Caring is Sharing – Exploiting the Value in Data for Health and Innovation
M. Hägglund et al. (Eds.)

doi:10.3233/SHTI230113

Monitoring Distributed Business Processes in Biomedical Research

Christoph ZILSKE[a], Maximilian KURSCHEIDT[a,1], Simon T SCHWEIZER[a], Hauke HUND[a], Simon MÖDINGER[a] and Christian FEGELER[a]
[a] *GECKO Institute, Heilbronn University of Applied Sciences, Heilbronn, Germany*

Abstract. In biomedical research, business processes, such as data-sharing or feasibility queries, span across several healthcare organizations. Due to the growing number of data-sharing projects and connected organizations, the management of distributed processes gets more complex over time. This leads to an increased need for administrating, orchestrating, and monitoring all distributed processes of a single organization. A proof of concept for a decentralized and use case agnostic monitoring dashboard was developed for the Data Sharing Framework, which most German university hospitals have deployed. The implemented dashboard can handle current, changing, and upcoming processes using only information for cross-organizational communication. This differentiates our approach from other existing use case specific content visualizations. The presented dashboard is a promising solution to provide administrators with an overview of the status of their distributed process instances. Therefore, this concept will be further developed in upcoming releases.

Keywords. Visualization, Decentralized Monitoring, Distributed Business Process, Data Sharing

1. Introduction

Large amounts of medical routine data are documented across different healthcare organizations, which provides great potential for biomedical research. To consolidate and analyze this data, distributed processes such as feasibility studies [1], record linkage, and data sharing [2] need to be implemented between bio-medical research organizations. The requirements to facilitate such distributed processes include process orchestration, technical connectivity, and data protection. In this context, the Data Sharing Framework (DSF) was introduced as a use case independent communication infrastructure for executing cross-organizational distributed processes. Currently, the DSF is deployed at approx. 40 organizations in production, including most German university hospitals and several other organizations across multiple research consortia, such as the *Medical Informatics Initiative* (MII) [3,4] and the *Network University Medicine* (NUM) [5].

The ever-growing number of federated data-sharing projects, respective processes, and participating organizations result in increased communication relationships that require a higher degree of automation. This trend leads to a situation where monitoring the status, tasks, and errors of distributed processes becomes more challenging for users

[1] Corresponding Author: Maximilian Kurscheidt, GECKO Institute, University of Applied Sciences Heilbronn, Max-Planck-Straße 39, 74081 Heilbronn, Germany; E-mail: maximilian.kurscheidt@hs-heilbronn.de.

and administrators. Due to changing requirements and a distributed development community, the exact design of future processes is unknown. With this article, we want to answer the question if a decentralized dashboard is suitable for monitoring the status of distributed processes. Therefore, a proof of concept was implemented.

1.1. Data Sharing Framework

The concept of the DSF represents a peer-to-peer network where no central entity exists, and each organization using the DSF has various communication partners executing distributed processes. The DSF was implemented using commonly known standards such as FHIR R4 and BPMN 2.0. It consists of a FHIR server, accessible by external communication partners, acting as a "mailbox"; and a private Business Process Engine (BPE) executing BPMN 2.0 processes to interact with local and remote systems [3].

The decentralized authentication, authorization, and role definition specify the relationship between organizations and are handled through the DSF-*AllowList*[2], which consists of O*rganization*, *OrganizationAffiliation*, and *Endpoint* FHIR resources.

FHIR *Task* resources are used to start and continue processes and include mandatory attributes in the DSF context: requester; recipient; status of the process; a business key; and the name and version of the process that should be triggered. Input parameters can be used to configure processes and output parameters to save results.

Organizations can read *Task* resources stored on remote FHIR servers if they have created them. They can also access all *Task* resources stored on their local FHIR server. Each *Task* is associated with a specific process instance by their business key, which is shared across all *Task* resources. During the execution of a process, *Task* resources are updated with the status, changing from *requested* to *in-progress* to *completed* or *failed*.

2. Materials and Methods

Requirements for the proof of concept were conducted by four expert interviews, where interviewees had multiple roles (DSF core developers, plugin developers, technical support engineers and system administrators). The interviews included questions regarding the functionality, interactions, challenges and suggested improvements of the DSF and processes. Distributed processes from the research consortia MII and NUM were analyzed regarding communication patterns, entity roles and structure. Additionally, protocols of past DSF hackathons were reviewed for common challenges.

The dashboard was developed for the DSF using common web technologies, such as HTML, CSS, and the JavaScript framework React.js[3]. The DSF was used in version 0.9[2], widely deployed at German university hospital *data integration centers* (DIC).

3. Results

Most organizations communicate simultaneously with other organizations in multiple projects through different processes via the DSF communication infrastructure. An

[2] https://github.com/highmed/highmed-dsf/tree/v0.9.0 (accessed March 3, 2023)

[3] https://reactjs.org (accessed March 3, 2023)

example of such a peer-to-peer network constellation from the perspective of *Organization A* with the role DIC is displayed in *Figure 1*. Each of the edges represents a line of communication associated with a distributed process, distinguished by color.

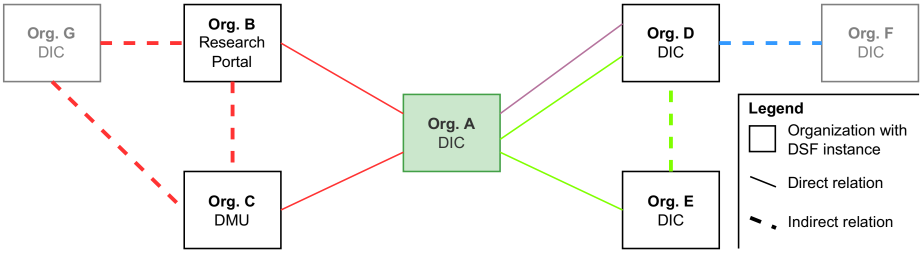

Figure 1. Simplified distributed communication infrastructure with multiple distributed processes. DIC: *Data integration center*; DMU: *Data management Unit*.

The red communication lines in *Figure 1* are derived from the data sharing process of the MII[4] . A simplified version of the BPMN 2.0 process is displayed in *Figure 2* and the model shows the interaction pattern, including multiple subprocesses from different roles. FHIR *Task* resources are represented through the colored BPMN message flows.

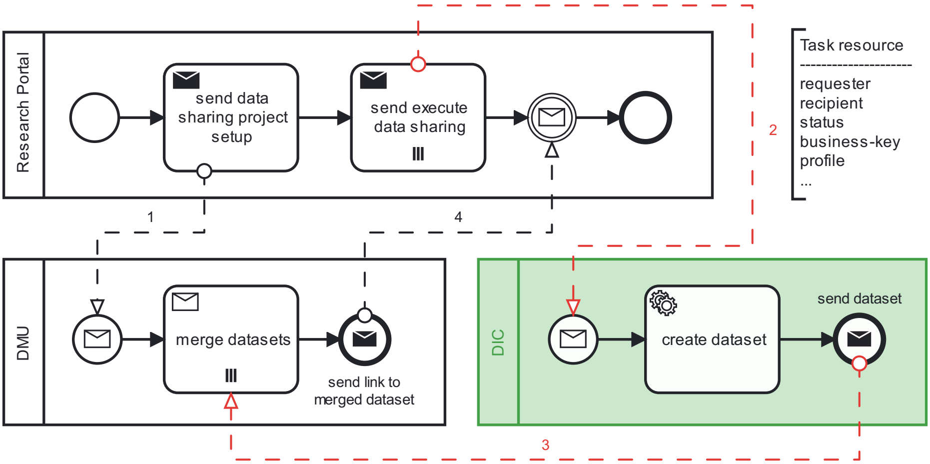

Figure 2. Simplified data sharing process between the roles Research Portal, Data Management Unit (DMU), and Data Integration Center (DIC).

The DIC, in this example, receives a *Task* resource from the *Research Portal* and sends a *Task* resource to the *Data Management Unit* (DMU). Focusing on the perspective of the DIC, only two *Task* resources marked *2* and *3* are accessible for the DIC. *Task* resources *1* and *4* are relevant for the overall process but are executed between other organizations and outside of the DIC's scope. This concept is consistent across all processes developed for the DSF.

The interviewees stated the following key requirements for the dashboard: all deployed processes with their corresponding process instances need to be presented; the

[4] https://github.com/medizininformatik-initiative/mii-dsf-processes/wiki (accessed March 3, 2023)

dashboard should be use case and process agnostic, deployed locally and independent from other participants; received and requested *Task* resources and their associated process instances need to be presented; *Task* resources and process instances should be filterable and searchable.

The information included in the *Task* resources was used to visualize the state of a distributed process in our monitoring dashboard. *Figure 3* shows the implemented monitoring dashboard from the perspective of the DIC referenced in *Figure 1* and *Figure 2*. The elements presented in *Figure 3 (section a)* allow the user to select between different distributed processes deployed on the DSF. If a primary process is selected, all corresponding process instances grouped by their individual business key are listed in *Figure 3 (section b).*

In *Figure 3*, the simplified data sharing process *from Figure 2* is selected. Selecting a process instance opens an overview of the received and sent *Task* resources marked by the icons in *Figure 3 (section c*). They present their status, which gets aggregated to the overall status of the process instance. *Task* resources and process instances can be sorted and filtered by different properties such as status, requester, or date of creation. This functionality allows a structured overview of the status of process instances. The *Task* resource can be selected to view the entire content.

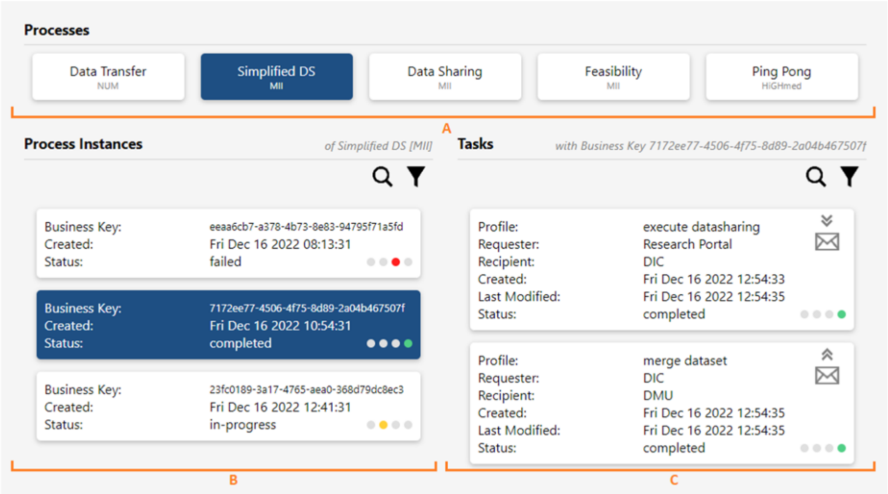

Figure 3. Monitoring dashboard: ***Section a***: Main processes deployed on the DSF. ***Section b***: Process instances of the selected process. ***Section c***: Associated task resources.

4. Discussion

We implemented a decentralized dashboard prototype to monitor the state of distributed processes. It bundles relevant information and visualizes the status of process instances for the system administrator. This eliminates the need to analyze *Task* resources and log files manually. The peer-to-peer architecture of the DSF enables organizations to execute individual processes with their distributed communication partners. No central entity oversees and coordinates the execution of process instances for all organizations. Thus, a decentralized monitoring dashboard was implemented, which enables every organization to operate and monitor their processes independently.

A use case independent approach was chosen to monitor known and unknown processes. This differentiates our approach from centralized use case specific dashboards (e.g., MII *Forschungsdatenportal für Gesundheit* [6]). These are developed to represent detailed process content but are difficult to adapt to other processes. Our approach switches the perspective from use case specific dashboards to an organization-oriented dashboard that visualizes all distributed processes associated with their DSF instance. No adaptations of existing processes or the DSF are needed for our implementation. Moreover, no modification of the specifically secured BPE component is required. Our monitoring approach uses only required *Task* resource attributes, representing communication metadata. An overview of the distributed process was achieved by grouping and labeling the information available within these resources. The level of detail might still be insufficient for users, depending on their role in the process. In future, through a standardized interface, additional information about a process could be integrated into the dashboard.

5. Conclusion

In this article, we presented a proof of concept for a decentralized monitoring dashboard for visualizing distributed processes executed on the communication infrastructure - Data Sharing Framework. Using only information contained in FHIR *Task* resources for cross-organizational communication, we generalized the dashboard functionality to handle existing and future distributed processes. This design decision makes the presented proof-of-concept use case-, process-, project- and consortia independent. The presented decentralized dashboard is a promising solution to provide administrators with an overview of the status of their distributed process instances that are parallelly deployed in their DSF. The dashboard will be further developed as part of the *DSF-Community-Project* during the MII phase 3 grant.

Acknowledgments

The project is funded by the German Federal Ministry of Education and Research (BMBF, grant id: 01ZZ1802E).

References

[1] R. Wettstein et al., Secure Multi-Party Computation Based Distributed Feasibility Queries – A HiGHmed Use Case. Stud Health Technol Inform. 296 (2022) 41-49. doi:10.3233/SHTI220802

[2] R. Wettstein et al. Data Sharing in Distributed Architectures – Concept and Implementation in HiGHmed. Stud Health Technol Inform. 283 (2021) 111-118. doi:10.3233/SHTI210548

[3] H. Hund et al. Executing Distributed Healthcare and Research Processes – The HiGHmed Data Sharing Framework. Stud. Health Technol. Inform. 278 (2021) 126-133. doi:10.3233/SHTI210060

[4] P. Knaup et al., Implementation of a National Framework to Promote Health Data Sharing, Yearb Med Inform, 27.01 (2018) 302-304. doi: 10.1055/s-0038-1641210

[5] H.U. Prokosch et al. The COVID-19 Data Exchange Platform of the German University Medicine. Stud. Health Technol. Inform. 294 (2022) 674–678. doi:10.3233/SHTI220554

[6] J. Gruendner et al. The Architecture of a Feasibility Query Portal for Distributed COVID-19 Fast Healthcare Interoperability Resources (FHIR) Patient Data Repositories: Design and Implementation Study. JMIR Med Inform 10:5 (2022). doi: 10.2196/36709