education sciences

MDPI

Article

Cloud-Based ICME Software Training

Lukas Koschmieder, Ralph Altenfeld, Janin Eiken, Bernd Böttger and Georg J. Schmitz *

ACCESS e.V., 52072 Aachen, Germany; L.Koschmieder@access-technology.de (L.K.);
R.Altenfeld@access-technology.de (R.A.); j.eiken@access-technology.de (J.E.);
b.boettger@access-technology.de (B.B.)
* Correspondence: g.j.schmitz@access-technology.de

Abstract: Hands-on type training of Integrated Computational Materials Engineering (ICME) is characterized by assisted application and combination of multiple simulation software tools and data. In this paper, we present recent experiences in establishing a cloud-based infrastructure to enable remote use of dedicated commercial and open access simulation tools during an interactive online training event. In the first part, we summarize the hardware and software requirements and illustrate how these have been met using cloud hardware services, a simulation platform environment, a suitable communication channel, common workspaces, and more. The second part of the article focuses (i) on the requirements for suitable online hands-on training material and (ii) on details of some of the approaches taken. Eventually, the practical experiences gained during three consecutive online training courses held in September 2020 with 35 nominal participants each, are discussed in detail.

Keywords: software training; simulation workflows; SimPhoNy; Simphony-Remote

check for updates

Citation: Koschmieder, L.; Altenfeld, R.; Eiken, J.; Böttger, B.; Schmitz, G.J. Cloud-Based ICME Software Training. *Educ. Sci.* **2021**, *11*, 5. https://dx.doi.org/10.3390/educsci11010005

Received: 1 December 2020
Accepted: 21 December 2020
Published: 24 December 2020

1. Introduction

The occurrence and pandemic spread of the Coronavirus in 2020 drastically changed social and professional life. Established on-site training formats could no longer be supported since personal contact between tutors and participants had—and still has—to be limited.

Especially hands-on type training events, where the participants are assisted in direct interaction with dedicated hardware and software has become challenging. While training on physical machinery naturally requires on-site presence, software training can alternatively be performed online with remote access. In fact, a number of software companies already provided online tutorials, webinars, and even hands-on training for their own, individual software tools well before the COVID-19 crisis.

Here we report on a hands-on software training course in the area of Integrated Computational Materials Engineering "ICME" held in late 2020. ICME and its use in materials and process design drastically decrease the development times of new products. By its name and by its nature, ICME draws on the combination of a diverse variety of models and software tools. Individual, monolithic tools are unlikely to be capable of tackling the host of phenomena occurring during the processing and service life-cycle of materials. Accordingly, there is a need for modular and freely configurable workflows orchestrating the simultaneous or consecutive execution of a variety of tools [1].

A variety of simulation platforms currently is emerging in the area of Computational Materials Science and Integrated Computational Materials Engineering. Prominent examples are nanoHUB [2], Materials Cloud [3], AiiDAlab [4], pyiron [5], AixViPMaP [6], and commercial platforms like SimStack (https://www.simstack.de), MedeA (https://www.materialsdesign.com), 3DEXPERIENCE (https://www.3ds.com/3dexperience), ANSYS Workbench (https://www.ansys.com), SIMIT (https://www.siemens.com) to name a few. Except for SimStack and MedeA, which focus on electronic/atomistic/mesoscopic models, most other commercial platforms serve for continuum models at the scale of components rather than for modeling of materials. The non-commercial platforms can be classified according to (i) the used hardware (high-performance computing (HPC), local installations on workstations, grid-based operations, or deployment in the cloud), (ii) the area of application (electronic, atomistic, mesoscopic, or continuum models), (iii) the types of software codes being run on the platform (commercial versus open access), (iv) the availability of a collaborative dataspace shared by the different tools and allowing for information exchange, (v) the availability of a workflow management tool orchestrating the sequential or parallel execution of different tools (vi) the use of a common namespace/ontology for all tools (ongoing work e.g., in the MarketPlace project).

The Simphony-Remote (S-R) infrastructure being developed and used for cloud-based ICME training depicted in the present article is a free and open-access platform infrastructure. In contrast to many other platforms, which draw on abundant open-access codes being freely available, the electronic/atomistic/mesoscopic domain, the S-R integrates both commercial and open-access applications originating from the world of continuum models of microstructures and thermodynamics. It provides a common dataspace shared by the different applications, which can be orchestrated, for example, by Jupyter notebooks (Interactive data science environment. http://jupyter-notebook.readthedocs.io). Jupyter notebooks seem to emerge as a standard to orchestrate workflows, for example, in AiiDAlab, AixViPMaP, pyiron. The unique feature of the S-R infrastructure being depicted in the present article, however, is that the entire platform including all its apps can be preconfigured and be deployed in the cloud. It can be operated from a normal web browser, which seems a highly valuable asset for educational purposes.

A major challenge to be met for this training was to provide a platform not limited to own software, but giving access to third-party tools—both commercial and open access. Respective platform concepts are not yet readily available, but currently under development, for example, in the European MarketPlace Project (http://the-marketplace-project.eu). The training course depicted in the present article is one of the early applications of such concepts under real-life education conditions.

The major differences between our regular on-site and the *online* training presented in this article are summarized in Table 1.

Table 1. Major differences between our on-site and online training courses.

Topic	On-Site	Online
Location	Physical site	Webconference
Hardware	One workstation per participant, located in a seminar room, local login	One virtual machine per participant, hosted by a cloud provider, web login
Software	Multiple software applications, installed on each workstation individually	Multiple software applications, deployed in a single virtual disk image
Training material	Presentation slides, printed tasks, all content being handed out after the training on a USB stick	PDFs, videos, interactive notebooks, all content being downloadable from a website
Social events	Coffee breaks, lunch breaks, social dinner	None
Expenses	Travelling and accommodation costs, participation fee	None
Class	Three tutors, max. 12 participants	Five tutors, max. 35 participants per group, three groups
Duration	Three days	Two days of six hours each per group

2. Infrastructure

The online training was intended to serve as a proper alternative to regular training that had been held on-site for many years. Particular attention during on-site training had been paid to the possibilities of holding lectures and accompanying exercises. Especially missing personal feedback during breaks and social events and the lack of direct monitoring of the participant's progress were anticipated to be drawbacks of the online training.

Therefore, the infrastructure for the training required a proper communication tool in the first place.

The great variety of software applications that had to be made available to the participants posed a further challenge. In particular, the questions of how to provide software licenses for commercial software and how to make an overall viable software infrastructure available had to be answered.

2.1. Communication Tools

Microsoft Teams (Microsoft Teams. Proprietary business communication platform. https://www.microsoft.com) (MS Teams) was used for communicating with the training participants. Besides its widespread use—especially since COVID-19—there was no dedicated evaluation of other possible tools suitable for the training purpose. MS Teams provides functionality for general meetings and chats, but also allows for one-to-one/group video calls, file exchange, and more. Especially, the latter functions were used in order to provide a file system comprising all tutorial materials and to allow for bilateral communication. Using these functionalities, however, requires an advance registration of the participants as team members. Guest links are only sufficient to join a meeting and to watch presentations. Some of the participants were not used to MS Teams or were not able to manage the MS Teams registration. In these cases, additional email communication was necessary to help with registration or to provide information during the training.

2.2. Software Infrastructure

The software infrastructure for hands-on training was based on Simphony-Remote (S-R). S-R has been developed in a series of European projects and is available for free on GitHub (SimPhoNy project repository. http://github.com/simphony) now. S-R is a web service allowing users to run web and desktop applications in a collaborative environment on a remote computer. Unlike remote desktop software, S-R does not grant access to the desktop environment on the remote computer. Instead, it exposes a set of software applications through a web user interface. S-R is compatible with most modern web browsers and does not require the installation of any additional software on the computers of the users. Administrative tasks such as (i) maintaining the software applications and licenses, (ii) creating user accounts, or (iii) granting users access to the file system on the remote computer, are handled by an expert user.

The layout of the web user interface consists of three components, Figure 1:

(i) A main area in the center of the screen provides access to the graphical user interface of a software application (app).
(ii) A vertical bar on the left-hand side shows a list of all available apps and gives users the ability to switch between them.
(iii) A horizontal bar on the top provides contextual functionality such as sharing the screen/controls of some app with another person (dropdown menu not depicted).

Once started, apps keep running on the remote computer until explicitly stopped by the user. Running apps are affected neither by accidental closing of the web browser nor by temporal loss of network connectivity, only to name two problems likely to occur during an online session. Having the option to run unattended software processes also enables the adaptation of a common training use case, in which computationally intense simulations are configured during one training session while the results are examined during another one.

The S-R software framework is very flexible in terms of the integration of software applications. Arguably, most desktop applications can be configured to run on it in case they support the Linux operating system. However, there are chances that also non-Linux applications may be run using additional compatibility software like Wine (Wine. Windows runtime system compatibility layer for Linux. http://winehq.org). In addition to desktop applications, S-R may also serve many popular web applications such as Jupyter. In either case, a software adapter must be created to allow users to run a particular software application on S-R. A growing number of such software adapters are available in the public S-R project repository. Most of the software applications, however, are related to the field of atomistic modeling and simulation, and not all their adapters are up-to-date. In preparation for the online training, several new software adapters were created targeting the domain of continuum models for microstructure evolution simulation, Table 2.

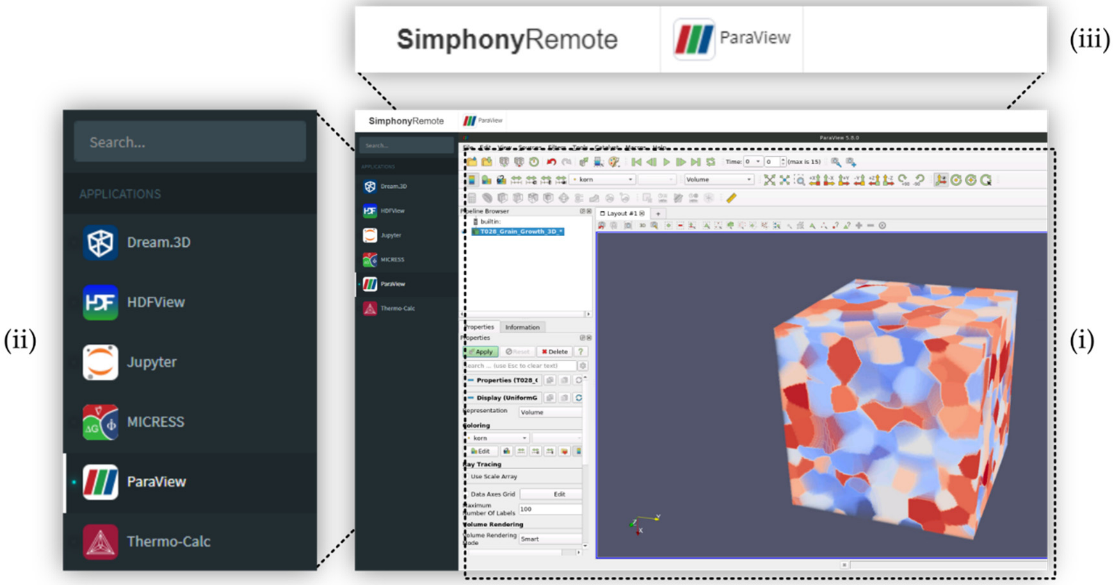


Figure 1. Example of Simphony-Remote (S-R) running a graphical desktop application for scientific data visualization (**i**). The depicted cube-shaped object portrays the results of a grain growth simulation and demonstrates the platform's ability to exchange data between different software applications. The adjacent user interface components are not relevant in this context and can be ignored. The dark menu on the left-hand side (shown enlarged in (**ii**)) shows the names and logos of all software tools that were implemented in S-R and thus available to each training participant. The stripes icon in the top menu (shown enlarged in (**iii**)) represents a running software application and reveals further options when clicked, for example, for terminating the app.

Table 2. Simphony-Remote (S-R) components including the core software (above dotted line) and the application adapters created for the hands-on training (below dotted line).

Software Component	Description
simphony-remote	S-R web service
simphony-remote-docker-base	S-R base images
simphony-remote-docker-scripts	S-R image build system
simphony-remote-docker-dream3d	Dream.3D [1] S-R image (vncapp)
simphony-remote-docker-hdview	HDFView [2] S-R image (vncapp)
simphony-remote-docker-jupyter	Jupyter [3] S-R image (webapp)
simphony-remote-docker-micress	MICRESS [4] S-R image (vncapp)
simphony-remote-docker-paraview	ParaView [5] S-R image (vncapp)
simphony-remote-docker-thermocalc	Thermo-Calc [6] S-R image (vncapp)

[1]: Dream.3D. Scientific data analysis and processing software. http://bluequartz.net; [2]: HDFView. HDF5 file format viewer. http://hdfgroup.org; [3]: Jupyter Notebook Server. Interactive data science environment. http://jupyter.org; [4]: MICRESS. Microstructure evolution simulation software. http://micress.de; [5]: ParaView. Scientific data visualization software. http://paraview.org; [6]: Thermo-Calc. Thermodynamic database/calculation software. http://thermocalc.com.

The task of creating a software adapter requires a technical understanding of the S-R software framework and should therefore be ideally performed by a skilled system administrator. Respective software adapters comprise (i) a computer program to automatically install a given application, (ii) a computer program to automatically start the user interface of the software application, (iii) detailed information on the software application, for example, a comprehensive description of the software application and its purpose. In terms of the installation of software applications, S-R basically offers a similar degree of freedom as the underlying Linux operation system. For example, the installation process may involve the integration of a license management system for commercial applications or some additional auxiliary software, for example, to account for compatibility. Although software adapters are typically supposed to wrap a single application, there is no technical limitation that requires following this practice. If needed and meaningful, even an entire desktop environment may be deployed in a single S-R app.

S-R software adapters are based on so-called Docker (Operating-system-level virtualization software. http://docker.com) images. The Docker mechanism allows starting software processes in an isolated operational environment, which encapsulates all computer resources of a given software application. S-R offers two Docker base images, which serve as templates for creating software adapters.

1. A *vncapp* base image is used to create adapters for desktop applications. The name is derived from the VNC (Virtual Network Computing). Remote desktop software. http://tools.ietf.org/html/rfc6143) technology used to handle remote access. A software adapter for desktop applications will first initialize an X11 session (LXSession. Session manager for the LXDE desktop environment. http://lxde.org) in a virtual framebuffer (X Virtual Framebuffer (Xvfb). http://x.org). It will then start a VNC server (x11vnc. VNC remote access server. http://github.com/libvnc) to enable remote access to the X11 session. Finally, a web server (NGINX. Web server providing reverse proxy functionalities. http://nginx.com) will be launched, connecting a VNC web client (noVNC. VNC remote access web client. http://github.com/novnc) to the VNC server.
2. A *webapp* base image is used to create adapters for web applications. It is less complex consisting of only a reverse proxy redirecting incoming requests to a specific port that the wrapped web application is expected to listen to.

Although knowledge of internal mechanics is beneficial, in many cases it is not necessary to understand every single bit to be successful in creating an application adapter for S-R. This is especially true for open-source applications which usually can be easily installed in Linux via a package manager such as Conda (Python package manager. https://conda.io).

S-R is based on the JupyterHub (JupyterHub. Multi-user hub for Jupyter Notebook Server. http://jupyter.org/hub) architecture, which offers many options for user authentication, e.g., via LDAP (Lightweight Directory Access Protocol (LDAP). http://tools.ietf.org/html/rfc4511) or OAuth (Web authentication system. http://oauth.net). JupyterHub includes a spawner, a software component responsible for launching the interactive user environments. The default spawner relies on system user accounts to isolate the interactive environments of different JupyterHub users. This mechanism is redundant in S-R because an even higher level of isolation is achieved encapsulating interactive environments in Docker containers. Therefore, S-R offers an alternative virtual user spawner that does not depend on system users and thus enables additional flexibility in terms of setup.

During ICME training, whether conducted as on-site or online training, it is most important that participants can familiarize themselves with the training environment as fast as possible in order to be able to dive into the exercises. In contrast to classical on-site training, during the online training, all participants had to be introduced to the use of the S-R platform beforehand. They had to learn how to start and stop applications, how to switch between different applications, how to transfer data between applications, how to download data to their local computers, and much more. Based on the relatively small number of technical support requests (only two to three per group), most participants were able to internalize the usability concept of S-R after only 30 min of introduction. This observation is also supported by the results of a System Usability Score (SUS) [7] study, which was performed after the training and is detailed in Section 5.

2.3. Hardware Infrastructure and Deployment

The hardware infrastructure for operating S-R was rented from a cloud provider (DigitalOcean. Cloud service provider. http://digitalocean.com) in view of a large number of registrations and the lack of sufficient computer hardware on the local site. Each participant was provided with an own individual virtual machine comparable to the well-proven hardware used during regular on-site training, effectively accounting for 8 central processing units (CPUs), 32 gigabytes of random-access memory (RAM), and 100 gigabytes of solid-state drive (SSD) per participant. Each user was sent a unique link to his/her personal virtual machine via email shortly before the training. The user accounts were anonymous and not linked to the participants' identities. An alternative strategy would have been to run multiple users on a single virtual machine. This concept was however dropped for multiple reasons:

- Even the largest virtual machines available on the cloud market were not able to serve more than just a few participants.
- A more complex authentication system would have been needed to handle multi-user login and, therefore, the assignment of participants to virtual machines would have resulted in increased organizational overhead.
- Even if a reduction of the number of virtual machines were possible, some kind of automatic deployment would still have been necessary to launch the virtual machines just in time and to avoid unnecessary upkeep costs.

A pre-configured S-R installation including all required software applications was deployed to the cloud in the shape of a custom Virtual Disk Image (VDI) created using the VirtualBox (http://virtualbox.org) virtualization environment. The VDI was equipped with a couple of mechanisms eventually enabling the operator to launch as many S-R instances as needed within seconds.

- VDIs contain a snapshot of the entire file system representing the content and original state of the hard disk that every new instance of a virtual machine will start with. However, some static configuration files stored on the hard disk need to be replaced or adjusted dynamically during the booting process of the operating system for the virtual machine to function properly within the environment of a specific cloud provider. A standard multi-distribution method for cross-platform cloud instance initialization has been used to cope with these issues (cloud-init. Cross-platform cloud instance initialization system. https://cloud-init.io).
- In a fully automatic deployment, all services must start automatically and be embedded in an environment, which handles irregular behavior. For this purpose, a Linux system manager (systemd. http://freedesktop.org) was configured in such a way that it started S-R as soon as the operating system was up and running. This manager further restarts the software process automatically in case of a crash.

Several modifications had to be made to S-R to accommodate cloud deployment. First and foremost, a method to provide persistent data storage and to allow file exchange between different apps had to be established. The reason is that S-R apps run in Docker containers, in which the hard disk content is volatile. If the user stops an app, the corresponding Docker container will cease to exist. Starting an app again will result in the virtual hard disk to be restored to its original state, leaving no traces of any modifications made during previous use of the app. Files created during earlier use will no longer exist and modifications to already existing files will be nullified. This mechanism ensures that applications will always start under the same well-defined conditions and thus are unlikely to fail. Furthermore, apps running in Docker containers will operate on separate file systems, each of which occupies a different location on the actual physical hard disk of the virtual machine. Different apps often depend on conflicting versions of files. Providing each app with its own individual file system solves this problem elegantly. Isolating containers from the file system on the host also improves the security of the system because malicious users may only cause damage to the isolated file system within the scope of their own container while all other users' file systems and, even more importantly, the host file system will remain out of reach. To overcome, these technical constraints, a single persistent working directory—or workspace in S-R terminology—was attached to each container/app, acting as a shared space for data exchange between software applications. Moreover, all training materials—for example, exercises and solutions—were added to this central location in a sensible structure, and participants were instructed to perform all their training activities within the parameters of their assigned workspace. A complete list of training materials is available in Section 3.2.

Interactive notebooks were created and installed in S-R to facilitate challenging processes, that most participants were expected to have difficulties with. One such notebook included a computer program for automatically compressing and downloading the entire workspace, enabling participants to save all their work and the training materials to a personal computer. Without automation or proper knowledge, backing up the workspace would have been very tedious or even impossible, given the fact that the training materials alone comprised of over 1700 files and 4.5 gigabytes of data.

Visual customizations—including the name, logo, color scheme, and font of the ICME training event—were added to the front page of S-R to provide for a familiar and professional look, which was coherent with the corporate identity and branding of the organization conducting the online training.

A data center in Frankfurt, Germany, was eventually selected to host the S-R infrastructure due to (i) its central location in continental Europe, (ii) being well-known as a major international Internet exchange point, and (iii) being one of the few locations offering virtual machines in the desired quality and quantity as required for the training.

2.4. Pre-Training Tests

It was anticipated that once the online training has started, there would be little room for corrections. Therefore, every single software component had to go through a series of thorough tests. Each S-R app and modification was tested both individually and in interplaying with each other. Functional testing was conducted solely in a virtual machine on a local computer using the VirtualBox software. Non-functional testing—especially in terms of measuring software reliability and scalability—was performed on the actual cloud infrastructure to guarantee the desired operability under the expected conditions in the training situation. System testing was mostly performed on local hardware, and—to a limited degree—on the cloud infrastructure since most conditions were known to be similar. Multiple pragmatic stress tests were performed to determine the stability of S-R on one hand, and the behavior of the cloud infrastructure on the other, for example, a series of computationally intensive long-term simulations were launched on the cloud infrastructure to estimate the impact of multiple layers of virtualization on the runtime. Latency was tested between different data centers to evaluate the effect on the responsiveness of the system, which was assumed to decrease with growing distance as a general rule. However, the impact of the connection quality was unknown and therefore was reviewed. An accessibility test was conducted by Asia based partners to validate that the training service was also accessible from their countries.

Although the handling of MS Teams for meetings is quite simple, several tests were required to get acquainted with the special class team template and the general communication coordination between us as teachers and virtual training attendees. Also, the S-R infrastructure was extensively tested by the tutors to become familiarized with the workflows.

3. Dedicated Training Materials

On-site training events typically comprise presentations by the tutors and additional hands-on exercises performed by the participants with the assistance of the tutors. In general, on-site trainings start with an introduction of the participants and the tutors highlighting their background and their interests, which—in the area of materials science and engineering—can be classified into the material types (e.g., steels, superalloys, Al- and Mg alloys, etc.) and into methods for their production (e.g., casting, welding, forming, additive manufacturing, joining). Such a short introduction facilitates the communication between the participants as it allows for the identification of common interests for bilateral discussion during the course, especially during breaks. In general, on-site training further requires some time for a basic introduction to the topic and its relevance, and some short tutorials summarizing the theoretical and phenomenological background of the trained methods trained.

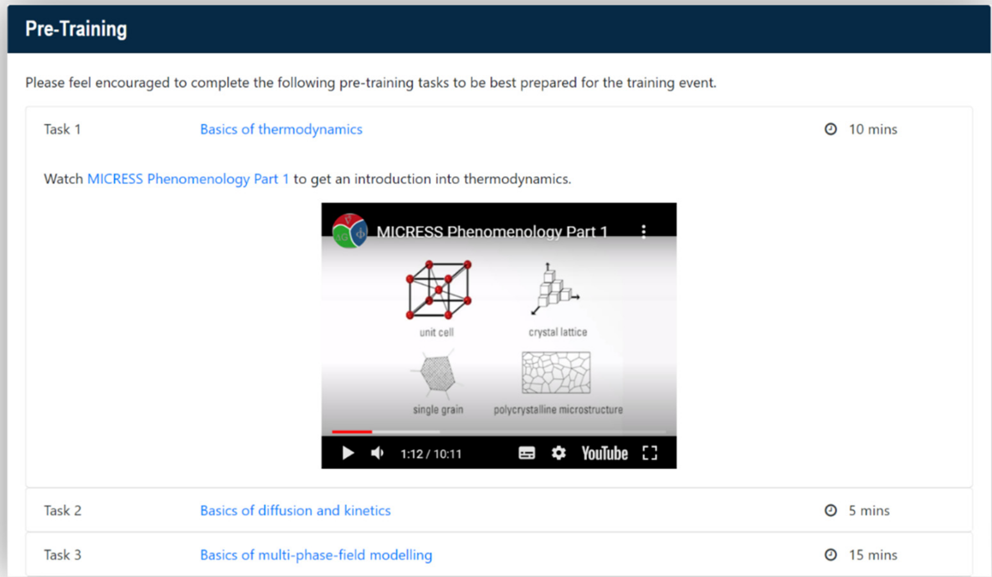
Major differences between on-site and online training are (i) the overall time being allocated to the event and (ii) the degree of distraction acting on the participants. Physical presence in an on-site training lasts, for example, three days, and breaks and social evening events provide possibilities for intense discussions and exchange. Distraction by e-mails is observed to happen to a limited extend only, while other sources of distraction (phone calls, colleagues) are negligible.

Online conferences, workshops, and also training courses—in contrast—in general, are limited in time and several sources of distraction are around. The strategy was thus to focus the online activities on anything actually requiring the computational infrastructure meaning that a clear focus was set on performing the hands-on activities. This strategy was motivated by the assumption that performing and solving hands-on challenges is less prone to distractions as compared to mere listening presentations.

To use the online training time most efficiently for such real hands-on exercises and for the solution of challenges by the participants, all participants were encouraged to make use of dedicated pre-training materials to acquaint themselves with the topic.

3.1. Pre-Training Materials

A number of existing video tutorials have been compiled and participants already during registration were encouraged to watch these presentations in preparation for the hands-on training itself, Figure 2.



Pre-Training

Please feel encouraged to complete the following pre-training tasks to be best prepared for the training event.

Task 1 Basics of thermodynamics 10 mins

Watch MICRESS Phenomenology Part 1 to get an introduction into thermodynamics.

Task 2 Basics of diffusion and kinetics 5 mins

Task 3 Basics of multi-phase-field modelling 15 mins

Figure 2. Compilation of video tutorials offered for pre-training.

The participants were further motivated to acquaint themselves with some of the freely available tools and by exploiting some of the freely available examples.

3.2. Training Materials

The training materials offered to the participants can be classified into presentations, challenges, and solution videos and were complemented by some auxiliary software tools.

3.2.1. Presentations

Short presentations of 20 to 30 min each were provided by the tutors to introduce new topics and to pose new challenges. These presentations were made available to all participants in the MS Teams workspace already prior to their presentation by the tutors.

3.2.2. Challenges

The first challenge to be met by the participants was getting access to the S-R simulation platform providing the basis for the entire hands-on training. This ramp-up of the training took some time, especially during the first of the three training sessions, where neither the tutors nor the participants had major experience with operating the setting. Once having access to the platform, further challenges to be solved by the participants were compiled in a way allowing demonstrating some strategic approaches towards own simulations in the area of ICME, for example:

- How to generate and import microstructures?
- How to modify the initial conditions for a simulation?
- How to alter the boundary conditions for a simulation?
- How to add a further chemical element into an existing simulation?
- How to add a further phase into an existing simulation?
- How to go from a 2D simulation to a 3D simulation?
- How to use different models for grain growth?
- How to set different nucleation conditions in solid-state transformations?
- How to couple to thermodynamic datasets?

Each of these challenges was shortly presented by the tutors and was made available to the participants as a file in the MS Teams workspace.

3.2.3. Solution Videos

During on-site training, hands-on challenges are typically tackled with the personal assistance of the tutors. During past on-site training in the area of ICME, a ratio of one tutor per four participants turned out to be adequate to offer individual support to the participants.

During the early stages of registration, an unexpectedly high interest in the online training became obvious. A large number of participants had to be supported by only a few tutors. Additional "solution videos" were thus generated for each of the challenges showing in detail all the individual solution steps and allowing the participants to look up the solutions without the need of individual assistance by a tutor. The duration of these videos typically was between 5 to 10 min, which corresponds to the time a skilled user needs to perform the challenge. The time allocated to the participants for the solution of each challenge was longer by a factor of 5 to 6.

3.2.4. Auxiliary Software

To facilitate the solution of the challenges, a number of pre-calculated files were provided which could be used as an intermediate result to build on in case of getting stuck. Further, a number of pre-configured Jupyter notebooks were provided to facilitate post-processing of the simulated data, for example, to generate plots or to generate regression curves to the data. Eventually, a specific notebook was provided allowing the participants to compress the entire user workspace into a zip archive and to download it at the end of the training course.

4. The Hands-On Training Courses

4.1. Registration and Attendance

Computational scientists from both academia and industry were encouraged to register for the training. The participation in the event was free of charge and the number of participants a priori was limited to 25 per group, but later was raised to a maximum of 35 to account for the high interest. The training slots were allocated on a first-come-first-serve basis, and a maximum of two participants per institution were allowed to register. Different time slots were provided to allow the accommodation of participants from different time zones. A total number of 97 participants from 21 countries in four continents eventually had formally registered for the three training periods, with almost 50% coming from India. Approximately 50% of the registered participants held a Ph.D., 30% a Master's degree, and 20% a Bachelor's or other degree.

During online registration, the participants were requested to fill a questionnaire indicating (i) their prior knowledge in different ICME related simulation tools, Figure 3, (ii) their interest in terms of materials, (iii) their interest in terms of manufacturing processes and, (iv) their expectations for the training.

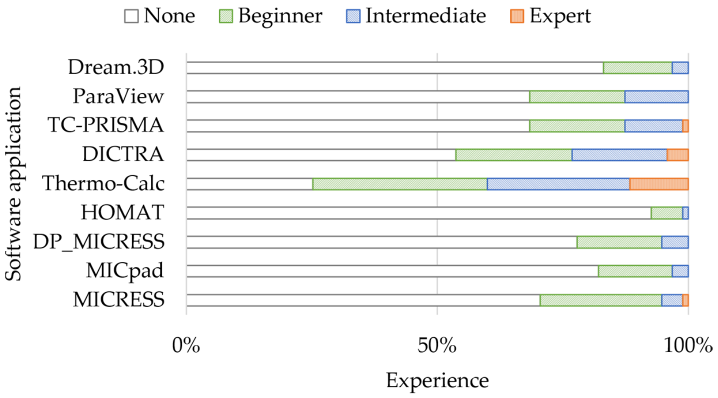


Figure 3. Summary of the self-assessment of the course participants regarding their experience with the software applications used in the training course.

The dominant materials classes of interest to the participants were Al-based alloys, steels, superalloys, Ti-alloys, and high entropy alloys (HEA), followed by Mg-alloys and Li-alloys. Only a few participants were interested in ceramics or semiconductors like CdZnTe.

The processes and phenomena of interest to the participants covered the entire ICME process chain starting from casting/solidification via diffusion/heat-treatment, forming/recrystallization/grain growth, to brazing/welding/joining, and coating/metal-deposition. Eventually, quite a number of participants were interested in additive manufacturing.

A number of 54 persons out of 97 formally being registered eventually really actively joined the three training courses, which thus had a number of 15–20 participants each, Figure 4. The agenda for the training courses is detailed in Appendix A.

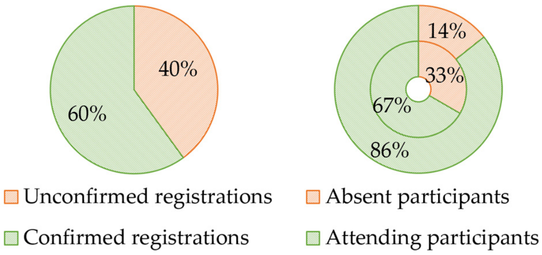


Figure 4. Exemplary statistics for Group A: Percentage of confirmed registrations (**left**) and attendance in percent of the confirmed registrations (**right**) on Day 1 (outer ring) and Day 2 (inner ring).

4.2. Communication Channels

Video and chat functionality as a communication channel was provided using MS Teams. Each of the three training groups—including the tutors—was organized as a "team" in MS Teams leading to training groups A, B, and C, respectively. Presentations by the tutors to all members of a team were given in a team meeting covering the entire training day. Short sections subsequent to the presentations provided the opportunity for the attendees to raise their questions either in the chat or by video communication being moderated by one of the tutors. The MS Teams chat functionality turned out to be the favorite communication channel for questions. The general meeting chat was also used to get feedback during the hands-on session.

One to one chats were used to give advice mainly during the breaks and during the hands-on sessions. However, these chats were also possible any time with all tutors (except the actual presenter). Especially posting of screenshots in the chats—by both the participants and by the tutors—turned out to be very effective in terms of both the communication of troubles/questions and also of successful solutions.

5. Evaluation

5.1. Technical Issues

MS Teams requires a Microsoft account to join a team. Our strategy was to register the participants for a team in MS Teams a week before sending the first meeting invitation. Nevertheless, it obviously was confusing when mixing up meeting invitations and team invitations. A participant who only enters the meeting cannot experience the full MS Teams functionality, esp. one-to-one video calls and chats, and also cannot access the common file system of the team. Furthermore, some organizations seem not to allow the use of MS Teams. In such cases, the participants were encouraged to use private accounts.

The virtual machines started on demand were assigned dynamic Internet Protocol (IP) addresses by the cloud provider. Around 15% of these IP addresses could not be reached because web browsers mistakenly perceived them as a potential source of danger. Presumably, these IP addresses had been used in the past for dubious purposes and consequently have been placed on a corresponding blacklist. Unfortunately, as a customer, you do not have an influence on the process of assigning IP addresses. An additional validation step was introduced as a workaround. Virtual machines that received an unreachable IP address were restarted until the system assigned them a valid IP address. A single restart was sufficient on average.

During the hands-on exercises, a bug occasionally occurred in S-R that caused the connection to the software application to be lost. The error could be traced back to certain triggers, while the actual cause remains unknown. Fortunately, these connection errors only occurred relatively rarely. In such cases, the connection could be re-established by refreshing the web browser without any loss of data or progress. The bug has been reported to the developers.

In the anonymous survey, some participants reported having experienced performance problems using S-R, the cause of which can unfortunately only be speculated due to the imprecise descriptions.

5.2. Tutors Perception

The perception of the actual training by the tutors is summarized as follows:

- Video function should be switched on at least once during a "getting acquainted session" to make the training less anonymous.
- Shared channels like chat are useful but often get dominated by a few very active participants.
- During hands-on sessions, direct bilateral interaction was possible by one-to-one video calls and screen sharing. However, a direct interaction, for example, via screen sharing also on the hands-on computers combined with an audio call or chat would be even more efficient.

- Presentation sessions were sometimes hard because of missing direct feedback. Video contact could perhaps have been helpful.
- Muting participants is required mainly because of suboptimal technical equipment of participants. Unfortunately, this increases response times and widely prevents spontaneous questions or discussions.
- Timeshift in an international context did not create a major problem, although some of the participants had to work during night time.
- In order to improve the efficiency of this digital format, personal commitment should be increased. This can be achieved by the personal introduction of each participant at the beginning, intermediate presentation of results, or other requirements to get a certificate of attendance.

5.3. Feedback from Participants

(i) Immediate feedback from participants during and shortly after the course

It was mandatory to have feedback on the usability of the S-R platform already during the initial period of the training as all subsequent activities required the participants to be fully acquainted with the hitherto unknown platform.

(ii) Feedback form

A feedback form was sent to all 97 registered participants and not only to those 54 who actually participated in one of the three courses. The scope was not only to collect feedback from those who participated but also to detect reasons for non-participation. In total, 27 out of 97 originally registered participants returned the filled form. Four of them could not join the training at all because of other obligations (1), difficulties in login in to the platform (1), or other reasons (2). Of 23 people, two attended the presentations but did not perform the hands-on exercises. They reported troubles in using the platform (1) and unclear instructions for its use (1). The remaining 21 actively participated in the hands-on challenges and judged the complexity of the tasks to be solved as adequate (14), too difficult (4), or too easy (3).

They further provided detailed and interesting feedback about the Simphony-Remote platform used for the first time for such a training course, Figure 5. The average SUS score was 66 points, which corresponds to an adjective rating between "ok" and "good", with a tendency towards the latter [8]. It can be assumed that under better conditions an even higher score could have been achieved. The following factors may have had a negative impact on the score: (i) insufficient introduction, (ii) participants being unfamiliar with the hosted software applications (supported by registration form data; Figure 3), (iii) inadequate connection bandwidth and/or high latency (supported by feedback form comments; Appendix B), (iv) software bugs (supported by own experience; Section 5.1).

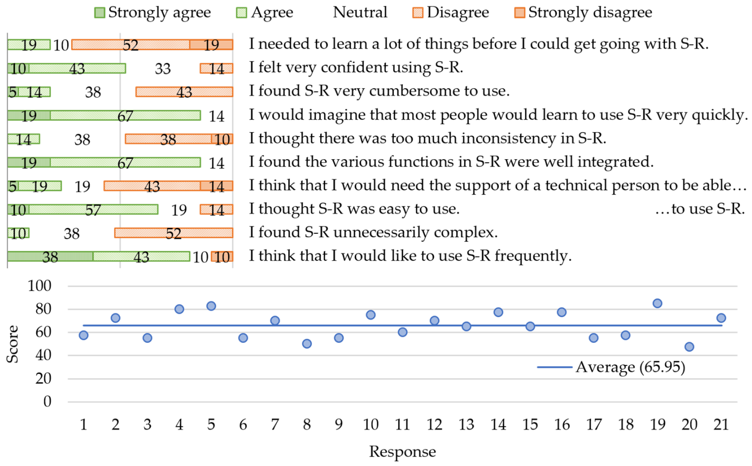


Figure 5. Results of a System Usability Scale study on Simphony-Remote showing the distribution of the responses (**top**), and the total scores per response/participant (**bottom**).

The participants further communicated their views on different aspects of the training course. The overall average rating of 23 participants for the training course was 4.3 "stars" (out of 5), Figure 6.

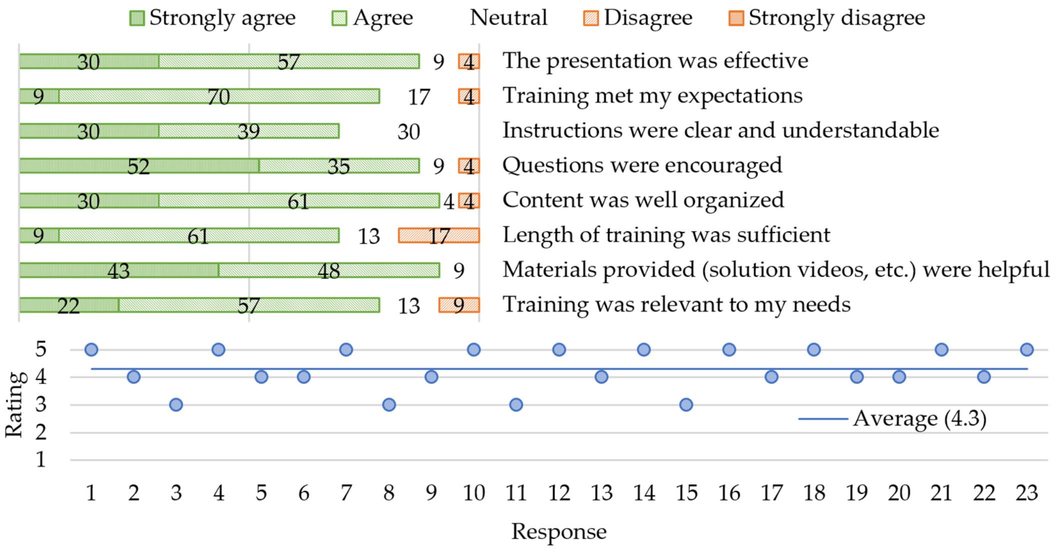


Figure 6. Participants' feedback on different aspects of the training (top), and the overall training experience rated on a scale between 1 (worst) and 5 (best).

Some participants eventually added further encouraging comments (for all comments see Appendix B) like "It was a great experience, tailored for people of different backgrounds and levels of expertise. That's difficult enough for an online training session".

6. Conclusions and Lessons Learned

The lessons learned during this first cloud-based ICME training event can be categorized into organisatorial, communicational, educational, and operational aspects.

From an organisatorial point of view, registration of the participants via the internet worked very smoothly and the number of about 35 registered people per team was at the upper limit of what could be handled. In fact, however, only approximately two-thirds of the formally registered people took part in the training, while we took some effort in setting up the infrastructure for 35 people. For future training sessions, we would consider a small participation fee and only about 20 participants, which was found to allow quite effective work, as meaningful. This participation fee would generate a stronger commitment and cover some of the cost of renting the hardware. A free tutorial aiming at introducing the background of the models used in the different training challenges in the future should be provided in addition and prior to the hands-on sessions. A quite strong demand arose to continue work on the simulation platform after the end of the official training period. This demand could not be satisfied for Group A, because Group B needed the infrastructure the next day. It could however be realized for Group B, because of a weekend between B and C and also for the final Group C. Future courses should be organized as events being separated by at least one week to allow for exploiting the possibilities of the platform and using the tools by the participants on their own.

From a communicational point of view, a number of participants were not yet used to MS Teams. Moreover, the use of MS Teams seemed to be blocked by some institutions. The situation would probably have been similar for other web conferencing software like Zoom, Webex, GoToMeeting, FreeConferenceCall to name just a few. In spite of the fact that people get increasingly used to such tools, there still remains the difficulty of switching between the conferencing tool and the computer, where the hands-on simulations are run. In the wake of this first cloud-based ICME training, we started to exploit other options and plan to use Webex including its additional training module (Webex Training. Online training solution. http://webex.com) for the next training in view of its subjective ease of use and of the option to integrate the training computers in hands-on sessions.

From an educational perspective, the training has been very successful according to the feedback from the participants. It was, however, quite an effort creating all the tutorial videos—especially those detailing the solutions. Now, this material is available for re-use and also to the public (http://micress.rwth-aachen.de/online-training-2020.html). These videos now allow for self-learning by people interested in learning about the different tools, which might be at their disposal as local installations.

From an operational perspective, the deployment of the software infrastructure as a cloud service with one virtual machine being allocated to each of the participants and also the tutors was very effective. In future training sessions, care will be taken in checking and using the allocated IP names of these virtual machines as some of them were blacklisted. The major positive operational experience relates to the S-R platform being used for educational purposes the first time according to our knowledge. S-R was originally developed to eliminate the need of deploying sophisticated scientific software and large-scale research datasets to all workstations and people involved in collaborative work [9]. The practical experiences made during the cloud-based ICME training and being reported by the participants demonstrate quick familiarization and ease of use of S-R. The application of S-R is not at all limited to such a specific type of educational use. In view of the open nature of the training course not comprising any exchange of confidential information, security issues were not analyzed in detail. Its flexibility, including options to customize its open source code, makes S-R applicable in a number of other scenarios, in which a dedicated set of software applications has to be made hands-on accessible via web browsers to an audience being spread over the world. A thorough study of non-functional user requirements [10] to the simulation platform and its usability would have been beneficial. Such a detailed study could however not be performed in the time available for the preparation of the training. Further, there were no options to modify the software applications installed on the platform and used in the training, as most of these applications are proprietary codes and/or developed by third-party domain experts. Focus thus was set on meeting the functional requirements first, while considering non-functional requirements on the basis of own user experience only. Optimization of the platform infrastructure towards non-functional requirements is subject to future work.

Author Contributions: Conceptualization, G.J.S. and L.K.; methodology, L.K., R.A., J.E.; software, L.K.; validation, J.E., G.J.S., B.B.; formal analysis, L.K.; resources, L.K.; writing—original draft preparation, G.J.S., L.K.; writing—review and editing, J.E., B.B., R.A.; visualization, L.K.; supervision, G.J.S.; project administration, G.J.S.; funding acquisition, G.J.S. All authors have read and agreed to the published version of the manuscript.

Funding: Parts of the depicted work—especially the work on the simulation platform—compiling this publication and the APC for this article were funded by the European Commission in the frame of the MarketPlace project, grant agreement number 760173.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A Training Schedule

Table A1. Training Schedule Day 1.

Group A/C	Group B	Schedule
from 08:45	from 15:45	Web conference sign in
09:00–09:30	16:00–16:30	Introduction *Welcome* *Training agenda* *Introduction of participants and tutors*
09:30–10:00	16:30–17:00	Introduction to the training platform *Overview* *Getting started*
10:00–10:30	17:00–17:30	Introduction to MICRESS Introduction to MICpad *How to rename an existing input file* *How to modify the output file path* Your First MICRESS simulation *How to run a simple grain growth simulation* *How to inspect the results of a simulation*
10:30–11:00	17:30–18:00	Coffee break/Q&A
11:00–12:00	18:00–19:00	Introduction to the DRI file format *Step-by-step examination of a pre-configured DRI file* Customizing your DRI file *Recommended workflows and best practices* *Troubleshooting strategies* *How to modify the initial microstructure*
12:00–12:30	19:00–19:30	Introduction to pre-processing *How to create a custom GES5 thermodynamic database with Thermo-Calc*
12:30–13:30	19:30–20:30	Lunch break/Q&A
13:30–14:00	20:30–21:00	Introduction to simulation conditions *How to define the initial and boundary conditions*
14:00–14:30	21:00–21:30	Moving from 2-D to 3-D simulation *How to define the dimensions of a RVE* Introduction to post-processing *How to examine simulation data with DP_MICRESS* *How to export simulation data to VTK* *How to visualize simulation data with ParaView*
14:30–15:00	21:30–22:00	Adding a further phase
optional	optional	Adding a further alloy element *How to replace an existing GES5 file*
12:30–15:00	19:30–22:00	Summary Q&A

Table A2. Training Schedule Day 2.

Groups A/C	Group B	Schedule
from 08:45	from 15:45	Web conference sign in
09:00–09:30	16:00–16:30	High-end applications *Continuous casting HSLA steels* *ICME of superalloy components*
09:30–10:00	16:30–17:00	Initial microstructures I: Creation *How to create a microstructure from experimental data* *How to create a synthetic microstructure* *How to create a microstructure with Dream.3D*
10:00–11:00	17:00–18:00	Initial microstructures II: Importing *How to import a microstructure from Dream.3D to MICRESS*
11:00–12:30	18:00–19:30	Isothermal grain growth Grain growth models *How to inspect average grain radius evolving over time* *How to use different grain growth models*
12:30–13:30	19:30–20:30	Lunch break/Q&A
13:30–14:30	20:30–21:30	Phase transformations Nucleation models *How to inspect phase fraction evolving over time* *How to modify seed types, seed undercooling, and seed locations*
14:30–15:00	21:30–22:00	Summary Farewell
after 15:00	after 22:00	Q&A (optional)

Appendix B

Table A3. Feedback form: Comments/Ideas for Improvement.

People that use Micress for the first time don't know the program language and don't find it easy to find the targeted change in the domain to change parameters of simulations. There needs to be more explaining of the domain and parameters and how to change them properly before we jump from challenge 1 to challenge 6.
I like to download the presentations/instructions so I can have them in front of me, but I found downloading them from SimphonyRemote was a bit cumbersome. Maybe providing one single zip with all the documents included would have been useful. (or maybe you did have this and I just missed it . . .). GREAT EFFORT BY EVERYONE INVOLVED, THANKS VERY MUCH!!
Simphony-Remote was slow and there were issues with Jupyter
The bugs faced during working over S-R beta version can be sorted out and S-R version can be better in terms of connectivity.
SimphonyRemote system was good, but it was just unstable.
It is a bit difficult to follow without knowing the basics of the MICRESS program and how to enter the commands correctly.
It was a tremendous effort by Team Access to put together such a wonderful training. Though there were a few potholes, it wasn't an entirely bumpy ride. The team was well prepared and the "Solutions Videos" was an excellent thought, given the limited amount of time. The online interaction through the Chat Box was quite educative. I don't think even in a physical meeting we would have had the opportunity to interact to that extent.
Having said that, there are a few areas/ideas for improvement.
Though Simphony-Remote is a beta version, it kept crashing and it took some time to figure out how to avoid that. After opening any application such as Micress or Jupyter or Dream3D, if I immediately click 'File' on the top tab, the application was crashing. This was quite annoying. Hope this will be fixed in the deployment version.
It was taking quite a bit of time when one switches between applications (for example Micress and Jupyter). This not only happens during the initialization alone but subsequently during entire sessions.
Dream3D session could have been effectively communicated with example problems or sub-problems in an application area rather than explaining every part of the code.
The training was limited to steels while I was expecting at least one example in non-ferrous systems. Or at least, some training example could have been provided for trying out on our own. One cannot take an easy decision to buy the software unless he/she tries out some problem in their own technical area. On a similar note, some of the thermodynamic software also provide training on steels, and the actual experience with non-ferrous systems springs some unwanted surprises.
Further, I feel that access to SimphonyRemote could have been kept open for a few more days (or at least a trial version limited to ternary systems). Though effectively we had about 36 h of access, it was extremely taxing to try out something new (after a long day of training and timezone differences).
The Simphony-Remote access may be extended for one or two weeks for practice purposes.
Would enjoy a training that builds on what we have learned so far.
The training was helpful in learning new things.
It could have been better if you can allow us for original database so that we can use for our projects
The time to solve the exercises was short. Perhaps in the next training sessions, you can give the participants more time to practice the exercises.
It was a great experience, tailored for people of different backgrounds and levels of expertise. That´s difficult enough for an online training session. Maybe to divide the day into shorter modules with better designed hands-on activities, perhaps making presentations a bit more simple (not so biased to showing off the software, we know already it is great).

References

1. Schmitz, G.J.; Prahl, U. (Eds.) *Handbook of Software Solutions for ICME*; Wiley VCH: Weinheim, Germany, 2016; ISBN 978-3-527-33902-0.
2. Faltens, T.; Strachan, A.; Klimeck, G. nanoHUB as a Platform for Implementing ICME Simulations in Research and Education. In Proceedings of the 3rd World Congress on Integrated Computational Materials Engineering, Colorado Springs, CO, USA, 31 May–4 June 2015. [CrossRef]

3. Talirz, L.; Kumbhar, S.; Passaro, E.; Yakutovich, A.V.; Granata, V.; Gargiulo, F.; Borelli, M.; Uhrin, M.; Huber, S.P.; Zoupanos, S.; et al. Materials Cloud, a platform for open computational science. *Sci. Data* **2020**, *7*, 1–12. [CrossRef] [PubMed]
4. Yakutovich, A.V.; Eimre, K.; Schütt, O.; Talirz, L.; Adorf, C.S.; Andersen, C.W.; Ditler, E.; Du, D.; Passerone, D.; Smit, B.; et al. AiiDAlab—An Ecosystem for Developing, Executing, and Sharing Scientific Workflows. *Comput. Mater. Sci.* **2021**, *188*, 110165. [CrossRef]
5. Janssen, J.; Surendralal, S.; Lysogorskiy, Y.; Todorova, M.; Hickel, T.; Drautz, R.; Neugebauer, J. Pyiron: An integrated development environment for computational materials science. *Comput. Mater. Sci.* **2019**, *163*, 24–36. [CrossRef]
6. Koschmieder, L.; Hojda, S.; Apel, M.; Altenfeld, R.; Bami, Y.; Haase, C.; Lin, M.; Vuppala, A.; Hirt, G.; Schmitz, G.J. AixViPMaP®—An Operational Platform for Microstructure Modeling Workflows. *Integr. Mater. Manuf. Innov.* **2019**, *8*, 122–143. [CrossRef]
7. Brooke, J. *SUS—A 'Quick and Dirty' Usability Scale. Usability Evaluation in Industry*; Taylor and Francis: London, UK, 1986.
8. Brooke, J. SUS—A Retrospective. *J. Usability Stud.* **2013**, *8*, 29–40.
9. Borini, S. Simphony-Remote—Accessing Containerized Desktop and Web Applications with a Web Browser. EuroSciPy 10 2017. Available online: https://av.tib.eu/media/38209 (accessed on 20 December 2020).
10. Sun, P.; Yang, J.; Ming, H.; Chang, C.K. A Multi-layered Desires Based Framework to Detect Users' Evolving Non-functional Requirements. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018.