ZeitGeist: A Generic Tool Supporting the Dissemination of Time Series Data Following FAIR Principles

Andreas Schmidt[1,2,a], Mohamad Anis Koubaa[1,b], Jan Schweikert[1,c], Karl-Uwe Stucky[1,d], Wolfgang Süß[1,e] and Veit Hagenmeyer[1,f]

[1]*Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany*

[2]*Department of Computer Science and Business Information Systems, University of Applied Sciences, Karlsruhe, Germany*

fi

Keywords: FAIR Data, RO-Crate, Time Series Data, Export-Query-Configurator.

Abstract: An important point for the widespread dissemination of FAIR-data is the lowest possible entry barrier for preparing and providing data to other scientists according to the FAIR criteria. If scientists have to manually extract, transform and annotate the data according to the FAIR criteria and then export it to make it available to the public, this requires a significant investment of time that does not primarily reward the scientist who prepares and provides the data. The Energy Lab at KIT is running a large cluster of an Influx database management system with energy related time series data being stored in a variety of individual databases over periods of up to 15 years. In order to increase the willingness to make data available to the scientific public, we develop a tool that greatly supports and automates the publication and annotation process of time series data stored in Influx databases.

1 INTRODUCTION

The results of a survey published in Nature (Baker, 2016) revealed that more than 70% of the scientists surveyed had tried to reproduce experiments of other scientists and failed. The article also mentions other studies in the field of cancer research and psychology, where it is estimated that only between 10% and 40% of the experiments described can be reproduced. This fact has serious consequences. On the one hand, it decreases the trust in publications and thus in science as a whole, and on the other hand, it is a big waste of resources if a lot of time has to be invested in verifying results from other papers in order to be able to build on these results afterwards.

One way to increase reproducibility is to make the original data on which the experiments are based available. This is now also being driven forward by a number of research institutions around the world. In Germany, for example, by the *Helmholtz Association of German Research Centers*, the largest scientific organization in Germany with over 44,000 employees and an annual budget of 5.8 billion euros (as of 2020), which recently launched the Helmholtz Metadata Collaboration (HMC) project. The goal of HMC is to develop and establish novel methods and tools to document research data using enriched metadata (HMC, 2023). Another large organisation in Germany is the *German National Research Data Infrastructure (NFDI)* (NFDI, 2023). The NFDI aims to create a permanent digital repository of knowledge as an indispensable prerequisite for new research questions, findings and innovations. NFDI consortia, associations of various institutions within a research field, work together in an interdisciplinary manner to implement the goal. An important consortium is NFDI4Energy (NFDI4Energy, 2023),a national research data infrastructure for the interdisciplinary energy system research.

In 2016, Wilkinson et. al. published a paper (Wilkinson et al., 2016) in which they formally described the *FAIR Guiding principles*, which had been postulated for the first time two years earlier in a workshop in Leiden/Netherlands. FAIR stands for **F**indable, **A**ccessibility, **I**nteroperability,

[a] https://orcid.org/0000-0002-9911-5881
[b] https://orcid.org/0000-0001-8552-2008
[c] https://orcid.org/0000-0003-4774-2717
[d] https://orcid.org/0000-0002-0065-0762
[e] https://orcid.org/0000-0003-2785-7736
[f] https://orcid.org/0000-0002-3572-9083

Schmidt, A., Koubaa, M., Schweikert, J., Stucky, K., Süß, W. and Hagenmeyer, V.
ZeitGeist: A Generic Tool Supporting the Dissemination of Time Series Data Following FAIR Principles.
DOI: 10.5220/0012254300003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 3: KMIS*, pages 303-310
ISBN: 978-989-758-671-2; ISSN: 2184-3228

and **R**eusability of digital artifacts. A very central idea is the description of science artifacts by metadata. For this reason, it is not surprising that the FAIR principles also play an important role in HMC (Buttigieg et al., 2022) and NFDI4Energy (NFDI4Energy, 2023). Section 3 summarizes the most important aspects of FAIR.

An important point for the widespread dissemination of FAIR-data is a low entry barrier for preparing and providing data according to the FAIR criteria. If scientists have to manually extract, transform and annotate the data according to the FAIR criteria and then export it to make it available to the public, this requires a significant investment of time that does not primarily reward the scientist who prepares and provides the data.

In order to increase the willingness to make data available to the scientific public, we develop *ZeitGeist*, a tool that greatly supports and automates the publication and annotation process of time series data stored in an Influx database. The tool is developed in the context of the Energy Lab 2.0 (ELAB, 2023).

The energy transition raises many questions: How can energy be generated in an environmentally friendly way and stored efficiently? What happens when the sun does not shine and the wind does not blow? And what happens if more electricity is suddenly needed? To answer these questions, the Energy Lab 2.0 researches the intelligent interaction of various options to generate, store, and supply energy. As Europe's largest research infrastructure for renewable energy, the Energy Lab 2.0 finds answers to all these questions. There, the intelligent networking of environmentally friendly energy generators and storage methods are investigated. In addition, energy systems of the future are simulated and tested based on real consumer data. A plant network links electrical, thermal, and chemical energy flows as well as new information and communication technologies. The research aims at improving the transport, distribution, storage, and use of electricity and thus creates the basis for the energy transition.

The Energy Lab 2.0 has a large cluster of an Influx time series database, in which a wide variety of energy-related data are stored in a large number of individual databases over periods of up to 15 years. These data in turn form the basis for a wide variety of research projects like SEKO[1] (Sector Coupling), Living Lab Energy Campus[2], Kopernikus 2X[3], and others. In order to make the experiments performed at KIT reproducible for research, it is necessary to make these data available. So far, this has mostly been done within git or DVC (DVC, 2023) repositories.

ZeitGeist is a web application consisting of a backend service and an interactive frontend. The backend provides arbitrary, predefined and annotated time series data of a measurement (an Influx database structure that corresponds to a table in a relational database) via an URL without requiring any further information for access. The specification of the data is undertaken via HTTP-GET parameters. These include the desired time interval and specific conditions on the attributes as well as a configuration file in which the Influx server access information is stored. The actual request is made by a series of REST-API (Inf, 2021) calls to the InfuxDB. In order to be able to extract arbitrarily large amounts of data, a stream-based approach was chosen. The data is returned as an *RO-Crate* dataset (Soiland-Reyes et al., 2022). The column data types are extracted from metadata calls to the Influx database (InfluxMeta, 2022). Further information about the attributes (like quantity, unit), provided as metadata in the *RO-Crate*, can be additionally specified in the configuration file.

The frontend implements the interactive construction of the URL for reading out the time series data. The first step is to select the specific configuration file stored for a particular measurement, which contains the information for accessing a specific database, etc. This information is used, to access the measurement and determine the time interval for which data is available. Meta information of the measurement is read out including the attributes with their data types. In addition, for attributes which act as tags (descriptive attributes), the existing tag values are extracted. These attributes can be used to interactively formulate extraction conditions (e.g. only data of certain buildings, devices, ...). Finally, the time interval of the data to be extracted must be specified. The result of this step is a URL, conforming to the backend API, to export the data.

The rest of the paper is structured as follows: Section 2 provides an overview of the characteristic features of the InfluxDB database management system. Section 3 explains the four FAIR guiding principles (Findable, Accessible, Interoperable, and Reuse), which should apply to scientific data management and stewardship. Section 4 introduces *RO-Crate*, a lightweight approach to packaging research artifacts along with their metadata in machine-readable form in a container. Section 5 then introduces our tool *ZeitGeist*, its architecture and internal functionality as well as the configuration possibilities. Section 6 concludes the paper with a summary and a research outlook.

[1] https://www.esd.kit.edu/85.php

[2] https://www.fz-juelich.de/de/llec

[3] https://www.kopernikus-projekte.de/en/projects/p2x

2 INFLUX DATABASE

The InfluxDB database management system is optimized for storing and querying time series data.

Figure 1 shows the structure of a data set. Each record consists of a mandatory timestamp, zero or more tags, describing the dataset (e.g the location of a sensor), and at least one field for storing a value (e.g. a sensor value). Furthermore, you can see that both the timestamp and the optional tags have an index for quick access, but the fields do not. This means that for read requests, datasets can be quickly selected by their tag values or grouped by them, but not by the actual measured values. The timestamp is represented as `RFC 3339 UTC` timestamp, with nanosecond precision, all tags have the datatype `string`, while the fields can have one of the datatypes `float`, `integer`, `boolean`, or `string`. Each record is stored in a *measurement*, which is an organizational element of the database, similar to a table in a relational database. In contrast to a relational table a measurement is not based on a schema, so that in principle each record can have its own fields and tags. For efficiency reasons, the tag values for a dataset are not stored directly with the dataset, but a hash value is determined for this combination of tag values, which is then stored with the dataset.

The characteristic of time series data, it's chronological order, as well as the lack of transactional support and less query facilities compared to ie.g. SQL, enables the database to perform a series of internal optimizations that result in a much higher write and read rate than would be possible with a multi-purpose database.

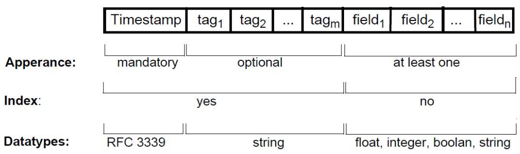

Figure 1: InfluxDB record.

An interesting aspect of time series databases is the `retention policy`. This specifies how long the data should be stored in the database. Records whose age is greater than the *retention policy* are automatically deleted by the server.

Continuous queries are closely linked to the *retention policy*. These are executed cyclically by the database system and are used to "downsample" the data records. This means that older data records are stored in aggregated form before they have reached their lifetime, as defined by the *retention policy*.

An InfluxDB server can run on a single machine as well as in a cluster. It supports sharding as well as replication. An InfluxDB server hosts multiple databases. Each database can have multiple measurements, in which the data records are stored.

InfluxDB comes with a REST API (Inf, 2021). This allows communication with the database from almost any programming language. In addition, there are a number of language bindings, all of which are based on the REST API. InfluxDB currently supports two query languages. One is *InfluxQL*, an SQL like query language and the newer *flux* query language which works stream-based. *InfluxQL* also supports the formulation of queries to the data dictionary, so that information about the structure of the data can be read out. This feature is used with the Influx Exporter developed by us.

3 FAIR PRINCIPLES

An important aspect of FAIR is the possibility of machine processing, since huge amounts of data, its constant growth, and high data complexity make purely manual processing impossible (Go-fair, 2022).

The principles formulated in the following do not recommend any technologies, standards or implementation recommendations, but serve as guidelines for possible implementations.

Findable: Data and metadata must be findable for both humans and computers. For this purpose, the data must be described by rich metadata. Furthermore, data and metadata must be identifiable by a globally unique and persistent identifier (PID). A metadata record should refer to the record of the described data by its PID. In order for data and metadata to be found, they both must be registered and indexed in a searchable resource.

Accessible: Data and metadata must be accessible by its PID through a standardized communication protocol that supports authorization and authentication. Even in the event that data is no longer available, it should be possible to access at least the metadata.

Interoperable: Metadata are described by a formal, common, accessible and widely applicable language for knowledge representation. Furthermore, it must be possible to qualitatively describe relationships between the data sets, which makes it necessary to identify the data sets according to their PID. Example of such languages include *RDF*, *JSON-LD*, or *OWL*.

Reusable: Metadata should be described by a variety of precise and relevant attributes. This should help the client (human, computer) to decide if the data is relevant or not. Also the data and metadata are provided with a unique and accessible data usage license and with provenance information. Further, if there are domain-specific standards or best practices for archiving and sharing data, they should be followed.

4 RO-Crate

The FAIR principles presented above are described independently of any implementation aspects and leave a wide scope for interpretation. This Section will specifically address how research objects (files, workflows, ...) can be described using metadata. In the ideal case, complete experiments can be repeated on the basis of the data and associated metadata, thus ensuring reproducibility and reusability. In our opinion, one of the most promising approaches is *RO-Crate* (Soiland-Reyes et al., 2022). It is a lightweight approach to pack research artifacts together with their metadata in a machine-readable form in a container. This can be done, for example, through a zip archive or a github repository. The semantic of the metadata is described by *schema.org* vocabularies in JSON-LD (JSON-LD, 2018) syntax.

The structure of a *RO-Crate* container consists of the following artifacts:

Data entities are files that can either exist locally in the container as bytestream, reference to external files outside the container, or they are directories. The *data entites* are described in more detail by the *contextual entities*.

Contextual entities exist outside the container and are stored inside the container only by their metadata, like a Person, referenced by their *ORCID*.

The root directory of the container contains the *RO-Crate* metadata file (`ro-crate-metadata.json`), which describes the contents of the *RO-Crate*, the metadata and their relations to each other. The description is done in the linked data JSON-LD format.

In *RO-Crate* it is also possible to define so-called profiles, which simplify the domain-specific use in the sense that certain assumptions can be made about the structure and content of the *RO-Crates*, thus facilitating programmatic use.

5 INFLUX EXPORTER

5.1 Architecture

ZeitGeist consists of a backend service and an interactive frontend. The backend provides arbitrary, predefined and annotated time series data of a *measurement* via an URL without requiring any further information for access. The web-based frontend implements the interactive construction of the URL for reading out the time series data.

Figure 2 gives a high level overview of the involved components. The `ExportConfigurator` (1) allows the selection of a previously defined configuration file. In the configuration, information for accessing the Influx database (server, port, database, user, password) as well as the *measurement* to be read out are specified. Further optional specifications for default values are described in Section 5.2. An example for a configuration is given in the Listings 3 and 4.

After selecting a specific configuration, this information is used to perform a series of queries via the InfluxDB REST API (3). One of the calls determines the time interval within which data is available in the *measurement*. Also, the meta information about the *measurement* is read out. This includes the possible attributes (tags and fields) with their data types. In addition, for tags (descriptive attributes), the existing values are extracted. These attributes can be used to interactively formulate extraction conditions (e.g. only data of certain buildings, devices, ...). Additionally, the time interval of the data to be extracted must be specified. The result of this step is a URL (4), conforming to the backend API, to export the data. The URL contains a number of HTTP GET parameters that specify the desired data as well as the configuration file. An example of a generated URL is shown in Listing 1. Beside the configuration-file `kit.cn.buildings.tapwater.ini`, the begin and end of the time interval are specified `2019-09-23T02:00:00Z`, `2019-10-23T02:00:00Z`, as well as a restriction on the tag *building* (`0101` or `0121`).

```
https://zeitgeist.clients.iai.kit.edu/↩
    InfluxExporter.php?config=kit.cn.↩
    buildings.tapwater.ini&start=2019-09-23↩
    T02:00:00Z&end=2019-10-23T02:00:00Z&↩
    select_building[]=0101&select_building↩
    []=0121
```

Listing 1: Generated URL.

The script behind the URL is `InfluxExporter` (5). It is responsible for delivering the specified data as an *RO-Crate* object. The program expects a number of key-value pairs, which

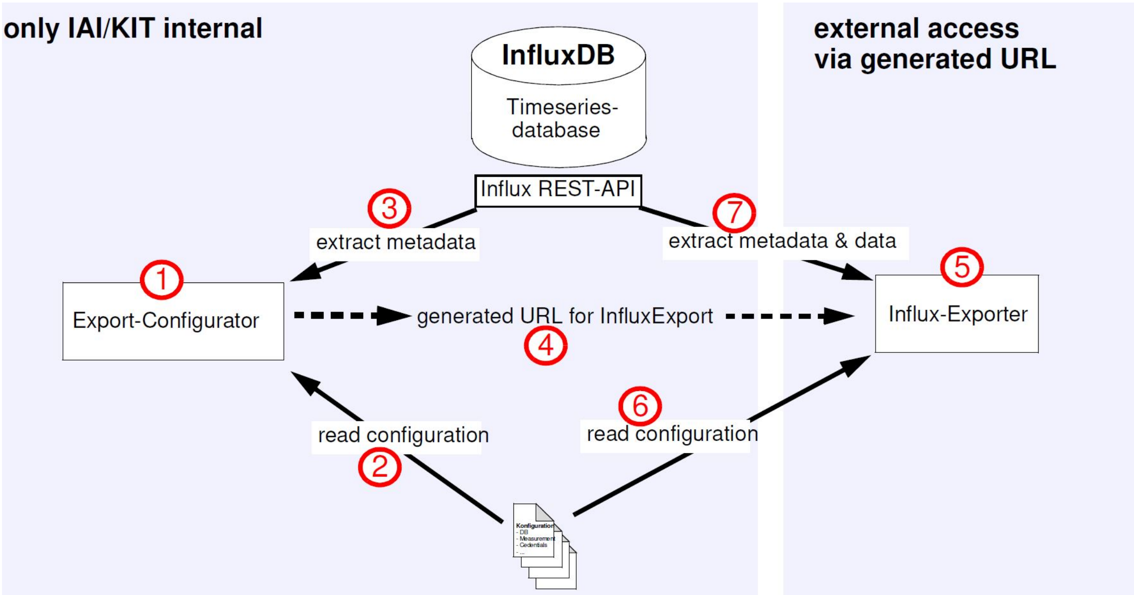

Figure 2: General Architecture.

are delivered as HTTP GET parameter. The script reads the configuration file, specified in the URL for obtaining the information to access the Influx database and then transforms the given parameter to an *InfluxQL* query. The query for the URL in Listing 1 can be seen in Listing 2.

```
select *
  from "kit.cn.buildings.tapwater"
 where time >= '2019-09-23T02:00:00Z'
   and time <= '2019-10-23T02:00:00Z'
   and ("building" = '0101' or
        "building" = '0121')
 order by time
```

Listing 2: Generated InfluxQL query.

5.2 Configuration

A configuration is normally split into two files. The reason for this is that one configuration is specific to one *measurement*, but the connection information for accessing the Influx cluster is the same for many *measurements*. In order to avoid having the complete login information in each configuration file, these are moved to a separate file and referenced from the *measurement*-specific configuration file. This has the particular advantage that if the access information changes (e.g. the database is moved to a new computer), the changes only have to be made in one file. Listing 3 and 4 illustrate this fact. The access information is located in Listing 3 (file: `ini-files/elab-ml4t.ini`) and is included in line 3 of the *measurement*-specific configuration file (Listing 4).

In addition to the settings already described in the previous section, there are a number of further optional settings. These are either default values, which are displayed in the configurator (export time interval), or information, which is transferred to the *RO-Crate* container as meta information.

Unless otherwise specified, the entire time interval during which data is available is displayed in the GUI as the default interval for export. By setting one or more of the properties `default_interval_start`, `default_interval_end` and/or `default_interval_range` the default interval to be exported can be customized. Beside absolute timestamps in UTC format, also relative timestamps using `now()`, a function supported by the InfluxDB, can be used. For example, the property-setting `default_interval_start = "now() - 1h"` returns all datasets within the last hour. If only absolute timestamps are used, the property `cacheable` can be set to `true`. In this case, the returned data record is also stored in a cache, so that in the case of subsequent queries with the same conditions and time interval, the query does not have to be sent to the database again, but can be served directly from the cache, thus relieving the database. However, this only makes sense if it can be ensured that the data has not changed in the meantime (which is typically the case with historical data).

Further properties allow a more precise specification of the provided meta-information of the output columns. While the data type (`string, float, integer, boolean, timestamp`) of the individual tags and fields are determined automatically by special queries regarding the structure of the schema of an Influx database, further information such as ”quantity” or ”unit” of a result column must then be defined by setting the corresponding properties. In Listing 4, for example, starting from line 22 on, it is specified that the quantity kind of field `value` is `ActiveEnergy` and furthermore that the unit is specified in kilowatt-hours (KiloWHR in QUDT notation).

To provide information about the publisher (person, organisation), the two properties `ror` and `orcid` can be set in the configuration file. These entries are displayed as default values in the Export Configurator’s GUI but can be overwritten. The same applies to the property *license*.

```
server = "https://elab-influx-d1.server.elab2↩
    .kit.edu:8086"
username = "iai-ml4t-flx-r-001"
password = "${IAI_ML4T_FLX_R_001}"
```

Listing 3: file `ini-files/elab-ml4t.ini`.

```
1 ; mandatory entries:
2 ;
3 connection=ini-files/elab-ml4t.ini
4 database=fm_efficio_mirror
5 measurement=efficio_raw
6
7 ; optional entries:
8 ;
9 default_interval_end="now()"
10 default_interval_range = "1 hour"
11 description="efficio_raw (DB:↩
    fm_efficio_mirror)"
12 cacheable=false
13
14 [ro-crate]
15 ror="https://ror.org/04t3en479"
16 orcid="https://orcid.org/0000-0002-9911-5881"
17 licence="https://spdx.org/licenses/CC-BY-4.0.↩
    html"
18
19 ; optional datatype information
20 ;
21 [columns]
22 quantity[value] = "http://qudt.org/vocab/↩
    quantitykind/ActiveEnergy"
23 unit[value] = "http://qudt.org/vocab/unit/↩
    KiloW-HR"
```

Listing 4: Configuration file.

5.3 Output Format

The export of a measurement is provided as *RO-Crate* container. Figure 3 shows the content of the zipped *RO-Crate*. The name of the ZIP file is formed from the measurement name and the exported time interval.

Within the ZIP file there are three files. the file `data.csv` contains the exported time series

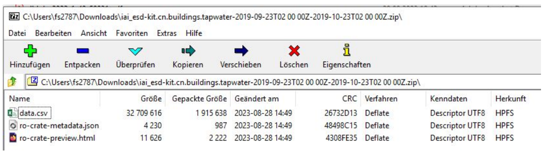

Figure 3: Result RO-Crate object.

dataset. The file `ro-crate-metadata.json` contains the metadata in JSON-LD format. Additionally the file `ro-crate-preview.html` was created and packed. This file is not mandatory and contains a human-readable representation in HTML of the content of the `ro-crate-metadata.json` file.


```
{
    "@id": "ro-crate-metadata.json",
    "about": {
        "@id": "./"
    },
    "conformsTo": {
        "@id": "https://w3id.org/ro/crate/1.1"
    },
    "@type": "CreativeWork"
},
{
    "@id": "data.csv",
    "encodingFormat": "text/csv",
    "fileSize": 32.709616,
    "name": "data",
    "csvw:dialect": {
        "@id": "#csv-dialect"
    },
    "startDate": "2019-09-01T02:00:00Z",
    "endDate": "2019-09-30T23:59:59.999999999Z",
    "csvw:tableSchema": {
        "@id": "kit.cn.buildings.tapwater-651a8f4891afc"
    },
    "@type": [
        "File",
        "csvw:Table"
    ]
},
{
    "@id": "#csv-dialect",
    "csvw:delimiter": ",",
    "csvw:lineTerminator": "\n",
    "@type": "csvw:Dialect"
},
{
    "@id": "kit.cn.buildings.tapwater-651a8f4891afc",
    "@type": "csvw:Schema",
    "csvw:columns": [
        {
            "@id": "measurement-651a8f4891b22"
        },
        {
            "@id": "time-651a8f4891b2a"
        },
        {
            "@id": "building-651a8f48a22b4"
        },
        {
            "@id": "device-651a8f48a22e6"
        },
        {
            "@id": "value-651a8f4b17d8e"
        }
    ]
},
{
    "@id": "measurement-651a8f4891b22",
    "csvw:name": "measurement",
    "csvw:datatype": "http://www.w3.org/2001/XMLSchema#dateTime#string",
    "@type": "csvw:Column"
},
{
    "@id": "time-651a8f4891b2a",
    "csvw:name": "time",
    "csvw:datatype": "http://www.w3.org/2001/XMLSchema#dateTime",
    "@type": "csvw:Column",
    "qudt:unit": "http://qudt.org/vocab/unit/UNITLESS",
    "qudt:quantity": "http://qudt.org/vocab/quantitykind/Time"
},
{
    "@id": "building-651a8f48a22b4",
    "csvw:name": "building",
    "csvw:datatype": "http://www.w3.org/2001/XMLSchema#dateTime#string",
    "@type": "csvw:Column"
},
{
    "@id": "device-651a8f48a22e6",
    "csvw:name": "device",
    "csvw:datatype": "http://www.w3.org/2001/XMLSchema#dateTime#string",
    "@type": "csvw:Column"
},
{
    "@id": "value-651a8f4b17d8e",
    "csvw:name": "value",
    "csvw:datatype": "http://www.w3.org/2001/XMLSchema#double",
    "@type": "csvw:Column"
}
```


Figure 4: Extract from `ro-crate-metadata.json`.

CSVW (CSVW, 2017) syntax specification is used to describe the structure of the CSV file. In addition to general aspects such as separators and the use of quotation marks, *csvw* can be used to specify the data types of the columns. Figure 4 is an excerpt from the `ro-crate-metadata.json file`. It shows the connection between the file `data.csv`, the schema `kit.cn.buildings.tapwater-...` and the associated columns together with their data types, described in *csvw*.

5.4 Implementation Aspects

The current prototype is developed as a proof of concept in PHP and runs on an Apache server. The cache is implemented using a simple filesystem directory. The streaming component used is realized using the *ZipStream-PHP*-library (Männchen, 2023) from Jonatan Männchen. It is planned to undertake a complete reimplementation in Python and to make it available as open source to the community, with the hope to contribute to the wide spreading of the FAIR principles. The future features discussed in Section 6 will also be implemented in this next version.

6 CONCLUSION AND OUTLOOK

The FAIR principles seek to ensure sustainable research data management. By enriching data with metadata, it should be possible for both humans and computers to (1) find relevant data (Findable), (2) access it (Accessible), (3) integrate it with other data (Interoperable), and (4) be able to decide (based on the given metadata) if it could be used in different contexts (Reuse).

In order to be able to publish research data in the future very easily and without significant time effort according to the FAIR principles, we develop *ZeitGeist* that can greatly simplify the publication process of Influx time series data. *ZeitGeist* allows the configuration of the data to be exported and automatically adds meta-information during the subsequent export, according to the FAIR principles and following the *RO-Crate* approach.

A re-implementation of the prototype in Python and the availability as open source is planned. For this version we have planned the following enhancements:

Currently, configuration files are created directly in a directory of the web server, which requires access to the directory, but this is typically reserved for administrators. In the new version, it should be possible to create and administer configuration files using the GUI of the Export Configurator. By also integrating a user/group management concept, the configuration files created in this way can then also be reserved for specific persons/groups.

In the next version we also plan that not always all attributes of the *measurement* will be exported, but that the attributes to be exported can be specified.

Automatic resolution of ORCID and ROR: The RO-Crate specification requires that referenced contextual entities (metadata) should at least be described with a name and type in the *RO-Crate* metadata file. The reason for this is, that clients need not to follow all links when displaying the provided information.

Another possible extension would be to allow the export of different temporal resolutions of the data.

ACKNOWLEDGEMENTS

This publication was supported within the Hub Energy of the Helmholtz Metadata Collaboration (HMC), an incubator platform of the Helmholtz Association within the framework of the Information and Data Science strategic initiative.

REFERENCES

Baker, M. (2016). Is there a reproducibility crisis? *Nature*, pages 452–454.

Buttigieg, P. L., Curdt, C., Ihsan, A. Z., Jejkal, T., Kubin, M., Mannix, O., Mohr, D. P., Pirogov, A., Port, B., and Stucky, K.-U. (2022). An interpretation of the FAIR principles to guide implementations in the HMC digital ecosystem. Projektbericht, Geomar, Kiel, Germany.

CSVW (2017). CSVW Namespace Vocabulary Terms. https://www.w3.org/ns/csvw.

DVC (2023). DVC Documentation. https://dvc.org/doc.

ELAB (2023). Welcome to the Energy Lab 2.0. https://www.elab2.kit.edu/english/index.php.

Go-fair (2022). FAIR Principles. https://www.go-fair.org/fair-principles/.

HMC (2023). Helmholtz Metadata Collaboration (HMC). https://helmholtz-metadaten.de/.

Inf (2021). InfluxDB API reference. https://docs.influxdata.com/influxdb/v1.8/tools/api/.

InfluxMeta (2022). Explore your schema using InfluxQL. https://docs.influxdata.com/influxdb/v1.8/query_language/explore-schema/.

JSON-LD (2018). JSON-LD 1.1: A JSON-based Serialization for Linked Data. https://www.w3.org/2018/jsonld-cg-reports/json-ld/.

Männchen, J. (2023). ZipStream-PHP. https://github.com/maennchen/ZipStream-PHP.

NFDI (2023). German National Research Data Infrastructure (NFDI). https://nfdi.de/.

NFDI4Energy (2023). NFDI4energy Consortia. https://nfdi4energy.uol.de/.

Soiland-Reyes, S., Sefton, P., Crosas, M., Castro, L. J., Coppens, F., Fernández, J. M., Garijo, D., Grüning, B., La Rosa, M., Leo, S., Ó Carragáin, E., Portier, M., Trisovic, A., Community, R.-C., Groth, P., Goble, C., and Peroni, S. (2022). Packaging research artefacts with RO-Crate. *Data Science*, 5(2):97–138.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data*, 3.