Kadi4Mat: A Research Data Infrastructure for Materials Science

RESEARCH PAPER

NICO BRANDT
LARS GRIEM
CHRISTOPH HERRMANN
EPHRAIM SCHOOF
GIOVANNA TOSATO
YINGHAN ZHAO
PHILIPP ZSCHUMME
MICHAEL SELZER

*Author affiliations can be found in the back matter of this article

ABSTRACT

The concepts and current developments of a research data infrastructure for materials science are presented, extending and combining the features of an electronic lab notebook and a repository. The objective of this infrastructure is to incorporate the possibility of structured data storage and data exchange with documented and reproducible data analysis and visualization, which finally leads to the publication of the data. This way, researchers can be supported throughout the entire research process. The software is being developed as a web-based and desktop-based system, offering both a graphical user interface and a programmatic interface. The focus of the development is on the integration of technologies and systems based on both established as well as new concepts. Due to the heterogeneous nature of materials science data, the current features are kept mostly generic, and the structuring of the data is largely left to the users. As a result, an extension of the research data infrastructure to other disciplines is possible in the future. The source code of the project is publicly available under a permissive Apache 2.0 license.

CORRESPONDING AUTHOR:
Nico Brandt
Institute for Applied Materials (IAM-CMS), Karlsruhe Institute of Technology (KIT), Straße am Forum 7, 76131 Karlsruhe, Germany
nico.brandt@kit.edu

KEYWORDS:
research data management; electronic lab notebook; repository; open source; materials science

TO CITE THIS ARTICLE:
Brandt, N, Griem, L, Herrmann, C, Schoof, E, Tosato, G, Zhao, Y, Zschumme, P and Selzer, M. 2021. Kadi4Mat: A Research Data Infrastructure for Materials Science. *Data Science Journal*, 20: 8, pp. 1–14. DOI: *https://doi.org/10.5334/dsj-2021-008*

1 INTRODUCTION

In engineering sciences, the handling of digital research data plays an increasingly important role in all fields of application (Sandfeld et al. 2018). This is especially the case, due to the growing amount of data obtained from experiments and simulations (Hey & Trefethen 2003). The extraction of knowledge from these data is referred to as a data-driven, fourth paradigm of science, filed under the keyword data science (Hey 2009). This is particularly true in materials science, as the research and understanding of new materials are becoming more and more complex (Hill et al. 2016). Without suitable analysis methods, the ever-growing amount of data will no longer be manageable. In order to be able to perform appropriate data analyses smoothly, the structured storage of research data and associated metadata is an important aspect. Specifically, a uniform research data management is needed, which is made possible by appropriate infrastructures such as research data repositories. In addition to uniform data storage, such systems can help to overcome inter-institutional hurdles in data exchange, compare theoretical and experimental data and provide reproducible workflows for data analysis. Furthermore, linking the data with persistent identifiers enables other researchers to directly reference them in their work.

In particular, repositories for the storage and internal or public exchange of research data are becoming more and more widespread. Especially the publication of such data, either on its own or as a supplement to a text publication, is increasingly encouraged or sometimes even required (Naughton & Kernohan 2016). In order to find a suitable repository, services such as re3data (Pampel et al. 2013) or FairSharing (The FAIRsharing Community et al. 2019) are available. These services also make it possible to find subject-specific repositories for materials science data. Two well-known examples are the Materials Project (Jain et al. 2013) and the NOMAD Repository (Drax & Scheffler 2018). Indexed repositories are usually hosted centrally or institutionally, and are mostly used for the publication of data. However, some of the underlying systems can also be installed by the user, such as for internal use within individual research groups. Additionally, this allows full control over stored data as well as internal data exchanges, if this function is not already part of the repository. In this respect, open-source systems are particularly important, as this means independence from vendors and opens up the possibility of modifying the existing functionality or adding additional features, sometimes via built-in plug-in systems. Examples of such systems are Ckan (CKAN Association 2014), Dataverse (King 2007), DSpace (Smith et al. 2003) or Invenio (CERN 2016), where the latter is the basis of Zenodo (CERN & OpenAIRE 2013). The listed repositories are all generic and represent only a selection of the existing open-source systems (Amorim et al. 2017).

A second type of system in addition to the repositories, which is also increasingly used in experimentally oriented research areas, are the electronic lab notebooks (ELN) (Rubacha, Rattan & Hosselet 2011). Nowadays, the functionality of ELNs goes far beyond the simple replacement of paper-based lab notebooks, and can also include aspects such as data analysis, as seen, for example, in Galaxy (Afgan et al. 2018) or Jupyter Notebooks (Kluyver et al. 2016). Both systems focus primarily on providing accessible and reproducible computational research. Specifically, the boundary between unstructured and structured data is more and more blurred, the latter being traditionally only found in laboratory information management systems (LIMS) (Bird, Willoughby & Frey 2013; Elliott 2009; Taylor 2006). Most existing ELNs are domain-specific and limited to research disciplines such as biology or chemistry (Taylor 2006). According to current knowledge, a system specifically tailored to materials science does not exist. For ELNs, there are also open-source systems such as eLabFTW (CARPi, Minges & Piel 2017), SciNote (SciNote LLC 2015) or Chemotion (Tremouilhac et al. 2017). Compared to the repositories, however, the selection of ELNs is smaller. Furthermore, only the first two mentioned systems are generic.

Thus, generic research data systems and software are available for both ELNs and repositories, which, in principle, could also be used in materials science. The listed open-source solutions are of particular relevance, as they can be adapted to different needs and are generally suitable for use in a custom installation within single research groups. However, both aspects can be a considerable hurdle, especially for smaller groups. Due to a lack of resources, a structured research data management and the possibility of making data available for subsequent use is therefore particularly difficult for such groups (P. Bryan Heidorn 2008). What is finally missing is a system that can be deployed and used both centrally and decentrally, as well as internally and publicly, without major obstacles. The system should support researchers throughout the entire research process, starting with the generation and extraction of raw data, up to the structured storage, exchange and analysis of the data, resulting in the final publication of the corresponding results. In this way, the features of the ELN and the repository are combined, creating a virtual research environment (Carusi & Reimer 2010) that accelerates the generation of innovations by facilitating the collaboration between researchers. In an interdisciplinary field like materials sciences, there is a special need to model the very heterogeneous workflows of the researchers (Hill et al. 2016).

For this purpose, the research data infrastructure Kadi4Mat (Karlsruhe Data Infrastructure for Materials Sciences) is being developed at the Institute for Applied Materials (IAM-CMS) of the Karlsruhe Institute of Technology (KIT). The current logo of the project is shown in ***Figure 1***. The aim of the software is to combine the possibility of structured data storage with documented and reproducible workflows for data analysis and visualization tasks, incorporating new concepts with established technologies and existing solutions. In the development of the software, the FAIR principles (Wilkinson et al. 2016) for scientific data management are taken into account. Instances of the data infrastructure have already been deployed and show how structured data storage and data exchange are made possible (Brandt 2020). Furthermore, the source code of the project is publicly available under a permissive Apache 2.0 license (Brandt et al. 2020).

Figure 1 Logo of Kadi4Mat.

2 CONCEPTS

Kadi4Mat is logically divided into the two components ELN and repository, which have access to various tools and technical infrastructures. The components can be used by web- and desktop-based applications, via uniform interfaces. Both a graphical and a programmatic interface are provided, using machine-readable formats and various exchange protocols. In ***Figure 2***, a conceptual overview of the infrastructure of Kadi4Mat is presented.

2.1 ELECTRONIC LAB NOTEBOOK

In the ELN component, the so-called *workflows* are of particular importance. A workflow is a generic concept that describes a well-defined sequence of sequential or parallel steps, which are processed as automatically as possible. This can include the execution of an analysis tool or the control and data retrieval of an experimental device. To accommodate such heterogeneity, the concrete steps must be implemented as flexibly as possible, since they are highly user- and application-specific. In ***Figure 2***, the types of tools shown in the second layer are used as part of the workflows, so as to implement the actual functionality of the various steps. These can be roughly divided into analysis, visualization, transformation and transportation tasks. In order to keep the application of these tools as generic as possible, a combination of provided and user-defined tools is accessed. From a user's perspective, it must be possible to provide such tools in an easy manner, while the execution of each tool must take place in a secure and functional environment. This is especially true for existing tools, e.g. a simple MATLAB (The MathWorks, Inc. 2021) script, which require certain dependencies to be executed and must be equipped with a suitable interface to be used within a workflow. Depending on their functionality, the tools must in turn access various technical infrastructures. In addition to the use of the repository and computing infrastructure, direct access to devices is also important for more complex data analyses. The automation of a typical workflow of experimenters is only fully possible if data and metadata, created by devices, can be captured. However, such an integration is not trivial, due to a heterogeneous device landscape and proprietary data formats and interfaces (Hawker 2007; Potthoff et al. 2019). In Kadi4Mat, it should also be possible to use individual tools separately, where appropriate, i.e. outside a workflow. For example, a visualization tool for a custom data format may be used to generate a preview of a datum that can be directly displayed in a web browser, when using the web-based interface.

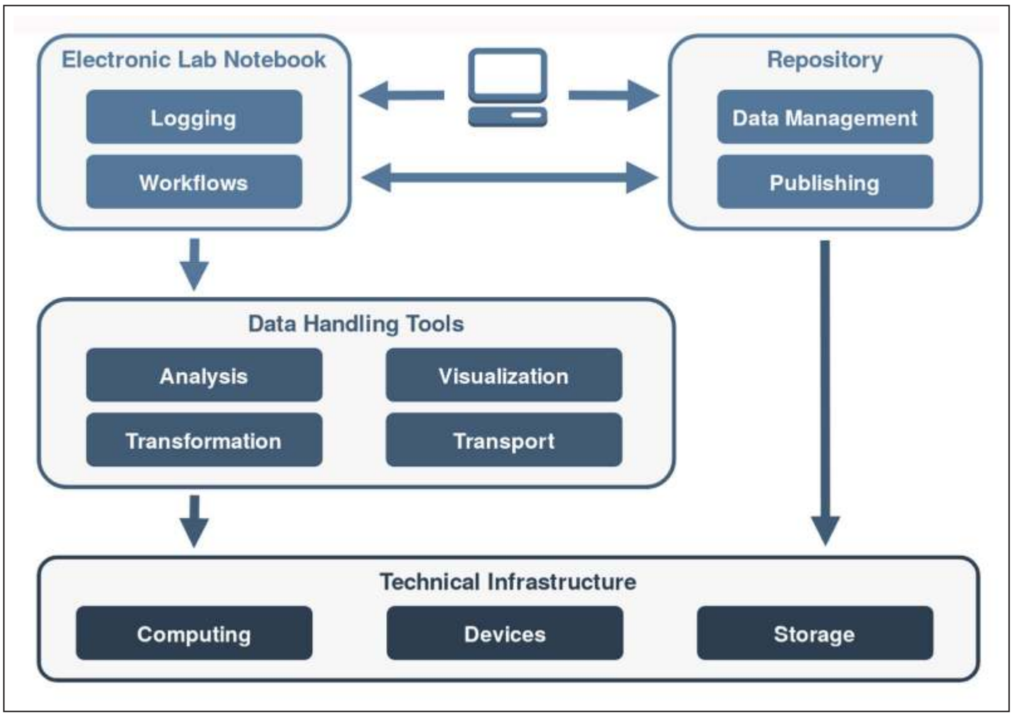


Figure 2 Conceptual overview of the infrastructure of Kadi4Mat. The system is logically divided into the two components ELN and repository, which have access to various data handling tools and technical infrastructures. The two components can be used both graphically and programmatically via uniform interfaces.

In *Figure 3*, the current concept for the integration of the workflows in Kadi4Mat is shown. Different steps of a workflow can be defined with a graphical node editor. Either a web-based or a desktop-based version of such an editor can be used, the latter running as an ordinary application on a local workstation. With the help of such an editor, the different steps or tools to be executed are defined, linked and, most importantly, parameterized. The execution of a workflow can be started via an external component called *process manager*. This component in turn manages several *process engines*, which take care of executing the workflows. The process engines potentially differ in their implementation and functionality. A simple process engine, for example, could be limited to a sequential execution order of the different tasks, while another one could execute independent tasks in parallel. All engines process the required steps based on the information stored in the workflow. With appropriate transport tools, the data and metadata required for each step, as well as the resulting output, can be exported or imported from Kadi4Mat, using the existing interfaces of the research data infrastructure. With similar tools, the use of other external data sources becomes possible, and with it the possibility to handle large amounts of data via suitable exchange protocols. The use of locally stored data is also possible when running a workflow on a local workstation.

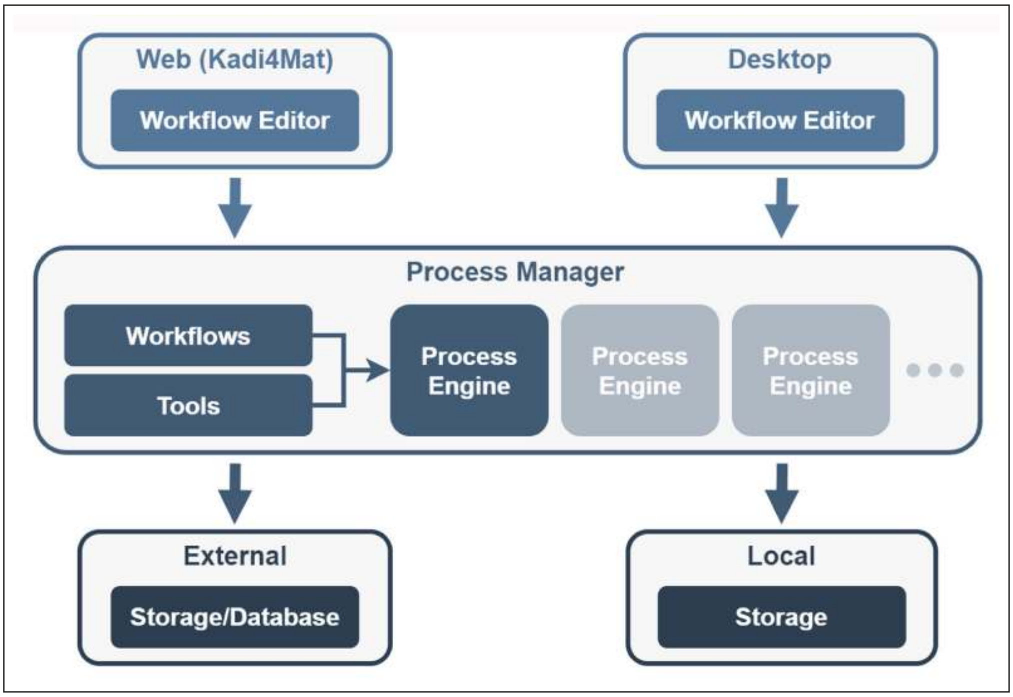


Figure 3 Conceptual overview of the workflow architecture. Each workflow is defined using a graphical editor that is either directly integrated into the web-based interface of Kadi4Mat or locally, with a desktop application. The process manager provides an interface for executing workflows and communicates on behalf of the user with multiple process engines, to which the actual execution of workflows is delegated. The engines are responsible for the actual processing of the different steps, based on the information defined in a workflow. Data and metadata can either be stored externally or locally.

Since the reproducibility of the performed steps is a key objective of the workflows, all meaningful information and metadata can be logged along the way. The logging needs to be flexible, in order to accommodate different individual or organizational needs, and therefore is also part of the workflow itself. Workflows can also be shared with other users, for example via Kadi4Mat. Manual steps may require interaction during the execution of a workflow, for which the system must prompt the user. In summary, the focus of the ELN component thus points in a different direction than in classic ELNs, with the emphasis on the automation of the steps performed. This aspect in particular is similar to systems such as Galaxy (Afgan et al. 2018), which focuses on computational biology, or Taverna (Wolstencroft et al. 2013), a dedicated workflow management system. Nevertheless, some typical features of classic ELNs are also considered in the ELN component, such as the inclusion of handwritten notes.

2.2 REPOSITORY

In the repository component, data management is regarded as the central element, especially the structured data storage and exchange. An important aspect is the enrichment of data with corresponding descriptive metadata, which is required for its description, analysis or search. Many repositories, especially those focused on publishing research data, use the metadata schema provided by DataCite (DataCite Metadata Working Group 2019), and are either directly or heavily based on it. This schema is widely supported and enables the direct publication of data, via the corresponding DataCite service. For use cases that go beyond data publications, it is limited in its descriptive power, at the same time. There are comparatively few subject-specific schemas available for engineering and material sciences. Two examples are EngMeta (Schembera & Iglezakis 2020) and NOMAD Meta Info (Ghiringhelli et al. 2017). The first schema is created a priori and aims to provide a generic description of computer-aided engineering data, while the second schema is created a posteriori, using existing computing inputs and outputs from the database of the NOMAD repository.

The second approach is also pursued in a similar way in Kadi4Mat. Instead of a fixed metadata schema, the concrete structure is largely determined by the users themselves, and thus is oriented towards their specific needs. To aid with establishing common metadata vocabularies, a mechanism to create templates is provided. Templates can impose certain restrictions and validations on certain metadata. They are user-defined and can be shared within workgroups or projects, facilitating the establishment of metadata standards. Nevertheless, individual, generic metadata fields, such as a title or description of a data set, can be static. For different use cases such as data analysis, publishing or the interoperability with other systems, additional conversions must be provided. This is not only necessary because of differing data formats, but also to map vocabularies of different schemas accordingly. Such converted metadata can either represent a subset of existing schemas or require additional fields, such as a license for the re-use of published data. In the long run, the objective in Kadi4Mat is to offer well-defined structures and semantics, by making use of ontologies. In the field of materials science, there are ongoing developments in this respect, such as the European Materials Modelling Ontology (EMMC 2019). However, a bottom-up procedure is considered as a more flexible solution, with the objective to generate an ontology from existing metadata and relationships between different data sets. Such a two-pronged approach aims to be functional in the short term, while still staying extensible in the long term (Greenberg et al. 2009), although it heavily depends on how users manage their data and metadata with the options available.

In addition to the metadata, the actual data must be managed as well. Here, one can distinguish between data managed directly by Kadi4Mat and linked data. In the simplest form, the former resides on a file system accessible by the repository, which means full control over the data. This requires a copy of each datum to be made available in Kadi4Mat, which makes it less suitable for very large amounts of data. The same applies to data analyses that are to be carried out on external computing infrastructures and must access the data for this purpose. Linked data, on the other hand, can be located on external data storage devices, e.g. high-performance computing infrastructures. This also makes it possible to integrate existing infrastructures and repositories. In these cases, Kadi4Mat can simply offer a view on top of such infrastructures or a more direct integration, depending on the concrete system in question.

A further point to be addressed within the repository is the publication of data and metadata, including templates and workflows, that require persistent identifiers to be referenceable.

Many existing repositories and systems are already specialized in exactly this use case and offer infrastructures for the long-term archiving of large amounts of data. Thus, an integration of suitable external systems is to be considered for this task in particular. From Kadi4Mat's point of view, only certain basic requirements have to be ensured in order to enable the publishing of data. These include the assignment of a unique identifier within the system, the provision of metadata and licenses, necessary for a publication, and a basic form of user-guided quality control. The repository component thus also goes in a different direction than classic repositories. In a typical scientific workflow, it is primarily focused on all steps that take place between the initial data acquisition and the publishing of data. The component is therefore best described as a *community repository* that manages *warm* data, i.e. unpublished data that needs further analysis, and enables data exchange within specific communities, for example within a research group or project.

3 IMPLEMENTATION

Kadi4Mat is built as a web-based application that employs a classic client-server architecture. A graphical front end is provided to be used with a normal web browser as a client, while the server is responsible for the handling of the back end and the integration of external systems. A high-level overview of the implementation is shown in ***Figure 4***. The front end is based on the classic web technologies JavaScript, HTML and CSS. In particular, the client-side JavaScript web framework Vue.js (Vue Core Development Team 2014) is used. The framework is especially suitable for the creation of single-page web applications (SPA), but can also be used for individual sections of more classic applications, to incrementally add complex and dynamic user interface components to certain pages. Vue.js is mainly used for the latter, the benefit being a clear separation between the data and the presentation layer, as well as the easier re-use of user interface components. This aspect is combined with server-side rendering. Due to the technologies and standards employed, the use of the front end is currently limited to recent versions of modern web browsers such as Firefox, Chrome or Edge.

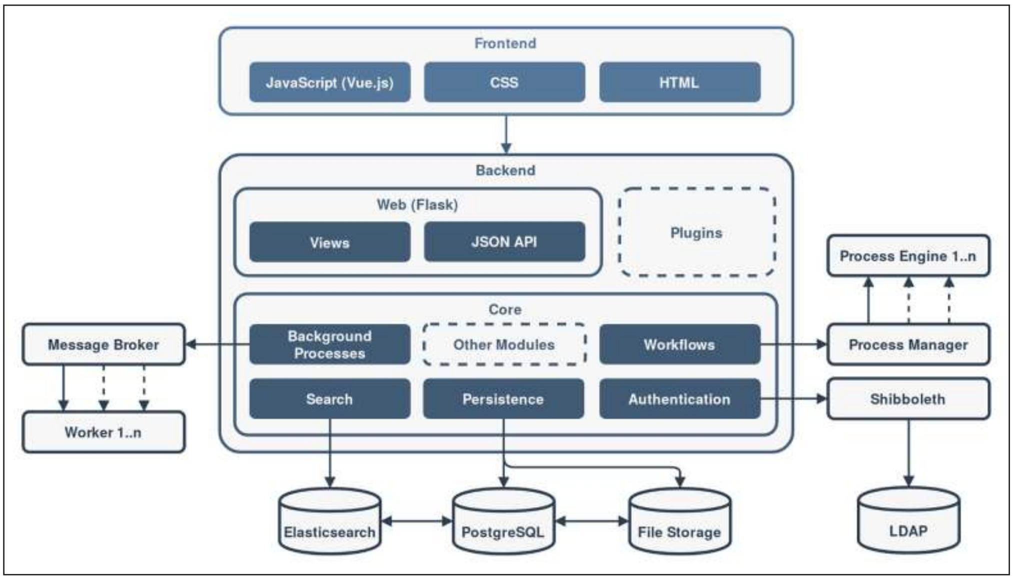


Figure 4 Overview of the implementation of Kadi4Mat, separated into front end and back end. The front end uses classic web technologies and is usually operated via a web browser. In the back end, the functionality is split into the web and the core component. The former takes care of the external interfaces, while the latter contains most of the core functionality and handles the interfaces of other systems. A plugin component is also shown, which can be used to customize or extend the functionality of the system.

In the back end, the framework Flask (The Pallets Projects 2015) is used for the web component. The framework is implemented in Python and is compatible with the common web server gateway interface (WSGI), which specifies an interface between web servers and Python applications. As a so-called microframework, the functionality of Flask itself is limited to the basic features. This means that most of the functionality, which is unrelated to the web component, has to be added by custom code or suitable libraries. At the same time, more freedom is offered in the concrete choice of technologies. This is in direct contrast to web frameworks such as Django (DSF 2005), which already provides a lot of functionality from scratch. The web component itself is responsible for handling client requests for specific endpoints and assigning them to the appropriate Python functions. Currently, either HTML or JSON is returned, depending on the endpoint. The latter is used as part of an HTTP API, to enable an internal and external programmatic data exchange. This API is based on the representational state transfer (REST) paradigm (Fielding & Taylor 2000). Support for other exchange formats could also be relevant in the future, particularly for implementing certain exchange formats for interoperability, such as OAI-PMH (OAI 2015). Especially for handling larger amounts of data, other exchange protocols besides HTTP are considered.

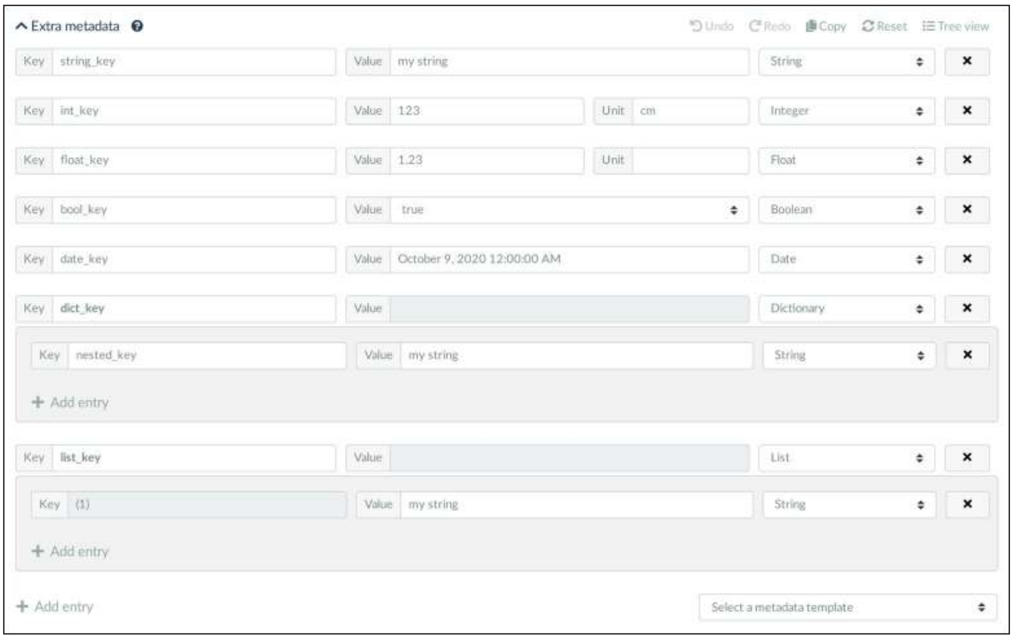
A large part of the application consists of the core functionality, which is divided into different modules, as shown in ***Figure 4***. This structure is mainly of an organizational nature. A microservice architecture is currently not implemented. Modules that access external components are particularly noteworthy, which is an aspect that will also be increasingly important in the future. External components can either run on the same hardware as Kadi4Mat itself or on separate systems available via a network interface. For the storage of metadata, the persistence module makes use of the relational database management system PostgreSQL (PostgreSQL Global Development Group 1996), while the regular file system stores the actual data. Additionally, the software Elasticsearch (Elastic NV 2010) is used to index all the metadata that needs to be efficiently searchable. The aforementioned process manager (Zschumme 2021b), which is currently implemented as a command line application, manages the execution of workflows by delegating each execution task to an available process engine (Zschumme 2021a). While the current implementation of the process engine primarily uses the local file system of the machine on which it is running, users can add steps to synchronize data with the repository to their workflow at will. To increase performance with multiple parallel requests for workflow execution, the requests can be distributed to process engines running on additional servers. By wrapping the process manager with a simple HTTP API, for example, its interface can easily be used over a network. A message broker is used to decouple longer running or periodically executed background tasks from the rest of the application, by delegating them to one or more background worker processes. Apart from using locally managed user accounts or an LDAP system for authentication, Shibboleth (Cantor & Seavo 2005) can be used as well. From a technical point of view, Shibboleth is not a single system, but the interaction of several components, which together enable a distributed authentication procedure. Depending on the type of authentication, user attributes or group affiliations can also be used for authorization purposes in the future.

Another component shown in ***Figure 4*** are the plugins. These can be used to customize or extend the basic functionality of certain procedures or actions, without having to modify or know the corresponding implementation in detail. Unlike the tools in a workflow, plugins make use of predefined hooks to add their custom functionality. While such a plugin has to be installed centrally by the system administrator for all users of a Kadi4Mat instance, the possibilities are also evaluated to be able to make use of individual plugins on the user level.

4 RESULTS

The current functionalities of Kadi4Mat can either be utilised via the graphical user interface, with a browser, or via the HTTP API, with a suitable client. On top of the API, a Python library is developed, which makes it especially easy to interact with the different functionalities (Schoof & Brandt 2020). Besides using the library in Python code, it offers a command line interface, enabling the integration with other programming or scripting languages.

In the following, the most important features of Kadi4Mat are explained, based on its graphical user interface. The focus of the features implemented so far is on the repository component, the topics of structured data management and data exchange in particular, as well as on the workflows, which are a central part of the ELN's functionality. After logging in to Kadi4Mat, it is possible to create different types of resources. The most important type of resource are the so-called *records*, which can link arbitrary data with descriptive metadata and serve as basic components that can be used in workflows and future data publications. In principle, a record can be used for all kinds of data, including data from simulations or experiments, and it can be linked to other records of related data sets, e.g. to the descriptions of the software and hardware devices used. The metadata of a record includes both basic metadata, such as title or description, and domain-specific metadata, which can be specified generically, in the form of key/value pairs. The latter can be defined using a special editor, as shown in ***Figure 5***. With the help of such metadata, a description of subject- and application-specific records becomes



Key string_key | Value my string | String

Key int_key | Value 123 | Unit cm | Integer

Key float_key | Value 1.23 | Unit | Float

Key bool_key | Value true | Boolean

Key date_key | Value October 9, 2020 12:00:00 AM | Date

Key dict_key | Value | Dictionary

Key nested_key | Value my string | String

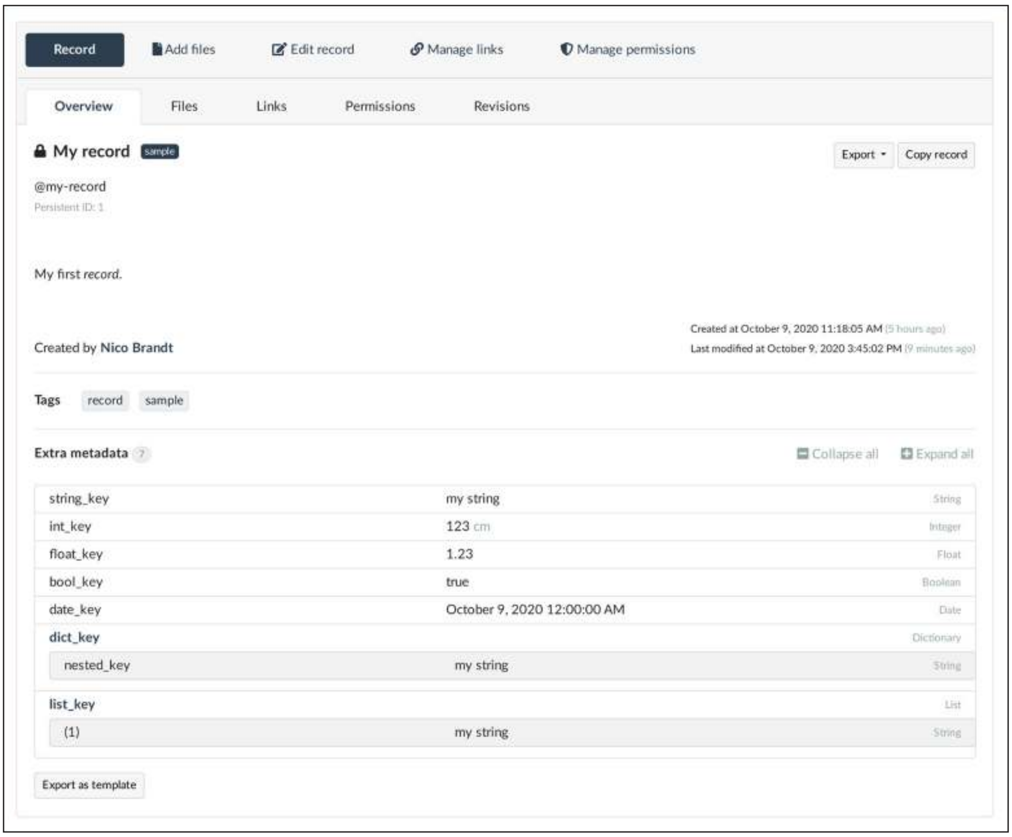
Key list_key | Value | List

Key (1) | Value my string | String

Figure 5 Screenshot of the generic metadata editor, showing the different types of metadata entries currently possible. The last two examples of type *dictionary* and *list* contain nested metadata entries. In the upper right corner, a menu is displayed that allows performing various actions, one of which switches to a tree-based overview of the metadata. The ability to select metadata templates is shown in the lower right corner.

possible. This is particularly relevant in an interdisciplinary research field such as materials science, where using a fixed schema would be impracticable, due to the heterogeneity of the data formats and the corresponding metadata. The value of each metadata entry can be of different types, such as simple character strings or numeric types like integers and floating point numbers. Numeric values can also be provided with an arbitrary unit. Furthermore, nested types can be used to represent metadata structures of almost any complexity, for example in the form of lists. The input of such structures can be simplified by templates, which are specified in advance and can be combined as desired. While templates currently offer the same possibilities as the actual metadata, it is planned to add further validation functionalities, such as the specification of a selection of valid values for certain metadata keys. Wherever possible, automatically recorded metadata is also available in each record, such as the creator of the record or the creation date. The actual data of the record can be uploaded by the users and are currently stored on a file system, accessible by Kadi4Mat. It is possible to upload any number of files for each record. This can be helpful when dealing with a series of several hundred images of a simulated microstructure, for example, which all share the same metadata.

The created record can be viewed on an overview page that displays all metadata and linked files. Some common file formats include a preview that is directly integrated into the web browser, such as image files, PDFs, archives or textual data. Furthermore, the access rights of the record are displayed on its overview page. Currently, two levels of visibility can be set when creating a record: public and private visibility. While public records can be viewed by every logged-in user, i.e. read rights are granted implicitly to each user, private records can initially only be viewed by their creator. Only the creator of a record can perform further actions, such as editing the metadata or deleting the entire record. ***Figure 6*** shows the overview page of a record, including its metadata and the menu to perform the previously mentioned actions. In order to grant different access rights to other users, even within private records, different roles can be defined for any user in a separate view. Currently, the roles are static, which means that they can be selected from a predefined list and are each linked to the corresponding fine-grained permissions. Because of these permissions, the possibility of custom roles or certain actions being linked to different user attributes becomes possible. In addition to roles for individual users, roles can also be defined for *groups*. These are simple groupings of several users which, similar to records, can be created and managed by the users themselves. The same roles that can be defined for individual users can be assigned to groups as well. Each member of the group is granted the corresponding access rights automatically. Finally, the overview page of a record also shows the resources linked to it. This refers in particular to the so-called *collections*. Collections represent simple, logical groupings of multiple records and can thus contribute to a better organization of resources. In terms of an ontology, collections can be regarded as classes, while records inside a collection represent concrete instances of such a class. Like records and groups, collections can be created and managed by users. Records can also be linked to other records. Each record link represents a separate resource, which in turn can contain certain



My record sample

@my-record
Persistent ID: 1

My first *record*.

Created by Nico Brandt

Created at October 9, 2020 11:18:05 AM (5 hours ago)
Last modified at October 9, 2020 3:45:02 PM (9 minutes ago)

Tags record sample

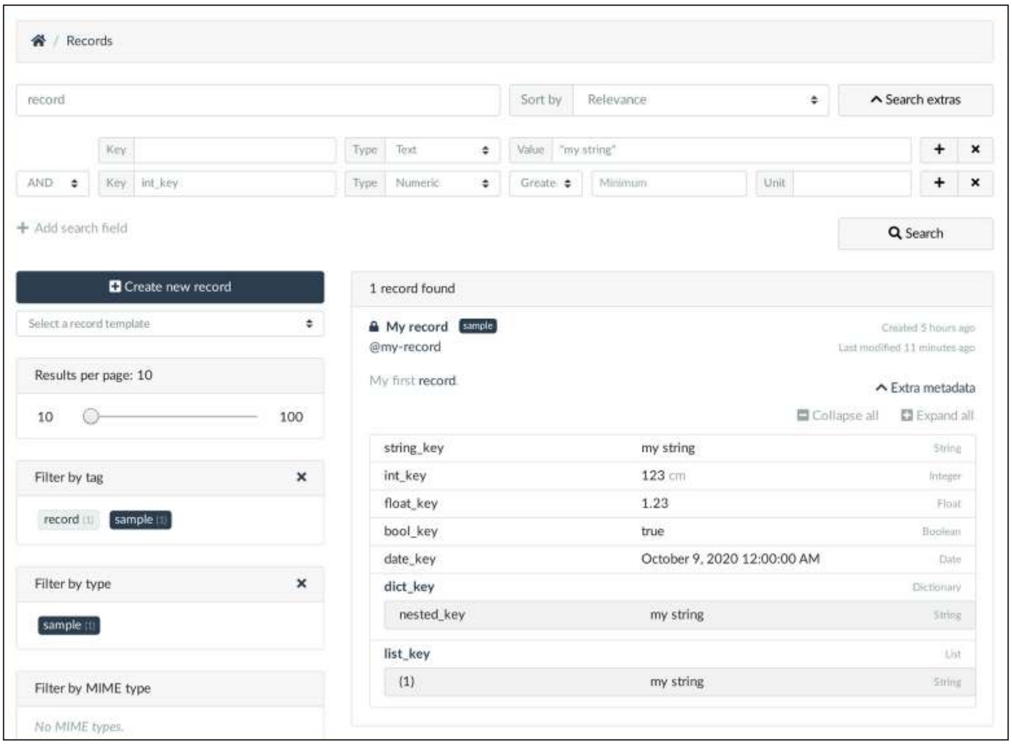
Extra metadata

string_key	my string	String
int_key	123 cm	Integer
float_key	1.23	Float
bool_key	true	Boolean
date_key	October 9, 2020 12:00:00 AM	Date
dict_key		Dictionary
nested_key	my string	String
list_key		List
(1)	my string	String

Figure 6 Screenshot of a record overview page. The basic metadata is shown, followed by the generic metadata entries (shown as *extra metadata*). The menu on the top allows various actions to be performed on the current record. The tabs below the menu are used to switch to other views that display the files and other resources associated with the current record, as well as access permissions and a history of metadata revisions.

metadata. The ability to specify generic metadata and such resource links already enables a basic ontology-like structure. This structure can be further improved in the future, e.g. by using different types of links, with varying semantics, and by allowing collections to be nested.

To be able to find different resources efficiently, especially records, a search function is included in Kadi4Mat. This allows searching in the basic metadata of resources and in the generic metadata of records via keywords or full text search. The values of nested generic metadata entries are flattened before they are indexed in Elasticsearch. This way, a common search mapping can be defined for all kinds of generic metadata. The search results can be sorted and filtered in various ways, for example, by using different user-defined tags or data formats, in the case of records. ***Figure 7*** shows an example search of records, with the corresponding results.



record

Sort by Relevance

Key | Type Text | Value "my string"

AND | Key int_key | Type Numeric | Greate | Minimum | Unit

1 record found

My record sample
@my-record
Created 5 hours ago
Last modified 11 minutes ago

My first record

string_key	my string	String
int_key	123 cm	Integer
float_key	1.23	Float
bool_key	true	Boolean
date_key	October 9, 2020 12:00:00 AM	Date
dict_key		Dictionary
nested_key	my string	String
list_key		List
(1)	my string	String

Results per page: 10

10 100

Filter by tag

record (1) sample (1)

Filter by type

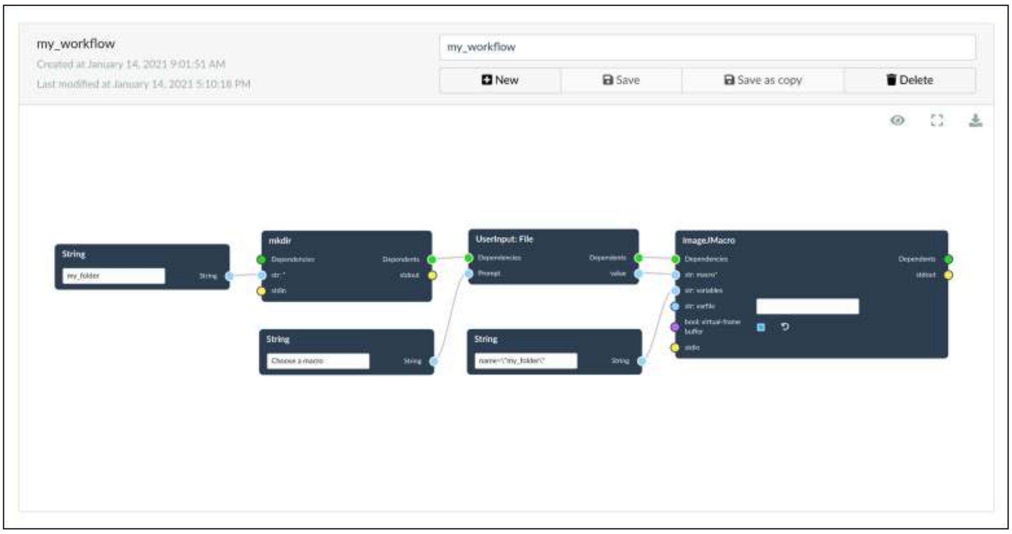
sample (1)

Filter by MIME type

No MIME types.

Figure 7 Screenshot of the search functionality of records with the corresponding search results. In addition to providing a simple query for searching the basic metadata of a record, the generic metadata can also be searched by specifying desired keys, types or values. The searchable types are derived from the actual types of the generic metadata entries, e.g. integers and floating point numbers are grouped together as numeric type. Various other options are offered for filtering and sorting the search results.

While the execution of workflows, via the web interfaces, and the ability to add user-defined tools are still under development, it is possible to define a workflow using a graphical node editor, running in the web browser. ***Figure 8*** shows a simple example workflow created with this editor. A selection of predefined nodes can be combined and parameterized, while the resulting workflow can be downloaded. A custom JSON-based format is currently used to store the representation of a workflow. This format contains all the information for the node editor to correctly display the workflow and to derive a functional workflow representation for execution. The downloaded workflow file can be executed directly on a local workstation by using the command line interface of the process manager. All tools to actually run such a workflow need to be installed beforehand. A selection of tools is provided for various tasks (Zschumme et al. 2020), including connecting to a Kadi4Mat instance by using a suitable wrapper on top of the aforementioned API library. Several common use cases have already been implemented, including the task of extracting metadata from Excel spreadsheets, often used to replace an actual ELN system, and importing it into Kadi4Mat. An overview of such a workflow is shown in ***Figure 9***. Developments are also underway for data science applications, especially in the field of machine learning. The combination of the ELN and the repository fits particularly well with the requirements of such applications, which typically require lots of high-quality input data to function well.



my_workflow
Created at January 14, 2021 9:01:51 AM
Last modified at January 14, 2021 5:10:18 PM

my_workflow

Figure 8 Screenshot of a workflow created with the web-based node editor. Several *String* input nodes are shown, as well as a special node that prompts the user to enter a file (*UserInput: File*). The two tools *mkdir* and *ImageJMacro* are used to create a new directory and to execute an ImageJ (Schindelin et al. 2015) macro file, respectively. The latter uses the input file the user was asked for. Except for the input nodes, all nodes are connected via an explicit dependency.

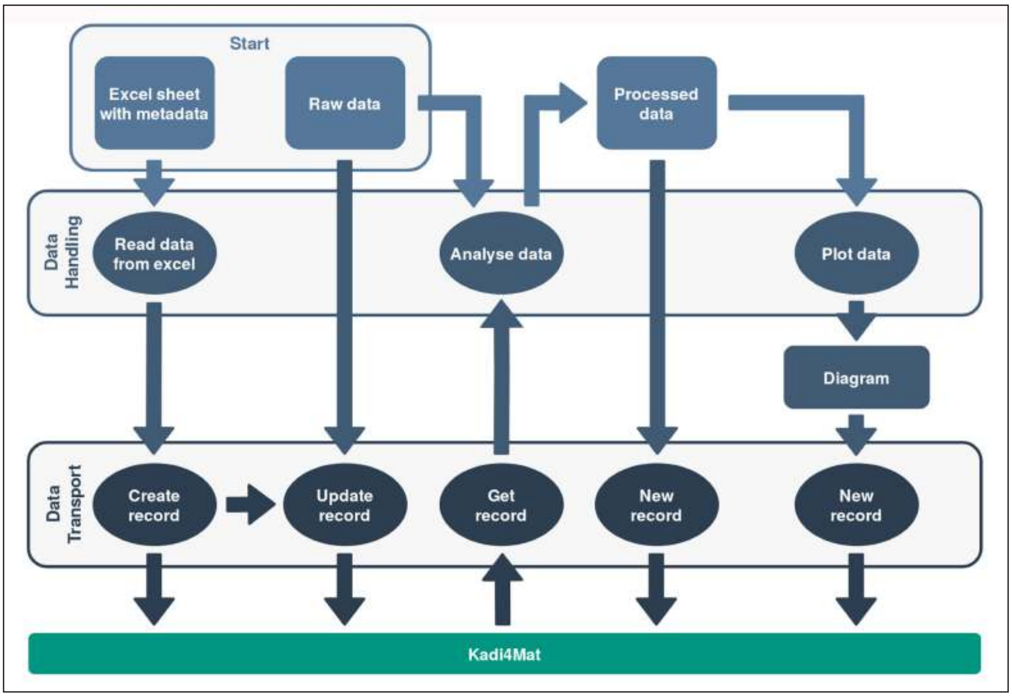


Figure 9 Overview of an exemplary workflow using Kadi4Mat. The starting point is raw data and corresponding metadata stored in an Excel spreadsheet. The tools used in this workflow are divided into tools for data handling and tools for data transport, the latter referring to the Kadi4Mat integration. In a first conversion step, the metadata are transformed into a format readable by the API of Kadi4Mat and linked to the raw data by creating a new record. The raw data is further analysed using the metadata stored in Kadi4Mat. Finally, the result of the analysis is plotted and both data sets are uploaded to Kadi4Mat as records. All records can be linked to each other in a further step, either as part of the workflow or separately.

5 CONCLUSION

The development and current functionality of the research data infrastructure Kadi4Mat is presented. The objective of this infrastructure is to combine the features of an ELN and a repository in such a way that researchers can be supported throughout the whole research process. The ongoing development aims at covering the heterogeneous use cases of materials science disciplines. For this purpose, flexible metadata schemas, workflows and tools are especially important, as is the use of custom installations and instances. The basic functionality of the repository component is largely given by the features already implemented and can be used with a graphical as well as a programmatic interface. This includes, above all, uploading, managing and exchanging data as well as the associated metadata. The latter can be defined with a flexible metadata editor to accommodate the needs of different users and workgroups. A search functionality enables the efficient retrieval of the data. The essential infrastructure for workflows is implemented as a central part of the ELN component. Simple workflows can be defined with an initial version of the web-based node editor and executed locally using provided tools and the process manager's command line interface. Both main components are improved continuously. Various other features that have not yet been mentioned as part of the concept are planned or are already in the conception stage. These include the optional connection of several Kadi4Mat instances, a more direct, low-level access to data and the integration of an app store, for the central administration of tools and plugins.

The development of Kadi4Mat largely follows a bottom-up approach. Instead of developing concepts in advance, to cover as many use cases as possible, a basic technical infrastructure is established first. On this basis, further steps are evaluated in exchange with interested users and by implementing best practice examples. Due to the heterogeneous nature of materials science, most features are kept very generic. The concrete structuring of the data storage, the metadata and the workflows is largely left to the users. As a positive side effect, an extension of the research data infrastructure to other disciplines is possible in the future.

ACKNOWLEDGEMENTS

This work is supported by the Federal Ministry of Education and Research (BMBF) in the projects FestBatt (project number 03XP0174E) and as part of the Excellence Strategy of the German Federal and State Governments, by the German Research Foundation (DFG) in the projects POLiS (project number 390874152) and SuLMaSS (project number 391128822) and by the Ministry of Science, Research and Art Baden-Württemberg in the project MoMaF – Science Data Center, with funds from the state digitization strategy digital@bw (project number 57). The authors are also grateful for the editorial support of Leon Geisen.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Nico Brandt *orcid.org/0000-0002-3860-1376*
Institute for Applied Materials (IAM-CMS), Karlsruhe Institute of Technology (KIT), Straße am Forum 7, 76131 Karlsruhe, Germany

Lars Griem *orcid.org/0000-0002-8093-6356*
Institute for Applied Materials (IAM-CMS), Karlsruhe Institute of Technology (KIT), Straße am Forum 7, 76131 Karlsruhe, Germany

Christoph Herrmann *orcid.org/0000-0001-8208-4356*
Institute for Applied Materials (IAM-CMS), Karlsruhe Institute of Technology (KIT), Straße am Forum 7, 76131 Karlsruhe, Germany

Ephraim Schoof *orcid.org/0000-0001-6821-7263*
Helmholtz Institute Ulm for Electrochemical Energy Storage (HIU), Helmholtzstraße 11, 89081 Ulm, Germany

Giovanna Tosato *orcid.org/0000-0001-5128-4080*
Institute for Applied Materials (IAM-CMS), Karlsruhe Institute of Technology (KIT), Straße am Forum 7, 76131 Karlsruhe, Germany

Yinghan Zhao *orcid.org/0000-0001-9440-138X*
Institute for Applied Materials (IAM-CMS), Karlsruhe Institute of Technology (KIT), Straße am Forum 7, 76131 Karlsruhe, Germany

Philipp Zschumme *orcid.org/0000-0002-4821-5719*
Institute for Applied Materials (IAM-CMS), Karlsruhe Institute of Technology (KIT), Straße am Forum 7, 76131 Karlsruhe, Germany

Michael Selzer *orcid.org/0000-0002-9756-646X*
Institute for Applied Materials (IAM-CMS), Karlsruhe Institute of Technology (KIT), Straße am Forum 7, 76131 Karlsruhe, Germany; Institute for Digital Materials Science (IDM), Karlsruhe University of Applied Sciences, Moltkestraße 30, 76133 Karlsruhe, Germany

REFERENCES

Afgan, E, et al. July 2, 2018. The Galaxy Platform for Accessible, Reproducible and Collaborative Biomedical Analyses: 2018 Update. *Nucleic Acids Research,* 46(W1): W537–W544. DOI: *https://doi.org/10.1093/nar/gky379*

Amorim, RC, et al. Nov. 2017. A Comparison of Research Data Management Platforms: Architecture, Flexible Metadata and Interoperability. *Universal Access in the Information Society,* 16(4): 851–862. DOI: *https://doi.org/10.1007/s10209-016-0475-y*

Bird, CL, Willoughby, C and **Frey, JG.** 2013. Laboratory Notebooks in the Digital Era: The Role of ELNs in Record Keeping for Chemistry and Other Sciences. *Chemical Society Reviews,* 42(20): 8157. DOI: *https://doi.org/10.1039/c3cs60122f*

Brandt, N. 2020. *Kadi4Mat – Karlsruhe Data Infrastructure for Materials Science*. URL: *https://kadi.iam-cms.kit.edu* (visited on Sept. 30, 2020).

Brandt, N, et al. Oct. 16, 2020. IAM-CMS/Kadi: Kadi4Mat. Version 0.2.0. *Zenodo*. DOI: *https://doi.org/10.5281/ZENODO.4088270*

Cantor, S and **Scavo, T.** 2005. Shibboleth Architecture. *Protocols and Profiles,* 10: 16. DOI: *https://doi.org/10.26869/TI.66.1*

CARPi, N, Minges, A and **Piel, M.** Apr. 14, 2017. eLabFTW: An Open Source Laboratory Notebook for Research Labs. *The Journal of Open Source Software,* 2(12): 146. DOI: *https://doi.org/10.21105/joss.00146*

Carusi, A and **Reimer, T.** Jan. 2010. Virtual Research Environment Collaborative Landscape Study. *JISC Report*.

CKAN Association. 2014. *Ckan – The Open Source Data Portal Software*. URL: *https://ckan.org/* (visited on May 19, 2020).

DataCite Metadata Working Group. 2019. *DataCite Metadata Schema Documentation for the Publication and Citation of Research Data v4.3*. Version 4.3. DataCite.

Django Software Foundation. 2005. *Django – The Web Framework for Perfectionists with Deadlines*. URL: *https://www.djangoproject.com/* (visited on May 25, 2020).

Draxl, C and **Scheffler, M.** Sept. 2018. NOMAD: The FAIR Concept for Big Data-Driven Materials Science. *MRS Bulletin* 43(9): 676–682. DOI: *https://doi.org/10.1557/mrs.2018.208*

Elastic NV. 2010. *Elasticsearch – The Official Distributed Search & Analytics Engine*. URL: *https://www.elastic.co/elasticsearch* (visited on June 2, 2020).

Elliott, MH. 2009. Thinking beyond ELN. *Scientific computing,* 26(6): 6–10.

European Materials Modelling Council. 2019. *European Materials and Modelling Ontology*. URL: *https://github.com/emmo-repo* (visited on May 24, 2020).

European Organization For Nuclear Research. 2016. *Invenio – Open Source Framework for Large-Scale Digital Repositories*. URL: *https://invenio-software.org/* (visited on May 19, 2020).

European Organization For Nuclear Research & OpenAIRE. 2013. *Zenodo*. DOI: *https://doi.org/10.25495/7GXK-RD71*

Fielding, RT and **Taylor, RN.** 2000. *Architectural Styles and the Design of Network-Based Software Architectures*. Vol. 7. Irvine: University of California.

Ghiringhelli, LM, et al. Dec. 2017. Towards Efficient Data Exchange and Sharing for Big-Data Driven Materials Science: Metadata and Data Formats. *npj Computational Materials* 3(1). DOI: *https://doi.org/10.1038/s41524-017-0048-5*

Greenberg, J, et al. Nov. 30, 2009. A Metadata Best Practice for a Scientific Data Repository. *Journal of Library Metadata,* 9(3–4): 194–212. DOI: *https://doi.org/10.1080/19386380903405090*

Hawker, CD. Dec. 2007. Laboratory Automation: Total and Subtotal. *Clinics in Laboratory Medicine,* 27(4): 749–770. DOI: *https://doi.org/10.1016/j.cll.2007.07.010*

Hey, AJG, (ed.). 2009. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Redmond, Washington: Microsoft Research. pp. 251.

Hey, T and **Trefethen, A.** Mar. 11, 2003. The Data Deluge: An e-Science Perspective. In: Berman, F, Fox, G and Hey, T (eds.), *Wiley Series in Communications Networking & Distributed Systems*. Chichester, UK: John Wiley & Sons, Ltd. pp. 809–824. DOI: *https://doi.org/10.1002/0470867167.ch36*

Hill, J, et al. May 2016. Materials Science with Large-Scale Data and Informatics: Unlocking New Opportunities. *MRS Bulletin,* 41(5): 399–409. DOI: *https://doi.org/10.1557/mrs.2016.93*

Jain, A, et al. July 2013. Commentary: The Materials Project: A Materials Genome Approach to Accelerating Materials Innovation. *APL Materials,* 1(1): 011002. DOI: *https://doi.org/10.1063/1.4812323*

King, G. Nov. 2007. An Introduction to the Dataverse Network as an Infrastructure for Data Sharing. *Sociological Methods & Research,* 36(2): 173–199. DOI: *https://doi.org/10.1177/0049124107306660*

Kluyver, T, et al. 2016. Jupyter Notebooks – a Publishing Format for Reproducible Computational Workflows. In: Loizides, F and Scmidt, B (eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Netherlands: IOS Press. pp. 87–90. DOI: *https://doi.org/10.3233/978-1-61499-649-1-87*

Naughton, L and **Kernohan, D.** Mar. 7, 2016. Making Sense of Journal Research Data Policies. *Insights the UKSG journal,* 29(1): 84–89. DOI: *https://doi.org/10.1629/uksg.284*

Open Archives Initiative. 2015. *Open Archives Initiative Protocol for Metadata Harvesting*. URL: *http://www.openarchives.org/pmh/* (visited on May 25, 2020).

P. Bryan Heidorn. 2008. Shedding Light on the Dark Data in the Long Tail of Science. *Library Trends,* 57(2): 280–299. DOI: *https://doi.org/10.1353/lib.0.0036*

Pampel, H, et al. Nov. 4, 2013. Making Research Data Repositories Visible: The *Re3data.Org* Registry. In: Suleman, H (ed.), *PLoS ONE,* 8(11): e78080. DOI: *https://doi.org/10.1371/journal.pone.0078080*

PostgreSQL Global Development Group. 1996. *PostgreSQL: The World's Most Advanced Open Source Database*. URL: *https://www.postgresql.org/* (visited on Sept. 30, 2020).

Potthoff, J, et al. Mar. 2019. Procedures for Systematic Capture and Management of Analytical Data in Academia. In: *Analytica Chimica Acta: X,* 1: 100007. DOI: *https://doi.org/10.1016/j.acax.2019.100007*

Rubacha, M, Rattan, AK and **Hosselet, SC.** Feb. 2011. A Review of Electronic Laboratory Notebooks Available in the Market Today. *Journal of Laboratory Automation,* 16(1): 90–98. DOI: *https://doi.org/10.1016/j.jala.2009.01.002*

Sandfeld, S, et al. 2018. *Strategiepapier – Digitale Transformation in der Materialwissenschaft und Werkstofftechnik*.

Schembera, B and **Iglezakis, D.** 2020. EngMeta: Metadata for Computational Engineering. *International Journal of Metadata, Semantics and Ontologies,* 14(1): 26. DOI: *https://doi.org/10.1504/IJMSO.2020.107792*

Schindelin, J, et al. July 2015. The ImageJ Ecosystem: An Open Platform for Biomedical Image Analysis. *Molecular Reproduction and Development,* 82(7–8): 518–529. DOI: *https://doi.org/10.1002/mrd.22489*

Schoof, E and **Brandt, N.** Oct. 16, 2020. IAM-CMS/Kadi-Apy: Kadi4Mat API Library. Version 0.2.1. *Zenodo*. DOI: *https://doi.org/10.5281/ZENODO.4088276*

SciNote LLC 2015. *SciNote – Electronic Lab Notebook & Inventory Management*. URL: *https://www.scinote.net/* (visited on May 21, 2020).

Smith, M, et al. Jan. 2003. DSpace: An Open Source Dynamic Digital Repository. *D-Lib Magazine,* 9(1). DOI: *https://doi.org/10.1045/january2003-smith*

Taylor, KT. 2006. The Status of Electronic Laboratory Notebooks for Chemistry and Biology. *Current Opinion in Drug Discovery and Development,* 9(3): 348.

The FAIRsharing Community, et al. Apr. 2019. FAIRsharing as a Community Approach to Standards, Repositories and Policies. *Nature Biotechnology,* 37(4): 358–367. DOI: *https://doi.org/10.1038/s41587-019-0080-8*

The MathWorks, Inc. 2021. *MATLAB – MathWorks*. URL: *https://www.mathworks.com/products/matlab.html* (visited on Jan. 19, 2021).

The Pallets Projects. 2015. *Flask – The Pallets Projects*. URL: *https://palletsprojects.com/p/flask/* (visited on May 25, 2020).

Tremouilhac, P, et al. Dec. 2017. Chemotion ELN: An Open Source Electronic Lab Notebook for Chemists in Academia. *Journal of Cheminformatics,* 9(1). DOI: *https://doi.org/10.1186/s13321-017-0240-0*

Vue Core Development Team. 2014. *Vue.Js – The Progressive JavaScript Framework*. URL: *https://vuejs.org/* (visited on May 25, 2020).

Wilkinson, MD, et al. Dec. 2016. The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Scientific Data,* 3(1). DOI: *https://doi.org/10.1038/sdata.2016.18*

Wolstencroft, K, et al. July 1, 2013. The Taverna Workflow Suite: Designing and Executing Workflows of Web Services on the Desktop, Web or in the Cloud. *Nucleic Acids Research,* 41(W1): W557–W561. DOI: *https://doi.org/10.1093/nar/gkt328*

Zschumme, P. Jan. 15, 2021a. IAM-CMS/Process-Engine. Version 0.1.0. *Zenodo*. DOI: *https://doi.org/10.5281/ZENODO.4442563*

Zschumme, P. Jan. 15, 2021b. IAM-CMS/Process-Manager. Version 0.1.0. *Zenodo*. DOI: *https://doi.org/10.5281/ZENODO.4442553*

Zschumme, P, et al. Oct. 16, 2020. IAM-CMS/Workflow-Nodes. Version 0.1.0. *Zenodo*. DOI: *https://doi.org/10.5281/ZENODO.4094719*

TO CITE THIS ARTICLE:
Brandt, N, Griem, L, Herrmann, C, Schoof, E, Tosato, G, Zhao, Y, Zschumme, P and Selzer, M. 2021. Kadi4Mat: A Research Data Infrastructure for Materials Science. *Data Science Journal*, 20: 8, pp. 1–14. DOI: *https://doi.org/10.5334/dsj-2021-008*

Submitted: 16 October 2020
Accepted: 27 January 2021
Published: 10 February 2021

Data Science Journal is a peer-reviewed open access journal published by Ubiquity Press.