AKjDPG 1: Hacky Hour (joint session AKjDPG/AGI)

Time: Thursday 14:00–17:30 Location: ZEU/0148

AKjDPG 1.1 Thu 14:00 ZEU/0148

Adamant: A JSON-Based Metadata Editor for Researchers — •Ihda Chaerony Siffa, Marjan Stankov, and Markus M. Becker — Leibniz Institute for Plasma Science and Technology (INP), Felix-Hausdorff-Straße 2, 17489 Greifswald, Germany

Adamant is a browser-based research data management (RDM) tool, specifically developed to systematically collect research metadata that is both machine- and human-readable. It utilizes the JavaScript Object Notation (JSON) schema specifications, where any valid schema can be rendered as an interactive and user-friendly web form. Users may create a JSON schema from scratch or provide an existing schema. Subsequently, users can provide inputs to the rendered form and generate a JSON document, which can be downloaded for further use. Adamant has found its usage in several research settings; namely, compilation of structured experiment metadata in conjunction with a generic electronic laboratory notebook, scientific instrument job requesting, and preparation of input data for plasma simulations. Overall, Adamant is an emerging generic RDM tool that eases day-to-day research activities as far as structured metadata is concerned.

AKjDPG 1.2 Thu 14:45 ZEU/0148

Hands-on data management with open-source software: CaosDB — •Florian Spreckelsen and Daniel Hornung — IndiScale GmbH, Göttingen, Germany

Data management involves the storing, searching, retrieving and analyzing of data sets and their connections and circumstances. Good data management makes valuable data reusable, for current and future users. It also makes data findable (*Where is the training data for sensor X of setup Y again?*) and adds real utility to data, because data can be embedded into context (*Which experimental settings were used for obtaining the data for project P, and how many failures were there?*).

The open-source toolkit CaosDB is a practical implementation of an agile data management approach designed to handle all these tasks, and much more: The structure of data can be modified later without losing old information and without the need to migrate existing data. This encourages agile implementation of data management workflows instead of delaying until the *perfect master plan* is ready. And CaosDB comes with a powerful Python client, so access is as easy as a few lines of code. This session consists of a short live demonstration of the CaosDB Python client, and participants are encouraged to follow along on their own machines. For this, they can install CaosDB's Python library and additional tools with `pip install caosdb caosadvancedtools` and make sure that they can load the library in Python with `import caosdb`. A Jupyter notebook will be made available online before the session.

AKjDPG 1.3 Thu 15:30 ZEU/0148

ELN integration into the open-source data management solution CaosDB — •Daniel Hornung, Florian Spreckelsen, Henrik tom Wörden, Timm Fitschen, and Thomas Weiss — IndiScale GmbH, Göttingen, Germany

Scientific research still often lacks professional data management, mostly because the dynamically evolving research environments lack suitable software tools. In contrast, standardized industrial processes can be integrated easily with existing data management software. Research work in the lab is increasingly documented with electronic lab notebooks (ELNs), which allow to conveniently enter device and experimental settings in a semi-structured way. This data is usually critical in the analysis of acquired raw data from instruments, e.g. for searching specific data sets or filtering by parameters. We successfully integrated the data management software CaosDB with the eLabFTW ELN, thus combining flexible lab input methods with an agile open source approach to data management.

We chose CaosDB over other solutions, because it allows to flexibly adjust the data model when necessary. This agility is required by the dynamic nature of scientific research activities and cannot be provided by rigid, SQL based approaches.

20 min. break

Invited Talk AKjDPG 1.4 Thu 16:00 ZEU/0148

Open data and open-source tools throughout research data life cycle: KCDC example — •Victoria Tokareva — Karlsruhe Institute of Technology, Institute for Astroparticle Physics, Karlsruhe, Germany

Open science essentials include open data, open source software, open access materials, open educational resources, etc. They provide substantial benefits to society like reproducibility of research, increased transparency and public acceptance of studies, simplified publication process, and enhanced public education. Ultimately, new opportunities become available for unique interdisciplinary studies performed by large diverse teams of specialists on publicly available datasets. Established in 2013, the KASCADE Cosmic Ray Data Centre (KCDC) exists simultaneously as an open archive for data of high-energy astroparticle physics experiments (such as KASCADE, KASCADE-Grande, LOPES, Maket-Ani, etc.), open source software and a web portal providing access to open educational resources. KCDC allows data selection with custom user data cuts using GUI or REST API and interactive online analysis of the selected data with integrated Jupyter Notebooks. From this talk, one can learn more about KCDC's functionality and get better understanding of open science and research data life cycle concepts. An example of machine learning based analysis employing the KCDC platform and deployment of the results as an application using Streamlit will be discussed. This work is partially supported by the DFG fund "NFDI 39/1" for the PUNCH4NFDI consortium.

AKjDPG 1.5 Thu 16:45 ZEU/0148

Interactive USB measurement device controlling with Python — •Benedikt Bieringer — Institute for Nuclear Physics, University of Münster

Although USB devices are central part of most lab experiments, in many cases their use is significantly limited by the provided manufacturers' software and drivers. This talk covers writing a (graphical) readout and controlling software in Python both by using proprietary drivers and by writing own user-space Python-based drivers in cases where the manufacturers' drivers limit the usage potential of the USB device. It gives an overview over writing Python modules for existing drivers in C++, writing USB drivers in Python using PyUSB by analyzing USB packets with Wireshark and writing an interactive plotting and controlling GUI with PyQt and PyQtGraph.