



Agile Research Data Management with Open Source: CaosDB

Daniel Hornung  ¹

Florian Spreckelsen  ¹

Thomas Weiß  ¹


1. IndiScale GmbH, Göttingen.



Date Received:

2021-08-01

Licenses:

This article is licensed under: 

Keywords:

Data Management, Research Data Management, Agile Data Management, Software Tools, FAIR Data, Good Scientific Practice

Data availability:

Software availability:

Software can be found here:

<https://gitlab.com/caosdb>

and at DOI:10.5281/zenodo.7752417

Abstract.

Research data management (RDM) in academic scientific environments increasingly enters the focus as an important part of good scientific practice and as a topic with big potentials for saving time and money. Nevertheless, there is a shortage of appropriate tools, which fulfill the specific requirements in scientific research. We identified where the requirements in science deviate from other fields and proposed a list of requirements which RDM software should answer to become a viable option.

We analyzed a number of currently available technologies and tool categories for matching these requirements and identified areas where no tools can satisfy researchers' needs. Finally we assessed the open-source RDMS (research data management system) CaosDB for compatibility with the proposed features and found that it fulfills the requirements in the area of *semantic, flexible data handling* in which other tools show weaknesses.

1 Introduction

2 Research units, from small research groups at universities to large research and development
3 departments are increasingly confronted with the challenge to manage large amounts of data, data
4 of high complexity[1], [2] and changing data structures[3], [4]. The necessary tasks for research
5 data management include storage, findability and long-term accessibility for new generations of
6 researchers and for new research questions[4]–[6].

7 In spite of the advantages of implementing data management solutions[7], we found a lack of
8 standard methods or even standard software so far for research data management, especially in
9 the context of quickly evolving methods and research targets. We hypothesize that the reason for
10 this deficit is that scientific research poses unique challenges for data management, since it is
11 characterized by constant innovation, short lived research questions, trial-and-error approaches,
12 and the continuous integration of new insights.

13 We propose *agile research data management* as a promising approach to meet the special
14 requirements of scientific research and to fully leverage the benefits of increased research
15 digitalization, automated data acquisition methods and storage capabilities.

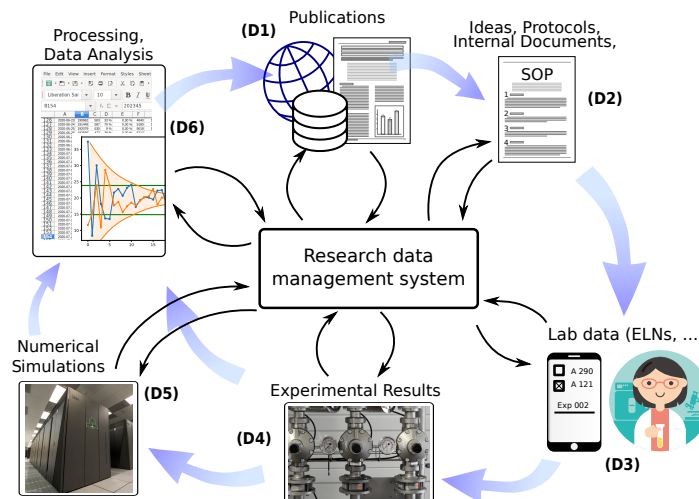


Figure 1: Schematic illustration of the scientific data lifecycle. Data can be obtained from every step, and in most cases the relationship between data entities is just as relevant as the raw data. Blue thick arrows denote the direction in which information flows in normal research. Thin black arrows indicate data flow to and from a research data management system. While this example focuses on experimental and laboratory centered disciplines, comparative lifecycles also exist for theoretical sciences and most fields in the humanities.

16 For this article, we identified the specific challenges for research data management (RDM) and
 17 defined eleven requirements which suitable RDM software should have to (a) fulfill the practical
 18 needs and (b) be accepted by the potential users. We then matched existing tools against these
 19 requirements and found areas where the tools show substantial need for improvement.
 20 Finally we present the CaosDB[8], [9] toolkit as a viable approach to satisfy all the proposed
 21 requirements.

22 2 Challenges for research data management

23 2.1 The scientific data lifecycle: the need for proper tooling

24 Data which accrues in scientific research is more than just experimental readings, field notes
 25 or interview recordings. In order to fully represent the research journey and eventually enable
 26 reproducible science, the data from every research step may become relevant. We identify
 27 the challenges to make this data usable in a way that leads to reproducible, and time-efficient,
 28 research.

29 Figure 1 shows a schematic of different research steps during the research lifecycle, during
 30 which important data is generated. For full reproducibility, it is not sufficient however to simply
 31 store any data that one acquires, but also to represent the semantic connections and make these
 32 connections searchable.

33 In more detail, the most relevant sources and targets for data in scientific research are (numbered
 34 from (D1) to (D6)):

35 **Prior publications (D1)** An important part of good scientific practice (GSP) is to credit the

36 influence of prior work, written by the scientists themselves or third parties. Linking one's
37 own work to previous publications — articles or published data from repositories — and
38 making these connections public helps to assess reproducibility and may lead to fruitful
39 data-reuse in unforeseen contexts.[10] An RDMS should be able to trace back each data
40 item to previous scientific publications on which it is based.

41 **Ideas and SOPs (D2)** The data here consists mostly of text documents which describe thoughts,
42 hypotheses and planned standard operating procedures (SOPs). These documents fill the
43 gap between previous work and the next round of data acquisitions, they often also work
44 as a blueprint for the data acquisition phase.[11] A scientist may consult their RDMS
45 to answer questions like “Which SOP was used when experiment X was carried out to
46 generate data file Y?”.

47 **Lab data (D3)** Environmental data, device settings, used SOPs and ingredients and other inci-
48 dental data typically accrues during the course of experiments and was traditionally stored
49 in paper laboratory notebooks. Currently, a lot of laboratories switch to electronic lab
50 notebooks (ELNs) for the same purpose. While this data is often seen as second-class
51 “metadata”, we hold that since often conclusions can be drawn from it, it deserves the
52 same handling as final instrument readings.[12]–[14]

53 During work in the lab, software must be as unintrusive as possible, with efficient user
54 interfaces.

55 **Experimental results (D4)** These are what is often considered the *main* data. For meaningful
56 analysis, data from experimental results mostly must be enriched with additional data from
57 experimental or device settings or from processed samples, to filter for special conditions,
58 to compare settings or to verify that values are compatible with standard literature.[15]

59 **Numerical simulations (D5)** Similarly to experimental results, data obtained from numerical
60 procedures can not be interpreted without knowledge about used software and parameters,
61 possibly hardware conditions and input from laboratories or third-party data sources.[16]
62 Since bit-for-bit reproducibility is possible in theory, all relevant settings should be stored
63 unchanged.

64 **Data analysis (D6)** When analyzing data from previous steps, storing not only the used pro-
65 grams, scripts, and their parameters, but also the semantic connections enables later
66 researchers to reconstruct which method was used, which assumptions were made and
67 under which conditions the input data was gathered.[17], [18]

68 **Next publication (D1)** Formally the end of the lifecycle, but of course also the beginning of
69 many new ones, a publication contains a number of statements which are supported by
70 data from previous steps. A comprehensive RDMS could quickly answer a question like
71 “In figure X, which methods were used to analyze the data, which devices and software
72 were used to acquire the raw data, and which assumptions were made when planning the
73 experimental setting?”

74 This list focuses on experimental and laboratory centered disciplines like engineering or natural
75 sciences, but of course in the humanities and theoretical sciences, there are equivalent steps
76 which are equally important to preserve and link to each other.

77 2.2 Specifics of scientific research data management

78 There are some needs for data management which are specific to or more pronounced in scientific
79 research, which we will label by (S1) through (S5):

80 **Interoperability** Scientists tend to work with their own custom-written software[19]–[21],
81 which often requires files with data to be directly accessible to the OS via a file system
82 **(S1)**, remote or locally. Also programmatic access (query, retrieve, update) to data via
83 network APIs **(S2)** is a necessity for many scientific data uses.

84 **Agility** Traditional DMS require users to define a data model and stick to it[22]. All data to be
85 entered has to conform to the data model as it was defined. Research however is defined
86 by having undefined outcomes, the research questions, experimental setup or analysis
87 methods change more often than not over the course of one investigation.[23] We therefore
88 identify **(S3)** as the special need for flexibility regarding the data model.

89 **Learning curve** Scientific research is founded upon the contribution of many participants, with
90 different qualifications, varying research foci and high fluctuations. As a consequence, a
91 steep learning curve for using an RDMS would be detrimental to its adoption **(S4)**.

92 **Early usefulness** Systems which only store data, but do not provide short-term advantages, have
93 high acceptance barriers. Especially in academic research, junior scientists with short-term
94 contracts have little incentive to invest time and money in systems which only may pay
95 out on longer timescales.[22] Therefore, RDMS should offer some tangible advantages on
96 the short run **(S5)**.

97 3 Requirements for a scientific RDMS

98 Based upon the challenges from the previous section, we propose a set of requirements for an
99 RDMS to be a useful tool for scientific research.

100 3.1 General requirements

101 **(R1) Semantic linkage** In order to retain the semantic context in which data is embedded, it
102 must be possible in the RDMS to link data sets with each other in a meaningful way, i.e.,
103 the links must bear some meaning. The default linking possibilities and properties of the
104 data types in the RDMS form the *data model*.

105 **(R2) Flexible data model** Researchers require an RDMS for structured storage of data, where
106 the data model can be changed on the fly, without the need to migrate or discard existing
107 data **((S3))**. When the data model is changed, for example due to new machines, protocols
108 or evolving research questions, the existing data must remain valid and usable. A change
109 in ontological semantics *now* must be compatible with previous semantics *then*.

110 **(R3) Searchability** The RDMS should have easily accessible search options not only for prop-
111 erty values of stored entities, but also for links to other entities and properties (and link)
112 thereof. This deep search allows the traversal of the structured knowledge graph and
113 delivers actual utility value.

114 **(R4) Sustainability** In order to assure long-term access to stored data, software solutions must
115 have some safeguard against becoming unmaintained. This could be achieved by being
116 either open-source software or “too big to fail”. In the case of open-source software,
117 either the community or other companies could step in, if the original maintainers stopped
118 their support. On the other hand, if a software system is very widely adopted and thus
119 indispensable, it is unlikely to be abandoned or left unsupported.

120 **(R5) Open APIs** For interaction with third-party programs, the RDM must have an API with
121 low entrance barriers ((S2)). In research contexts, these third-party programs are often
122 custom-written by scientists without explicit computer science background, so extensive
123 documentation of the API is very desirable.

124 3.2 Automation

125 Automation of repetitive data integration reduces error rates and frees users to concentrate on
126 more challenging tasks. It is therefore desirable for an RDMS to have:

127 **(R6) Synchronization** The RDMS should make it easy for its administrators to integrate existing
128 data sources (for example databases or file systems with structured folder hierarchies)
129 into the RDMS: The RDMS should be synchronized automatically with data from these
130 sources, which makes these data available in a unified manner via the RDMS interface.
131 Note that the RDMS can not solve the conceptual problem of a single source of truth when
132 synchronizing data from different sources, but it can at least highlight potential conflicts
133 and where they first occurred to administrators.

134 **(R7) ELN integration** Research work in the lab is increasingly documented with electronic lab
135 notebooks (ELNs)[24], [25], which allow to conveniently enter device and experimental
136 settings in a semi-structured way. This data is usually critical in the analysis of acquired
137 raw data from instruments, e.g., for searching specific data sets or filtering by parameters.
138 There should be a possibility that the RDMS integrates the ELN data and presents it like
139 data from other sources.

140 **(R8) Workflow representation** While following one SOP, the laboratory workflow is often
141 highly standardized, which makes it suitable for representation within the RDMS. The
142 RDMS should support workflows with different states, which can only be switched in an
143 admin-defined pattern. This simplifies the work for users, because they may e.g., only see
144 the interfaces which are relevant for the current sample processing step.

145 3.3 Specific requirements for scientific work

146 As introduced in section [Specifics of scientific research data management](#), some requirements
147 arise from scientific research specifically.

148 **(R9) Versioning** Mistakes during data acquisition happen, and it must be possible to correct
149 existing data sets. At the same time, this editing must be made transparent and the history
150 of each data set must be kept for future inspection.

151 **(R10) File system integration** For interaction with third-party programs, raw data files must be

152 available on standard file systems ((S1)). Ideally the scientists' workflows should remain
153 unchanged by the RDMS.

154 **(R11) Gentle learning curve, early pay-off** To accommodate for the short employment lifecycles in science, RDMS should offer straightforward and simple to learn usage possibilities
155 which give some early sense of achievement ((S4), (S5))[26]. One example could be
156 simplified search options which help users understand that an RDMS will make their work
157 easier when handling with data.
158

159 3.4 Relation to FAIR data management

160 FAIR data management is seen as a general requirement by the scientific community at large.
161 We hold that a research data management system fulfilling (R1) – (R11) can enable research
162 groups to implement a FAIR data management.

163 Specifically, *Findability* can be achieved because each data set and collections of data can be
164 assigned persistent identifiers, data and metadata can be intimately connected and data can be
165 found through the search functionality of the RDMS.

166 Scientific RDMS can enable *Accessibility* through open and standardized APIs and separation of
167 raw data and metadata. RDMS allow for *Interoperability* when users can incorporate existing
168 ontologies for data model, descriptions and references between data sets. *Reusability* is fostered
169 by rich data models including licenses, provenance information and which follow the respective
170 communities' standards.

171 4 Current state of the tools landscape

172 We give a short overview over existing solutions, tools and approaches and over their possibilities.
173 We also classify the extent to which they cover the required features.

174 4.1 Technologies and approaches

175 Currently, DMS tools exist for a range of fields and use numerous technological and methodological approaches.
176

177 **ELNs** Electronic laboratory notebooks (for example eLabFTW[27], Chemotion[28], RSpace[29],
178 eLabJournal[30] and other[24]) replace paper-based physical solutions to document the
179 scientific workflow in laboratories, but also partly planning and analysis of obtained
180 data. They focus on the user experience while entering data and on collaboration between
181 multiple users and allow to enter data in a semi-structured way, often by means of user
182 editable templates.

183 **Field-specific solutions** Many scientific field have specialized data management solutions for
184 their fields which cater to the specific needs, such as chemical structure searches, material
185 property tables or domain specific data visualization. Often, these solutions excel in their
186 purposes but beyond that offer little or no customization options or interaction possibilities.
187 Examples are Nomad[31], C6H6.org[32], Chemotion[28], among others.

188 **Data, article and software repositories** Most scientific journals and some funding agencies
 189 require scientists to publish the data underlying their publications in a publicly accessible
 190 data repository. There are data repositories with custom software, and an increasing number
 191 of public repository instances using off-the-rack software like Dataverse[33], Invenio[34],
 192 DSpace[35] or CKAN[36]. Data repositories cover **(D1)** in the data lifecycle and offer
 193 some search functionality, in all but very few cases they are intended for immutable data
 194 at the time of publication. Data models range from very simple (only authors and text
 195 description) over completely user defined key-value pairs to domain specific fixed data
 196 models for domain repositories.

197 Similarly, software and articles are stored in specialized repositories, which often have
 198 extensive metadata capabilities for the entities stored within them.

199 **Data storage systems** Data storage is a necessary prerequisite for scientific research and thus
 200 there are many well established systems: mirrored network file systems (e.g., NFS, CIFS)
 201 with regular backups to tape archives on the one hand and object stores (e.g., S3) on the
 202 other hand, which store binary blobs outside classical file system structures.

203 **SQL databases** Plain SQL databases use tables where rows represent records and columns
 204 represent the data sets' attributes or properties. Each table with a fixed set of columns of
 205 mostly fixed types represent one type or class of data, defining the properties available for
 206 that type.

207 Because SQL databases are readily available and can be integrated into most programming
 208 languages, they are often used as the technical base for both self-written ad-hoc data
 209 management solutions and existing commercial data management systems alike[37], [38].

210 **Key-value stores** A contrasting approach to SQL databases (therefore categorized as NoSQL
 211 databases, popular examples are CouchDB or MongoDB), key-value stores manage data as
 212 a collection of key-value pairs. They trade the structure of the SQL paradigm for flexibility,
 213 allowing users to store whatever they deem appropriate.

214 **RDF, SPARQL** A common concept from academic knowledge representation research, RDF
 215 (resource description framework)[39] is a framework and representation standard for
 216 subject-predicate-object triples. It has found adoption in the standardization community
 217 and some applications. SPARQL is a query language for accessing RDF data and used by
 218 knowledge services such as Wikidata.[40], [41]

219 4.2 Do existing tools meet the requirements?

220 We discuss to what degree these technologies and tools are able to fulfill the requirements **(R1)**
 221 – **(R11)** listed above. Here we differentiate between technologies, which may be used when
 222 implementing applications on the one hand, and tools on the other hand which are candidates for
 223 data management solutions.

224 4.2.1 Technologies

225 **RDF, SPARQL** RDF was designed and is well suited to represent semantic relationships be-
 226 tween entities and local RDF collections can be extensively searched with SPARQL by

Technology	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
RDF + SPARQL	●	●	●	●	●	○	∅	∅	○	○	○
SQL	●	○	●	●	●	○	∅	∅	○	○	○
Key-value stores	◐	◐	◐	●	●	○	∅	∅	○	○	○
Data storage	○	○	◐	●	●	∅	∅	∅	◐	●	◐
Tools	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
ELNs	◐	○	◐	●	●	○	∅	●	●	○	●
FSS ^a	◐	○	◐	◐	◐	◐	◐	●	◐	◐	●
Repositories	○	○	◐	●	◐	◐	○	○	●	○	●

a. field-specific solutions

R1	Semantic linkage	R7	ELN integration
R2	Flexible data model	R8	Workflow representation
R3	Searchability	R9	Versioning
R4	Sustainability	R10	File system integration
R5	Open APIs	R11	Gentle learning curve, early pay-off
R6	Synchronization		

Table 1: Data technologies, tools and if they meet the requirements.

Symbols used: ●: yes, ○: no, ◐: partly, ◑: may be possible to implement, ∅: not applicable.

227 trained experts. There is a number of standardized RDF serializations which can be generated and read by a many programming languages. Data models can be implemented
 228 using *RDF Schema*, which is based upon RDF. Entities can reference entities located on
 229 other instances, which brings greater flexibility, but raises issues about data mutability
 230 and searchability.
 231

232 **SQL databases** Relational databases thrive on relations between tables and thus allow some
 233 degree of semantic linking, albeit with very limited flexibility. Searching is possible, but
 234 requires a certain degree of expertise, which can be mitigated by external helper tools.
 235 There are standardized implementations, open source and proprietary alike, which can be
 236 expected to continue for the foreseeable future.

237 **Key-value stores** NoSQL databases allow users a comprehensive degree of freedom when
 238 storing data, but at the same time often provide no overarching structure to enforce certain
 239 data model properties. Semantic linkage thus often is limited to convention instead of
 240 internalized structures. Searchability is comparable to traditional SQL databases, and there
 241 is a large number of implementations.

242 **Data storage systems** As a basic technology to store raw files or objects, data storage systems
 243 do not have the ability to link data or provide a data model. Searching data for associated
 244 metadata or file content is possible for some storage systems. Higher-level functionality is
 245 not available within the data storage systems themselves.

246 These base technologies have in common that they mostly do not provide functionality such
 247 as high-level network APIs, graphical user interfaces, integration with other components or
 248 versioning. Also they target technical audiences and thus feature steep learning curves for data
 249 manipulation and searching alike.

250 4.2.2 Tools

251 **ELNs** ELNs target at a non-technical audience and thus generally aim to have low entrance
252 barriers, with tutorials and graphical help functions. Most generic ELNs allow basic
253 linking between stored records and searches thereof, and users are guided in their work
254 of entering data by means of templates. These templates often do not have a semantic
255 meaning however, but serve only as a means of suggesting data fields. Data is organized
256 around lab sessions, the main datatype are notes from the laboratory. ELNs only started to
257 become the de-facto standard in laboratories over the last decade, so the market is far from
258 settled. There are open-source and proprietary software solutions, by large players and by
259 solo enterprises. Nearly all ELNs developed over the last five years now offer APIs for
260 third-party access, and many allow users to organize their workflows, such as different
261 processing steps for a sample.

262 Synchronization with other data sources or integration with file systems is not a core
263 element of ELNs and as such rarely seen. Similarly, synchronization with other data
264 sources exists only on a case-to-case base. Versioning of stored entities is possible to some
265 extent for most ELNs.

266 **Field-specific solutions** Semantic linking may be possible to a certain amount as permitted by
267 the data model, which typically is limited to the use cases foreseen by the developers.
268 Similarly, searching the data often is limited to key-value filters on the specialized data
269 types. Some solutions (e.g., NOMAD) implement their own ELNs, but integration with
270 third-party ELNs and synchronization with other data sources does not exist generally: it
271 could be implemented via APIs, in those cases where they exist. Support for workflows is
272 generally quite good, and the learning curves are adapted to the audience. Versioning of
273 data and integration of existing file systems may be present in some systems. Long-term
274 availability of software support may be an issue when these solutions are only developed
275 by a small set of people or even individuals, often in time-limited funding situations. In
276 these cases, open-source software can be an insurance for the future, especially if there is
277 sufficient development documentation.

278 **Repositories** Data repositories only cover a small subset of data management use cases and as
279 such generally do not implement many of the requirements. They may allow semantic
280 linkage between entities, but do not have encompassing data models at all. Searching is
281 limited to key-value filters and full-text, sometimes referenced datasets can also be used as
282 filters, but there may be APIs which allow external tools to improve on this shortcoming.
283 Repositories generally have institutional funding so that long-term availability can be
284 seen as guaranteed. Synchronization with other data sources, local file system or ELN
285 integration or workflow representation does not make sense, since repositories are meant
286 for manual data archive upload at the end of the scientific life cycle. Upload of data
287 to archives is very straightforward in most cases, and editing of uploaded data does not
288 invalidate the original version, but only marks it as out of date.

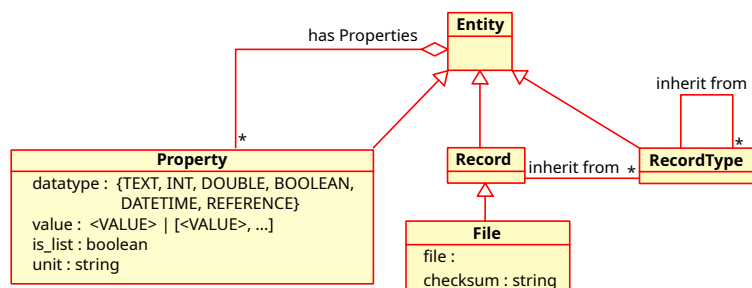


Figure 2: The metadata model of CaosDB.

289 4.2.3 Summary of existing tools

290 The requirements coverage of the examined technology and tool classes are shown in table 4.2.1.
 291 We see that while existing tools cover a wide range of the required features, there are significant
 292 shortcomings in two areas: flexible data models, semantic linkage and searchability on the one
 293 hand, and integration with ELNs, other devices and file systems on the other hand.

294 We stated earlier that these topical fields are especially relevant in scientific research. As an effect
 295 DMS have been widely successful in many areas such as finance, administration, and high-tech
 296 industries[42], [43], but remain scarce in both academic and private sector research[43], [44].
 297 In summary, we find the need for a tool which fills the requirements for semantic, flexible data
 298 management and has sufficient synchronization and ELN integration capabilities.

299 5 CaosDB

300 We hold that CaosDB[9], an agile data management framework, fulfills the proposed requirements
 301 from section Requirements for a scientific RDMS. CaosDB was initially developed by one of
 302 our colleagues, Timm Fitschen, during his time at the Max Planck Institute for Dynamics and
 303 Self-Organization, and others.[45] In 2018, CaosDB was released under the AGPLv3 license on
 304 gitlab.com.[8] Since 2020, CaosDB has found increased adoption in multiple research facilities.

305 In this section, we first describe CaosDB in detail, then we assess to which extent CaosDB
 306 fulfills the proposed requirements and finally we give an overview over limitations and possible
 307 enhancements in the future.

308 5.1 Detailed description

309 CaosDB was developed out of the need for a data management solution that can cope with
 310 large data amount from automated sources and from existing filesystems alike and that allows
 311 researchers to quickly adapt the way how data sets are connected or described. These needs
 312 reflect on the design choices which were taken over the course of development.

313 5.1.1 Data Model

314 CaosDB's *meta* data model is shown schematically in Figure 2. The base type for everything
 315 is ENTITY, with the inheriting types PROPERTY (attributes of ENTITIES, may be list values and
 316 references to other ENTITIES), RECORDTYPE (templates for actual data sets) and RECORD. Actual

317 data is typically stored in RECORDS, which *inherit* from one or more RECORDTYPES and thus
318 have all the PROPERTIES defined therein. The RECORDTYPES may form a complex inheritance
319 hierarchy themselves. FILE entities are similar to Records, but additionally are connected to files
320 which may reside on conventional file systems or potentially in abstracted cloud storage systems.
321 This approach to use files at their current locations instead of duplicating file content not only
322 increases CaosDB’s scalability, but also lower the entrance barrier, since scientists can access
323 the managed file in their traditional ways.

324 Details of this metadata model in CaosDB are elaborated on in [9], but it should be clear now
325 already that CaosDB provides the *Semantic linkage* feature.

326 In CaosDB, the *data model* of the stored data refers to the RECORDTYPES and their PROPERTIES,
327 which together describe the pattern to which newly created data sets should conform. The data
328 model in CaosDB can be modified at any time, but the changes only take effect for data to
329 be inserted *after* this modification. Existing data is not affected and remains unchanged. This
330 property fulfills the proposed *Flexible data model* feature.

331 PROPERTIES of RECORDTYPES are allocated a graded *importance*, which denotes if this PROPERTY
332 is either *obligatory*, *recommended* or merely *suggested* for RECORDS which inherit from this
333 RECORDTYPE, when a user creates a new RECORDS. This system of importances and the fact
334 that *legacy* data is not necessarily consistent with a *modified* data model was a deliberate design
335 decision. The rationale was that when the data model changes, the meaning at the time of data
336 creation should have priority over consistency with later data models.

337 This possibility to completely change the data model, while not giving up on a general structure,
338 places CaosDB between traditional SQL based relational databases and NoSQL approaches (c.f.
339 Figure 3). While we described above why rigid SQL databases are not suited for use in dynamic
340 research environments, giving no structure (the NoSQL paradigm) tends to lead to incoherent
341 data which is hard to search. A common implementation of NoSQL approaches in the context of
342 data management are *data lakes*, where raw data can be stored and annotated with metadata. The
343 missing structure in Data Lakes however has lead to the tongue-in-cheek colloquialism “Data
344 Swamp”. A third approach, using graph databases to represent semantic information, has not
345 found its way into general adoption to our knowledge, presumably because the query languages
346 tend to become very unwieldy, compare the appendix [Appendix: Query language comparison](#)
347 for an example.

348 5.1.2 Architecture and Libraries

349 CaosDB uses a client/server based architecture, as depicted in Figure 4a. CaosDB has is a REST
350 API for simple access by traditional clients and a web interface for browsers, as well as a gRPC
351 API which allows for more complex operations, such as atomic content manipulations. The
352 existing client libraries¹ and the open APIs provide the proposed *Interoperability* requirement.

353 One particularly useful client library component is the *CaosDB Crawler* framework. This
354 extensible framework simplifies the work to synchronize external data sources with CaosDB

1. A list of the available libraries with the respective source code repositories are given in the Appendix section [List of CaosDB libraries](#).

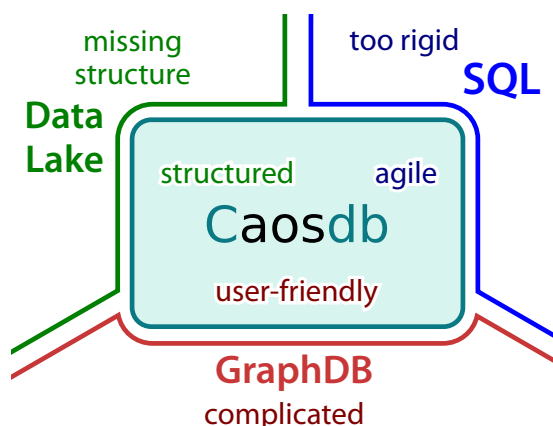


Figure 3: CaosDB compared to other database approaches.

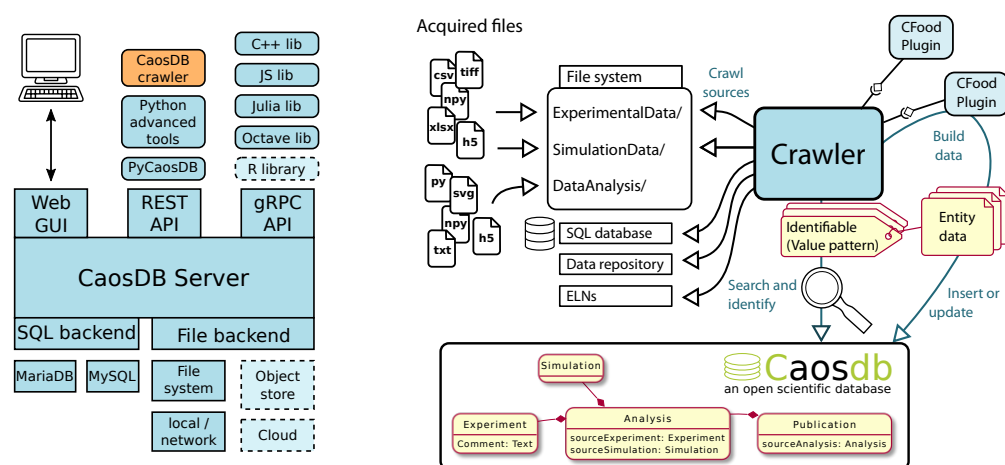


Figure 4: (a) CaosDB's server-client architecture with client libraries and backend components. Dotted elements are under development. (b) The crawler framework facilitates fast development of custom data integration from a diversity of sources.

355 through a plugin system. The crawler workflow can be characterized as follows:

- 356 1. The crawler checks its data sources for new or changed data stores, such as file systems or
 357 the content of other databases. This may happen periodically or be triggered manually by
 358 users.
- 359 2. Each new data source is fed to a so-called *CFood plugin* for consumption. There is a
 360 choice of existing plugins, or administrators can write their own. The CFood plugin's job
 361 is to build CaosDB entities from the consumed data and to specify *Identifiables*, which
 362 work as search patterns. Administrators can mostly define simple CFood plugins by
 363 YAML configuration files[46] which is a more user-friendly approach than for example
 364 the mappings defined by the W3C's R2RML standard.[47]
- 365 3. The crawler checks for each *Identifiable* if a corresponding entity exists already in CaosDB.
 366 If there is no corresponding entity, the entity as returned by the CFood plugin is inserted
 367 into CaosDB. If there is already an existing entity, the Crawler will attempt to merge the

368 existing with the new entity and notify the data curators in case of merge conflicts.

369 This tool set provides the *Synchronization* requirement, and if ELNs are used as external data
370 source, the *ELN integration*. Practical use of CaosDB crawler framework has previously been
371 demonstrated in [48] and ELN integration was implemented as a working proof-of-concept
372 in [49].

373 5.1.3 Miscellaneous features

374 **Deep search** CaosDB offers a simple semantic query language, which borrows some semantics
375 from SQL, but has a focus on usability for non-technical users. The CaosDB query
376 language makes deep search easy with expressions like the following:

```
377 FIND Analysis WITH quality_factor > 0.5  
378 AND WITH Sample WITH weight < 80g
```

379 This convenient nesting of query expressions circumvents the JOIN operations from
380 traditional SQL languages. A full documentation of CaosDB's query language is available
381 online[50] and in CaosDB's sources.

382 **Search templates** CaosDB's web interface provides customizable search templates which allow
383 more advanced users to create their own query templates, which can then be shared with
384 novice users for *simplified searches*. In query templates, users can insert custom strings
385 into pre-defined locations of a search query, see Figure 5.

386 **Versioning** When entities are modified in CaosDB, time and user of the change are recorded
387 and CaosDB puts the previous version onto a history stack and amends the current version
388 with link to the previous version. Over time, each entity may thus grow to a tree of linked
389 versions, which can be retrieved via the web UI or programmatically through the APIs.
390 This feature of CaosDB enables scientific research data management users to adhere to
391 the principles of good scientific practice.

392 **State management** In CaosDB, users may declare a state machine of states and allowed transi-
393 tions. Users may then affix states to entities, and these states can then only be changed
394 according to the rules of the state machine. In this way, users can implement a *workflow*
395 *representation* which ensure that for example laboratory samples run through a specified
396 list of preparation steps in order.

397 5.1.4 Availability and documentation

398 CaosDB is available on the public Git repository gitlab.com at <https://gitlab.com/caosdb>,
399 a detailed list of CaosDB's sub projects is given in the annex. CaosDB's source code is licensed
400 under the AGPLv3 (Affero GNU Public License, version 3). Community contribution workflows,
401 a code of conduct and general development guidelines are outlined in [https://gitlab.com/c](https://gitlab.com/caosdb/caosdb-meta)
402 [aosdb/caosdb-meta](https://gitlab.com/caosdb/caosdb-meta) and in the sub project specific code repositories. The community chat[51]
403 is currently populated with 33 members, the contributors file lists 19 active contributors[45].

404 For the interested public, there is a live demo server at <https://demo.indiscale.com>, hosted
405 by IndiScale GmbH. This demo server is actually running LinkAhead, a commercially supported

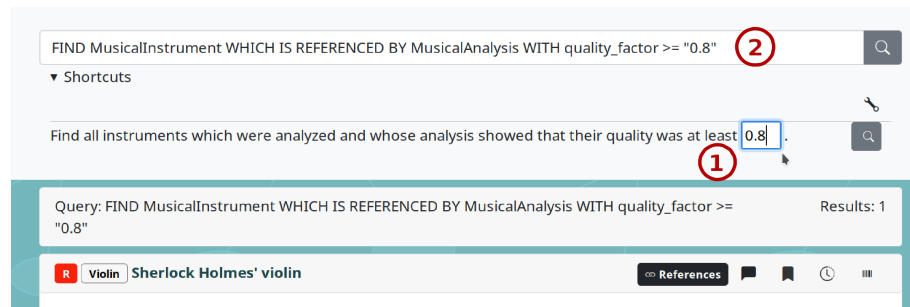


Figure 5: A query template in CaosDB's web UI. The user can enter a custom value into an input field ① and the template is then executed as a plain CaosDB query ②. Screenshot from <https://demo.indiscale.com>.

406 distribution of CaosDB. IndiScale GmbH also provides commercial support, development and
 407 customization services for CaosDB. There are also Debian/Ubuntu packages to run precompiled
 408 LinkAhead/CaosDB for download at <https://indiscale.com/download>.

409 CaosDB's sub projects each have their own documentation in their source directories. The
 410 documentation is also available online at <https://docs.indiscale.com>.

411 5.2 Requirements matching

412 In the following list, we evaluate if and how CaosDB matches the requirements proposed in
 413 section [Requirements for a scientific RDMS](#):

414 **(R1) Semantic linkage** Links between ENTITIES in CaosDB are implemented as reference typed
 415 PROPERTIES, these PROPERTIES can be restricted to Entities with certain parents, adding
 416 an additional ontological level. All PROPERTIES can have a description and higher-order
 417 properties and thus can fulfill the requirements for typical predicates in subject-predicate-
 418 object relationships in predicate logic oriented triple stored such as RDF.

419 **(R2) Flexible data model** In CaosDB, the data model, i.e., the set of RECORDTYPES can be
 420 modified at any time. Existing RECORDS are not affected by these modifications and keep
 421 their properties and inheritance information.

422 **(R3) Searchability** CaosDB's query language allows to deeply search the available data for
 423 simple key-value relations and also for nested relations on the knowledge graph and the
 424 related entities' properties.

425 **(R4) Sustainability** CaosDB is fully open-source and freely available on gitlab.com, with
 426 options for commercial support.

427 **(R5) Open APIs** The REST and GRPC APIs included in CaosDB enable interaction with scien-
 428 tists' custom-written programs. Additionally the existence of client libraries simplifies the
 429 usage by programmers without formal software development training.

430 **(R6) Synchronization** CaosDB's *crawler* framework simplifies the synchronization between
 431 existing data sources and the RDMS and allows to make a diversity of data accessible at a
 432 single resource.

433 **(R7) ELN integration** The CaosDB crawler may use ELNs as a data source, thus integrating
434 the content acquired by ELNs into the RDMS. This makes ELN data searchable and usable
435 equivalently to data from their sources.

436 **(R8) Workflow representation** The state machine in CaosDB can be used to represent stan-
437 dardized workflows. For example laboratory samples or interview partners or publications
438 may have a state whose possible transitions and conditions can be specified.

439 **(R9) Versioning** Entities in CaosDB are versioned and previous content may be displayed and
440 recovered. The content history of entities is stored: which user changed what value at
441 which time.

442 **(R10) File system integration** CaosDB does not make copies of data files but only references
443 the file locations. The file path or resource identifier is returned upon queries, so that users
444 can use the location in their accustomed software.

445 **(R11) Gentle learning curve, early pay-off** Search queries in CaosDB can be made more ac-
446 cessible to users by templates where only specific values need to be filled in. The agile
447 data model allows scientists to start with a structured data management without the need to
448 develop a seemingly overwhelming master plan for their data. Instead they can start small
449 in an area where they expect the most immediate benefits such as improved findability of
450 linked data, and grow the data management at a later time.

451 We find that CaosDB fulfills the requirements **(R1)–(R5)**, **(R9)–(R11)** and that **(R6)–(R8)**
452 (synchronization, ELN integration and workflows) can be readily implemented using on-board
453 means. CaosDB therefore qualifies as a promising candidate for a scientific RDMS.

454 5.3 Critical evaluation and outlook

455 A common misunderstanding about CaosDB is what it provides out of the box and what it can
456 be used for. CaosDB is not a tool to describe data objects following a specific ontology, but
457 ontologies can be implemented with CaosDB in a straightforward manner, and it makes it easy
458 to manage data according to that ontology.

459 Similarly, CaosDB does not enforce data to be FAIR. However researchers can use CaosDB to
460 implement a FAIR data management and to assure that they handle their data in a FAIR manner.
461 Data transferred over the REST and GRPC interfaces use standardized formats such as XML for
462 data serialization, which can be understood by most programming interfaces. Additionally, the
463 internal infrastructure of CaosDB is being reworked to use UUIDs or other unique identifiers as
464 primary keys for all ENTITIES.

465 As outlined in the previous section, CaosDB fulfills most of the requirements and makes others
466 feasible for administrators and users. This also implies that there is room for improvement,
467 for example by providing integrated connectors to ELNs or other data sources or templates for
468 workflow representations.

469 Along similar lines, CaosDB is still lacking tools to seamlessly interchange data and data
470 models with RDF based systems. In order to accelerate the general interoperability between
471 data management tools, CaosDB has become part of the *ELN consortium*[52], an association of

472 interested parties with the aim to develop a common interchange format, based upon the RO-
473 Crate[53], [54] specification. While it is possible now already by external tools, full integration of
474 existing vocabularies represented in RDF serializations will further simplify FAIS data handling
475 with CaosDB.

476 When synchronizing data with CaosDB, special attention has to be given to the relationship
477 between data from external sources (e.g., crawled files, ELNs) and records in the RDMS. Different
478 sources can (usually by some error) have conflicting data, or entries in the RDMS can be changed
479 manually by users after their insertion. In our experience, this problem can not be solved in a
480 general and purely technical way. Instead, best practices have to be implemented as to where
481 possible errors should be corrected and whether some sources have precedence above each other.
482 An RDMS like CaosDB, together with the crawler framework, can help administrators identify
483 inconsistencies in the case of two or more data sources. Through versioning, it is visible who
484 and when maybe changed data manually. How to optimize the help in recognizing potential
485 conflicts, and in the end curate data both in the RDMS and in the external sources, is subject of
486 the authors' ongoing research.

487 Since CaosDB does not receive institutional funding, the direction of its future development
488 depends on the actions of the community. Therefore the immediate advancements will be shaped
489 by the needs of the current users of CaosDB and of the company which currently provides
490 commercial support for it. A current list of feature requests can be generated online.[55] The
491 authors know of about a dozen institutions where CaosDB is currently in use. Together with the
492 growing user base we expect the software to persist for a significant amount of time.

493 CaosDB may fall short in terms of performance against traditional SQL databases for very
494 large amounts of data. To address this issue there is currently development underway to add a
495 virtualization layer which may use existing tabular data sources and present them in a configurable
496 way as native CaosDB ENTITIES.[56]

497 We are aware that the perceived "usability" is subject to personal preferences unless evaluated in
498 a controlled study. We see the potential for a separate survey in the future which systematically
499 evaluates user experiences, workflows and the time and effort spent or gained by users of different
500 software approaches to a previously defined set of data management challenges.

501 **6 Conclusion**

502 We found that scientific research has specific needs to data management: Interoperability, agility,
503 adequate learning curves and early practical use. Altogether we identified a set of eleven
504 requirements which we applied to multiple classes of technologies and tools and to CaosDB,
505 an agile RDMS. Especially in the requirements cluster "Semantic linkage, flexible data model,
506 semantic search", previously existing tools show significant weaknesses, whereas CaosDB offers
507 a promising outlook.

508 We hope that the open source license of CaosDB will inspire more scientists to contribute to
509 CaosDB and improve it in the areas of interoperability with existing standards.

510 7 Appendix: Software

511 7.1 CaosDB

512 The CaosDB suite with the main libraries is published at Zenodo:

513 <https://zenodo.org/record/7752417> (DOI:10.5281/zenodo.7752417)

514 7.2 List of CaosDB libraries

515 The following libraries for programming client applications are publicly available:

516 **Python** <https://gitlab.com/caosdb/caosdb-pylib> The Python client library can be
517 used for third-party applications and is the foundation for several other libraries:

518 **Advanced Python tools** [https://gitlab.com/caosdb/caosdb-advanced-user-t](https://gitlab.com/caosdb/caosdb-advanced-user-tools)
519 [ools](https://gitlab.com/caosdb/caosdb-advanced-user-tools) Additional high-level tools building upon the Python library, including a legacy
520 implementation of the CaosDB crawler. These tools also include converters from
521 JSON Schema to CaosDB's data model.

522 **Crawler** <https://gitlab.com/caosdb/caosdb-crawler> A new implementation of
523 the CaosDB crawler, also using the Python library. Allows to validate data against a
524 JSON Schema.

525 **JavaScript** <https://gitlab.com/caosdb/caosdb-webui> The JavaScript library is part of
526 the web user interface component.

527 **Protobuf API** <https://gitlab.com/caosdb/caosdb-protobuf> The gRPC API is defined via
528 these protobuf files.

529 **C++** <https://gitlab.com/caosdb/caosdb-cpplib> The C++ library uses the gRPC API
530 of CaosDB.

531 **Octave** <https://gitlab.com/caosdb/caosdb-octavelib> The Octave/Matlab library is
532 based upon the C++ library.

533 **Julia** <https://gitlab.com/caosdb/caosdb-julialib> The Julia library also is based upon
534 the C++ library.

535 8 Appendix: Query language comparison

536 As an example for nested queries in different query languages, we consider the search for female
537 UK-based writers in a certain time period, whose given or family name starts with the letter
538 "M". We used the RDF query language SPARQL with Wikidata (<https://www.wikidata.org>)
539 identifiers and CaosDB's query language with fictional but realistic identifier names.

540 The SPARQL query is as follows:

```
541 SELECT DISTINCT ?item ?itemLabel ?givenName ?familyName WHERE {
542     ?item wdt:P31 wd:Q5; # Any instance of a human.
543     ?item wdt:P27 wd:Q145; # citizenship in the United Kingdom
544     ?item wdt:P21 wd:Q6581072; # female
```

```

545         wdt:P106 wd:Q36180; # writer
546         wdt:P569 ?birthday;
547         wdt:P570 ?diedon;
548         wdt:P734 [rdfs:label ?familyName];
549         wdt:P735 [rdfs:label ?givenName].
550     FILTER(?birthday > "1870-01-01"^^xsd:dateTime
551           && ?diedon < "1950-01-01"^^xsd:dateTime)
552     FILTER(regex(?givenName, "M.*") || regex(?familyName, "M.*"))
553     SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
554 }

```

555 In contrast, the CaosDB query looks like this:

```

556 SELECT given_name, family_name FROM Writer
557 WITH gender=f AND citizenship=UK AND birthday > 1870 AND death < 1950
558 AND (given_name LIKE "M*" OR family_name LIKE "M*")

```

559 We understand that SPARQL und CaosDB's query language have non-overlapping sets of
560 features. For example, CaosDB does not know about aliases for names, such as in multilingual
561 environments. On the other hand, SPARQL has no native understanding of SI units and their
562 conversion and it focuses on experts instead of casual users.

563 9 Acknowledgements

564 We acknowledge the previous work on the CaosDB software by its main authors and independent
565 contributors[45], especially Timm Fitschen .

566 10 Conflicts of interest

567 The authors work for IndiScale GmbH, which provides commercial support and other services for
568 CaosDB and the derived free and open-source LinkAhead distribution. DH and FS contributed
569 to the development of CaosDB.

570 11 Roles and contributions

571 **Daniel Hornung:** Conceptualization, Visualization, Writing – original draft

572 **Florian Spreckelsen:** Conceptualization, Writing – review & editing

573 **Thomas Weiß:** Conceptualization, Visualization

574 **References**

- 575 [1] C. R. Bauer, N. Umbach, B. Baum, K. Buckow, T. Franke, R. Grütz, L. Gusky, S. Y.
576 Nussbeck, M. Quade, S. Rey, T. Rottmann, O. Rienhoff, and U. Sax, “Architecture of a
577 Biomedical Informatics Research Data Management Pipeline,” *Exploring Complexity in*
578 *Health: An Interdisciplinary Systems Approach*, pp. 262–266, 2016. doi: [10.3233/978-](https://doi.org/10.3233/978-1-61499-678-1-262)
579 [1-61499-678-1-262](https://doi.org/10.3233/978-1-61499-678-1-262). [Online]. Available: [https://ebooks.iospress.nl/doi/10](https://ebooks.iospress.nl/doi/10.3233/978-1-61499-678-1-262)
580 [.3233/978-1-61499-678-1-262](https://ebooks.iospress.nl/doi/10.3233/978-1-61499-678-1-262).
- 581 [2] C. L. Borgman, “The conundrum of sharing research data,” *Journal of the American*
582 *Society for Information Science and Technology*, vol. 63, no. 6, pp. 1059–1078, 2012,
583 issn: 1532-2890. doi: [10.1002/asi.22634](https://doi.org/10.1002/asi.22634). [Online]. Available: [https://onlinelib](https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.22634)
584 [rary.wiley.com/doi/abs/10.1002/asi.22634](https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.22634).
- 585 [3] A. M. Khattak, K. Latif, and S. Lee, “Change management in evolving web ontologies,”
586 *Knowledge-Based Systems*, vol. 37, pp. 1–18, Jan. 1, 2013, issn: 0950-7051. doi: [10.101](https://doi.org/10.1016/j.knsys.2012.05.005)
587 [6/j.knsys.2012.05.005](https://doi.org/10.1016/j.knsys.2012.05.005). [Online]. Available: [https://www.sciencedirect.com](https://www.sciencedirect.com/science/article/pii/S0950705112001323)
588 [/science/article/pii/S0950705112001323](https://www.sciencedirect.com/science/article/pii/S0950705112001323) (visited on 01/18/2023).
- 589 [4] T. Schneider and M. Šimkus, “Ontologies and Data Management: A Brief Survey,” *KI -*
590 *Künstliche Intelligenz*, vol. 34, Aug. 13, 2020. doi: [10.1007/s13218-020-00686-3](https://doi.org/10.1007/s13218-020-00686-3).
- 591 [5] M. D. Wilkinson, M. Dumontier, IJ. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N.
592 Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes,
593 T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-
594 Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. ’t Hoen, R.
595 Hoof, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B.
596 Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag,
597 T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J.
598 Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, and B. Mons, “The
599 FAIR Guiding Principles for scientific data management and stewardship,” *Scientific Data*,
600 vol. 3, no. 1, p. 160 018, 1 Mar. 15, 2016, issn: 2052-4463. doi: [10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18).
601 [Online]. Available: <https://www.nature.com/articles/sdata201618> (visited on
602 11/22/2021).
- 603 [6] R. Higman, D. Bangert, and S. Jones, “Three camps, one destination: The intersections of
604 research data management, FAIR and Open,” *Insights*, vol. 32, no. 1, p. 18, 1 May 22,
605 2019, issn: 2048-7754. doi: [10.1629/uksg.468](https://doi.org/10.1629/uksg.468). [Online]. Available: [http://insight](http://insights.uksg.org/articles/10.1629/uksg.468/)
606 [s.uksg.org/articles/10.1629/uksg.468/](http://insights.uksg.org/articles/10.1629/uksg.468/) (visited on 01/18/2023).
- 607 [7] G. W. Hruby, J. McKiernan, S. Bakken, and C. Weng, “A centralized research data reposi-
608 tory enhances retrospective outcomes research capacity: A case report,” *Journal of the*
609 *American Medical Informatics Association*, vol. 20, no. 3, pp. 563–567, May 1, 2013,
610 issn: 1067-5027. doi: [10.1136/amiajnl-2012-001302](https://doi.org/10.1136/amiajnl-2012-001302). [Online]. Available: <https://doi.org/10.1136/amiajnl-2012-001302>.
- 611
612 [8] “CaosDB.” Website: <https://caosdb.org>, Source code: [https://gitlab.com/caos](https://gitlab.com/caosdb)
613 [db](https://gitlab.com/caosdb), GitLab. (Feb. 10, 2023).

- 614 [9] T. Fitschen, A. Schlemmer, D. Hornung, H. tom Wörden, U. Parlitz, and S. Luther,
615 “CaosDB— Research Data Management for Complex, Changing, and Automated Research
616 Workflows,” *Data*, vol. 4, no. 2, p. 83, 2 Jun. 2019, issn: 2306-5729. doi: [10.3390/data](https://doi.org/10.3390/data4020083)
617 [4020083](https://doi.org/10.3390/data4020083). [Online]. Available: <https://www.mdpi.com/2306-5729/4/2/83> (visited
618 on 01/18/2023).
- 619 [10] F. Radicchi, S. Fortunato, and A. Vespignani, “Citation Networks,” in *Models of Science*
620 *Dynamics: Encounters Between Complexity Theory and Information Sciences*, ser. Un-
621 derstanding Complex Systems, A. Scharnhorst, K. Börner, and P. van den Besselaar,
622 Eds., Berlin, Heidelberg: Springer, 2012, pp. 233–257, isbn: 978-3-642-23068-4. doi:
623 [10.1007/978-3-642-23068-4_7](https://doi.org/10.1007/978-3-642-23068-4_7). [Online]. Available: <https://doi.org/10.1007>
624 [/978-3-642-23068-4_7](https://doi.org/10.1007/978-3-642-23068-4_7).
- 625 [11] K. Manghani, “Quality assurance: Importance of systems and standard operating proce-
626 dures,” *Perspectives in Clinical Research*, vol. 2, no. 1, pp. 34–37, 2011, issn: 2229-3485.
627 doi: [10.4103/2229-3485.76288](https://doi.org/10.4103/2229-3485.76288). pmid: 21584180. [Online]. Available: [https://www](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3088954/)
628 [.ncbi.nlm.nih.gov/pmc/articles/PMC3088954/](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3088954/) (visited on 07/11/2023).
- 629 [12] M. A. Abujayyab, M. Hamouda, and A. Aly Hassan, “Biological treatment of produced
630 water: A comprehensive review and metadata analysis,” *Journal of Petroleum Science*
631 *and Engineering*, vol. 209, p. 109 914, Feb. 1, 2022, issn: 0920-4105. doi: [10.1016/j.p](https://doi.org/10.1016/j.petrol.2021.109914)
632 [etrol.2021.109914](https://doi.org/10.1016/j.petrol.2021.109914). [Online]. Available: [https://www.sciencedirect.com/scie](https://www.sciencedirect.com/science/article/pii/S0920410521015308)
633 [nce/article/pii/S0920410521015308](https://www.sciencedirect.com/science/article/pii/S0920410521015308) (visited on 07/11/2023).
- 634 [13] A. Nicholson, D. McIsaac, C. MacDonald, P. Gec, B. E. Mason, W. Rein, J. Wrobel,
635 M. de Boer, Y. Milián-García, and R. H. Hanner, “An analysis of metadata reporting
636 in freshwater environmental DNA research calls for the development of best practice
637 guidelines,” *Environmental DNA*, vol. 2, no. 3, pp. 343–349, 2020, issn: 2637-4943. doi:
638 [10.1002/edn3.81](https://doi.org/10.1002/edn3.81). [Online]. Available: [https://onlinelibrary.wiley.com/doi](https://onlinelibrary.wiley.com/doi/abs/10.1002/edn3.81)
639 [/abs/10.1002/edn3.81](https://onlinelibrary.wiley.com/doi/abs/10.1002/edn3.81).
- 640 [14] J. Griss, Y. Perez-Riverol, H. Hermjakob, and J. A. Vizcaíno, “Identifying novel biomarkers
641 through data mining—A realistic scenario?” *PROTEOMICS—Clinical Applications*, vol. 9,
642 no. 3-4, pp. 437–443, 2015, issn: 1862-8354. doi: [10.1002/prca.201400107](https://doi.org/10.1002/prca.201400107). [Online].
643 Available: [https://onlinelibrary.wiley.com/doi/abs/10.1002/prca.201400](https://onlinelibrary.wiley.com/doi/abs/10.1002/prca.201400107)
644 [107](https://onlinelibrary.wiley.com/doi/abs/10.1002/prca.201400107).
- 645 [15] G. Silvello, G. Bordea, N. Ferro, P. Buitelaar, and T. Bogers, “Semantic representation and
646 enrichment of information retrieval experimental data,” *International Journal on Digital*
647 *Libraries*, vol. 18, no. 2, pp. 145–172, Jun. 1, 2017, issn: 1432-1300. doi: [10.1007/s007](https://doi.org/10.1007/s00799-016-0172-8)
648 [99-016-0172-8](https://doi.org/10.1007/s00799-016-0172-8). [Online]. Available: [https://doi.org/10.1007/s00799-016-017](https://doi.org/10.1007/s00799-016-0172-8)
649 [2-8](https://doi.org/10.1007/s00799-016-0172-8).
- 650 [16] B. G. Fitzpatrick, “Issues in Reproducible Simulation Research,” *Bulletin of Mathematical*
651 *Biology*, vol. 81, no. 1, pp. 1–6, Jan. 1, 2019, issn: 1522-9602. doi: [10.1007/s11538-01](https://doi.org/10.1007/s11538-018-0496-1)
652 [8-0496-1](https://doi.org/10.1007/s11538-018-0496-1). [Online]. Available: <https://doi.org/10.1007/s11538-018-0496-1>.

- 653 [17] R. A. Poldrack, K. J. Gorgolewski, and G. Varoquaux, “Computational and Informatic
654 Advances for Reproducible Data Analysis in Neuroimaging,” *Annual Review of Biomedical*
655 *Data Science*, vol. 2, no. 1, pp. 119–138, 2019. doi: [10.1146/annurev-biodatasci-0](https://doi.org/10.1146/annurev-biodatasci-072018-021237)
656 [72018-021237](https://doi.org/10.1146/annurev-biodatasci-072018-021237). [Online]. Available: <https://doi.org/10.1146/annurev-biodatasci-072018-021237>.
- 658 [18] F. Strozzi, R. Janssen, R. Wurmus, M. R. Crusoe, G. Githinji, P. Di Tommaso, D. Belha-
659 chemi, S. Möller, G. Smant, J. de Ligt, and P. Prins, “Scalable Workflows and Reproducible
660 Data Analysis for Genomics,” in *Evolutionary Genomics: Statistical and Computational*
661 *Methods*, ser. Methods in Molecular Biology, M. Anisimova, Ed., New York, NY: Springer,
662 2019, pp. 723–745, isbn: 978-1-4939-9074-0. doi: [10.1007/978-1-4939-9074-0_24](https://doi.org/10.1007/978-1-4939-9074-0_24).
663 [Online]. Available: https://doi.org/10.1007/978-1-4939-9074-0_24.
- 664 [19] J. E. Hannay, C. MacLeod, J. Singer, H. P. Langtangen, D. Pfahl, and G. Wilson, “How
665 do scientists develop and use scientific software?” In *2009 ICSE Workshop on Software*
666 *Engineering for Computational Science and Engineering*, May 2009, pp. 1–8. doi: [10.11](https://doi.org/10.1109/SECSE.2009.5069155)
667 [09/SECSE.2009.5069155](https://doi.org/10.1109/SECSE.2009.5069155).
- 668 [20] J. Segal, “When Software Engineers Met Research Scientists: A Case Study,” *Empirical*
669 *Software Engineering*, vol. 10, no. 4, pp. 517–536, Oct. 1, 2005, issn: 1573-7616. doi:
670 [10.1007/s10664-005-3865-y](https://doi.org/10.1007/s10664-005-3865-y). [Online]. Available: [https://doi.org/10.1007/s1](https://doi.org/10.1007/s10664-005-3865-y)
671 [0664-005-3865-y](https://doi.org/10.1007/s10664-005-3865-y).
- 672 [21] G. Wilson, “Software Carpentry: Getting Scientists to Write Better Code by Making Them
673 More Productive,” *Computing in Science & Engineering*, vol. 8, no. 6, pp. 66–69, Nov.
674 2006, issn: 1558-366X. doi: [10.1109/MCSE.2006.122](https://doi.org/10.1109/MCSE.2006.122).
- 675 [22] N. R. Anderson, E. S. Lee, J. S. Brockenbrough, M. E. Minie, S. Fuller, J. Brinkley, and
676 P. Tarczy-Hornoch, “Issues in Biomedical Research Data Management and Analysis:
677 Needs and Barriers,” *Journal of the American Medical Informatics Association*, vol. 14,
678 no. 4, pp. 478–488, Jul. 1, 2007, issn: 1067-5027. doi: [10.1197/jamia.M2114](https://doi.org/10.1197/jamia.M2114). [Online].
679 Available: <https://doi.org/10.1197/jamia.M2114>.
- 680 [23] M. Bron, J. Van Gorp, and M. de Rijke, “Media studies research in the data-driven age:
681 How research questions evolve,” *Journal of the Association for Information Science and*
682 *Technology*, vol. 67, no. 7, pp. 1535–1554, 2016, issn: 2330-1643. doi: [10.1002/asi.23](https://doi.org/10.1002/asi.23458)
683 [458](https://doi.org/10.1002/asi.23458). [Online]. Available: [https://onlinelibrary.wiley.com/doi/abs/10.1002](https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23458)
684 [/asi.23458](https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23458).
- 685 [24] S. Kanza, C. Willoughby, N. Gibbins, R. Whitby, J. G. Frey, J. Erjavec, K. Zupančič,
686 M. Hren, and K. Kovač, “Electronic lab notebooks: Can they replace paper?” *Journal of*
687 *Cheminformatics*, vol. 9, no. 1, p. 31, May 24, 2017, issn: 1758-2946. doi: [10.1186/s13](https://doi.org/10.1186/s13321-017-0221-3)
688 [321-017-0221-3](https://doi.org/10.1186/s13321-017-0221-3). [Online]. Available: [https://doi.org/10.1186/s13321-017-02](https://doi.org/10.1186/s13321-017-0221-3)
689 [21-3](https://doi.org/10.1186/s13321-017-0221-3).
- 690 [25] S. G. Higgins, A. A. Nogiwa-Valdez, and M. M. Stevens, “Considerations for implementing
691 electronic laboratory notebooks in an academic research environment,” *Nature Protocols*,
692 vol. 17, no. 2, pp. 179–189, 2 Feb. 2022, issn: 1750-2799. doi: [10.1038/s41596-021-0](https://doi.org/10.1038/s41596-021-00645-8)
693 [0645-8](https://doi.org/10.1038/s41596-021-00645-8). [Online]. Available: [https://www.nature.com/articles/s41596-021-00](https://www.nature.com/articles/s41596-021-00645-8)
694 [645-8](https://www.nature.com/articles/s41596-021-00645-8) (visited on 01/20/2023).

- 695 [26] F. Abdullah, R. Ward, and E. Ahmed, “Investigating the influence of the most commonly
696 used external variables of TAM on students’ Perceived Ease of Use (PEOU) and Perceived
697 Usefulness (PU) of e-portfolios,” *Computers in Human Behavior*, vol. 63, no. C, pp. 75–90,
698 Oct. 1, 2016, issn: 0747-5632. doi: [10.1016/j.chb.2016.05.014](https://doi.org/10.1016/j.chb.2016.05.014). [Online]. Available:
699 <https://doi.org/10.1016/j.chb.2016.05.014>.
- 700 [27] N. Carpi, A. Minges, and M. Piel, “eLabFTW: An open source laboratory notebook for
701 research labs,” *Journal of Open Source Software*, vol. 2, no. 12, p. 146, Apr. 14, 2017,
702 Sources: <https://github.com/elabftw/>, issn: 2475-9066. doi: [10.21105/joss.00](https://doi.org/10.21105/joss.0014146)
703 [146](https://doi.org/10.21105/joss.0014146). [Online]. Available: [https://joss.theoj.org/papers/10.21105/joss.0014](https://joss.theoj.org/papers/10.21105/joss.0014146)
704 [146](https://joss.theoj.org/papers/10.21105/joss.0014146).
- 705 [28] P. Tremouilhac, A. Nguyen, Y.-C. Huang, S. Kotov, D. S. Lütjohann, F. Hübsch, N. Jung,
706 and S. Bräse, “Chemotion ELN: An Open Source electronic lab notebook for chemists in
707 academia,” *Journal of Cheminformatics*, vol. 9, no. 1, p. 54, Sep. 25, 2017, issn: 1758-
708 2946. doi: [10.1186/s13321-017-0240-0](https://doi.org/10.1186/s13321-017-0240-0). [Online]. Available: [https://doi.org/10](https://doi.org/10.1186/s13321-017-0240-0)
709 [.1186/s13321-017-0240-0](https://doi.org/10.1186/s13321-017-0240-0).
- 710 [29] *RSpace ELN*, Formerly eCAT from the University of Wisconsin., Edinburgh, Scotland:
711 Research Space. [Online]. Available: <https://www.researchspace.com/> (visited on
712 07/12/2023).
- 713 [30] *eLabJournal*, eLabNext. [Online]. Available: [https://www.elabnext.com/products](https://www.elabnext.com/products/elabjournal/)
714 [/elabjournal/](https://www.elabnext.com/products/elabjournal/).
- 715 [31] C. Draxl and M. Scheffler, “The NOMAD laboratory: From data sharing to artificial
716 intelligence,” *Journal of Physics: Materials*, vol. 2, no. 3, p. 036 001, May 2019, Sources:
717 <https://gitlab.mpcdf.mpg.de/nomad-lab/nomad-FAIR>, issn: 2515-7639. doi:
718 [10.1088/2515-7639/ab13bb](https://doi.org/10.1088/2515-7639/ab13bb). [Online]. Available: [https://dx.doi.org/10.1088](https://dx.doi.org/10.1088/2515-7639/ab13bb)
719 [/2515-7639/ab13bb](https://dx.doi.org/10.1088/2515-7639/ab13bb).
- 720 [32] L. Patiny, M. Zasso, D. Kostro, A. Bernal, A. M. Castillo, A. Bolaños, M. A. Asencio,
721 N. Pellet, M. Todd, N. Schloerer, S. Kuhn, E. Holmes, S. Javor, and J. Wist, “The C6H6
722 NMR repository: An integral solution to control the flow of your data from the magnet to
723 the public,” *Magnetic Resonance in Chemistry*, vol. 56, no. 6, pp. 520–528, 2018, issn:
724 1097-458X. doi: [10.1002/mrc.4669](https://doi.org/10.1002/mrc.4669). [Online]. Available: [https://onlinelibrary](https://onlinelibrary.wiley.com/doi/abs/10.1002/mrc.4669)
725 [.wiley.com/doi/abs/10.1002/mrc.4669](https://onlinelibrary.wiley.com/doi/abs/10.1002/mrc.4669).
- 726 [33] G. King, “An Introduction to the Dataverse Network as an Infrastructure for Data Sharing,”
727 *Sociological Methods & Research*, vol. 36, no. 2, pp. 173–199, Nov. 1, 2007, issn: 0049-
728 1241. doi: [10.1177/0049124107306660](https://doi.org/10.1177/0049124107306660). [Online]. Available: [https://doi.org/10](https://doi.org/10.1177/0049124107306660)
729 [.1177/0049124107306660](https://doi.org/10.1177/0049124107306660).
- 730 [34] J. Caffaro and S. Kaplun. “Invenio: A Modern Digital Library for Grey Literature.”
731 Sources: <https://github.com/inveniosoftware>. (2010), [Online]. Available: [http](http://cds.cern.ch/record/1312678)
732 [s://cds.cern.ch/record/1312678](http://cds.cern.ch/record/1312678) (visited on 07/12/2023), preprint.
- 733 [35] M. Smith, M. Barton, M. Bass, M. Branschofsky, G. McClellan, D. Stuve, R. Tansley, and
734 J. H. Walker, “DSpace: An Open Source Dynamic Digital Repository,” Jan. 2003, issn:
735 1082-9873. doi: [10.1045/january2003-smith](https://doi.org/10.1045/january2003-smith). [Online]. Available: [https://dspace](https://dspace.mit.edu/handle/1721.1/29465)
736 [e.mit.edu/handle/1721.1/29465](https://dspace.mit.edu/handle/1721.1/29465) (visited on 07/12/2023).

- 737 [36] J. Winn, “Open data and the academy: An evaluation of CKAN for research data man-
738 agement,” presented at the IASSIST 2013, Cologne, May 2013. [Online]. Available:
739 <http://eprints.lincoln.ac.uk/id/eprint/9778/> (visited on 07/12/2023).
- 740 [37] A. Ailamaki, V. Kantere, and D. Dash, “Managing scientific data,” *Communications of*
741 *the ACM*, vol. 53, no. 6, pp. 68–78, Jun. 1, 2010, issn: 0001-0782. doi: [10.1145/1743546](https://doi.org/10.1145/1743546.1743568)
742 [6.1743568](https://doi.org/10.1145/1743546.1743568). [Online]. Available: <https://doi.org/10.1145/1743546.1743568>.
- 743 [38] A. Nourani, H. Ayatollahi, and M. S. Dodaran, “Clinical Trial Data Management Software:
744 A Review of the Technical Features,” *Reviews on Recent Clinical Trials*, vol. 14, no. 3,
745 pp. 160–172, Sep. 1, 2019. doi: [10.2174/1574887114666190207151500](https://doi.org/10.2174/1574887114666190207151500).
- 746 [39] O. Hartig, P.-A. Champin, and G. Kellogg. “RDF 1.2 Concepts and Abstract Syntax.”
747 (Jun. 29, 2023), [Online]. Available: [https://www.w3.org/TR/2023/WD-rdf12-con](https://www.w3.org/TR/2023/WD-rdf12-concepts-20230629/)
748 [cepts-20230629/](https://www.w3.org/TR/2023/WD-rdf12-concepts-20230629/) (visited on 07/12/2023).
- 749 [40] S. Harris and A. Seaborne. “SPARQL 1.1 Query Language.” (Mar. 21, 2013), [Online].
750 Available: <https://www.w3.org/TR/2013/REC-sparql11-query-20130321/>
751 (visited on 07/12/2023).
- 752 [41] D. Vrandečić and M. Krötzsch, “Wikidata: A free collaborative knowledgebase,” *Com-*
753 *munications of the ACM*, vol. 57, no. 10, pp. 78–85, Sep. 23, 2014, issn: 0001-0782,
754 1557-7317. doi: [10.1145/2629489](https://doi.org/10.1145/2629489). [Online]. Available: [https://dl.acm.org/doi](https://dl.acm.org/doi/10.1145/2629489)
755 [/10.1145/2629489](https://dl.acm.org/doi/10.1145/2629489).
- 756 [42] T. P. Raptis, A. Passarella, and M. Conti, “Data Management in Industry 4.0: State of the
757 Art and Open Challenges,” *IEEE Access*, vol. 7, pp. 97 052–97 093, Jul. 16, 2019, issn:
758 2169-3536. doi: [10.1109/ACCESS.2019.2929296](https://doi.org/10.1109/ACCESS.2019.2929296).
- 759 [43] S. V. Tuyl and A. Whitmire, “Investigation of Non-Academic Data Management Practices
760 to Inform Academic Research Data Management,” *Research Ideas and Outcomes*, vol. 4,
761 e30829, Oct. 31, 2018, issn: 2367-7163. doi: [10.3897/rio.4.e30829](https://doi.org/10.3897/rio.4.e30829). [Online].
762 Available: <https://riojournal.com/article/30829/> (visited on 01/19/2023).
- 763 [44] M. A. Kennan and L. Markauskaite, “Research Data Management Practices: A Snapshot
764 in Time,” *International Journal of Digital Curation*, vol. 10, no. 2, pp. 69–95, 2 Jun. 30,
765 2015, issn: 1746-8256. doi: [10.2218/ijdc.v10i2.329](https://doi.org/10.2218/ijdc.v10i2.329). [Online]. Available: [http://i](http://ijdc.net/index.php/ijdc/article/view/10.2.69)
766 [jdc.net/index.php/ijdc/article/view/10.2.69](http://ijdc.net/index.php/ijdc/article/view/10.2.69) (visited on 01/19/2023).
- 767 [45] CaosDB. “HUMANS.md,” GitLab. (Nov. 10, 2022), [Online]. Available: [https://gitl](https://gitlab.com/caosdb/caosdb-meta/-/blob/main/HUMANS.md)
768 [ab.com/caosdb/caosdb-meta/-/blob/main/HUMANS.md](https://gitlab.com/caosdb/caosdb-meta/-/blob/main/HUMANS.md) (visited on 07/13/2023).
- 769 [46] “Tutorial: Parameter File — caosdb-caoscrawler 0.6.0 documentation.” (2023), [Online].
770 Available: [https://docs.indiscale.com/caosdb-crawler/tutorials/paramet](https://docs.indiscale.com/caosdb-crawler/tutorials/parameterfile.html#getting-started-with-the-cfood)
771 [erfile.html#getting-started-with-the-cfood](https://docs.indiscale.com/caosdb-crawler/tutorials/parameterfile.html#getting-started-with-the-cfood) (visited on 07/12/2023).
- 772 [47] “R2RML: RDB to RDF Mapping Language.” (Sep. 27, 2012), [Online]. Available: [http](https://www.w3.org/TR/2012/REC-r2rml-20120927/)
773 [s://www.w3.org/TR/2012/REC-r2rml-20120927/](https://www.w3.org/TR/2012/REC-r2rml-20120927/) (visited on 07/12/2023).
- 774 [48] F. Spreckelsen, B. Rüchardt, J. Lebert, S. Luther, U. Parlitz, and A. Schlemmer, “Guidelines
775 for a Standardized Filesystem Layout for Scientific Data,” *Data*, vol. 5, no. 2, p. 43, 2
776 Jun. 2020, issn: 2306-5729. doi: [10.3390/data5020043](https://doi.org/10.3390/data5020043). [Online]. Available: [https:](https://www.mdpi.com/2306-5729/5/2/43)
777 [//www.mdpi.com/2306-5729/5/2/43](https://www.mdpi.com/2306-5729/5/2/43) (visited on 01/18/2023).

- 778 [49] “CaosDB / CaosDB Crawler Cfoods / ELabFTW Cfood · GitLab,” GitLab. (Mar. 31,
779 2023), [Online]. Available: <https://gitlab.com/caosdb/crawler-cfoods/elabftw-cfood> (visited on 07/13/2023).
- 781 [50] “CaosDB Query Language Examples — caosdb-server 0.10.0 documentation.” (2023),
782 [Online]. Available: <https://docs.indiscale.com/caosdb-server/CaosDB-Query-Language.html> (visited on 07/12/2023).
- 784 [51] “#Caosdb on Matrix chat.” (Jul. 13, 2023), [Online]. Available: <https://matrix.to/#/#caosdb:matrix.org> (visited on 07/13/2023).
- 786 [52] “The ELN Consortium,” GitHub. (2023), [Online]. Available: <https://github.com/TheELNConsortium> (visited on 07/12/2023).
- 788 [53] P. Sefton, E. Ó Carragáin, S. Soiland-Reyes, O. Corcho, D. Garijo, R. Palma, F. Coppens,
789 C. Goble, J. M. Fernández, K. Chard, J. M. Gomez-Perez, M. R. Crusoe, I. Eguinoa, N.
790 Juty, K. Holmes, J. A. Clark, S. Capella-Gutierrez, A. J. G. Gray, S. Owen, A. R. Williams,
791 G. Tartari, F. Bacall, T. Thelen, H. Ménager, L. Rodríguez-Navas, P. Walk, b. whitehead, M.
792 Wilkinson, P. Groth, E. Bremer, L. J. Castro, K. Sebby, A. Kanitz, A. Trisovic, G. Kennedy,
793 M. Graves, J. Koehorst, S. Leo, M. Portier, P. Brack, M. Ojsteršek, B. Droesbeke, C. Niu,
794 K. Tanabe, T. Miksa, M. La Rosa, C. Decruw, A. Czerniak, J. Jay, S. Serra, R. Siebes,
795 S. de Witt, S. El Damaty, D. Lowe, X. Li, S. Gundersen, and M. Radifar, “RO-Crate
796 Metadata Specification 1.1.3,” Apr. 26, 2023. doi: [10.5281/zenodo.7867028](https://doi.org/10.5281/zenodo.7867028). [Online].
797 Available: <https://zenodo.org/record/7867028> (visited on 07/12/2023).
- 798 [54] S. Soiland-Reyes, P. Sefton, M. Crosas, L. J. Castro, F. Coppens, J. M. Fernández, D. Garijo,
799 B. Grüning, M. La Rosa, S. Leo, E. Ó Carragáin, M. Portier, A. Trisovic, R.-C. Community,
800 P. Groth, and C. Goble, “Packaging research artefacts with RO-Crate,” *Data Science*, vol. 5,
801 no. 2, pp. 97–138, Jan. 1, 2022, issn: 2451-8484. doi: [10.3233/DS-210053](https://doi.org/10.3233/DS-210053). [Online].
802 Available: <https://content.iospress.com/articles/data-science/ds210053>
803 (visited on 07/12/2023).
- 804 [55] “CaosDB Enhancement Requests,” GitLab - CaosDB. (Jul. 13, 2023), [Online]. Available:
805 [https://gitlab.com/groups/caosdb/-/issues?or\[label_name\]\[\]=Enhance
806 ment%3A%3AAccepted&or\[label_name\]\[\]=Enhancement%3A%3AProposed&state
807 =opened](https://gitlab.com/groups/caosdb/-/issues?or[label_name][]=Enhancement%3A%3AAccepted&or[label_name][]=Enhancement%3A%3AProposed&state=opened) (visited on 07/13/2023).
- 808 [56] “CaosDB Trino branch,” GitLab CaosDB. (May 25, 2023), [Online]. Available: [https://gitlab.indiscale.com/caosdb/src/caosdb-server/-/commits/f-experim
809 ent-trino](https://gitlab.indiscale.com/caosdb/src/caosdb-server/-/commits/f-experiment-trino) (visited on 07/13/2023).