Evaluating Peer-to-Peer for Loosely Coupled Business Collaboration: A Case Study

Fabian Stäber[1] and Jörg P. Müller[2]

[1] Siemens AG, Corporate Technology, Information and Communications
Otto-Hahn-Ring 6, Munich, 81739, Germany
[2] Clausthal University of Technology,
Julius-Albert-Str. 4, Clausthal-Zellerfeld, 38678, Germany

Abstract. Built-in support for self-organization, reliability, and decentralized management makes peer-to-peer an inherently suitable paradigm for loosely coupled business collaboration applications. However, current raw peer-to-peer algorithms are not sufficient to fulfill the requirements of distributed business process management. In this paper, we make the case for a generic service layer between peer-to-peer overlay and business application; we identify a number of important service layer components, and we evaluate these components with respect to requirements gathered from an industrial case study: automotive collaborative product development (CPD).

1 Introduction

Over the past few years, the peer-to-peer paradigm has been receiving broad attention in research and industry alike. Within the ATHENA IP[1], we have investigated the applicability of peer-to-peer protocols and architectures for a number of *collaborative business processes*, one of them being automotive collaborative product development (CPD, [1]). We have gathered further experience by building a peer-to-peer based Business Resource Management Framework (BRMF, [2]), and by applying BRMF to the automotive application. Studying business integration in the automotive industry, we learned that second tier suppliers join and leave the supplier network very dynamically. Handling the fluctuation of suppliers is a great challenge for business collaboration. A software platform enabling business integration among suppliers must support this churn.

Considering the capability of peer-to-peer systems to support easy to use plug-and-play networks in combination with resilience, reliability, decentralized management, and loosely coupled control, it seems that peer-to-peer technologies fit perfectly as a basis for implementing the type of dynamic collaboration processes as mentioned above. However, while the use of peer-to-peer technologies for business integration has been proposed in several research papers (e.g., [2,3]), peer-to-peer has not yet become a significant technology on the business applications market.

[1] http://www.athena-ip.org

G. Alonso, P. Dadam, and M. Rosemann (Eds.): BPM 2007, LNCS 4714, pp. 141–148, 2007.

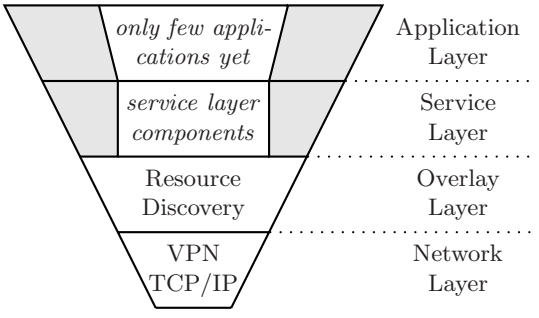

Fig. 1. Layers for Peer-to-Peer Based Applications

The reason for this becomes clear when decomposing peer-to-peer applications into different layers, as illustrated in Figure 1. The raw peer-to-peer overlay does not match the requirements of real-world business collaboration scenarios. Therefore, a service layer needs to be introduced, providing the functionality required by the application. However, although the overlay layer has been a research topic for several years, to our knowledge there is no significant related work addressing service layer components as independent building blocks that can be composed to meet the application requirements. Pushing this research forward is the key towards enabling collaborative business process management to benefit from peer-to-peer computing.

This paper is organized as follows. In the next section, we give the problem statement. In Section 3, we introduce the use case and present the required background in peer-to-peer computing. Section 4 identifies the requirements of the use case regarding the peer-to-peer service layer, and Section 5 presents the service layer components that can be used to meet the requirements. Finally, we summarize the interdependencies of the service layer components.

2 Problem Statement

The reason for the gap between the application requirements and the service layer is not that there are too few service layer components available. Rather, the problem is that current peer-to-peer projects provide monolithic, domain specific solutions, and do not distinguish generic service layer components. This makes it hard to benefit from peer-to-peer in new domains, like peer-to-peer based CPD applications.

In this paper, we analyze the requirements of a CPD scenario in the automotive industry and review the concepts behind current peer-to-peer projects with similar requirements. We extract generic, domain independent service layer components needed to implement our scenario, and analyze the interdependencies between these components.

Based on the analysis, we evaluate the feasibility of applying peer-to-peer technologies in CPD. The goal is to enable business collaboration to benefit

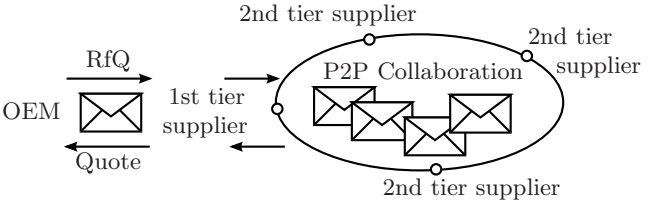

Fig. 2. P2P-Based CPD Scenario

from the self-organization and resilience offered by peer-to-peer systems. Our goal is to provide a novel view of peer-to-peer based applications, with a service layer as an independent layer, and to investigate interrelations between different generic service layer components are analyzed.

3 Background

In this section we briefly introduce the background necessary to identify the gap between the requirements of the CPD application and the services offered by the overlay layer. We first introduce the CPD scenario, and then briefly give an overview of peer-to-peer technologies.

3.1 Use Case Scenario

Figure 2 shows a Collaborative Product Development (CPD) scenario in the automotive industry, that was developed as part of the ATHENA project. A car manufacturer (OEM, Original Equipment Manufacturer) issues Requests for Quotations (RfQs) to its first tier suppliers. The engineers on the supplier side analyze the technical specifications in the RfQ and discuss them with the second tier suppliers. After this, the first tier supplier generates a proposal for alternative technical specifications and returns this proposal to the OEM. In turn, the OEM revises and updates its RfQ, and issues a new version. This negotiation cycle repeats until all parties agree on a feasible specification.

We learned that second tier suppliers join and leave the environment very dynamically. In this paper, we address the requirements on a business collaboration platform that is able to support the churn on the supplier side. The collaboration platform has two tasks. First, it must serve as a messaging platform, allowing the business partners to notify each other about new documents in a fluctuating environment. Second, the platform must serve as a data store, keeping the published technical specifications available in the face of churn.

3.2 Peer-to-Peer Technologies

In this section we will briefly introduce peer-to-peer background, and motivate why we focus on Distributed Hash Tables (DHTs) for the rest of this paper.

From an abstract point of view, a peer-to-peer overlay can be seen as a distributed routing protocol, mapping *keywords* to *peers* being responsible for the

given keywords. The most common use of peer-to-peer overlays is to build data sharing applications on top of them. A piece of data to be shared in a peer-to-peer application is called a *resource*. Each resource must be associated with one or more keywords that can be used to find the peers being responsible for the resource. In the simplest case, a keyword could be the filename of the resource.

In the CPD scenario presented above, the resources to be shared are business documents, like RfQs or Quotes. It is necessary to provide the recipients of these documents guaranteed access to the resources being available. The type of peer-to-peer overlays being able to provide guaranteed access is known as *Distributed Hash Tables* (DHTs). There are several DHT implementations, but conceptually all of them provide the same functionality. Some DHTs provide a unique one-to-one mapping between a single keyword and a unique peer being responsible for the keyword, and some of them provide a generic n-to-m mapping, yielding a set of responsible peers for a set of keywords.

In the rest of this paper, we will view the DHT as an abstract layer being able to look up peers for a given set of keywords. In Section 5 we show how to compose service layer components providing rich features on top of the raw peer-to-peer layer.

4 Application Requirements

Analyzing the CPD scenario, we identified eight requirements to be considered when evaluating the service layer components in Section 5. The choice of these requirements is based on the following considerations: First, we only consider requirements regarding the underlying service layer. More requirements can be found on the application layer, but these do not directly correspond to service layer components. Second, we only choose requirements of a generic nature, which means that these requirements can also be found in other application scenarios in a similar way. That way, we can benefit from ideas that are found in other peer-to-peer based applications.

Messaging. As shown in Section 3, the collaboration platform must not only serve as a data store for business documents, but also notify the respective business partners if documents are added, updated or removed. Doing so requires some messaging functionality.

Traffic Load Balancing. The peers in the peer-to-peer infrastructure are all operated by the participating business partners. Each peer acts as a router for other peers. The network traffic should be equally distributed among all peers. It must be avoided that a single peer is flooded with all the traffic.

Data Load Balancing. Each peer should store roughly the same amount of data.

Data Consistency. Due to the decentralized nature of peer-to-peer systems, there is no central instance defining which version of a document is the current one. Therefore, it is required that the collaboration infrastructure provides means for maintaining a consistent view of the versions of all resources.

Security. Business partners may be both, collaborators and competitors at the same time. Therefore, secure communication must be guaranteed, which means that the communication must be confidential, reliable and authenticated. Additionally, data stored in the network must be encrypted and resistant to malicious modifications or removal.

Resilience. We observe that in the use case business partners join and leave the supplier network very dynamically. The underlying infrastructure must catch up with this churn.

Rich Queries. There must be a way for business partners to describe documents they are interested in. The underlying collaboration platform must offer some rich query language for formulating complex queries.

Low Network Load. All the requirements above could be easily implemented if we had infinitely low delay and unbounded throughput. However, small suppliers often have limited bandwidth connections. Deploying the system in these environments must be feasible.

5 Evaluation of Service Layer Components

In this section, we evaluate service layer components clustered by their functionality. The choice of the service layer components is derived from the requirements we analyzed regarding the CPD scenario. As this is only a short paper, we restrict ourselves to giving a very brief survey of existing service layer solutions, and to providing an evaluation matrix giving an idea of how a methodology for evaluating the interrelationships between these service layer components should look like.

Subscriptions are used to notify business partners if documents of interest are added, updated, or removed. This is essential if the application does not only require a data store, but also messaging functionality. A survey of multicast solutions including references to related work can be found in [4]. Besides adding messaging functionality, subscriptions also reduce network traffic, as polling can be avoided. On the downside, certain security challenges are introduced, as the peer being responsible for a certain keyword learns who of its competitors is subscribed for that keyword.

Replication means that backup copies of the documents are stored on different peers in the peer-to-peer overlay. If the peer being responsible for a document fails, a backup peer can take over the responsibility, and the document remains available. Apart from increasing reliability, replication also fosters traffic load balancing, as there are more than one peers that can be queried for each document.

There are simple replication strategies, copying each resource to a fixed number of neighboring peers, and there are more sophisticated replication strategies, adapting on the popularity of the documents. A survey and evaluation of different strategies can be found in [5].

Fuzzy Hashing is a way to avoid *hotspots* in terms of data load, i.e. to relieve peers suffering from too much data to be stored. This is achieved by replacing the strict mapping of resources to keywords with an adaptable, fuzzy

mapping [6,7]. Apart from fostering load balancing, fuzzy hashing also has a positive effect on security, because with fuzzy hashing no peer has the full control of all resources with a certain keyword.

Consensus Protocols must be combined with replication strategies in order to avoid concurrent modifications. If a document is replicated in the peer-to-peer overlay, it cannot be guaranteed that all modifications to the document will be consistent among all the copies of the document. Using consensus protocols, atomic transactions can be implemented [8,9,10].

Additionally, the use of consensus protocols has a positive impact on security, because a single peer can be kept from tampering with documents.

Confidential Communication is essential in the CPD scenario, as the suppliers may be both, cooperators and competitors at the same time. Confidential communication must fulfill three properties: Sender authentication, content encryption, and anonymous communication paths, securing that intermediate peers on the communication path cannot learn who is communicating with whom.

Implementing confidential communication on the service layer requires that the peer-to-peer overlay is built on top of a *virtual private network*, (VPN), providing a *public key infrastructure* (PKI). The unique PKI certificates can be used to prevent Sybil attacks [11], and the public key encryption can be used to implement Onion Routing [12], providing anonymous communication paths.

Redundant Paths are used to prevent attacks on the DHT's routing mechanism. The confidential communication introduced above a secure environment for the CPD scenario, but it relies on the overlay's lookup algorithm to work correctly. The reliability of the lookup mechanism can be increased using redundant lookup paths [13]. Additionally, redundant lookup paths have a positive impact on traffic load balancing, as there is no single lookup paths being a potential bottleneck.

Search Indexes are used to implement rich queries on top of the peer-to-peer overlay's trivial keyword lookup mechanism. We identified four major technologies to implement high level queries: Ontologies, as applied in the Edutella project [14], SQL, as implemented in the PIER project [15], XPath, as in the Active XML project [16], and index servers providing full text search, as with Lucene [17]. Although the related work on distributed ontologies, SQL, and XPath looks promising, one must always keep in mind that queries that affect a large number of peers do not scale. Therefore, distributed query languages will always be restricted to a subset of their non-distributed counterparts.

6 Conclusions and Outlook

Figure 3 shows an evaluation matrix of the components introduced in Section 5. The + and − signs stand for positive or negative impact on the corresponding requirement area. The results are the following:

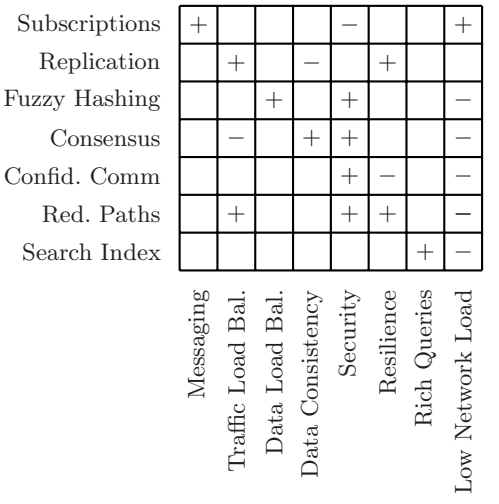

Fig. 3. Benefits and Conflicts of Service Layer Components with Requirements

1. All the requirements we identified in the CPD scenario are supported by one or more service layer components.
2. There is no one-to-one mapping of service layer components and requirements. Most service layer components have impact on several requirements.
3. All service layer components have positive and negative impacts. While some requirements are fulfilled using these components, other requirements are corrupted. That means that applying any of these service layer components is always a trade-off.

The evaluation framework presented in this paper provides one step towards enabling collaborative business processes to benefit from the self-organization and resilience of decentralized peer-to-peer systems. Our next steps are to apply our view of the service layer to other scenarios and other requirements. The comparison of the results will result in an evaluation of what interrelations on the service layer are triggered by specific use case requirements, and what interrelations are more universal.

References

1. Stäber, F., Sobrito, G., Müller, J.P., Bartlang, U., Friese, T.: Interoperability challenges and solutions in automotive collaborative product development. In: Proc. of the 3rd International Conference on Interoperability for Enterprise Software and Applications, Enterprise Interoperability, vol. 2, pp. 709–720. Springer, London (2007)
2. Friese, T., Müller, J.P., Smith, M., Freisleben, B.: A robust business resource management framework based on a peer-to-peer infrastructure. In: CEC '05: Proc. of the 7th International IEEE Conference on E-Commerce Technology, pp. 215–222. IEEE Press, Los Alamitos (2005)

3. Androutsellis-Theotokis, S., Spinellis, D., Karakoidas, V.: Performing peer-to-peer e-business transactions: A requirements analysis and preliminary design proposal. In: Karmakar, N., Isaías, P. (eds.) Proc. of the IADIS International e-Commerce 2004 Conference, IADIS Press, pp. 399–404 (2004)
4. Abad, C.L., Yurcik, W., Campbell, R.H.: A survey and comparison of end-system overlay multicast solutions suitable for network-centric warfare. In: Suresh, R. (ed.) Battlespace Digitization and Network-Centric Systems IV. SPIE, vol. 5441, pp. 215–226 (2004)
5. Leslie, M., Davies, J., Huffman, T.: Replication strategies for reliable decentralised storage. In: ARES '06: Proc. of the First International Conference on Availability, Reliability and Security, pp. 740–747 (2006)
6. Mitzenmacher, M.: The power of two choices in randomized load balancing. IEEE Transactions on Parallel and Distributed Systems 12(10), 1094–1104 (2001)
7. Rieche, S., Petrak, L., Wehrle, K.: A thermal-dissipation-based approach for balancing data load in distributed hash tables. In: LCN '04: Proc. of the 29th Annual IEEE International Conference on Local Computer Networks, pp. 15–23. IEEE Computer Society Press, Los Alamitos (2004)
8. Lamport, L., Shostak, R., Pease, M.: The byzantine generals problem. TOPLAS: ACM Transactions on Programming Languages and Systems 4(3), 382–401 (1982)
9. Lamport, L.: Paxos made simple. SIGACT News 32(4), 18–25 (2001)
10. Muthitacharoen, A., Gilbert, S., Morris, R.: Etna: a fault-tolerant algorithm for atomic mutable dht data. Technical Report MIT-LCS-TR-993, MIT Computer Science and Artificial Intelligence Laboratory (2005)
11. Douceur, J.R.: The sybil attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)
12. Stäber, F., Bartlang, U., Müller, J.P.: Using onion routing to secure peer-to-peer supported business collaboration. In: Proc. of eChallenges 2006, Exploiting the Knowledge Economy: Issues, Applications and Case Studies, vol. 3, pp. 181–188. IOS Press, Amsterdam (2006)
13. Srivatsa, M., Liu, L.: Vulnerabilities and security threats in structured overlay networks: a quantitative analysis. In: ACSAC 2004, pp. 252–261. IEEE Press, Los Alamitos (2004)
14. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., Risch, T.: Edutella: A p2p networking infrastructure based on RDF. In: WWW '02: Proc. of the 11th Int. World Wide Web Conference, pp. 604–615 (2002)
15. Huebsch, R., Chun, B., Hellerstein, J., Loo, B.T., Maniatis, P., Roscoe, T., Shenker, S., Stoica, I., Yumerefendi, A.R.: The architecture of PIER: an internet-scale query processor. In: CIDR '05: Proc. of the 2nd Conference on Innovative Data Systems Research, pp. 28–43 (2005)
16. Abiteboul, S., Benjelloun, O., Milo, T.: The active xml project: an overview. Technical Report 331, Gemo, INRIA-Futurs (10, 2005)
17. Gospodnetić, O., Hatcher, E.: Lucene in Action. Manning Publications Co., Greenwich, CT (2005)