Federated Search and Recommendation

Dileepa Jayakody[a], Nirojan Selvanathan[a] and Violeta Damjanovic-Behrendt[a]

[a] *Salzburg Research, Jakob Haringer Strasse 5/II, 5020 Salzburg, Austria*

Abstract

This paper presents an index-time merging approach to federated search designed for a European Connected Factory Platform for Agile Manufacturing (EFPF) project and its federated digital ecosystem. The presented approach supports search and recommendations for products, services as well as suitable business partners for the creation of networks within the business ecosystem. We discuss the overall architecture of the search and recommendation system, major challenges and the current state of development.

Keywords 1

Federated search, recommendation, digital platforms, platform search

1. Introduction

Federated search systems emerged in the late 1990s in order to support library users to simultaneously search multiple library sources and catalogues [1]. Early federated search systems were designed to collect search criteria from users, submit search queries to sources to be retrieved, wait for a response to arrive, and filter and organize the query results before sending the results back to the users. For example, Windows 7 services differentiate between two types of search, Windows Search and Federated Search [2]. Windows Search enables searching of the entire local system and a limited amount of external resources such as shared drives. Likewise, the selection-based search in MacOS Spotlight requires an index of all items and files that exist in the system to be created.

By contrast, Federated Search enables the simultaneous search of multiple sources, including document repositories, databases, remote repositories and external Web sites. To enable federated features for websites to be effectively searched by the user, search connectors need to be implemented and configured for each website [2].

With the maturing of Cloud Computing, Distributed Machine Learning (DML), big data, fog and edge computing technologies, new opportunities for exploiting computing resources on demand emerged, creating huge possible impacts for various sectors. However, implementing such federated systems is still a challenging task due to cross-cutting requirements related to search and distributed data management, data governance and policies that need to be simultaneously implemented and administered.

The core issue related to federated search is about resource accessibility, which is required to create one large index from those multiple sources that constitute the federation. Federated search allows for each source to be indexed locally and searched across many sources. Another issue of federated search is about the relevance of results that are obtained from a variety of sources. The relevance of results needs to be synchronized and compared across sources, in order to merge the results in a meaningful way. In practice, even the results from the same source are not comparable when different search engines are used to retrieve these sources.

In this paper, we present the design of federated search and recommendation systems for digital platforms that is currently ongoing work in the EFPF (European Connected Factory Platform for Agile Manufacturing) project. The aim of EFPF is to create a federation of several digital platforms, while overcoming the interoperability challenges between different platforms and their services.

Proceedings of the Workshops of I-ESA 2020, 17-11-2020, Tarbes, France

EMAIL: dileepa.jayakody@salzburgresearch.at

2. Literature review

The authors in [3] differentiate between federated search and metasearch. Metasearch engines crawl open websites and return results of all kinds, such as apps, webpages, contacts and documents that are drawn from different sources. In contrast to metasearch, federated search systems focus on subscribed databases that are processed through an indexing mechanism that enables structured, in-depth, content-oriented retrieval [4]. A number of attempts to implement federated search arose from the 1990s onwards, e.g. MetaLib and SFX implemented at Boston College [5], MetaLib and SFX with Ex Libris at the Five Colleges Libraries in Massachusetts [6], Georgia Library Learning Online (GALILEO) with its hybrid approach to federated search products [7], and more. Many interviews showed that the users were dissatisfied with federated search performance of that time, due to limitations in database availability, search speed, retrieval precision, and result comprehensiveness [8]. Users preferred simple interfaces over federated search and had little interest in advanced searching techniques. In recent years, several attempts to improve federated search were conducted. In 2004, the University of Nevada at Reno experimented with the Google Search Appliance to be used as a federated search utility. This required partnership between Google, the library and a test vendor, and it still experienced technical and interoperability challenges [9].

3. Problem statement

Federated search is a technique for simultaneous search of multiple data sources using a single query and search interface. The examples of federated search can be found in many sectors, facilitating search of thousands of products, each sorted into different categories, e.g. search of numerous websites serving various purposes and stakeholders [11]. The majority of the research on federated search is focused on system performance and technical development, system usability testing, user interaction, interface customization, authentication, design, database communication protocol, system platform [10].

The two fundamental components of federated search systems comprise of an index that is a reference to the data to be searched and a search algorithm. These two components interact using either search-time, index-time or a federated search interface [11]:

- Search-time merging runs separate search algorithms on each data source that is searched. It uses multiple indices and aggregates the returned search results into a final list, which is then presented to the user. Obtaining the results can be slow, if the central search engine needs to wait until all of the local search engines have responded.
- Index-time merging requires a central index to be built for all of the data that need to be included in the search results. It requires one search engine and one index, and is faster than search-time merging. To aggregate data from multiple sources and formats into a single index, interoperability issues need to be addressed.
- Federated search indexing is an extension of the search-time merging method that requires a robust search solution in order to index different types of content in different indices and create the unified federated search interface.

4. Federated search and recommendation in EFPF

Our solution to a federated search and recommendation system, designed in EFPF, includes the following tree steps: firstly, in order to achieve a technical federation of digital platforms, we design a common federated search capability. Secondly, we collect the results of federated search to support federated recommendations. Finally, both federated search and recommendations support advanced matchmaking and agile network creation mechanisms for business transactions across the digital platform ecosystem. Federated search in EFPF supports queries for products, services and business partners. The search criteria and the results are collected to support the recommendation processes, based on different techniques of information pattern matching, e.g. information retrieval and similarity matching techniques, which are both based on Machine Learning (ML) and data analysis. After the recommendation algorithm has identified the most suitable products, services and/or partners, the user evaluates the results based on selected indicators, e.g. cost, reliability, quality, etc. In the final step, the user makes a decision and initiates a business transaction.

5. Design of a federated search and recommendation system

In EFPF, federated search is based on the index-time merge architectural approach (Figure 1). This approach requires content from all base platforms participating in the federation, to be channeled into a central index of the EFPF platform. The four base platforms in EFPS are NIMBLE (https://www.nimble-project.org/), COMPOSITION (https://www.composition-project.eu/), vf-OS (https://www.vf-os.eu/) and DIGICOR (https://www.digicor-project.eu/). The EFPF Data Spine is designed as an integration flow engine, enabling toolsets and services from the base platforms to connect with each other and offers their capabilities to the users of the different platforms. For the implementation of the Federated Search Index, we use Apache Solr (https://lucene.apache.org/solr/) that enables distributed indexing, replication and load-balanced querying, automated failover and recovery, configuration, etc.

One major disadvantage of the index-time merge architecture concerns the data connectors that need to be implemented to integrate different types of data sources. In addition, acquiring the content from different repositories and data sources requires considerable efforts; e.g., it needs to be done using scheduled read-only processes that need to be designed and implemented at the data integration layer. This also requires a decision about the frequency of the data ingestion into the central index. Data ingestion frequency needs to be configured e.g. hourly, daily or weekly, depending on the data velocity of the base platforms.

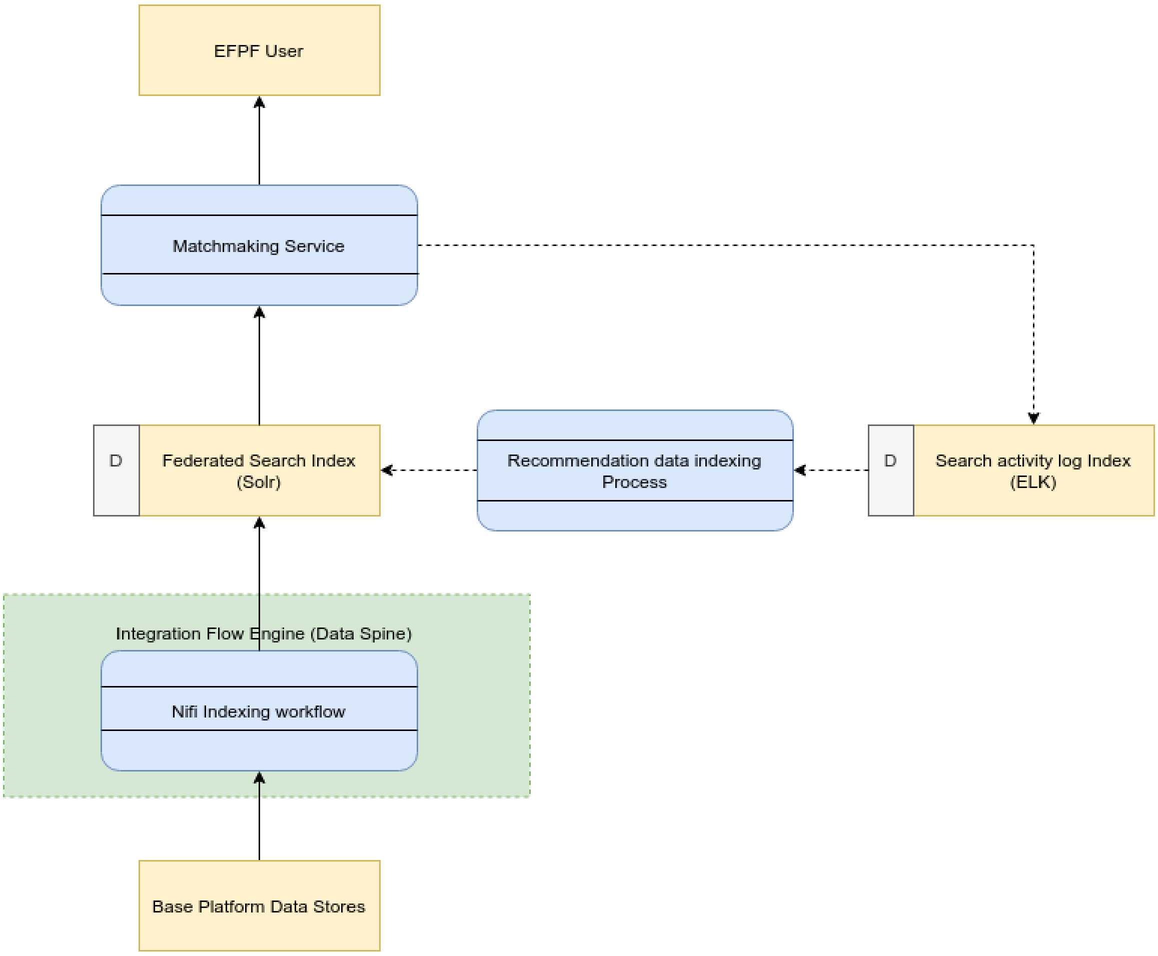
EFPF User
Matchmaking Service
D
Federated Search Index (Solr)
Recommendation data indexing Process
D
Search activity log Index (ELK)
Integration Flow Engine (Data Spine)
Nifi Indexing workflow
Base Platform Data Stores

Figure 1: The index-time merge architecture in EFPF

The user interaction data in EFPF is stored using the User Activity Log Service, which listens to all user interaction events generated from the EFPF portal (https://efpf-portal.ascora.eu/), e.g. purchased items, etc. The user interaction data are also stored using Solr Index, in order to support ML model creation. The outcomes of the matchmaking are presented to the EFPF users through an intuitive UI that is integrated in the portal.

6. Conclusion

Federated search, ML and recommendation techniques can be combined in an innovative way to deliver better results for businesses, improving user experience, increasing user engagement through incentives and rewards mechanisms, boosting conversion rates and fully operationalizing businesses using digital platform solutions. However, there are still many questions to be explored in the intersection of these fields, e.g.: Which recommendation algorithms are more suitable for federated platform ecosystems? What is the minimum amount of data to run an effective recommendation system? How symbolic AI methods can compensate for the missing data and better support recommendation? This paper has presented the main challenges and the initial design of the federated search architecture. The above questions will be further answered over the course of the project implementation. In 2021, an open call for third-party use of the platform infrastructure will offer the opportunity for large-scale validation of the search and recommendation system.

7. Acknowledgements

This research has been co-funded by the European Commission within the H2020 project NIMBLE, No. 723810 and the H2020 DT-ICT-07-2018-2019 project EFPF, No. 825075.

8. References

[1] A. C. Elguindi, K. Schmidt, "Discovery systems, layers and tools, and the role of the electronic resources librarian". Electronic Resource Management, 2012, p. 109-139.

[2] J. Orchilles, "Introduction to Windows 7", Microsoft Windows 7 Administrator's Reference, p. 1–91, doi:10.1016/b978-1-59749-561-5.00001-2, 2010.

[3] X. Fei, "Implementation of a Federated Search System: Resource Accessibility Issues", Serials Review – SER REV. 35, 2009, p. 235-241.

[4] R. Tang, I., Hsieh-Yee, S. Zhang, "User Percep. of MetaLib Comb. Search: An Inv. of How Users Make Sense of Federated Search," Intern. Ref. Serv. Q. 12, 2007, p. 211–236.

[5] B. Geritty, T. Lyman, E. Tallent, "Blurring Services and Resources: Boston College's Implementation of MetaLib and SFX", Ref. Serv. Rev.30, no.3, 2002, p. 229–241.

[6] L.S. Mestre C. Turner, B. Lang, B. Morgan, "Do We Step Together, in the Same Direction, at the Same Time? How a Consortium Approached a Federated Search Implementation", Internet Reference Services Quarterly 12, no. 1/2, 2007, p. 111–132.

[7] L. Fancher, "Wanted, Dead or Alive: Federated Searching for a Statewide Virtual Library", Internet Reference Services Quarterly 12, no. 1/2, 2007, p. 133–158.

[8] K. Calhoun, "An Integrated Framework for Discovering Digital Library Collections", Journal of Zhejiang University Science 6A, no. 11, 2005, p. 1318–1326.

[9] M. Taylor, "Using the Google Search Appliance for federated searching", Internet Reference Services Quarterly, 10(3), 2006, p. 45–55.

[10] D.G. Dorner, A.M. Curtis, "A Comparative Review of Common User Interface Products," Library Hi Tech 22, no. 2, 2004, p. 182–197.

[11] Algolia, "What is Federated Search?", URL: https://blog.algolia.com/what-is-federated-search/, 2019.