HORIZON2020 European Centre of Excellence
Grant Agreement n. 824143

D5.5

Report on the scalability of the Materials Cloud Jupyter section; code coverage and adoption of Quantum Mobile

Carl Simon Adorf, Nicola Marzari, Elsa Passaro, Giovanni Pizzi, Christopher J. Sewell,and Aliaksandr Yakutovich

Due date of deliverable: 30/11/2021
Actual submission date: 30/11/2021

Lead beneficiary: EPFL (participant number 6)
Dissemination level: PU - Public

Document information

Project acronym:	MaX
Project full title:	Materials Design at the Exascale
Research Action Project type:	European Centre of Excellence in materials modelling, simulations and design
EC Grant agreement no.:	824143
Project starting / end date:	01/12/2018 (month 1) / 31/05/2022 (month 42)
Website:	www.max-centre.eu
Deliverable No.:	D5.5

Authors: C.S. Adorf, N. Marzari, E. Passaro, G. Pizzi, C.J. Sewell, and A. Yakutovich

To be cited as: C.S. Adorf et al., (2021): Report on the scalability of the Materials Cloud Jupyter section; code coverage and adoption of Quantum Mobile. Deliverable D5.5 of the H2020 project MaX (final version as of 30/11/2021). EC grant agreement no: 824143, EPFL, Lausanne, Switzerland.

Disclaimer:

D5.5 Report on the scalability of the Materials Cloud Jupyter section; code coverage and adoption of Quantum Mobile

Content

1 Executive summary

An important goal of MaX is making access to advanced simulation tools for materials properties accessible, easy and straightforward to use for a broad class of users. These include not only experienced computational researchers (for which the goal is to reduce the technical burdens associated with running HPC simulations in high-throughput mode for tens of thousands of different materials), but also for non-experts (e.g. experimental scientists, industry, …), and for education purposes.

To this aim, within MaX we developed and currently maintain two platforms. The first is AiiDAlab (formerly known as the Materials Cloud Jupyter version), which makes it possible to provide simple GUIs for advanced AiiDA workflows. Thanks to AiiDAlab (that is based on Jupyter), any AiiDA workflow (e.g. all those described in D5.4, and in particular the common workflows described there) becomes easily accessible to anyone, directly in their browser, without the need to install any software. In the next sections we describe all results and improvements to the platform developed within the context of MaX, as well as one example application to run Quantum ESPRESSO simulations (relaxation of crystal structures, band structures etc.) with just a few clicks.

The second platform is Quantum Mobile, a virtual machine that includes all MaX flagship codes, including AiiDA preconfigured to run with all of them (as well as a few more codes). Quantum Mobile has already proven to be an outstanding platform to run courses and tutorials to teach how to use the codes. Indeed, having a virtual machine ready removes the complex steps of installing and setting up the codes on very diverse hardware (laptops and workstations typically owned by the students, with a variety of software, operating systems, libraries, …). Thanks to Quantum Mobile, tutorials can start in less than 30 minutes, going straight instead to the course content. The advantages provided by Quantum Mobile have become even more clear during the past few years marked by COVID-19, where most events have turned into virtual ones: without Quantum Mobile it would have been almost impossible to provide customized help to each participant on potential issues regarding the installation of software for the tutorials). In this report we describe the current state and updates to Quantum Mobile, the software installed in it, and its usage in schools and events.

2 General overview and status

The Materials Cloud Jupyter is now developed and distributed under the name "AiiDAlab''. A quick introduction to AiiDAlab with videos demonstrating the capability of the service can be found on the corresponding section of Materials Cloud [https://www.materialscloud.org/aiidalab]. The project provides an ecosystem that makes scientific workflows accessible and shareable. The AiiDAlab platform consists of multiple components that, when composed together, provide the full capability of the service. The base component of the whole concept is an AiiDAlab application. Typically, an application is a set of AiiDA workflows that encode simulation tasks, combined with graphical user interfaces (GUI) to manage them. The GUI part is based on Jupyter technology. The workflows might be a part of the application or can be installed as a Python dependency. To facilitate the distribution of applications we have developed a powerful App Store that relies on standard tools like Git and Pip. Once a developer registers an application in the AiiDAlab registry

[https://github.com/aiidalab/aiidalab-registry], the AiiDAlab users will get it listed in the App Store. In addition to that, the application management notifies the users about available updates in case that the application developer released a new version.

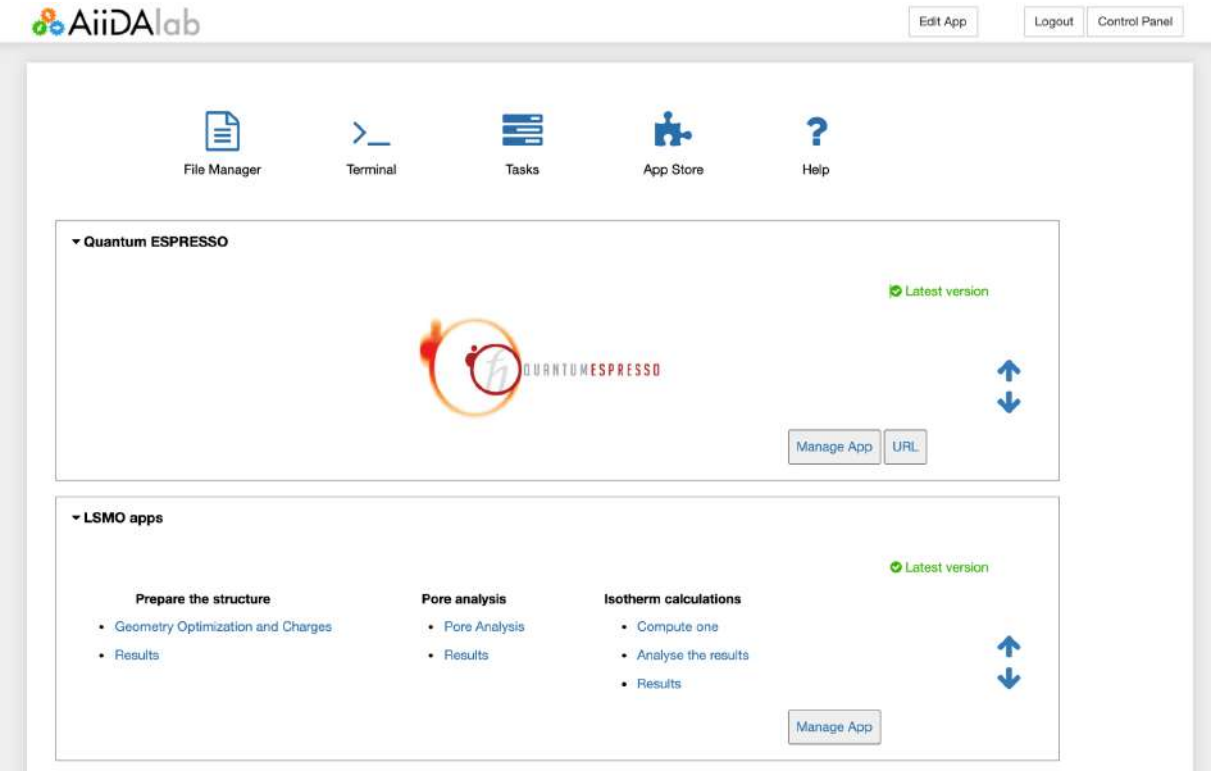

Fig. 1: Overview of the AiiDAlab home page.

Tasks like working with a 3D structure of a material, searching and downloading crystal structures from databases using the OPTIMADE API [https://www.optimade.org/] client, or setting up access to the computational resources are of general applicability. Therefore, we implemented them in the form of small widgets in the AiiDAlab widgets library [https://github.com/aiidalab/aiidalab-widgets-base]. Developers can simply reuse those when creating their application tailored for a specific computational task. Additionally, we provide a cookie cutter recipe [https://github.com/aiidalab/aiidalab-app-cutter] that helps kick off the application development process by automatically providing a scaffolding for a new app.

```
from aiidalab_widgets_base import CodQueryWidget, SmilesWidget, StructureExamplesWidget
from aiidalab_widgets_base import StructureBrowserWidget, StructureManagerWidget, StructureUploadWidget
from aiidalab_widgets_base import BasicStructureEditor
```

```
widget = StructureManagerWidget(
    importers=[
        StructureUploadWidget(title="From computer"),
        CodQueryWidget(title="COD"),
        StructureBrowserWidget(title="AiiDA database"),
        SmilesWidget(title="SMILES"),  # requires OpenBabel!
        StructureExamplesWidget(
            title="From Examples",
            examples=[
                ("Silicon oxide", "miscellaneous/structures/SiO2.xyz")
            ]),
    ],
    editors = [
        BasicStructureEditor(title="Basic Editor"),
    ],
)
display(widget)
```

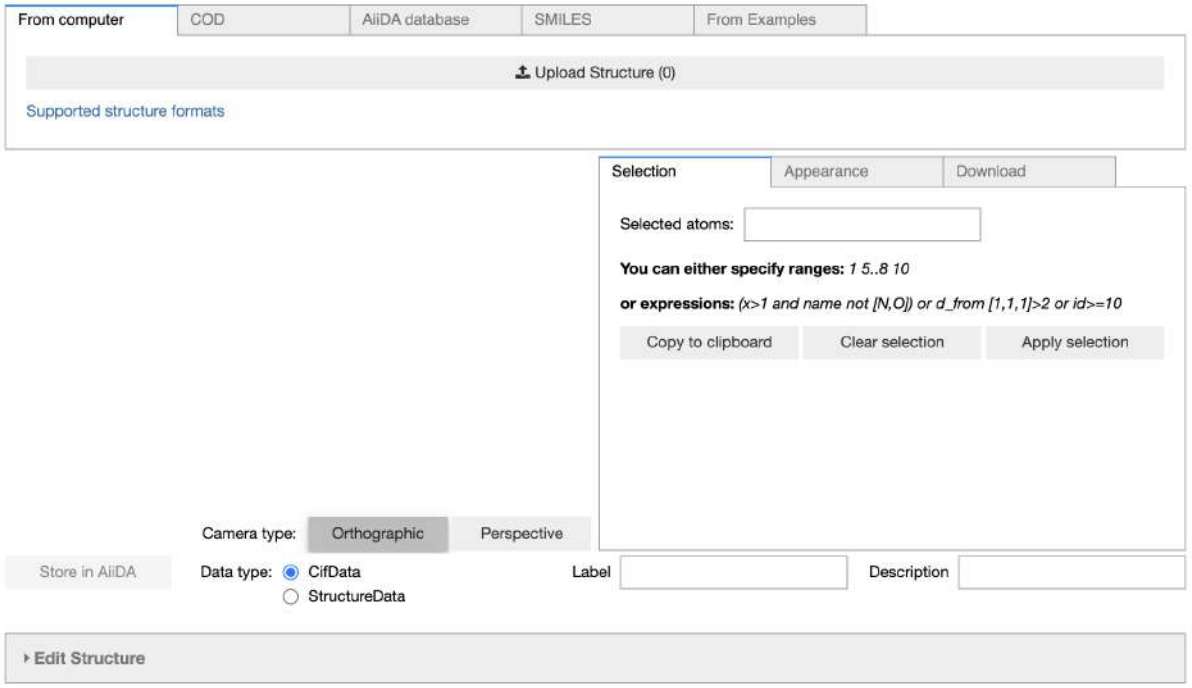

Fig. 2: Structure Visualiser/Editor widget available within the AiiDAlab Widgets package.

Although AiiDAlab is a collection of Python packages and Jupyter notebooks that can be installed directly on Linux or Mac OS, it is primarily distributed as a Docker image named AiiDAlab Docker Stack [https://github.com/aiidalab/aiidalab-docker-stack]. It also offers the possibility to configure which AiiDAlab applications should be pre-installed when the container is run for the first time. That allows us to more easily customize the deployment for a specific user group. Additionally, we provide the functionality to perform a factory reset: either to remove locally installed software and apps or remove everything including the data. The AiiDAlab Docker Stack is based on the AiiDA docker container [https://hub.docker.com/r/aiidateam/aiida-core] that is released automatically together with every release of AiiDA.

Until recently, installing the AiiDAlab application meant cloning the corresponding Git repository. The

Python dependencies of the application had to be provided together with the AiiDAlab Docker image. This approach had several drawbacks. First, the container was quite large in size. Second, we had to release the container quite often to keep up with the application developers to make sure to provide a working container. Any significant update of an AiiDAlab application would require the application developer to request a container update, which is impractical. Enabling such a dependency installation was a challenge as we had to make sure to properly resolve dependency clashes by different apps. In essence, this required a development of a proper package manager. Currently, AiiDAlab App Store is not only capable of installing the apps and their dependencies but also detecting when an application is incompatible with the current environment. It also informs the users about newer versions of the installed applications with a possibility to update them with one click only. All these features are implemented within the "aiidalab" application [https://github.com/aiidalab/aiidalab]. Additionally, the application provides all the necessary tools to create and serve an independent AiiDAlab application registry.

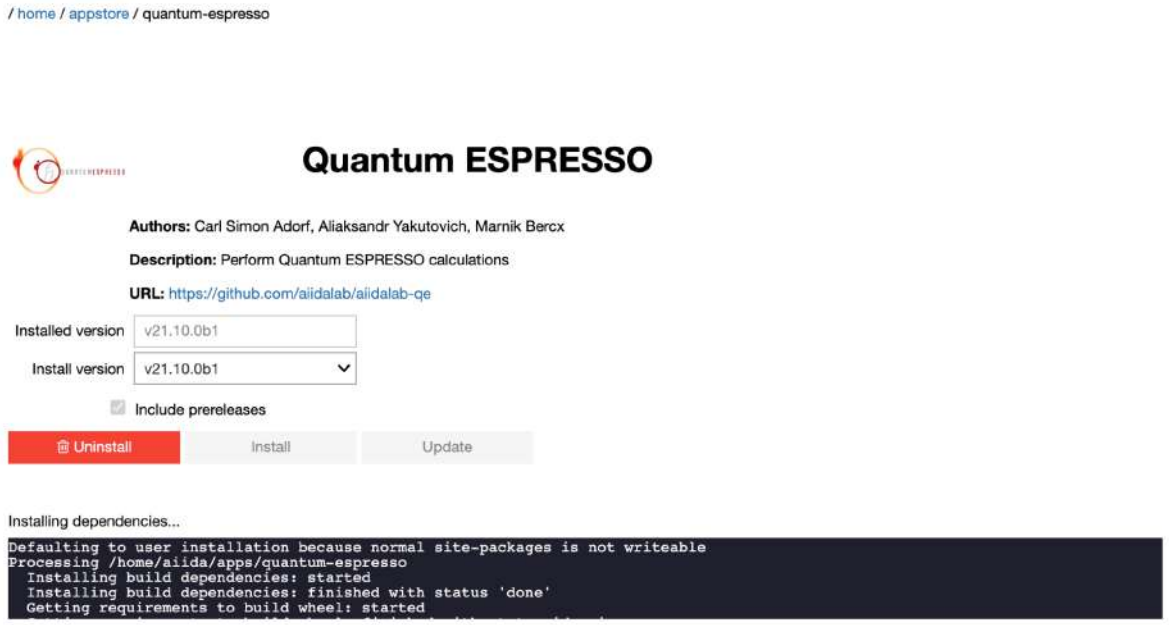

Fig. 3: Page of the App Store to install the Quantum ESPRESSO AiiDAlab application together with its dependencies.

In order to facilitate uptake by both users and stakeholder developers and institutions, we created comprehensive documentation for AiiDAlab and published it online [https://aiidalab.readthedocs.io/]. The documentation is divided into 4 main parts: user guide, app developer guide, administrator guide, and development guide. The division makes it easier for specific user groups to find documentation relevant to them. The user guide presents different ways to start using the platform. The app developer guide is there to explain the application development process: all the way from blank slate to registering the app in the AiiDAlab registry. The administrator guide explains how to create and maintain your own AiiDAlab instance together with your own application registry, and is relevant e.g. for research institutions, HPC centres or industries interested in (re)deploying AiiDAlab with their premises or networks. The development guide is intended for

developers of the AiiDAlab platform itself and covers topics such as the overall architecture, release workflow, etc.

3 Scalability and infrastructure

AiiDAlab has been further developed such that it can be deployed as "Quantum-as-a-Service" on highly scalable cloud resources, specifically on Kubernetes clusters without compromising the option of executing it as part of the Quantum Mobile VM [https://quantum-mobile.readthedocs.io/], directly on a local workstation or larger server via the Docker Engine, e.g. for on-premise installations. To automate a reproducible deployment of AiiDAlab in the Cloud we provide Terraform scripts [https://github.com/aiidalab/aiidalab-k8s] that can be used to create a Kubernetes cluster on AWS resources meeting the needs of one or multiple AiiDAlab deployments. Such deployments on AWS resources have been successfully employed as computational environments to be used by workshop participants, e.g., at AiiDA tutorials and similar workshops. Additionally, we maintain an Ansible role [https://github.com/aiidalab/ansible-role-aiidalab-server] to streamline the creation of a multi-user setup on a single server.

4 Applications

The optimized workflows for the Quantum ESPRESSO Density Functional Theory (DFT) suite were developed as part of the aiida-quantumespresso AiiDA plugin [https://github.com/aiidateam/aiida-quantumespresso] and can be directly run by AiiDAlab users through the Quantum ESPRESSO application (short: "QE app", https://github.com/aiidalab/aiidalab-qe). The QE app was completely redesigned from scratch, such that the interface is significantly more responsive and the overall user experience is now streamlined. In particular we adopted a "wizard" interface (see Fig. 4), meaning that users are guided linearly through the individual steps from structure selection, to workflow configuration, and the final results output via a series of consecutive forms. This structure makes it significantly easier and more efficient for users to provide all necessary inputs and to provide feedback on missing or invalid inputs.

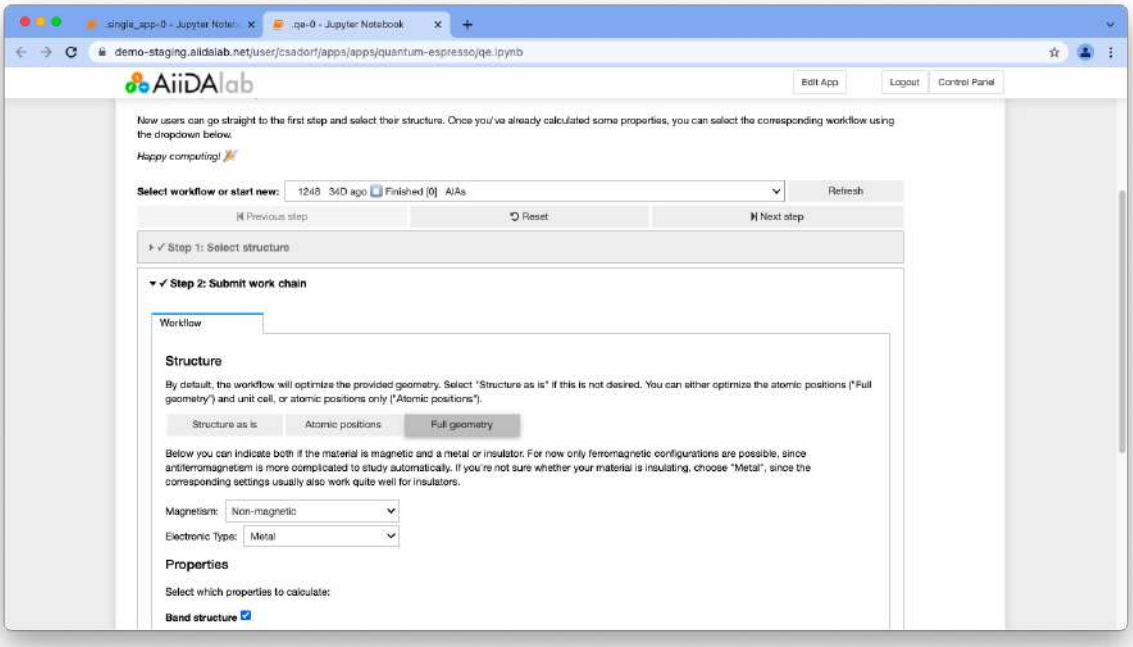

Fig. 4: The Quantum ESPRESSO AiiDAlab application (QE app) is structured through a wizard style sequence of forms, guiding the users through the individual steps from structure selection to results output and download. Here we show a completed work chain, with step 2 selected. The icons in the top left indicate the status of each step (here: check marks indicate that both step 1 and 2 are completed).

In addition to the direct upload and the selection of structures stored within a user's AiiDA database, structures can now also be directly selected from the OPTIMADE client interface (see Fig. 5). The client interface is an embedded version of the OPTIMADE client hosted on the Materials Cloud (https://www.materialscloud.org/work/tools/optimadeclient). This enables users to directly run DFT calculations for all structures that are available through a data provider that implements the OPTIMADE API (https://www.optimade.org/optimade), e.g. the Materials Cloud (https://materialscloud.org) or The Materials Project (https://materialsproject.org), without the need to download them locally and then re-upload them from their computer.

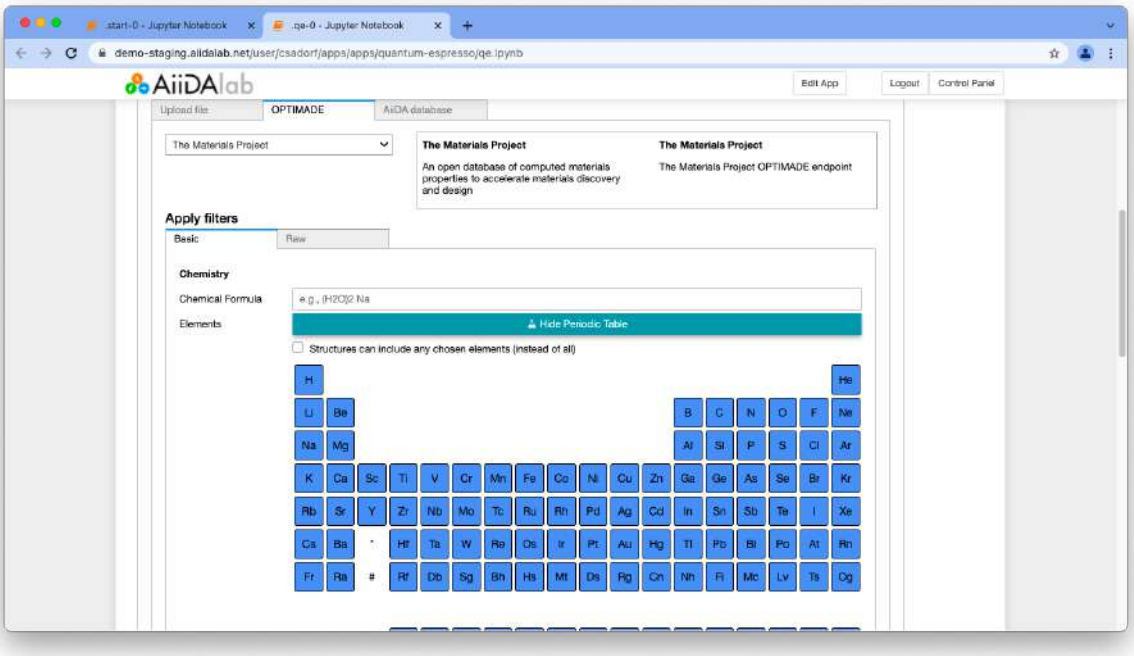

Fig. 5: The OPTIMADE client interface is directly integrated into the structure selection step of the QE app (step 1) making it very easy for users to run DFT calculations on structures available from data providers that support the OPTIMADE API. Here, we have selected the Materials Project Database for subsequent querying.

In addition, it is now possible to switch between multiple running workflows and retrieve previously completed workflow runs without the need to leave the app interface or to open a second browser window (see. Fig. 6). The workflow progress is visualized via a tree view (see Fig. 7) that provides detailed information about the execution status of individual calculation steps, and their respective outputs, including the terminal output. This allows the user to better understand what calculations are performed and how the overall workflow progresses.

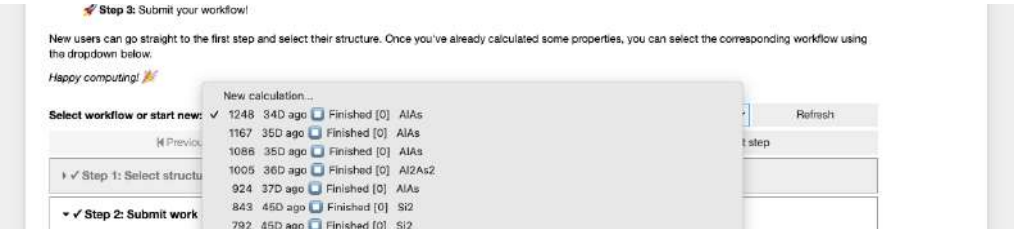

Fig. 6: The QE app work chain selector makes it easy to switch between multiple running work chains or retrieve results from previous runs and optionally restart a calculation with the same structure and different parameters.

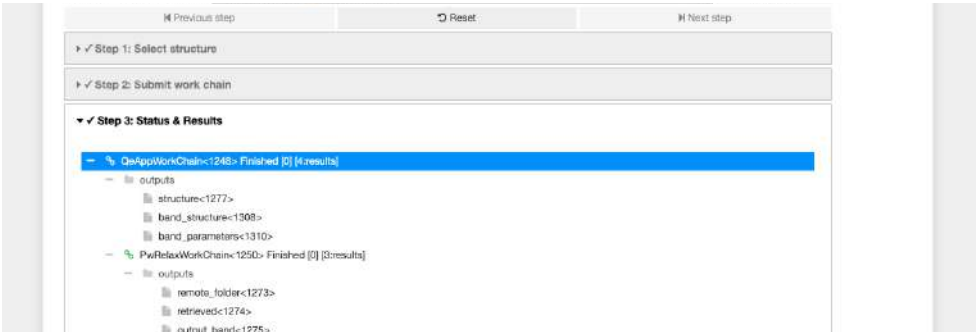

Fig. 7: The QE app process tree view visualizes the hierarchical structure of the underlying work chain and enables users to get both a quick overview of the overall execution status as well as inspect the progression and outputs of individual steps.

Finally, the results from band structure calculations are visualized via the integration of a much improved bands plot widget that automatically chooses appropriate zoom levels and allows for custom ordering of symmetry points (see Fig. 8).

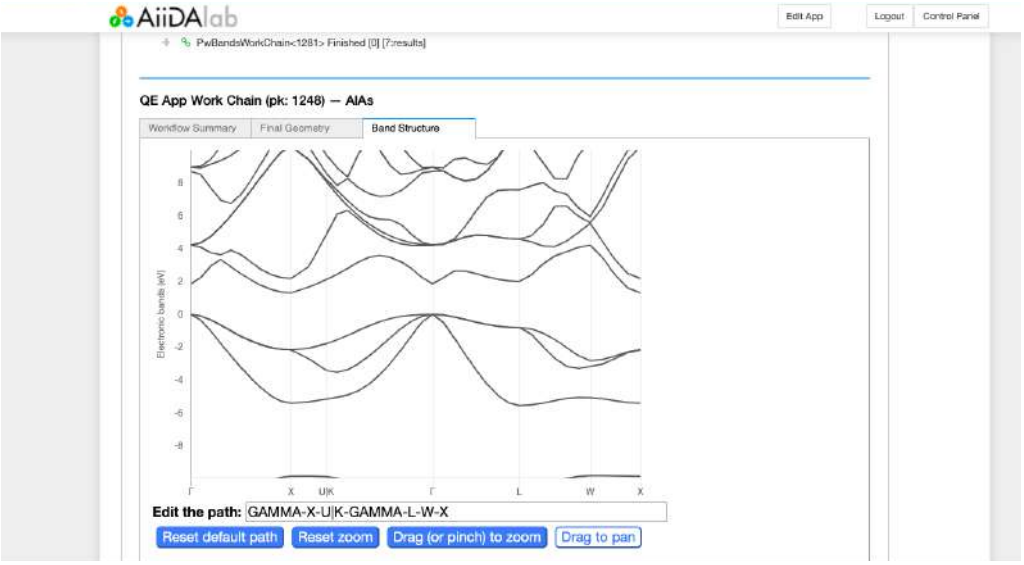

Fig. 8: The integrated bands plot widget allows users to directly visualize the band structure results in a very interactive fashion.

5 Interaction with the EU landscape

The production AiiDAlab machine [https://aiidalab.materialscloud.org] is deployed on the resources of Swiss National Supercomputing Centre (CSCS) and serves all users from the funding and supporting partners of Materials Cloud, so in particular MAX members as well as the other projects listed here: https://www.materialscloud.org/home#partners.

Additionally, AiiDAlab is fully integrated in the European EOSC. AiiDAlab is indeed registered as one of the services on the EOSC Marketplace (see Fig. 9).

Furthermore, we provide a demonstrator instance [https://aiidalab-demo.materialscloud.org] running on the kubernetes cluster hosted on EOSC resources, at CESNET. This deployment is supported by EGI-ACE [https://www.egi.eu/projects/egi-ace/] and, more specifically, by a EOSC-hub Early adopter Programme grant [https://eosc-hub.eu/research-communities/open-aiida-lab-platform-cloud-computing-materials-science]. Any user from an academic or research institution can login thanks to our pairing with the authentication server provided by EGI (EGI Check-in).

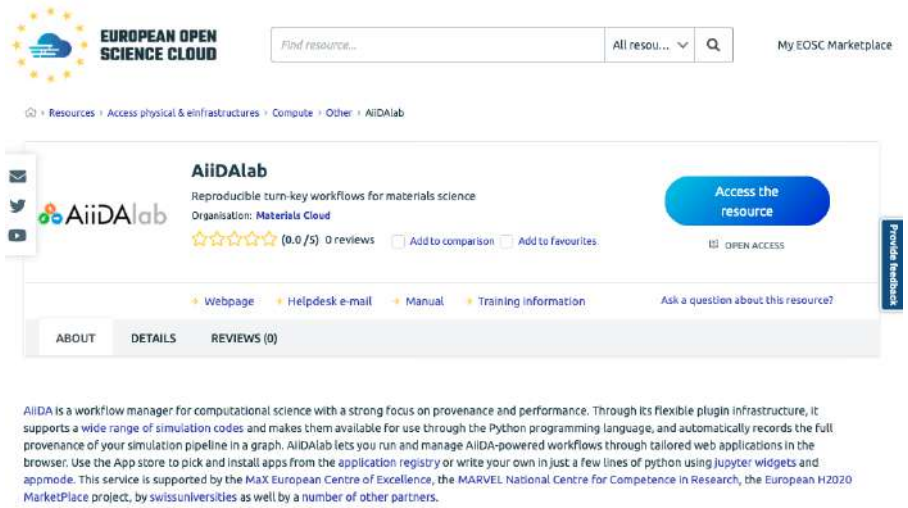

Fig. 9: AiiDAlab is provided as a service inside the EOSC Marketplace. https://marketplace.eosc-portal.eu/services/aiida-lab

Finally, to simplify connections to the supercomputing centers, we started to develop a computer and code database [https://github.com/aiidateam/aiida-code-registry] where the necessary AiiDA configurations are provided. The database makes it significantly easier for users to set up remote connections and codes with tested configurations directly through the AiiDAlab interface, and provides a first step to making access to current and future (pre-)exascale HPC centers much easier for users and researchers. An example of this can be easily used to set up the connection to the Daint supercomputer using the simple GUI offered in AiiDAlab is shown in Fig. 10.

6 Use in schools and tutorials

The AiiDA virtual tutorial 2021 [https://aiida-tutorials.readthedocs.io/en/tutorial-2021-intro/] was fully run on the AiiDAlab instance deployed on AWS resources. Additionally, the hands-on sessions of the Computational methods in Chemistry and Chemical Engineering course at EPFL are done using the AiiDAlab LSMO application.

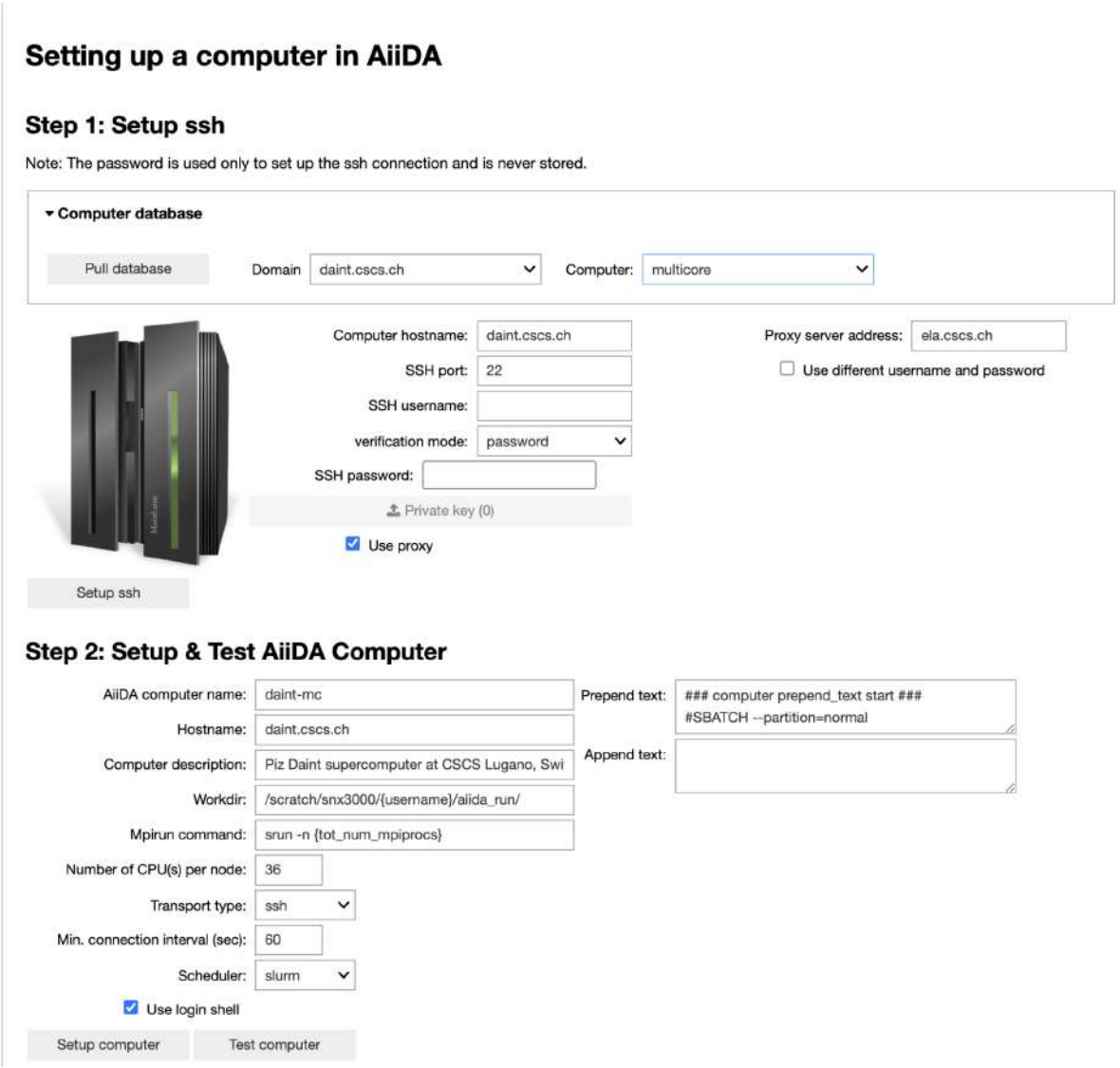

Fig. 10: Configure access to Piz Daint supercomputer (CSCS, Switzerland) using the AiiDAlab app to set up new computers; the details were pulled automatically from our centralized registry ("Computer Database") of computers and HPC centres.

7 Code coverage and adoption of Quantum Mobile

Quantum Mobile is a Virtual Machine (VM) solution to provide a uniform environment for quantum-mechanical simulations and to make it easy for users to access advanced HPC and high-throughput capabilities. The VM provides a full Linux Ubuntu distribution, including a wide range of open-source quantum mechanical simulation codes, ready to be used either directly or through AiiDA, and additional applications such as the SLURM job scheduler and Jupyter Notebook. Once the virtual machine is started, the user is offered with a standard Ubuntu Desktop, as shown in Fig. 11.

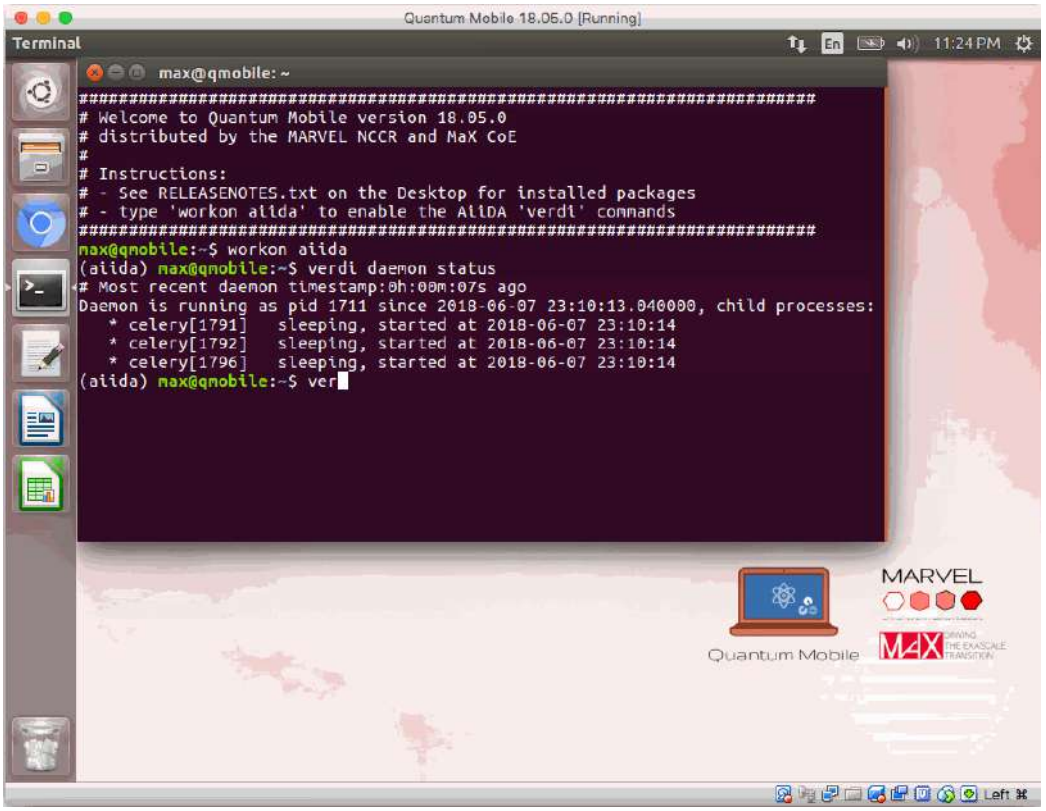

Fig. 11: Quantum Mobile Virtual Machine interface.

Quantum Mobile has been implemented with a fully autonomous build process (utilising the open-source Red-Hat Ansible automation platform [www.ansible.com]), allowing for a modular approach to updating existing codes and adding new components. This allows for a release schedule of approximately every six months to update the VM with cutting-edge versions of each simulation code, and also for users to generate their own custom-builds with only a subset of simulation packages.

In the last 18 months, Quantum Mobile has undergone a major update, in conjunction with the AiiDA Common Workflows project (see deliverable D5.4). In particular, recent versions now include, beside all MAX flagship codes, also a number of other open-source DFT simulation codes (see table 1). Moreover, by releasing the common workflows on Quantum Mobile, the results documented in the

Common Workflow scientific article [1] are fully and completely reproducible in an easy way through Quantum Mobile, without the need of installing additional software.

Additional improvements include compatibility upgrades to the modern Ubuntu 20.04 distribution, the implementation of adaptive SLURM job scheduling based on the available CPU resources, and the integration of AiiDAlab.

Simulation Code	Code Version	AiiDA Plugin	Plugin Version
Abinit*	9.2.1	aiida-abinit	0.2.0
BigDFT*	1.9.1	aiida-bigdft	0.2.6
CP2K	7.1	aiida-cp2k	1.3.0
NWChem*	7.0.2	aiida-nwchem	2.0.0
Fleur	0.30 MaX4	aiida-fleur	1.1.4
Quantum Espresso	6.7 MaX	aiida-quantumespresso	3.4.1
Siesta	1.2.0 MaX	aiida-siesta	1.1.0
Yambo	4.5.2	aiida-yambo	1.1.3
Wannier90	3.1.0	aiida-wannier90	2.0.1

Table 1: Overview of the simulation codes included in QM v21.06.04 (* indicate new codes).

In addition to these VM updates, a modern website has been created with comprehensive documentation aimed to onboard both users and developers of custom builds; the new website is available at https://quantum-mobile.readthedocs.io (see Fig. 12).

Quantum Mobile has proved to be essential as a teaching resource: since May 2018, Quantum Mobile has been downloaded more than 10'000 times and has been used in 12+ workshops, university courses and tutorials [2-9], with extremely positive feedback both from students and lecturers. A number of enthusiastic testimonials can be found on the Quantum Mobile website.

Therefore, Quantum Mobile is becoming an established platform to run courses and tutorials in the field of Materials Science, removing the burden of software installation at the beginning of a tutorial, and allowing students to focus first on learning the science and the code, rather than the technicalities associated with compiling and optimizing codes.

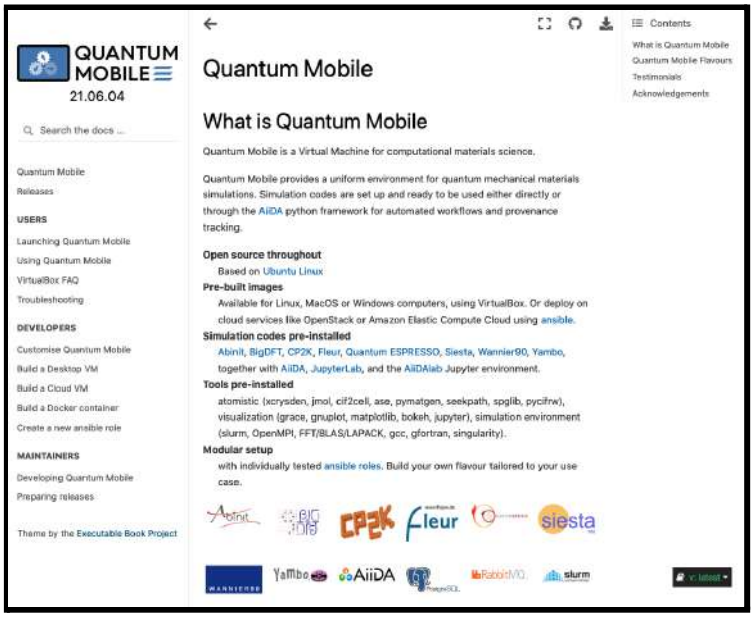

Fig. 12: Quantum Mobile website front-page.

8 Conclusions

AiiDAlab is now a robust platform for running simulations “on the cloud”, where the workflow engines and the data live on cloud servers and can be accessed from the browser. It provides easy ways to create GUIs (using jupyter notebooks and python syntax), making it possible to create GUIs to make advanced high-throughput HPC simulations and data analysis very easy to run, and straightforward to use. In combination with the workflows described in D5.4, MaX is creating all the tools to make not only data FAIR, but also access to simulation FAIR (in particular, accessible). Combined with the Quantum Mobile virtual machine, access to the simulation codes becomes very easy also for students and in schools and tutorials, providing a platform that actively contributes to MaX’s goals on education, training and dissemination.

References

[1] S. P. Huber et al., npj Comput. Mater. 7, 136 (2021): https://doi.org/10.1038/s41524-021-00594-6

[2] Understanding Advanced Molecular Simulations (2018, EPFL, Switzerland): https://edu.epfl.ch/coursebook/en/understanding-advanced-molecular-simulation-CH-420

[3] Computational Methods in Molecular Quantum Mechanics (2018, EPFL, Switzerland): https://edu.epfl.ch/coursebook/en/computational-methods-in-molecular-quantum-mechanics-CH-452

[4] MolSim school (2019, Amsterdam, Netherlands): http://www.acmm.nl/molsim/molsim2019/

[5] Molecular and Materials Modelling (2018, ETH Zurich, Switzerland): http://www.vvz.ethz.ch/Vorlesungsverzeichnis/lerneinheit.view?lerneinheitId=120724&semkez=2018S&ansicht=KATALOGDATEN&lang=en

[6] Computational Materials Physics online course (University of Ghent, Belgium): https://compmatphys.org/

[7] Ethiopia School on Electronic Structure Methods (2019, Addis Ababa, Ethiopia): https://agenda.ethernet.edu.et/event/33/

[8] Computational School on Electronic Excitations in Novel Materials Using the Yambo Code (2020, Trieste, Italy): http://indico.ictp.it/event/9018/

[9] Wannier90 v3.0 school, virtual edition 2020: https://www.materialscloud.org/learn/sections/C8ad91/wannier90-v3-0-school-virtual-edition-2020