Web of Needs: A Process Overview

Florian Kleedorfer, Soheil Human, Heiko Friedrich
Research Studio Smart Agent Technologies
Research Studios Austria
Thurngasse 8
A-1090 Wien, Austria
florian.kleedorfer, soheil.human, heiko.friedrich@researchstudio.at

Christian Huemer
Institute for Software Technology and Interactive Systems
Vienna University of Technology
Favoritenstrasse 9-11
A-1040 Wien, Austria
huemer@big.tuwien.ac.at

ABSTRACT

The Web of Needs (WoN) is an approach for expressing and publishing human needs on the Internet as linked data to allow automatic matching of the expressed needs and communication between users who expressed them. In this paper we summarize the processes realized by the different architectural components and explain the decision to use a linked data based messaging subsystem.

CCS Concepts

•Information systems → World Wide Web; •Human-centered computing → Collaborative and social computing systems and tools;

Keywords

linked data, electronic marketplaces, messaging

1. INTRODUCTION

This work is an extension to our publication on verifiability of the message exchange in the Web of Needs (WoN) [5], providing details on the application that uses this messaging approach.

The main idea of WoN is to allow users to publish objects that indicate intention to interact. These objects are referred to as *needs*; they allow users to be informed when the intended interaction may be possible, and support have such interactions by facilitating the exchange of messages between needs. [4]. Using linked data for the description of the intentions has the advantage that the semantics of the data is portable across different participants and it creates a unified, decentralized data view of the whole content on the Web. However, in the application scenarios we envisage, the descriptions of needs (intentions) are only one part of the content; the actual messages that are exchanged are at least of the same importance because the state of each interaction is defined by the description of the needs and the set of all messages that have been exchanged. In order to provide said unified view not only of the needs, but also of the communication, a linked data based messaging system was designed such that messages (payload and metadata) are formulated in RDF [6] and made available as linked data.

© 2016 Copyright held by the author/owner(s).
SEMANTICS 2016: Posters and Demos Track
September 13-14, 2016, Leipzig, Germany

The messaging aspect is integrated with the application in such a way that all interaction with it is realized via messages - not only, as one might assume, user-to-user communication. All state changes - creation and manipulation of need objects and the management of their connections with each other - are all done using linked data messages. The messages are cryptographically signed by their senders and all intermediate relays; their signatures are linked in chains or trees [7], so this approach generates an audit trail of all state changes and conversations, guarding against the manipulation of published data and of messages exchanged. This provenance data is part of the data layer of the system, which has the advantage that the same tools and methods can be used to create, analyze, or store the data that are used for managing the main application content.

2. ARCHITECTURAL OVERVIEW

Figure 1 shows a deployment diagram of the Web of Needs. It depicts a distributed network that consists of three different types of nodes: *WoN nodes*, *owner applications*, and *matching services*. WoN nodes, or for short *nodes*, are the data services that relay messages and serve linked data. Needs (intentions), created in the owner applications, are stored on WoN nodes. Matching services search for pairs of needs that are suitable for an interaction, and if such a pair is identified, they send a *hint* message to each of the needs.

If a user finds the remote need indicated in a hint message suitable for an interaction, they can open a connection from their need to the suggested need.

For example, consider a typical classified ads use case: user Alice is looking for a power drill for some home improvement work. User Bob wants to sell a power drill. Both users can publish their intentions on the WoN. A matching service suggests that the two connect, which they eventually do. They organize the transaction by exchanging chat messages.

Finding matches for a given need is a crucial functionality of the Web of Needs, and the usefulness of the system depends greatly on the quality of the matches. The matching task is challenging, in most cases, it is domain specific, and the idea of developing a single monolithic matching service that could fulfill all expectations is unrealistic. To allow for free evolution of good matching, WoN was designed as an open system to support any number of matching services. In the demonstrator application[1] [2], two matching approaches

[1]The latest stable version is available at https://www.

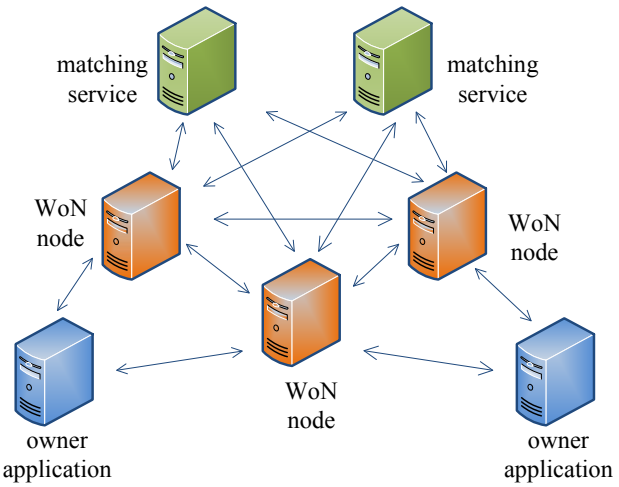

Figure 1: Deployment diagram indicating types of nodes and communication paths.

are implemented:

1. *Online Matching.* This approach is required for providing results as fast as possible after a need has been published. In order to allow for this, WoN nodes offer updates about needs in the form of publish-subscribe queues[2]. Matching services can subscribe to these events and be updated whenever a new need is created. The matcher was built using an information retrieval approach [3] by indexing all encountered needs and querying that index with queries generated for each new need.

2. *Offline Matching.* While some use cases require instantaneous results, it is acceptable under some circumstances that matches are discovered at a later point in time. Such situations allow a matching service to analyze a larger number of needs in bulk, which is what is required, for example, for a tensor factorization approach we applied for detecting new matches [8].

3. PROCESSES IN THE WEB OF NEEDS

In order to provide an overview of the functionality provided by WoN, we illustrate the main processes using the Business Process Model and Notation (BPMN) graphical representation [1] in Figure 2. The process includes three pools, one for each type of service or application.

1. *Owner Application.* This denotes a mobile, desktop, or Web application that offers a graphical user interface to access the Web of Needs. The pool shown here represents some of the very basic functions of the owner application that are related to the matching system:

   (a) **Manage needs.**

   - **Create need.** When a need is created in the owner application, a CREATE message, which includes the need's RDF description, is sent to a WoN node.
   - **Deactivate need.** When a user no longer wants to receive any messages on behalf of a need, a DEACTIVATE message is sent to the WoN node hosting the need.
   - **Activate need.** When a user wants to reactivate a deactivated need, an ACTIVATE message is sent to the WoN-node hosting the need.

   (b) **Manage hints.**

   - **Show hints to user.** When matches are found, and the respective HINT messages are received from matching services (and forwarded by the WoN node), the owner application shows them to the user.
   - **Send feedback for matchers.** Users have the option to give feedback on the relevance of the matches in order to improve the matching system. The owner application constructs a HINTFEEDBACK message and sends it to the WoN node that holds the need's data. Feedback data is published along with the data describing the connection between two needs. It is intended for use by matching services to improve the matching quality over time.

   (c) **Manage connections to other needs.**

   - **Connect with remote need.** When a hint is received, the user can decide to establish a communication channel with the remote need. This is done by sending a CONNECT message.
   - **Accept or deny a remote need's connection request.** When a CONNECT message is received, the user can decide to establish a communication channel with the remote need. This is done by sending a CONNECT message. If the communication is not desired, a CLOSE message is sent.
   - **Exchange messages.** When a connection is established, CONNECTIONMESSAGE messages containing arbitrary RDF graphs as payload can be exchanged between needs.
   - **Disconnect from remote need.** When no more communication is desired, a CLOSE message is sent to the remote need via the WoN node.

2. *WoN Node* The following processes are executed on the service that holds all the data.

   - **Create need.** When a CREATE message is received from an owner application, the WoN node extracts the need description and stores it in its database. Subsequently, a message indicating the creation of the new need is published and consumed by the clients subscribed to the respective topic.
   - **Process a matcher's hint.** When matches are found, they are sent as HINT messages to the WoN node hosting the respective need. The node stores

matchat.org/

[2]In the current implementation, these are managed by an ActiveMQ message broker; however, other messaging subsystems that support publish-subscribe can be used as well.

them in its database, creating a new *Connection* entry between the need and its matched remote counterpart. The HINT message is sent to the owner application to be shown to the user.

- **Store feedback for matchers.** User feedback is stored in the database of the node.
- **Process and relay messages on behalf of the needs.** When a message is received from an owner application or a remote node (e.g., a CONNECT or CONNECTIONMESSAGE), the message is stored locally. If, according to the communication protocol[3], the message causes server-side actions, these are executed (e.g. a state change to a need or connection). Subsequently, the message is forwarded to its destination.

3. *Matching Service* This pool shows the processes typically performed by the matching service. It consists of two main modules (represented as lanes): a) node management b) need management c) hint generation and sending.

   (a) **Node Management.** The matching service tries to keep up to date with data in WoN. It does this by executing the following processes.
   - **Discover new WoN node.** When the matching service discovers a new WoN node, it creates an entry about it in its database.
   - **Subscribe to receive updates when needs are created.** The matching service identifies the message queues for being updated about new needs and subscribes to them.
   - **Crawl existing needs.** The matching service crawls newly discovered WoN nodes.

   (b) **Need management.**
   - **Discover new need.** New needs are discovered either through crawling or as a result of receiving a message from a WoN node's publish-subscribe queue the matching service is subscribed to. The need data is downloaded and stored in an RDF store.
   - **React to need state changes.** Whenever a state change is detected, the matching service updates the RDF data associated with the need and decides whether to react by calculating new matches.

   (c) **Hint generation and sending.**
   - **Execute online matcher.** Whenever a new need is encountered, the online matcher is activated. In the demonstrator application, the online matcher is realized using an information retrieval approach [3].
   - **Execute offline matcher.** At regular intervals, offline matchers can be started that perform bulk calculations and produce hints after a prolonged period of time. This matcher reads from the RDF store containing the needs known by the matching service.
   - **Store hints.** Hints are stored in a database when they are created. This allows for aggregating hints and for avoiding sending duplicate hints or hints of lower relevance than those that have already been sent.
   - **Send hints.** The hints are sent to the WoN nodes hosting the respective needs.

[3]The basic communication protocol and extensions thereof are not explained in detail here.

These processes and the ontologies used to describe the application data in RDF[4] provide a framework suitable for many content domains. Domain specific applications are realized by owner applications and matching services that can handle the data model (ontologies or vocabularies) used in the respective domain. To cover the motivating use case out of the box, the demonstrator application comes with a basic ontology for the classified ads domain.

4. CONCLUSION

In this work we provide an overview of the functionality realized based on a linked data messaging architecture in the Web of Needs. We list the distinct processes executed by the services that together provide the functionality of connecting users (or other agents) based on their intentions and enable interactions between them.

5. ACKNOWLEDGMENTS

This work was supported by the Austrian Research Promotion Agency (FFG) in the COIN project *USS WON – Usability, Scalability and Security on the Web of Needs.*

6. REFERENCES

[1] Business process model and notation (BPMN), version 2.0. Object Management Group, January 2011.

[2] Web of Needs on Github. https://github.com/researchstudio-sat/webofneeds/, 2016.

[3] Renaud Delbru. SIREn: Entity retrieval system for the web of data. In *Proceedings of the 3rd Symposium on Future Directions in Information Access (FDIA)*, 2009.

[4] Florian Kleedorfer and Christina Maria Busch. Beyond data: Building a web of needs. In *Proceedings of the WWW2013 Workshop on Linked Data on the Web (LDOW 2013)*, 2013.

[5] Florian Kleedorfer, Yana Panchenko, Christina Maria Busch, and Christian Huemer. Verifiability and traceability in a linked data based messaging system. In *Proceedings of the 12th International Conference on Semantic Systems*, Leipzig, Germany, September 2016. to appear.

[6] Frank Manola and Eric Miller. Rdf primer, 2004. [Last accessed on 2016/06/06].

[7] Ralph C. Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology - CRYPTO'87*, pages 369–378. Springer, 1987.

[8] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280. ACM, 2012.

[4]See http://purl.org/webofneeds/model and http://purl.org/webofneeds/message

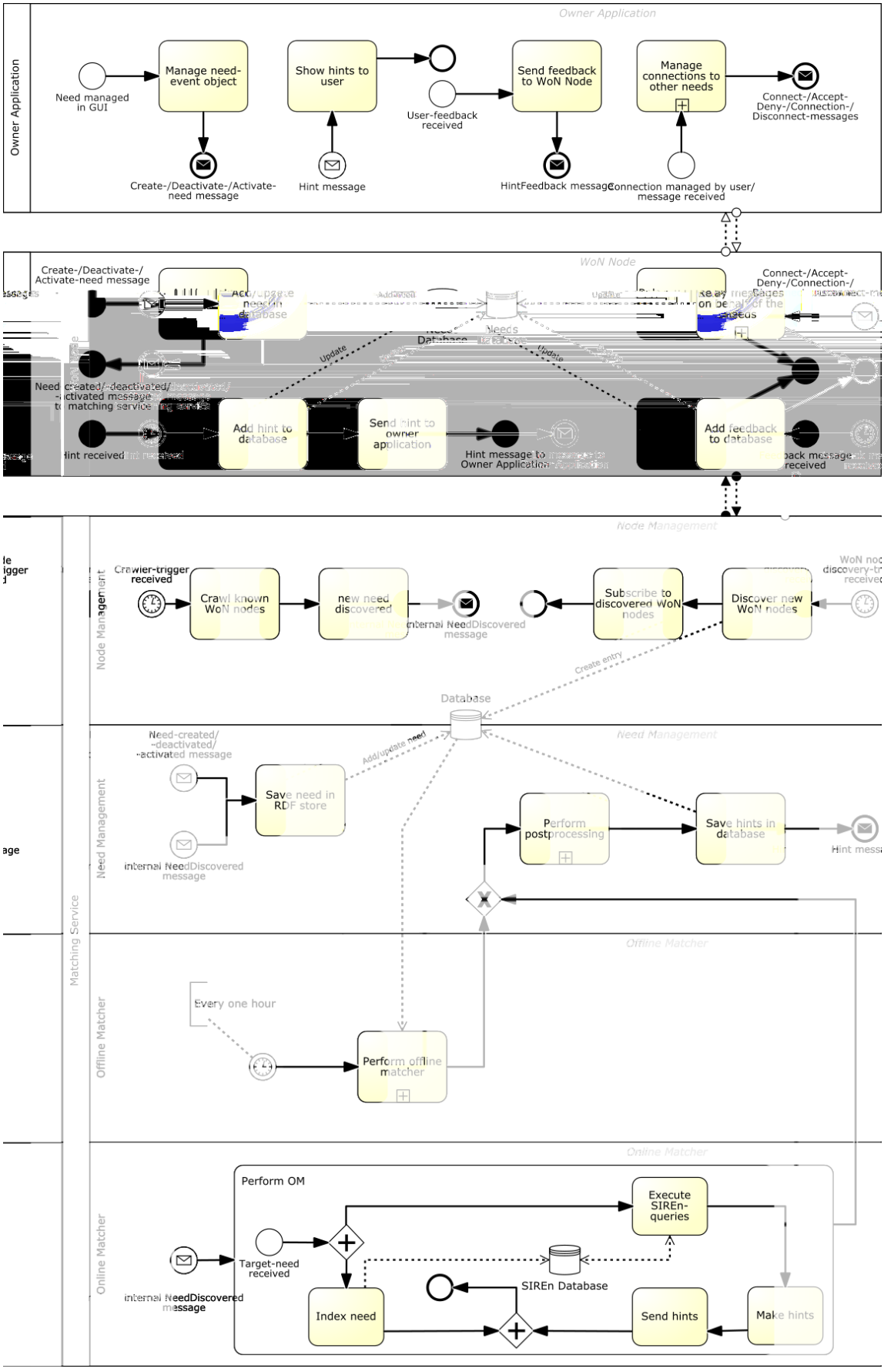

Figure 2: Main processes in the Web of Needs, organized by service.