FAIR Sensor Ecosystem: Long-Term (Re-)Usability of FAIR Sensor Data through Contextualization

Matthias Bodenbenner*, Jan Pennekamp†, Benjamin Montavon*, Klaus Wehrle†, Robert H. Schmitt*‡
*WZL, *RWTH Aachen University*, Germany · {m.bodenbenner, b.montavon, r.schmitt}@wzl-mq.rwth-aachen.de
†COMSYS, *RWTH Aachen University*, Germany · {pennekamp, wehrle}@comsys.rwth-aachen.de
‡*Fraunhofer IPT*, Germany

***Abstract*—The long-term utility and reusability of measurement data from production processes depend on the appropriate contextualization of the measured values. These requirements further mandate that modifications to the context need to be recorded. To be (re-)used at all, the data must be easily findable in the first place, which requires arbitrary filtering and searching routines. Following the *FAIR guiding principles*, fostering findable, accessible, interoperable and reusable (FAIR) data, in this paper, the *FAIR Sensor Ecosystem* is proposed, which provides a contextualization middleware based on a unified data metamodel. All information and relations which might change over time are versioned and associated with temporal validity intervals to enable full reconstruction of a system's state at any point in time. A technical validation demonstrates the correctness of the *FAIR Sensor Ecosystem*, including its contextualization model and filtering techniques. State-of-the-art FAIRness assessment frameworks rate the proposed *FAIR Sensor Ecosystem* with an average FAIRness of 71%. The obtained rating can be considered remarkable, as deductions mainly result from the lack of fully appropriate FAIRness metrics and the absence of relevant community standards for the domain of the manufacturing industry.**

***Index Terms*—FAIR Data; Cyber-Physical Systems; Data Management; Data Contextualization; Internet of Production**

I. Introduction and Motivation

The measurement data from manufacturing and production processes bears the potential for manifold improvements of these processes and product quality in the long run [1]–[3]. Examples of such reuse of usually large amounts of historical process data, range from retrospective fault analysis, predictive approaches [4] or process optimization in general [5]. However, this potential can only be fully released, if the data is richly annotated with meta and context information. By that, the data can be unambiguously interpreted and reused later [6]. The contextualization of data becomes even more vital when facing use cases and setups, which are subject to frequent changes and modifications. This happens very often in research and science when the setup of experiments is altered to investigate the behavior of systems under different conditions [7]. Similar trends are observable in production systems, which need to be adapted constantly to improve the quality or efficiency of the processes or meet the demands of customers [8]. Specific examples of such changing setups are large-scale metrology applications, like the tracking and position estimation of mobile units. Even in large volumes of several meters, these systems yield high accuracy. Thus, they are often used to track and measure multiple different objects [9]. These measurements can happen simultaneously or successively. Moreover, depending on the measuring system the device's configuration or parametrization is regularly modified, such as repositioning or calibration.

Today, this time-sensitive contextualization of measured data and the seamless description and collection of the context is performed insufficiently in practice [10]. Thus, the reuse of such data is almost impossible due to missing context information and can be considered lost [11], [12]. To facilitate the degree of documentation and hence reusability of such data, this paper adopts the FAIR guiding principles [13] that are well-known for data management. The FAIR principles have already been adopted by other approaches [14], however, related work usually insufficiently covers a subset of the FAIR principles only or is not suited for industrial application.

Contributions. This paper addresses this research gap and proposes the novel *FAIR Sensor Ecosystem* for contextualization and persistent storage of measurement data. The *FAIR Sensor Ecosystem* is a centrally hosted backend consuming measurement data sent via a publish/subscribe protocol. A *FAIRification middleware* unambiguously assigns each value to a measured object and the device's configuration at the time of measurement, utilizing a data structure that persistently links measured data to objects of interest, even if the setup is subject to frequent changes. Due to the persistent data linking, the proposed solution enables high findability, high interoperability and long-term reusability of the stored data.

Organization. Section II introduces a specific scenario to illustrate the research gap, reviews the FAIR guiding principles and derives domain-specific prerequisites for FAIR measurement data in the domain of manufacturing. Section III reviews relevant scientific literature, succeeded by the description of the architecture and data model of the *FAIR Sensor Ecosystem* in Section IV. After that, in Section V, the prototypic implementation and validation are explained. The paper is concluded in Section VI with a summary and outlook.

II. Scenario and Background

To ensure the suitability of the developed *FAIR Sensor Ecosystem*, this section derives domain-specific prerequisites for the envisaged system. For that, a specific, common use-case scenario in industry and science is illustrated and ex-

Author manuscript.

plained. Additionally, the FAIR guiding principles are reviewed.

A. Industrial Metrology Applications

As motivated, contextualization is required to acquire reusable measurement data from scientific experiments as well as from industrial production processes. Whereas the developed *FAIR Sensor Ecosystem* should be widely applicable, a *large-scale metrology (LSM)* application is used as a specific example to emphasize the needs for and benefits of the ecosystem. In general, measuring devices for LSM track and measure *targets*, which are attached to the object of interest. Depending on the device and the desired accuracy these targets are expensive and thus often used in different setups (a detailed description of *LSM* applications have been described by Montavon [9]). However, for applications demanding very high accuracy, such as machine-tool calibration, the measurement uncertainty induced by the specific target needs to be considered. Thus, it must be unambiguously preserved, which target was attached to which object of interest at which point in time. In this paper, this need is further illustrated and investigated using the exemplary setup explained and shown in Figure 1.

B. FAIR Guiding Principles for Data

To facilitate the value of data Wilkinson et al. [13] proposed the *FAIR guiding principles* for data. The objective of the principles is to improve the findability, accessibility, interoperability and reusability (FAIR) of scientific data. To adhere to the four basic principles FAIR data must fulfill the requirements shown in Table I according to Wilkinson et al.

These principles can be utilized to improve the value of data in general and not of scientific data only. The described challenges in the manufacturing industry can be addressed by adopting the FAIR principles for this domain. However, to build the *FAIR Sensor Ecosystem* upon the guiding principles, it is required to derive domain-specific requirements the system must fulfill to provide long-term reusable sensor data.

C. Prerequisites for Reusability of Sensor Data

Whereas some principles are more clear and specific, and thus, comparably easy to realize (in particular **F1**, **A1** and **R1.1**), most are quite vague [16] and therefore, not suitable to directly develop a technical solution. Depending on the perspective and the domain these principles might be differently interpreted and thus realized and implemented differently. Leading to the question of how the long-term reusability of measurement data can be realized based on domain-specific requirements derived from the FAIR guiding principles. Based on the described scenario, the FAIR principles have been adopted for the manufacturing domain by deriving specific prerequisites that have to be fulfilled.

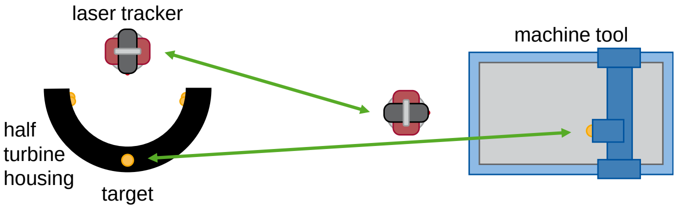

Fig. 1. Close to industry scenario of a LSM application using a laser tracker for measuring the position of targets attached to different objects. The green arrows indicate a temporary repositioning of the tracker and one of the targets. To be able to reuse the data correctly, the change of the tracker's position and association of the target with the two different objects (turbine housing and machine tool) must be preserved in the data. The figure is not drawn to scale.

P1 Upon **F1** and **F3**, in case the configuration, setup or relations between objects change, the identifiers of data must be persisted.

P2 To increase findability and reusability (**F2** and **R1**), objects and measurement values should be describable by metadata of any type.

P3 To preserve interoperability (in particular **I1**), it must be ensured that (meta) data of the same type always has the same format.

P4 To fulfill principle **A2** metadata must not be deletable. Instead, a new revision of the metadata should be created and linked to the older version.

P5 All data and metadata need to be enriched with semantic annotations to fulfill principles **I1**, **R1**, i.e. the terms used for description should be defined in available ontologies or terminologies.

P6 To fulfill **F4**, it must be possible to filter the data by any field of the data model. Thus, the search must also include the metadata fields which can store any type as per **P2** and **P3**.

On top of these FAIR-derived functional prerequisites, performance and scalability have been identified as crucial non-functional requirements for any *FAIR Sensor Ecosystem*, because such a system must be able to ingest and process

TABLE I
A SUMMARY OF THE FAIR DATA PRINCIPLES AS SPECIFIED IN [13]
(RECREATED WITH PERMISSION OF THE AUTHORS FROM [15]).

Findability	**Accessibility**
F1 (meta)data are assigned a globally unique and persistent identifier F2 data are described with rich metadata (defined by R1 below) F3 metadata clearly and explicitly include the identifier of the data it describes F4 (meta)data are registered or indexed in a searchable resource	A1 (meta)data are retrievable by their identifier using a standardized communications protocol A1.1 the protocol is open, free, and universally implementable A1.2 the protocol allows for an authentication and authorization procedure, where necessary A2 metadata are accessible, even when the data are no longer available
Interoperability	**Reusability**
I1 (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation. I2 (meta)data use vocabularies that follow FAIR principles I3 (meta)data include qualified references to other (meta)data	R1 meta(data) are richly described with a plurality of accurate and relevant attributes R1.1 (meta)data are released with a clear and accessible data usage license R1.2 (meta)data are associated with detailed provenance R1.3 (meta)data meet domain-relevant community standards

continuously incoming measurement data and deal with high data rates of more than 1k data points per second.

III. Related Work

Based on these prerequisites, related work is studied which aims to increase the FAIRness of (measurement) data. Some focus on specific aspects of FAIR only. Among others, ontologies and terminologies for the interoperable description of measurement data have been published, such as the *Semantic Sensor Network Ontology (SSN)* [17], the *OGC SensorThings API* [18], or the *Web of Things (WoT) Things Description* [19].

Semantic Sensor Network Ontology. Using SSN the relations between measurements, the measuring device, the observed property and the measurement procedure can be described in detail. Defining a common vocabulary and structure for modeling measurement data, interoperability of measuring devices, sensors and measurements is facilitated. However, SSN is not capable of tracking changes in the configuration or specification of devices, as required by **P1** and **P4**. So, to persist and preserve relations between measurements, the measuring devices and the configuration of the device at the time of measuring, additional concepts must be introduced.

OGC SensorThings API. The *OGC SensorThings API* provides a similar set of terms like SSN for the definition of data structures for measurement data. Moreover, it offers a REST API for accessing and manipulating data from devices for which a data structure is defined according to the SensorThings API's data model. The API also supports the exchange of sensor data in various formats, including JSON and XML. Due to the API, it is possible to alter the data model and meta-information of a measuring device after deployment and in use. But, the OGC SensorThings API has the same drawback as SSN. Moreover, the OGC SensorThings API does not provide any functionality for data curation. So, **P3** can only be ensured using additional tools or frameworks.

Web of Things - Things Description. A more general approach is the *WoT Things Description*, which does not only target measurement data and devices. It provides a data model for describing the characteristics, capabilities and behavior of IoT devices. Based on the Resource Description Framework and using JSON and JSON-LD as data formats it paves the way for interoperable and compatible devices. On top of the description, there exist drafts for WoT *Profiles* [20] and WoT *Discovery* [21], which deal with the provision of a RESTful HTTP API and the discovery and exploration of connected devices, respectively, based on a *Thing description*, i.e., a corresponding data model. Compared to the SSN and the OGC SensorThings API, with the WoT Things Description the setting of a device can be versioned, but relations between different things and other elements must be qualified explicitly by users. Moreover, all three approaches mainly address the interoperability of data but do not explicitly consider the other three aspects of FAIR.

FactDAG. Gleim et al. [15] proposed FactDAG, an approach fully dedicated to the FAIR guiding principles for describing production data and its provenance. To persistently identify data, keep track of changes and preserve provenance, each data point is considered as an immutable *Fact*, which is identified by a triple called *FactID* and stored in a directed acyclic graph. Further, the authors did a reference implementation called *FactStack* [22] build upon standardized protocols and data formats in accordance with the FAIR principles. But, the approach is missing a dedicated way of filtering and searching the data by custom keys and attributes, which is not compliant to **P6** and challenges the retrieval of data in practice.

MontiThings. Another approach for the development of IoT applications for handling measurement data has been published by Kirchoff et al. [23]. The *MontiThings* modeling infrastructure enables the definition of IoT devices, the parameters and data they provide and how the data is processed. For that, the authors apply model-driven software development and generative software implementation. However, the approach is more focused on the data flow between distributed data processing entities. Suitability for long-term storage and data management following FAIR principles is not considered.

No approach can provide a solution, which completely fulfills the FAIR guiding principles and the defined prerequisites, Thus, we stress the research gap of having a FAIR contextualization model for measurement data, which

(i) facilitates long-term storage with persistent data identification,
(ii) considers flexible filtering and retrieving of stored data,
(iii) and can deal with frequently changing measurement setups and configurations.

IV. FAIR Sensor Ecosystem

To overcome the identified research gap, the authors propose the *FAIR Sensor Ecosystem* that consumes measurement data and stores it in a database having a schema reflecting the prerequisites mentioned in Section II-C. Thereby, the data structure is fully aligned with the FAIR guiding principles. For that, the individual components are tailored to meet the FAIR guiding principles and the defined prerequisites.

TABLE II
Overview of the individual software components of the proposed *FAIR Sensor Ecosystem* and the requirements met by each of the components.

Component	FAIR Principles	Prerequisites
Schema validation	**I1**	**P3**
User *API*	**A1**, **I1**, **R1**	**P4**, **P5**
Persistent database	**F4**	
Protocol and data format Ingestion Service	**A1.1**, **I2**	
Persistent identifiers	**F1**, **F3**	**P1**
Structured metadata model	**F1**, **R1**	**P2**
Strong data typing	**I1**	**P3**
Data versioning	**A2**	**P4**
Metadata linking	**F3**, **I1**, **I3**, **R1**	**P5**
Mandatory license	**R1.1**	
Filtering technique	**F4**	**P6**

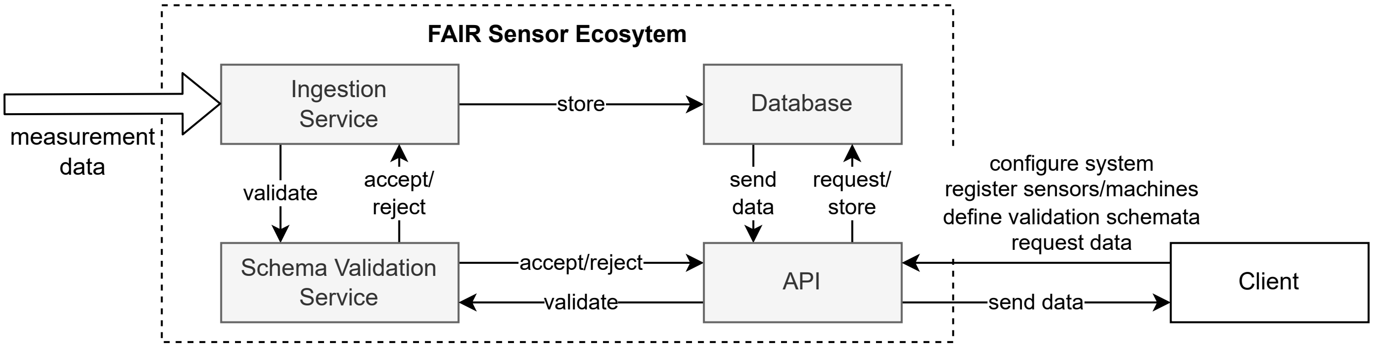

Fig. 2. Architecture of the *FAIR Sensor Ecosystem* for processing and storing measurement data according to an object model defined by users via an *API*.

The next subsection describes the system architecture of the *FAIR Sensor Ecosystem* (c.f. Figure 2), its software components and the flow of measurement data from ingestion to storage (or rejection). An overview of which software component addresses which principle and prerequisite, is given in Table II. After that, the contextualization model is presented, which is used to structure and validate ingested measurement data. The section closes with an explanation of how stored data can be filtered and retrieved via a *RESTful API*.

A. Architecture of the Ecosystem

According to **P3**, ingested data must be validated against defined schemata. Storage in the *persistent database* includes updates of links and references between related resources. For the creation and maintenance of the relationships between modeled objects, data and metadata schemata, the ecosystem must provide endpoints for that via an *API*.

As shown in Fig. 2 the architecture consists of four software components. The *API* provides endpoints for creating virtual objects, which represent real-life objects such as machines, sensors or assets, relations between them and meta-information about the objects according to the contextualization model presented in Section IV-B. The relations and meta information can further be updated via *API* according to **P4** and **P5**. Furthermore, schemata for data and metadata can be defined as required by **P3**, which are used to validate incoming data and metadata via the *Schema Validation Service*. To fulfill **P4**, the *API* does not offer an endpoint to delete data.

Measurement data sent from external devices and services via a publish/subscribe protocol is handled by the *Ingestion Service*. According to the stored object model, specified topics are subscribed by the *Ingestion Service*, which thereby receives corresponding data. The data is checked for the validity of its type against defined types via the *Schema Validation Service* and rejected if the type is invalid. Otherwise, the data is stored in the *Database* (following **F4**) and linked to the currently valid meta information of the objects, which produced the data.

Non-functional, but essential features of the system regarding the FAIR principle is the usage of standardized, open communication protocols and data formats, as required by principle **A1.1**. In addition, the meta-information must use publicly defined terms following principle **I2**.

B. Contextualization Model

Contextualization requires preserving the relations between the measurement data and objects of the physical world. Thus, the corresponding data points need to be linked. Besides temporal aspects, these relations are generally hard to qualify. The kind of relation and thus requirements and aspects of the relations differ between different scenarios and use cases. Thus, in this work, the relations between objects have been simplified to a generic composition scheme, where each relation is associated with an interval of temporal validity, as can be seen in Figure 3. Each object is modeled as a component, which can have arbitrarily many subcomponents, but only one parent component. The resulting hierarchical tree-like structure might not fully express the usually complex connections of objects in production but gives a basic structure that allows to flexibly relate measurement data to superordinate objects. The assumption that the relationships between the objects change frequently over time is reflected in the model by storing a validity interval for all relations defined by timestamps of the beginning and the end for each relation.

Although the configuration or parametrization of a single device, i.e., component, might change frequently, it must be ensured, that the node in the model graph representing the component does not change to preserve correct linking. Thus, in the model, the metadata on parametrization is decoupled from the component itself. For that, a dedicated entity is introduced, which is linked to the component it describes and to its preceding and following versions of the metadata, by using the PROV ontology [24] fulfilling **P4**. The structure of the metadata itself is not further restricted except for being representable as JSON object. Thus, users can store theoretically arbitrary (semi) structured data conforming to **P2**

Ingested measurement data is always (indirectly) linked to at least two components, which fulfills principle **I3**. First, measurement data is linked to the component information of the measuring device. Thus, the metadata includes a reference to the data they describe, so that principle **F3** is satisfied. Second, measurement data is linked to the component representing the object of interest of the measurement, i.e., the object of which a property is measured. Both together fulfill **P5**. Taking into account that the object of interest is not necessarily the

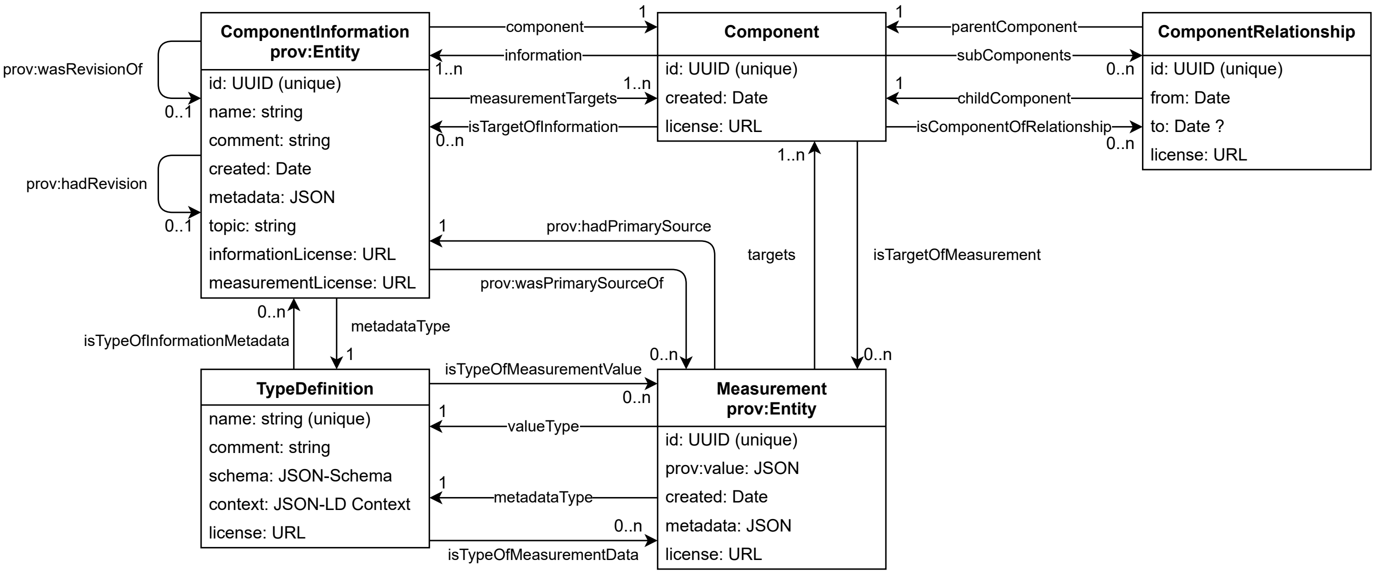

Fig. 3. The contextualization model linking objects, their metadata and measurement data. All the data and metadata are always associated with a `TypeDefinition` to ensure **P3**. Measurements, as the central element, are linked to the system they have been measured with, to the component of interest, and to the configuration and setup of both at the time of measuring. For increased findability all relations are bidirectional.

parent component of the measuring device, measurements are explicitly linked via a dedicated relation to the component of interest.

All data points are identified by using an identifier according to UUID4 to ensure uniqueness. Because the identifier will never change, **P1** is fulfilled. Lastly, to fulfill **P3** all component information and all measurements must conform to a specific type definition, to which the data is linked. In general, all relations are bidirectional for easy browsability and discoverability. By adding a mandatory license field to each entity of the model principle **R1.1** is achieved.

C. Filtering the Stored Data

According to **P6** it must be possible to search and filter the database by any field in the data model and the stored metadata. This allows users to retrieve the data by setting the context in which the data was created. This context can primarily be described by specifying an interval in the past during which the data was produced, the types of the data itself and the related metadata and values of attributes of related objects. So, the following filtering techniques capable of utilizing a multitude of filter arguments are introduced:

Date Filtering To retrieve data and metadata of a certain interval, two arguments can be supplied. If the argument `from` is used, only data which is younger than the specified date is returned. If the argument `to` is given, only data which is older than the specified date is returned.

Associated Type Filtering This technique allows filtering data by a given metadata or value type. Furthermore, it allows retrieving all measurement values, which are associated with specific component information or produced by a specific component.

Custom Filter Expressions The ecosystem has to support filtering by any arbitrary attribute or field of the model and of the stored metadata. Thus, it must enable users to write custom filtering expressions, which address the a priori unknown properties of the metadata stored. Otherwise, these fields are neither filterable nor searchable.

Based on the concepts of architecture, model and search routines, a prototypical ecosystem and middleware have been implemented at the author's institutes, which will be explained in the next section.

V. Implementation and Validation

The architecture, model and filtering techniques have been prototypically implemented to demonstrate and validate the approach. The resulting software artifacts have been deployed in a real-world test bed and tested with field data from machines and sensors. Details of the specific implementation and validation are described in the following.

A. Implementation

The authors' implementation is completely based on open-source technologies and libraries. The system is implemented in TypeScript, which is translated into JavaScript so that NodeJS libraries can be used without restrictions. The *API* is realized via a *REST API* using the HTTP protocol, enriched by the HTTP Memento protocol [25] to provide the time-based filtering technique. Measurement data is exchanged in JSON format and sent via the MQTT protocol. JSON is fully compatible with JSON-LD used to define and describe the context of the data and the type schemata. To achieve scalability, the ingestion service has been split into three subcomponents: ingress, buffering and ingestion. By that, incoming measurement data can be validated, even if the amount of data is very high. For that RabittMQ [26] has been used, as the quality of service feature of RabbitMQ ensures, that no data buffered is lost. This way, potential bottlenecks in the validation of incoming data and writing accepted data to the database can be mitigated.

The influence of the latter should have been further reduced by using PostgreSQL with the TimescaleDB extension which splits the table into smaller chunks, so-called *hypertables*. This strongly increases the insertion rate on tables with huge amounts of data. However, hypertables do not support foreign keys [27], which conflicts with the contextualization models that require a foreign key from `Measurement` to `Component`, `ComponentInformation` and `TypeDefinition`. Thus, the TimeScaleDB extension has not been used in the final version of the prototype, which theoretically reduces the insertion rate of the database if the table contains large amounts of data [28]. The data model is implemented using TypeORM [29], which integrates well with the other tools used and reduces implementation and maintenance effort due to its easy-to-use *API*. To deal with multiple unrelated object trees in the database, a hidden root node is introduced, which can not be queried but serves as the parent node for all objects which do not have a parent.

All data in the database is stored in JSONB. Thus SQL operators and functions, both defined in ISO/IEC 19075-6:2021 [30], are utilized to implement the *Associated Type Filtering* and *Custom Filter Expressions*.

B. Validation and Evaluation

While the fulfillment of the six prerequisites, defined in Section II-C, has already been discussed in Section IV (c.f. Table II), the validation and evaluation of the system have been conducted in three ways:

1) An exemplary technical validation of the implemented system has been conducted by modeling the varying real-world test-bed shown in Figure 1. Based on that data of that test bed has been collected and stored and queried later.
2) A performance evaluation of the implemented system regarding its ingestion capabilities and scalability.
3) Theoretic FAIRness assessment of the developed architecture and the contextualization model employing state-of-the-art FAIR evaluation frameworks.

1) Technical Validation: For technical validation, the implemented prototype has been integrated into the IoT infrastructure. The measurements produced by measuring devices are represented and distributed according to the SOIL data model [31]. For ingestion in the developed *FAIR Sensor Ecosystem* the definitions of measurements according to the SOIL model are mapped to `TypeDefinitions` of the *FAIR Sensor Ecosystem*. The test setup consisted of a laser tracker producing high-frequency position data and different targets measured and attached to different machines or parts. By that, the setup and configuration were modified between different measurement runs. Additionally, environmental data collected by multiple different sensors have been included in the test. To approve the system design and implementation concerning the requirements defined in Section II-C four steps have been taken: (i) ingestion of valid real data, (ii) ingestion of real data modified to be invalid, (iii) retrieval of data using the custom filter functions, (iv) reconstruction of the setup and configuration at a given point in time. The technical validation was carried out successfully and showed that the concept and the implementation worked as desired. The high-frequency data could be ingested without loss of data, and invalid data was rejected without exceptions. Moreover, all different setups and configurations could be fully reconstructed.

2) Performance Analysis: For the performance evaluation, the implementation has been deployed to a dedicated computer (Windows 10, Version 21H2, OS build 19044.1706, AMD FX-8350 8 Core CPU, 32 GB DDR3 RAM, Samsung MZ-7TE120BW SSD connected via SATA). To ensure that the evaluation of the performance of the implementation does not interfere with the rest of the required tools, RabbitMQ, MQTT, PostgreSQL and the performance evaluation script have been set up on another computer (MacBook Pro 2021, M1 Max, 32 GB RAM, macOS Monterey 12.4). Real-life data has been mimicked using ESP32 microcontrollers, generating a constant data stream of at least 3200 measurements each second, which is comparable to the expected amount of data in a real-life production environment [32]. First, the ingress capability of the system has been investigated. The ingress service was able to handle even more than 7k measurements per second and the performance cap was still not reached. Second, the performance of the ingestion service including the schema validation was tested. To determine the overhead of schema validation the tests have been executed with and without schema validation. The result of the performance of the ingestion service is shown in Figure 4 and demonstrates the scalability of the system using parallelized ingestion workers. Nevertheless, the type validation and processing of metadata are a significant overhead compared to processing and storing the values only.

While these two validations only proved the technical applicability of the system and validated the specific implementation, the authors also assessed the FAIRness of the data model and data stored according to the model using multiple state-of-the-art FAIRness assessment tools and frameworks.

3) FAIRness Assessment: As there does not exist a standard approach for the evaluation of the FAIRness of data, we use three different approaches for assessing the FAIRness of the data stored in the *FAIR Sensor Ecosystem* according to the contextualization model and compare their scores. Namely, the *FAIR Metrics* proposed by Wilkinson et al. [33], the *RDA FAIR Data Maturity Model* [34] and the *Data Stewardship Wizard* [35].

The *FAIR Metrics* framework allows for manual and automatic evaluation, which both have been applied. All metrics are evaluated on a binary scale. The automatic evaluation has been done with the tool from Wilkinson et al. The automatic evaluation yields a score of 12 out of 22, where deductions are mainly due to the custom unrecognized data model, a non-fitting identification scheme and missing specific attributes (such as `title`). The manual assessment reveals a score of 12 out of 14, caused by the custom data model and missing links to externally stored data. Similar results have been obtained using the *RDA FAIR Data Maturity Model* and the *Data Stewardship Wizard*. The assessment based on the latter can be viewed at [36]. Low scores in interoperability are mainly caused by the custom contextualization model, although all the attributes' keys are taken from available ontologies.

To evaluate the overall FAIRness, the individual results of all considered frameworks have been mapped to a percentage scale to be comparable. The outcome is shown in Table III. The score of zero for reusability for the Maturity Model results from the indicators which assess the use of a community standard. The indicators are stated as essential, but the criteria can not be fulfilled when no standard exists.Except for the Maturity Model, which defines a formula for weighting the indicators, the rating for each category is computed as the mean of the individual metrics. The overall rating was also calculated as the mean value of the four categories. The high discrepancy between the evaluation of two different frameworks for one category mainly results from differences in the defined metrics and different weighting of these. Moreover, the number of metrics differs significantly between categories and frameworks. Summing up, the overall FAIRness of the stored data can be considered high, especially for findability and accessibility. The interoperability of the system and the data could be improved by using terms of established ontologies for the elements and relations of the contextualization model.

The satisfactory result of the FAIRness analysis shows, that the adaption of the FAIR guiding principles in terms of the prerequisites defined in Section II-C has been successful. This is further underlined, by the evaluation of the design and implementation of the *FAIR Sensor Ecosystem* demonstrating the capabilities and applicability of the ecosystem, which is built upon these prerequisites. It has been shown, that the ecosystem covers all defined prerequisites and can process and store large amounts of FAIR sensor data.

TABLE III
CONSOLIDATED ASSESSMENT OF THE FAIRNESS OF DATA STORED IN THE PROPOSED *FAIR Sensor Ecosystem* GIVEN IN PERCENT. WHEREAS THE OVERALL FAIRNESS RATINGS DIFFER SIGNIFICANTLY FROM FRAMEWORK TO FRAMEWORK, THE AVERAGE RATINGS FOR THE FOUR BASE PRINCIPLES ARE COMPARABLE.

Framework	F	A	I	R	**Average**
FAIR Metrics (manual) [33]	100	100	60	100	**90**
FAIR Metrics (automatic) [33]	25	20	71	100	**54**
Maturity Model [34]	100	100	20	0	**55**
Stewardship Wizard [35]	80	83	90	84	**84**
Average	**76**	**76**	**60**	**71**	**71**

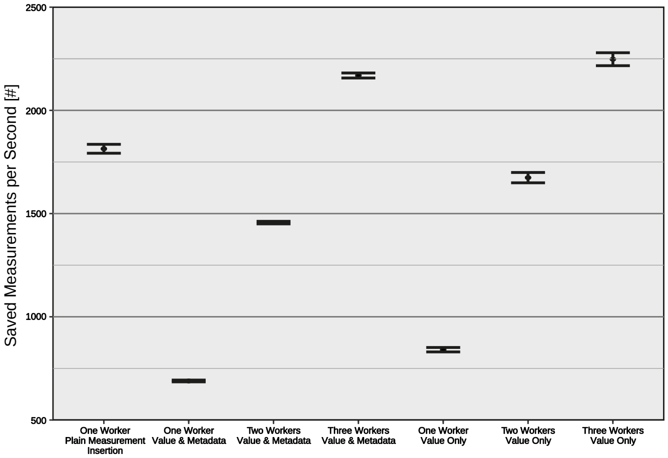

Fig. 4. Evaluation results of the performance analysis of the ingestion service. The plot shows the averages of how many sensor values could be stored per second, where the bars represent a 95% confidence interval. Tests have been carried out with a different number of ingestion workers and with and without additional metadata and type validation. The performance tests reveal a significant overhead introduced by the validation of (meta-) data but also prove the scalability of the system.

VI. Conclusion and Outlook

Adoption of the FAIR guiding principle is vital to establish data management for the manufacturing industry which fully enables data-driven applications. Although contextualization is crucial to make sense of data and reuse it properly, previously published work does not address the contextualization of sensor data sufficiently. Thus, the authors focus on this research gap, by introducing the *FAIR Sensor Ecosystem*: a middleware and storage application adhering to the FAIR principles, which validates measurement data, links the data with relevant meta information and stores it in the database persistently. The system can store arbitrary but structured data. To ensure interoperability all data stored must conform to user-defined data schemas. Via a *REST API*, it is possible to query and reconstruct the context of measured data, even if the setup and configuration have been changed since the time of data acquisition. The approach has been successfully validated by the implementation of a prototype system. Moreover, the FAIRness of data stored in the system has been evaluated employing state-of-the-art FAIRness assessment frameworks, and a positive FAIRness rating was obtained.

The presented work focused on relating measurements and objects within a closed system so that all data is stored in a single database. In the future, the authors plan to extend the system so that a link to external data sources is possible. The interoperability could be improved by extending the data model so that ontologies, like SSN, can be incorporated, which requires a more fine-grain representation of objects and the possibility to qualify relations more freely. This flexibility in the definition of the data model allows for representing the real-world scenario even more precisely. Furthermore, the actual FAIRness of the stored data still depends heavily on the users' diligence in defining and maintaining the object model. This could be improved by including automated tests that assess the FAIRness of the stored data and report the FAIRness score to the user.

Concluding, the presented work shows, that the conceptual adoption and technical realization of the FAIR guiding principles for the domains of production and manufacturing can advance the utility and long-term (re-)usability of sensor data. By using the presented *FAIR Sensor Ecosystem* for persistent storage of measurement data and its context information, long-term utilization of measurement data could be facilitated. Nevertheless, the FAIRness assessment reveals that state-of-the-art FAIRness assessment frameworks are mainly tailored to scientific data. To fully leverage stakeholders to make optimal reuse of stored measurement data, the data must further be compliant with community-specific standards, which are currently not sufficiently defined for the domains of production and manufacturing. Thus, as an orthogonal research direction, the establishment of these community-specific standards is fundamental for obtaining FAIR production data.

Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Project-ID 432233186 – AIMS. The authors would further like to thank Jonas Schlabertz for his contributions to the presented FAIR Sensor Ecosystem as part of his Master's thesis.

References

[1] J. Pennekamp, R. Glebke, M. Henze *et al.*, "Towards an Infrastructure Enabling the Internet of Production," in *Proceedings of the 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS '19)*. IEEE, 2019, pp. 31–37.

[2] M. Khan, X. Wu, X. Xu, and W. Dou, "Big Data Challenges and Opportunities in the Hype of Industry 4.0," in *Proceedings of the 2017 IEEE International Conference on Communications (ICC '17)*. IEEE, 2017.

[3] P. Brauner, M. Dalibor, M. Jarke *et al.*, "A Computer Science Perspective on Digital Transformation in Production," *ACM Transactions on Internet of Things*, vol. 3, no. 2, 2022.

[4] L. Spendla, M. Kebisek, P. Tanuska, and L. Hrcka, "Concept of Predictive Maintenance of Production Systems in Accordance with Industry 4.0," in *Proceedings of the 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI '17)*. IEEE, 2017, pp. 000 405–000 410.

[5] P. Dahlem, M. P. Sanders, H. Birck Fröhlich, and R. H. Schmitt, "Hybrid model approaches for compensating environmental influences in machine tools using integrated sensors," *at - Automatisierungstechnik*, vol. 68, no. 6, pp. 465–476, 2020.

[6] R. Isele and N. Arndt, "Mit semantischer Datenverwaltung Big Data in den Griff bekommen," *Wirtschaftsinformatik & Management*, vol. 8, no. 4, pp. 56–63, 2016.

[7] M. Grönewald, P. Mund, M. Bodenbenner *et al.*, "Mit AIMS zu einem Metadatenmanagement 4.0: FAIRe Forschungsdaten benötigen interoperable Metadaten," in *E-Science-Tage 2021: Share Your Research Data*. heiBOOKS, 2022, pp. 91–104.

[8] T. Stehle and U. Heisel, *Konfiguration und Rekonfiguration von Produktionssystemen*, 1st ed. Springer, 2017, pp. 333–367.

[9] B. Montavon, "Virtual Reference Frame Based on Distributed Large-Scale Metrology Providing Coordinates as a Service," Ph.D. dissertation, RWTH Aachen University, 2021.

[10] J. Pennekamp, A. Belova, T. Bergs *et al.*, "Evolving the Digital Industrial Infrastructure for Production: Steps Taken and the Road Ahead," in *Internet of Production: Fundamentals, Applications and Proceedings*. Springer, 2023.

[11] T. Tanhua, S. Pouliquen, J. Hausman *et al.*, "Ocean FAIR Data Services," *Frontiers in Marine Science*, vol. 6, 2019.

[12] P. Groth, H. Cousijn, T. Clark, and C. Goble, "FAIR Data Reuse – the Path through Data Citation," *Data Intelligence*, vol. 2, no. 1-2, pp. 78–86, 2020.

[13] M. D. Wilkinson, M. Dumontier, I. J. J. Aalbersberg *et al.*, "The FAIR Guiding Principles for scientific data management and stewardship," *Scientific Data*, vol. 3, 2016.

[14] M. Behery, P. Brauner, H. A. Zhou *et al.*, "Actionable Artificial Intelligence for the Future of Production," in *Internet of Production: Fundamentals, Applications and Proceedings*. Springer, 2023.

[15] L. Gleim, J. Pennekamp, M. Liebenberg *et al.*, "FactDAG: Formalizing Data Interoperability in an Internet of Production," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3243–3253, 2020.

[16] M. D. Wilkinson, S.-A. Sansone, E. Schultes *et al.*, "A design framework and exemplar metrics for FAIRness," *Scientific Data*, vol. 5, no. 1, p. 180118, 2018.

[17] M. Compton, P. Barnaghi, L. Bermudez *et al.*, "The SSN ontology of the W3C semantic sensor network incubator group," *Journal of Web Semantics*, vol. 17, pp. 25–32, 2012.

[18] Open Geospatial Consortium, "OGC SensorThings API Part 1: Sensing," OGC 15-078r6, 2016.

[19] S. Käbisch, T. Kamiya, M. McCool, V. Charpenay, and M. Kovatsch, "Web of Things (WoT) Thing Description," W3C Rec., 2020.

[20] M. Lagally, B. Francis, M. McCool *et al.*, "Web of Things (WoT) Profile," W3C Working Draft, 2023.

[21] A. Cimmino, M. McCool, F. Tavakolizadeh, and K. Toumura, "Web of Things (WoT) Discovery," W3C Candidate Rec. Snapshot, 2023.

[22] L. Gleim, J. Pennekamp, L. Tirpitz *et al.*, "FactStack: Interoperable Data Management and Preservation for the Web and Industry 4.0," in *Proceedings of the 19th Symposium for Database Systems for Business, Technology and Web (BTW '21)*, vol. P-311. Gesellschaft für Informatik, 2021, pp. 371–395.

[23] J. C. Kirchhof, B. Rumpe, D. Schmalzing, and A. Wortmann, "MontiThings: Model-Driven Development and Deployment of Reliable IoT Applications," *Journal of Systems and Software*, vol. 183, 2022.

[24] T. Lebo, S. Sahoo, and D. McGuinness, "PROV-O: The PROV Ontology," W3C Rec., 2013.

[25] H. van de Sompel, M. L. Nelson, R. Sanderson *et al.*, "Memento: Time Travel for the Web," arXiv:0911.1112, 2009.

[26] VMware, Inc., "Messaging that just works – RabbitMQ," https://www.rabbitmq.com/, 2007.

[27] jvujcic, "Foreign key to hypertable," https://github.com/timescale/timescaledb/issues/498, 2018 (accessed March 25, 2023).

[28] Timescale, Inc., "Why use TimescaleDB over PostgreSQL?" https://docs.timescale.com/timescaledb/latest/overview/how-does-it-compare/timescaledb-vs-postgres, 2021 (accessed March 25, 2023).

[29] TypeORM, "TypeORM - Amazing ORM for TypeScript and JavaScript," https://typeorm.io/, 2017.

[30] ISO, "Information technology — Guidance for the use of database language SQL — Part 6: Support for JSON," ISO/IEC 19075-6:2021, 2021.

[31] M. Bodenbenner, M. P. Sanders, B. Montavon, and R. H. Schmitt, "Domain-Specific Language for Sensors in the Internet of Production," in *Proceedings of the 10th Congress of the German Academic Association for Production Technology (WGP '20)*, vol. 20. Springer, 2020, pp. 448–456.

[32] W. Yu, T. Dillon, F. Mostafa, W. Rahayu, and Y. Liu, "A Global Manufacturing Big Data Ecosystem for Fault Detection in Predictive Maintenance," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 183–192, 2020.

[33] M. D. Wilkinson, M. Dumontier, S.-A. Sansone *et al.*, "Evaluating FAIR maturity through a scalable, automated, community-governed framework," *Scientific Data*, vol. 6, no. 1, 2019.

[34] C. Bahim, C. Casorrán-Amilburu, M. Dekkers *et al.*, "The FAIR Data Maturity Model: An Approach to Harmonise FAIR Assessments," *Data Science Journal*, vol. 19, no. 1, 2020.

[35] R. Pergl, R. Hooft, M. Suchánek, V. Knaisl, and J. Slifka, ""Data Stewardship Wizard": A Tool Bringing Together Researchers, Data Stewards, and Data Experts around Data Management Planning," *Data Science Journal*, vol. 18, no. 1, 2019.

[36] J. Schlabertz, "Assessement results with the Data Stewardship Wizard," https://researchers.ds-wizard.org/projects/54f6e55d-919a-4f1d-8a39-6c27f0b30ac8/metrics, 2022 (accessed March 25, 2023).