Research Ideas and Outcomes 11: e171656
doi: 10.3897/rio.11.e171656

Grant Proposal

FAIRJupyter4AI: A Corpus of Computational Notebooks for AI

Daniel Mietchen‡, Sheeba Samuel§

‡ FIZ Karlsruhe — Leibniz Institute for Information Infrastructure, Berlin, Germany
§ Chemnitz University of Technology, Chemnitz, Germany

Corresponding author: Daniel Mietchen (daniel.mietchen@fiz-karlsruhe.de),
Sheeba Samuel (sheeba.samuel@informatik.tu-chemnitz.de)

Reviewable v 1

Received: 10 Sep 2025 | Published: 15 Oct 2025

Citation: Mietchen D, Samuel S (2025) FAIRJupyter4AI: A Corpus of Computational Notebooks for AI. Research Ideas and Outcomes 11: e171656. https://doi.org/10.3897/rio.11.e171656

Abstract

Computational notebooks like Jupyter have transformed scientific and educational workflows in computational fields by combining code, text, and visualizations. They have also become a popular mechanism to share computational workflows. However, ensuring their reproducibility remains a persistent challenge due to often insufficiently documented direct and indirect dependencies, missing data, and inconsistencies in execution environments. Existing datasets lack the multimodal, fine-grained structure needed for AI applications. FAIRJupyter4AI aims to address this gap by creating a large-scale, AI-ready corpus of Jupyter notebooks enriched with executable code, markdown, outputs, and structured annotations. The project integrates these into a hybrid knowledge graph (KG) that incorporates symbolic, statistical, and execution-based representations. Key objectives include: curating diverse notebooks (initially Python, later R, with provisions for additional languages); automating reproducibility testing; building a KG for cross-notebook queries; training AI models for tasks like error repair and notebook generation; and fostering community use via APIs and integration with community platforms like NFDI or Hugging Face.

The project will be implemented using the infrastructure established by the NFDI Basic Service Jupyter4NFDI, in the upcoming Integration Phase of which (October 2025-September 2027) the applicants are actively involved. Its central JupyterHub provides

cross-consortial and cross-institutional access to scalable computing and data resources and associated software stacks for both research and training purposes.

The FAIRJupyter4AI work programme is structured around five interlinked work packages: (1) Data Collection & Curation, (2) Reproducibility Assessment, (3) Knowledge Graph Development (4) AI Model Training, and (5) Communication, Community & Sustainability. Key innovations include continuous updates and enrichment pipelines (avoiding static snapshots), unifying multimodal content for AI, and bridging reproducibility with AI. Building on prior work involving 27,000+ notebooks and the FAIR Jupyter Knowledge Graph, FAIRJupyter4AI will curate, annotate and release over 20,000 notebooks that are research-related and openly licensed. In addition, we will share a metadata corpus for 50,000 research-related notebooks, along with open-source tools, models, and associated documentation. By making Jupyter notebooks metadata FAIR, reusable, and machine-understandable, this project will set a new standard for reproducible and AI-enhanced computational science, and it will open up new opportunities for learning and teaching about computational reproducibility across multiple domains of research.

Keywords

Jupyter Notebook, Computational Reproducibility, AI, Jupyter4NFDI, Dataset

List of participants

- Daniel Mietchen, FIZ Karlsruhe Leibniz Institute for Information Infrastructure (FIZ Karlsruhe)
- Sheeba Samuel, Chemnitz University of Technology, Chemnitz

State of the art and preliminary work

Computational notebooks like Jupyter (Kluyver et al. 2016) have become central to modern scientific workflows, seamlessly integrating code, narrative text, outputs, and data. Despite their widespread adoption, reproducibility remains a significant challenge due to issues such as undocumented dependencies, code-documentation mismatches, and environmental drift. The heterogeneous and often unstructured nature of notebooks complicates automated analysis, reproducibility assessment, and the effective application of advanced AI methods. Existing KGs for reproducibility focus primarily on metadata and artifact relationships, lacking access to the rich, raw multimodal content and fine-grained annotations necessary for training sophisticated AI models that comprehend and interact with both computational logic and narrative context inside notebooks.

FAIRJupyter4AI addresses these gaps by developing a large-scale, high-quality, AI-ready corpus of Jupyter notebooks that aggregates the full spectrum of notebook content

- executable code, markdown narratives, rich outputs (e.g., plots, tables) - with associated data files and auxiliary information like error messages. This corpus is carefully cleansed and semantically annotated at a detailed level (linking code to explanations, tracing data provenance, and highlighting reproducibility-critical components). A knowledge graph that integrates symbolic, statistical (embeddings), and sequence-based representations (execution traces) to index and interconnect notebooks and their internal elements, enabling powerful structured queries and cross-notebook inferences while maintaining access to raw, annotated content.

The project encompasses the full lifecycle of corpus creation and utilization: from identifying use cases driven by AI modes and community priorities, defining an extensible and customizable corpus structure, to continuous data ingestion with strict quality control, FAIR (Wilkinson et al. 2016) and CARE (Carroll et al. 2020) compliance, and ethical considerations including bias mitigation and privacy.

FAIRJupyter4AI will deploy an automated, scalable reproducibility assessment pipeline that continuously tests notebook execution, environment dependencies, and resource use. Visualization tools will cater to both humans—through workflow diagrams illustrating key reproducibility steps—and machines—via structured JSON matrices. A nanopublication-based “news feed” mechanism will transparently report on reproducibility assessments, fostering community engagement.

A range of AI models, including large language models and code analyzers, will be trained and benchmarked on this richly annotated multimodal corpus. These models will support automated notebook generation and completion, fine-grained reproducibility diagnostics, intelligent error detection and debugging, and workflow optimization, pushing beyond conventional metadata querying to generation and validation aligned with best practices. Real-world benchmarking on imperfect datasets will improve AI robustness, reduce hallucinations, and enhance trustworthiness. To support extensibility and augment training resources, FAIRJupyter4AI will generate synthetic notebooks and cell sequences based on learned statistical patterns, provide APIs for community-driven data and functionality extensions, and align corpus standards with initiatives like NFDI, Wikimedia, and The Carpentries. By creating a continuously enriched, hybrid-structured corpus with integrated benchmarking, visualization, and community tools, FAIRJupyter4AI will become a foundational resource for researchers, educators, and infrastructure providers.

State-of-the-art

Computational reproducibility has emerged as a critical concern in modern scientific research, particularly with the increasing reliance on computational notebooks such as Jupyter for analysis, data processing, and result communication. Multiple studies have tackled various aspects of computational reproducibility, each highlighting unique challenges in methodology, tooling, and data accessibility.

In the life sciences, Grüning et al. (2018) emphasized domain-specific barriers to reproducibility, such as undocumented dependencies and inconsistent environments. Nüst et al. (2020) evaluated Docker's role in encapsulating computational environments, showcasing containerization as a promising solution. Trisovic et al. (2022) examined the reproducibility of R scripts in institutional repositories, illustrating that even well-archived code can suffer from execution failures due to evolving software ecosystems. Several studies focused specifically on the reproducibility of Jupyter notebooks (Rule et al. 2019, Pimentel et al. 2019, Schröder et al. 2019, Wang et al. 2020, Willis et al. 2020). Rule et al. (2018) analyzed over one million notebooks on GitHub, revealing patterns in code organization, execution order, and package use. This study proposed ten best practices for creating reproducible and well-structured Jupyter notebooks (Rule et al. 2019), underscoring the importance of both technical and human-centered design choices in computational workflows. Another large-scale study (Pimentel et al. 2021) assessed the reproducibility of 1.4 million notebooks, uncovering systemic issues such as undeclared dependencies, missing data files, and inconsistent environments that affect execution success rates.

More targeted investigations, such as that by Schröder et al. (2019), involved manual reproducibility testing of notebooks linked to biomedical publications, providing qualitative insight into the disconnect between published code and executable workflows. Similarly, Chattopadhyay et al. (2020) surveyed over 150 data scientists to identify practical challenges encountered when working with notebooks in real-world research settings—ranging from lack of documentation to hidden dependencies. The paper by Nguyen et al. (2025) challenges the prevailing notion of notebook executability by demonstrating that many non-executable notebooks are restorable through minimal intervention, such as resolving module issues or generating synthetic inputs via Large Language Models (LLMs). FAIRJupyter4AI builds on this insight by introducing a reproducibility score metric and AI-based repair tools within a large, curated corpus, enabling reuse potential across scientific domains.

Despite the breadth and scale of these efforts, existing reproducibility datasets suffer from several key limitations. First, they are typically static snapshots, lacking mechanisms for continuous updates or dynamic enrichment. Second, their structure and metadata are often insufficiently standardized or semantically rich to support advanced queries, making integration with federated sources like Wikidata or DBpedia difficult. Third, most datasets are not optimized for AI applications, failing to offer the multimodal annotations or granularity needed for training or evaluating models on reproducibility tasks.

While no prior efforts have produced comprehensive knowledge graphs specifically focused on Jupyter notebooks or their computational reproducibility, there has been substantial progress in adjacent areas (Goble et al. 2020) that informed and motivated our approach for constructing the FAIR Jupyter Knowledge Graph. Several KGs have been developed to structure information across diverse facets of computational science, scholarly communication, and software development.

One of the most relevant contributions is the Open Research Knowledge Graph (ORKG) (Auer et al. 2018), which provides a platform for representing, curating, and discovering scholarly knowledge in a structured and machine-interpretable format. ORKG has demonstrated utility in enhancing transparency and even assisting with aspects of scientific reproducibility (Hussein et al. 2023, Jaradeh et al. 2019). However, its granularity typically focuses on metadata and conceptual contributions of scholarly articles rather than the executable components embedded in computational notebooks. Other large-scale efforts like the Microsoft Academic Knowledge Graph (Färber 2019) have structured metadata on publications, authors, institutions, and research fields. While they have been leveraged to analyze open-source project metrics and contributions (Färber 2020), including gender-specific dynamics in GitHub repositories (Levitskaya et al. 2022), they lack deep integration with the actual computational processes or the granular structure of scientific code artifacts.

Some technical foundations for structuring code exist in projects like Graph4Code (Abdelaziz et al. 2021), which creates KGs to represent the structure and data flow within Python scripts. This supports use cases such as program search, bug detection, and automation, but does not directly address reproducibility or capture the dynamic execution environment and narrative context present in Jupyter notebooks. GitGraph is another prototype tool aiming to extract KG-style metadata from Git repositories, although it remains limited in scope and maturity.

Despite these advancements, a significant gap persists: no existing work has constructed a semantically rich, large-scale KG, an AI-ready corpus derived specifically from reproducibility assessments of Jupyter notebooks. Such a resource would uniquely bridge symbolic representations (e.g., entities, dependencies, provenance) with statistical views (e.g., embeddings) and potentially enrich this further with execution traces of computational workflows.

Several works have also explored datasets centered on Jupyter notebooks, primarily targeting code generation or comprehension tasks, thereby limiting their applicability to maintenance-focused scenarios (Yin et al. 2022, Agashe et al. 2019, Jin et al. 2025). Notably, there have been no prior benchmarks addressing developer edits specific to Jupyter notebooks. One recent effort fills this gap by introducing the first dataset of 48,398 notebook edits from 20,095 revisions across 792 machine learning repositories (Jin et al. 2025). This dataset captures detailed cell- and line-level modifications, shedding light on real-world maintenance patterns and highlighting that edits are typically small and highly localized.

In contrast, FAIRJupyter4AI focuses specifically on reproducibility-related repairs, creating a dedicated benchmark for error detection and correction in research notebooks. Unlike prior work on general maintenance, our benchmark will target issues such as missing dependencies, broken paths, or environment drift, supporting the development and evaluation of AI models that restore and preserve computational reproducibility in scientific workflows.

Preliminary work

This proposal builds upon our prior research (Samuel and Mietchen 2024a, Samuel and Mietchen 2024b, Samuel and König-Ries 2021), notably the large-scale study titled "*Computational Reproducibility of Jupyter Notebooks from Biomedical Publications*" (Samuel and Mietchen 2024a). In that work, we systematically analyzed over 27,000 Jupyter notebooks linked to biomedical publications via PubMed Central (Roberts 2001) and GitHub. We developed automated pipelines adapted from (Samuel and König-Ries 2021, Pimentel et al. 2019) to retrieve, clean, and structure paper and notebook metadata —including author affiliations, journal details, and dependency specifications—into a centralized SQLite database. This enabled detailed exploration of the landscape of computational reproducibility in published scientific workflows.

Our pipeline tested the reproducibility of these notebooks at scale, employing Conda-based environment reconstruction followed by notebook execution. Out of the total corpus, 10,388 notebooks were executed, with only 1,203 completing without errors, and a mere 879 producing output identical to the original. This highlighted critical issues such as undocumented dependencies, missing data files, and environment drift. We categorized and analyzed the most common failure modes—including ModuleNotFoundError, ImportError, and FileNotFoundError—and observed patterns across journals, research fields, and article types. These findings underscored the fragility and variance of reproducibility across published computational artifacts. The dataset is publicly available (Samuel and Mietchen 2023) as a 1.5GB SQLite database contained within a ZIP archive of 415.6 MB (compressed).

To provide structured access to these insights, we developed the FAIR Jupyter Knowledge Graph (Samuel and Mietchen 2024b). This semantic infrastructure converted execution and provenance data into machine-readable RDF triples (Samuel and Mietchen 2024c). The knowledge graph enables SPARQL queries to explore execution metadata, dependency structures, and reproducibility outcomes across the dataset. This forms the semantic backbone for the current proposal's integration, enrichment, and AI-readiness objectives. Our previous work also established a reusable pipeline architecture for reproducibility analysis, integrating notebook harvesting, metadata parsing, execution monitoring, and semantic enrichment. The infrastructure developed—comprising public code, notebooks, metadata repositories, and an open SPARQL endpoint—demonstrates both the technical feasibility and the community value of creating a high-quality, reproducibility-aware notebook corpus. The FAIR Jupyter service with links to the SPARQL endpoint, code and all the corresponding resources is available at https://w3id.org/fairjupyter under the GPL-3.0 license (Samuel and Mietchen 2024c). We are committed to keeping it up for five years, i.e. until April 2029. The CSV files, the YARRRML and RML mappings used for constructing the KG are available in Zenodo (Samuel and Mietchen 2024e). The resulting RDF triples are available via Samuel and Mietchen 2024d. A dedicated Jupyter notebook (Samuel and Mietchen 2024f) provides example SPARQL queries and demonstrates basic benchmarking metrics for exploring and evaluating our knowledge graph.

FAIRJupyter4AI builds directly upon this groundwork to deliver a sustainable, extensible, and impactful platform for reproducible and AI-enhanced computational science.

Objectives, concept and approach

Objectives

The primary objective of FAIRJupyter4AI is to develop a comprehensive, large-scale, and AI-ready corpus of computational notebooks that transcends existing reproducibility knowledge graphs by integrating the full multimodal content of Jupyter notebooks. This corpus will include executable code, narrative markdown, outputs (plots, tables), and associated data files, all meticulously cleansed, semantically annotated, and enriched to enable advanced AI-driven analysis and interaction. With an initial focus on Python, the corpus will be gradually expanded to include notebooks coded in other languages.

Key goals include:

- Curate and harmonize notebooks from diverse scientific domains, ensuring data quality, semantic richness, and alignment with FAIR and CARE principles.
- Construct a foundational knowledge graph interlinking notebooks and their internal components to enable structured queries and cross-notebook inference.
- Enable next-generation AI models to perform complex tasks such as automated notebook generation, fine-grained reproducibility analysis, intelligent error detection, workflow optimization, and benchmarking AI robustness.
- Provide an infrastructure that supports continuous enrichment through automated analysis and community contributions, fostering open, transparent, and reusable computational research.
- Support broad usage scenarios including reproducibility benchmarking, bias detection, metadata generation, workflow standardization, and integration with existing infrastructures like NFDI and Wikimedia.
- Facilitate visualization of reproducibility assessments for both human users (e.g., workflow diagrams) and machines (e.g., JSON/matrix formats).
- Explore synthetic data generation methods to create realistic yet anonymized notebook segments for AI training and benchmarking.
- Support educational workflows related to computational reproducibility.

Anticipated total duration of the project

The proposed project is planned for a total duration of 24 months. This time frame is appropriate to achieve the scientific objectives and complete the planned work packages given the requested resources and the use of Jupyter4NFDI infrastructure.

Work programme including proposed research methods

Methods

Fig. 1 shows the workflow for the construction of the FAIRJupyter4AI dataset.

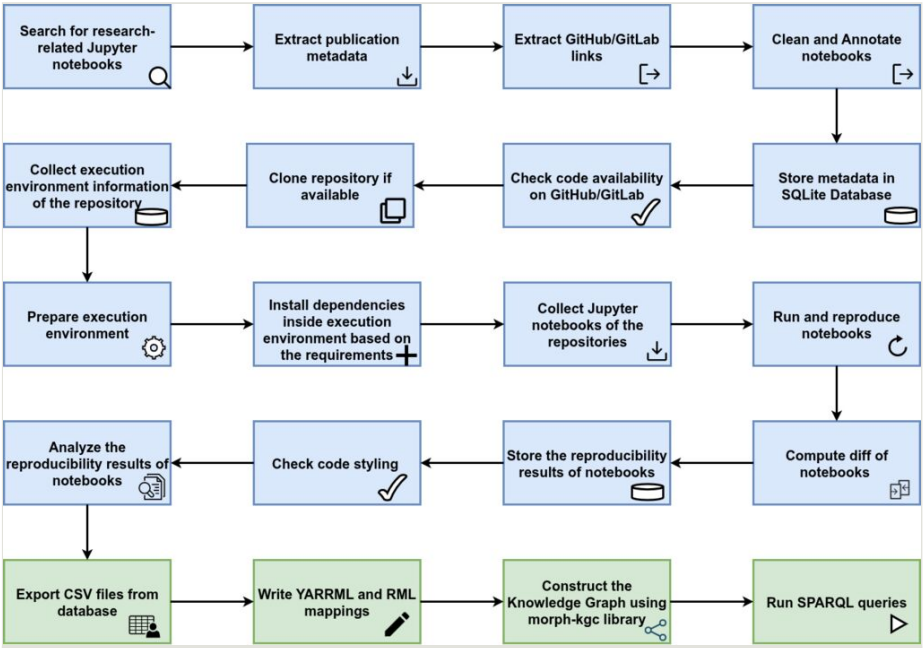


Figure 1. doi

The modular workflow for the construction of the FAIRJupyter4AI dataset will be based on that originally used for the construction of the FAIR Jupyter dataset (Samuel and Mietchen 2024a, Samuel and Mietchen 2024b), with provisions for addressing sources other than PubMed Central and non-Python programming languages. It will be deployed on Jupyter4NFDI infrastructure.

Work programme

The FAIRJupyter4AI work programme is organized into the following interconnected work packages (WPs) (Fig. 2) whose Tasks (T) are designed to reach Milestones (M) and to yield Deliverables (D) within the indicated amounts of Person Months (PM) through iterative development, rigorous validation, and community engagement:

WP1: Data Collection and Curation (Lead: FIZ)

- **T1.1: Collect 50k+ research-related Jupyter notebooks (3 PM)**. The existing FAIR Jupyter dataset (Samuel and Mietchen 2024b, Samuel and Mietchen 2024d, Samuel and Mietchen 2023) contains about 27k Jupyter notebooks, sourced from biomedical research publications indexed in PubMed Central (PMC), as of 2023.

In this task, we will expand the dataset by including - in a recurring fashion - (1) newer notebooks from PMC, (2) notebooks from other scholarly repositories (e.g. arXiv, SSRN, Zenodo), (3) notebooks originating from NFDI workflows (e.g. those deployed through Jupyter4NFDI, on an opt-in basis). We will also consider (4) notebooks from domain-general code repositories if a clear connection to research or certain corpus characteristics can be established and the available metadata is of sufficient quality. In the process, we will pay special attention to potential research and educational use cases for such a dataset or its derivatives, in particular the community needs that it may help address.

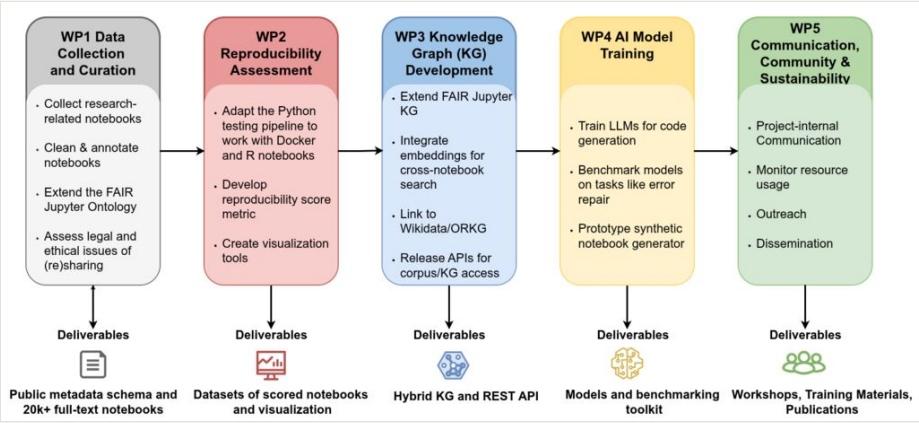


Figure 2. doi

Overview over the FAIRJupyter4AI work programme and some key deliverables.

- **T1.2: Clean notebooks & annotate with licensing/ provenance metadata (3 PM)** In this task, we will process the notebooks to add (cases 1 and 2) and remove (cases 3 and 4) some kinds of information as follows: (1) In order to build a corpus of Jupyter notebooks, the provenance and licensing terms need to be tracked (or extracted from README or CFF files and similar contexts) and compliance with licensing terms needs to be ensured. (2) In order to assess reproducibility, dependencies need to be tracked and taken into account, ideally in a recursive fashion (Nesbitt et al. 2024). (3) Notebooks might contain personally identifiable information or other sensitive information that would not be suitable for inclusion in a corpus for AI workflows. (4) In light of potential uses of the corpus for AI model training, bias is another dimension of concern, and we will document the biases we find in the data (e.g. in terms of programming and natural languages, data sources and subject matter) and explore ways to address them in ways that assist AI workflows..

- **T1.3: Extend the FAIR Jupyter Ontology to suit this broader corpus (1 PM)**. The existing FAIR Jupyter knowledge graph is using the FAIR Jupyter Ontology (Samuel and Mietchen 2024b), which needs to be adapted to be suitable for handling information regarding the additional sources and AI use cases. These

adaptations will be made with future changes of both the corpus and the ontology in mind, so as to facilitate further development.

- **T1.4: Assess legal and ethical issues of (re)sharing Jupyter notebooks (2 PM)**. In the process of collecting, analyzing and annotating the notebooks, we will come across issues that stand in the way of including the notebooks (individually or at scale, whether redacted or not) in corpora for AI. Likewise, questions around the enrichment of the notebooks with related information - e.g. about associated papers, authors, or conferences - will be explored, as well as the sharing of use case-specific subsets, or requests by third parties for inclusion or non-inclusion of specific notebooks or reproducibility attempts. In this task, such issues will be categorized, quantified and analyzed such that the analysis can inform the cleaning workflows in T1.2 in a way that is compliant with the FAIR and the CARE principles. In doing so, we will build on past activities regarding license tagging (Mietchen et al. 2013), open licensing (Hagedorn et al. 2011), and data publishing guidelines (Penev et al. 2017).

- **M1.1 (Month 6): Pipeline successfully deployed on Jupyter4NFDI infrastructure**. Deploying ReproduceMeGit - the core component of our reproducibility pipeline - to analyze the reproducibility of individual notebooks from NFDI workflows is a Deliverable (by Q3/ 2026, overseen by FIZ) in the Integration Phase of Jupyter4NFDI. We will expand on that and use the Jupyter4NFDI infrastructure to deploy our full ingestion and reproducibility pipelines (which are themselves a set of Jupyter notebooks) and further develop them there for FAIRJupyter4AI purposes.

- **M1.2 (Month 12): Licensing information available for 20k+ notebooks**. Sharing notebooks without clear licensing information essentially prevents their inclusion in AI corpora. Having 20k+ notebooks with clear licensing information thus represents an important milestone in corpus construction, particularly if the licensing terms are compatible with inclusion in such a corpus.

- **D1.1 (Month 3): Public metadata schema and ingestion pipeline**. One prerequisite to make the corpus useful is to have a clear and standards-aligned metadata schema. We will share this early on in order to invite community feedback. Alongside, we will share the pipeline that ingests notebooks and their metadata.

- **D1.2 (Month 11): Curated metadata corpus of 50k+ notebooks**. Besides licensing, other metadata (e.g. programming languages, dependencies) are essential for reproducibility-focused use cases.

- **D1.3 (Month 14): Report on legal and ethical issues related to (re)sharing Jupyter notebooks at scale**. This will summarize the findings from T1.4 and provide recommendations on what notebooks and metadata to include in our corpus, and how (technically as well as in terms of policies/ governance).

- **D1.4 (Month 24): Public full-text corpus of 20k+ research-related Jupyter notebooks**. The notebooks included in this full-text corpus will be a subset of those included in the metadata corpus of D1.2, filtered for availability of licensing information and for compatibility of the license with inclusion in a full-text corpus. Whenever possible, we will also archive these notebooks via Software Heritage.

WP2: Reproducibility Assessment (Lead: TUC)

- **T2.1: Adapt the Python testing pipeline to work with Docker (3 PM)**. The existing FAIR Jupyter pipeline did not analyze containerized notebooks. Here, we will adapt the workflow such that (1) notebooks shared with Docker containers can be handled and (2) Python notebooks assessed as fully reproducible can be containerized via Docker. In addition, running the pipeline again for notebooks we already assessed in the past allows to assess reproducibility decay. A likely starting point for these Docker efforts will be Repo2Docker (Forde et al. 2018) workflows, which are already in use at Jupyter4NFDI.

- **T2.2: Adapt the testing pipeline to work with R notebooks (3 PM)**. The existing pipeline covered notebooks in a number of languages other than Python but did not analyze their reproducibility. In this task, we will demo the use of the pipeline for non-Python notebooks by adapting it to notebooks written in R. To this end, we will review and leverage prior efforts regarding the reproducibility of R-based workflows, such as the R package flowR and the project SocEnRep .

- **T2.3: Develop a non-binary "reproducibility score" metric (1 PM)**. The existing pipeline kept track of whether required dependencies were indicated, whether they could be successfully installed, whether the notebooks ran through (and if so, whether their results matched the original ones) or whether they caused an exception, and in what cell. These kinds of information - along with relevant changes over time - could be combined into a more holistic reproducibility score, in line with the idea (Nguyen et al. 2025) that reproducibility of a Jupyter notebook could be defined as the percentage of (code) cells that execute successfully (in a Run-All mode). The holistic reproducibility score would then be included into the notebook's metadata in our corpus.

- **T2.4: Create visualization tools (1 PM)**. Here, we will build on the existing static visualizations of our initial corpus and on queries for the existing knowledge graph to assist human users of our pipeline and the corpus to visualize aspects relevant to them in a more interactive fashion, including at runtime. We will also explore visualization-like approaches for machines (e.g. structured JSON or matrix formats) and mechanisms to automatically signal information related to reproducibility attempts and results.

- **M2.1 (Month 9): ReproduceMeGit supports Docker.** ReproduceMeGit (Samuel and König-Ries 2021) is the core component of our reproducibility pipeline, so

enabling Docker support in the former is a key step in providing Docker support in the latter.

- **M2.2 (Month 18): ReproduceMeGit with R support and public dashboard.** Likewise, providing R support and dashboard functionality for our reproducibility pipeline requires adjustments to the core component, ReproduceMeGit.

- **D2.1 (Month 10): Dataset of 10k+ scored Python notebooks (e.g. execution logs, error types)**. The existing pipeline (Samuel and Mietchen 2024a) has run about 10k Python notebooks already but most of them failed in some way, so they were considered not reproducible. With the non-binary reproducibility score, perhaps some additional patterns can be observed that may be useful for corpus construction.

- **D2.2 (Month 11): Dataset of 1k+ scored non-Python notebooks (e.g. execution logs, error types)**. The existing pipeline did not assess non-Python notebooks, and their scoring might pose some additional challenges, so a smaller corpus seems to be an appropriate start here.

- **D2.3 (Month 21): Interactive dashboard (hosted demo)**. The variety of facets that potential users of our corpus might want to explore is large, so we will highlight some key facets in a hosted demo. This will include runtime characteristics not covered so far.

WP3: Knowledge Graph (KG) Development (Lead:FIZ)

- **T3.1: Extend FAIR Jupyter KG with additional information (4 PM)**. In line with the changes to the ingestion and assessment pipelines and the FAIR Jupyter Ontology as well as due to the application of the pipelines to new notebooks, the FAIR Jupyter KG would need to be adapted accordingly. For instance, the existing workflows have only partially taken into account notebook execution traces, keeping track of cell execution order and of outputs but not of kernel states during runtime, which we would now like to include. Likewise, decisions would have to be made about whether and how fixed versions of low-reproducibility notebooks would be included in the graph.

- **T3.2: Integrate embeddings (e.g., code2vec) for cross-notebook search (1 PM)** . The existing KG supports SPARQL queries, which can yield good results for searches aligned with the ontology. For similarity-based searches, SPARQL is usually not a good choice, but embeddings often are. Here, it likely makes sense to compute similarity separately for code, Markdown, outputs and errors. In addition to standard text-based approaches, code-aware embeddings like code2vec (Alon et al. 2019) look promising here. RDF and embeddings can be combined in different ways that come with different trade-offs, e.g. in terms of efficiency, complexity or scalability. We will explore some of these options and then implement the most suitable one.

- **T3.3: Link to Wikidata/ORKG (publications, libraries) (1 PM)**. The FAIR Jupyter KG has some overlap with other KGs. For instance, Wikidata and/ or ORKG might have information about the paper associated with a notebook, and federated queries based on the paper's DOI can in principle find out whether that is the case, but this approach is not very user-friendly. The same goes for other kinds of information, e.g. the libraries or datasets used in a notebook might also have Wikidata entries. In this task, we thus plan to enrich entries in the FAIR Jupyter KG with links to corresponding entries in other KGs, particularly Wikidata and ORKG.

- **T3.4: Release APIs for corpus/KG access (1 PM)**. The existing FAIR Jupyter KG can already be accessed programmatically but the planned changes to the KG imply changes to programmatic access and its documentation. Likewise, the corpus itself should be accessible programmatically, which this task will ensure.

- **M3.1 (Month 12): KG v1.0 (metadata only)**. The FAIR Jupyter KG has all the features deemed necessary for metadata corpus construction, along with a useful amount of content and associated documentation.

- **M3.2 (Month 21): Full multimodal KG release**. At this point, the full text of suitably licensed notebooks is also represented in the KG, at least by way of embeddings.

- **D3.1 (Month 16): Hybrid KG (RDF + embeddings) with SPARQL endpoint**. We provide at least one way in which RDF and embeddings can be used together to search by both semantic and similarity criteria.

- **D3.2: (Month 19) Public REST API**. Both the KG and the corpus are programmatically accessible via RESTful APIs that cater to use cases identified through community engagement.

WP4: AI Model Training (Lead:TUC)

- **T4.1: Train LLMs (e.g., CodeLlama) on corpus for code generation (5 PM)**. Once the corpus reaches certain thresholds in terms of size, quality and feature richness, it can be used for experimentation with AI models. While general-purpose models like Llama might be useful for the Markdown part of notebooks (e.g. to improve textual documentation), more specialized ones like CodeLlama (Roziere et al. 2023) are expected to perform better on code-related tasks such as code generation, code simplification, debugging, visualization and documentation. We will explore pretrained models, transfer learning, knowledge distillation and related approaches to identify potential matches between what our corpus might enable and what its potential users might want to do. On the way, we will also assess different ways to use LLMs via Jupyter (e.g. direct execution, interactive UI, container, cloud) for specific research-related use cases.

- **T4.2: Benchmark models on key tasks like error repair (5 PM)**. Some tasks that we expect to be popular amongst users of our corpus include (1) error repair, debugging or other improvements to a notebook (or its environment) to increase

its reproducibility score, (2) cross-language refactoring of code, (3) code suggestions or code completion, (4) assessment of the correspondence between notebook content and the relevant sections of associated publications. For a set of tasks like these and for a range of models, we will set up benchmarking workflows. These workflows will include tests on robustness against hallucinations (e.g. as per Xu et al. 2024).

- **T4.3: Prototype synthetic notebook generator (privacy-safe) (3 PM)**. When notebooks and/ or associated data are not readily shared (for whatever reason), synthetic notebooks and/ or data might provide a useful avenue to explore some key aspects nonetheless. For instance, if the methods section of a paper claims to have processed certain types of data in a certain way to yield certain results, then that text can serve as the basis for prompts to create notebooks (perhaps along with suitable data files) that allow to explore computational and data aspects of that methodology. In this task, we will explore several scenarios for synthetic notebooks, and how our corpus and suitable models can assist with them. Suitably designed synthetic notebooks can also be used to augment training data and reduce biases.

- **M4.1 (Month 15): Initial model for code completion**. Based on the model exploration activities, we will share a model that performs well on common code completion tasks in research-related Jupyter contexts, and invite community feedback.

- **M4.2 (Month 24): Full benchmarking suite**. For a range of tasks and benchmarking workflows, we provide results and enable others to submit theirs.

- **D4.1 (Month 17): Hugging Face repository with five models**. As Hugging Face is a common, popular and robust platform for sharing AI models and corpora, we will use it as one way to reach out to the community of potential FAIRJupyter4AI users.

- **D4.2 (Month 22): Benchmarking toolkit (leaderboard for reproducibility tasks)**. Here, we will share the setup we are using for running our reproducibility benchmarks. The setup is expected to be readily adaptable to other tasks and other data corpora (e.g. subsets of our corpus).

WP5: Communication, Community & Sustainability (Lead: FIZ)

- **T5.1: Project-internal communication (4 PM)**. The project involves complex technical work in a rapidly changing landscape, to be performed by individuals yet to be hired into two institutions, under the supervision of two PIs experienced in this space. In such a setting, good and regular internal communication is paramount. To this end, we will share workspaces, progress and problems on an ongoing basis whenever possible.

- **T5.2: Monitor resource usage (e.g. CPU, memory, cores) (2 PM)**. This task has several facets: (1) execution traces also depend on the underlying hardware, and

the use of resources like memory, CPUs and cores might have direct implications for reproducibility, which we want to track and include in the reproducibility assessment and its representation in the ontology, KG and corpus, (2) for certain types of resources (e.g. TPUs or large RAM), high reproducibility scores might only be achievable on specialized infrastructure, the merits of which would need to be explored via community engagement, especially since (3) the environmental footprints of AI models (both during training and inference) as well as of large datasets or large-scale reproducibility assessments are considerable. Furthermore, (4) the scalability of our own workflows relates to a balance between resource use and resource availability.

- **T5.3: Outreach (incl. 3 workshops, nanopublications) (2 PM)**. Our ingestion and reproducibility pipelines have been developed in the open from the beginning with lots of community feedback through channels like GitHub, JupyterCon or review requests. We want to continue working in this open fashion - e.g. publishing this proposal is another step in this direction - and expect that this approach will allow us to leverage community insights whenever we are not sure how to proceed. In addition to that, we plan to run a series of workshops for the project that are inspired by the ReproHack, Carpentry and NFDI communities and aimed at (1) assisting in the design of the corpus and (2) promoting its use by tool developers, educators and other stakeholders. Last but not least, we intend to experiment with nanopublications as a way to communicate the results of reproducibility assessments and generate additional interest in our workflows.

- **T5.4: Dissemination (e.g. publish best practice guide) (2 PM)**. With our methodology, code and data being open already, what remains to be communicated are essentially the insights we get in the process. We have done this for the initial pipeline construction (Samuel and Mietchen 2024a) and the construction of the KG (Samuel and Mietchen 2024b), which would form a good basis for a best practice guide to expand the pipeline and the KG to build a corpus for AI applications.

- **M5.1 (Month 8): First workshop**. At that point, we expect to have running infrastructure, a good amount of data and some useful ideas about what could realistically be done with the corpus using contemporary models. This seems like a good set of circumstances to engage with the community of potential users in a workshop. The feedback we expect to get there can help us finetune some key aspects of the project like features of the corpus or of the associated ingestion, assessment or benchmarking workflows.

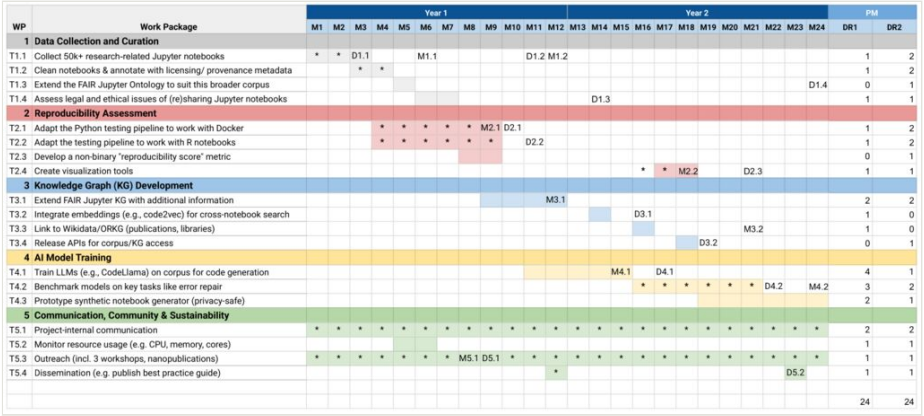
- **D5.1 (Month 9): Workshop materials are published**. Most of these materials will actually be made public beforehand but we want to incorporate some community feedback into these materials, so as to have a better basis for the other workshops.

- **D5.2 (Month 23): Preprint on the FAIRJupyter4AI corpus**. A narrative summary of our approach, with pointers to data, code and other associated materials, as well as with highlights of what we learned, how others are using the corpus or our workflows, and what the next steps might look like.

This structured approach ensures FAIRJupyter4AI delivers a continuously evolving, high-quality, and ethically sound resource that advances AI-driven reproducibility and research transparency at scale.

Timeline

Fig. 3 shows the FAIRJupyter4AI project plan.



WP	Work Package	Year 1												Year 2												PM	
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24	DR1	DR2
1	**Data Collection and Curation**																										
T1.1	Collect 50k+ research-related Jupyter notebooks	*	*	D1.1			M1.1					D1.2	M1.2													1	2
T1.2	Clean notebooks & annotate with licensing/ provenance metadata			*	*																					1	2
T1.3	Extend the FAIR Jupyter Ontology to suit this broader corpus																								D1.4	0	1
T1.4	Assess legal and ethical issues of (re)sharing Jupyter notebooks														D1.3											1	1
2	**Reproducibility Assessment**																										
T2.1	Adapt the Python testing pipeline to work with Docker				*	*	*	*	*	M2.1	D2.1															1	2
T2.2	Adapt the testing pipeline to work with R notebooks				*	*	*	*	*	*		D2.2														1	2
T2.3	Develop a non-binary "reproducibility score" metric																									0	1
T2.4	Create visualization tools																*	*	M2.2			D2.3				1	1
3	**Knowledge Graph (KG) Development**																										
T3.1	Extend FAIR Jupyter KG with additional information												M3.1													2	2
T3.2	Integrate embeddings (e.g., code2vec) for cross-notebook search																D3.1									1	0
T3.3	Link to Wikidata/ORKG (publications, libraries)																					M3.2				1	0
T3.4	Release APIs for corpus/KG access																			D3.2						0	1
4	**AI Model Training**																										
T4.1	Train LLMs (e.g., CodeLlama) on corpus for code generation														M4.1			D4.1								4	1
T4.2	Benchmark models on key tasks like error repair																*	*	*	*	*	D4.2			M4.2	3	2
T4.3	Prototype synthetic notebook generator (privacy-safe)																									2	1
5	**Communication, Community & Sustainability**																										
T5.1	Project-internal communication	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	2	2
T5.2	Monitor resource usage (e.g. CPU, memory, cores)																									1	1
T5.3	Outreach (incl. 3 workshops, nanopublications)	*	*	*	*	*	*	*	M5.1	D5.1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1	1
T5.4	Dissemination (e.g. publish best practice guide)												*											D5.2		1	1
																										24	24

Figure 3. doi

FAIRJupyter4AI Project Plan: Work Packages, Tasks, Deliverables, Milestones, and Timeline. This Gantt chart provides a 24-month project schedule, detailing the progression of work packages and individual tasks. The person-month allocation for Doctoral Researcher 1 (DR1) and Doctoral Researcher 2 (DR2) is shown on the right. Tasks marked with an asterisk (*) highlight involvement and contributions from student assistants. Key deliverables (Dx.y) and milestones (Mx.y) are indicated at their respective completion points.

Risk Analysis

The FAIRJupyter4AI project faces several risks across technical, legal, community, and resource domains. Technical challenges include scalability issues in the reproducibility pipeline and its integration into the knowledge graph, mitigated through modular design and adherence to existing standards. AI model underperformance is addressed by leveraging pretrained models and hybrid approaches. Legal risks, such as licensing conflicts and data bias, are managed through automated filtering and stratified sampling, while privacy concerns are minimized with scrubbing tools. Community engagement risks, including low adoption or tool complexity, are countered with partnerships and user-friendly interfaces. Resource risks like personnel turnover or infrastructure delays are mitigated by workflow documentation and written commitments from providers. A

contingency budget is reserved for potential problem areas like emergency cloud costs, legal reviews or community bounty initiatives, ensuring adaptability.

Supplementary information on the project context

General ethical aspects

No facet of the proposed research is expected to trigger a requirement for formal ethical review. Nonetheless, we will carefully consider relevant ethical aspects throughout the project. Our dataset, FAIRJupyter4AI, is based on publicly available content from PubMed Central and other research repositories, which may contain personally identifiable information or other sensitive information, as well as different types of biases that would impact AI use. In WP1, we will evaluate these issues (T1.4) and address them in our processing pipeline (T1.2). We also plan to include information on the environmental footprint of our analyses to promote transparency and accountability. These measures are intended to ensure that our study of reproducibility in AI respects ethical principles and contributes constructively to open science.

Generative AI tools (e.g., Deepseek and Gemini) were used for language editing and refinement of certain sections of this proposal. The scientific content remains the responsibility of the authors.

Considerations on aspects of ecological sustainability in the planning and implementation of the project

We recognize the environmental implications of large-scale computational experiments and have therefore incorporated metadata related to the environmental footprint of notebook execution—such as estimated energy consumption and hardware utilization—to support the documentation of resource use as a step towards more sustainable digital research practices.

Measures to meet funding requirements and handle project results

We will implement comprehensive measures for the management, storage, and dissemination of all results and data generated during the project. All research data, including the FAIRJupyter4AI corpus, metadata, and analysis outputs, will be stored using institutional repositories that support long-term preservation and persistent identifiers (e.g., DOIs). Workflows developed in this project will not only be used to generate the FAIRJupyter4AI corpus but they will be incorporated into the Jupyter4NFDI service portfolio and thereby be accessible to NFDI consortia and through NFDI to the wider research community. Where legally and ethically permissible, we plan to publish our metadata corpus under CC0 and our full-text corpus such that only openly licensed notebooks are included, to ensure accessibility and reusability by the scientific community. In addition, we will make all code and workflows available via GitHub and

archive them via Zenodo. Documentation and metadata will follow the FAIR principles (Wilkinson et al. 2016) to promote transparency and reproducibility.

Formal assurances

Publications resulting from the project and any relevant documentation will be available via open access, making them widely accessible for use by third parties. The source code for the software developed under the project will be documented in accordance with established standards, licensed with an open-source license, and made available for use by third parties free of charge.

People/collaborations/funding

Employment status information

Mietchen, Daniel - Postdoctoral Researcher (permanent)

Samuel, Sheeba - Postdoctoral Researcher (permanent)

Composition of the project group

Fodstad, Franziska - Secretary at the Department of Mathematics, FIZ Karlsruhe: She will provide administrative support for this project.

Herklotz, Dajana - Secretary at the Institute of Computer Science, Chemnitz University of Technology: She will provide administrative support for this project.

Other submissions

This project is currently not funded and no funding proposal has been previously submitted for it.

Financial contributions

TUC and FIZ Karlsruhe will provide essential workplace equipment and infrastructure, including IT support, computing facilities, and collaborative tools. In addition, Jupyter4NFDI is providing infrastructure for hosting and managing the FAIRJupyter4AI workflows, including storage, computing, and long-term availability mechanisms. Dr. Mietchen and Dr. Samuel, funded by FIZ and TUC, respectively, will each contribute approximately five hours per week to the project, providing expert supervision to the doctoral researchers and support staff, and ensuring close coordination across partners.

Requested modules/funds

Requested modules/funds

Requested funding for Staff:

Doctoral Researcher (Wissenschaftliche Mitarbeiter; Table 1): To successfully carry out the proposed research, two full-time (100%) doctoral researcher positions (DR1 and DR2) are essential. The work requires advanced skills in computer science and computational linguistics, particularly in areas such as artificial intelligence, semantic web, and software engineering. Offering less than a full position would not be competitive given the strong demand for such graduates in both academia and industry. We are confident the position can be filled by two qualified MSc graduates from a high-quality computer science program.

Table 1.
Requested funding for Staff.

Name	Staff	Quantity	Months	Sum (€)	Applicant
N.N	Doctoral Researcher or Comparable (100%)	1	24	163,200	Mietchen, Daniel
N.N	Doctoral Researcher or Comparable (100%)	1	24	163,200	Samuel, Sheeba
Total				**326400**	

Support Staff (Studentische/Wissenschaftliche Hilfskräfte; Table 2): Support staff (student assistants) are required to assist with essential tasks such as data collection, preprocessing, literature review, documentation and software maintenance. These tasks are critical to the efficiency of the project but do not require the full qualification level of the doctoral researcher. Employing qualified student assistants will ensure smooth project operations and allow the core research staff to focus on higher-level scientific work.

Table 2.
Requested Fund for Support Staff.

Name	Staff	Quantity	Hours per Month	Sum (€)	Applicant
N.N	Support Staff	1	80	20000	Mietchen, Daniel
N.N	Support Staff	1	80	20000	Samuel, Sheeba
Total				**40000**	

Job Description of staff (requested funds):

N.N. DR1 (Doctoral Researcher 1, 100%) will primarily work on WP3: Knowledge Graph (KG) Development and WP4: AI Model Training. In WP1, they will be involved in collecting and cleaning notebooks. For WP3, their main focus will be on extending the

FAIR Jupyter KG with additional information (T3.1), integrating embeddings for cross-notebook search (T3.2), and linking to external knowledge bases like Wikidata/ORKG (T3.3). This researcher will also contribute to WP2's reproducibility pipeline (T2.1, T2.2, & T2.4) and project-internal communication (T5.1), and outreach (T5.3). For WP4, they will lead the training of LLMs (T4.1), benchmarking models on error repair (T4.2), and prototyping the synthetic notebook generator (T4.3).

N.N. DR2 (Doctoral Researcher 2, 100%) will primarily focus on WP1: Data Collection & Curation and WP2: Reproducibility Assessment. In WP1, they will be heavily involved in collecting and cleaning notebooks, and critically, extending the FAIR Jupyter Ontology (T1.3). In WP2, their core responsibilities include adapting the Python and R testing pipelines to work with Docker (T2.1) and R notebooks (T2.2), and developing the non-binary "reproducibility score" metric (T2.3).

Support staff will primarily focus on providing crucial assistance across all work packages, particularly in data-intensive and foundational tasks. Their main contributions will be in WP1: Data Collection & Curation, specifically in the collection (T1.1), cleaning, and annotation of notebooks (T1.2). They will also support WP2: Reproducibility Assessment by assisting with the adaptation of the Python and R testing pipelines (T2.1, T2.2) and with WP4: AI Model Training in tasks like LLM training (T4.1) and benchmarking (T4.2). They will be actively involved in WP5: Communication, Community & Sustainability through project-internal communication (T5.1) and outreach activities including workshops (T5.3).

Requested funding for direct costs:

Travel Costs:

We request travel funds to support dissemination, outreach, and coordination activities that are essential to the success of the FAIRJupyter4AI project (Table 3). These activities will primarily involve two doctoral researchers, who will each attend up to two international conferences per year. Target venues include leading conferences such as the International Semantic Web Conference (ISWC), the Extended Semantic Web Conference (ESWC), and major AI and benchmarking venues (e.g., NeurIPS datasets track, AAAI, or SemEval), where we plan to publish our datasets and reproducibility benchmarks. In addition to academic dissemination, travel funds are necessary to support active participation in the National Research Data Infrastructure (NFDI) ecosystem. This includes attending NFDI conferences and community workshops to present our methods, gather feedback, and promote reuse of the FAIRJupyter4AI corpus. We will engage with relevant NFDI consortia as well as the Basic Service Jupyter4NFDI, so as to align our development with their infrastructure and deployment goals. Regular visits and collaborative meetings with Jupyter4NFDI partners (national) are planned to coordinate efforts related to dataset integration, service deployment, and API compatibility. We will also coordinate our workshops with those organized by Jupyter4NFDI to ensure complementarity.

Table 3.
Travel Costs.

Funding for	Sum (€)	Applicant
Travel	10,000	Mietchen, Daniel
	7000	Samuel, Sheeba

Table 4 shows the funding requested for project-related publications.

Table 4.
Project-related Publications.

Funding for	Sum (€)	Applicant
Publications	1500	Mietchen, Daniel
	1500	Samuel, Sheeba

Other Modules:

Project Specific Workshops

We request a total of 10,000 EUR to support the organization of three project-specific workshops over the two-year funding period (Table 5). These workshops are essential for community engagement, capacity building, and sustainable adoption of the outcomes of FAIRJupyter4AI. Workshops will target researchers, developers, and members of the NFDI and open science communities, and may be co-located with relevant events (e.g., NFDI consortium meetings) or at applicant institutions.

Table 5.
Project Specific Workshops.

Funding for	Sum (€)	Applicant
Project Specific Workshops	5000	Mietchen, Daniel
Project Specific Workshops	5000	Samuel, Sheeba

Requested funding for instrumentation

None

Total funding requested

Table 6 shows the total funds requested.

Table 6.

Total funding requested.

Applicant	Requested funding in €
Daniel Mietchen (FIZ)	199700
Sheeba Samuel (TUC)	196700
Total: 396400	

Grant title

LIS Funding Programme or Call: Data Corpora for Artificial Intelligence (DFG 2025)

Hosting institution

FIZ Karlsruhe Leibniz Institute for Information Infrastructure (FIZ Karlsruhe)

Chemnitz University of Technology, Chemnitz

Author contributions

Daniel Mietchen and Sheeba Samuel contributed equally to this proposal.

Conflicts of interest

The authors have declared that no competing interests exist.
Disclaimer: This article is (co-)authored by any of the Editors-in-Chief, Managing Editors or their deputies in this journal.

References

- Abdelaziz I, Dolby J, McCusker J, Srinivas K (2021) A Toolkit for Generating Code Knowledge Graphs. Proceedings of the 11th Knowledge Capture Conference137-144. https://doi.org/10.1145/3460210.3493578
- Agashe R, Iyer S, Zettlemoyer L (2019) JulCe: A Large Scale Distantly Supervised Dataset for Open Domain Context-based Code Generation. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)5435-5445. https://doi.org/10.18653/v1/d19-1546
- Alon U, Zilberstein M, Levy O, Yahav E (2019) code2vec: learning distributed representations of code. Proceedings of the ACM on Programming Languages 3: 1-29. https://doi.org/10.1145/3290353

- Auer S, Kovtun V, Prinz M, Kasprzik A, Stocker M, Vidal ME (2018) Towards a Knowledge Graph for Science. Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics1-6. https://doi.org/10.1145/3227609.3227689
- Carroll SR, Garba I, Figueroa-Rodríguez O, Holbrook J, Lovett R, Materechera S, Parsons M, Raseroka K, Rodriguez-Lonebear D, Rowe R, Sara R, Walker J, Anderson J, Hudson M (2020) The CARE Principles for Indigenous Data Governance. Data Science Journal 19 https://doi.org/10.5334/dsj-2020-043
- Chattopadhyay S, Prasad I, Henley A, Sarma A, Barik T (2020) What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems1-12. https://doi.org/10.1145/3313831.3376729
- DFG (2025) Data Corpora for Artificial Intelligence. Information for Researchers 28 (3 April 2025). [In English]. URL: https://www.dfg.de/en/news/news-topics/announcements-proposals/2025/ifr-25-28
- Färber M (2019) The Microsoft Academic Knowledge Graph: A Linked Data Source with 8 Billion Triples of Scholarly Data. Lecture Notes in Computer Science113-129. https://doi.org/10.1007/978-3-030-30796-7_8
- Färber M (2020) Analyzing the GitHub Repositories of Research Papers. Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020491-492. https://doi.org/10.1145/3383583.3398578
- Forde JZ, Head TD, Holdgraf C, Panda Y, Nalvarete G, Ragan-Kelley B, Sundell E (2018) Reproducible Research Environments with Repo2Docker.URL: https://api.semanticscholar.org/CorpusID:49255709
- Goble C, Cohen-Boulakia S, Soiland-Reyes S, Garijo D, Gil Y, Crusoe M, Peters K, Schober D (2020) FAIR Computational Workflows. Data Intelligence 2: 108-121. https://doi.org/10.1162/dint_a_00033
- Grüning B, Chilton J, Köster J, Dale R, Soranzo N, van den Beek M, Goecks J, Backofen R, Nekrutenko A, Taylor J (2018) Practical Computational Reproducibility in the Life Sciences. Cell Systems 6 (6): 631-635. https://doi.org/10.1016/j.cels.2018.03.014
- Hagedorn G, Mietchen D, Morris R, Agosti D, Penev L, Berendsohn W, Hobern D (2011) Creative Commons licenses and the non-commercial condition: Implications for the re-use of biodiversity information. ZooKeys 150: 127-149. https://doi.org/10.3897/zookeys.150.2189
- Hussein H, Farfar KE, Oelen A, Karras O, Auer S (2023) Increasing Reproducibility in Science by Interlinking Semantic Artifact Descriptions in a Knowledge Graph. Lecture Notes in Computer Science220-229. https://doi.org/10.1007/978-981-99-8088-8_19
- Jaradeh MY, Oelen A, Farfar KE, Prinz M, D'Souza J, Kismihók G, Stocker M, Auer S (2019) Open Research Knowledge Graph. Proceedings of the 10th International Conference on Knowledge Capture243-246. https://doi.org/10.1145/3360901.3364435
- Jin B, Wang J, Nie P (2025) Suggesting Code Edits in Interactive Machine Learning Notebooks Using Large Language Models. https://doi.org/10.48550/arXiv.2501.09745
- Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, Kelley K, Hamrick J, Grout J, Corlay S, Ivanov P, Avila D, Abdalla S, Willing C, Team JD (2016) Jupyter Notebooks - a publishing format for reproducible computational workflows. Positioning and Power in Academic Publishing: Players, Agents and Agendas https://doi.org/10.3233/978-1-61499-649-1-87

- Levitskaya E, Korkmaz G, Mietchen D, Rasberry L (2022) Analysis of linked Github and Wikidata. https://doi.org/10.5281/zenodo.7443339.
- Mietchen D, Maloney C, Moskopp ND (2013) Inconsistent XML as a barrier to reuse of Open Access Content. Balisage Series on Markup Technologies 12 https://doi.org/10.4242/balisagevol12.mietchen01
- Nesbitt A, Veytsman B, Mietchen D, Brown EM, Howison J, Pimentel JF, Hébert-Dufresne L, Druskat S (2024) Biomedical open source software: Crucial packages and hidden heroes. https://doi.org/10.48550/arXiv.2404.06672
- Nguyen T, Gill W, Gulzar MA (2025) Are the Majority of Public Computational Notebooks Pathologically Non-Executable? 2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)396-407. https://doi.org/10.1109/msr66628.2025.00070
- Nüst D, Sochat V, Marwick B, Eglen S, Head T, Hirst T, Evans B (2020) Ten simple rules for writing Dockerfiles for reproducible data science. PLOS Computational Biology 16 (11). https://doi.org/10.1371/journal.pcbi.1008316
- Penev L, Mietchen D, Chavan V, Hagedorn G, Smith V, Shotton D, Ó Tuama É, Senderov V, Georgiev T, Stoev P, Groom Q, Remsen D, Edmunds S (2017) Strategies and guidelines for scholarly publishing of biodiversity data. Research Ideas and Outcomes 3 https://doi.org/10.3897/rio.3.e12431
- Pimentel JF, Murta L, Braganholo V, Freire J (2019) A Large-Scale Study About Quality and Reproducibility of Jupyter Notebooks. 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR) https://doi.org/10.1109/msr.2019.00077
- Pimentel JF, Murta L, Braganholo V, Freire J (2021) Understanding and improving the quality and reproducibility of Jupyter notebooks. Empirical Software Engineering 26 (4). https://doi.org/10.1007/s10664-021-09961-9
- Roberts R (2001) PubMed Central: The GenBank of the published literature. Proceedings of the National Academy of Sciences 98 (2): 381-382. https://doi.org/10.1073/pnas.98.2.381
- Roziere B(ClOfmfcapad1, Gehring J, Gloeckle F, Sootla S, al. e (2023) Code llama: Open foundation models for code. https://doi.org/10.48550/arXiv.2308.12950
- Rule A, Tabard A, Hollan J (2018) Exploration and Explanation in Computational Notebooks. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems1-12. https://doi.org/10.1145/3173574.3173606
- Rule A, Birmingham A, Zuniga C, Altintas I, Huang S, Knight R, Moshiri N, Nguyen M, Rosenthal SB, Pérez F, Rose P (2019) Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. PLOS Computational Biology 15 (7). https://doi.org/10.1371/journal.pcbi.1007007
- Samuel S, König-Ries B (2021) ReproduceMeGit: A Visualization Tool for Analyzing Reproducibility of Jupyter Notebooks. Lecture Notes in Computer Science201-206. https://doi.org/10.1007/978-3-030-80960-7_12
- Samuel S, Mietchen D (2023) Dataset of a Study of Computational reproducibility of Jupyter notebooks from biomedical publications. https://doi.org/10.5281/zenodo.8226725
- Samuel S, Mietchen D (2024a) Computational reproducibility of Jupyter notebooks from biomedical publications. GigaScience 13 https://doi.org/10.1093/gigascience/giad113
- Samuel S, Mietchen D (2024b) FAIR Jupyter: A Knowledge Graph Approach to Semantic Sharing and Granular Exploration of a Computational Notebook Reproducibility Dataset. Transactions on Graph Data and Knowledge 2 (2). https://doi.org/10.4230/TGDK.2.2.4.

- Samuel S, Mietchen D (2024c) FAIR Jupyter. Service,. https://doi.org/10.4230/artifacts.22527
- Samuel S, Mietchen D (2024d) FAIR Jupyter Knowledge Graph [Data set]. https://doi.org/10.5281/zenodo.13845701.
- Samuel S, Mietchen D (2024e) FAIR Jupyter Knowledge Graph: v1.0. Software, version 1.0. https://doi.org/10.5281/zenodo.14197755
- Samuel S, Mietchen D (2024f) FAIR Jupyter Knowledge Graph: SPARQL Queries and Performance Evaluation and Benchmark. https://doi.org/10.5281/zenodo.13847627
- Schröder M, Krüger F, Spors S (2019) Reproducible research is more than publishing research artefacts: A systematic analysis of jupyter notebooks from research articles. https://doi.org/10.48550/arXiv.1905.00092.
- Trisovic A, Lau M, Pasquier T, Crosas M (2022) A large-scale study on research code quality and execution. Scientific Data 9 (1). https://doi.org/10.1038/s41597-022-01143-6
- Wang J, Kuo T, Li L, Zeller A (2020) Restoring reproducibility of Jupyter notebooks. Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings288-289. https://doi.org/10.1145/3377812.3390803
- Wilkinson M, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten J, da Silva Santos LB, Bourne P, Bouwman J, Brookes A, Clark T, Crosas M, Dillo I, Dumon O, Edmunds S, Evelo C, Finkers R, Gonzalez-Beltran A, Gray AG, Groth P, Goble C, Grethe J, Heringa J, 't Hoen PC, Hooft R, Kuhn T, Kok R, Kok J, Lusher S, Martone M, Mons A, Packer A, Persson B, Rocca-Serra P, Roos M, van Schaik R, Sansone S, Schultes E, Sengstag T, Slater T, Strawn G, Swertz M, Thompson M, van der Lei J, van Mulligen E, Velterop J, Waagmeester A, Wittenburg P, Wolstencroft K, Zhao J, Mons B (2016) The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data 3 (1). https://doi.org/10.1038/sdata.2016.18
- Willis A, Charlton P, Hirst T (2020) Developing Students' Written Communication Skills with Jupyter Notebooks. Proceedings of the 51st ACM Technical Symposium on Computer Science Education1089-1095. https://doi.org/10.1145/3328778.3366927
- Xu Z, Jain S, Kankanhalli M (2024) Hallucination is inevitable: An innate limitation of large language models. https://doi.org/10.48550/arXiv.2401.11817.
- Yin P, Li WD, Xiao K, Rao A, Wen Y, Shi K (2022) Natural language to code generation in interactive data science notebooks. https://doi.org/10.48550/arXiv.2212.09248