PERIODIC TECHNICAL REPORT III (MONTHS 37-48), PART B

LUCA DE FEO, MICHAEL KOHLHASE, MIN RAGAN-KELLEY, CLÉMENT PERNET, VIVIANE PONS, AND NICOLAS M. THIÉRY

ABSTRACT. This is OpenDreamKit's Technical Report for Reporting Period 3. As such, it provides an overview of the project achievements by objectives, work packages, and tasks from September 2018 to August 2019 and their impact. It also explains how reviewers' recommendations were followed up and discusses risk and quality management and deviations from the original plan.

[1] EdN:1

CONTENTS

[1] EDNOTE: Fix capitalization of section titles

1. Explanation of the work carried out by the beneficiaries and Overview of the progress

In this section, we give a general overview of the progress of the project during the third reporting period, ranging from September 2018 to August 2019. For new readers, we start by recalling some context of OpenDreamKit's approach that is important to understand and evaluate the progress; except for the last paragraphs, this piece of text is unchanged from the preview reporting periods. Then we provide a brief overview of the work carried out by objective of the project; finally we detail the progress work package by work package.

Some context: OpenDreamKit's approach

OpenDreamKit's approach to delivering a Virtual Research Environment (VRE) for mathematics is not to build a monolithic one-size-fits-all VRE, but rather a toolkit from which it is easy to set up VRE's that are customised to specific needs by combining the appropriate components (collaborative workspaces, user interfaces, computational software, databases, ...) on top of available physical resources (from personal laptops to cloud infrastructure). This approach — chosen by design — allows users to flexibly put together lean computational environments and tools for particular research challenges. These tools provide the required functionality but due to the component based approach carry no unnecessary bloat that would reduce effectiveness in terms of installation process, size, computation time, and reproducibility.

Most of the components preexist as an ecosystem of open source software, developed by well established communities of developers. For example, for interactive computing and data analysis, OpenDreamKit promotes Jupyter, a web-based general purpose flexible notebook interface[1] that targets all areas of science. A number of Virtual Research Environment already exist, e.g. powered by CoCalc (formerly SageMathCloud) or JupyterHub.

Hence most of the work in OpenDreamKit is to foster this ecosystem, improving the components themselves and their composability. The technical work is distributed over the work packages:

- *Component Architecture* (**WP3**): ease of deployment: modularity, packaging, portability, distribution, for individual components and combinations thereof. sustainability of the ecosystem: improving the development workflows.
- *User Interfaces* (**WP4**): enable Jupyter as uniform notebook interface, and further improve it; foster the collaboration between CoCalc and JupyterHub; generally speaking investigate collaborative, reproducible, and active documents.
- *Performance* (**WP5**): make the most of available hardware (multi-core, HPC, cloud), for individual computational components and combinations thereof.
- *Data/Knowledge/Software* (**WP6**): enable rich and robust interaction between computational components, data bases, knowledge bases, and users through explicit common semantic spaces, a language to express them, and tools to leverage them.

These technical work packages are supported by[2]

- *Community Building and Dissemination* (**WP2**): developer and training workshops, conferences, teaching material with focus on making the created value accessible to a wide, varied and growing user community.

[1] a notebook is a document that contains live code, equations, visualizations and explanatory text

[2] The project originally had another work package on *Studies of Social Aspects* (**WP7**); following the formal review for reporting period 1, it was decided in agreement with the reviewers and advisory board to shut down the work package after reporting period 1, moving some of its tasks to other work packages, and redirecting the man power for the others to more central tasks.

As a result of OpenDreamKit's approach, the work programme for OpenDreamKit consists of a large array of loosely coupled tasks, each being useful in its own right, and none being absolutely critical.

All three reporting periods confirmed that this is a strong feature of OpenDreamKit's approach. Indeed, as analysed in the proposal, this kind of project is subject to the following risks:

(1) Recruitment of qualified personnel;
(2) Different groups not forming an effective team;
(3) Implementing infrastructure that does not match the needs of end-users;
(4) Lack of predictability for tasks that are pursued jointly with the community;
(5) Reliance on external software components.

Together with ambitious software challenges, this made the accurate prediction of workload and precise timeline of work packages difficult, especially over a period of four years in a field of rapidly evolving technologies.

And indeed the project actually faced each of the risks above, especially 1, 4, and 5. The toughest situation we encountered was the volatility of the personnel at Sheffield, where the PIs and hired personnel all left for industry at various stage of the project, some after an intermediate move to Leeds. However, thanks to the flexibility enabled by the loose coupling, those risks could be mitigated by adapting the tasks schedule and human resources allocation, with little influence on the general aims and objectives.

OpenDreamKit's approach has one downside: it impedes formal evaluation, as much for us to assess the adequacy and impact of our tools, as for our reviewers to assess the depth and value of our contribution. Indeed, we do not have a main well-defined product and we capture a very diverse range of end-users and use-cases; hence quantitative methods of evaluation of the adequacy for end users, like satisfaction surveys, are rather elusive. In addition, when tools are jointly developed with the community, how should one attribute the merit of their success to the project or to the community?

In practice, this certainly added complexity to our reporting efforts and to our reviewers efforts. It is our *belief* however that this did not impact the work itself. Indeed, co-design is intrinsic to the by-users for-users development model of the ecosystem, most of the participants were also end-users themselves, and we maintained deep contact with the user community notably through our continuous dissemination actions. Therefore informal evaluation through first hand experience or witnessing was largely sufficient to inform the design and execution of the project.

1.1. Explanation of work carried out per Objective

For reference, let us recall the aims of OpenDreamKit.

Aim 1 Improve the productivity of researchers in pure mathematics and applications by promoting collaborations based on mathematical **software**, **data**, and **knowledge**.

Aim 2 Make it easy for teams of researchers of any size to set up custom, collaborative *Virtual Research Environments* tailored to their specific needs, resources and workflows. The VRE should support the entire life-cycle of computational work in mathematical research, from initial exploration to publication, teaching and outreach.

Aim 3 Identify and promote best practices in computational mathematical research including: making results easily reproducible; producing reusable and easily accessible software; sharing data in a semantically sound way; exploiting and supporting the growing ecosystem of computational tools.

Aim 4 Maximise sustainability and impact in mathematics, neighbouring fields, and scientific computing.

Those aims are backed up in our proposal by nine objectives; we now highlight our main contributions during this reporting period toward achieving each of them.

Obj 1 **Virtual Research Environment Kit:** "*To develop and standardise an architecture allowing combination of mathematical, data and software components with off-the-shelf computing infrastructure to produce specialised VRE for different communities.*"

By building around the JUPYTER ecosystem, all OpenDreamKit components can be assembled in different combinations to make a VRE suitable for a variety of communities, especially in mathematics and education. This has been highly successful, enabling VREs in fields from mathematics to micromagnetics, all using common OpenDreamKit components, which can be mixed and matched to serve the target community.

This objective is by nature multilevel; achievements include:

- Collaborative workspaces: major JupyterHub and JupyterLab developments, for collaboration, both real-time and via version control systems such as git, see **T4.2**: "Notebook improvements for collaboration";
- User interface level: enabling JUPYTER as uniform interface for all computational components; see **T4.1**: "Uniform notebook interface for all interactive components".
- Interfaces between computational or database components:
  - *short term*: refactoring of existing ad-hoc interfaces, see **T4.12**: "Python/-Cython bindings for PARI";
  - *long term*: investigation of patterns to share data, ontologies, and semantics uniformly across components, see **T3.2**: "Interfaces between systems", and Section 1.2.6 about WP6, where we report on the "Math-in-the-Middle" (MitM) paradigm for semantic system integration and non-trivial mathematical use cases. In RP3, we have added interoperability of mathematical data sets to the mix.

Obj 2 **Core Components:** "*To develop open source core components for VRE where existing software is not suitable. These components will support a variety of platforms, including standard cloud computing and clusters. This primarily addresses Aim 2, thereby contributing to Aim 1 and 3.*"

At this stage, it has been possible to implement most of the required developments within existing components or extensions thereof. For the Math-in-the-Middle paradigm for semantic system interoperability we have developed knowledge-based Mediator based on the MMT system. In RP3 we have concentrated on mathematical data sets. We have added a data aspect to MathHub.info which allows dataset authors to semantically describe data sets and then generate database schemata, management functionality, and user interfaces from that.

Real-time collaboration has been developed as a feature in JupyterLab, as part of : “”, and : “” has delivered nbdime, an important tool for collaboration on JUPYTER notebooks.

These core components can be deployed as part of a VRE. The task of assembling a VRE from these components has been greatly facilitated by work in **T3.3**: “Modularisation and packaging”, including packaging of numerous OpenDreamKit components in the conda packaging system.

Obj 3 Community Building across Disciplines: “*To bring together research communities (e.g. users of* JUPYTER*,* SAGE*,* SINGULAR*, and* GAP*) to symbiotically exploit overlaps in tool creation building efforts, avoid duplication of effort in different disciplines, and share best practice. This supports Aims 1, 3 and 4.*”

We have organized or co-organized a dozen users or developers workshops (see **T2.3**: “Community Building: Development Workshops”) which brought together several communities. Some key outcomes include:

- Enabling JUPYTER as uniform interface for all computational components; see **T4.1**: “Uniform notebook interface for all interactive components”.
- Sharing best practices for development, packaging, building containers (see **T3.3**: “Modularisation and packaging”), and continuous integration (see **T3.1**: “Portability”);
- A smooth collaboration between JUPYTERHUB, and SIMULAGORA; see **T3.4**: “Simulagora integration” and Section 1.4;
- Work on interfaces between systems; see **T3.2**: “Interfaces between systems”, **T4.7**: “Active Documents Portal”, and **T4.12**: “Python/Cython bindings for PARI”;
- Sharing of best practices and tools for authoring live structured documents (see **T4.6**: “Structured documents”);
- Developing interactive documentation
- Sharing of best practices when using VRE’s like COCALC or JUPYTER for research and education;
- Sharing of best practices in developing modularized parallel linear algebra kernels and there integration into SAGEMATH (see **T3.5**: “Component architecture for High Performance Computing and Parallelism”)
- Collaboration on interactive visualization **T4.8**: “Visualisation system for 3D data in web-notebook”, **T4.9**: “Visualisation of 3D fluid dynamics data in web-notebook”, **T4.5**: “Dynamic documentation and exploration system”.
- Jump-starting a community on semantically described, interoperable mathematical data sets around `data.mathhub.info`. A Math Data workshop (8 Days) in Cernay included external mathematicians and will be continued by external partners in 2020;
- Packaging of OpenDreamKit components in the conda-forge ecosystem for easier and wider dissemination and use by a large community;
- Integration of OpenDreamKit participants into the core JupyterLab team, to ensure development of JupyterLab to serve OpenDreamKit needs, in particular live collaboration features : “”;
- Adoption of modern software engineering practices throughout OpenDreamKit components, such as Continuous Integration, leveraging cross-domain expertise in mathematics and software engineering, see D6.9: “Shared persistent Memoisation Library for PYTHON/SAGE”.

Obj 4 Updates to Mathematical Software Components: “*Update a range of existing open source mathematical software systems for seamless deployment and efficient execution within the VRE architecture of objective 1. This fulfils part of Aim 2.*”

Achievements include:

- Continuous efforts of development, release and integration within SAGEMATH have been put for
  - the linear algebra computational kernels of LINBOX, FFLAS-FFPACK and GIVARO (Deliverables D5.12: "Exact linear algebra algorithms and implementations. Library maintenance and close integration in mathematical software for LINBOX library" and D5.14: "Implementations of exact linear algebra algorithms on distributed memory et heterogenous architectures: clusters and accelerators. Solving large linear systems over the rationals is the target application."),
  - the PARI library for computational number theory (Deliverable D5.16: "PARI suite release (LIBPARI, GP and GP2C) that fully support parallelisation allowing individual implementations to scale gracefully between single core / multicore / massively parallel machines." still ongoing),
  - the GAP software for computational group theory (Deliverable D5.15: "Final report and evaluation of all the GAP developments."),
  - the FLINT and Singular software for commutative algebra (Deliverable : "").
- Packaging efforts: docker containers (delivered and regularly updated), Debian and Conda packages; see **T3.3**: "Modularisation and packaging", in particular D3.10.
- Continued efforts on portability of SAGEMATH and its dependencies (see **T3.1**: "Portability").
- Improved continuous integration and development workflow.
- Integration of all the relevant mathematical software in the uniform JUPYTER user interface, in particular for integration in the VRE framework (delivered, ongoing); see **T4.1**: "Uniform notebook interface for all interactive components".
- Work in WP5 and WP3 to better support HPC in the individual mathematical software system and combinations thereof; see Section 1.2.5 and D3.11.
- The SAGEMATH and GAP systems have been extended by a persistent memoization package, which allows to cache computational results and even share them between any system that implementes the memoization format; see D6.9: "Shared persistent Memoisation Library for PYTHON/SAGE" for details.
- Ongoing work on the MMT system which forms the basis of the WP6: Work on **T6.11** has led to a tight integration with the Isabelle theorem prover and a complete revamp of the code for indexing theory morphisms (crucial for the MitM-based integration of VRE components).

Obj 5 A Sustainable Ecosystem of Software Components: "*Ensure that our ecosystem of interoperable open source components is* sustainable *by promoting collaborative software development and outsourcing development to larger communities whenever suitable. This fulfils part of Aims 3 and 4.*"

Achievements include:

- Continued work on outsourcing the computational system user interfaces by migrating to JUPYTER, eventually deprecating outdated interfaces such as the SAGE notebook; see **T4.1**: "Uniform notebook interface for all interactive components";
- Refactoring SAGE's documentation build system to contribute many local developments upstream (SPHINX) **T4.4**: "Refactor SAGE's SPHINX documentation system";
- Outsourcing and contributing upstream as PYTHON bindings the existing SAGE bindings for many computational systems; see **T4.12**: "Python/Cython bindings for PARI".
- Migrating SAGE to Python 3, see **T3.3**: "Modularisation and packaging".

Obj 6 Engineering Social Interactions in Open Source VRE:

This objective was the social science research side of WP7. Following the work plan revisions after Reporting Period 1, the manpower originally allocated to this objective was reallocated to other objectives. There thus was no new achievements in Reporting Period 2 and 3.

Obj 7 **Next Generation Mathematical Databases:** "*Identify and extend ontologies and standards to facilitate safe and efficient storage, reuse, interoperation and sharing of rich mathematical data whilst taking account of provenance and citability. This fulfills parts of Aims 2 and 3.*"

This objective is at the core of WP6; see Section 1.2.6 for details. In the first two reporting periods WP6 has developed the Math-in-the-Middle ontology that acts as the pivot point mediating between system languages in the MitM interoperability framework. This work has been reported in deliverables D6.5 and D6.8.

In the third reporting period the focus of WP6 has been on instantiating the FAIR principles for mathematics (we call the result **deep FAIR**) and turning mathematical datasets into deep FAIR VRE components. This work has been reported in D6.10. The `data.mathhub.info` system, which implements deep FAIR datasets from scratch meets exactly the objectives stated above – but the system is still very young and needs to attract a critical mass of datasets and community. The LMFDB system which has both has been retrofitted with aspects of deep FAIR in OpenDreamKit, and is much more interoperable than at the start of OpenDreamKit. In parallel, and somewhat dual (lightweight/ad-hoc persistent data caching for mathematical software systems), is the work on **T6.9**: *Memoisation and production of new data*. Here we have developed a data memorization format and corresponding memoization packages for PYTHON (for SAGE) and GAP. These have the potential to lead (by collecting computation results on the side) to informal data sets, which can be semantified later.

Obj 8 **Collaborative Research Environments that Transcend Domains:** "*Demonstrate the effectiveness of Virtual Research Environments built on top of OpenDreamKit components for a number of real-world use cases that traverse domains. This addresses part of Aim 2 and through documenting best practices in reproducible demonstrator documents Aim 3.*"

The Ubermag (previously JOOMMF) project is a demonstrator for the versatility of the research environment infrastructure developed in this project. Users of Ubermag can control simulation tools for devices based on nanomagnetism through the Jupyter notebook interface, document and share designs for new devices through static html and pdf versions of the notebook, and complement publications with executable notebooks hosted on myBinder. The required computations are of numerical nature (not symbolic), the simulation results are floating point numbers, and the users are engineers and material scientists from academia and industry – this is very different from doing mathematical computation as pushed forward in most other activities in the project. Nevertheless, the tool sets we have developed and improved can be applied nearly independent of the application domain: the Jupyter Notebook, the tools nbval, nbdime, nbconvert, Binder and 3d rendering of output using k3d.

Another use case of our virtual research environment infrastructure is that of interactive executable textbooks (see **T2.9**). We have demonstrated that such novel and executable text books covering physics, mathematics and data science aspects can all be put together in this framework, and that they benefit from the infrastructure methodology improvements we have achieved. This applicability is independent of the application domain.

A significant outcome of this project is to pave the way for other research projects (EC-funded or not) that make use of Jupyter and its growing ecosystem. While we have made significant progress, it is fair to say that there is substantial work left. By solving large challenges the Jupyter ecosystem, OpenDreamKit has made addressing new use cases feasible that were not possible when OpenDreamKit started.

We note that a number of projects working towards the European Open Science Cloud (EOSC) have chosen the Jupyter Notebook and tools as a technology to realise the EOSC – in particular for remote access, remote analysis and visualisation of data, and for improved reproducibility. These projects will directly build on and benefit from OpenDreamKit. It is also clear that further features are requested by the community and that these will improve the EOSC projects and the EOSC itself. There is currently no resource to provide this further development of the Jupyter ecosystem for EOSC in particular – the funded projects understand themselves as users of Jupyter, not developers.

Obj 9 **Training and Dissemination:** "*Promote and disseminate OpenDreamKit to the scientific community by active communication, workshop organisation, and training in the spirit of open-source software. This addresses Aim 4.*"

[2] This objective is at the core of WP2, with in particular 29 organized events during the third reporting period: 7 development workshops, 10 training and community building workshops (including one targeted at women and one in a developing country),1 research conference and 11 communication and participation in external events. See Section 1.2.2 and D2.15: "Community building: Impact of development workshops, dissemination and training activities, year 4" for more details.

[2]EdNote: @Izabela: update number of meetings and workshops

1.2. Explanation of the work carried per Work Package

We will now tabulate and explain the achievements of the OpenDreamKit project by the work packages.

1.2.1. Work Package 1: Project Management

1.2.1.1. **Overview.** As in the previous reporting periods, UPSud coordinated OpenDreamKit in close collaboration with the other beneficiaries to ensure that:

(1) the objectives of the project were met within the agreed budget and the timeframe specified by milestones and deliverables;
(2) all the risks jeopardising the success of the project are managed and that the final results are of high quality;
(3) the innovation process within the project is fully aligned with the objectives set up in the Grant agreement.

1.2.1.2. **Tasks.**

T1.1: "Project and financial management".

- UPSud took care of the budget management together with the administration body, the D.A.R.I. (Direction des Activités de Recherche et de l'Innovation) and its finance service. This included prefinancing, funds transfer to cater for the moving of personnel across sites and from old sites to new sites, and the coordination of financial reports.
- In earlier reporting periods, UPSud led four amendment processes to the Grant Agreement, to manage work plan revisions and to reallocate staff and all remaining resources from the four terminated beneficiaries UZH, USFD, JacobsUni, SOUTHAMPTON to the added beneficiaries UGent, FAU, XFEL.
  During Reporting Period 3, UPSud led a fifth amendment upon the request of FAU to add a subcontractor to conduct a new task **T6.11**: "Isabelle Case Study". Following the suggestions of the Project officer, UPSud used the occasion to formalize budget transfers between beneficiaries to optimize the use of remaining resources to achieve the project aims; this was notably required to exploit resources left at LEEDS following the early departure of all its personnel.
- In earlier reporting periods, UPSud had organized the first project review in Brussels on April 2017, and steering committee meetings in Orsay (September 2015), St Andrews (January 2016), Edinburgh (January 2017), Brussels (March 2017), online (February 2018), and at XFEL (June 2018).
  During Reporting Period 3, UPSud organized the second project review in Luxembourg on October 2018 and steering committee meetings in Luxembourg (October 2018) and Marseille (February 2019). It also organized a one week "report writing sprint" in Cernay (August 2019) to collectively write the project reports. The final review meeting will take place in Luxembourg on October 30, 2019, and will gather 20 OpenDreamKit participants to present the project final results.
- As in earlier reporting periods, UPSud ensured that all the milestones and deliverables of Reporting Period 3 were achieved within its timeframe, and reported on in a timely manner.
- As in earlier reporting periods, UPSud maintained the internal and external communication tools that were described in D1.1: "Basic project infrastructure (websites, wikis, issue trackers, mailing lists, repositories)". The project website was continuously updated with new content, and virtually all work in progress is openly accessible on the Internet to external experts and contributors (for example through open source software repositories on Github).

- Concerning the future of OpenDreamKit and of its infrastructure toolkit, the consortium kept accessing information and getting involved in the development of the European Open Science Cloud that is currently promoted by the European Commission. The project manager participated to the following events:
  - EOSC Hub week, April 10-12 of 2019, Prague, Czech Republic.
  - Building Open Science in Europe: The road ahead for the EOSC community and the EU Member States, June 20th of 2019, Tallinn, Estonia.
  - ICT Proposers' Day 2019, September 19-20th of 2019, Helsinky, Finland.

  In addition, two spin-off proposals were submitted on January 29th to the H2020 European E-Infrastructure call INFRAEOSC-02-2019:
  - BOSSEE: Building Open Science Services on European E-Infrastructure, with a focus on Jupyter and applications;
  - FAIRMAT: FAIR Mathematical Data for the European Open Science Cloud.

  Both were prepared with the same open strategy as OpenDreamKit, and involved new combinations of OpenDreamKit and external beneficiaries. None was accepted but the respective consortia are determined to resubmit them or variants thereof at the earliest opportunities.

T1.2: "Quality assurance and risk management". We recall that the Quality Assurance Plan is described in detail in D1.3: "Internal Progress Reports year 1, including risk management and quality assurance plan".

- As in the previous reporting periods, UPSud organized the interaction with the Advisory Board, composed of seven members, some of which specifically represent End Users:
  - Lorena Barba from the George Washington University
  - Jacques Carette from the McMaster University
  - Istvan Csabai from the Eötvös University Budapest
  - Françoise Genova from the Observatoire de Strasbourg
  - Konrad Hinsen from the Centre de Biophysique Moléculaire
  - William Stein, CEO of SageMath Inc.
  - Paul Zimmermann from the INRIA

  This Advisory Board is composed of academics and/or software developers from different backgrounds, countries and communities. It is a strong asset to understand the needs of a variety of end-user profiles.
- As in the previous reporting periods, The Quality Review Board monitored the quality and the relevance of the software development relative to the end-user needs. This board – chaired by Hans Fangohr and composed of four members with a track record of caring about the quality of software in computational science – is responsible for ensuring key deliverables do reach their original goal and that best practice is followed in the writing process as well as in the innovation production process. It met after the end of each Reporting Period (RP), and before the Review following that RP. More details are given in Section 4.3.
- UPSud has also been managing risks. Up to the Leeds situation, the assessment we present in Section 4.2 has only marginally deviated from earlier assessments at Month 12 (D1.3) and Month 36 (D1.5).

T1.3: "Innovation management".

- At month 18, UPSud had with the help of the consortium a first version of Innovation Management Plan (D1.4), with focuses on:
  - The open source aspect of the innovation produced within OpenDreamKit;
  - The various implementation processes the project is dealing with;
  - The strategy to match end-users needs with the promoted VREs.

During Reporting Period 3, UPSud produced a second version of the Innovation Management Plan (D1.6). It confirms the elements of strategy of the previous plan, and adds two additional sections: one on the choice and impact of open licenses in the context of OpenDreamKit, and one on the different types of outcome of the project and their respective sustainability. This second version also includes minor updates to the earlier sections, notably to reflect the work plan revisions that occurred after Reporting Period 1.

1.2.2. WorkPackage 2: Community Building, Training, Dissemination, Exploitation, and Outreach

1.2.2.1. **Overview.** We continued the line of work from the previous reporting periods, especially on **T2.1**: "Dissemination and Communication activities" and **T2.5**: "Dissemination: reaching towards users and fostering diversity": we organized or participated in about 30 events throughout the last year, including the large dissemination conference *Free Computational Mathematics* at CIRM in February 2019. Over the whole project, this accumulates to 110 events.

We have also pursued our effort towards greater diversity in the open-source community, organizing the first ever SAGE workshop in Nigeria as well as our second Women in Sage event in Archanes, Crete.

About our communication, we continued our work to reach the community through multimedia. We produced comics with Juliette Belin from Logilab and short motion graphics videos with the Pix Videos company to explain the project and common use cases. This self-explanatory visual material reinforces our website to keep informing the community about the project long after it is officially over.

1.2.2.2. **Tasks.**

T2.1: "Dissemination and Communication activities". Press Releases were considered an important dissemination and communication tool at the start of the project and will also be at the end. During the first year, the project has covered six press releases describing the general goals of the project. To promote OpenDreamKit innovative method and highlight its results to the general public, we plan to submit press releases at the beginning of November, after the Final review meeting. The procedure for the press release production and distribution is still under revision. The text proposal was made available to all the partners inviting them to finalize its publication through their press offices. The final press releases will be published in French, the Coordinator's and 3 partners main language, but also translated in English for the others beneficiaries to enable its publication in local media. We also plan to send this article proposal to our European communication officer to publish it in the EC newsletter and submit it for publication in the Horizon magazine. These Press releases will be addressed to the general press in the high education, research area but also in local press, to audiences that do not require a detailed knowledge of the work carried out.

Beside that, to raise interest of the scientific community on the project topic and its impact, our communication strategy was accompanied by audio-visually enhanced materials targeted at non-specialist. With the help of Pix Videos, we created several explainer comics and life motion-design videos based on the sketches by Juliette Belin from Logilab. These multimedia creations describe different common use-cases of tools either directly developed by OpenDreamKit or that can be used in conjunction with our software. As an example, it describes how to use Binder (an external tool) and JUPYTER (developed by OpenDreamKit) in the context of scientific collaboration. We plan to use those videos and comics to promote open source software now that the project itself is ending.

T2.2: "Training and training portal". The website for the project has been continuously updated with new content, and virtually all work in progress was openly accessible. It became a repository for a wide type of information and communication material. It was used to update on new technical results, and events that might be of interest for our targeted communities, and also to help to share the demos experience, facilitate adoption of project results by the users, to support best practices, *etc.*. Some of our "Use Cases" were illustrated by comics designed by Juliette Belin, giving a quick overview on how to use some of OpenDreamKit tools.

T2.3: "Community Building: Development Workshops". Development workshops are a key aspect of OpenDreamKit development model. The aim of these workshops is to bring together developers from the different communities to design and implement some of the wanted features such as user interface, and documentation and to ensure cross compatibility. As reported in D2.15: “Community building: Impact of development workshops, dissemination and training activities, year 4”, we have organized or co-organized 7 of these workshops throughout year 4 of the project. The thematic varies for each event: PARI/GP, LINBOX, Data, and cross-thematic events such as GAP-SAGE and GAP-SINGULAR days. They were aimed at a specific software components to improve joint developments. It fostered collaboration between scientists and developers from different backgrounds to build tools that are needed by all. These workshops were essential in order to disseminate our work while improving it.

T2.4: “Reviewing emerging technologies”. By nature, most of the work on this task occurred in the earlier reporting periods, especially through D2.3: “Review on emerging technologies”. Of course, we continued keeping track of new technologies and writing about them on our website.

T2.5: “Dissemination: reaching towards users and fostering diversity”. During Reporting Period 3, we organized 10 more training workshops on various components of OpenDreamKit such as SAGE, JUPYTER, GAP, Ubermag and more, to disseminate them to the scientific community. Other the four years of OpenDreamKit, this accumulate to 45 training events and about 1800 attendees.

One of our main dissemination event was the CIRM conference *Free Computational Mathematics*, in Marseille which was aimed at the general scientific community. It was an occasion to showcase many of the tools developed and supported by OpenDreamKit and to promote the spirit of collaboration, free software, and best practices. We had 58 participants mixing different level of expertise, from newcomers to advanced developers, and different software communities (GAP, JUPYTER, LINBOX, MPIR, PARI/GP, SAGE, SINGULAR). Another important dissemination event was the Sage Days 105, organized as a satellite event to the main yearly international conference on algebraic combinatorics involving 50 participants. It featured several tutorial demos and presentations by OpenDreamKit participants including best practices.

We pursued our effort towards better diversity in the open source software community. We widely advertised our *Free Computational Mathematics* conference to reach a large audience and provided funding to many attendees. This allowed in particular the attendance of three researchers from University of Ibadan, Nigeria. They were very enthusiastic about the conference and it was then decided to organize a SAGE workshop directly in Ibadan for the benefit of Nigerian and West-African mathematical community. This event happened in July 2019 and welcomed 80 participants, mostly from Nigeria and neighboring countries.

We also had another *Women in Sage* event, following the one we organized in Paris in 2017. The event was co-organized by Viviane Pons from OpenDreamKit and Eleni Tzanaki, who attended the 2017 Women in Sage workshop. It was held in the village of Archanes in Crete. We welcomed 22 women from 8 different countries, many of them SAGE beginners. The organization of the event was also an occasion for a series of SAGE lectures at the mathematics department of University of Crete which initiated the inclusion of SAGE in the students curriculum.

All the material we developed for presentation at all events organized throughout the project were made publicly available. The impact of development and training workshops was the awareness rising of project results and of the possibilities to strengthen our collaborative open source development model.

T2.6: “Introduce OpenDreamKit to Researchers and Teachers”. Training and disseminating to Researchers and Teachers is at the heart of OpenDreamKit and the participants doubled up their efforts during the last reporting period. This included the organization of training events (see **T2.5**: “Dissemination: reaching towards users and fostering diversity” above), but also many more evaluation and dissemination activities: teaching with OpenDreamKit technology (thereby training students and other instructors alike), local consulting, contributing course material, templates and utilities. This is reported on in D2.17: “Introduce OpenDreamKit to Researchers and Teachers as laid out in Task 2.6”, together with some reflection on the lessons learned at the occasion of these activities: adoption, adequateness for the needs, best practice.

It should be noted that Sheffield (now Leeds) has been the lead on this task until its participants got compelling opportunities in the industry in Fall 2018. This did not reduce the overall dissemination activities of the project: indeed, the freed resources were redistributed to other participants that were eager to organize more activities than originally planned. There was some impact however: with continued leadership some more of the lessons learned at the occasion of those activities could have been formally collated, when currently many are in the state of shared folklore. Luckily this information is still spreading in the community through many channels: informal discussions, blog posts, mailing lists, etc.

T2.7: “Open source dissemination of micromagnetic VRE”. This task was mostly carried out during the first reporting period. The Ubermag (previously called JOOMMF) project is working and available on GitHub (Ubermag repo). For each Ubermag package we use continuous integration on both Travis CI and AppVeyor, where we perform tests and monitor the test coverage, which we then make available on Codecov. Documentation for each package consists of APIs (automatically generated from the code) and different tutorials created in Jupyter notebooks. Both of them are tested on Travis CI. Documentation is built and made publicly available on Read the Docs. After every major milestone, we upload each package to the Python Package Index repository and build a Conda package, which can later be easily installed on different operating systems. We encourage the early use of our software and invite for feedback for which we provide several different communication channels. Ubermag can also be used in the cloud as a Virtual Research Environment, by using Binder services.

T2.8: “Micromagnetic VRE dissemination workshops”. We had several workshops and tutorials during major events where we demonstrated the use of our Micromagnetic VRE, received feedback and feature requests from the community:

- IOP Magnetism in April 2017, univ. of York.
- Intermag in April 2017, Dublin.
- MMM in November 2017, Pittsburgh.
- Advances in Magnetism in February 2018, Italy.

T2.9: “Demonstrator: Interactive books”. In D2.14: “Demonstrators: Problems in Physics with Sage, Computational Mathematics for Engineering” we report on the delivery of two new open interactive textbooks. Together with the two books delivered during RP2 (D2.9: “Demonstrator: interactive books on Linear Algebra and Nonlinear Processes in Biology”), this was the occasions to explore various approaches to exploit OpenDreamKit technology for authoring textbooks. In the deliverable report, we reflect on their respective merits and suggest some best practice.

T2.10: “Demonstrator: Computational mathematics resources indexing service”. Not applicable for this period. The web toolkit *planetaryum* (D2.7) has been delivered in the 2nd reporting period, closing the task.

1.2.3. WorkPackage 3: Component Architecture

1.2.3.1. **Overview.** This Work Package focuses on the structure of the components that make up a mathematical software and their interactions. Such components can be separate modules inside a unique software, or separate software interacting through library calls and/or through APIs. The last reporting period has focused mainly on integration and packaging, producing, maintaining and consolidating official packages for Linux distributions and Conda.

Milestones. Helping end users perform computations on whatever hardware they possess is one of the major goals of OpenDreamKit, and of WP3 in particular. The only milestone involving WP3 is

M5: "ODK's computational components available on major platforms" (month 42). *"User story: users shall be able to easily install ODK's computational components on the three major platforms (Windows, Mac, Linux) via their standard distribution channels."*

The completion of D3.10: "Packaging components and user-contributed code for major Linux distributions" completes the work done in the previous reporting periods by making all OpenDreamKit components available in the major Linux distributions (Debian, Ubuntu, Fedora, ...) through their official channels[3].

Windows and MacOS binaries have also been regularly distributed since at least two years. The milestone has thus been successfully achieved, and we consider the overall work done in this work package a success.

1.2.3.2. **Tasks.**

T3.1: "Portability". The first task of this workpackage is to improve the portability of computational components.

The fundamental milestone in this task was achieved during the second reporting period, with the release of SAGE 8.0 in July 2017, the first release with native support for Windows. At the start of the current reporting period all OpenDreamKit components were thus available for all three major platforms (Windows, Mac, Linux), through several channels: executable installers, official packages, Conda packages and Docker images.

The work in this reporting period has thus essentially consisted in maintaining and consolidating the available software.

T3.2: "Interfaces between systems". In this task we investigate patterns to share data, ontologies, and semantics across computational systems, possibly connected remotely.

The work concerning this work package was essentially completed in Year 1. During this reporting period, a refactoring of the GAP–SAGE interface, in the form of an official C API named LIBGAP, has taken place. For conciseness, this achievement has been reported in a different work package: **T5.2**.

T3.3: "Modularisation and packaging". In this task we investigate best practices for composing, sharing and interfacing computational components and data for connected mathematical systems.

Thanks to the joint efforts of OpenDreamKit and of the community, SAGE has been available as a Debian/Ubuntu package since June 2017, as an experimental Conda package since the summer of 2017, and as a package for several other Linux distributions (Fedora, Arch, Gentoo, ...) for even longer. The work in this reporting period has consisted in consolidating and helping update these packages, as reported in detail in deliverable D3.10.

More maintenance work has gone into keeping the Windows and MacOS versions up to date. In particular, since version 8.2 SAGE is compatible with the C compiler clang, ensuring long-term sustainability for the MacOS port.

[3]Note that the role of OpenDreamKit is to facilitate packaging for Linux distributions, by simplifying dependency management and build chains, and keeping up to date with dependencies. It is not OpenDreamKit's goal to directly take the lead on packaging for the dozens of available distributions, as this would not be sustainable.

An effort has been put into unvendoring system/external libraries and tools historically packaged in SAGE, towards the goal of making SAGE installable via the standard Python tools such as pip. In the reporting period a general unvendoring structure has been designed and implemented, and over 40 non-Python components of SAGE (about 50% of such components) received configuration scripts allowing the use of the corresponding system libraries or tools. Most of this work has been done by OpenDreamKit members.

One of the main roadblocks for this task is porting SAGE to Python 3, indeed as Python 2 is coming to its end of life in 2020, finalizing the port is crucial for the future of SAGE. Fortunately this task was essentially completed during the last reporting period: the current stable version, SAGE 8.8, is already compatible with Python 3, and SAGE 9.0, expected at the beginning of 2020, will be the first SAGE version to be officially based on Python 3.

T3.4: “Simulagora integration”. The goal of this task was to deliver Simulagora VM images that can be used by the end-users to run computations with the tools enhanced or developed in the project.

In the Simulagora platform, we have integrated Jupyter as a key product that can be used by scientists (expert users) to develop dedicated tools with very little development skills : a notebook (containing fields) is used to collect and pre-process the data, this data is used by a script that runs computation codes, and finally a notebook containing the results and their visualization is produced. The scientist only has to write two notebooks and one script in order to produce a customized scientific tool that runs in the cloud and that can be used by several end-users (regular users).

During the project, the improvment in the Jupyter ecosystem and the emergence of the binder tool decrease the usefulness of a platform such as Simulagora. Today, its main benefit is the ability to easily build a Web tool around a legacy heavy computation code (that might be difficult to install with the binder process), or to run parametric studies that need thousands computations.

T3.5: “Component architecture for High Performance Computing and Parallelism”. The goal of this task is to investigate and implement parallelism-friendly ways of combining components together, so that calling components can benefit from the parallelism features of called components.

We used SAGE and its components as a test-bed, by producing an HPC-enabled distribution of SAGE. This distribution is able to exploit parallel features in LINBOX, GAP, SINGULAR and PARI/GP, and some of those have been integrated in the official SAGE distribution. Nevertheless, several roadblocks still exist, preventing these components from cooperating seamlessly in a parallel environment. The work accomplished and the limitations have been reported in deliverable D3.11.

This task was the occasion for a tightly knit collaboration between all OpenDreamKit participants involved in High Performance Mathematical Computing, leading to rich exchanges of expertise and best practice, and the building of a shared vision.

T3.6: “Document and modularise SAGEMATHCLOUD’s codebase”. Not applicable for this period.

T3.7: “Improving the development workflow in mathematical software”. Not applicable for this period.

T3.8: “Python interface for OOMMF micromagnetic simulation library”. Not applicable for this period.

1.2.4. **WorkPackage 4: User Interfaces**

1.2.4.1. **Overview.** The objective of WorkPackage 4 is to provide modern, robust, and flexible user interfaces for computation, supporting real-time sharing, integration with collaborative problem-solving, multilingual documents, paper writing and publication, links to databases, etc. This work is focused primarily around the JUPYTER project, in the form of:

- Enhancing existing JUPYTER tools (**T4.2**)
- Building new tools in the JUPYTER ecosystem (**T4.3**, **T4.2**, **T4.8**)
- Improving the use of OpenDreamKit components in JUPYTER and SAGE environments (**T4.1**, **T4.4**, **T4.5**, **T4.12**)
- Demonstrating effectiveness of WorkPackage 4 results in specific scientific applications (**T4.9**, **T4.11**, **T4.14**, **T4.13**)
- Work on Active Documents, which have some goals in common with JUPYTER notebooks (**T4.6**, **T4.7**)

All deliverables for WorkPackage 4 have been delivered and highly successful in previous reporting periods. There are no new deliverables in Reporting Period 3. However, the work of software is never really complete. Work has continued on some tasks to further improve, mature, and maintain the results of WorkPackage 4 toward sustainability and to best serve OpenDreamKit objectives based on feedback from OpenDreamKit and the wider user community.

Milestones.

M7: “Collaborative VRE for mathematical researchers and beyond” (month 48). *“The prototype VRE shall be extended with improved ease of deployment, new functionality such as interactive 3D visualization and real-time collaboration, enabling researchers to collaborate productively in a shared computational environment. Finally, integrating notebooks and semantic knowledge into a publication / knowledge system enable a continuous process of leveraging OpenDreamKit components from research to publication.”*

The JUPYTER-based prototype for this has been previously delivered in **M6**: “Prototype VRE for mathematical researchers” (month 36), and is extended in **T4.2**: “Notebook improvements for collaboration” to more mature functionality.

WorkPackage 4 has resulted in a number of useful pieces of software for mathematical researchers, sometimes creating new software, improving existing software, or establishing new or improved connections between two existing systems.

Combining the above, Milestone **M7**: “Collaborative VRE for mathematical researchers and beyond” (month 48) has been reached: from the obtained toolkit, we can produce a JUPYTER-based VRE, integrating OpenDreamKit components. The Jupyter kernels delivered in **T4.1** enable access to a broader collection of mathematical software. The interactive utility of software such as PARI is improved in **T4.12**, and general interactivity and exploration of mathematical objects in SAGE is improved in **T4.5**. The scope of what classes of work can be made interactive is increased by the development of interactive three-dimensional visualization tools in **T4.8**. Further, the process of collaboration on notebook documents is improved by **T4.2** and prototype support for live collaboration with **T4.2**. By focusing on JUPYTER as our User Interface of choice, all of these tools can be combined in a single VRE, hosted in the cloud or and made accessible to any researcher, building on the Docker images created in D3.1: “Virtual images and containers”.

The work in this final reporting period has focused on stabilising and maturing the software delivered in previous periods.

1.2.4.2. **Tasks.**

T4.1: “Uniform notebook interface for all interactive components”. All deliverables for this task have been delivered in previous reporting periods.

Kernels for OpenDreamKit components GAP, PARI, SAGE, and SINGULAR, had been delivered in the form of D4.4 in RP1 and D4.7: “Full featured JUPYTER interface for GAP, PARI/GP, Singular” in RP2. Work has continued to develop these kernels in this reporting period to bring them to further maturity and sustainability.

T4.2: “Notebook improvements for collaboration”. All deliverables for this task have been delivered in previous reporting periods.

Prototype components and plan for D4.15 had been delivered in RP2. This has been developed to further complete prototypes of real-time collaboration in JupyterLab in collaboration with the JUPYTER community. We are optimistic about its completion and adoption in JupyterLab in the near future. Real-time collaboration has proven to be the largest and most challenging effort in WP4, both in terms of technical effort and in community engagement. The reason being that real-time collaboration needs extensive work in development in the core of JupyterLab itself, which required collaboration and coordination with the JupyterLab community for assembling plans and implementation, aligning with other goals of the JupyterLab project, including development of new features in the phosphorjs framework on with JupyterLab is based, and a complete refactor of the JupyterLab data model. This work has involved participation in workshops and meetings, as well as addition of OpenDreamKit team members to the core JupyterLab team. As of August 2019, real-time collaboration has been implemented in JupyterLab in a `datastore` branch on the official jupyterlab repository on GitHub, and is expected to arrive in a public release of JupyterLab soon.

In addition, further releases of `nbdime` from D4.6 have been made to better support asynchronous collaboration.

This work furthers OpenDreamKit objective 5 of promoting sustainable software in math and science.

T4.3: “Reproducible Notebooks”. All deliverables for this task have been delivered in previous reporting periods.

D4.8: “Facilities for running notebooks as verification tests” was delivered in the form of a new Python package, `nbval`, which enables testing and verification of existing notebooks via a plugin to the Python testing framework **pytest**. In this reporting period, nbval has received further activity and contributions and new releases. nbval integrates with nbdime from D4.6 to deliver testable, reproducible notebooks via traditional software development testing practices. This work furthers OpenDreamKit objective 5 of promoting sustainable software in math and science.

T4.4: “Refactor SAGE’s SPHINX documentation system”. Even though this reporting period contains no explicit deliverables for this task, significant foundation work was carried out which we now describe. Documentation tools such as Sphinx rely on introspection to harvest the documentation out the sources. For performance, a large fraction of the SageMath sources is however written in Cython (compiled Python) which, until recently, had an incompatible and limited introspection API. This forced SageMath and other projects to maintain bespoke and fragile Sphinx extensions to harvest their documentation.

Tackling this required to dig deep into the system and design, implement, and get accepted a change to Python itself: PEP (Python Enhancement Proposal) 590. PEP 590 makes available Python’s fast calling protocol to custom code, thereby enabling full support for introspection and documentation to Python functions implemented in C – e.g. Cython functions –, with no performance loss. This has been implemented in the upcoming Python 3.8 and Cython 3.0 releases. We expect not only Cython and therefore SageMath to benefit from this, but also other similar projects such as Pythran or Numba.

T4.5: “Dynamic documentation and exploration system”. Due M36 (D4.16)

All deliverables for this task have been delivered in previous reporting periods.

As planned in D4.16, OpenDreamKit packages *Sage-Combinat-Widgets* and *Sage-Explorer* were further developed during RP3. *Sage-Combinat-Widgets* has gained in flexibility and has been applied to a range of new mathematical objects. User interfaces features like feedback have been enhanced, and documentation has been augmented and gained a tutorial. *Sage-Explorer* has gone through a complete new design and reengineering process, at the same time for better modularity in the code and for better ergonomics. Finally, the *Francy* Jupyter-based graph visualisation library was generalized to support PYTHON– and therefore SAGEMATH– in addition to GAP. All three benefited from feedback, if not contributions, from end-users.

T4.6: “Structured documents”. All deliverables for this task have been delivered in previous reporting periods.

During RP3, we developed the JupyterLab extension JupyterLabTraining dedicated to teaching programming, e.g. in Python or Sage. It provides an environment where learners can autonomously do a series of exercises in order to learn a new programming language. Each exercise is an independant Jupyter notebook containing the questions, a cell where the learner will write her code, a hidden cell containing automated tests, and a button to run these tests and check the code that has been written answers the questions. The left panel shows the list of all the exercises; they can be sorted by topic (keyword), complexity or learning track. Thanks to this environment, each learner can do the exercises at his own pace and choose the exercises that focus on his own points of interest. The learning process is thus much more efficient for each person.

We also developed further the thebelab software for interactive computing in traditional web-based documentation.

T4.7: “Active Documents Portal”. All deliverables for this task have been delivered in previous reporting periods.

One of the most prominent features of a virtual research environment (VRE) is a unified user interface. The OpenDreamKit approach is to create a mathematical VRE by integrating various pre-existing mathematical software systems. There are two approaches that can serve as a basis for the OpenDreamKit UI: computational notebooks and active documents. The former allows for mathematical text around the computation cells of a read-eval-print loop of a mathematical software system and the latter makes semantically annotated documents active.

MATHHUB is a portal for active mathematical documents ranging from formal libraries of theorem provers to informal – but rigorous – mathematical documents lightly marked up by preserving LaTeX markup.

As the authoring, maintenance, and curation of theory-structured mathematical ontologies and the transfer of mathematical knowledge via active documents are an important part of the OpenDreamKit VRE toolkit, the editing facilities in MATHHUB play a great role for the project, as delivered in D4.3: “Distributed, Collaborative, Versioned Editing of Active Documents in MathHub.info”.

T4.8: “Visualisation system for 3D data in web-notebook”. All deliverables for this task have been delivered in previous reporting periods.

The software developed for this task has been delivered in earlier reporting periods. Packages such as k3d-jupyter have received further development, improved compatibility with JupyterLab, and developed toward maturity and stability, with growing community adoption in collaboration with related community projects such as ipyvolume towards common goals in interactive visualisation. Several contributions have been made to JupyterLab and the JUPYTER ecosystem to further support similar work, benefiting a wide user community.

T4.9: “Visualisation of 3D fluid dynamics data in web-notebook”. No work to report in this period.

T4.10: “Common option system for various displays in Sage”. No work to report in this period.

T4.11: "Case study: micromagnetic VRE built from OpenDreamKit". The micromagnetic virtual research environment is hosted in the JUPYTER Notebook. The computational backend is the existing OOMMF (Object Oriented MicroMagnetic Framework) simulation tool, which is accessible through the new Python interface that has been created as part of OpenDreamKit (**T3.8**). The JUPYTER Notebook allows us to integrate the micromagnetic model specification, the execution of the simulation, and the postprocessing and data representation within a single executable document; providing a new computational research environment for micromagnetic simulation that uses the most widely used simulation code. We have enhanced this environment further by exploiting that the notebook allows objects to represent themselves in different ways within the notebook. For example, Python objects that represent mathematical equations in the micromagnetic VRE appear rendered as LaTeX in the notebook. It allows users to interactively compose and explore computational models, and to be able to inspect what they have put together in the language of the scientist (i.e. through equations) rather than through the language of the computer (i.e. code). The addition of this representation options does not stop the code from being valid PYTHON that can be run outside the notebook. We have also provided a graphical representation of the mesh and discretisation cell as the appropriate representation of a finite difference mesh to further assist the effective communication between code and science user and graphical representation of vector field objects. We have used dissemination workshops to seek feedback from users and to refine interface.

T4.12: "Python/Cython bindings for PARI". No work to report in this period.

T4.13: "Demonstrator: micromagnetic VRE notebooks" has been merged into **T2.7**: "Open source dissemination of micromagnetic VRE".

T4.14: "Online portal for micromagnetic VRE demonstrator" has been merged into **T2.7**: "Open source dissemination of micromagnetic VRE".

1.2.5. **WorkPackage 5: High Performance Mathematical Computing**

1.2.5.1. **Overview.** This work package is about better exploiting modern parallel computer architectures in computational mathematics software, notably when deployed within a Virtual Research Environment. It is addressed at the level of individual computational components (PARI, GAP, LINBOX, MPIR, SAGE, SINGULAR, ...), and also at the level of interfacing and exposing core parallel features to higher level programming interfaces.

Key results obtained over the reporting period are the following:

- An MPI based distributed rational linear system solver based on Chinese remainder algorithm in LINBOX.
- A new hybrid algorithm for rational linear system solving based on p-adic lifting and Chinese remainder algorithm, and its parallel implementation in LINBOX for multi-core servers.
- A full-featured parallelisation engine, supporting POSIX threads and MPI, for PARI/GP in production release of the software.
- The release of GAP-4.9 allowing compilation in HPC-GAP compatibility mode.
- A major redesign of the polynomial arithmetic used in Singular, delivering state of the art efficiency.

In addition, we investigated how to exploit parallelism when combining computational software; see **T3.5**: “Component architecture for High Performance Computing and Parallelism” and the following milestone.

1.2.5.2. **Milestones.**

M8: “Seamless use of parallel computing architecture in the VRE (proof of concept)” (month 36). *“User story: Astrid wants to run compute intensive routines involving both dense linear algebra and combinatorics. She has access through a JupyterHub-based VRE to a high end multi-core machine which includes a vanilla* SAGE *installation. She automatically benefits from the HPC features of the underlying specialized libraries (*LINBOX*, ...). This is a proof of concept of the overall framework to integrate the HPC advances of specialized libraries into a general purpose VRE. It will prepare the final integration of a broader set of such parallel features for the end of the project.”*

This milestone was already adressed at the second review meeting. Nonetheless, we now report on further developments bringing much of this proof of concept to production.

The solution then proposed was limited to the use of a multi-threaded BLAS and required a minor configuration change at the install time of the SAGE software. D3.11: “HPC enabled SAGE distribution” made this process more flexible to the user, while offering a broader range of parallel routines. Indeed, the Singleton class `Parallelism` now provides the user with total control over the number of threads to be used for the linear algebra computations. The following session demonstrates the gain in parallelizing the product of a random 8000×8000 matrix over $\mathbb{Z}/65521\mathbb{Z}$ with itself using 16 cores, using a vanilla Sage 8.9 installation:

```
pernet@dahu34:~/soft/sage$ ./sage
SageMath version 8.9 Release Date: 2019-09-29
sage: a=random_matrix(GF(65521),8000)
sage: Parallelism()
Number of processes for parallelization:
 - linbox computations: 1
 - tensor computations: 1
sage: time b=a*a
CPU times: user 17.5 s, sys: 1.04 s, total: 18.5 s
Wall time: 18.5 s
```

```
sage: Parallelism().set("linbox",16)
sage: Parallelism()
Number of processes for parallelization:
 - linbox computations: 16
 - tensor computations: 1
sage: time b=a*a
CPU times: user 28.9 s, sys: 4.85 s, total: 33.8 s
Wall time: 2.41 s
```

Moreover, as announced in the previous technical report, the parallel code exposed to the user now includes many more routines, especially those related to exact Gaussian elimination provided by the FFLAS-FFPACK library: `Det`, `Rank`, `Solve`, `Echelon Forms`, etc.

1.2.5.3. **Tasks.**

T5.1: "PARI". Deliverable D5.10: "Devise a generic parallelisation engine for PARI and use it to prototype selected functions (integer factorisation, discrete logarithm, modular polynomials)" was merged with D5.16: "PARI suite release (LIBPARI, GP and GP2C) that fully support parallelisation allowing individual implementations to scale gracefully between single core / multicore / massively parallel machines." in the revised workplan. The deliverable D5.16 is released on time.

The release 2.12 of the PARI/GP suite features a MultiThread engine, used transparently in all tools from the suite: the PARI library, the command line interface `gp` and the GP2C compiler. Written in 2015 and 2016, the engine supports sequential evaluation (no parallelism), POSIX threads and MPI within the same code base. It is now being progressively used wherever it makes sense in the code base and this is by nature work in progress. In PARI-2.12, the MT engine is a central component of

- fast (near linear time) Chinese remaindering;
- fast linear algebra over $\mathbb{Q}$ and cyclotomic fields, a critical component of the new "Modular Forms" package;
- polynomial resultants in $\mathbb{Z}[X] \times \mathbb{Z}[X, Y]$ (via fast Chinese remainders and evaluation / interpolation), a basic tool for algebraic number theory;
- computation of classical modular polynomials for about 20 classical invariants (j, Weber functions, small eta quotients...);
- discrete logarithm over finite fields (prime fields and $\mathbb{F}_{p^e}$ for word-sized prime p) ;
- Adleman-Pomerance-Rumely-Cohen-Lenstra primality proof;
- Fourier coefficients of L-functions (Hasse-Weil and Artin L-functions);
- hi-resolution plot of mathematical functions using parallel evaluation.

The `master` branch on the public development server includes further

- values of complex L-functions (via parallel computation of Meijer G-functions);
- a new thread-safe version of the Multiple Polynomial Quadratic Sieve (MPQS) integer factoring algorithm, ready to be parallelized;

The parallel-enabled components of the PARI/GP suite have been advertised (including tutorial sessions) and tested by participants during PARI/GP workshops in Grenoble (2016), Lyon (2017), Besançon (2018) and Bordeaux (2019).

T5.2: "GAP". Deliverable D5.15: "Final report and evaluation of all the GAP developments." was completed at the end of this period, reporting all of the developments in the GAP system during the project which are relevant to, or provide essential context for, this workpackage.

During reporting period 3 the main areas of effort were:

- Follow-up work to the integration of HPC-GAP into the main codebase reported in the previous period. This work improves the robustness of the system and dramatically reduced the differences between the two versions of the source code.
- Development of the GAP interface to the meataxe64 high performance linear algebra library (to which we also contributed significant development effort). This system targets large calculations over small finite fields on multi-core shared memory computers.

  These calculations are the key kernel underpinning higher level computations in representation theory (eg modular character tables) as well as computations in matrix groups and a wide range of other problems.

  The interface makes almost all of the capabilities of meataxe64 callable from GAP, something which is not only a quantum leap in performance for GAP in this critical area, but also allows easy prototyping in GAP of new algorithms for meataxe64. This library can make use of multiple cores whether or not it is being called from HPC-GAP. A full set of benchmarks are included in D5.15, but as a highlight, two dense random $320\,000 \times 320\,000$ matrices over $GF(2)$ can be multiplied in just over 1000 seconds on a 64 core AMD "bulldozer" system.
- Release of "libGAP" a general-purpose C API for GAP. This allows any program, including in particular HPC code, to call on the functionality of GAP efficiently and without the need to run a separate GAP process. This makes fine grained interaction possible.

  To interface with GAP, SageMath formerly used a bespoke implementation of "libGAP", requiring heavy patching of GAP. Having the functionality available upstream reduced considerably the maintenance burden for SageMath developers and packagers alike.
- Development and release of a new linguistic reflection API in GAP, allowing GAP programs to access and modify the executable representation of their own functions at run-time. This will be the basis of future automatic parallelisation and optimisation tools.
- Very much improved profiling tools
- Release of the new package "ferret" which achieves world-leading performance in partition backtrack, a critical, and notoriusly challenging computational kernel
- Extensive developments in our testing and release infrastructure with the overall goal of ensuring that GAP users have easy access to a reliable, up-to-date and mutually compatible set of versions of the large suite of packages redistributed with GAP.

T5.3: "Linbox". During this reporting period, we delivered D5.14: "Implementations of exact linear algebra algorithms on distributed memory et heterogenous architectures: clusters and accelerators. Solving large linear systems over the rationals is the target application.".

A first focus was made on distributed computing, with an MPI parallelization of a Chinese remainder based algorithm. The first proof-of-concept implementation was then cleanly integrated in the mainstream code of the library. Its performance shows a very nice scaling with the number of compute nodes on a 256 cores cluster.

Although this approach is best suited for parallelization on a large number of nodes, its total computational complexity ($\tilde{O}(n^4)$) becomes a major concern on large instances. The usual alternative approach based on p-adic lifting has a better total computational complexity ($\tilde{O}(n^3)$), but is intrinsically more sequential, and therefore less suited for large scale parallelization. A major contribution in this task is a new algorithm combining p-adic lifting and Chinese remaindering in order to expose more parallelism without sacrifying the gain in complexity. We also provide a full-featured implementation of this new algorithm in LinBox, which delivers high sequential efficiency and nevertheless scales well up to 16 cores. This compromise is a good fit for personal computers or the typical lab-wide computational server researchers have access to.

Lastly, we introduced support for GPUs in the FFLAS-FFPACK library and showed how matrix product over a finite field benefit from these accelerators.

All these software improvements are closely integrated in the mainstream code of the FFLAS-FFPACK and LINBOX libraries.

T5.4: “Singular”. The only deliverable under consideration for this reporting period is D5.13: “Parallelise the Singular sparse polynomial multiplication algorithms and provide parallel versions of the Singular sparse polynomial division and GCD algorithms.”

Multivariate polynomials are represented in Singular using the sdmp format. While this data structure is generally amenable to parallelization, the implementation and some of the algorithms in Singular were not. Much work has been invested in updating the algorithms and data structures and making Singular polynomial arithmetic competitive with other systems. This work has been done in the Singular submodule Flint, whose code is available at `https://github.com/wbhart/flint2`.

We now support polynomial exponents of unlimited size with the three basic monomial orderings of lex, deglex, and degrevlex over the integers mod p and rationals. Continued engagement by colleagues in the HPC community including Bernard Parisse, Michael Monagan, Roman Pearce, and Mickaël Gastineau has been invaluable.

Parallel and serial implementations of the operations of multiplication, division and GCD are complete and perform well in both the dense and sparse cases. The performance is more than competitive with all other systems we are aware of, both on a single core and on multiple cores.

Basic arithmetic in Singular now benefits directly from the implementation and researchers are already working on leveraging the new implementation in other areas, e.g. Gröbner bases over rational functions, Gröbner bases with a bottleneck on multivariate arithmetic and polynomial factorisation. Early indications are that all of these are going to experience a huge improvement for many real-world research applications.

Other systems such as the new Oscar computer algebra system already benefit directly from the new ODK implementation.

Sage will automatically benefit directly at the next update of the Singular version in Sage.

T5.5: “MPIR”. Not applicable for this period.
T5.6: “HPC infrastructure for combinatorics”. Not applicable for this period.
T5.7: “Pythran”. Not applicable for this period.
T5.8: “Sun Grid Engine Integration in Project JUPYTER Hub”. Not applicable for this period.

1.2.6. **WorkPackage 6: Data/Knowledge/Software-Bases**

1.2.6.1. **Overview.** In a series of workshops (September 2015 in Paris, January 2016 in St. Andrews, June 2016 in Bremen, and July 2016 in Białystok, 2017 in Orsay, 2018 in Cernay, 2019 in Cernay), the participants working on WP6 met and discussed the topic of integrating the OpenDreamKit systems into a mathematical VRE toolkit. Additionally, Florian Rabe was employed at both FAU and UPSud throughout 2018 and 2019 to deepen the integration.

Key results of the first two reporting periods were

R1. the observation that *knowledge-aware interoperability of software and database-systems is the most critical objective* for WP6 in the OpenDreamKit project.

R1. the consensus that this can be achieved by *aligning the mathematical knowledge underlying the various systems*,

R1. the existing integration of mathematical computation systems in the Sage and Jupyter systems must be complemented with a similar integration of mathematical databases.

This requires explicitly representing the three aspects of math VREs – Data (D), Knowledge (K), and Software (S) – and basing computational services and inter-system communication on a joint $\mathcal{DKS}$-base. These results are engrained in the "Math-in-the-Middle" (MitM) paradigm [Deh+16], which gives a representational basis for specification-based interoperability of mathematical software systems – so that they can be integrated in a VRE toolkit. In the MitM paradigm, the mathematical knowledge (K) underlying the VREs and the interface for each system (S) are represented as modular theory graphs in the OMDoc/MMT format. For the data aspect (D) we have extended the concept of OMDoc/MMT theories to "virtual theories" that allow the practical management of possibly infinite theories, see [D6.518] for details.

Through the concerted effort of the WP6 participants, we have been able to implement this design and instantiate it with theory graphs for the GAP and SAGE systems and integrating the LMFDB (see [D6.518]). Based on this, we were able to generically integrate GAP, SAGE, and LMFDB via the standardised SCSCP protocol [HR09]. This case study shows the feasibility of the design.

In the **third reporting period**, the focus was on the **representation and curation of mathematical data**, building on the earlier work. We have refined the original notion of $\mathcal{DKS}$-bases from the grant proposal into a tetrapodal structure which captures the four primary aspects of "doing Maths" that have to be supported in a VRE toolkit: narration (papers and textbooks), computation (algorithms and software), inference (theorems and proofs), and tabulation (database schemas and datasets). These are joined via a fifth modular organization aspect – see the introduction of [D6.1019] and [Car+19] for a discussion.

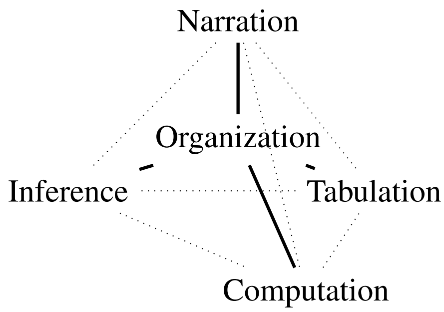

FIGURE 1. Five Aspects of Math VREs, a Tetrapod Structure

We have taken up the general discussion of research data, the FAIR principles, and have adapted them to the case of mathematical research data. The outcome of this was the observation that – even though mathematics deals with ideal and abstract objects – it is often possible to describe these objects concisely by representing them as database structures in a way that complements their formal symbolic descriptions. The codecs from the "virtual theories" approach developed in WP6 tie these two representations together: they link the database level of mathematical data sets with the MitM ontology – and from there, via the interface theory graphs, interface it to the mathematical software systems in OpenDreamKit.

We undertook three larger case studies to bring this about:

- developing the system data.mathhub.info for managing mathematical data sets MitM-style and equipping it with a search UI; see the report on **T6.8** below,

- exporting the Isabelle knowledge base (via a subcontract), and equipping it with a semantic search facility; see the report on **T6.11** below,
- extending the formula search capabilities developed in the first reporting period to Jupyter notebooks; see the report on **T6.10** below.

This wraps up and integrates the work in WP6 combining aspects of Data (D) (now captured by tabulation), Knowledge (K) (now captured by organization and inference), and Software (S) (now captured by computation). Importantly, the joint system addresses semantically the central aspects of all four FAIR requirements for the open sharing of research data. For a joint and integrated final report on this, see [D6.1019].

1.2.6.2. **Milestones.**
M9: "First Math-In-The-Middle-based interoperability prototype" (month 36). This milestone was addressed in the second reporting period.

M10: "Second Math-In-The-Middle-based interoperability prototype" (month 48). *"The goal of this milestone was to take into account all the operational experiences with the first prototype and add more systems and integrate some of the UI components from WP4. The experiences with the preparation of this prototype allow us to estimate the joining costs of adding a system to the OpenDreamKit VRE toolkit, which is an important measure of the flexibility of the Math-In-the-Middle approach."*

The state of the MitM VRE middleware is sufficiently mature that most of the functionality can be configured by writing domain and system knowledge in form of OMDoc/MMT theories, but without requiring extensions of the system (e.g., changes to the programming of the VRE systems or the MMT mediator). This means that additional computational systems can be added at the cost of generating system API theories, extending the MitM ontology, and supplying alignments.

In the *third review period* we have concentrated on mathematical data sources. Our analysis of systems in **T6.6**, **T6.7**, and even more **T6.8** has revealed that, to achieve deep and meaningful FAIRness of mathematical research data – in particular of interoperability (I) – we have to semantically model the mathematical objects in math-aware representations (as described above). As a consequence, we have to integrate mathematical data systematically into the OpenDreamKit VRE toolkit, giving rise to the refined tetrapodal model described above. In particular, this integration must consider individual datasets rather than dataset-related systems like OEIS, FindStat, or LMFDB. The `data.mathhub.info` system was expressly designed to do this: we can just specify a dataset in the MDDL language (see [BKR19; D6.1019]) and then generate a database infrastructure including user interface and import facility from it, resulting in a MitM-Interoperable VRE component for that dataset. We have tested and evaluated this setup at the occasion of a special Math Data Workshop [WDM], using five data sets supplied by one internal and two external mathematicians. This showed that – after a learning period of a day and with some help from OpenDreamKit knowledge engineers, external users can MitM-integrate datasets by specifying their structure and semantics and supplying them in a specification-conforming format in under a day of work. We anticipate that creating the alignments (see [D6.518]) that relate the mathematical background of these datasets to algorithms in computational MitM systems like GAP, SAGE, and SINGULAR is of the same order of magnitude.

1.2.6.3. **Tasks.**
T6.1: "Survey of existing $\mathcal{DKS}$ bases, Formulation of requirements". This task was addressed in the first reporting period.

T6.2: “Triform Theories in OMDoc/MMT”. This task was addressed in the first reporting period.

T6.3: “$\mathcal{DKS}$ Base Design”. This task was addressed in the first reporting period.

T6.4: “Computational Foundation for Python/Sage”. This task was addressed in the first reporting period.

T6.5: “Knowledge-based code infrastructure”. This task was addressed in the second reporting period. In this period, the infrastructure has matured and been extended with a data component `data.mathhub.info`.

T6.6: “OEIS Case Study (Coverage and automated Import)”. This task was addressed in the first reporting period.

T6.7: “FindStat Case Study (Triformal Theories)”. This task was addressed in the second reporting period.

T6.8: “LMFDB Case Study (Triformal Theories)”. Work on this task had already started in the second report period. There we had used the concept of virtual theories developed in **T6.2** to MitM-integrate (parts of) the LMFDB and make it interoperable as a VRE component. During this process, it became apparent that to achieve meaningful interoperability (the I in FAIR), we have to model the mathematical datasets as described above. This prompted us to refocus on an integration into the OpenDreamKit VRE toolkit at the dataset level, not at the level of systems like OEIS, FindStat, or LMFDB. Indeed, LMFDB is itself a collection of over 80 datasets and the work on the semantics and representation of datasets catalyzed a large inventory of datasets in LMFDB. At the start of the OpenDreamKit project, LMFDB had been growing organically based on the schema-less MongoDB database, and no-one had an complete overview on the details and extent of the content. The Warwick group led a move to inventory all the data sets, and to (manually) recover their specifications at the mathematical and data base level (schema information), which has in turn facilitated the recent move from MongoDB to PostGreSQL (independent of OpenDreamKit). Another fruit of the OpenDreamKit work was a vastly improved and more semantic API for LMFDB (see `http://www.lmfdb.org/api2/`) that has recently come online. LMFDB’s earlier API was just a very thin HTTP wrapper over the database core of LMFDB. API2 adds full SQL querying support and first steps towards “semantic/mathematic” queries. A SAGE interface based on API2 is currently under development.

While the LMFDB has “retrofitted” a more semantic treatment of datasets on LMFDB, the FAU group has explored building a dataset hosting system and user interface based on the MitM model from scratch. The `data.mathhub.info` system leverages the MitM ontology for specifying the mathematical objects in a dataset and extends MitM with a set of MDDL (Math Dataset Description Language) specifications that link the mathematical specifications with database schemata. The MDDL specifications are implemented as a set of codecs that implement the transformations between the layers, encapsulate the translations of mathematical queries to SQL queries in the underlying database generated from MDDL, and provide user interface widgets for the corresponding mathematical types. We have tested this setup on five external data sets – we did not want to duplicate work with LMFDB – and so far the `data.mathhub.info` design seems to scale. When the set of codecs collected in `data.mathhub.info` and made available for reuse by other datasets reaches a point of saturation, we expect joining the costs for `data.mathhub.info` to become restricted to the dataset-immanent information.

T6.9: “Memoisation and production of new data”. We have developed persistent memoization modules for Sage and Gap that can use both local and remote data stores. The design is meant to be completely configurable to support a variety of use cases. In particular, when a common storage format exists (e.g. via OpenMath or MitM), the two systems can share the same data stores. This development bridges the gap between computation and data, enabling for example the progressive tabulation of data sets. as users run their usual computations.

We report on this task in detail in D6.9.

T6.10: "Math Search Engine". This task ran over the whole duration of the OpenDreamKit project. The third reporting period with its refined model of the semantic level of mathematical VRE components has given us a new view on this task as well.

Generally, search is one of the FAIR principles of research data (F), and adapting it to mathematical data/knowledge/software has been one of the central topics in WP6 (and WP4). In the grant proposal we had concentrated on formula- and full-text search, but our deeper understanding of the categories of mathematical data (see [D6.1019]) developed during the OpenDreamKit project showed that, for mathematical semantic search, we need to better take into account the kind of data:

- For **symbolic data** (organization and inference in Figure 1), formula search as reported on in [D6.116] is sufficient as long as the context of all formulas is included in the search index. For this, we have pioneered an export of the knowledge base of the Isabelle system (and others) into symbolic OMDoc formulae and RDF triples that they can be searched via SPARQL queries. See our description of the new **T6.11** below for details.
- In **narrative data**, we need full-text search capabilities; this has been addressed in the first reporting period – see [D6.116] again.
- For **tabulated data**, we need a mathematical query language in which users can express information needs at the mathematical level, and which can be compiled into e.g. SQL queries at the database level. This was initiated in the second reporting period. In the third reporting period, it has greatly matured, been integrated into the new `data.mathhub.info` system, and extended by a user interface generation system.
- For **computational data**, i.e. mathematical software, there are two options: we can search source code, or we can search the mathematical artefacts in the programs. As source code search is already provided by repository hosting systems like GitHub or GitLab we have concentrated on the latter. Concretely, we built a system to harvest mathematical formulae from Jupyter notebooks, index them in the MathWebSearch engine, and provide a specialized user interface for searching them. The main technical development has been to make the MathWebSearch engine – which had been mostly experimental – more deployable and manageable so that it can be used as a VRE component and so that it can be integrated into the OpenDreamKit VRE toolkit without developing instance-specific code.

We have reported on all the aspects of this task in detail in [D6.1019].

T6.11: "Isabelle Case Study". For many decades, the development of a universal database of all mathematical knowledge, as envisioned, e.g., in the QED manifesto [Qed], has been a major driving force of computer mathematics. Today a variety of such libraries are available. These are most prominently developed in proof assistants such as Coq [**coq**] or Isabelle [Isa] and are treasure troves of detailed mathematical knowledge. Within OpenDreamKit, we have developed interface standards, specifically OMDoc for symbolic and ULO for relational knowledge, that allow maintainers of formal libraries to make their content available to outside systems.

In this task (which has been added in the last amendment of the grant agreement), we have exported the large Isabelle knowledge base as both OMDoc/MMT and ULO format. Concretely, we have built an exporter from the Isabelle Theorem prover library (Archive of Formal Proof) to both MMT and RDF data. This exporter is now part of the latest releases of both Isabelle and MMT, and the exported data is available online.

We report on this task in detail in D6.10.

1.3. Impact

All the information of section 2.1 of the DoA is still relevant.

There is for now no change to bring to Key Performance Indicators. The evolution of the measures between Month 18 and Month 48 allowed the Coordinator to assess that selected KPI were as appropriate as could be given the complexity of formal evaluation discussed in the context Section 1.

1.4. Infrastructures

Per design, OpenDreamKit focuses on delivering “a flexible toolkit enabling research groups to set up Virtual Research Environments”. As such, there is no e-infrastructure deployed and managed by OpenDreamKit. Instead, there are many e-infrastructures that use the software developed or contributed to by OpenDreamKit, and we regularly help with new or updated deployments.

Some of the typical content of this section (e.g. Selection Panel, ...) is therefore irrelevant for OpenDreamKit, and we simply provide some informal information and figures on the main existing deployments and their typical public, together with some assessment of the impact we had on them.

- cloud.sagemath.org With 500k accounts worldwide and 30k active projects both for research and education, SAGEMATHCLOUD is the largest Virtual Research Environment based on the ecosystem OpenDreamKit contributes to. Predating OpenDreamKit, it benefits back from most of our actions. OpenDreamKit has been contributing to a healthy collaboration/competition relation between JUPYTERHUB and SAGEMATHCLOUD, with the competition occurring only at the level of specific individual components and both teams learning from each other.
- JupyterHub @ EGI We have partnered with EGI, and helped them deploy an experimental instance of JupyterHub on their infrastructure. This service is now in their catalog, available to all academics in Europe, `https://jupyterhub.fedcloud-tf.fedcloud.eu/`, and actively advertised by EGI `https://www.slideshare.net/EGI_Foundation/reproducible-open-science-with-egi-notebooks-bin` We are jointly seeking for ways to make this service sustainable in the long run, in particular as part of the EOSC.
- jupyter.math.cnrs.fr We have helped setup this JUPYTERHUB service, deployed by the French CNRS for the benefit of the personnel of all math labs in France. This service includes all the OpenDreamKit computational components.
- mybinder.org Binder is a web service that makes it easy for any user to publish live notebooks based on an arbitrary reproducible executable environments. It thus fosters dissemination and reproducible research. At the time of our previous report, (`http://mybinder.org/`) used to be often overloaded by the demand, proving that it had identified just the right service for a critical need. Meanwhile, binder has started to evolve into a federation of instances, two of which are hosted by Google and OVH respectively, as free services to the community.

  Our work on Docker and conda packaging **T3.3** and JUPYTER integration **T4.1** enabled the easy definition of executable environments based on OpenDreamKit’s computational math software, for Binder and beyond.

  Within our EGI partnership, we have started exploring ways to contribute additional computing resources to the mybinder federation by setting a new one for the EC community: `http://binderhub.fedcloud-tf.fedcloud.eu/`. For the current status, see `https://github.com/OpenDreamKit/OpenDreamKit/issues/205`.

We are also supporting the convergence between Binder and JupyterHub, to bring the flexibility of Binder computing environments to JupyterHub; see: https://opendreamkit.org/2018/03/15/jupyterhub-binder-convergence/.

Finally, we contributed to the dissemination of Binder, notably by the authoring of an explainer comic and motion graphics video. These have grown popular and thanks to their Creative Commons licence are being regularly reused by disseminators all around the world.

- JupyterHub @ JUPYTERHUB instance deployed on USheffield's HPC system.
- JupyterHub @ UPSud, UVSQ, ... We have helped with the definition, deployment and maintenance of several University wide instances, exercising them with large classes (e.g. 400 students at UPSud. Lessons learned at the occasion have been shared through blog posts such as:
  - https://opendreamkit.org/2018/10/17/jupyterhub-docker/
  - [3] https://blog.jupyter.org/how-to-deploy-jupyterhub-with-kubernetes

Many more institutions are deploying JupyterHub instances. We are keeping track of the instances we are aware of at https://github.com/OpenDreamKit/OpenDreamKit/issues/174.

[3] EDNOTE: Does anyone know a way to make this URL break appropriately (or should we use bit.ly?

2. Update of the plan for exploitation and dissemination of result (if applicable)

Not applicable

3. Update of the Data Management Plan

There is a generally accepted consensus in the Mathematical community on sharing data, and the OpenDreamKit team members are long time proponents of the FAIR principles. In addition, by nature, ethical issues are nonexistent. Finally, *producing* data was not at the core of OpenDreamKit; in particular, we did not produce huge data sets requiring expensive infrastructure just to store them. The preparation of the original Data Management Plan during Reporting Period 1 (D1.2: “Data Management Plan V1”) was therefore straightforward. Only marginal updates were required during Reporting Period 2 (D1.6: “Data Management Plan V2”) and this latest version remains adequate for this last reporting period: only a few additional data sets have been produced or curated during the project, and are hosted and disseminated under the same FAIR conditions. The process was eased by the semantic data management facilities in LMFDB and `data.mathhub.info` (a from-scratch mathematical dataset hosting/search/management system; see WP6 description).

4. Follow-up of recommendations and Quality Management

In this section, we will detail our actions in response to the recommendations and comments of the reviewers and review the risk management and quality assurance procedures adopted in OpenDreamKit.

4.1. Follow-up of recommendations

We are extremely grateful for the very constructive comments and recommendations that were provided during the review itself and in the formal report.

Recommendation 1. *A minor aspect: In the deliverable D5.11, authors have to clarify the reason why the speedup with the use of cores is not so high when you increment the number of cores. The presentation has also to be improved.*

Deliverable 5.11 was polished, complemented with a clarification and resubmitted after the review.

Recommendation 2. *To include the KPIs in a centralized way in the technical report (KPI table)*

All KPI's are presented within a single section of the Technical Report for reporting period. We tried making this section into a table; however many of our KPI take the form of qualitative narratives and do not fit well in such a table. Instead, we made individual tables for each quantitative KPI.

Recommendation 3. *Demonstration of capabilities due to project results is crucial, especially for test cases/show cases. Often, such demonstrations are extremely technical. A higher level approach to such demonstrations is needed, so that potential users are not taken aback by the many actions they need to undertake. It would be good if the project team discusses this, and takes action to make demonstrations more attractive and appealing. This is also vital for the sustainability of the project results.*

We have discussed the matter within the project and we will be trying our best during the next review. We are facing an intrinsic difficulty due to OpenDreamKit's toolkit strategy. We indeed have a long experience of delivering very progressive demonstrations at our training workshops; in fact we often introduce some of the technology stack even to hundreds of freshmen students at the occasion of our classes. However much of that technology is not a single product of OpenDreamKit per se. Rather it is an ecosystem of products, to which OpenDreamKit makes many contributions. Often the contributions are by nature almost invisible from the end-user perspective, and it takes some technical context to highlight their relevance and impact on usability or sustainability.

Recommendation 4. *Financial statements must be made available to reviewers no later than 15 days before Review Meeting in final form and to the Commission much earlier.*

Our previous project manager had left the project in August 2018, and despite a rehiring process started as early as May 2018 the position remained vacant until December 2018. We are happy to report however that we very lucky in the recruitment. Our new project manager Izabela Faguet – together with the project coordinator – took early and active steps to devise, advertise, and enforce a strict time line to ensure that all partners – including UPSud itself – would submit their financial reports well on time.

Recommendation 5. *It is to be hoped that spend can be accelerated in the next year to make best advantage of the funds available and to ensure maximum benefit to the communities.*

On the day after the review a brainstorm was run among the participants to explore opportunities of funds reallocation. Practical feasibility was then explored and selection of best opportunities were then discussed in the following weeks.

The largest source of tentatively unused funds was from Leeds, following the departure to industry of its members. Most of the resources ($\approx$180 k euros) were redistributed to the other partners. This proved extremely useful to organize many additional dissemination events – making up for Leeds departure – add a new case study exploring the MitM approach in the context of proof systems, and generally speaking increase the participant involvement on existing tasks. This is formalized and explained in the 5th grant agreement amendment.

Another source of tentatively unused funds was from UNIKL, Indeed, those funds were originally reserved for the organization of conferences early in the project which could finally be covered from other sources. It was decided that UNIKL would cover some of the expenses ($\approx$60 k euros) for the organisation of our large dissemination event at CIRM, enabling UPSud to reallocate funds for other dissemination events.

Recommendation 6. *Greater attention must be paid to acknowledgement of EU funding in all areas. For exemple, include the name of the project in Software Carpentry: Related projects:* `https://software-carpentry.org/join/projects/`

We made sure that EU funding was acknowledged by the projects we developed or contributed to: Jupyter, JOOMMF, GAP, LinBox, pypersist, MathHub, Memoisation, MPIR, MMT, PARI/GP, SageMath, Sage Combinat widgets, Sage Explorer, Sage GAP Semantic Interface, Singular, Ubermag.

We have reached to Software Carpentry; their site is under reconstruction and the aforementioned page is about disappear. We are working with them for a proper location for the acknowledgment. Presumably this will be on the pages of the lessons that ODK contributed to.

We also asked the partners to double check their publications.

Recommendation 7. *To develop a comic explaining the MitM approach.*

The comic has been published on: `https://github.com/OpenDreamKit/OpenDreamKit.github.io/blob/master/public/images/use-cases/MitM.png`. It has already been used in the MitM use case description at `https://opendreamkit.org/2018/05/16/lmfdb-usecase/`, in conference presentations and posters.

Recommendation 8. *To disseminate the Adoption by Logipedia of the MitM principle of integrating (logical) systems by aligning concepts.*

We have made a blog post about this, see `https://opendreamkit.org/2019/01/24/logipedia/`.

Recommendation 9. *Some guidelines (set of recommendations) for using the different tools provided by OpenDreamKit would be recommendable.*

We have expanded our use case section on `opendreamkit.org` and will keep doing so.

Recommendation 10. *Some guidelines (set of recommendations) for using the different hardware architectures would be recommendable.*

[4] EdN:4

[4] EDNOTE: @ClementPernet: hardware architecture recommendations

4.2. Risk management

4.2.1. Recruitment of highly qualified staff

Recruitment of highly qualified staff was planned to be a high risk when the proposal was written. And unfortunately it turned out we were right. In such a field as computer science and software development, potential candidates who are likely to be fairly young considering only temporary positions are offered, are very scarce. Furthermore they need to make a choice between public and private bodies which are very attractive, and the choice between pure development and research. The risk concerns not only recruits, but also senior members of the project, and includes the usual hazards of life.

Over the whole duration of the project, OpenDreamKit partners faced the following Human Resources issues:

- Five unplanned departures for industry (Neil Lawrence, Tania Allard, Michael Croucher, Markus Pfeiffer, Benoît Pilorget),
- Three PIs moving to another site with some or all of their team (Michael Kohlhase, Hans Fangohr, Michael Croucher),
- An early retirement (Ursula Martin),
- Two deaths (Hans-Peter Langtanger, Jan Aksamit),
- One long term leave,
- Four paternity and maternity leaves of various duration (Martin Sandve Alnæs, Jeroen Demeyer, Viviane Pons, Min Ragan-Kelley).
- Some complex hiring processes due to visa issues and the like.

Most of them were reported on in previous reporting periods. The main event in the last reporting period was, in Fall 2018, the departure for the industry (at NAG) of Michael Croucher, lead PI for Leeds (formerly Sheffield), leading to the termination of all activities there. As documented in Section **??**, work and resources were successfully reallocated.

Altogether, this confirmed that the recruitment and keeping of highly qualified staff is indeed a risky endeavour, which induced delays on several deliverables. However the planned mitigation measures – taking into account the pool of potential candidates in the design of the positions, aggressive advertisement, weak coupling between tasks, spreading of critical tasks (e.g. dissemination) over several beneficiaries – worked adequately: with proactive reshuffling of the work plan, this did not impact the overall progress of the project. We were also really happy with the top notch quality of our recruits all along the project.

Throughout the project, we proactively sought to improve the gender balance in our community in general, and our personnel in particular. The latter is a tough endeavor given the tight pool of experienced research software engineers on temporary positions and its high gender imbalance – notably in mathematics. This effort went essentially unsuccessful earlier on in the project. We are happy to report that we recruited four women near the beginning of Reporting Period 3: Theresa Pollinger as Junior Researcher and Katja Bercic as PostDoc at FAU, and Odile Benassy as Research Software Engineer and Izabela Faguet as Project Manager at UPSud. Even if very far from achieving gender equality, this is a step forward.

4.2.2. Different groups not forming effective team

A recurrent risk in large projects such as OpenDreamKit is to have different groups not forming effective teams. Put in other words, having developers of the different pieces of software working solely for the benefit of the program they were initially working on and for.

As expected, this risk was tamed by the existence of many preexisting collaborations between the partners and of “joint itches to scratch together” (to use a common open source software metaphor). In addition, this risk was continuously tackled by the Coordinator and the consortium at large, notably through the organization of dozens of joint workshops which brought together members from the different groups and of the community to bootstrap joint activities through brainstorms and coding sprints. It was quite satisfying to witness diverse groups formed of e.g. a Micromagnetism researcher, a mathematician, and a dev-op expert solving together a shared issue.

The evolution of the social organization of software development communities with a long history is by nature a slow process; nevertheless, ever since the beginning of the project, collaborative efforts across communities are becoming a standard practice, within and outside OpenDreamKit.

4.2.3. Implementing infrastructure that does not match the needs of end-users

In general this is a major risk for infrastructure projects. Tackling it is at the heart of OpenDreamKit's approach. All of the following points contributed to naturally steer by demand the design of the proposal and the governance of the project:

- Co-design is intrinsic to the by-users for-users development model of OpenDreamKit's ecosystem;
- We contributed to existing project with well established user communities;
- The open bottom-up approach strongly involved the community from the inception;
- Most of the participants were also end-users themselves;
- We maintained deep contact with the user community notably through our continuous dissemination actions.
- We sought counsel and feedback from our Advisory Board which includes end-user representatives.

In addition, the toolkit approach mitigates the silo effect. End-users can adopt and adapt individually each tool, or a whole infrastructure combining them, according to their needs.

As mentioned in the introduction, the approach has however the drawback of making formal evaluation – of adequacy and adoption on the one hand, and of our contribution on the other hand – a more complex matter; this reflects in our multiform KPI's (Section **??**).

4.2.4. Lack of predictability for tasks that are pursued jointly with the community

As planned, we regularly shifted manpower around to adapt for the variability of the involvement of the community in the different tasks. For example, the SageMath Jupyter kernel of D4.4: "Basic JUPYTER interface for GAP, PARI/GP, SAGE, Singular" was mostly implemented by the community which allowed to focus on other tasks such as the long term task D3.7: "One-click install SAGE distribution for Windows with Cygwin 32bits and 64bits". On the other hand many other deliverables were implemented with very little help from the community.

4.2.5. Reliance on external software components

There is not much to report on this front: none of the external software component we relied on have failed us. Quite on the contrary, critical software like JUPYTER have continued to blossom. Besides the high modularity of the design meant few components were critical to the overall success of the project.

4.3. Quality assurance plan

4.3.1. Deliverables quality: Quality Review Board

The Quality Review Board is the Consortium Body that fosters best possible quality in the delivered work of the project. All four members of the board have a research interest in the quality of software in computational science, and use and share their experience to benefit the quality of the work.

The board was chaired by Hans Fangohr, from the University of Southampton and European XFEL GmbH (Germany). He is supported in this task by Mike Croucher from the University of Leeds and now Numerical Algorithm Group (UK), Alexander Konovalov from the University of St Andrews (UK), and by Konrad Hinsen from the Centre de Biophysique Moléculaire (France) with whom a Non-Disclosure Agreement was signed.

These board members engage with European initiatives working towards improvement of the software quality in research, in particular in computational and data science; both as voluntary activities and key of their professional roles. Mike Croucher was the head of research computing at Leeds, and is well known through his outreach blog; Alexander Konovalov is a fellow of the Software Sustainability Institute and an active member of the Software Carpentry community; Konrad Hinsen has founded and is editing the ReScience Journal for reproducible Science, and Hans Fangohr is the founder of the UK's only centre for doctoral training in computational modelling with focus on software engineering training for scientists, a fellow of the Software Sustainability Institute, was chairing the EPSRC's national scientific advisory committee on high performance computing, is heading data analysis infrastructure development at the European XFEL research facility, and leading the data analysis work package in the Photon and Neutron Open Science Cloud H2020 project that works towards implementation of the European Open Science Cloud.

The quality review board has reviewed deliverables after the reporting period 1 and 2, identified good practice - both in terms of software engineering content but also presentation of the work -, produced reports, and shared the findings with all members in the project to improve the quality of the remaining deliverables. An improvement of the quality of deliverable reports at the end of reporting period 2 in comparison to reporting period 1 was noted. The reports are available on request. The board has stuck to its no-blame culture in its reporting, but has pointed out deliverable reports of very high quality.

4.3.2. Infrastructure quality: End-user group

It was decided by the Steering Committee during the kick-off meeting to slightly modify the management structure by having only one gender-friendly Advisory Board composed of 6 people (as agreed a few months later at the Bremen meeting), some of which to be end-users.

Members of the board are: Jacques Carette from the McMaster University, Istvan Csabai from the Eötvös University Budapest, Françoise Genova from the Observatoire de Strasbourg, Konrad Hinsen from the Centre de Biophysique Moléculaire, William Stein who is CEO of SageMath, Inc. (SME), and Paul Zimmermann from INRIA.

BOP:5

5. Deviations from Annex 1

There was no major deviation from Annex 1. All deliverables due for M18 were delivered within the timeframe of the 1st Reporting Period, and all milestones in this period were reached. Slight modifications were brought to WP5 and WP6 and were included in the AMD-676541-13.

5.1. Tasks

No deviation from the tasks. All workplan is on time at the end of the Reporting Period.

However, there were four deliverables that were handed in late. We will explain the situation and implications in each case. Therefore, administratively speaking, Milestone 2 (Implementations), originally due on Month 24, was only reached in Month 36, though with little, if any, consequences on the project as a whole.

5.1.1. D2.7: Community-curated indexing tool (open source)

This deliverable, a tool for publishing and organizing curated collections of Jupyter notebooks, was delivered in month 36, a delay of 12 months after the initially planned schedule.

The initial plan had been to build upon an already existing prototype, whose development had started at a Sage meeting back in 2015. The original tool was specific to Sage, but broader in scope and not tied to Jupyter. Given the constantly evolving context, it quickly became apparent that this tool did not properly address the community needs. We thus took on evaluating other available open source solutions, which considerably delayed the deliverable.

As we were not able to find an appropriate solution to build upon, we finally decided to bootstrap a new project, called *planetaryum*. Once the development of *planetaryum* started, we were able to complete the version 0.1 in the planned time frame.

Since other solutions were available before *planetaryum*, albeit less powerful, the delay in the deliverable did not impact other tasks in the project.

5.1.2. D4.7: Full featured Jupyter interface for GAP, PARI/GP, Singular

This deliverable of full-featured kernels for GAP, Pari, Singular, etc. has been delivered in month 36 after a delay of 12 months. The initial plan was for delivery in Month 24, but was delayed to ensure a high quality of the delivered software, as more time-sensitive resources were directed away from this task during months 12-24. The delay had no impact beyond the deliverable itself, as no other tasks relied strongly on this deliverable being ready, and the result is a much stronger collection of Jupyter kernels for mathematical software.

5.1.3. D4.13: Refactorisation of Sage's Sphinx documentation system

This deliverable is delivered in month 36, a delay of 12 months after the initial schedule of month 24. Progress was slower than planned, due to the nature of coordinating with large software collaborations. Additionally, work was shifted to other tasks during early stages, resulting in the delay of this deliverable. There have been no negative consequences of the delay, as its delivery was not a prerequisite for other tasks. As a result of the delay, we have delivered much greater work than initially planned, including significant improvements to the Sphinx documentation system itself used by projects all over the world, and an Enhancement Proposal to improve the Python language itself, ensuring wide impact for this work.

[5] Old Part: MK: this seems to be from the previous version and needs to be dealt with somehow

5.1.4. **D6.5: GAP/SAGE/LMFDB Interface Theories and Alignment in OMDoc/MMT for System Interoperability**

This deliverable report was delayed, as we found it useful to extend the scope of the work reported from just the format of the interface theories and alignments – these are extensively discussed in the report as well – to a full account of the Math-in-the-Middle interoperability paradigm for OpenDreamKit and discuss two full-scale use cases. It just made more sense to deliver this report together with D6.8 (the resources for the use cases) for Milestone M9 *First Math-In-The-Middle-based interoperability prototype* in month 36, in particular, since the delay of D6.5 did not delay the research and development in WP6 (after all, an earlier version of much of the content of D6.5 has been pre-published as [WKR17; Koh+17] near the original deadline of D6.5 and was therefore available to the OpenDreamKit partners).

This way D6.5 can serve as as reference for opening the MitM paradigm to outside users. We are currently working on a high-visibility Journal publication based on D6.5 and D6.8 (presumably Journal of Symbolic Computation).

5.2. **Use of resources**

All changes of use of resources were included in the two amendments previously cited and were due to modifications in the personnel. Those adjustments were due to the change of positions of some key OpenDreamKit participants and expected difficulties in hiring planned staff. The work plan has been updated accordingly, with no foreseeable impact on the achievement of tasks, deliverables, and milestones.

Another minor deviation in the proposed use of resource was that FAU hired students to do some routine jobs (simple formalizations, and the creation of alignments in WP6 and the creation of example documents in WP4) that did not require the attention of a mature researchers. As the pay grade of student assistant is roughly 1/4 of that of full researchers, this action was cost-effective. An unplanned effect was that the reported person months went up considerably, exceeding the planned amount, without incurring additional cost.

Due to cancellation of WP7, the efforts of UOXF were redirected to other WPs, including WP2, where they were originally not active. Within WP2, UOXF delivered mini-courses on various ODK components, such as GAP and SAGE.

A major deviation from the work plan for CNRS was an overspending of PM in WP5 and underspending in WP4. This reallocation is due to a change of priorities in the tasks assigned to CNRS and the specific competences of the hired engineer. In particular we decided to drop **T4.10**: this task is generally useful in the long term, but low priority; since the writing of the proposal, it appeared that its implementation would require much larger amounts of code refactoring and backward incompatibilities than originally expected; in addition the landscape of visualization libraries (in particular in javascript) has been evolving at a very fast pace which is likely to quickly; hence the refactoring will be more efficiently achieved in a later project, once the landscape will have stabilized.

Taking this decision enabled us to reinvest the freed resources (26PMs) on more critical and urgent tasks, as listed below. Additional efforts are made to provide native Windows support for the OpenDreamKit components, in particular the cysignals library (6PM); this was requested by many users and developers and will enable the native use of PARI/GP from Python via cypari on Windows. Some extra efforts are invested in the fine grain parallelization of PARI/GP (4PM). Finally, most efforts are dedicated to modularity and Python3 compatibility for SAGE (16PM).

A major deviation for USTAN is the reduced cost of manpower due to exchange rate changes. In the application we used an expected rate of € 1 = £0.69. In our most recent financial report we used an actual rate of € 1 = £0.88. The monthly rates paid in pounds have been in line with our budget as had the effort in terms of man months, but the cost of this in Euros has fallen dramatically. We anticipate a modest increase in effort beyond the original budget over the final reporting period, but expect to fall well within the financial budget.

UVSQ claiming effort in WP6 is due to Luca De Feo spending 5 hours on WP6 from April 24th to 28th, 2017. He helped Michael Kohlhase's team (from FAU) to prepare the WP6 presentations. The benefits of his collaboration are here: `https://link.springer.com/chapter/10.1007%2F978-3-319-72453-9_14`.

5.2.1. Unforeseen subcontracting (if applicable)

Not applicable.

5.2.2. Unforeseen use of in kind contribution from third party against payment or free of charges (if applicable)

Not applicable. EOP:5

6. Deviations in the third Report Period

6.1. Tasks

6.1.1. New Task T6.11: *Isabelle Case Study*

An amendment to the grant agreement[6] introduced the new task **T6.11**. During the course of the project, it became evident that we should also test the practical coverage of the trifunctional modules, by transforming an existing, high-profile library of formalized mathematics (in contrast to computational mathematics) into OMDoc/MMT. We introduced the new task to this end. EdN:6

It conducted a case study on the Isabelle library. This task was carried out in collaboration Dr. Makarius Wenzel, the main developer of Isabelle on a subcontract (as explained below).

6.2. Deliverables

6.2.1. Changed the title of D6.10: Towards Mathematical Data as VRE components

The title of this deliverable was originally *Full-text search (Formulae + Keywords) in OpenDreamKit*, but in the last grant proposal amendment the scope was broadened to a report on the remaining WP6 activities and achievements – also to account for the new task **T6.11**. As the focus last reporting period was on integrating mathematical data, the title was changed to better account for this.

6.2.2. New Task T6.11: *Isabelle Case Study*

The report on this new task is included in Deliverable D6.10: Towards Mathematical Data as VRE components.

6.3. Unforeseen subcontracting (if applicable)

FAU has subcontracted the Isabelle case study (**T6.11**) to Dr. Makarius Wenzel's company sketis.net.

Note that the task as a whole was unforeseen — once the task was added, the decision to subcontract was rather obvious: Dr. Wenzel is the main developer of Isabelle who has spent the last ~ 10 years building the technological prerequisites for such a case study. There is no alternative way of conducting such a case study since developing the necessary expertise in-house would have been prohibitively expensive (if possible at all). Because, since the start of OpenDreamKit, Dr. Wenzel has left academia and started his own company specializing on Isabelle development, the above-mentioned amendment to the grant amendment included a subcontract for his company.

[6] EdNote: @NT: Is there a formal reference to this, e.g., by number of date? (It's OK to ignore this ednote.)

7. Key Performance Indicators (KPIs)

The following Key Performance Indicators (KPIs) show how OpenDreamKit addresses the specific impacts listed in the work programme. KPIs were thought through by the members of OpenDreamKit so that they are meaningful, reusable, realistic and easily measurable. the following qualitative and quantitative indicators are divided into the four aims of OpenDreamKit. If quantitative indicators are more useful for reporting and internal evaluation, qualitative indicators will give content for father dissemination and communication purposes, for example through the project website.

7.1. KPIs for Aim 1

Recommendation 11. *Aim 1 Improve the productivity of researchers in pure mathematics and applications by promoting collaborations based on mathematical software, data, and knowledge.*

7.1.1. Success story

(1) RP2 **How OpenDreamKit supported the RSE revolution**
In a blog post on October 2018, OpenDreamKit fellow Mike Croucher describes How OpenDreamKit supported the RSE revolution, saying that ”By 2015, there were a small number of central ‘Research Software Engineering Groups’ within UK Universities with his group at Sheffield being among the first. OpenDreamKit was one of the first projects they won that demonstrated that funders would support RSEs on major grants – this improved credibility of the new role a great deal and helped secure its future at Sheffield.” Mike Croucher is a Research Software Engineer at the university of Sheffield, passionate about improving the quality of research software. He enables researchers to ask larger and more complex research questions by improving the software they develop. Along with the Software Sustainability Institute, the UK Research Software Engineering Association and the EU-funded OpenDreamKit project, Mike Croucher actively campaign to improve the career prospects of the talented people who underpin a huge amount of computational research. . . .

7.2. KPIs for Aim 2 and adoption of OpenDreamKit's technologies

Recommendation 12. *Aim 2 Make it easy for teams of researchers of any size to set up custom, collaborative Virtual Research Environments tailored to their specific needs, resources and workflows. The VRE should support the entire life-cycle of computational work in mathematical research, from initial exploration to publication teaching and outreach;*

7.2.1. Success stories

(1) RP2 **Jupyter's ACM Software System award**
In June 2018, Jupyter was awarded prestigious 2017 ACM Software System. Award previous winners include: UNIX, TCP/IP, the Web, TeX, Java, GCC, LLVM.

(2) RP2, RP3 **Partnership with EGI**
there is an ongoing collaboration between EGI (main stakeholder of the EOSC) and OpenDreamKit to deploy JupyterHub and BinderHub-based EGI services. Proofs of concepts for both have been deployed.Both parties are very satisfied with the collaboration and want to strengthen it. A Technology Provider Agreement was signed between EGI and Simula, on behalf of OpenDreamKit/Jupyter developers, and a joint application submitted to the EOSC call INFRAEOSC-02-2019.

(3) RP2 **Use of Jupyter components by Logilab's Simulagora**
One of the flagship product of Logilab is Simulagora, an industry-grade VRE for collaborative computational simulation; it eases the deployment of micro web applications

to leverage the access to complex simulations to end users. Since Logilab joined OpenDreamKit, it was able to outsource several of the core components of Simulagora to replace them by analog standard Jupyter components, bringing in more interactivity.

(4) RP2: **Blogs about how to deploy Jupyter-based VREs**
- Luca De Feo: Deploying a containerized JupyterHub server with Docker
- Nicolas Thiéry: Toward versatile JupyterHub deployments, with the Binder and JupyterHub convergence
- Loic gouarin: Deploying JupyterHub with Kubernetes on OpenStack

(5) RP3: **Multiple European Binder instances run as a free service for the community**
Multiple instances of the Binder VRE-deployment tool are deployed throughout the world, which can leverage ODK components. In addition to the original deployment by the Binder team, sponsored by Google Cloud, French cloud provider OVH and GESIS, the Liebniz institute for Social Sciences in Germany, sponsor and operate instances in the Binder Federation. Additionally, The UK-based Turing Institute is joining the Binder Federation by the end of 2019. This is an indication of the wide adoption of Jupyter ecosystem for VRE-deployment, and an indicator of the strong choice made by ODK to center much of the work on Jupyter. By focussing on Jupyter, users of ODK components can easily benefit from developments such as the Binder service.

(6) **Multi-Site involvement of Researchers (Mobility of Researchers)**
The excellent collaboration between the OpenDreamKit partners has led to an increased mobility of researchers. Rehires between partners and joint appointments are a good measure of this:
- PD. Dr. Florian Rabe (Joint appointment UPSud/FAU)
- Felix Schmoll Summer Internship (From JacU to St.Andrews)
- Prof. Nathan Carter (Bentley Univ.) in St. Andrews (Sabbatical)

(7) RP2: **A MitM- and OpenDreamKit-inspired initiative in the the theorem proving community**
Logipedia (`http://logipedia.science`) was launched in 2018 and aims at integrating proof assistants in a way similar to what OpenDreamKit did for computation systems. The FAU site of OpenDreamKit has collaborated with the group behind Logipedia throughout OpenDreamKit, and several ideas of OpenDreamKit, especially the MitM approach, are closely related to the ideas underlying Logipedia. An EU-infrastructure proposal centered around Logipedia (with participation of FAU) is in preparation.

(8) RP3: **Immediate Community Building around `data.mathhub.info`**
- On the Math Data Workshop in Cernay in August 2019 we invited two external researchers who specialize in mathematical datasets in combinatorics. They contributed five datases (data, descriptions, schema theories, and provenance) for inclusion in `data.mathhub.info` and have been active after the workshop. They indenpendently recruited two of their colleagues for the Slack channel that focuses the nascent community.

7.2.2. Quantitative metrics

(1) **OpenDreamKit based VRE deployments**
We are tracking collaborations with various institutions and projects to deploy instances of JupyterHub and CoCalc (formerly SageMathCloud). There are **seven major OpenDreamKit based VRE deployments** so far: seven at the scale of academic institutions (universities of Zurich, Paris Sud / Paris Saclay, Sheffield, UVSQ, Gent, University of Oslo, NTNU), and two at the scale of research networks (EGI, Mathrice).
There are many more than those listed above, however due to the success of the VRE toolkit components, it is impossible to track all such deployments, as OpenDreamKit

Year	Jupyter kernels
2015	49
2016	75
2017	94
2018	113
2019	132

FIGURE 2. Jupyter kernels in September of each year during the course of OpenDreamKit.

Year	Total (incl non-ODK)	SageMath	Xeus-cling C++	GAP	Singular	PARI/GP	MMT
2016	467836		0	0	0	0	0
2017	1220829						0
2018	3087257	6199	684	63	8	3	1
2019	5540456	8047	1216	94	11	5	2

FIGURE 3. Count of Jupyter notebooks with ODK kernels in September of each year during the course of OpenDreamKit. Per-kernel counts were only tracked from 2018.

members need not be contacted or involved in any such deployments. This is a great advantage of the open source approach used in OpenDreamKit, but presents a challenge for measurement of success.

(2) **Systems and languages integrated in Jupyter**
At the beginning of ODK (September, 2015), there were 49 Jupyter kernels (languages or systems that could be used in Jupyter). This has increased significantly over the course of ODK. There are now 132 of them, 6 of which were contributed or significantly improved by ODK. Growth has been steady throughout the course of ODK, as seen in Figure 3

(3) **Use of Jupyter notebooks**
Millions of Jupyter notebooks are already online with over five million on GitHub alone. Close to ten thousands of them are using Jupyter kernels (co)developed by ODK, as seen in Figure **??**.

(4) **Download statistics for SageMath's Windows installer**
[7] OpenDreamKit invested 6 months worth of work in a native Windows application for SageMath which was delivered in mid 2017. In 2018, it counts for 44% of SageMath downloads.

(5) **Download statistics via platform-specific distribution channels**
At the beginning of OpenDreamKit, not all of ODK's computational systems were available in the standard package repositories of major Linux distributions (e.g., Debian, Ubuntu, Fedora, . . .). Following major efforts from OpenDreamKit and the community this was resolved in 2017. with the availability of SageMath on Debian/Ubuntu.

The "popularity contests"[4] organized by the various Linux distributions give widely different numbers, reflecting the varying interests of the communities tied to each distribution. Figure 4 shows historical package installation statistics for the duration of OpenDreamKit in Arch and Debian (the only distributions that make historical data available). Non-historical data for Ubuntu shows a similar trend to Debian. These figures are based on voluntary reporting, and are thus subject to important biases; nevertheless, we observe an increase in adoption of OpenDreamKit packages starting in 2017, in particular for JUPYTER.

[7] EDNOTE: @defeo, @embray: update Windows installer download stats

[4] Public statistics on package use, based on voluntary submissions by users.

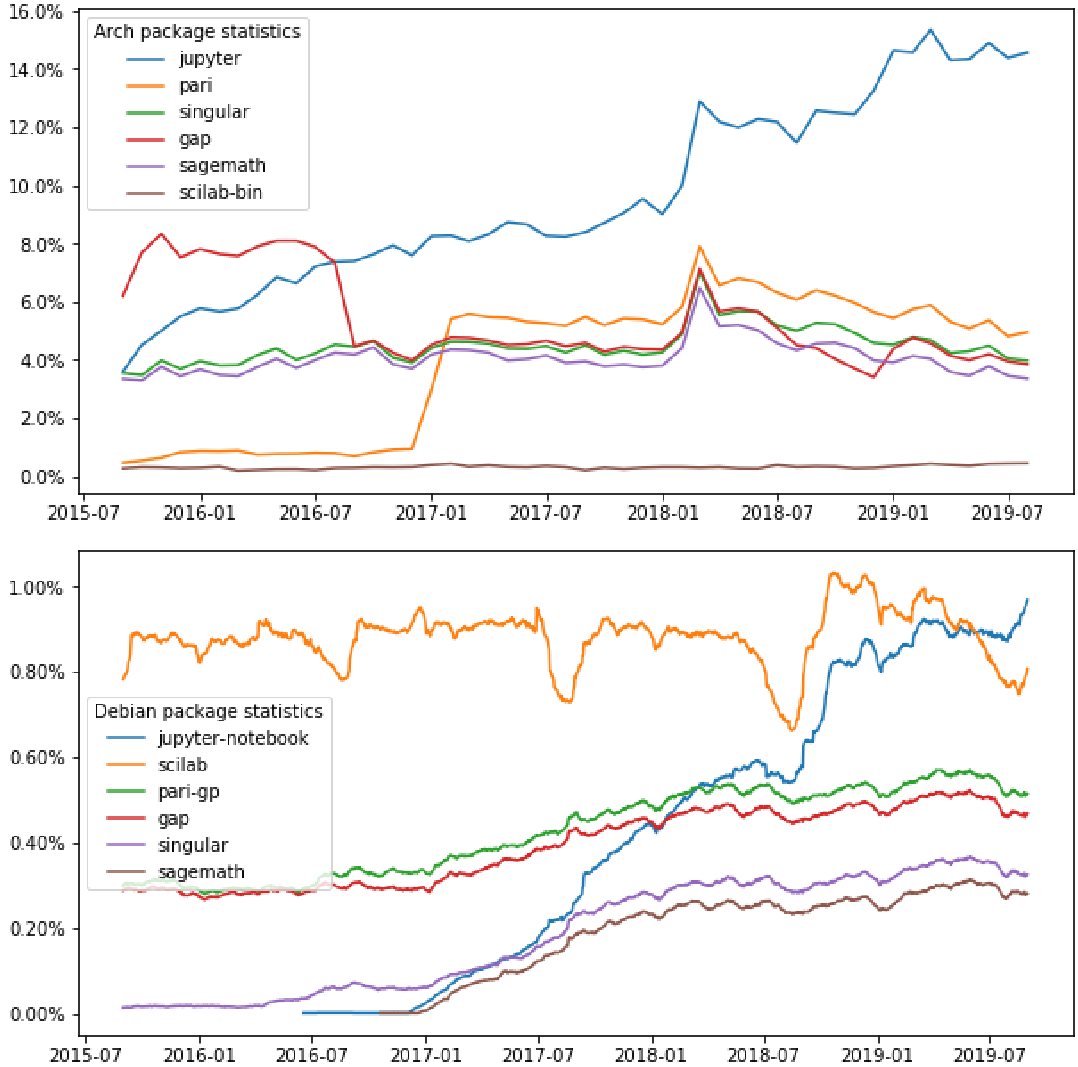

FIGURE 4. Arch and Debian package statistics for the project duration (2015-2019).

Because Jupyter is used in many research areas outside of mathematics, it is interesting to compare the percentage of ODK users to that of Jupyter users. For example, in 2019, in the Arch Linux distribution 34% of Jupyter users are also OpenDreamKit users; this percentage jumps to 58% on Debian and Ubuntu. Both figures are down from their values in 2018, because, unsurprisingly, the popularity of JUPYTER is growing faster than that of other OpenDreamKit components.
It is also (mildly) interesting to compare the number of *pulls* on DockerHub for Jupyter and for OpenDreamKit's containers (available for GAP and SageMath). In 2018, Jupyter's `scipy-notebook` container has more than 1 million pulls, whereas ODK components have between 10 and 100 thousands.
This numbers should be taken with a grain of salt, since they are automatically incremented by continuous integration/testing systems deployed by the projects themselves.
OpenDreamKit has made most of its components available on Conda Forge. The SageMath 8.9 package currently has 25K downloads (compare to 750K downloads for Jupyter).

7.3. KPIs for Aim 3

Recommendation 13. *Aim 3 Identify and promote best practices in computational mathematical research including: making results easily reproducible; producing reusable and easily accessible software; sharing data in a semantically sound way; exploiting and supporting the growing ecosystem of computational tools.*

7.3.1. Success stories

(1) **Mike Croucher's talk "Is your research software correct"**
This excellent talk highlights crucial best practice whenever software is used in research, including open code and data sharing, automation, use of high level languages, software training, version control, pair programming, literate computing, or testing. A lot of the work in ODK relates to disseminating this set of best practice (WP2: "Community Building, Training, Dissemination, Exploitation, and Outreach"), and enabling it through appropriate technology (WP4: "User Interfaces"). Just to cite a few examples, D4.6: "Tools for collaborating on notebooks via version-control", and D4.8: "Facilities for running notebooks as verification tests" enable respectively version control and testing in the JUPYTER literate computing technology, while Mike's talk is and will be delivered in several of ODK's many training events.

(2) **RSE Conference Workshop: "Reproducible research with Jupyter"** (blog post)
OpenDreamKit member, Tania Allard, ran a hands-on workshop on Jupyter notebooks for reproducible research at the UK RSE Conference. This workshop focused on the use of Jupyter notebooks as a means to disseminate reproducible analysis workflows and how this can be leveraged using tools such as nbdime and nbval. Both nbdime and nbval were developed by members of the OpenDreamKit project as a response to the growing popularity of the Jupyter notebooks and the lack of native integration between these technologies and existing version control and validation/testing tools. An exceptional win was that this workshop was, in fact, **one of the most popular events of the conference** and we were asked to run it twice as it was massively oversubscribed. This reflects, on one hand, the popularity of Jupyter notebooks due to the boom of literate programming and its focus on human-readable code. Allowing researchers to share their findings and the code they used along the way in a compelling narrative. On the other hand, it demonstrates the importance of reproducible science and the need for tools that help RSE and researchers to achieve this goal, which aligns perfectly with the goals of OpenDreamKit.

(3) **Best presentation: 3D at the rescue for visualizing large mathematical ontologies** blog post
The Math-in-the-Middle (MitM) ontology and the system API theories in the MitM paradigm are big theory graphs with thousands of nodes and edges. Understanding and interacting with such large and complex objects is very difficult.The FAU group has conducted research into whether virtual reality technologies are helpful for this task. We have presented a first working prototype at the Conference on Intelligent Computer Mathematics CICM 2018 and the author: Richard Marcus - a master's student at FAU has received a prize for best presentation.

(4) **Explainer cartoon on publishing reproducible logbooks**
As part of our training and dissemination activities, we authored a series of Use Case blog posts: examples of applications that have been made possible through the OpenDreamKit project. The most important ones are illustrated by an explainer cartoon authored by Juliette Belin, graphic designer originally working at Logilab and nowadays as free-lance. The cartoon about Binder was retweeted by Chris Holdgraf from the Binder team: "I love this visual description of BinderHub from @JulietteTaka - obviously I'm a fan of

Binder itself, but it's particularly great to see a visual take on the subject! I'd love for the open community to find more ways to invite these kinds of non-code contributions!". It had much success and we henceforth received several requests for reuse:

- Becky Arnold, from the Alan Turing Institute, London, UK, for The Turing Way, *a lightly opinionated guide to reproducible data science*.
- Mark Hanly PhD, Lecturer at Centre for Big Data Research in Health of the University of Sydney, for "a report I am preparing for the Australian Research Data Commons and aims to inform how the ARDC can support researchers to use platforms like Jupyter Notebooks and Binder to disseminate their research. Juliette's cartoon is a very effective way to communicate the motivation and workflow; congratulations on an excellent visualisation!".
- Sarah Gibson, from the Alan Turing Institute, London, UK, for a blog post "Diving into Leadership to Build Push-Button Code for BinderHub".
- Enol Fernandez, Gergely Sipos, from the EGI fundation for a EOSC-Hub talk "Reproducible Open Science with EGI Notebooks, Binder and Zenodo".

7.3.2. **Quantitative metrics**

(1) **GAP packages: activity and adoption of best practices**
GAP has a powerful extension system that allows users to share their research code through an official channel. Thanks to GAP's continuous integration/testing, extension developers have been encouraged and helped to keep their code up to date. The *code coverage*[5] of GAP has gone up from 69% for the 4.9 release, to 75% for the 4.10 release. During the year 2018, 50% of GAP extension packages had been updated; by June 2019 this number had grown to 80%.

(2) **SageMath packages on PyPI**
Following the lead of GAP, SageMath has been advocating, with the support of OpenDreamKit, the Python package repository PyPI for users to share their research code. This channel is still in its infancy. In 2015 there were a handful of packages on PyPI for SageMath; in 2018 the number grew to 117.

(3) **Systems made interoperable with the Math-in-the-Middle architecture**

The Math-in-the-Middle interoperability architectures is a foundational new way of making open-API systems interoperable (see [D6.518] for an overview). Thanks to a fully functional prototype integrating GAP, SAGE, SINGULAR, and LMFDB via the SCSCP Protocol, users can run calculations involving any combination of those systems from any of them.

The following numbers highlight the state of play as of M36 (details in [D6.818]):

MitM-connected Systems: four (GAP, Sage, LMFDB, Singular)(See D6.5);
Formal MitM Ontology: 55 files, 2600 LoF, 360 commits (See D6.8);
Informal MitM Ontology: 815 theories, 1700 concepts in English, German,(Romanian, Chinese);
MitM System API Theories (GAP, Sage, LMFDB, Singular): 1.000+; Theories, 22.000 Concepts;
SageMath Ontology: 512 CDs with 2800 entries;
GAP Ontology: 218 CDs with 2996 entries.
data.mathhub.info: six new datasets with together about 11M mathemtical objects with between 6 and 16 computed properties. Provenance sketches for all of them, one formalized in MMT.

[5] Percentage of code covered by tests

	2016–2017	2017–2018	2018–2019	Total	
North America	1,504	3,638	8,069	13,211	53.2%
Europe	936	2,745	5,354	9,035	36.4%
Asia	98	353	1,244	1,682	6.8%
Unknown	85	145	297	540	2.2%
South America	11	43	211	265	1.1%
Oceania	4	29	53	86	0.3%
Africa		12	18	30	0.1%
Central America		3	1	4	0%
Total	2,638	2,795	6,968	15,247	24,853

TABLE 1. Visits to `https://opendreamkit.org/`, by region and by year.

7.4. KPIs for Aim 4

Recommendation 14. *Aim 4 Maximise sustainability and impact in mathematics, neighbouring fields, and scientific computing.*

7.4.1. OpenDreamKit's web site statistics

- 6 video interviews and explainer material totaling 400 views
- 6 press releases
- 35 blogs (32 blogs and 3 technical blogposts)
- 517 Twitter followers, 1862 tweets
- Official website visitor statistics have been collected since March 2017, the visitor breakdown by year and country is reported in Table 1.

7.4.2. Workshops, conferences, events

An important part of the success of the ODK project is linked to its ability to foster a community in the spirit of the open source projects it is built on. Part of this relies on the organization and participation to scientific and development events of many different scales and objectives.

Over the last four years OpenDreamKit has organized or coorganized 110 events:

- 5 project meetings
- 24 development workshops
- 45 training and building community workshops or sessions (including 5 targeted at women and 10 in developing countries) adding 1800 trainees
- 7 research workshops

In addition, it communicated at and participated to 30 external events.

7.4.3. Diversity success stories

(1) **Women in SAGE workshops** blog post
The under-representation of women in the scientific world is even more visible if we intersect science with software development. During the four years of the projects we have organzided 5 events primarily targeted at women as to help form a team of women developers and to reduce the gap between men and women developers in Sage Math in particular and in open scientific software in general. Over 290 women received training.

On January 2017, Viviane Pons, Jessica Striker and Jennifer Balakrishnan organized the first WomenInSage event in Europe with OpenDreamKit. 20 women spent a week together coding and learning in a rented house in the Paris area. We took advantage of the diverse knowledge background of our group to work together and learn from each other. It was an occasion for many ”first times” among participants who had very little experience with Sage.

This sparked the organization of another Women in Sage workshop in Montreal in June 2018. OpenDreamKit organized a new one in April 2019 in Crete. It was co-organized by Eleni Tzanaki who had been a participant in the 2017 Women in Sage event. It was also an occasion to organize SAGE sessions at University of Crete and to add SAGE to the math curriculum.

(2) **Women in computing**
In partnership with CodeFirstGirls, OpenDreamKit developed training materials and provided training for over 130 women during Year 3 at Sheffield and Manchester

(3) **ODK RSE Tania Allard was invited by NumFocus to participate in the Diversity and Inclusion in Scientific Computing unconference**

(4) **ODK RSE Tania Allard diversity chair for the 2017 International Research Software Engineering conference.**

(5) **Training workshops in developing countries**
10 training workshops were organized in developing countries including Algeria, Lebanon, Tunisia, Morocco, Colombia, Mexico, Nigeria and attended by about 460 trainees.

The story of our event in Nigeria is particularly telling of the dissemination work we do. Indeed, we first met Ini Adinya from University of Ibadan at the CIRM conference *Free Computational Mathematics* organized by OpenDreamKit in February 2019. She was there along with two other Nigerian mathematicians who had received funding from OpenDreamKit to attend. They were thrilled by the conference and we decided to organize a SAGE workshop at Ibadan the following summer where we welcomed 80 participants mostly from Nigeria and neighboring countries.

The Workshop was run by Erik Madisson Bray from OpenDreamKit along with Yaé Ulrich Gaba, Evans Doe Ocansey, and Dr. Chimere Stanley Anabanti. The three extra instructors had been found through SAGE mailing lists and community. They had prior experience with teaching SAGE in Africa but, to our knowledge, it was the first time that such an event was held in Nigeria. The workshop itself was a technical challenge with unreliable network and power outages but the impact on the attendees was undeniable. Actually, the most frequent feedback we got on our questionnaire was that the workshop should have lasted more than a week! You can read more about the workshop on our blogpost here: `https://opendreamkit.org/2019/07/29/SageDays102/`.

7.4.4. Adoption of ODK technologies for teaching

We collect some success stories and statistics on courses and departments we worked with directly and an estimate of how many students this subsequently affected.
[8] EdN:8

(1) **African Institute for Mathematical sciences**
In early spring 2017, Prof. Dr. W. Decker and Prof. Dr. G. Pfister gave a three-week course on computational algebraic geometry at the African Institute for Mathematical sciences (Cape Town, South Africa) with lectures and computer lab sessions. The course was attended by about 50 students from all over Africa. In the lab sessions, the students learned how to experiment with the computer algebra system Singular. It proved extremely valuable that the students could run Singular in the Jupyter notebook.

(2) **PGTC "Software tools for mathematics" workshop 2018**
Alexander Konovolov use GAP Jupyter interface in teaching Reproducible GAP experiments on Binder.

[8] EDNOTE: @nthiery, @VivianePons: resolve redundancy between the list of courses here and in D2.17. Maybe mention here only courses not delivered by ODK as success stories + statistics on courses delivered by ODK members (number of students, levels, domains)

(3) **University of Sheffield**
Cocalc was adopted for teaching in UK after ODK's Kickoff: Support was given to a number of lecturers in Sheffield to migrate to Jupyter and CoCalc (formerly SageMathCloud) but also to those that had already been using CoCalc and Jupyter notebooks for their courses (D2.17, T2.6):
- Courses in physics: 500 students yearly.
- Courses inBiomedical Sciences
- Courses inComputer Science
- Courses inBioinformatics
- Courses inMaterials Science

In addition, a previously generated CoCalc tutorial was extended by adding tutorial sections for students having courses in CoCalc as well as with a hands-on tutorial for lecturers to get started. The material can be found as a website at `https://tutorial.cocalc.com/`.

(4) **Université Paris Sud**
Since 2017, Jupyter is used at Université Paris Sud for teaching C++ to over 400 students.This was initiated in particular by OpenDreamKit participants Loïc Gouarin, Viviane Pons, and Nicolas M. Thiéry. The mix of narrative documents and interactive programming fostered active participation from the students while our web-based deployment made it easier for them to work from home.The course material is available from `http://Nicolas.Thiery.name/Enseignement/Info111`.

In addition, ODK participants were involved in the following Jupyter-based courses:
- Projet Math-info (Bachelor). Tools: SAGE. 40 students yearly.
- Algèbre et Calcul Formel (Master). Tools: SAGE. 25 students yearly.
- Algorithmique (Engineers). 20 students yearly.
- Algorithmique (Master). 20 students yearly.
- Combinatorics (Master). 20 students yearly.

(5) **University of Granada**
The GAP Jupyter kernel was used by Pedro Garcia-Sanchez to teach a master course in mathematical software at the University of Granada. See `https://github.com/pedritomelenas/Software-Matematicas-GAP`. Pedro has taken on the technology, and is now involved in the development of interactive visualization widgets for discrete maths (package Francy; see also D4.16: "Exploratory support for semantic-aware interactive widgets providing views on objects represented and or in databases") in particular for use in other courses.

(6) **University of Ghent**
- Courses in mathematics (Bachelor, Master). Tools: CoCalc, Jupyter, SageMath. 31 students yearly.
- Courses in chemistry (Master). Tools: SageMath. 37 students yearly.

(7) **FAU Erlangen-Nürnberg**
- Logic-based Natural Language Semantics. Tools: MMT, domain modeling and reasoning system. 5-10 students yearly.
- Knowledge Representation for Mathematics and Technology. Tools: MMT, domain modeling and reasoning system. 5-10 students yearly.

(8) **Université Versailles Saint Quentin**
- Algèbre Commutative et Effective (M). Tools: Jupyterhub, SageMath, Planetaryum. 22 students yearly.
- Algorithmique et Programmation C (M). Tools: JupyterHub, JupyterLab, C. 22 students yearly
- Introduction au calcul formel (M). Tools: Jupyterhub, SageMath. 20 students yearly

- Algorithmique (M). Tools: Jupyterhub, JupyterLab, Python. 20 students yearly.
- Database (M). Tools: JupyterHub, Apache Spark. 60 students yearly.

(9) **Bioinformatics Awareness Days**
An event that took place in The Sheffield Institute for Translational Neuroscience on November 2017 where were given training activities about bioinformatics workflows using Jupyter notebooks with computation provided by the free Micosost Azure Notebook service.The event demonstrated that OpenDreamKit supported technologies could be applied to the field of Bioinformatics and led to a new collaboration between Dr Cutillo and OpenDreamKit member Mike Croucher.

Following the success of this workshop, Dr Cutillo independently taught an introductory workshop on statistics using Jupyter notebooks on Azure at Parthenope University of Naples (Materials at `https://github.com/luisacutillo78/RbasicStats`) r Cutillo has since moved to University of Leeds where she will be teaching statistics to 200+ undergraduates. She plans to use OpenDreamKit developed technologies in collaboration with the Research Software Engineering group at Leeds.

The event required also the development of a website that was linked to the Jupyter notebooks (`https://bitsandchips.me/BAD_days/`). The website caught the attention of Eleni Vasilaki, Head of Machine Learning at University of Sheffield who wanted to do something similar for her course on Adaptive Intelligence. We supported her in this endeavour and the result is at (`http://bitsandchips.me/COM3240_Adaptive_Intelligence/`).

In order to better support this, OpenDreamKit member Tania Allard, developed a Jekyll template for use by academics and researchers using Jupyter notebooks for course materials and dissemination. This led to the development of a Python package: nbjekyll (`https://github.com/trallard/nbjekyll`) that complements the Jekyll template.As well as being used internally at Sheffield, The nbjekyll package received some attention on twitter `https://twitter.com/jdblischak/status/1009800776305332224` and `https://twitter.com/walkingrandomly/status/1009414151716909057` receiving a total of 42 retweets and 80 'likes'

7.4.5. Impact of some tools developed by OpenDreamKit

Measuring success of software can be a challenge. One way to measure is to observe metrics of engagement on a public development platform such as GitHub. A 'star' means that an individual is following development of the project. Issues are used to discuss the project, e.g. to report bugs, ask questions, request features, etc.. Comments indicate how much people are discussing the project, and the number of authors of commits and discussion comments indicates how broadly we are reaching. Note that the number of participants in development discussion tends to be a very small fraction of total users, which is difficult to estimate. To compare, the Jupyter project estimates millions of users, while the flagship notebook repository sees only thousands of participants on the repository in the last twelve months.

We report success for some OpenDreamKit components related to WorkPackage 4, showing significant community engagement, comparing the results to the previous twelve months, showing an increase in popularity over time.

(1) **nbdime**: Tool for diffing Jupyter notebooks
GitHub statistics: 1336 stars on GitHub, 61 contributors (45 in 12 months prior), 388 comments, 107 new issues (94 closed). Delivered in the first period (D4.6), NBDIME has been met with enthusiasm and widely adopted. This led to further developments, including integration in the Jupyter Notebook, Jupyter Lab, and the version control git through extensions.

(2) **Thebelab**: Jupyter-based javascript library for integrating live code in static web pages
GitHub statistics: 96 stars, 28 contributors (15 in 12 months prior), 124 new issues (126 closed), 292 comments.
Thebelab has been integrated into documentation tools such as jupyter-sphinx plugin and the documentation of several software packages, including some SAGEMATH documentation and other software beyond the OpenDreamKit consortium.

(3) **K3D-Jupyter**: 3D visualisation in the Jupyter notebook.
GitHub statistics: 131 stars, 14 contributors, 140 new issues (129 closed), 303 comments.
This visualization tool is now used by OpenDreamKit component ubermag for interactive visualisation in Jupyter notebooks.

References

[BKR19] Katja Berčič, Michael Kohlhase, and Florian Rabe. "Towards a Unified Mathematical Data Infrastructure: Database and Interface Generation". In: *Intelligent Computer Mathematics (CICM) 2019*. Conferences on Intelligent Computer Mathematics. Ed. by Cezary Kaliszyck et al. LNAI 11617. Springer, 2019, pp. 28–43. DOI: 10.1007/978-3-030-23250-4. URL: https://kwarc.info/kohlhase/papers/cicm19-MDH.pdf.

[BKS17] Johannes Blömer, Temur Kutsia, and Dimitris Simos, eds. *MACIS 2017*. LNCS 10693. Springer Verlag, 2017.

[Car+19] Jacques Carette et al. "Big Math and the One-Brain Barrier – A Position Paper and Architecture Proposal". accepted at the Mathematical Intelligencer. 2019. URL: https://arxiv.org/abs/1904.10405.

[D6.1019] Katja Berčič et al. *Report on OpenDreamKit deliverable D6.10: Full-text search (Formulae + Keywords) in OpenDreamKit*. Deliverable D6.10. OpenDreamKit, 2019. URL: https://github.com/OpenDreamKit/OpenDreamKit/raw/master/WP6/D6.10/report-final.pdf.

[D6.116] Michael Kohlhase and Alexandru Glontaru. *Full-text Search (Formulae + Keywords) over LaTeX-based Documents*. Deliverable D6.1. OpenDreamKit, 2016. URL: https://github.com/OpenDreamKit/OpenDreamKit/raw/master/WP6/D6.1/report-final.pdf.

[D6.518] John Cremona et al. *Report on OpenDreamKit deliverable D6.5: GAP/SAGE/LMFDB Interface Theories and alignment in OMDoc/MMT for System Interoperability*. Deliverable D6.5. OpenDreamKit, 2018. URL: https://github.com/OpenDreamKit/OpenDreamKit/raw/master/WP6/D6.5/report-final.pdf.

[D6.818] John Cremona et al. *Report on OpenDreamKit deliverable D6.8: GCurated Math-in-the-Middle Ontology and Alignments for GAP/SAGE/LMFDB*. Deliverable D6.8. OpenDreamKit, 2018. URL: https://github.com/OpenDreamKit/OpenDreamKit/raw/master/WP6/D6.8/report-final.pdf.

[Deh+16] Paul-Olivier Dehaye et al. "Interoperability in the OpenDreamKit Project: The Math-in-the-Middle Approach". In: *Intelligent Computer Mathematics 2016*. Conferences on Intelligent Computer Mathematics (Bialystok, Poland, July 25, 2016–July 29, 2016). Ed. by Michael Kohlhase et al. LNAI 9791. Springer, 2016. ISBN: 978-3-319-08434-3. URL: https://github.com/OpenDreamKit/OpenDreamKit/blob/master/WP6/CICM2016/published.pdf.

[HR09] Peter Horn and Dan Roozemond. "OpenMath in SCIEnce: SCSCP and POPCORN". In: *MKM/Calculemus Proceedings*. Ed. by Jacques Carette et al. LNAI 5625. Springer Verlag, July 2009, pp. 474–479. ISBN: 978-3-642-02613-3.

[Isa] *Isabelle*. Mar. 9, 2013. URL: http://isabelle.in.tum.de (visited on 03/27/2013).

[Koh+17] Michael Kohlhase et al. "Knowledge-Based Interoperability for Mathematical Software Systems". In: *MACIS 2017: Seventh International Conference on Mathematical Aspects of Computer and Information Sciences*. Ed. by Johannes Blömer, Temur Kutsia, and Dimitris Simos. LNCS 10693. Springer Verlag, 2017, pp. 195–210. URL: https://github.com/OpenDreamKit/OpenDreamKit/blob/master/WP6/MACIS17-interop/crc.pdf.

[Qed] *The QED Project*. http://www-unix.mcs.anl.gov/qed/. 1996. URL: http://www-unix.mcs.anl.gov/qed/.

[WDM] *Workshop on Data in Mathematics*. URL: https://opendreamkit.org/2019/08/17/WorkshopOnDataInMathematics/ (visited on 08/27/2019).

[WKR17] Tom Wiesing, Michael Kohlhase, and Florian Rabe. “Virtual Theories – A Uniform Interface to Mathematical Knowledge Bases”. In: *MACIS 2017: Seventh International Conference on Mathematical Aspects of Computer and Information Sciences*. Ed. by Johannes Blömer, Temur Kutsia, and Dimitris Simos. LNCS 10693. Springer Verlag, 2017, pp. 243–257. URL: `https://github.com/OpenDreamKit/OpenDreamKit/blob/master/WP6/MACIS17-vt/crc.pdf`.