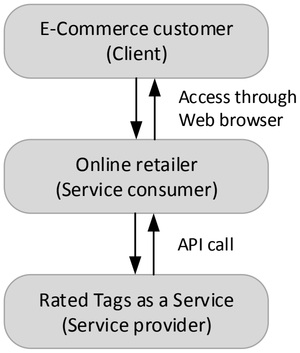


Figure 3: Conceptual model for Rated Tags as a Service

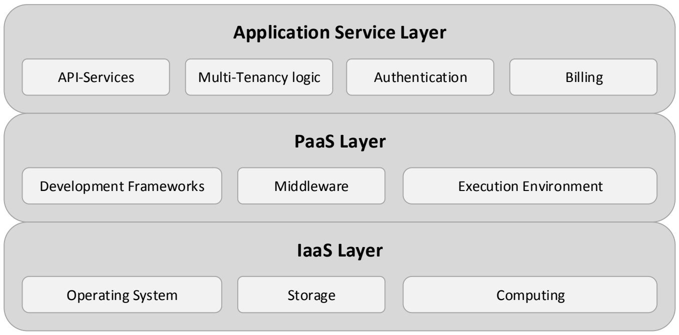


Figure 4: Proposed Cloud-based architecture for the service provider

Rated Tags group also had access to the corresponding Rated Tags (as shown in Figure 1). The instructions for the participants were to select the product with the lowest operating noise. The study results show, that Rated Tags users had a significant increase in decision quality. This was measured by comparing the selected product of the participants with the dominating product, i.e., the product with the lowest operating noise. Participants of the Rated Tags group significantly more often chose the dominating product than participants of the control group. The results also show a decrease in decision effort. This was measured by the time required to chose a product. The participants of the Rated Tags group were significantly faster in their decision making.

Similar approaches to Rated Tags were proposed by Vig et al. [5] and Lee et al. [6]. Their research also conducted the combination of user-generated tags and ratings. However, their research focused on different aspects, for example the user acceptance or user interface for the created tags. The research for Rated Tags primarily focuses on an improvement of the decision making process of e-commerce customers.

The combination of e-commerce and Cloud-based services is subject to several scientific articles [7][8] [9][10]. However, these works only discuss the integration of Cloud-based e-commerce services from a theoretical point of view. In contrast to that, this paper presents a conceptual model and architecture for a concrete e-commerce service. Therefore it is more likely to find solutions for real world problems, which can later be generalized to a common Cloud-based social commerce framework.

III. Rated Tags as a Service

This section constitutes the core of this article. It describes the proposed client- and server-side architecture to make the Rated Tags system available in a service-oriented way. As depicted in Figure 3, the three relevant actors for the architecture are service provider, service consumer and the e-commerce customer.

A. *Service provider architecture*

Because the proposed service needs to support a rising number of service consumers, an important aspect for the underlying system is scalability. Scalability means that the underlying system can cope with an increasing amount of load or traffic without modifying the system's architecture [11]. For this reason, a Cloud-based solution is proposed, because Cloud-resources allow a dynamic, on-demand scaling without the need to maintain an own data center [12].

The proposed Cloud-based architecture for the service provider is depicted in Figure 4. It is based on the service orchestration architecture from the NIST cloud reference architecture [13, p. 3]. The topmost layer is the application service layer. This layer contains all the application specific logic for the service, while only a subset of all the components is displayed in Figure 4. The service layer makes use of the underlying PaaS and IaaS layers. The PaaS layer provides the execution environment for the service among other things. Finally, the bottom (IaaS) layer provides access to various resources, e.g., files or other storage structures.

An important aspect for the architecture is the underlying data model. To fully take advantage of the Software as a Service (SaaS) paradigm, a multi-tenant architecture is proposed. Multi-tenancy means, that a single software instance runs on a server, which serves multiple, independent service consumers (tenants) [14]. Therefore the data model needs to be designed for data partitioning, which is also a requirement to improve the scalability of Cloud-based applications [14].

The proposed partitioning scheme is horizontal data partitioning by tenants. This means that every tenant has its dedicated table. An example for this is shown in Table I. The name of the table is “RatedTag_ACME”, whereas the name or id of the tenant (in this case “ACME”) appears after the underscore. The shown columns do not contain any tenant-specific data. This is different from the shared table approach, where a table is shared among tenants and the identification is done via a column that identifies the tenant [15]. The advantage of the proposed dedicated table approach is that the data of tenants is physically separated from each other. It is also

TABLE I: An example of a tenant-specific table

RatedTag_ACME						
Id	ProductCategoryId	ProductId	ReviewId	UserId	Tag	Rating

assumed that this approach performs better, but this needs to be evaluated in a future simulation.

As shown in Table I, specific ids from the service consumer need to be stored to connect the reviews of the service consumers to the saved Rated Tags, for example the id of the review. Because the used format for these ids is tenant-specific and can be numeric or text-based, a common format is required. The proposed type for these columns is text, because the common types of ids (numeric ids and GUIDs) can be transformed into a textual format.

Another important aspect is the type of data storage. Traditionally, data for web applications is stored in relational databases like MySQL or Microsoft SQL Server, but with the rise of cloud computing and the increasing need for scalability the paradigm of NoSQL emerged. NoSQL databases do not require a fixed table scheme and mostly use an eventually consistent model in favor of availability and partition tolerance. Eventually consistent means that "the storage system guarantees that if no new updates are made to the object, eventually all accesses will return the last updated value" [16]. Because the Rated Tags system has no strict consistency requirements, an eventually consistent model is proposed.

B. Service consumer architecture

The main challenge for service consumers is the integration of the service in their own infrastructure. Three imaginable integration scenarios for service consumers are presented and discussed. These integration scenarios are depicted as sequence diagrams in Figures 5, 6 and 7.

One possible approach for service integration is shown in Figure 5. It shows a client (e-commerce customer), who requests a product page from the online store where Rated Tags are used. The online store (service consumer) then forwards an API call to the service provider to get all relevant data for the current context. After the completed API call, the retrieved data is included in the HTML and the rendered response is delivered back to the client. This can be considered as a synchronous approach, because the page rendering waits for the API call to complete.

Another possible approach is shown in Figure 6. This approach is similar to the previous one, but it applies an asynchronous communication. The requested page is rendered immediately and includes JavaScript code to reload additional content while loading the website. The JavaScript code calls the server of the service consumer, which forwards the API call to the service provider to fetch additional data and deliver the rendered results back to the client. This approach is more responsive than the previous one, because the Rated Tags-specific content is asynchronously loaded after the main web site is loaded.

Finally, a third approach is shown in Figure 7, where the client directly communicates with the service provider. This is different to the two previous approaches, where the service consumer acted as a proxy, that forwarded an API call to the service provider. Such a direct communication approach is typically taken by mashups or widgets, that include mainly JavaScript and do not have any server side code. An exemplary service that uses this technique is Disqus [17], which is a discussion service, that can be integrated in any website to facilitate social interactions between users of this website. The approach has the advantage that it is easy to integrate, but it also has issues that needs to be considered. Some of these issues are discussed below.

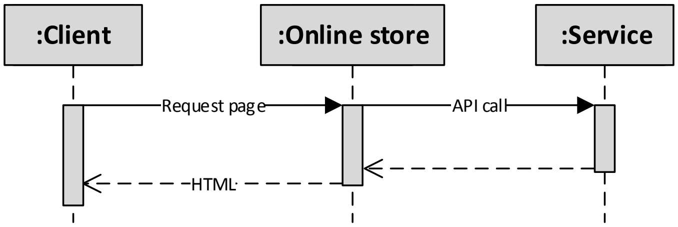


Figure 5: Integration scenario 1: Load content at once (no JavaScript required)

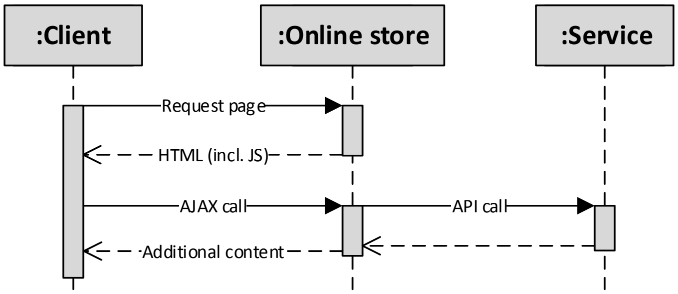


Figure 6: Integration scenario 2: Load additional content via AJAX

A problem of the above approach is the required cross-domain request, because the JavaScript same-origin policy forbids browsers to send and receive content from a different domain. There are two solutions to get around this problem: JSON with padding (JSONP) and cross-origin resource sharing (CORS). JSONP is a technique to include and execute JavaScript from a different domain by specifying a callback function [18]. A restriction of JSONP is that it only supports HTTP-GET requests. CORS can be used to allow JavaScript requests from other domains. For this to work, the client (browser) and server must specify special CORS HTTP-headers [19]. Unfortunately, CORS is a newer mechanism, which means this technique is not supported by older browsers. The direct communication approach also lacks a server-side component, which is required to control the access to the service. Because of these issues, the integration scenario from Figure 7 is not suited for the architecture.

Eventually, the decision about the integration scenario depends upon the service consumer. If it is acceptable to use JavaScript, the approach from Figure 6 should be preferred, because it is more responsive. Alternatively, the approach from Figure 5 can be used when the online store of the

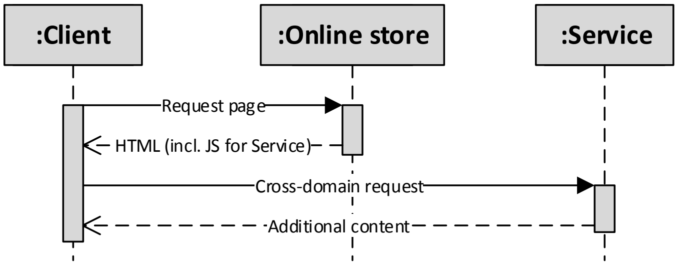


Figure 7: Integration scenario 3: Client-side call to the API

service consumer needs to be available for customers who have JavaScript disabled. The proposed architecture supports both approaches.

IV. Conclusion and Future Work

This paper presented the conceptual model and a Cloud-based architecture for a novel service called Rated Tags as a Service. The challenges for client- and server-side implementation were discussed. An important aspect for the client-side is the ease of integration, for which possible integration scenarios were presented and assassed. From the server-side perspective, the overall cloud-based architecture, a tentant-specific architecture and an eventually consistent model for data storage were proposed.

In a future work, a prototype of the proposed model will be implemented and evaluated. This includes not only the server-side components of the framework, but also the client-side. We will demonstrate by example, how third party online stores can integrate the social commerce service into their system. The evaluation of the prototype will concentrate on the non-functional requirements of the service, e.g., the scalability or the ease of integration.

References

[1] Z. Huang and M. Benyoucef, “From e-commerce to social commerce: A close look at design features,” Electronic Commerce Research and Applications, vol. 12, no. 4, 2013, pp. 246–259.

[2] J. H. Kietzmann, K. Hermkens, I. P. McCarthy, and B. S. Silvestre, “Social media? get serious! understanding the functional building blocks of social media,” Business Horizons, vol. 54, no. 3, 2011, pp. 241–251.

[3] C. Wang and P. Zhang, “The evolution of social commerce: An examination from the people, business, technology, and information perspective,” Communications of the AIS (CAIS), vol. 31, no. 1, 2012, pp. 105–127.

[4] D. Kailer, P. Mandl, and A. Schill, “Rated tags: Adding rating capability to collaborative tagging,” in 2013 IEEE Third International Conference on Social Computing and Its Applications 2013 (SCA 2013), Karlsruhe, Germany, 2013, pp. 249–255.

[5] J. Vig, M. Soukup, S. Sen, and J. Riedl, “Tag expression: Tagging with feeling,” in ACM Symposium on User Interface and Technology, 2010, pp. 323–332.

[6] S. E. Lee, D. K. Son, and S. Han, “Qtag: Tagging as a means of rating, opinion-expressing, sharing and visualizing,” in 25th Annual International Conference on Design of Communication, 2007, pp. 189–195.

[7] J. Yu and J. Ni, “Development strategies for sme e-commerce based on cloud computing,” in 2013 Seventh International Conference on Internet Computing for Engineering and Science (ICICSE), Sept 2013, pp. 1–8.

[8] X. Wang, “Research on e-commerce development model in the cloud computing environment,” in 2012 International Conference on System Science and Engineering (ICSSE), June 2012, pp. 539–542.

[9] H. L. Qing, “Research on the model and application of e-commerce based on cloud computing,” in 2012 International Conference on Computer Science Service System (CSSS), Aug 2012, pp. 142–146.

[10] G. Lackermair, “Hybrid cloud architectures for the online commerce,” Procedia Computer Science, vol. 3, no. 0, 2011, pp. 550–555, world Conference on Information Technology.

[11] A. B. Bondi, “Characteristics of scalability and their impact on performance,” in Proceedings of the 2Nd International Workshop on Software and Performance, ser. WOSP ’00. New York, NY, USA: ACM, 2000, pp. 195–203.

[12] M. Armbrust et al., “A view of cloud computing,” Communications of the ACM, vol. 53, no. 4, Apr. 2010, pp. 50–58.

[13] F. Liu et al., “NIST Cloud Computing Reference Architecture,” http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505 (retrieved: March, 2014), 2011.

[14] W.-T. Tsai, Q. Shao, Y. Huang, and X. Bai, “Towards a scalable and robust multi-tenancy saas,” in Proceedings of the Second Asia-Pacific Symposium on Internetware, ser. Internetware ’10. New York, NY, USA: ACM, 2010, pp. 8:1–8:15.

[15] Z. H. Wang et al., “A study and performance evaluation of the multi-tenant data tier design patterns for service oriented computing,” in IEEE International Conference on e-Business Engineering, 2008. ICEBE ’08, Oct 2008, pp. 94–101.

[16] W. Vogels, “Eventually consistent,” Communications of the ACM, vol. 52, no. 1, Jan. 2009, pp. 40–44.

[17] http://www.disqus.com (retrieved: March, 2014).

[18] B. Ippolito, “Remote json – jsonp,” http://bob.ippoli.to/archives/2005/12/05/remote-json-jsonp (retrieved: March, 2014), 2005.

[19] A. van Kesteren, “Cross-origin resource sharing: W3C Recommendation 16 January 2014,” http://www.w3.org/TR/2014/REC-cors-20140116 (retrieved: March, 2014), 2014.