Danish e-infrastructure Cooperation - DeiC
Att: Gitte Julin Kudsk, Head of DeiC Sekretariat

Expression of interest Integration Portal

In relation to request of expression of interest circulated amongst Danish Universities as of 2. April 2020 (bilag 2).

Consortium Description

SDU, AU and DTU hereby express interest in the software development project of the Integration Portal. The consortium members have many years of experience in providing access to national HPC services and developing complete software solutions for research. In particular, the consortium has the strongest developer team available in Denmark for research cloud services.

The partners in the consortium and their role in the project are presented here:

- **SDU**: SDU will coordinate the project and guarantee that timely actions are taken for the success of the project and its deliverables. The SDU team, based at the eScience center, will develop the front-end graphical user interface in WP1, together with the help of AU and DTU for the design of the user experience. SDU will develop the functionality needed for the integration in WP2 into the "core" of the UCloud/integration portal service. Finally SDU will facilitate and develop the core functionalities needed for the integration of national data resources in WP3.
- **AU**: AU will help with the design of the user experience for the graphical user interface of the integration portal in WP1. AU experience in supporting new HPC users, in particular from the humanities and social sciences, will be important to develop an easy-to-use platform. AU will also contribute to the development of the solution for the integration to future data management systems in WP3.
- **DTU**: DTU's role in this project is focused on the development of the integration with HPC systems and to act as a large-scale pilot user for the integration portal based on UCloud. DTU will further contribute to testing and improving the solution towards a production-ready national integration portal addressing this DeiC call, and in timely preparation for a future national data management call. DTU is running one of the largest HPC clusters in Denmark called "Sophia", with 16.000+ compute cores and state-of-the-art high-speed interconnect, which will serve as a test case during the development of the UCloud integration portal. The local HPC team managing Sophia

will participate mainly in WP2, and WP1 as needed. With respect to WP3, concerning data management, DTU will participate in the testing of the solution and thus explore the feasibility of integration into it's local storage solution. Feedback from this experience can provide valuable input for identifying blind spot bugs or short-comings of the solution.

Introduction

In the past three years, SDU has developed the **UCloud** platform, which is the most complete, user-friendly and advanced private cloud system nationally and in the Nordic countries.

For the services offered, UCloud is already at the level of usability of commercial cloud solutions, like e.g. AWS, Google Cloud or MS Azure. This project will expand its functionality to provide a national access point to compute and data resources available at national providers and Danish universities.

The UCloud solution is an excellent match for the present integration portal project, as it was designed precisely for this purpose. The overall architecture of UCloud is designed to be able to easily interface with generic computing or data storage resources, such as the national HPC centers or the future national data repositories. UCloud is based on a modular, modern microservice based design; this makes it perfect to easily add new functionalities and allows the system to scale up to match the needs of national users.

UCloud already contains the elements, implemented as sets of microservices, which can be used as the core of the envisioned components in this call such as, e.g., the "center proxy" (compute microservices), the "data proxy" (storage microservices), identity providers (authentication microservices; WAYF is already supported, as well as multi-factor authentication).

In addition, UCloud provides a clean and modern Graphical User Interface (GUI), implemented as a modern React-based web application, which is easy to extend to match the needs of this project.

UCloud is in production at SDU and it has been built from the ground up by the developer team at the SDU eScience Center. UCloud will be released as open-source software.

Finally, the UCloud platform has been built for handling sensitive data based on security and privacy-by-design principles. The SDU eScience is ISO/IEC27001:2013 certified since in February 2020, the first Danish university or public institution certified in the country.

Table of content

Vision and expected function of the portal

The vision of the project is to create a user-friendly portal for researchers in Denmark, which allows easy access to national compute and data resources. We propose to base the solution on the UCloud platform; this will offer a very solid and flexible foundation and it will allow us to match all the requirements in the DeiC call and exceed them as described in this section.

In this project we plan to expand the capabilities of UCloud to:

- provide a GUI for the added functionalities;
- be able to connect to all national HPC centers for a number of common tasks such as, e.g.: batch job management, access to files, accounting of HPC resources, list the hardware or softwares available at the HPC centers, monitor national HPC utilization, etc;

and in the future, in relation to data management:

- be able to connect to/integrate with national data storage resources and national archives;
- initiate data movement to/from said data resources;
- allow queries of metadata in said data repositories to search for datasets;
- allow publication of datasets in international (e.g. Zenodo) or national repositories and archives.

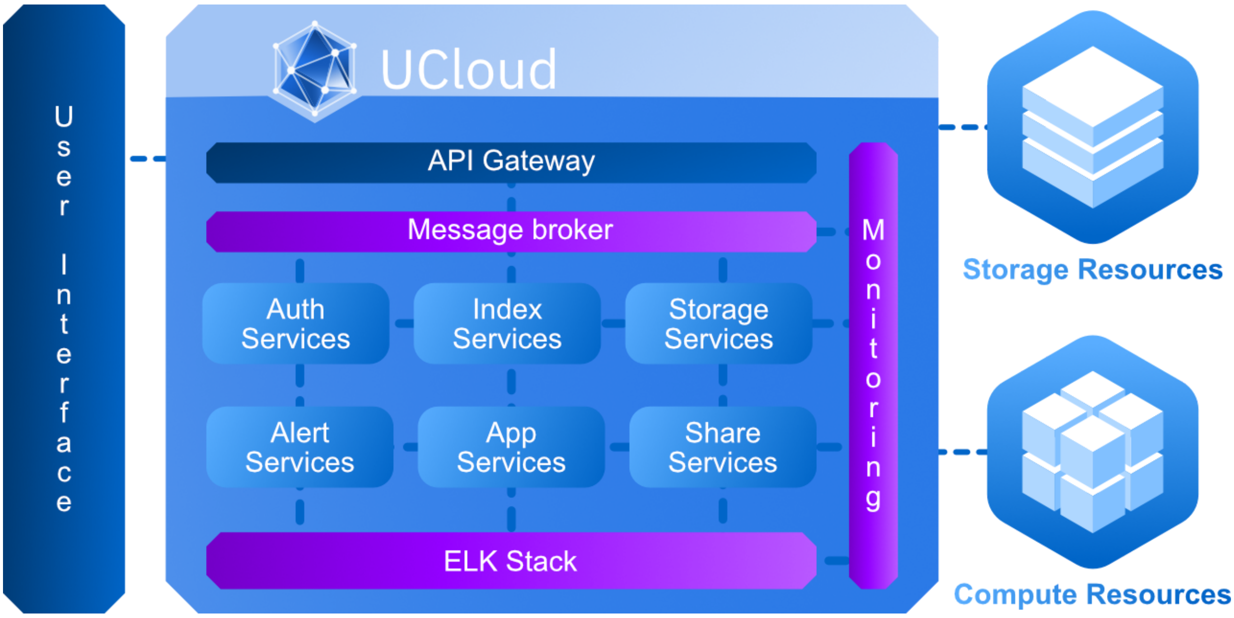
It is envisioned that the portal can also interface to resources available in the Nordic and European context, such as the pre-exascale EuroHPC facility of the LUMI consortium and EOSC. Integration with both resources is possible, however interfaces are not clearly defined at present. For example, the ongoing Puhuri project by NeiC will define the interfaces to access LUMI resources, and once these interfaces and APIs are defined, the UCloud architecture makes it easy to integrate them.

For EOSC, we note that UCloud itself is already among the services that will be integrated as part of the EOSC-Nordic project. Services in EOSC have different levels of maturity and have very heterogeneous interfaces. Until a common API for the EOSC services is defined, it will be possible to select specific EOSC services that provide API access and integrate these in the national portal. This could for example be a selection of data archives providing APIs for access. Once a common API for the EOSC services is defined this will be integrated in the UCloud platform.

Overview of functionality

UCloud acts as a high-level orchestration platform for compute and storage resources. UCloud is based on a modern microservice architecture, in which different services manage specific resources (e.g., compute or storage) or perform specific tasks (e.g. authentication or logging). A microservice architecture has many advantages compared to a traditional, monolithic software: each service has a clearly defined API interface and data sharing dependencies across services are avoided; it is easy to scale up services to serve more users; the service is inherently more robust against failures; it is easy to upgrade functionalities without affecting the service as individual microservices can be updated independently.

The diagram below gives a high-level overview of the main components of UCloud (each box is implemented by several microservices). Besides microservices to manage storage and computing resources, the platform also contains a metadata engine, which can be used for several functionalities such as, e.g., indexing of resources, classification of datasets, authorization engines, etc.



Integration portal work plan

The execution of the integration portal project will fall in three distinct work packages and is expected to last 36 months in total.

Work Package 1 - Web Portal / GUI

UCloud comes complete with a modern web application that provides a graphical user interface (GUI) to the services. The web application is mainly written in Typescript and it uses the React library to provide an interactive user interface.

We will use the UCloud web UI as the foundation for this project, and add the necessary components to expose the new functionality to the users.

The UCloud GUI already includes most of the required functionalities, such as:

- authentication via WAYF and two-factor authentication;
- file browser;
- dashboard for resource utilization;
- submission/cancellation of job to computing backends;
- present a list of available software (apps store).

In this work package we will extend the GUI to include:

- an overview of available computing resources, such as the national HPC centers;
- a list specific software present at each HPC center in the apps store section;
- project management functionalities for:
    - resource allocation and monitoring
    - access management to project resources (e.g. add members to projects with different roles).

The web portal will thus also be extended to allow DeiC and funding agencies to show the overall utilization of national HPC facilities and job submission statistics.

Later on one can envision to include more functionalities related to data management such as: overview of available national data sources; queries to search for specific datasets at national archives; or publish datasets for preservation.

Since most of the work for the GUI/web portal is already made, this proposal will allow to focus on a better integration with national HPC centers and data facilities.

Work Package 2 - Integration with HPC centers

UCloud provides a high-level abstraction to manage storage and compute resources. In this project, we will expand the core UCloud functionality to support the national HPC centers as providers of compute and storage resources. **Access to national resources will be seamlessly integrated within the platform.**

Interaction with the core UCloud platform will be provided by a clean API. The functionality of the “compute proxy” described in the DeiC call, is already present in the core UCloud design which abstracts the computing resources in a set of “apps” microservices. These microservices act in many ways as a meta scheduler, where jobs can be executed on different computing resources.

We will expand the capabilities of the platform to allow seamless integration with providers of computing resources, such as the national HPC centers.

Computing resource providers will be able to register metadata about their services/resources with the UCloud platform. This metadata will provide information about which features the provider supports. For example, a resource provider might declare that it supports storage and batch computing.

We plan to provide an API to interface to external resource providers which will cover:

- user mapping (between the national portal/UCloud and resource providers);
- storage/files at HPC centers (usage, quotas, and read/write operations);
- compute/jobs (usage, quotas, start, cancel, list and interact with jobs);
- project management (e.g. add/remove users to a slurm account).

Most of these APIs will be optional. A provider can declare which APIs are supported in the metadata document. This will allow providers of resources to interface gradually with the national portal.

As part of this project, we will provide a reference implementation for all of these APIs both for the national portal/UCloud and the HPC centers. The implementations will be modular and support best practices already in use at the national HPC centers. Feedback from the initial workshop will determine which systems currently in use at the national HPC facilities to support.

Project management functionalities will be implemented which will allow to

- create projects and allocate resources to them from the national HPC center;
- manage the user membership of project;
- monitor usage resources of projects.

External resource providers will be able to query UCloud for information, such as resource allocation and project membership to automatically set up accounts at the national HPC centers.

The core UCloud framework is written in Kotlin/JVM: a modern, free, open-source programming language with a large user base. The server components use Ktor, a modern asynchronous web-server with support for coroutines. For data storage, the core utilities well-established databases such as PostgreSQL, Redis and ElasticSearch.

Work Package 3 - Integration with national data resources

At a later stage the platform can be expanded to include APIs to interface with national data storage resources and archives. UCloud already provides the user with a unified environment for data and computing. By interfacing the future national storage facilities with the platform, the users can be presented a common view for both computing and storage resources. By providing an integration layer on top of the existing data repositories and future national data facilities, a powerful national data management system can be put into force.

The envisioned integration will provide APIs for operations such as:

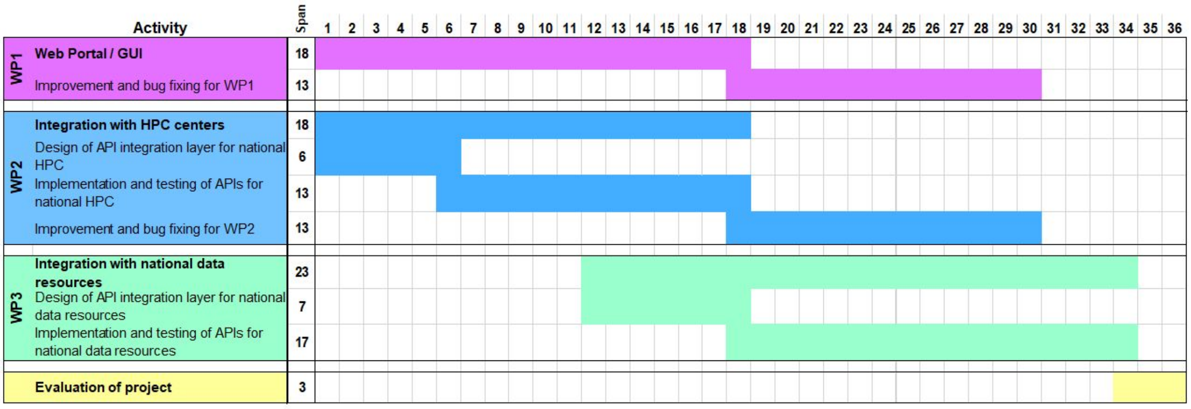
- interface with national authentication and authorization system;
- access datasets located in national data storage resources and national archives;
- initiate data movement to/from said data resources;
- allow queries of metadata in said data repositories to search for datasets;
- allow publication of datasets in international (e.g. Zenodo) or national repositories and archives.

Such functionality can be integrated into this project and presented to the users in the provided GUI.

We stress that the national integration portal based on UCloud will act as an orchestrator, controlling/interfacing to lower-level components for, e.g., national authorization to access datasets or parallel data transfer between locations, etc. These lower-level components already exist in many if not all cases and can be reused by the national storage resources and archives (for example, among HPC centers one could use, e.g., gridftp as a standard tool for fast data transfer). When national archives already implement APIs for the envisioned functionalities, these will be used.

Timeline

In this proposal, we suggest starting the project in September 2020. A suggested timeline is presented in the Gantt chart below. The activities will start with a workshop where all the main stakeholders will be invited to participate and actively contribute to the design of high-level functionality. The stakeholders will also be asked for input on the relevant platforms in use at national HPC centers which should be integrated in the portal. We will work closely with national HPC centers, in particular their back-offices, and the end users to incorporate their feedback into the platform and ensure a good realization of the project.



	Activity	Span	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
WP1	**Web Portal / GUI**	18	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪																		
	Improvement and bug fixing for WP1	13																		🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪						
WP2	**Integration with HPC centers**	18	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦																		
	Design of API integration layer for national HPC	6	🟦	🟦	🟦	🟦	🟦	🟦																														
	Implementation and testing of APIs for national HPC	13						🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦																		
	Improvement and bug fixing for WP2	13																		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦						
WP3	**Integration with national data resources**	23												🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		
	Design of API integration layer for national data resources	7												🟩	🟩	🟩	🟩	🟩	🟩	🟩																		
	Implementation and testing of APIs for national data resources	17																		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		
	Evaluation of project	3																																		🟨	🟨	🟨

Indicative budget

	2020	2021	2022
Project management	100K dkk	100K dkk	100K dkk
Workshops	75K dkk		
Travel and consumables	50K dkk	50K dkk	50K dkk
Developers/A-TAP salaries	275K dkk	2850K dkk	2850K dkk
Total	**500K dkk**	**3000K dkk**	**3000K dkk**

- **SDU**: Project leadership and management. Design of high-level functionalities and APIs. Development of graphical user interface and core functionality of integration portal.
- **DTU**: Participation in project management and steering. Design of high-level functionalities and APIs. Development of integration layer to national HPC centers and the future national data storage.
- **AU**: Participation in project management and steering. Design of high-level functionalities and user-interaction for the web-based graphical user interface. Development of integration to the future national data storage systems and national archives.

Software Development and Deployment Methodology

UCloud uses an **agile methodology** for development and deployment, which will also be used for this project The method is commonly used for complex projects like this one, and it emphasizes collaboration, flexibility, continuous improvement, and high-quality results.

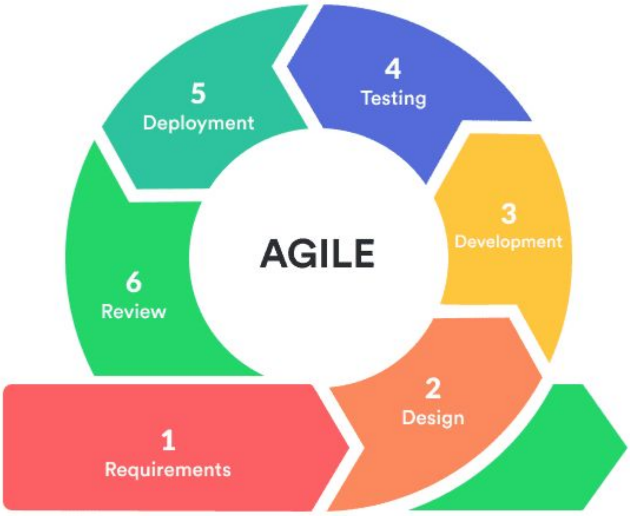
This project methodology is fundamentally designed around the concept of tasks, also known as *issues*. An issue is a description of some desired change within the project. Issues contain a description of either an observed bug to fix or a desired enhancement to the software. Issues can be assigned to team members by team or project leaders or developers can assign themselves to work on certain issues.

Large or complex issues, which need to be broken down into smaller tasks, are called *epics*. Associated with every epic is a number of smaller issues. For example, new features in UCloud commonly become an epic.

Roadmap

New features described by epics are placed into the project roadmap with a starting date and deadline. The goal of the roadmap is to provide an easy way to plan the work needed to complete the project in a timely manner. The roadmap provides an overview of the whole project and a detailed always up-to-date view of the scheduled work for the next 3-6 months.

All tasks/issues go through a fixed set of stages: “backlog”, “in progress”, “testing”, “review” and finally “closed”.



Backlog

The issue's life begins when it enters the *backlog*. The backlog is a list of tasks ordered by priority, and it describes the next tasks/issues to work on. For example, a critical bug will have a high priority and be put on top of the backlog as soon as it is created, even if many other tasks were created before it.

In progress: Design & Development

The software design begins when the work on an issue starts. Once a developer starts working on it, the issue enters the "in progress" stage. New requirements are often discovered during the lifetime of an issue. These new issues enter the backlog as any other issue. Epics corresponding to bigger features receive detailed sub-issues and developers are assigned to individual issues.

Initial testing

Eventually code will reach a functional stage. At this point the issue becomes "ready for initial testing". In this stage, the code is tested by the assigned developer. Code is tested both manually and automatically. Automatic tests are executed by Jenkins and include both unit and integration testing. Manual testing is performed on a development system. The development system contains a software and hardware stack similar to the one used in the production environment. This allows us to more accurately test code.

Code review

After the testing stage, the assigned developer will submit a pull request and the issue enters the “code review” stage. The code is reviewed by one or more developers in the team knowledgeable of the affected code. This typically includes the team leader. The review causes a feedback loop between reviewers and the developer. Once the reviewers accept the proposed change, the code is merged into the master branch and the associated issue is closed.

Staging and Alpha testing

Changes made to the software base of UCloud through issues are bundled together to form a *release-candidate*. This release-candidate is deployed to the *staging environment* of UCloud. The staging environment is similar to both the development and production environment. This release candidate goes through internal *alpha testing*. New issues may arise from alpha testing, which are inserted in the backlog. Depending on the nature of the issue, these may block the release candidate from release or can simply be dealt with in later releases. Once the release candidate has passed internal alpha testing, it is deployed to the production environment.

Deployment

The UCloud software is deployed using Kubernetes. Kubernetes is a very flexible container orchestrator system which is configured using Kubernetes “resources”. These resources define the desired state of the cluster and it is the job of Kubernetes to always match this state on the available hardware. The code of each microservice contains the code needed for deploying itself to a Kubernetes. As a result, this code goes through the same development and review process as all other code. New big features are introduced in production as “beta” and a testing phase by the users starts. Bugs reports are submitted by the users via the integrated bug reporting feature in the web UI. After the beta testing is complete, the feature is promoted to *stable*.

Code and project management

As UCloud, this project will be hosted on GitHub as a public repository under an open-source license. We use the GitHub issue system to create and close issues. We use ZenHub as a project management tool, which integrates seamlessly with Github, to track the various stages of the lifecycle of an issue and to manage both epics and the project roadmap.

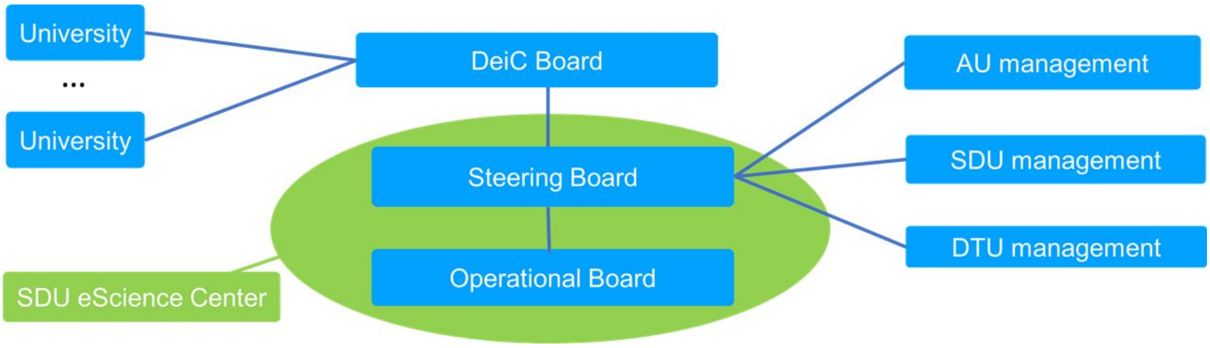
Documentation

Technical documentation of UCloud is kept in the same repository as the source code. This means that all documentation goes through the same review mechanisms as all other code. Technical documentation is automatically published via GitBook. User documentation is kept in a separate repository, and it will be provided in collaboration with the back-offices at the national HPC centers.

Management and governance

The partners in the consortium are SDU, DTU and AU. The governance structure of the consortium will partly overlap with the existing governance structure of SDU's eScience Center, which will be enlarged to also include representatives from DTU and AU. The governance model is illustrated in the diagram below.

The Steering Board will represent the management of SDU, DTU and AU. It will set long-term strategic directions for the consortium based on feedback and input from the DeiC Board. The director of the SDU eScience Center will be a member of the Steering Board ex-officio.



The Operational Board will oversee the execution of the project and take appropriate actions to ensure that the goals set up by the Steering Board are met. The Operational Board will be chaired by the director of the SDU eScience Center, professor Claudio Pica.

The secretariat of the SDU eScience Center will provide administrative support for both boards.

Finally, all universities can provide feedback to the center through the DeiC Board to ensure the best possible user experience and optimal long-term investments.

The DeiC Board representative from SDU is Dean Henrik Bindslev, From DTU University Director Claus Nielsen and the representative from AU is University Director Arnold Boon.

Key persons in the organization

- **Prof. Claudio Pica**, Director of the SDU eScience Center, Head of Section “Computational Science” at IMADA
- **Prof. Martin Svensson**, Head of Department of Mathematics and Computer Science (IMADA), member of SDU IT security committee
- **Associate Professor Kristoffer Laigaard Nielbo**, Head of Center for Humanities Computing Aarhus, Member of Operational Board for the project
- **Head of section Michael Rasmussen**, AIT Research IT, DTU
- **HPC Specialist Consultant Martin Rosgaard**, AIT Research IT Software Engineering, DTU
- **Special Consultant Dan S. Thrane,** team leader for Cloud Services at SDU eScience center