An Open Dataset Storage Standard for 6G Testbeds

Gilles Callebaut*, Michiel Sandra†, Christian Nelson†, Thomas Wilding‡,
Daan Delabie*, Benjamin J. B. Deutschmann‡, William Tärneberg†,
Emma Fitzgerald†, Anders J. Johansson†, Liesbet Van der Perre*
* KU Leuven, ESAT-Wavecore, Ghent Technology Campus, B-9000 Ghent, Belgium
† Department of Electrical and Information Technology, Lund University, SE-221 00 Lund, Sweden
‡ Graz University of Technology, Austria

***Abstract*—The emergence of sixth-generation (6G) networks has spurred the development of novel testbeds, including sub-THz networks, cell-free systems, and 6G simulators. To maximize the benefits of these systems, it is crucial to make the generated data publicly available and easily reusable by others. Although data sharing has become a common practice, a lack of standardization hinders data accessibility and interoperability. In this study, we propose the Dataset Storage Standard (DSS) to address these challenges by facilitating data exchange and enabling convenient processing script creation in a testbed-agnostic manner. DSS supports both experimental and simulated data, allowing researchers to employ the same processing scripts and tools across different datasets. Unlike existing standardization efforts such as SigMF and NI RF Data Recording API, DSS provides a broader scope by accommodating a common definition file for testbeds and is not limited to RF data storage. The dataset format utilizes a hierarchical structure, with a tensor representation for specific experiment scenarios. In summary, DSS offers a comprehensive and flexible framework for enhancing the FAIR principles (Findability, Accessibility, Interoperability, and Reusability) in 6G testbeds, promoting open and efficient data sharing in the research community.**

***Index Terms*—6G, dataset, measurements, testbed, channel sounding, simulations**

I. Introduction

Both academia and industry are actively developing new sixth-generation (6G) testbeds, such as sub-THz networks, cell-free (CF) systems, and 6G simulators. To benefit from these systems, the measurements, or more generally, the generated data should be publicly available and convenient to be reused by others. While the former has become the norm, the latter has not. Therefore, a common data format and interface are imperative to adhere to the Findability, Accessibility, Interoperability, and Reuse of digital assets (FAIR) principles [1].

We propose a Dataset Storage Standard (DSS) [2] i) to facilitate exchanging data and ii) to conveniently create processing scripts in a testbed-agnostic fashion. The latter enables developers and researchers to share processing scripts for the testbeds without intermediate scripts to tailor it to the specific output format of one testbed. Additionally, DSS is designed to also store data originating from simulations, enabling the processing of experimental and simulation with the same scripts and tools. The open-source interface documentation, including examples, is hosted on GitHub. This is a living document; for the most up-to-date information, please refer to the GitHub repository [2].

As evident from [3, Table XI and XII] and [4], *a significant number of testbeds are presently in the design and construction stages for 6G. This underscores the pressing need for standardized storage, processing, and documentation of experiments. Fortunately, DSS arrives at precisely the right moment to address this requirement.*

Several standards are being proposed and used in both industry and academia to record or store measurement data. Examples are Network Common Data Form (NetCDF), Hierarchical Data Format version 5 (HDF5), Digital RF [5], Signal Metadata Format (SigMF) [6], and NI RF Data Recording API [7]. NetCDF and HDF5 are general-purpose data formats used in various scientific domains, while Digital RF and SigMF are specialized formats for radio frequency (RF) and signal data. The NI RF Data Recording API is specific to National Instruments hardware and software solutions. Next to RF research, several data storage standards are proposed in astronomy, having similar requirements [8], [9] to DSS. Flexible Image Transport System (FITS) is one of the most widely used data storage standards in astronomy. It is a flexible and self-descriptive format, primarily designed for storing astronomical images and tables. FITS files include headers that contain metadata describing the content of the data. The format is highly extensible and supports various data types, making it suitable for a wide range of astronomical data. Another standard is Virtual Observatory Table (VOTable), which is an XML-based standard designed for representing tabular data, such as catalogs and data tables. It is widely used in the Virtual Observatory framework, facilitating the exchange of data between different observatories and data archives. VOTable files include metadata and descriptions of table columns, making them self-descriptive and interoperable. Although having some similarities in the requirements, FITS and VOTable are tailored to astronomy and are therefore not directly applicable to 6G testbeds. An overview and differences of these standards are provided

6GTandem has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101096302. The REINDEER project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101013425.

TABLE I: Comparison of Data Storage Standards

Standard	Data Stored	Storage Method	Focus	Hardware description
NetCDF	Structured scientific data	Hierarchical format	General-purpose	✗
HDF5	Structured and unstructured data	Hierarchical format	General-purpose	✗
Digital RF	RF and signal data	Custom format	RF signal data	✗
SigMF	Signal metadata	Lightweight format	Signal metadata	✗
NI RF Data API	RF data (NI-specific)	SigMF format	RF data management	✓(only NI)
FITS	Measurement-specific	Table structures	Astronomical observations	✗
VOTable	Measurement-specific	Hierarchical tabular structure	Astronomical observations	✗
DSS	Measurement-specific	Custom format (HDF5 + NetCDF)	6G experiments and data storage	✓

in Table I.

In contrast to other standardization endeavors in the context of 6G testbeds, such as SigMF and NI RF Data Recording API, DSS has a broader scope. It supports a common definition file for testbeds and is not limited to storing RF data exclusively. DSS emphasizes describing hardware-agnostic experiment descriptions. While not designed for experiment configuration, DSS does not impose limitations on the use of it for that purpose. Furthermore, DSS supports mobility, enabling the movement of the user equipments (UEs) about the testbed. This can also be very useful for creating a database of different experiments.

At its core, DSS is built upon the data model of the widely recognized HDF5 and NetCDF storage standards, and data can be stored using either of these formats. DSS introduces an additional layer in the form of metadata description files, customizing it to better suit the requirements of 6G testbeds.

II. DSS Architecture

The standard comprises three main components: i) human-readable description files detailing experiments and testbeds (Section II-A), ii) a specific data storage format based on the type of measurement (Section II-B), iii) and an application program interface (API) (Section II-B). These are illustrated in Fig. 2 and elaborated below.

The *description files* are used to i) programmatically extract relevant information regarding the stored dataset, and ii) to model the hardware setup to perform simulations. Consequently, the behavior of hardware is defined in the *hardware component*. This can range from the frequency response of the utilized cables to the power amplifier (PA) AM/AM profile. By defining how to store and describe this metadata, the FAIR principles are forced to be met and calibration procedures and hardware imperfections become transparent to the users of the testbed and the dataset.

A. The DSS Description Files

A number of files are required to interpret and explain a conducted experiment. The files are structured in a way such that they can conveniently be re-used in different experiments and testbeds. For instance, a *data source* can describe an SDR, as well as a DAQ system. It is a system that actively influences (e.g., through samples) and returns the data in a digital format. This is in contrast to *hardware components*, that passively affect the signals and data, e.g., an low-noise amplifier (LNA). A testbed consists of one of several chains of *data sources* and *hardware components*. An experiment can refer to an environment file, describing, e.g., the room dimensions, and 3D scans and models. The experiment includes the testbeds used and a number of measurements, each having different parameters.

1) Testbed Description Files: A testbed description file is a hierarchy of logical chains of data sources and hardware components that reflect the real measurement/testbed setup. An example of a testbed description file, based on the Techtile testbed [10], is depicted in Fig. 3 and is based on the example shown in Fig. 1. The testbed consists of two subsystems, an acoustic system, and an RF infrastructure. The RF data chain consists of 140 chains of an SDR with, in this example, only one antenna connected to it (while two are possible). The acoustic part of the testbed encompasses one central DAQ system connected, in this example, to 100 amplifiers and MEMS microphones. Each $\star$ in Fig. 3 represents a reference to the data source or hardware component, defined elsewhere (see the following sections).

2) Data Source Description Files: Each data source has a unique type associated with it, and each type in turn has a number of configurable parameters. An example of a data source is an SDR, which has a set of configurable parameters such as, e.g., the sampling rate or bandwidth.

3) Hardware Components: Various hardware components, including antennas, cables, PAs, filters, and sensors, are integral to testbeds and affect recorded datasets (see Fig. 1). Knowledge of these systems is often required to calibrate hardware or to remove the effect of the antenna pattern when processing the dataset. Each hardware component can have one or several attributes associated with it. An example is presented in Fig. 1, where the frequency response of the utilized amplifier is stored in a DSS-compliant dataset file.

4) Environments: Optionally, environment-related metadata can be stored, such as the physical properties of the room (air absorption, material properties, temperature, noise values, room dimensions, interference signals in surroundings, ...). However, in DSS there are currently no constraints on the type of data and the storage format.

5) Experiments: The experiment description file contains the information about a conducted experiment and thus refers to other description files. The experiment description file can also contain variables such as temperature and references to photo and video files.

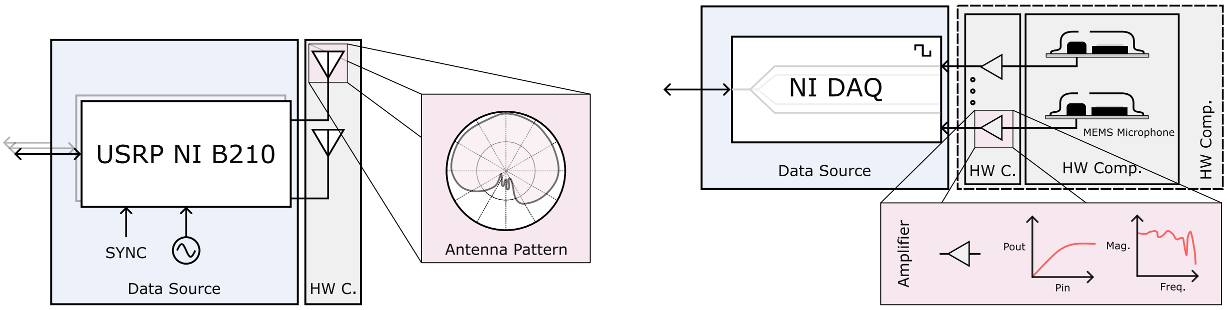


Fig. 1: Example of two testbed setups (inspired by the Techtile testbed [10]). **Left**: Illustrating multiple universal software radio peripheral (USRP) NI B210 software-defined radios (SDRs) having each two antennas connected. The antenna pattern is an attribute of the specific antenna hardware component. **Right**: One central data acquisition system (DAQ) samples multiple micro-electromechanical systems (MEMS) microphones. Each microphone is connected to an amplifier before being sampled. Two examples of attributes related to an amplifier is depicted, i.e., the amplitude modulation to amplitude modulation (AM/AM) and frequency response.

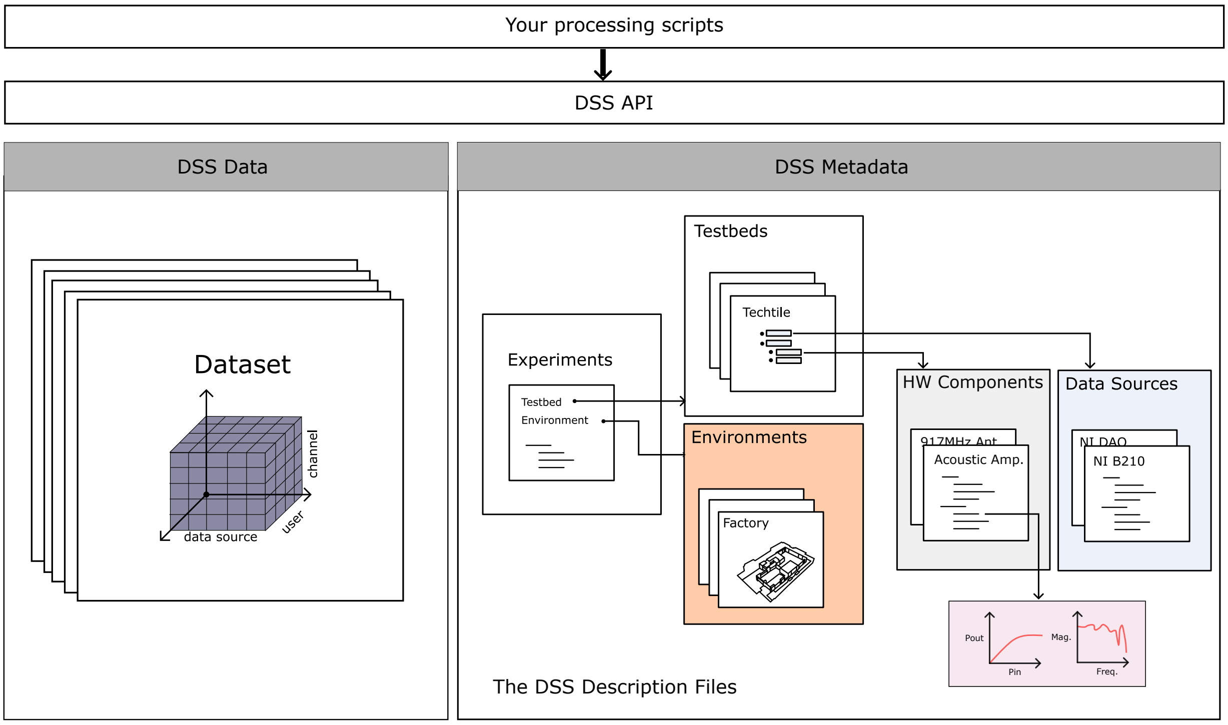


Fig. 2: Illustration of the proposed standard, consisting of the description files, datasets, and API.

B. Dataset

The description files are utilized when reading/storing the data in a common format. A simple dataset structure is used to store the data of a specific experiment scenario. A tensor is used, which has a data and a time dimension. These dimensions are tailored to the dataset type, exemplified by the type of channel measurement that is performed, e.g., RF channel-sounding or acoustic measurements. Next to predefined dataset types, a user of DSS is allowed to specify new types and thus extend the DSS.

The dataset is stored in an HDF5 [11] or NetCDF [12] format to maintain interoperability with a range of programming languages.

C. Application Programming Interface

The DSS API functions as an additional layer atop the standardized storage and description files, enhancing the utility and adaptability of DSS. It is imperative to emphasize that, while the API will be developed, DSS maintains its primary focus on the meticulous structuring of data and metadata.

The DSS API is laid out to facilitate streamlined interaction with the standard and provide users with a versatile platform for extending functionalities. By adhering to the standardized data storage and metadata management conventions established by DSS, the API becomes a powerful tool for implementing common workflows and procedures. This approach allows DSS to become the unification of different libraries and tools for a wide range of measurement types. This way the community can focus on the experiments, and use already available post-processing procedures embedded in the DSS API.

```yaml
Techtile: &Techtile
  name: "Techtile"
  description: "Small description"
  url: "https://github.com/techtile-by-dramco"
  level: "L3" # Level of testbed, see [4]
  data_chains:
    -   label: "RF"
        chain:
            data_source:
                <<: *B210
            channel_chain:
                hardware_components:
                    - <<: *Techtile915MHzAntenna
                data_source_channel: "0"
            num_data_source_chains: 140 # num_channels
↪   will be num_chains x 2 channels
            channel_locations: # physical location of
↪   each channel source
                file: techtile_antenna_locations.npy
                loc_unit: "m" # unit used for source
↪   location
    -   label: "Acoustic"
        chain:
            data_source:
                <<: *DAQ
            channel_chain:
                hardware_components:
                    - <<: *TechtileMicrophonePA
                    - <<: *TechtileMicrophone
                data_source_channel: "0:100"
        num_data_source_chains: 1 # num_channels will
↪   be num_chains x 2 channels
        channel_locations: # physical location of each
↪   channel source
            file: techtile_microphone_locations.npy
            loc_unit: "m" # unit used for source
↪   location
```

Fig. 3: Example of an incomplete testbed description file.

III. Dataset Types – DSS in the field

A. The Channel-Sounding Dataset

The dataset containing a channel-sounding experiment or scenario, see e.g., [13]–[23], has minimal three dimensions: the number of transmitters, the number of receivers, and the data in the delay or frequency domain. Each transmitter or receiver can have one or more channels, as specified by the testbed or data source description. The mapping between the transmit and receive channels should be described in the experiment description file. Furthermore, each channel can have a position and orientation associated with it, allowing to, e.g., take antenna characteristics into account. Similarly, data in the delay or frequency domain can have a delay or absolute frequency coordinate associated with it. Depending on the type of measurement hardware that is used, different means of (post-)processing can be necessary, such as reduction of the measurement bandwidth, selection of a specific frequency range for application scenarios, or pulse shaping.

Depending on the available data, examples for application of channel-sounding datasets are for testing of positioning or channel estimation algorithms (e.g., using array processing on suitable datasets), analysis of propagation characteristics for propagation/environment modeling or machine learning-based feature extraction, but also wireless power transfer applications, requiring as minimal knowledge the transmit power settings. Below we briefly discuss the data that can be expected in a general channel-sounding dataset as well as two applications in form of positioning and wireless power transfer.

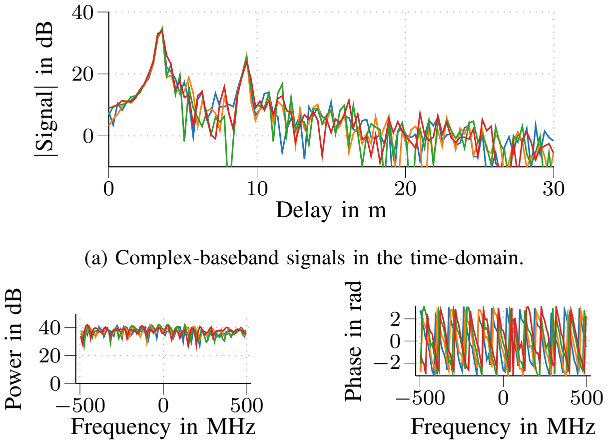
1) Measurement Data: Visualization of the measurement data, e.g., the channel impulse response (CIR) between a specific transmitter $i_{\text{tx}} = 0$ for each receive channel at a certain measurement time(index) $i_{\text{t}} = 0$ can be conveniently done via dedicated functions. These plot the channel response (CR), e.g., the received signal in complex baseband notation, or the complex-valued transfer function (TF) via

```python
dss.plot_cr((tx=0, rx=[0,1,2,3], t=0))
dss.plot_tf((tx=0, rx=[0,1,2,3], t=0))
```

which allows to quickly view and compare measurements. Examples are shown in Fig. 4 in form of the time- and frequency domain signals for a single transmitter location and 4 receiver locations spaced roughly $2\,\text{cm}$ apart for a measurement bandwidth of $1\,\text{Ghz}$ and carrier frequency of $6.95\,\text{GHz}$ (see [22] for details regarding the measurements).

2) Positioning: For developing positioning applications, a multitude of algorithms can be employed, including various combinations of range- or angle-based algorithms or even received signal strength (RSS)-based approaches. The data stored in DSS then needs to be formatted accordingly in post-processing to allow application of, e.g., channel estimation algorithms to extract ranges. This will commonly encompass the selection of the target bandwidth and center frequency, oversampling when required, or determination of parametric signal atoms for use with parametric channel estimators, see [22, Sec. IV-C]. A corresponding DSS-compliant dataset will thus necessitate a certain level of calibration to be performed to enable correct use of the measurement data, ideally de-embedding the properties of the measurement device from the measurement data up to a minimum level that cannot be performed by the end-user, or would be required for algorithm testing. This could encompass, e.g., calibration of the system response to remove effects that can be considered unique to the measurement hardware, but would generally require a description of the performed calibration procedure. Positioning algorithms will furthermore require the location of all sensor positions, encompassing the UE, agent or mobile device to localize, as well as the base stations (BSs), often termed (positioning) anchors. For the (positioning) anchors, information about the level of synchronization between these anchors is necessary. To incorporate hardware components such as antennas in algorithms, the orientation of these in the environment coordinate system needs to be specified in the DSS description file.

3) Wireless Power transfer: Using a dataset to evaluate wireless power transfer (WPT) requires accurate amplitude measurements or simulations. An essential part of the former is the calibration of linear systematic errors introduced by hardware components such as cables, antennas, connectors, etc. A DSS-compliant dataset includes the characterization of these hardware components. The manufacturer of a data source guarantees calibrated measurements at its ports, and thus the measurement reference planes are located there.



(a) Complex-baseband signals in the time-domain.

(b) Magnitude in the frequency domain.

(c) Phase in the frequency domain.

Fig. 4: Measurement signals between a single transmitter and 4 exemplary receiver locations spaced roughly 2 cm apart.

It is evident that adopting DSS will ensure that applications working with channel-sounding data can seamlessly process data from different DSS-compliant datasets without particular care about the involved error networks.

B. The Simulation Dataset

In the multifaceted realm of 6G research, both real-world experimentation and theoretical modeling are paramount. Here, the simulation dataset stands out as a powerful instrument, providing researchers with a refined method to emulate scenarios, validate theories, and build models, even before initiating real-world experiments. The DSS, with its intricate architecture as described earlier, bestows upon the simulation dataset several noteworthy features.

At its foundation, the simulation dataset ensures dimensional rigor. Every simulation run represents an iteration or instance of a chosen scenario. In contrast, the associated result captures the observational outcome from that specific iteration. There's also an inherent fluidity in parameter selection. Simulations within the DSS framework often pivot around varying input parameters, such as network arrival rates. This variance unfolds a rich spectrum of results, from metrics like mean throughput and delay to volumes of dropped packets. Such depth of outcomes allows researchers to understand system behavior across different conditions.

Additionally, the importance of temporal granularity is hard to overstate. Many simulations seek to delineate how systems evolve over time. The DSS accommodates this by offering time-based outputs, structuring data points as time-value pairs, painting a clear picture of temporal variations. An experiment can generate a plethora of results. With the DSS, researchers can decide whether each system node should be a unique data source or if they should consider the entire simulation logging as a singular unit, which is crucial for nuanced data interpretation and analysis.

There are several overarching advantages to employing the simulation dataset within the DSS. Its unmatched flexibility allows researchers to control every variable, recreating myriad scenarios, some of which might be challenging or even impossible to stage in the real world. Economically, simulations often trump real-world experiments, especially when the latter require specialized or rare resources. The ability to rerun simulations guarantees reproducibility, reinforcing the credibility of findings. In scenarios where real-world tests might pose risks, simulations provide a hazard-free environment to venture into. Moreover, the inherent scalability of the DSS simulations ensures systems can be studied at any scale, from the smallest networks to sprawling infrastructures.

In essence, the simulation dataset, meticulously crafted within the DSS paradigm, is poised to become an indispensable tool in 6G network research. It bridges the theoretical with the practical, ensuring that researchers are equipped with robust, reliable, and insightful tools, preparing them for the demands of the real world.

C. The Acoustic Dataset

A dataset containing the room impulse responses (RIRs) or audio fragments obtained from different speakers and microphone combinations is desired, e.g., [24]. The acoustic dataset shows a close analogy to the channel-sounding dataset, described in Section III-A. The three dimensions for this dataset consist of the number of speakers, the number of microphones and the data (RIRs or audio fragments). The location dependency of a speaker or microphone entails the use of different channels, when focusing on array configurations. In simulations, it may be possible, for example, that single microphones are placed at all desired locations simultaneously, meaning that only 1 channel is applicable. Measurements are more likely to use and move the same microphone (array). Nevertheless, this dataset can be used for both simulation and measurements.

Plotting the RIR of channel 0 between e.g., speaker 1 and microphone 14 can be conveniently done via:

```
dss.plot_rir((sp=1, mic=14, ch=0))
```

and results in, for example, Fig. 5.

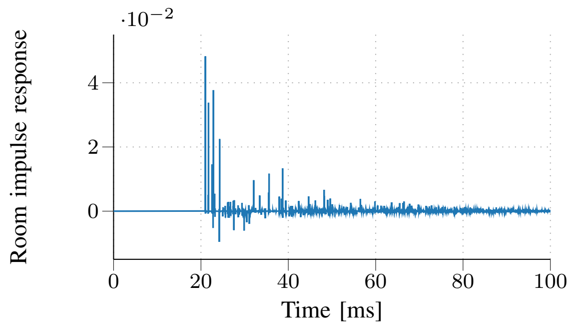


Fig. 5: RIR between the 14th microphone and 1st speaker.

IV. Conclusion and Future Extensions

The Dataset Storage Standard (DSS) architecture presented in this document represents a robust framework for the structured storage and management of scientific datasets. DSS offers a well-defined structure comprising description files, hardware component definitions, environments, and experiments, all working together to ensure the transparency, reusability, and interoperability of stored data. By focusing on standardized data storage and metadata management, DSS adheres to the Findability, Accessibility, Interoperability, and Reuse of digital assets (FAIR) principles, making datasets findable, accessible, and usable for a wide range of scientific and engineering applications. The inclusion of a flexible API enhances DSS's capabilities, enabling users to efficiently interact with and extend the standard for specific workflows and procedures. With its support for various dataset types, such as channel sounding and acoustic measurements, DSS proves versatile and adaptable to diverse research domains. In essence, the DSS architecture sets a strong foundation for effective data storage, retrieval, and analysis, contributing significantly to advancing scientific and experimental research through standardized and well-structured datasets.

The future outlook for DSS holds several promising opportunities for further development and extensions such as e.g., expanded dataset types, machine learning integration, and the extension of the API including data visualization tools.

In summary, the future of DSS is marked by its adaptability and openness to growth. By embracing emerging technologies and responding to the evolving needs of the scientific and engineering communities, DSS can continue to serve as a valuable standard for dataset storage and management, fostering greater collaboration and innovation in data-driven research.

REFERENCES

[1] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne *et al.*, "The FAIR Guiding Principles for scientific data management and stewardship," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[2] G. Callebaut, "Dataset Storage Interface," 2023. [Online]. Available: https://github.com/6G-Testbeds/Dataset-Storage-Interface

[3] C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas, J. S. Thompson, E. G. Larsson, M. D. Renzo, W. Tong, P. Zhu, X. Shen, H. V. Poor, and L. Hanzo, "On the Road to 6G: Visions, Requirements, Key Technologies, and Testbeds," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.

[4] G. Callebaut, L. Liu, T. Eriksson, L. V. der Perre, O. Edfors, and C. Fager, "6G Radio Testbeds: Requirements, Trends, and Approaches," 2023.

[5] R. Volz, W. C. Rideout, J. Swoboda, J. P. Vierinen, and F. D. Lind, "Digital RF." [Online]. Available: https://github.com/MITHaystack/digital_rf

[6] B. Hilburn, "SigMF Release v0.0.1," Jul. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1418396

[7] National Instruments, "NI RF Data Recording API v1.0.0." [Online]. Available: https://github.com/genesys-neu/ni-rf-data-recording-api

[8] M. Chen, H. Deng, F. Wang, and K. Ji, "A Survey on File Format for Data Storage in Radio Astronomy," in *2013 6th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 2013, pp. 324–327.

[9] A. W. Gunst and G. H. Kruithof, "Antenna data storage concept for phased array radio astronomical instruments," *Experimental Astronomy*, vol. 45, pp. 351–362, 2018.

[10] G. Callebaut, J. V. Mulders, G. Ottoy, D. Delabie, B. Cox, N. Stevens, and L. V. d. Perre, "Techtile Open 6G R&D Testbed for Communication, Positioning, Sensing, WPT and Federated Learning," in *2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2022, pp. 417–422.

[11] The HDF Group. (1997-NNNN) Hierarchical Data Format, version 5.

[12] R. Rew and G. Davis, "NetCDF: an interface for scientific data access," *IEEE Computer Graphics and Applications*, vol. 10, no. 4, pp. 76–82, 1990.

[13] S. Payami and F. Tufvesson, "Channel measurements and analysis for very large array systems at 2.6 GHz," in *2012 6th European Conference on Antennas and Propagation (EUCAP)*, Mar. 2012, pp. 433–437.

[14] A. O. Martínez, E. De Carvalho, and J. O. Nielsen, "Towards very large aperture massive MIMO: A measurement based study," in *2014 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2014, pp. 281–286.

[15] G. Sommerkorn, M. Käske, D. Czaniera, C. Schneider, G. Del Galdo, R. S. Thomä, and M. Walter, "Experimental and Analytical Characterization of Time-Variant V2V Channels in a Highway Scenario," in *2019 13th European Conference on Antennas and Propagation (EuCAP)*, Mar. 2019, pp. 1–5.

[16] D. Gaillot, P. Laly, N. Dahmouni, G. Delbarre, M. Van den Bossche, G. Vermeeren, E. Tanghe, E. Simon, W. Joseph, L. Martens, and M. Liénard, "Measurement of the V2I Massive Radio Channel with the MaMIMOSA Sounder in a Suburban Environment," in *2021 15th European Conference on Antennas and Propagation (EuCAP)*, Mar. 2021, pp. 1–4.

[17] S. Zelenbaba, B. Rainer, M. Hofer, D. Löschenbrand, A. Dakić, L. Bernadó, and T. Zemen, "Multi-Node Vehicular Wireless Channels: Measurements, Large Vehicle Modeling, and Hardware-in-the-Loop Evaluation," *IEEE Access*, vol. 9, pp. 112 439–112 453, 2021.

[18] M. Sandra, G. Tian, A. Fedorov, X. Cai, and A. J. Johansson, "Measurement-Based Wideband Maritime Channel Characterization," *The 17th European Conference on Antennas and Propagation (EuCAP 2023)*, 2023.

[19] M. Sandra, C. Nelson, and A. J. Johansson, "Ultrawideband USRP-Based Channel Sounding Utilizing the RFNoC Framework," in *2022 IEEE Conference on Antenna Measurements and Applications (CAMA)*, 2022, pp. 1–3.

[20] D. Löschenbrand, M. Hofer, L. Bernadó, S. Zelenbaba, and T. Zemen, "Towards Cell-Free Massive MIMO: A Measurement-Based Analysis," *IEEE Access*, vol. 10, pp. 89 232–89 247, 2022.

[21] D. Stanko, M. Döbereiner, G. Sommerkorn, D. Czaniera, C. Andrich, C. Schneider, S. Semper, A. Ihlow, and M. Landmann, "Time Variant Directional Multi-Link Channel Sounding and Estimation for V2X," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, Jun. 2023, pp. 1–5.

[22] T. Wilding, B. J. B. Deutschmann, C. Nelson, X. Li, F. Tufvesson, and K. Witrisal, "Propagation Modeling for Physically Large Arrays: Measurements and Multipath Component Visibility," 2023.

[23] G. Callebaut, S. Willhammar, A. P. Guevara, A. J. Johansson, L. Van Der Perre, and F. Tufvesson, "Experimental Exploration of Unlicensed Sub-GHz Massive MIMO for Massive Internet-of-Things," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 2195–2204, 2021.

[24] R. Glitza, L. Becker, and R. Martin, "Database of Simulated Room Impulse Responses for Acoustic Sensor Networks Deployed in Complex Multi-Source Acoustic Environments," in *31th European Signal Processing Conference (EUSIPCO)*, 2023.