REPORT ON OpenDreamKit DELIVERABLE D1.5

Internal Progress Reports year 2 and 3, including risk management and quality assurance plan

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

Due on	31/08/2018 (M36)
Delivered on	17/10/2018
Lead	Université Paris-Sud (UPSud)
Progress on and finalization of this deliverable has been tracked publicly at: `https://github.com/OpenDreamKit/OpenDreamKit/issues/21`	

DELIVERABLE DESCRIPTION, AS TAKEN FROM GITHUB ISSUE #21 ON 2018-10-17

- **WP1:** Project Management
- **Lead Institution:** Université Paris-Sud
- **Due:** 2018-08-31 (month 36)
- **Nature:** Report
- **Proposal:** p. 34
- **Final report**, sources

This document is meant as an internal draft for the project's Technical Report for Reporting Period 2. As such, it provides an overview of the project achievements, by objectives, work packages, and tasks from March 2017 to August 2018, and their impact. It also explains how reviewer's recommendations were tackled and discusses risk and quality management and deviations from the original plan. We recommend to read directly the technical report itself.

CONTENTS

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 second reporting period, ranging from March 2017 to August 2018. We start by recalling some context of OpenDreamKit's approach that is important to understand and evaluate the progress. Then we describe the general state, and in more detail the progress of the work packages.

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 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.

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.

[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.

The first and second reporting period 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 makes 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. 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.

1.1. Explanation of work carried out per Objective

For reference, let us recall the aims of OpenDreamKit.

A1: Improve the productivity of researchers in pure mathematics and applications by promoting collaborations based on mathematical **software**, **data**, and **knowledge**.

A2: 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.

A3: 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.

A4: 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.

O1: "*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.*"

This objective is by nature multilevel; achievements include:

- Collaborative workspaces: major JUPYTERHUB developments, see **T4.2**: "Notebook improvements for collaboration"; study and documentation of the SAGEMATHCLOUD architecture, see **T3.6**: "Document and modularise SAGEMATHCLOUD's codebase";
- 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.

O2: "*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. New software components includes the tools nbmerge, nbdiff and nbval (see D4.8 and D4.6), and planetaryum (see D2.7). For the Math-in-the-Middle paradigm for semantic system interoperability we have developed knowledge-based Mediator based on the MMT system.

O3: "*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, SAGEMATHCLOUD, and SIMULAGORA; see **T3.6**: "Document and modularise SAGEMATHCLOUD's codebase", **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");
- Sharing of best practices when using VRE's like COCALC or JUPYTER for research and education;
- 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".

O4: "*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 SAGE have been put for
  - the linear algebra computational kernels of LinBox, fflas-ffpack and Givaro (Deliverable D5.12: "Exact linear algebra algorithms and implementations. Library maintenance and close integration in mathematical software for LINBOX library")
  - 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." still ongoing)
- Packaging efforts: docker containers (delivered and regularly updated), Debian and Conda packages (beta); see **T3.3**: "Modularisation and packaging".
- Continued efforts on portability of SAGE and its dependencies (see **T3.1**: "Portability", in particular D3.7).
- Improved continuous integration and development workflow; (see **T3.7**: "Improving the development workflow in mathematical software"), and D3.8: "Continuous integration platform for multi-platform build/test.".
- 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".
- Ongoing work in WP5 to better support HPC in the individual mathematical software system and combinations thereof; see Section 1.2.5.

O5: "*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; 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".

O6: "*Promote collaborative mathematics and science by exploring the social phenomena that underpin these endeavours: how do researchers collaborate in Mathematics and Computational Sciences? What can be the role of VRE? How can collaborators within a VRE be credited and incentivised? This addresses parts of Aims 3, 1, and 2.*"

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.

O7: "*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 WP6 we have developed the Math-in-the-Middle ontology that acts as the pivot point mediating between system languages in the MitM interoperability framework. The work has been reported in D6.8.

O8: "*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.*"

Most of the work toward this objective is by nature planned for the last period of the OpenDreamKit project. Nevertheless, work has started e.g. toward the OOMMF demonstrator; see **T2.7**: "Open source dissemination of micromagnetic VRE" **T2.8**: "Micromagnetic VRE dissemination workshops", **T3.8**: "Python interface for OOMMF micromagnetic simulation library".

O9: "*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.*"

This objective is at the core of WP2, with in particular more than 30 meetings, developer, training, and community building workshops organized during the second reporting period. See Section 1.2.2 and D2.11: "Community building: Impact of development workshops, dissemination and training activities, year 2 and 3" for details.

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.

The general objectives of Work Package 1 are:

(1) Meeting the objectives of the project within the agreed budget and timeframe and carrying out control of the milestones and deliverables
(2) Ensure all the risks jeopardising the success of the projects are managed and that the final results are of good quality
(3) Ensuring the innovation process within the project is fully aligned with the objectives set up in the Grant agreement

WP1 has been divided into three tasks. In the following, we highlight the key achievement in this work package, when necessary report specifically on these individual tasks.

- UPSud lead the 2nd, 3rd and 4th amendment to the consortium, in particular to cater for the moving to other institutions of permanent and/or non-permanent researchers who are key personnel for the success of OpenDreamKit. A collateral effect is the addition of the sites , FAU, XFEL, and termination of the sites USFD, SOUTHAMPTON, JacobsUni, UZH since no relevant staff for OpenDreamKit remained at these institution.
- Following the recommendations from the reviewers after the first reporting period, UPSud also lead the work plan revision process that were implemented in the 3rd amendment.
- UPSud organized the first project review in Brussels with the Funding Agency, and regular project meetings, including three steering committee meetings in Brussels (March 2017), online (February 2018), and at XFEL (June 2018).
- UPSud made sure that all the milestones and deliverables of the Second Reporting Period were achieved and reported on within its timeframe.
- Concerning the communication, UPSud has been maintaining the internal communication tools that were described in D1.1: “Basic project infrastructure (websites, wikis, issue trackers, mailing lists, repositories)”. As for external communication the website for the project has been 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). A new version of the website was released in June 2018. Its end-user friendly interface and content makes it a tool not only for internal communication but very much for dissemination and progress tracking by the reviewers and the community.
- Concerning the future of OpenDreamKit and of its infrastructure toolkit, the coordinator took action to access information and get involved in the development of the European Open Science Cloud that is currently promoted by the European Commission. We took advantage of the EOSC stakeholder forum on 28-29 November and the 2017 edition of the DI4R (Digital Infrastructures for Research) in Brussels. During these events we gathered information on the potential of EOSC and how OpenDreamKit could fit in there. Furthermore we initiated a partnership with EGI – a key participant to EOSC – to deploy OpenDreamKit based infrastructure.
- The continued interaction with an Advisory Board and a Quality Review Board to control the quality and the relevance of the software development relative to the end-user needs. See below.
- UPSud produced a second version of the Data Management Plan (D1.6).
- Finally UPSud kept an eye on recruitment: the strategies we used (tailoring of the positions according to the known pool of potential candidates, in particular among previous related projects, strong advertisement, ...) seem to have paid off, and we are

really happy with the top notch quality of our recruits. However, despite many steps to foster women applications to apply (e.g. through reaching personally toward potential candidates or including women in the committees), we had almost no female candidate, and none made it to the short list. This is alas unsurprising in the very tight segment of experienced research software engineers for mathematics on temporary positions which is highly gender imbalanced; this is nevertheless a failure.

1.2.1.2. **Tasks.**
T1.1: "Project and financial management". As planned in WP1, UPSud has been coordinating OpenDreamKit and 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 and funds transfer to cater for the moving of personnel across sites and from old sites to new sites.
T1.2: "Quality assurance and risk management".

(1) Continued success in the recruitment of highly qualified staff.

The Quality Assurance Plan is described in detail in D1.3: "Internal Progress Reports year 1, including risk management and quality assurance plan". We will describe the main points below. UPSud had launched during RP1 a Quality Review Board which is chaired by Hans Fangohr. The four members of the board have a track record of caring about the quality of software in computational science. This board 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. The board meets after the end of each Reporting Period (RP), and before the Review following that RP. More details about the Quality Review Board are given in Section 4.3, including a summary of the recommendations it emitted after the first reporting period.

The other structure supporting OpenDreamKit to ensure the quality of the infrastructure it produce is the Advisory Board. It is 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. After the first reporting period, UPSud led the review by the Advisory Board of our Technical Report and later Work Plan Revision Proposal. This feedback, complementing that of the reviewers, was very helpful to orient the final work plan revisions.

UPSud has also been managing risks. In D1.3 all potential risks had been assessed by the Coordinator at Month 12. Here is a brief update on Risk 1 concerning the recruitment of highly qualified staff. This risk has been globally well managed thanks to a flexible workplan enabling adjustments in the timing of some tasks or deliverables, and thanks to legal actions taken by the Coordinator to allow key personnel, permanent or not, to remain in the Consortium even though their positions changed. The addition of the three partners is representative of these actions. The assessment for the other risks remain valid at Month 36, and we refer to D1.3 for details.
T1.3: "Innovation management". D1.4: "Innovation Management Plan v1" was produced at month 18, with a focus 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 VRE.

The second version of the Innovation Management Plan in RP3 will add content to explain all the innovations that the VRE is bringing to end-users. However the open source approach and the ”by users for users” development process will not change.

One of the assessed risks for 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 programme they were initially working on and for. This risk was continuously tackled by the Coordinator in order to reach the final goals of the VRE which are the unification of open source tools with overlapping functionality, the simplification of the tools for end-users without coding expertise, and the development of user-friendly interfaces. For this, the Scientific Coordinator is for example wilfully pushing for joint actions and workshops.

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 progressively becoming a standard practice, and not only for OpenDreamKit participants. More information on joint workshops can be found in the next section.

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

1.2.2.1. **Overview.** We have continued the work started in period one, especially on **T2.1**: "Dissemination and Communication activities" and **T2.5**: "Dissemination: reaching towards users and fostering diversity", organizing or participating in more than 50 events throughout the last two years. In particular, we have intensified our efforts in **training** the community to use the tools developed through OpenDreamKit. Indeed we have had 14 workshops or events directly organized by OpenDreamKit on subjects such as Jupyter, JOOMF (see **T2.7**: "Open source dissemination of micromagnetic VRE"), PARI/GP, and more. On top of that, we were also part of 5 SageDays.

One of our priorities is to increase the diversity of the open-source community in science. We have been organizing and supporting initiatives to support women developers and scientists such as: Women in Sage, Code First: Girl, and PyLadies. We have also organized specific events in developing countries: Colombia, Mexico, and Morocco.

Finally, this period was also an occasion to improve our online communication, following the advice of our last review. Indeed, we collaborated with students in communication to design a new website. We have also conducted interviews to explain the key points of the project and we are working on some multimedia content.

1.2.2.2. **Tasks.**

T2.1: "Dissemination and Communication activities". We have followed the advice from our reviewers and have worked at a better communication:

- We have worked in collaboration with a master degree in web communication and design. The OpenDreamKit website was the year project of a team of three students who delivered different reports to us, some new communication ideas and a whole new website design. We implemented the new design ourselves and organized a small workshop to think as a group about the website organization and content.
- To reach a wider audience, we decided to use different media for communication. We hired an interviewer and created short videos to share with our views on project's goals and achievement with our communities. These video are in final stage of edition and will be released shortly and made available on the website.
- We have worked with (motion) graphic designers to create infographic content for our project. The infography gives a clear and fast way to understand the tools we are developing. A first sketch illustrating one of our use cases will be posted on the web site by the review. More sketches will be posted later this fall and will serve as a basis for an explainer video to be produced by PixVideos over the winter.

T2.2: "Training and training portal". Training is a core and transversal aspect of our project. It is carried out through interventions and events as we discuss in **T2.5**: "Dissemination: reaching towards users and fostering diversity". These past two years especially, there is been a very strong effort from the OpenDreamKit team to organize training events and workshops. We are also working on creating better content on our webpage to help users understand in what aspect of their work can OpenDreamKit help. This is why we have create the "Use Cases" section.

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. As reported in D2.11: "Community building: Impact of development workshops, dissemination and training activities, year 2 and 3", we have organized or co-organized 12 of these workshops throughout years 2 and 3 of the project. The thematics varies for each event: PARI/GP, Linbox,

and many cross-thematic events such as GAP-Sage and GAP-Jupyter days, live structured documents, low level libraries, and more.

T2.4: "Reviewing emerging technologies". This task has been started during period one, especially through D2.3. We continue to keep track of new technologies and report by writing blog posts on our website.

T2.5: "Dissemination: reaching towards users and fostering diversity". Dissemination is a key aspect of the success of OpenDreamKit. Indeed, our development is carried out to help and support mathematical communities. One of the goals is to bring more users and more developers to the different projects we are involved in. The events that took place during Years 2 and 3 have been reported in D2.11: "Community building: Impact of development workshops, dissemination and training activities, year 2 and 3".

- **Training workshops and events.** This has been the most important aspect of this task for the past two years. We have been organizing 14 events covering subjects such as: Jupyter, JOOMMF, Bioinformatics, HPC, PARI/GP, GAP, experimental mathematics, web data, reproducible workflows.
- **Organization of Sage Days in established mathematical communities.** Sage Days have long been part of the SageMath tradition. By organizing and supporting Sage Days, OpenDreamKit can stay close the mathematical community, understand its needs, gather more users and developers, and improve the overall quality of the software. We have been involved in 5 different such events since the beginning of the project.
- **Training activities in developing countries.** OpenDreamKit has been present in Colombia for the second time at the conference ECCO. We also organized SageDays workshops in Mexico and Algeria as well as a PARI/GP event in Morocco.
- **Women in OpenDreamKit.** Following the organization of Women in Sage in January 2017, two female OpenDreamKit participants have given time and energy to this specific topic. Viviane Pons organized the Women in Sage event, she has also been an organizer of the *Pyladies Paris* chapter for the last two years. Another Women in Sage is planned for summer 2019. Tania Allard was a research software engineer for OpenDreamKit until July 2018, she worked at the *Code First: Girl* chapter in Sheffield, and was also invited to the *Diversity and Inclusion in Scientific Computing* event in 2018.

T2.6: "Introduce OpenDreamKit to Researchers and Teachers". Some work has been carried out on this topic by University of Sheffield and University of Leeds which can be seen by the many training activities and wokshops that have been organized on various topics (bioinformatic, HPC, Web data). The goal is to introduce the Jupyter notebook to researchers and teachers of various scientific fields. A frame work has also been built to allow teachers to easily create a course website based on Jupyer.

Besides, the University Paris-Sud has been experimenting teaching C++ programming using Jupyter on a big scale course of 400 students. They have been working closely with the developers of the C++ kernel and have been trying different technical solutions which will be documented in a blogpost.

T2.7: "Open source dissemination of micromagnetic VRE". This task was mostly carried out during the first period. The JOOMF project is working and available on GitHub (JOOMMF repo).For each JOOMMF package we use continuous integration on Travis CI 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. We encourage the early use of our software and invite for feedback for which we provide several different communication channels: Google group (joommf-news), Gitter channel, GitHub help repository, Twitter account, and a website.

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". Two books, *Linear Algebra and its Applications* (lectures for physicists) and *Nonlinear process in Biology* D2.8 have been delivered as planned. The source code can be found on our GitHub repo. We will keep updating and improving the content over time.

T2.10: "Demonstrator: Computational mathematics resources indexing service". The web toolkit *planetaryum* has been delivered D2.7. The joint development efforts under OpenDreamKit, and in particular the unifying force of the common Jupyter notebook infrastructure, have given a unique occasion to produce a unified solution applicable to all systems. Following the OpenDreamKit philosophy, planetaryum is not a single piece of software, but rather a toolkit, meant to power many different flavors of curated collections. The source code can be found on our GitHub repo.

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 softwares interacting through library calls and/or through APIs.

The latest reporting period has focused mainly on improving development workflows and user experience, in particular targeting notoriously "difficult" platforms such as Windows.
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."*

With the completion of D3.7: "One-click install SAGE distribution for Windows with Cygwin 32bits and 64bits", all OpenDreamKit components now run on Windows. Packages for the major Linux distributions (Debian, Ubuntu, Fedora, Arch, ...) have also been available for at least a year, thanks to the efforts of the community[3]. MacOS binaries are regularly released, albeit with the usual hiccups typical of the Apple ecosystem.

The bulk of the milestone is thus completed, ahead of schedule, although there is still work to do, the most important items on the agenda being better continuous integration, and support for Python 3 in SageMath[4]. We will focus on these items for the remaining year.

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

The most challenging target is the Windows platform, and indeed SageMath has not had native Window support for years. With the completion of D3.7: "One-click install SAGE distribution for Windows with Cygwin 32bits and 64bits", we are happy to announce Windows support for SageMath: since version 8.0, released in July 2017, a one-click installer based on Cygwin is the recommended way to install SageMath on Windows. With this deliverable, we achieved Windows support for 100% of OpenDreamKit's components.

In support of developing and maintaining OpenDreamKit's software on all platforms, we have also worked on infrastructures for continuous integration. No unique solution was found that could accommodate the needs of every project, however, through sharing information and experience returns, each of the software projects inside OpenDreamKit has managed to put in place the infrastructure better suited for its needs, leveraging various popular technologies such as Docker, Jenkins, etc. These efforts have been reported in D3.8: "Continuous integration platform for multi-platform build/test.".
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, through Symbolic Computation Software Composability Protocol (SCSCP). All subsequent planned work has been moved to WP6.

[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. We will keep monitoring the status of Linux packages, prioritizing more popular distributions such as Ubuntu, and continue our efforts to make our components easy to package.

[4]Python 3 support is vital for SageMath going into the year 2020, when Python 2 will be officially deprecated.

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.

The main deliverable in this task is D3.10, due in month 48. Thanks to the joint efforts of OpenDreamKit and of the community, SageMath is now available as a Debian package, and recently also as a Conda package.

This task is progressing as planned, and we expect to successfully complete it next year.

T3.4: "Simulagora integration". The goal of this task is to deliver every six months a new Simulagora VM image containing all the software components released over the period.

To this date, five OpenDreamKit VMs have been released in Simulagora. The latest version, released in March 2018, showcases virtual desktops available from a web browser and collaboration workflows based on "tools" that can be described as micro web applications that require very little development skills to set up, but make it easy to make available complex simulations to users.

T3.5: "Component architecture for High Performance Computing and Parallelism". Not applicable for this period.

T3.6: "Document and modularise SAGEMATHCLOUD's codebase". Recall COCALC used to be called SAGEMATHCLOUD at the beginning of this project. This task has been terminated early due to the cancellation of D3.4: "*Personal* SAGEMATHCLOUD: single user version of SAGEMATHCLOUD distributed with SAGE.", achieved by the COCALC developers *before the start of OpenDreamKit*.

The resources planned for this task were diverted to other work packages.

T3.7: "Improving the development workflow in mathematical software". This task seeks new ways of accepting contributions to mathematical software in a scalable way.

Deliverable D3.5: "Integration between SAGEMATHCLOUD and SAGE's TRAC server" was considerably reshaped to take into account the recent developments in the ecosystem. This caused a one year delay in the delivery.

Thanks to the work done, the entry barrier for developing SageMath has been considerably lowered. It is now possible for users with a GitHub or GitLab account to contribute to SageMath without having to go through a manual (and slow) registration process, and editing documentation is now easier then ever, even for the inexperienced user.

With the delivery of D3.5, this task is now complete.

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**)

Progress across WorkPackage 4 has been highly successful thus far. Several new software packages have been created, and existing projects in the SAGE and JUPYTER communities have been improved toward sustainability to serve OpenDreamKit objectives.
Milestones.
M6: "Prototype VRE for mathematical researchers" (month 36). *"User story: a group of mathematical researchers with access to common computational resources, such as a shared lab computer or cloud servers, shall be able to deploy a prototype VRE with* JUPYTERHUB*, integrating OpenDreamKit components. The Jupyter kernels for mathematical software developed as part of OpenDreamKit make computational mathematical components accessible in a* JUPYTER *environment, enabling a Jupyter-based deployment of the relevant tools for the researchers. The process of working on notebooks is greatly improved by review tools developed as part of WP4, enabling researchers to collaborate to some degree in a shared computational environment."*

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 **M6**: "Prototype VRE for mathematical researchers" (month 36) has been reached: from the obtained toolkit, we can produce a JUPYTER-based prototype 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**. 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".
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 is delivered in **M6**: "Prototype VRE for mathematical researchers" (month 36). **T4.8**: "Visualisation system for 3D data in web-notebook" adds interactive visualization to this environment, expanding what is possible in an interactive JUPYTER environment. We are on track to deliver real-time collaboration to this VRE via JupyterLab in **T4.2**: "Notebook improvements for collaboration" via D4.15: "Exploratory support for live

notebook collaboration". The final stage of taking work performed in this VRE into publication / knowledge system is enabled by **T4.6**: "Structured documents" and WorkPackage 6.

1.2.4.2. **Tasks.**

T4.1: "Uniform notebook interface for all interactive components". The first task for this workpackage is to enable the use of JUPYTER as uniform notebook interface for the relevant computational components. D4.4 has been delivered, providing basic JUPYTER kernels for GAP, PARI, SAGE, and SINGULAR. After the basic implementations of these kernels, they have been developed toward stability and maturity, as part of delivering D4.7: "Full featured JUPYTER interface for GAP, PARI/GP, Singular" in the first reporting period.

D4.5: "SAGE notebook / JUPYTER notebook convergence" has been delivered in the second reporting period, further enhancing SAGE's JUPYTER integration and preparing for the systematic transition from the legacy custom-built SAGE notebook application to JUPYTER in the coming months. Beside all the benefits of a uniform and actively developed interface for the user, outsourcing the maintenance of this key but non disciplinary component will save the SAGE community much needed resources and is an important step toward the sustainability of the OpenDreamKit ecosystem (Objective 5).

Progress was particularly fast thanks to a very active involvement of the SAGE community.

T4.2: "Notebook improvements for collaboration". D4.6: "Tools for collaborating on notebooks via version-control" has been delivered in the form of a new JUPYTER package, `nbdime`, enabling easier collaboration on notebooks via version control systems such as Git. This project was presented at the major Scientific Python conferences SciPy US in July 2016 and EuroSciPy in August 2016, and has been met with enthusiasm from the scientific Python community for its prospect of solving a longstanding difficulty in working with notebooks.

The JUPYTERHUB package has received significant updates and further development, specifically a *Services extension point*, which enables shared workspaces for collaboration, a step on the path toward real-time collaboration for D4.15. Prototypes and design for D4.15 have been delivered, with a plan for full production implementation in collaboration with the JUPYTER community during the next year.

T4.3: "Reproducible Notebooks". D4.8: "Facilities for running notebooks as verification tests" has been 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**. 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". SAGE has a vast documentation; maintaining and further fostering high quality requires strong documentation tools.

All along the first two reporting periods, a great deal of work has invested toward the maintainability of the SAGE documentation build system, including improving reproducibility of builds, updating contents, and increasing reliance on community-standard tools instead of less maintainable bespoke implementations. This included significant upstream contributions to the Sphinx documentation system has been significantly improved, especially for Cython-generated packages, such as those in **T4.12**. The current state and future plans for the last year are described in details in D4.13: "Refactorisation of SAGE's SPHINX documentation system".

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

D4.16: "Exploratory support for semantic-aware interactive widgets providing views on objects represented and or in databases" was delivered in the form of two new packages developed by OpenDreamKit *Sage Combinat Widgets* and *Sage Explorer*. Both build on the robust foundation of Jupyter Widgets, and explore what it can bring to interactive mathematics. The former focuses on interactive visualization and edition of mathematical objects, taking combinatorics

and discrete math as use case. The latter, which uses the former as building block, provides rich, detailed, and efficient interactive exploration of objects, their properties and interrelations. Both are demonstrated online via the Binder service.

T4.6: "Structured documents". Active structured documents are a common need with many use cases, and has many potential solutions. Requirements and venues for collaborations were explored through discussions between participants, in particular at the occasion of Sage Days 77 workshop (see the notes), and June's ODK meeting in Bremen. The findings were reported in D4.2: "Active/Structured Documents Requirements and existing Solutions".

In D4.9: "In-place computation in active documents (context/computation)", We have presented a general framework for in-situ computation in active documents. This is a contribution towards using mathematical documents – the traditional form mathematicians interact with mathematical knowledge and computations – as a user interface for a mathematical virtual research environments. This is also a step towards integrating the two main UI frameworks under investigation in the OpenDreamKit project: JUPYTER notebooks and active documents – see D4.2: "Active/Structured Documents Requirements and existing Solutions" – at a conceptual level. The system is prototypical at the moment, but can already be embedded into active documents via a Javascript framework and is ready for use in the OpenDreamKit project. The user interface and SCSCP connections are quite fresh and need substantial testing and optimizations.

OpenDreamKit hosted a workshop on live structured documents in October 2017, which resulted in the development of thebelab software for interactive computing on any website, enabling interactivity in traditional web-based documentation, and further development of the MATHHUB facilities for evaluation in structured documents.

T4.7: "Active Documents Portal". 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". The current landscape for 3D visualization in Jupyter has been explored and reported on, in order to identify where OpenDreamKit can best contribute to 3D visualization in the notebook, towards D4.12: "JUPYTER extension for 3D visualisation, demonstrated with computational fluid dynamics". There have been many community-led developments in this area, including the ipyvolume and K3D packages, and OpenDreamKit will further enhance existing community tools to best serve the needs of the community. In particular, developing tools for unstructured mesh visualization and CFD simulations. The SciViJS tool for interactive visualization in a webbrowser has been developed, and has gained better support for use in JUPYTER, also as part of D4.12. Following initial investigations, OpenDreamKit has contributed to many core projects in the JUPYTER widgets and visualization ecosystems. Including to the core JupyterLab and jupyter-widgets projects, as well as the *pythreejs* package for exposing the *threejs* javascript library to Python kernels. Additionally, we have created new packages as building blocks for visualization tools, including

ipyscales and *ipydatawidgets*. We have also produced new visualization tools in the form of *unray* for visualising unstructured mesh data (e.g. 3D biological or mechanical systems).
T4.9: "Visualisation of 3D fluid dynamics data in web-notebook". The tools in D4.12 have been demonstrated as useful in a particular scientific domain, that of computational fluid dynamics. The *K3D-Jupyter* project, developed by OpenDreamKit, is used to illustrate visualising Lattice-Boltzmann simulations interactively in a browser without any performance penalty.
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 engine is the (existing) OOMMF (Object Oriented MicroMagnetic Framework) 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 problem specification, the execution of the calculation, 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, graphical representation of vector field objects, and GUI elements for data exploration. We have used dissemination workshops to seek feedback from users and to refine interface.
T4.12: "Python/Cython bindings for PARI". There has been a great deal of progress delivering improved PARI. This work has resulted in benefits to the wider Python and SAGE communities via substantial contributions to the SAGE codebase, the benefits of which go well beyond this deliverable, being used by projects outside OpenDreamKit.

The end results of this first state of the work are the packages cysignals and CyPari2, both installable in a pure PYTHON environment via the standard tool `pip`. Starting from version 8.0, installation via `pip` is SAGE's default way of providing the PARI interface.

D4.10: "Second version of the PARI Python/Cython bindings" has been delivered, further improving the PARI packages by adding new features, in particular to the Python interface to PARI. *cypari2* has gained the ability produce high-resolution SVG plots. It now also supports the dynamic array type from PARI/GP, `t_LIST`. The source code of cypari2 is automatically generated. This automatic generation has been greatly improved and can be re-used outside cypari2 for any Python package that wants to interface efficiently with PARI. The cypari2 documentation is also greatly improved, as a direct result of improvements to the Sphinx documentation system in **T4.4**.
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.** Workpackage 5 is about the development of high performance computing tools in mathematical virtual research environments. It is addressed at the level of each kernel library composing the computational tools of the project (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 period for WorkPackage 5 are the following:

- A full-featured parallelisation engine, supporting POSIX threads and MPI, for PARI in production release of the software
- Release of GAP-4.9 allowing compilation in HPC-GAP compatibility mode.
- A new symmetric matrix factorization algorithm over finite fields, and its high-performance implementation in the `fflas-ffpack` library.
- Major redesign the the polynomial arithmetic used in Singular delivering state of the art efficiency.

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."*

With Deliverable D5.12, we developped, released and integrated in the SAGE the LinBox library and its core dependencies: fflas-ffpack and givaro. When installing the latest SAGE release on a multithreaded multicore server, it only takes one configure option to let fflas-ffpack use a multi-threaded BLAS and therefore expose its parallel speed-up to the end-user of Sage. This feature is compliant with the use of a higher level of parallelism, through process workstealing queues that *Astrid* may be using in her combinatorics code, as those exposed in D5.11. Now that this first proof of concept has been successfully achieved, we are working in exposing the more advanced parallel routines of fflas-ffpack into SAGE, following D5.9. It should in particular make Gaussian elimination and related routines enjoy a better scaling with respect to available CPU cores.

1.2.5.2. **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)" has been 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.

Deliverable D5.16 is on time or even ahead of schedule. The generic engine for Deliverable D5.10 was written and finalized in 2015 and 2016 and is in production since PARI-2.9 (released 11/2016), it supports sequential evaluation (no parallelism), POSIX threads and MPI within the same code base. Development work since then is targeted at using the engine wherever it makes sense in the code base. The current situation (PARI-2.11, released 07/2018) includes:

- fast (near linear time) Chinese remaindering;
- fast linear algebra over Q and cyclotomic fields, a critical component of the new "Modular Forms" package;
- polynomial resultant in $\mathbb{Z}[X]$ via Chinese remainders;

- 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) - polynomial resultant in $\mathbb{Z}[X] \times \mathbb{Z}[X, Y]$ via Chinese remainders / evaluation;
- APRCL primality proof.

More work is underway regarding the development of ECPP, integer factorization engines, class group and units computations.

Compared to the pre-existing sequential code, the parallel version adds about 15 extra lines of C on average (conveniently encapsulated in a separate "worker" object). As usual in "embarassingly parallel" code, the measured speedup is essentially linear in the number of available cores. The resulting GP binary has been successfully crash-tested by a panel of users during the 2017 PARI/GP Atelier in Lyon: basically, substituting the `Configure` instruction by `Configure --mt=pthreads` before compilation transparently yields the expected speedups for the target functions on participants dual-core laptops.

T5.2: "GAP". No deliverable is due for the evaluation period but steady progress was made on Deliverable D5.15: "Final report and evaluation of all the GAP developments.". Over this period, eight releases were cut incorporating contributions to Deliverable D3.11: "HPC enabled SAGE distribution".

Another major direction of efforts is the HPC-GAP integration: HPC-GAP is a fork of GAP initiated during the SCIENCE project, which enables multithreaded calculations. Now that HPC-GAP has reached maturity, it's critical for its widespread use and long term maintainability to merge it back into GAP's master branch. The first step towards this long-standing goal, which is at the core of Task **T5.2**, is the major release of GAP 4.9, published in 2018 and allowing compilation in HPC-GAP compatibility mode. It comes together with the new manual book called "HPC-GAP Reference Manual", which is also available online at `https://www.gap-system.org/Manuals/doc/hpc/chap0.html`. HPC-GAP integration required a major refactorization of GAP's build system mainly developed by our external collaborator Max Horn (Giessen) into GAP's master branch during the Sage-GAP-Days-85 we organized in March 2017. An overview of the most important changes introduced in GAP 4.9.1, with links to corresponding GitHub entries, can be found in the GAP manual: `https://www.gap-system.org/Manuals/doc/changes/chap2.html`. Many GAP packages also have been updated to work in GAP 4.9 and make use of its new features.

T5.3: "Linbox". The deliverable D5.12: "Exact linear algebra algorithms and implementations. Library maintenance and close integration in mathematical software for LINBOX library" due for this reporting period is delivered on time. The contribution, as detailed in the report is twofold:

- Several algorithmic innovations have been produced: a new matrix invariant for rank profiles [DPS16], a new symmetric matrix factorization algorithm and its high performance implementation has been proposed [DP18]; new representation and algorithms for quasi-separable matrices [Per16; PS17], and new developments on interactive certificates for the security of large scale distributed computations on unsafe resources [Dum+16; DLP17]. These results have been presented in the main venues in the domain: the international conference ISSAC'16-17-18 and the Journal of Symbolic computation.
- Major software developments have happened in the LinBox ecosystem, increasing robustness, maintainability and introducing new computational features, such as the new symmetric factorization algorithm or a parallel triangular matrix inversion implementation. Not only have these library been closely integrated within SAGE, but the interface has been fully rethought thanks to a broader support of C++ in Cython.

Deliverable 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." is making good progress thanks to the work of our

OpenDreamKit engineer Hongguang Zhu. Several code prototypes already implement a Chinese remainder based parallel rational solver on a distributed infrastructure. We are now working on improving the communication patterns, and on designing a parallel rational vector reconstruction algorithm, as it has now become the bottleneck in this computation.

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 are not. Since the last update 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 also now support polynomial exponents of unlimited size with the three basic monomial orderings of lex, deglex, and degrevlex. Suggestions by colleagues in the HPC community including Bernard Parisse, Michael Monagan, Roman Pearce, and Mickaël Gastineau have been invaluable.

The serial implementations of the operations of multiplication, division and GCD are complete in both the dense and sparse cases and the performance is competitive with other systems. The parallel implementation of multiplication is also complete with competitive performance as well. We are on track to have division and GCD parallelized and delivered on time.

We have also been able to parallelize polynomial root clustering, which is a major engine in Singular that benefits from fine-grained parallelization. Performance improvements continue to be made by Remi Imbach (now at NYU following his ODK contract). The fully working implementation is at `https://github.com/rimbach/Ccluster`.

T5.5: "MPIR". All deliverables for this task have been delivered at the previous reporting period.

T5.6: "HPC infrastructure for combinatorics". The goal of this task is to use combinatorics as a source of challenges to experiment on various HPC techniques. Recall that from the first reporting period, deliverable D5.1: "Turn the Python prototypes for tree exploration into production code, integrate to SAGE." successfully implemented a MapReduce programming model on large datasets described by a recursion tree. Since it is written purely in Python, the code doesn't perform well when the computation in each node is short. A good technology for handling such situations with fine grain parallelism, is CILK++. However since both Intel and GCC teams decided to discontinue CILK++ support, we looked for some replacement and also to scale from shared memory to cluster. It turns out that there isn't currently any widespread well maintained replacement.

On the other hand, while writing our code to experiment with various solutions, we came to realize that designing new algorithms based on the vector primitive of the modern processor (AVX instruction set) allows to gain large speedup ($2\times$ to $50\times$) on the handling of small combinatorial objects. By requiring the invention of new algorithms, this becomes definitely a research question. As a preliminary results, we started to develop a library called HPCombi which can already speed up some computation of GAP and SAGE through the LIBSEMIGROUP library. This work resulted in a keynote talk at the PASCO 2017 conference and an invitation to the Scottish Programming Language Seminar 2018.

On the less experimental side, progress has been made into Sage to compute with integer vector which are ubiquitous in combinatorics. Using Cython, we wrote a bridge called PPLPY for the PPL library which now provides access to the PPL library to any Python user. This package use general linear programming techniques to enumerate integer vectors. Improving SAGE capabilities in polytope computations and linear algebra is critical for combinatorics. Similarly two other interfaces have been developed to LATTE (counting integral points in polytopes) and POLYMAKE (the reference software for Polyhedral computations). Finally, some more

combinatorial objects were optimized using the Cython technology, namely permutations and Lyndon words (which are a combinatorial tool to deals with vectors up to a cyclic permutation).
T5.7: "Pythran". The goal of this task is to make Pythran easily integratable in large-scale project, taking into account native dependencies, compilation time, memory footprint, speed and size of compiled binaries as well as multi-platform support. Integration with CYTHON is a possible mean to achieve this goal.

Two projects have been selected for this task: SCIPY and SCIKIT-IMAGE. These projects are relevant for PYTHRAN because they have many small to medium kernels that can benefit from compilation. Even though PYTHRAN has not been selected as a scipy backend, the exchanges with the community have led to a great deal of improvements of which all PYTHRAN users take advantage. The SCIKIT-IMAGE community is still examining the possibility of using PYTHRAN as an acceleration mean.

Integration of PYTHRAN as a CYTHON backend for NUMPY has improved in various aspects: better error detection, more supported expression patterns and improved performance for the compiled expressions.

Deliverable D5.11: "Refactor and Optimise the existing combinatorics SAGE code using the new developed PYTHRAN and CYTHON features." is shared with Task **T5.6**: "HPC infrastructure for combinatorics", the status of which we reported on above.
T5.8: "Sun Grid Engine Integration in Project JUPYTER Hub". All deliverables for this task have been delivered at the previous reporting 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) the participants working on WP6 met and discussed the topic of integrating the OpenDreamKit systems into a mathematical VRE toolkit. Key results 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*.

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 underlying the VREs (K) 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 [D6.216] for details.

A side effect of **R1.** is that the verification aspects anticipated in the proposal are non-critical to the OpenDreamKit project. In particular the value of the exemplary verification of an LMFDB algorithm in **T6.8** and deliverable D6.8: "Curated Math-in-the-Middle Ontology and Alignments for GAP/ SAGE/ LMFDB" seems highly questionable.

Correspondingly we have refined the notion of "triformal theories" coined in the proposal into the concept of "$\mathcal{DKS}$ theory graphs", which can be formalized and implemented without the extension of OMDoc/MMT for "biformal theories" anticipated in the proposal.

Through the concerted effort of the WP6 participants, we have been able to implement this design into prototypical $\mathcal{DKS}$ base patterned after the MitM paradigm with virtual theories, generating interface theory graphs for the GAP and SAGE systems and integrating the LMFDB system via the MitM codec architecture described in [D6.216]. Based on this, we were able to generically integrate GAP, SAGE, and LMFDB via the standardised SCSCP protocol [HR09] – essentially remote procedure calls with OpenMath Objects. This case study shows the feasibility of the initial design of $\mathcal{DKS}$-bases; further investigations and the integration of additional systems will determine the practicability.

1.2.6.2. **Milestones.**

M9: "First Math-In-The-Middle-based interoperability prototype" (month 36). *"User story: thanks to a fully functional prototype integrating of at least the systems* GAP*,* SAGE*,* SINGULAR*, and* LMFDB *via the* SCSCP *Protocol, end users shall be able to run calculations involving any combination of those systems from any of them. This prototype will be the basis for integration work for additional systems and the user interface from WP4."*

Workpackage **WP6** is fully on track with this milestone. After first integration and DKS prototypes (the MitM VRE middleware framework) became available in late fall 2017 (see [Koh+17; WKR17]) we were able to develop more sophisticated – and mathematically more realistic/relevant – use cases [CL1818] and generalize those parts of the framework that had been overly specific to the first use cases. This involved non-trivial investments in all parts of the framework, as well as the system API theory generation systems and (in particular) the MitM ontology.

M10: "Second Math-In-The-Middle-based interoperability prototype" (month 42). *"The goal of this milestone is 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 will 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 not extending the system (programming the VRE systems or the MMT mediator). This means that additional systems can be added at the cost of generating system API theories, extending the MitM ontology and supplying alignments. We are targeting the knowledge bases OEIS, and FindStat (see **T6.7**) as well as PARI/GP. We plan to extend the worked use cases substantially. To this end we already have statements of interest from external researchers, who want to use the flexible integration in the MitM framework and do not mind the communication overheads involved. First work on UI integration work has already begun; see D4.11: "Notebook Import into MathHub.info (interactive display)", which presents a Jupyter kernel for MMT and prototypical MitM-based integration of Jupyter widgets.

1.2.6.3. **Tasks.**

T6.1: "Survey of existing $\mathcal{DKS}$ bases, Formulation of requirements". This task was directly addressed in the WP6 workshops in the first year.

T6.2: "Triform Theories in OMDoc/MMT". For this task we have specified and implemented the concept of virtual theories that can contain large – theoretically even infinite – numbers of declarations and objects (e.g. 3.5M declarations in the LMFDB data base for elliptic functions) in OMDoc/MMT. Virtual theories are characterized by the fact that they are too large to keep in main memory of the MMT System and have to be partially and lazily imported from an external data store. We have reported on the design in D6.2: "Initial $\mathcal{DKS}$ base Design (including base survey and Requirements Workshop Report)", on a first implementation on the international conference (MACIS 2017) [WKR17], and finally on an extended use-case in LMFDB in D6.5: "GAP/SAGE/LMFDB Interface Theories and Alignment in OMDoc/MMT for System Interoperability".

T6.3: "$\mathcal{DKS}$ Base Design". This task was directly addressed in the WP6 workshops in the first year and has led to the design and implementation in D6.2. A first implementation has been presented on the international conference (MACIS 2017) [WKR17], and finally on an extended use-case in LMFDB in D6.5: "GAP/SAGE/LMFDB Interface Theories and Alignment in OMDoc/MMT for System Interoperability".

T6.4: "Computational Foundation for Python/Sage". In the course of the deliberations in the WP6 workshops we saw a shift from the development of computational foundations and verification towards API/Interface function specifications to enable semantic system interoperability via the Math-in-the-Middle (MitM) Ontology. Consequently, emphasis has changed to the generation of system API theories for GAP, SAGE, SINGULAR, and LMFDB, which act as OpenMath content dictionaries. The computational foundations exist but are rather more simple than originally anticipated. Much of the functionality has been offloaded to the SCSCP standard – remote procedure call with OpenMath representations of the mathematical objects – developed in the SCIENCE Project. As a direct consequence of the work in OpenDreamKit the OpenMath Society has promoted the SCSCP protocol into as an OpenMath Standard.

Conversely, the GAP and SAGE CDs are rather more elaborated than anticipated in the proposal, and thus form a viable basis for alignment with the MitM Ontology.

The MitM integration paradigm is the result of our research and development on the computer algebra foundations in this task has been presented on the international conference MACIS 2017 [Koh+17] and is described in deliverable D6.5: "GAP/SAGE/LMFDB Interface Theories and Alignment in OMDoc/MMT for System Interoperability", which presents an advanced CAS

integration use case. The MitM ontology and the system API theories have been developed to the point, where the data model is fully developed and the contents cover the use cases corresponding to this task and **T6.3**: "$\mathcal{DKS}$ Base Design" are surveyed in D6.8: "Curated Math-in-the-Middle Ontology and Alignments for GAP/ SAGE/ LMFDB".

T6.5: "Knowledge-based code infrastructure". The MitM architecture developed in WP6 has given important impulses to make the code infrastructure of SAGE and GAP more declarative (knowledge-based). In SAGE, the category infrastructure was validated (it seems to be the right level of abstraction to generate API theories) and extended; we explore ways to enrich it with additional semantic trough the use of annotations, to maximize the chance of them being accepted and adopted by the Sage community.

In GAP, the facilities for "constructors" was reformed, extended by an infrastructure for documentation and static typing/type analysis, and the code base refactored for over 2000 constructors. Similarly, the online documentation subsystem for GAP has been regularized and synchronized with the constructor level. Already at this early stage of the task the new "knowledge-based perspective" has revealed a plethora of errors and inefficiencies and has contributed to the code quality in both systems.

T6.6: "OEIS Case Study (Coverage and automated Import)". For the OEIS case study we have parsed the OEIS data and converted it into OMDoc/MMT theories (ca. 260,000). The main problem solved here was to parse the formula section (generating functions, relations between sequences, ...): they are represented in a human-oriented ASCII syntax, which is highly irregular, ill-separated from surrounding text, and interpunctuation. Nonetheless we managed to recover ca. 90% of the formulae and

i) generate ca. 100,000 new relations between sequences and
ii) provide a package of ca. 50,000 generating functions to Sage (which can be used e.g. in the FindStat database).

We use this theory set to test the functionalities of "virtual theory graphs" (one step up from the "virtual theories" developed in **T6.3**).

T6.7: "FindStat Case Study (Triformal Theories)". We have seen that the LMFDB already shows all the complexities needed to develop full-coverage DKS functionality for the OpenDreamKit VRE toolkit. On the other hand our survey shows that our DKS design (OMDoc/MMT virtual theories) is sufficient for covering the FindStat use case as well. Therefore we have delayed this taks to the last year of the OpenDreamKit project, when the system API theories for SAGE and OEIS have matured. With the declarative design of the virtual theories, task **T6.7** becomes a matter of writing down the schema theories system API theories for FindStat and defining the requisite codecs. We expect this to be a matter of one of two weeks of joint development of the FAU team together with UPSud.

T6.8: "LMFDB Case Study (Triformal Theories)". Work on this task has started. Given the concept of virtual theories developed in **T6.2** the task is to build a database connector that converts the MongoDB tables in LMFDB into "mathematical objects". We have identified the problems – e.g. that objects are reduced to ad-hoc database records: for instance elliptic curves are represented as a quadruple of integers, where the last is represented as a string of digits as the range of MongoDB integers is too small. We have developed an architecture of language-specific Codecs which mitigate these problems in a knowledge-centered way (Codecs are OMDoc/MMT objects) that interpret database records as OMDoc/MMT objects and can thus be used populate virtual theories. The next step is to extend the existing MMT query language by a query compiler into the underlying data store system; concretely to MongoDB underlying LMFDB for **T6.8**.

T6.9: "Memoisation and production of new data". Not applicable for this period

T6.10: “Math Search Engine”. Work on the first work phase has proceeded as planned and has culminated in D6.1: “Full-text Search (Formulae + Keywords) over LaTeX-based Documents (e.g. the arXiv subset)”. The second work phase on this task presupposes the Math-in-the-Middle ontology (as we call it now.) Where we already have that, e.g. for the OEIS (see **T6.6**) we already have a running search engine. The main problem here is to devise intuitive query interfaces and integrate them into the OpenDreamKit VRE framework.

1.3. **Impact**

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

There is for now no change to bring to Key Performance Indicators. The evolution of the measures between Month 18 and Month 36 will allow the Coordinator to evaluate if the selected KPI are appropriate.

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 and available to all academics in Europe: `https://jupyterhub.fedcloud-tf.fedcloud.eu/`. 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. The current main instance (`http://mybinder.org/`) is often overloaded by the demand, proving that it has identified just the right service for a critical need.

  Our work on packaging **T3.3** and JUPYTER integration **T4.1** enabled the easy definition of executable environments for Binder and beyond that include OpenDreamKit's computational math software.

  Within our EGI partnership, we have started exploring ways to contribute additional computing resources to the main mybinder instance 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/`.
- JupyterHub @ JUPYTERHUB instance deployed on USheffield's HPC system.

- JupyterHub @ , , ... We have helped with the definition and deployment of those several University wide instances, exercising them with large classes (e.g. 400 students at . Lessons learned at the occasion have been shared through blog posts such as:
  - `https://opendreamkit.org/2018/10/17/jupyterhub-docker/`
  - `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`.

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

Not applicable

3. Update of the Data Management Plan

A second version of the Data Management Plan was released in D1.6: “Data Management Plan V2”. Up to minor updates to the list of data sets, there was no change since D1.2: “Data Management Plan V1”.

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 in OpenDreamKit project.

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. Most recommendations were implemented right away at the occasion of the Work Plan Revision process that followed the second reporting period and involved the reviewers and advisory board. In the sequel we explain how we took the recommendations into account.

Recommendation 1: *The deliverable D4.7 needs to be presented or reasons for not presenting it needs to be detailed in the progress report.*

Taken from our Work Plan Revision proposal: This was in fact a communication glitch, which we clarified since with the project officer. The deadline of M14 for D4.7 was a typo in the original proposal: it should have been M24, 12 months after the related deliverable D4.4: "Basic Jupyter interface for GAP, PARI/GP, Sage, Singular". After consulting our former project officer back in Fall 2016, this typo was fixed in the second amendment to the grant agreement. We oversaw that at the time of the review this agreement was not yet effective and available to the reviewers. We will make sure to mention any such change in the report next time.

Recommendation 2: *The final Progress Report must be presented and it should report on all deviations including cancelled or amalgamated deliverables.*

We followed the advice from our Project Officer: *"The review report says that you should amend the current progress report on all deviations etc but as we have already started to process the report and cost, you don't need to do that anymore for the past period. However, in the future please report on all changes/deviations in the progress report to be clear."*

In addition we made sure that changes and deviations for Reporting Period 2 were reported in this document.

Recommendation 3: *The workplan of WP7 should be assessed critically and the draft proposal of the revised plan submitted to the Commission as soon as possible and latest by 1 July 2017.*

This was implemented in the Work Plan Revisions.

Recommendation 4: *The deliverables presentation should be clearer concerning their key content. They should include a clear "executive" summary, state the purpose and target audience of the deliverable and include clear conclusions. The full title of the deliverable should be used instead of the often used form "Report on Dn.m". Consolidation of the deliverables should be considered to reduce their number.*

Deliverables were consolidated at the occasion of the Work Plan Revision, without affecting the content and work time line, beside enabling a bit of flexibility to adapt to a quickly evolving landscape (where e.g. some actions become more pressing and others less). Together with the refactoring of WP7 this consolidation reduced the number of remaining deliverables from 55 to below 40.

We tried hard to follow the presentation recommendations for all newly submitted deliverables.

Recommendation 5: *Provide clearer and more relevant work-package specific milestones to allow for effective monitoring of the project's progress.*

This was implemented in the Work Plan Revisions.

Recommendation 6: *The web presence should be made more attractive for broader dissemination and it is recommended to create separate externally facing websites. The internal project website could be separated from the external one, as their purpose and target audiences are not the same. The social media presence and blogging should be made more vivid and attractive.*

At the occasions of the Work Plan Revisions, we proposed a plan to improve our web presence, which we implemented during Reporting Period 2, with continued efforts on the content since then, including interview videos and an upcoming motion design explainer video. To this end, we used help from master students in communications, and two specialized companies. For details, see **T2.1**: "Dissemination and Communication activities".

Recommendation 7: *The KPIs should be refined and alternatives suggested and the currently very generic milestones should be revised to be more specific and appropriate for project monitoring purposes.*

Alternative KPI's were chosen at the occasion of the Work Plan Revisions.

Recommendation 8: *Regarding the WP2, it is recommended to deploy some additional resources to improve the externally facing website as well as improving the internal site.*

See Recommendation 6 above.

Recommendation 9: *WP3 work to be integrated into the proposed revisions of the website. Efforts need to be re-allocated as SMC developers have already done the work that was described to be in D3.4.*

This was implemented in the Work Plan Revisions.

Recommendation 10: *Regarding WP5, make contacts with HPC community in order to ascertain current state- of-the-art. The work in this WP needs to be nearer the leading edge.*

The following text is updated from our Work Revision Proposal.

We would like to clarify the context of high performance *mathematical* computing, which is subject of WP5, and its relation to high performance *scientific* computing.

High performance computing is mostly driven by scientific computing and its applications. As a consequence, it focuses on numerical computations using approximate floating point arithmetic and parallel numerical linear algebra. Decades of efforts in research and development in this field have produced a mature set of software and algorithmic practices and even impacted the design of most of nowadays computers.

On the other hand, computational mathematics, at the core of OpenDreamKit activity, differs from scientific computing primarily on the type of arithmetic being used. There is in fact a large variety of arithmetics to be supported depending on the application: finite fields of small, medium and large cardinalities, multiprecision integer and rationals, polynomials over these domains, etc. Obviously all these arithmetics need to be exact which defeats a direct use of floating point arithmetic. Another challenge is the deep interplay between these arithmetics which are often composed in high stacks of algebraic structures: e.g. tensor algebras built on top of algebras built on top of combinatorics and fractions, the latter being built on polynomials built on arithmetic.

Consequently, the experience of the numerical scientific computing community can not be blindly exploited in the development of high performance mathematical computing. Driven by emerging applications, such as experimental mathematics, cryptanalysis, discrete optimization, and bioinformatics, high performance mathematical computing has become an active field in computer algebra over the last two decades, but is comparatively at an earlier stage of development, given the smaller community and the broader scope to be adressed. However interactions with the numerical scientific computing community have always been intense and several crucial innovations resulted from a convergence between the two fields: floating point arithmetic can be used under control for finite field linear algebra, sparse solvers used in cryptography are adapted from iterative numerical methods, etc.

In the first evaluation period of the project, most efforts have been put towards tasks specific to mathematical computing: improving some core computational kernels, for polynomial and finite field arithmetic (D5.5 and D5.7), parallelization of recursive tree explorations D5.11, etc. This could explain a feeling that these contributions are not in line with mainstream numerical high performance computing research. Achievements during second reporting period, which

will be presented at the formal review, include significant contributions to parallel exact linear algebra, which is much closer in nature to numerical HPC.

Participants of the WP5 are (and have been for a long time) designing leading edge software for high performance mathematical computing and are in continuous interaction with the mainstream numerical HPC community. More precisely, we identify that the major forthcoming interactions will be in the following aspects:

Parallel runtime for task based parallelization: Parallel runtime systems are becoming a key component to properly harness the always growing number parallel cores. Shifting from low level thread management to higher level parallel programming languages and runtimes has become a hot topic in numerical HPC. Convergence in this area is already happening, with for instance our participation to a french national workshop "Journée Runtime" where we could exchange with the numerical HPC community on our experience using XKaapi, Cilk and OpenMP for the task based parallelization of exact linear algebra. During the 11th e-Concertation Meeting, we also started an interaction with the research group at the BSC (Barcelona Supercomputing Center) regarding their runtime OmpSS.

Automated SIMD optimization.: SIMD is the lowest level of parallelization available inside each processor. Automating the optimization of such low level code is a common target between communities of exact and numerical computation. Our project's contribution in D5.5 is a major step in this direction. On the topic of SIMD vectorization, we have started several interactions with the developpers of the BLIS project (a framework instanciating blas-like libraries for numerical linear algebra), regarding implementations of AVX512 kernels for matrix multiplication[5]. We also have interracted on the topic of implementations of Strassen's algorithms which is a core feature of `LinBox` (**T5.3**) and for which authors of BLIS have shown recent interest.

We nevertheless doubled our efforts to deepen our contacts with the mainstream HPC community, and very much appreciate the contacts that were and will be supplied by the reviewers.

Recommendation 11: *The activities of the WP7 should be assessed critically and a revised work plan for this WP should be presented that is of greater relevance to the aims and goals of the project. If a satisfactory resolution of the issues is not reached then this WP should be dropped and the effort re-assigned elsewhere within the project.*

See Recommendation 3 above.

[5] `https://github.com/flame/blis/issues/182`

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. Because of this difficulty to recruit in the past year, there have been slight changes in the workplan, which do not put the project results at risk.

The following people were hired in the past year or are in the hiring pipeline for next year

NAME	GENDER	PARTNER	POSITION	HIRING DATE
Theresa Pollinger	F	FAU	Junior Researcher	October 2017
Tom Wiesing	M	FAU	Junior Researcher	September 2017
PD. Dr. Florian Rabe	M	FAU/UPSud	PostDoc	December 2017
Dr. Katja Berčič	F	FAU	PostDoc	November 2018

Dr. Florian Rabe is a joint appointment and splits his time and research between FAU and UPSud, which reflects the close cooperation and cross-fertilization of methods in WP6.

OpenDreamKit partners had to face some Human Resources issues in the past year:

- UPSud: Thanks to an early start in the recruitment process, and despite some difficulties in attracting experienced candidates for a part time position, the project manager position (24PM) was filled by Benoît Pilorget shortly after the start of the project. Unfortunately at month 36 the project finds itself without a Project manager since the departure of B. Pilorget.

  The recruitment of UPSud's first Research Engineer (48PM) was delayed by four months because the top ranked candidate for this position, Erik Bray, was originating from the US and needed time to arrange for his moving; there were also some administrative delays (visa, ...).

  The second Research Engineer position (36PM) was more problematic for internal administrative reasons. The top ranked candidate, Jeroen Demeyer, had the perfect profile; however for family reasons, he wished to work most of the time from Ghent in Belgium. After eight months investigating an administrative solution to hire him at UPSud, and a temporary four month solution, it was decided with OpenDreamKit's Steering Committee and Project Officer to instead add Ghent's university as new partner, hire Jeroen Demeyer there, with an adequate budget transfer and amendment to the Grant Agreement.

  Those delays have induced late start on several tasks, and costed much management time. However the excellence of the recruitment, well confirmed by the results obtained so far, was worth it and will soon compensate for the late start.

  In addition to this, a three year PhD position was open to work on WP6, starting from Month 12. By lack of suitable candidate, this position will be converted into a two year PostDoc position, presumably starting at Month 24. Active advertising has started and there are some tentative candidates. The relevant deliverables being due late in the project, no delay is to be expected from this change.

- CNRS: Because the research engineer offer (48PM) was still not filled in the Summer 2016, the CNRS decided to divide the position in two full positions of 24 PM each. As a result, a candidate was already selected for one of the two positions and should begin his work this Fall 2016. Thanks to the PM division, there should be no delay in any task or

deliverable.

- JacobsUni: Michael Kohlase, lead PI for Jacobs University, has moved on 01/09/2016 to Friedrich-Alexander-Universität Erlangen-Nürnberg, and most of his team will follow him. The necessary changes have been implemented in a grant agreement amendment in 2017.

- UJF: The original tentative candidate for UJF's Research Engineer position (12PM, planned to start on Month 1), Pierrick Brunet, finally declined the position to accept an alternative permanent offer. The position will be filled by another candidate in Autumn 2016. This induced a delay of Deliverable **D5.2** from Month 12 to Month 18, without impact on other tasks.

- UNIKL: UNIKL had to split the 12 PM planned for a software developer into 2 shorter positions (Anders Jensen and Alexander Kruppa) in order to deliver the planned work on time. Indeed the few qualified persons for this job were not able to accept this 12 months position during the timelapse planned within the project.

- USFD: The University of Sheffield has also been struggling in the the hiring process of a postdoc (36PM). The position should be filled this Autumn.

- Southampton: Southampton faced administrative difficulties in the recruitment of Marijan Beg (38PM) as a post-doc, due to the Croatian nationality of Mr Beg. His recruitment was delayed of four months, and therefore some tasks and deliverables, planned to be borne around the end of the project, were postponed of four months. However no serious delay nor implication on the main tasks of OpenDreamKit followed these difficulties.

- UVSQ: Nicolas Gama is currently on a long-term leave until September 2017. This will not affect the project in any way.

- UZH: The University of Zürich partner is only composed of one person, Paul-Olivier Dehaye, who does not enjoy a permanent position there. There have been worries that Mr Dehaye's contract with his university might end earlier than planned within OpenDreamKit. But thanks to the action of the OpenDreamKit steering committee, Mr Dehaye has been technically rehired by UZH as a scientific consultant for as long as the project needs.

- Simula: Everything is fine concerning temporary staff recruitment on the Simula side, however we have had to endure the hazards of human ressources with Hans-Peter Langtanger (the PI when the Grant was signed) being on a long-term sick leave, and with Martin Alnaes replacing him as PI currently on a paternity leave. However Benjamin Ragan-Kelley has stepped in to lead the Simula contribution in the meantime and all planned tasks are on time.

Altogether, this first year confirmed that the recruitment 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 – worked adequately: with appropriate reshuffling of the work plan, we don't expect an impact on the overall progress of the project.

4.2.2. **Different groups not forming effective team**

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). The organization of many joint workshops (for example the Sage-GAP workshop, the Atelier Pari attended by SageMath developers, the WP6 workshops) helped bootstrap joint activities through brainstorms and coding sprints. Upcoming workshops are planned on Year 2 to strengthen collaborations with the social aspects team in Oxford and the Singular team in Kaiserslautern.

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

The consortium is keeping in their minds the end-user needs. Since OpenDreamKit is improving already existent software which have their own users, their needs are naturally met. However Key performance Indicators will evaluate the effects of OpenDreamKit on these software. KPIs, indicated in the Proposal, will be launched this Autumn with the help of the end-user group which was merged with the Advisory Board. Constant links between the accomplished work and the end-user needs should be made in WP2 deliverables and also in WP7 deliverables when relevant. Open tracking of KPIs evolution can be found on GitHub.

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

As planned, we are regularly shifting 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 yet: none of the external software component we rely on have failed us. Quite on the contrary, critical software like JUPYTER have continued to blossom. Besides the high modularity of the design means few components are 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 is chaired by Hans Fangohr, from the University of Southampton and European XFEL GmbH. He is supported in this task by Mike Croucher from the University of Leeds, Alexander Konovalov from the University of St Andrews, and by Konrad Hinsen from the Centre de Biophysique Moléculaire with whom a Non-Disclosure Agreement was signed.

The 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 is the head of research computing at Leeds, 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 founder and director of the UK's only centre for doctoral training in computational modelling, a fellow of the Software Sustainability Institute, was chairing the EPSRC's national scientific advisory committee on high performance computing, and is leading big data analysis infrastructure development at the European XFEL research facility.

The quality review board has reviewed deliverables after the reporting period 1, identified good practice - both in terms of software engineering content but also presentation of the work -, produced a report, and shared the findings with all members in the project to improve the quality of the remaining deliverables. A summary is included in Sec. 4.3.3. The board has stuck to its no-blame culture in its reporting.

A similar process is underway for Reporting Period 2.

4.3.2. Good practice

The quality review board notes that there is no firmly established view on what best practice establishes, and that (i) we expect our best practice checklist to grow and change, and (ii) that due to the variety of possible outputs not all categories will be appropriate for every item under review. Here is a summary of good practice, with particular focus on software and computational projects.

4.3.2.1. Software engineering. Use of

- ☐ version control
- ☐ tests
- ☐ automated tests
- ☐ continuous integration
- ☐ automatic building of releases

4.3.2.2. Dissemmination.

- ☐ Host code publicly (Github, . . .)
- ☐ Reference Manual (APIs)
- ☐ Tutorial (for beginning users)
- ☐ Examples
- ☐ Offer live interactive online demos (for example through Binder)
- ☐ Support mechanisms (email/forum/gitter/github issues/. . .)
- ☐ How to cite the output?

- ☐ Installation mechanism
- ☐ High level description of tool/activity accessible to non-experts
- ☐ URLs/Blog/etc to and from OpenDreamKit project
- ☐ Grant acknowledgements
- ☐ Open Source license
- ☐ Workshop
- ☐ Engaging users

4.3.2.3. **Pathways to impact.**

- ☐ Does the software address the needs of the users?
- ☐ Workshops to gather feedback

4.3.3. **Summary of recommendations for deliverable reports**

For reports that are well written, the quality review board found good software engineering practices. However, for some deliverables the reports were more difficult to assess. To address this, the following guidance has been developed:

- Context setting
  - what is the problem that is being addressed?
  - Why should we care about the deliverable?
  - what is the thing that has been created?
- Stating the obvious: for example if people (outside the project) are excited about it
- Provide introduction to topics, even for people not too familiar with the field
- Comment on other good (software) practices you may use without thinking about it (version control, testing, continuous integration, distribution)
- Comment on the testing that has been done, even if not automatic. If there is a prototype / demonstrator, explain it in more detail.
- Work relating to sustainability, should be mentioned, even if implicit (for example growing a community through workshops will help to make the project more sustainable).
- Anything with impact should be mentioned (contribution / uptake to Software carpentry, other computational science and software projects, github, users, . . .)
- Have an executive summary on the first page (just half a page).
  - why does this deliverable exist? (context)
  - have you achieved everything you set out to do?
  - what would be / are the next steps?

4.3.4. **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.

5. Deviations from Annex 1 (if applicable)

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.

5.1.4. 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 aligments – 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.

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.

REFERENCES

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

[CL1818] John Cremona and David Lowry-Duda. *Mixing Data and Computation to explore mathematical data sets: Knowledge to the rescue with LMFDB + SageMath+ PARI/GP+ MitM*. 2018. URL: https://raw.githubusercontent.com/OpenDreamKit/OpenDreamKit/master/WP6/usecase-notes/UseCaseHecke.pdf.

[D6.216] Paul-Olivier Dehaye et al. *Report on OpenDreamKit deliverables D6.2: Initial D/K/S base Design (including base survey and Requirements Workshop Report) and D6.3: Design of Triform (D/K/S) Theories (Specification/RNC Schema/Examples) and Implementation of Triform Theories in the MMT API*. Deliverable D6.2. OpenDreamKit, 2016. URL: https://github.com/OpenDreamKit/OpenDreamKit/raw/master/WP6/D6.2/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 25–29, 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.

[DLP17] Jean-Guillaume Dumas, David Lucas, and Clement Pernet. "Certificates for triangular equivalence and rank profiles". In: *Proceedings of the 42nd International Symposium on Symbolic and Algebraic Compuation*. ISSAC'17. ACM, 2017. URL: https://www.openaire.eu/search/publication?articleId=dedup_wf_001::b7bd720f30461e3101ffcc69db4bf4a8.

[DP18] Jean-Guillaume Dumas and Clément Pernet. "Symmetric Indefinite Triangular Factorization Revealing the Rank Profile Matrix". In: *Proceedings of the 2018 ACM on International Symposium on Symbolic and Algebraic Computation*. ISSAC '18. New York, NY, USA: ACM, 2018, pp. 151–158. ISBN: 978-1-4503-5550-6. DOI: 10.1145/3208976.3209019.

[DPS16] Jean-Guillaume Dumas, Clement Pernet, and Ziad Sultan. "Fast Computation of the Rank Profile Matrix and the Generalized Bruhat Decomposition". In: *Journal of Symbolic Computation* In Press (2016). DOI: 10.1016/j.jsc.2016.11.011. URL: https://www.openaire.eu/search/publication?articleId=dedup_wf_001::beb92e25f275ea5bb6b2a97251143703.

[Dum+16] Jean-Guillaume Dumas et al. "Linear Time Interactive Certificates for the Minimal Polynomial and the Determinant of a Sparse Matrix". In: *Proceedings of the 41st International Symposium on Symbolic and Algebraic Compuation*. ISSAC'16. ACM, 2016. URL: https://www.openaire.eu/search/publication?articleId=dedup_wf_001::d9ad2805d3ca077f96b1335f3f74b894.

[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.

[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.

[Per16] Clement Pernet. “Computing with Quasiseparable Matrices”. In: *Proceedings of the 41st International Symposium on Symbolic and Algebraic Compuation*. ISSAC’16. ACM, 2016, pp. 389–396. DOI: 10.1145/2930889.2930915. URL: https://www.openaire.eu/search/publication?articleId=dedup_wf_001::8a8f37f7a1ca153e335c708fd621110f.

[PS17] Clement Pernet and Arne Storjohann. “Time and space efficient generators for quasiseparable matrices”. In: *Proceedings of the 42nd International Symposium on Symbolic and Algebraic Compuation*. ISSAC’17. ACM, 2017. URL: https://www.openaire.eu/search/publication?articleId=dedup_wf_001::b7bd720f30461e3101ffcc69db4bf4a8.

[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.