D2.4

Materials-to-device and device-to-materials syntactic interconnections

Claudio Rosati, Matteo Bertocchi, Francesco Benedetto, Andrea Padovani, Valerio Lunardelli, Flaviano José Dos Santos, Nicola Marzari, and Arrigo Calzolari

Document information

Project acronym:	INTERSECT
Project full title:	Interoperable Material-to-Device simulation box for disruptive electronics
Research Action Project type:	Accelerating the uptake of materials modelling software (IA)
EC Grant agreement no.:	814487
Project starting / end date:	1st January 2019 (M1) / 31st January 2022 (M37)
Website:	www.intersect-project.eu
Final version:	29/01/2021
Deliverable No.:	D2.4
Responsible participant:	EPFL (participant number 2)
Contributing Consortium members:	AMAT, CNR
Due date of deliverable:	31/01/2021
Actual submission date:	29/01/2021
Dissemination level:	PU - Public

Authors: Claudio Rosati, Matteo Bertocchi, Francesco Benedetto, Andrea Padovani, Valerio Lunardelli, Flaviano José dos Santos, Nicola Marzari, and Arrigo Calzolari

To be cited as: Rosati C. et al., (2021): Interoperable Material-to-Device simulation box for disruptive electronics. Deliverable D2.4 of the H2020 project INTERSECT (final version as of 29/01/2021). EC grant agreement no: 814487, EPFL, Lausanne, Switzerland

Disclaimer:

Versioning and Contribution History

Version	Date	Modified by	Modification reason
v.01	20/01/2021	Matteo Bertocchi	First Version
v.02	27/01/2021	Andrea Padovani	Revision
v.03	29/01/2021	Arrigo Calzolari	Final Version

D2.4 Materials-to-device and device-to-materials syntactic interconnections

Table of contents

1 Executive Summary

The present document is the deliverable **D2.4 - Materials-to-device and device-to-materials syntactic interconnections** prepared under *Task 2.3. Interoperability hub: Materials-to-device and device-to-materials syntactic interconnections and requirements*.

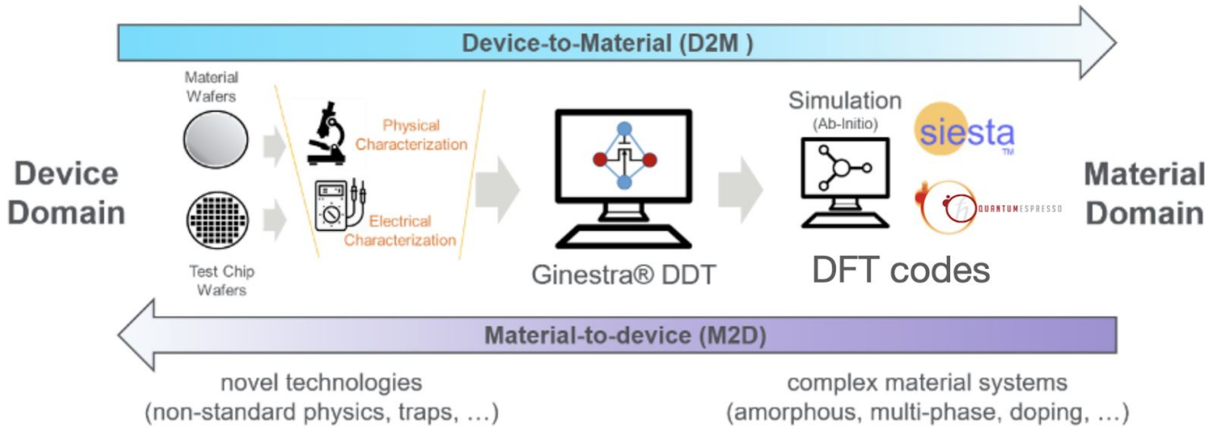

Figure 1 *M2D and D2M workflows.*

INTERSECT project [1] is developing an industry-ready integrated, standardized, interoperable software platform called Interoperable Materials to Device (IM2D). IM2D aims at integrating some of the most used open-source materials modelling codes, Quantum ESPRESSO (QE) [2] and SIESTA [3], with models and modelling software for emerging devices (Ginestra™) [4] via the SimPhony infrastructure [5] for semantic interoperability and ontologies, powered by the AiiDA [6] workflow engine, and its data-on-demand capabilities and apps interface. The scope of this Deliverable is to show the progress on the *interoperability Hub* (iHub) of IM2D platform. The iHub provides European Materials Modelling Ontology (EMMO) compliant standardized rich metadata that enable a seamless communication and integration between the different components of the IM2D platform and other external repositories and marketplaces. Through the iHub, IM2D platform is able to connect the device level simulation to the material modelling supporting both the material-to-device and the device-to-material workflows (Fig. 1). In the case of the material-to-device mode, iHub converts the requests for material/defect data needed for device simulations into requirements for materials' properties that are either stored or needed to be calculated by the material modeling codes (QE, SIESTA). In the case of the device-to-material mode, the defect/material data (e.g. band-gap, defect thermal ionization energy) extracted from the simulation of electrical device characteristics will be fed into the material modelling (both codes and database) in order to identify the material atomic structure (e.g. phase) and defect characteristics. For each workflow in D1.1 we hereby identify specific classes of use stories and application cases. Through the iHub data exchange to/from databases and processes, workflows (explained in D2.3) will be actuated, depending on the specific Ginestra™ requests and through the syntactic interconnection to AiiDA.

1.1 About this document

This document represents the status of deliverable D2.4 of the INTERSECT project concerning the material-to-device and device-to-material implementation in the interoperability hub of IM2D platform. There are no relevant Task deviations from the original DoA plan to be mentioned.

2 Material-to-device (M2D) and Device-to-materials (D2M) workflows

In developing the IM2D Interoperability infrastructure, we consider two main pathways: a top-down material-to-device (M2D) approach and a bottom-up device-to-material (D2M) approach. The M2D workflow improves the IM2D device-design capabilities by understanding and predicting the device performance starting from the material properties and fabrication process effects. On the other hand, D2M workflow improves the IM2D material-design capabilities by exploring new materials and new compounds, starting from the desired electrical performances of the device.

2.1 Material-to-device (M2D)

General concept: Material properties, including electrically active defects initially calculated by using electronic and atomistic models, are further exploited by Ginestra™ to simulate the electrical response of a selected device.

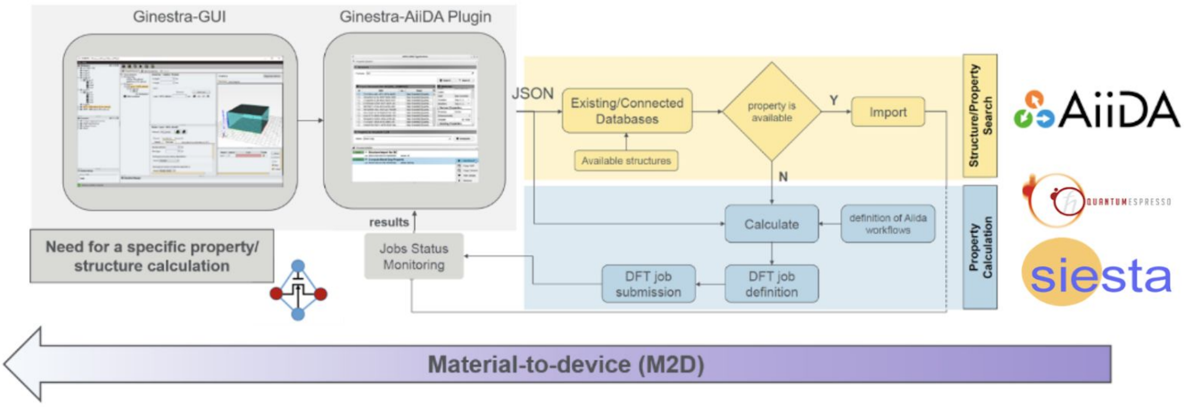

Figure 2 *Material-to-device (M2D) workflow.*

2.2 Device-to-material (D2M)

General concept: Specific electrical device characteristics (e.g. I-V curve) are interpreted by using the Ginestra™ code in the IM2D box to identify/determine specific material properties (e.g. spatial and energetic distribution of defects) that will be investigated at the Density Functional Theory (DFT) level to define the most promising material for the specific device application.

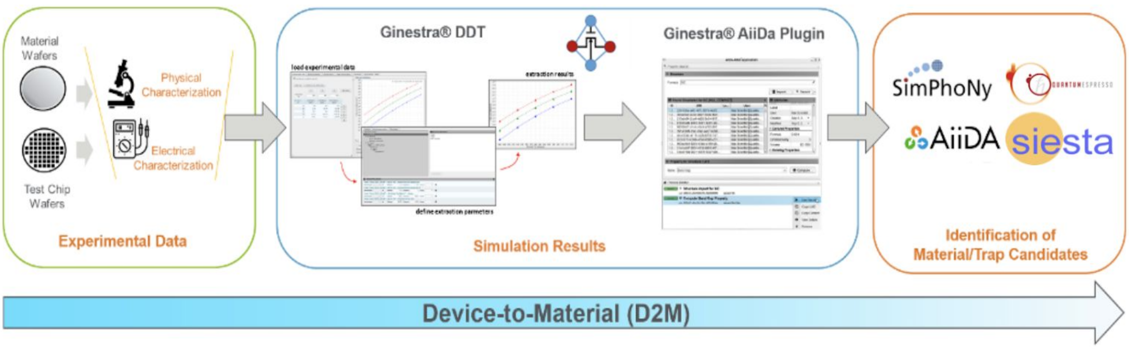

Figure 3 *Device-to Material (D2M) workflow.*

3 Syntactic Interoperability

3.1 Framework description

The Ginestra™ suite can interact with AiiDA through the AiiDA-POST, a plugin that extends AiiDA's Representational State Transfer (REST)-Application Programming Interface (API). This API allows Ginestra™ to remotely connect to AiiDA and submit workflows, retrieve data and query information from its database. On the Ginestra™ side, a plugin can handle the queries, the submission and monitoring of workflows, as well as the retrieval of the results and other general interactions with the AiiDA REST-API. Data type and formats mentioned below are extensively reported in the data management plans (D4.2, D4.6).

Upon HTTP queries, the plugin always returns a JavaScript Object Notation (JSON) file, a simple and well-supported interchange file type format that can be easily parsed and used. The JSON file provides information about the request and, additionally, error messages in the case of failures during the execution. In order to simplify the user experience, we have implemented a mapping that attributes keywords to some properties and directly relates them to the workflows. This enables us to submit the corresponding workflow, and to tie to the outputs for the retrieval of a specific data. The available properties list and description will be discussed later on in this document. The submission of a workflow can be easily performed with a "POST" HTTP request, by providing a JSON dictionary of the required and optional inputs of a workflow. On successful submission, the request JSON is stored in the database, and linked to the inputs of the workflow; the unique UUID identifier of the workflow is returned to the user, and can be used to query the AiiDA database for the status of the calculation, the output nodes, and the required property. Therefore, the provenance of the calculation directly ties the JSON request to the final results, connecting all the in-between steps of the workflow.

3.2 Status of the AiiDA-POST REST-API

The AiiDA-POST REST-API is an extension of the AiiDA REST-API, and is meant to satisfy the specific needs of the INTERSECT project. It creates a web interface that allows users to explore, query and control the AiiDA engine and databases, either locally or remotely. It is the main infrastructure that mediates the communication between the Ginestra™ plugin and the AiiDA infrastructure, and thus the connection with the DFT engine. *This constitutes the ground foundation of the syntactic code interoperability at the basis of the IM2D platform*. Since deliverable D2.1 ("Plugins for code interoperability"), we have maintained, improved and expanded the AiiDA-POST REST-API. In particular, we have updated the package making it compatible with the newest AiiDA version <1.5. We have also facilitated the query for available DFT codes and integrated it with the AiiDA-Defect package. The current available endpoints are (besides the standard AiiDA REST-API endpoints) the following:

Resource	**Method**	**Endpoint**
gdata	GET,	/api/v4/intersect/derived_data/structuredata
gexisting	GET,	/api/v4/intersect/existing/properties/<string:prop>/
gexisting	GET,	/api/v4/intersect/existing/properties/<string:prop>/<string:node_id>
gproperties	GET,	/api/v4/intersect/properties/
gproperties	GET,	/api/v4/intersect/properties/<string:entrypoint>/inputs/
gproperties	GET,	/api/v4/intersect/properties/<string:entrypoint>/list/
gproperties	GET,	/api/v4/intersect/properties/<string:entrypoint>/outline/
gproperties	GET,	/api/v4/intersect/properties/<string:entrypoint>/outputs/
gstatus	GET,	/api/v4/intersect/status/<string:node_id>
gsubmit	POST	/api/v4/intersect/submit/
gworkflows	GET,	/api/v4/intersect/workflows/
gworkflows	GET,	/api/v4/intersect/workflows/<string:entrypoint>/inputs/
gworkflows	GET,	/api/v4/intersect/workflows/<string:entrypoint>/outline/
gworkflows	GET,	/api/v4/intersect/workflows/<string:entrypoint>/outputs/

The resource ***gdata*** allows querying in the database by searching and filtering on derived properties, i.e., properties that can be trivially calculated from the material structure stored in the database, such as the chemical formula and the unit volume. The ***gproperties*** resource responds with information about the properties that have been mapped. It can respond, for example, with a list of required inputs for a given calculation. The currently mapped properties are: ground state (relaxed) crystal structure, total energy, band structure, band gap, effective mass, defect formation energy, and dielectric constant. These are the "*material's on demand*" properties identified in D1.1 "Report on use cases and system requirements" (WP1) and whose workflows have been implemented in Task 2.2 (WP2) and described in deliverables D2.2 and D2.3. As soon as other workflows for advanced materials parameters are implemented, we will include them in the *gproperties* map list. By defining and implementing calculations protocols that automatically determine many of the DFT codes inputs (see D2.3 "QE and SIESTA workflows for advanced materials parameters"), a user is able to launch the calculation of many of these properties with minimal input, whenever needed, and not previously calculated. However, few advanced workflows, such as those in the AiiDA-defects package, do require many inputs from the user as they are intrinsically meant for expert users. The ***qworkflows*** resource allows to expose all the available workflows, their input and output, even for properties that have not been mapped yet. The ***gsubmit*** is a POST resource and allows the user to submit calculations for any of the mapped properties or available workflows. Furthermore, it allows special processes for importing structures from the Crystallography Open Database (COD) or to import structures by uploading individual Crystallographic Information Files (CIF). The ***gstatus*** resource enables the Ginestra™ application to monitor the status of running and completed workflows. And, finally, the ***gexisting*** resource is meant to query the AiiDA database for available (previously calculated) mapped properties. Further information can be found in the AiiDA-POST repository documentation [https://gitlab.cc-asp.fraunhofer.de/intersect/ext_to_aiida].

As an example, the AiiDA-POST server can respond to HTTP query of the type:

HTTP 127.0.0.1:5000/api/v4/intersect/existing/properties/band_gap.pw

where a response similar to the following is expected.


```
▼ data:
  ▼ properties:
    ▶ 0:                 {…}
    ▼ 1:
        is_node:         false
        property_name:   "band_gap"
        property_value:  null
        structure:       "af79d8c2-f48a-42ad-a184-9235cd83b537"
        workflow:        "a3c7728d-fdaf-4e21-86ad-f42285a8872f"
    ▼ 2:
        is_node:         false
        property_name:   "band_gap"
        property_value:  3.9578492070523
        structure:       "c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e"
        workflow:        "e7711106-ce83-4400-a792-67f2ef4ab6c4"
```


Figure 4 *Response of the AiiDA-POST REST API for the resource properties when a particular property is specified, e.g., band_gap.pw. A null value indicates that the structure is metallic, therefore, not featuring a band gap.*

A dedicated cloud server for testing and development of the iHub is up and running. It was set to improve the collaborative effort among the different partners. In this server, we have installed the AiiDA core engine, the AiiDA-POST package, the AiiDA-defect package, the AiiDA-QuantumESPRESSO plugin, as well as the atomist quantum simulation codes (Quantum ESPRESSO and SIESTA). Thus, the Ginestra™ plugin can work from any external computer where it is installed connectable to the server. This complements the activity of Task 1.5 about the creation of a central code repository and versioning system.

3.3 Status of the Intersect Plugin for the Ginestra™ Application

As shown in Figs. 2 and 3, INTERSECT Ginestra™-AiiDA plugin allows to exchange EMMO compliant data from/to AiiDA database in both the M2D and D2M workflows. It is possible to retrieve/import a crystalline structure, retrieve a physical property, or submit/monitor the property computation.

A typical usage scenario in the M2D workflow is shown in the following picture (Fig. 5), where:

- A device property (Band Gap in the example) needs a value;
- The plugin button is pressed and its Graphical User Interface (GUI) is displayed;
- The material chemical formula is typed in the ***Formula*** field, and the ***Search*** is performed;
- The found structures are displayed;
- A structure is selected showing that the associated Band Gap property was not previously computed;
- The computation of the Band Gap property (together with some correlated ones) is started;
- The corresponding AiiDA process is monitored.

When the computation successfully ends, the structure can be inspected again, and/or the computed value can be copied into the device **Band Gap** property editor.

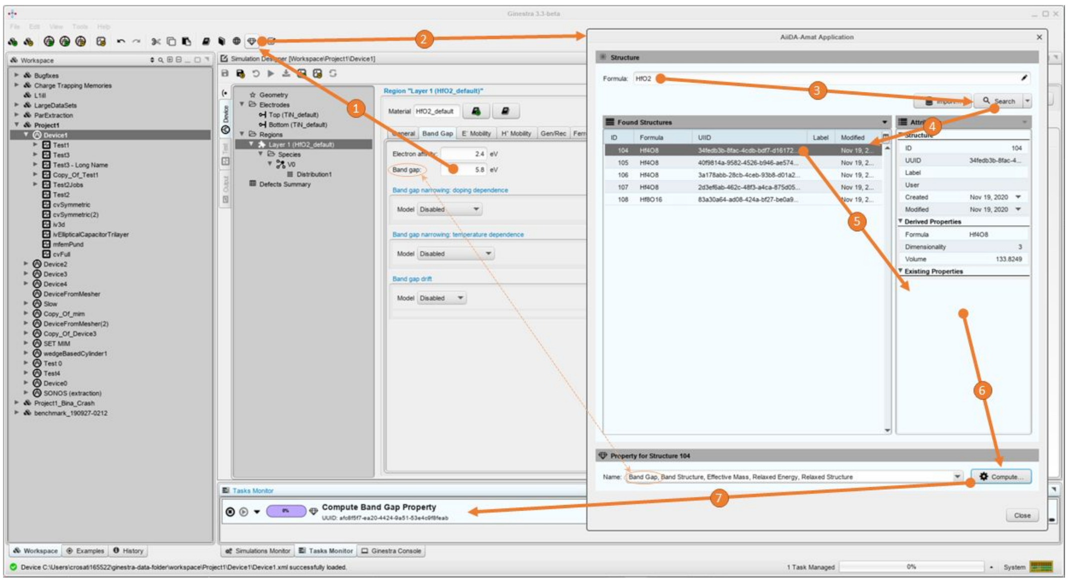

Figure 5 *Band Gap computation in Ginestra™.*

3.4 Ginestra™-AiiDA interconnection workflow

The workflow describing access to the AiiDA service in theM2D case is depicted in the following picture (Fig. 6).

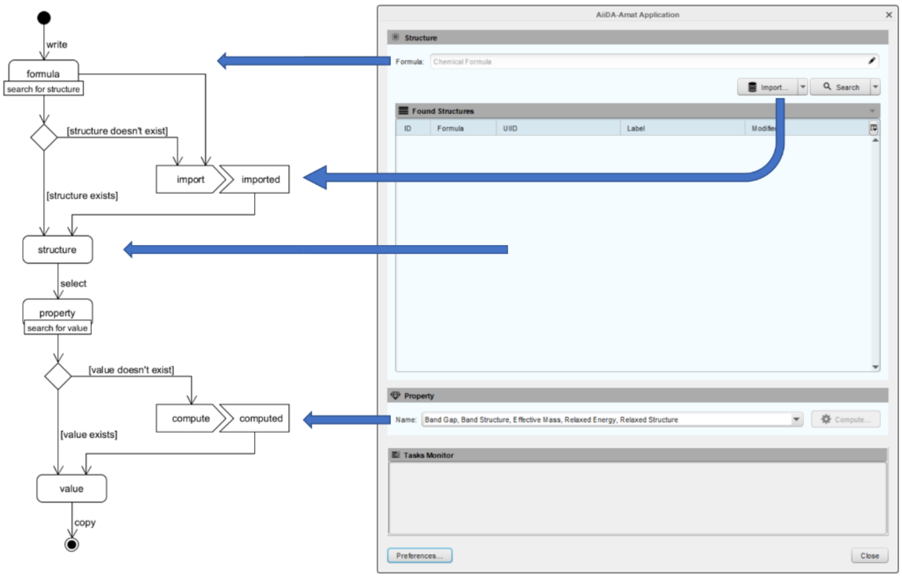

Figure 6 *Ginestra™-AiiDA interconnection workflow.*

The general idea is that the user searches for or imports a structure. Once the structure is shown in the search results table, then its existing properties can be viewed, and values taken (i.e., copied to the system clipboard). If a computable property doesn't exist for that structure, or a new value needs to be computed, then computation can be triggered and monitored up to its completion.

3.4.1 Search for Computable Properties

The first step of the plugin execution is to connect to the AiiDA service and to query for the list of computable properties.

REST API Used

```
/api/v4/intersect/properties
```

```
127.0.0.1 - - [14/Jan/2021 15:51:26] "GET /api/v4/intersect/properties HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2021 15:51:26] "GET /api/v4/users HTTP/1.1" 200 -
```

Output

The list of supported computable properties.

How Output Is Used

The returned list is used to populate the ***Name*** combo-box of the ***Property*** area.

The currently available computable properties are the following:

- *Band Gap*, *Band Structure*, *Effective Mass*, *Relaxed Energy*, *Relaxed Structure*; These are computed all together by a single algorithm;
- *Dielectric Constant*;
- *Defect Formation Energies*.

3.4.2 Structure Search

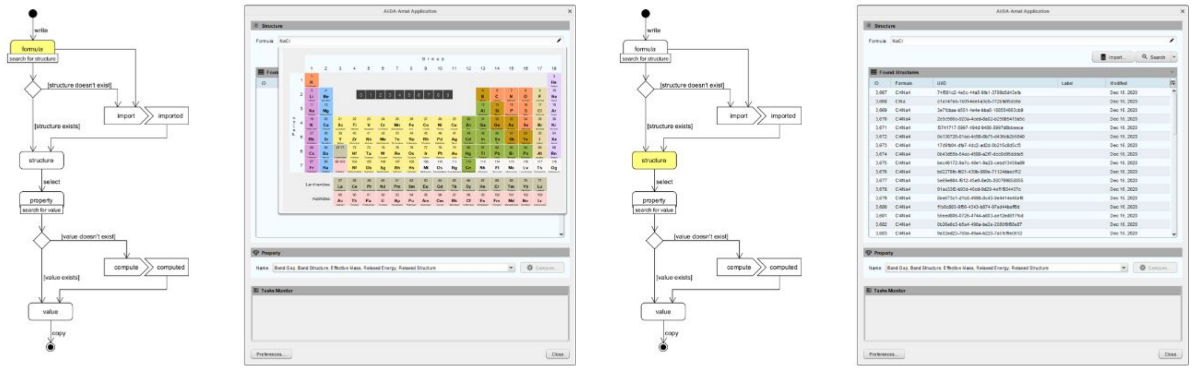

Figure 7 *Structure search.*

The user provides a chemical formula and the search returns the matching structures (Fig. 7).

REST API Used

/api/v4/intersect/derived_data/structuredata

```
127.0.0.1 - - [14/Jan/2021 16:06:17] "GET /api/v4/intersect/derived_data/structuredata?chemical_formula=ClNa&chemical_formula_type=hill_compact&limit=250&offset=250 HTTP/1.1" 200 -
```

Input

The formula string, the search mode (Hill or Hill Compact), whether to stop at the first found result or not.

Output

The list of structures matching the query parameters.

How Output Is Used

The returned list is used to populate the ***Found Structures*** table.

3.4.3 Structure Imports

If the material structure is not in the database, it can be imported either from COD or from a CIF file provided by the user.

Structure Import from COD

The user inserts the chemical formula of the structure to be imported then, after checking that from COD is selected in the ***Import…*** button drop-down menu, they press the ***Import…*** button. The application opens a dialog box showing the set of parameters the user can set before pressing the ***Import*** button (Fig. 8).

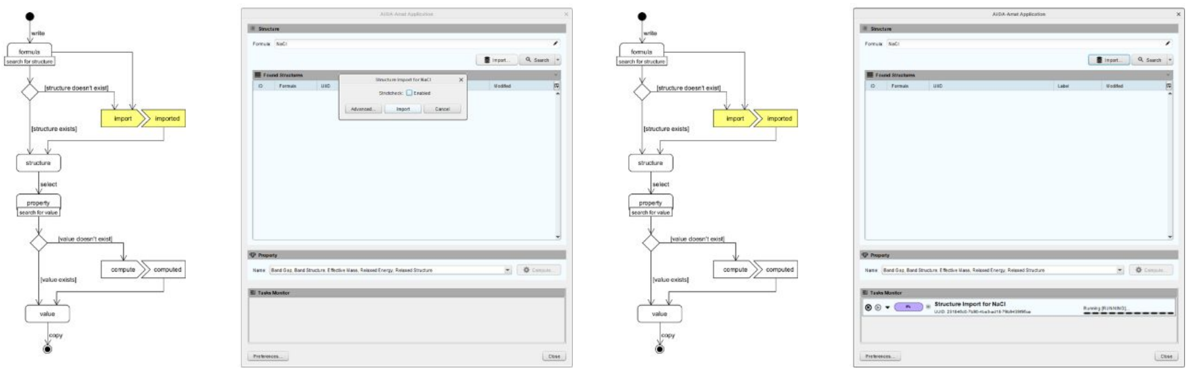

Figure 8 *Structure import from COD.*

For added flexibility, often desired by expert users, the ***Advanced…*** button allows the user to edit the raw JSON file to be submitted.

REST API Used

```
/api/v4/intersect/properties/structure.cod/inputs
/api/v4/intersect/submit
/api/v4/intersect/status/<node_id>
```

```
127.0.0.1 - - [14/Jan/2021 16:10:10] "GET /api/v4/intersect/properties/structure.cod/inputs HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2021 16:10:39] "POST /api/v4/intersect/submit HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2021 16:10:39] "GET /api/v4/intersect/status/e36f2246-25f0-4b36-9d58-4b5641bf97c8 HTTP/1.1" 200
```

Input

The formula string of the structure to be imported, and optional values for the `structure.cod` algorithm input parameters obtained from the first API call.

Output

The identifier of the processing request to be monitored.

How Output Is Used

A monitoring task is started by using the returned identifier as value for the `/api/v4/intersect/status/<node_id>` periodic API call.

Structure Import from CIF

The user, after checking that CIF is selected in the ***Import…*** button drop-down menu, presses the ***Import…*** button. The application opens a dialog box for selecting the CIF file to be imported (Fig. 9).

As in the previous case, for added flexibility the ***Advanced…*** button that allows the user to edit the raw JSON file to be submitted.

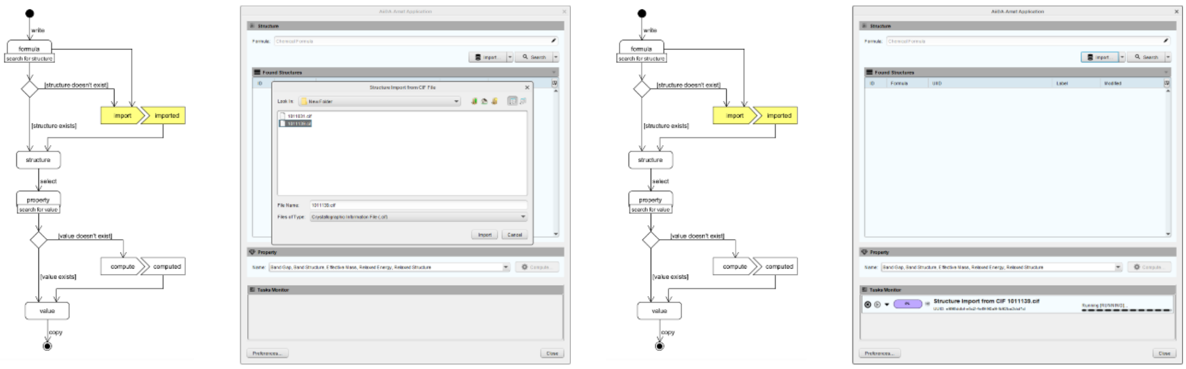

Figure 9 *Structure import from CIF.*

REST API Used

```
/api/v4/intersect/submit
/api/v4/intersect/status/<node_id>
```

```
(aiida) ubuntu@dev-aiidaginestra:~$ 127.0.0.1 - - [15/Jan/2021 14:35:22] "POST /api/v4/intersect/submit HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 14:35:22] "GET /api/v4/intersect/status/ac270e11-218f-47ea-8c60-040559f8b0ca HTTP/1.1" 200 -
```

Input

The CIF file to be imported.

Output

The identifier of the processing request to be monitored.

How Output Is Used

A monitoring task is started using the returned identifier as value for the `/api/v4/intersect/status/<node_id>` periodic API call.

3.4.4 Existing Properties Inspection

The user selects a found structure to inspect its content. The application opens a side panel showing the existing properties of the selected structure (Fig. 10).

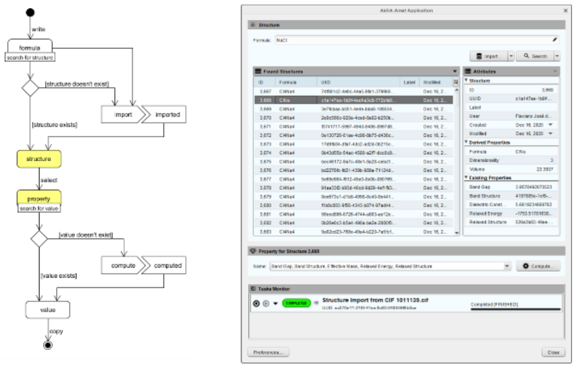

Figure 10 *Existing properties inspection.*

REST API Used

```
/api/ v4/nodes/<node_id>/contents/derived_properties
/api/v4/intersect/existing/properties/<property>/<node_id>
```

```
127.0.0.1 - - [15/Jan/2021 08:20:44] "GET /api/v4/nodes/4ac49247-47d5-499a-a645-0b227f838bce/contents/derived_properties HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:20:44] "GET /api/v4/nodes/84f0ddb1-d04c-4643-93fa-7bedb3b4fd65/contents/derived_properties HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:20:44] "GET /api/v4/nodes/7fd56c8f-8ebc-488f-9bbf-b93b8b45e081/contents/derived_properties HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:03] "GET /api/v4/intersect/existing/properties/band_gap.pw/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:04] "GET /api/v4/intersect/existing/properties/band_structure.pw/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:04] "GET /api/v4/intersect/existing/properties/band_gap.pw/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:04] "GET /api/v4/intersect/existing/properties/dielectric_constant.qe/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:04] "GET /api/v4/intersect/existing/properties/band_structure.pw/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:04] "GET /api/v4/intersect/existing/properties/dielectric_constant.qe/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:04] "GET /api/v4/intersect/existing/properties/effective_mass.pw/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:05] "GET /api/v4/intersect/existing/properties/formation_energies.qe/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:05] "GET /api/v4/intersect/existing/properties/effective_mass.pw/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:05] "GET /api/v4/intersect/existing/properties/relaxed_energy.pw/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:05] "GET /api/v4/intersect/existing/properties/formation_energies.qe/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:05] "GET /api/v4/intersect/existing/properties/relaxed_structure.pw/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:05] "GET /api/v4/intersect/existing/properties/relaxed_energy.pw/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:06] "GET /api/v4/intersect/existing/properties/relaxed_structure.pw/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:06] "GET /api/v4/intersect/existing/properties/structure.cif/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 08:21:06] "GET /api/v4/intersect/existing/properties/structure.cif/c1a147ee-1b9f-4eef-a3c8-772bfa95dc6e HTTP/1.1" 200 -
```

Input

The selected structure identifier and the names of the computable properties whose existing values are searched for.

Output

The derived and existing computable properties of the selected structure.

How Output Is Used

A side panel is displayed, showing:

- Information from the selected structure;
- List of the derived properties and their values;
- List of the already existing computable properties and their values. For each of the computable properties, more than one value can exist depending on how many computations were run on the structure.

Any element on the side panel can be selected and, through a contextual menu (triggered by a right-click), copied into the clipboard.

3.4.2 Property Computation

The user selects the sets of properties to be computed from the bottom combo-box, and then presses the ***Compute…*** button. The application opens a dialog box presenting the set of parameters, specific to the selected properties computation algorithm, the user can set before pressing the **Compute** button (Fig. 11).

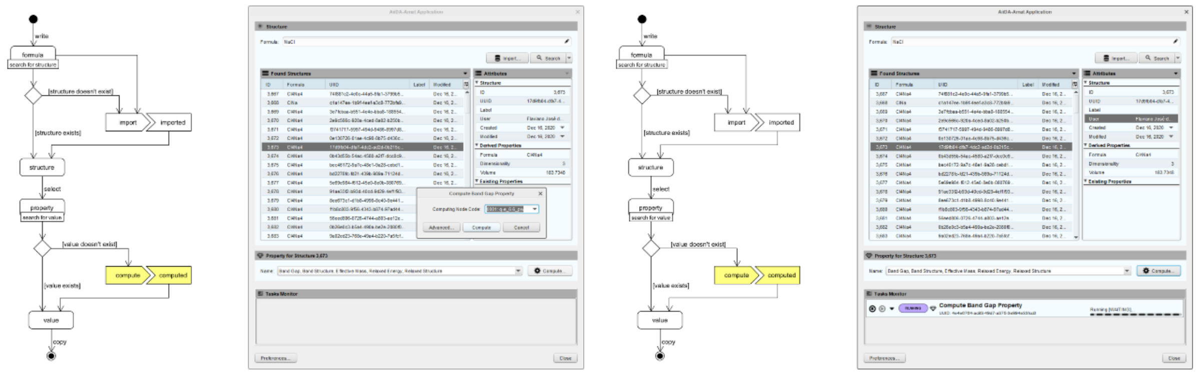

Figure 11 *Property Computation.*

For added flexibility, the ***Advanced…*** button allows the user to edit the raw JSON file to be submitted.

REST API Used

```
/api/v4/intersect/properties/<property>/inputs
/api/v4/intersect/submit
/api/v4/intersect/status/<node_id>
```

```
127.0.0.1 - - [15/Jan/2021 09:01:31] "GET /api/v4/intersect/properties/band_gap.pw/inputs HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 09:01:47] "POST /api/v4/intersect/submit HTTP/1.1" 200 -
127.0.0.1 - - [15/Jan/2021 09:01:47] "GET /api/v4/intersect/status/4e4e6764-ac93-49d7-a375-3e994e33fcc0 HTTP/1.1" 200 -
```

Input

The selected structure identifier, and the selected computable property to be computed.

Output

The identifier of the processing request to be monitored.

How Output Is Used

A monitoring task is started, using the returned identifier as value for the /api/v4/intersect/status/<node_id> periodic API call

3.5 Device-to-material (D2M) workflow

As shown in Fig. 3, Device-to-Material D2M workflow is responsible for querying the material database, and looking for properties that are consistent with the ones extracted by using the Ginestra™ Defect Discovery Tool (DDT), which allows to reproduce specific electrical device characteristics (e.g. I-V curve). The workflow starts from either experimental or device simulation data that can be processed by Ginestra™ to identify/determine specific material properties (the spatial and energetic distribution of defects, the dielectric constant, etc.). This responds to two specific industry needs:

- To characterize fabricated materials and devices to understand, for example, if/how different process/fabrication conditions affect material and trap properties. This is a common and key need during the development of a new technology. Test devices are fabricated and measured to evaluate if the electrical behavior satisfies the required specifications. The D2M workflow of the IM2D box can be applied to the measured electrical data, and provide critical feedback and understanding to guide the technology development

- To optimize an existing device/technology or design a new one. In this case, extensive device-level simulations (Ginestra™) can be used to screen different material/device solutions allowing to identify good candidates that meet target specifications. The output of this device simulations can be a set of desired material (e.g. band-gap, dielectric constant) and trap (e.g. energy level, defect concentration) parameters. The D2M workflow of the IM2D box can be applied to these outputs in order to identify the best material candidates with the required properties.

The D2M workflow is still under development. This paragraph describes the D2M flux diagram that will be implemented in the next months.

3.5.1 First block: Material Parameters extraction from Device characteristics.

The unknown physical parameters and the properties of the traps (e.g. energy distribution) of a material in a specific device can be obtained from two possible input sources, namely experimentally data input and simulation data input, as schematized in Fig. 12 and 14 respectively.

Experimental data input (Fig. 12)

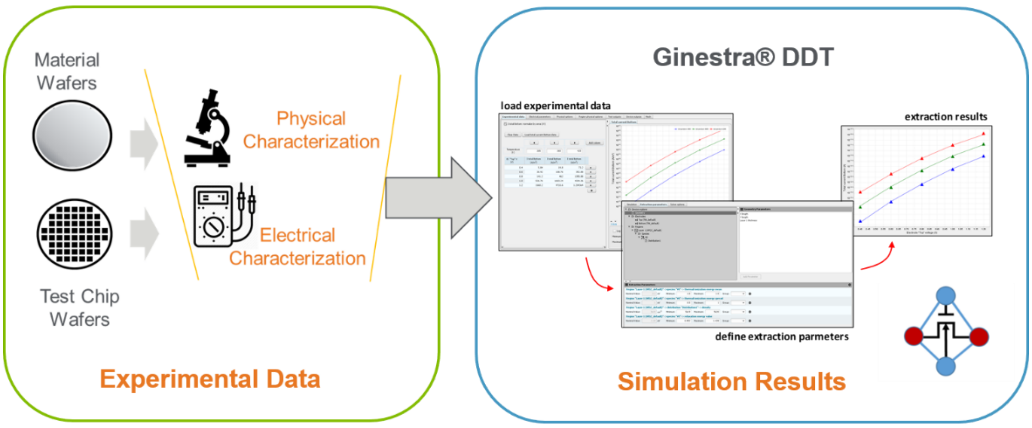

Figure 12 *D2M First Block, experimental data input.*

Input

Experimental data

Process Description

In Ginestra™, the DDT is used to extract relevant material parameters and trap energy/space distribution from experimental data (Fig. 13)

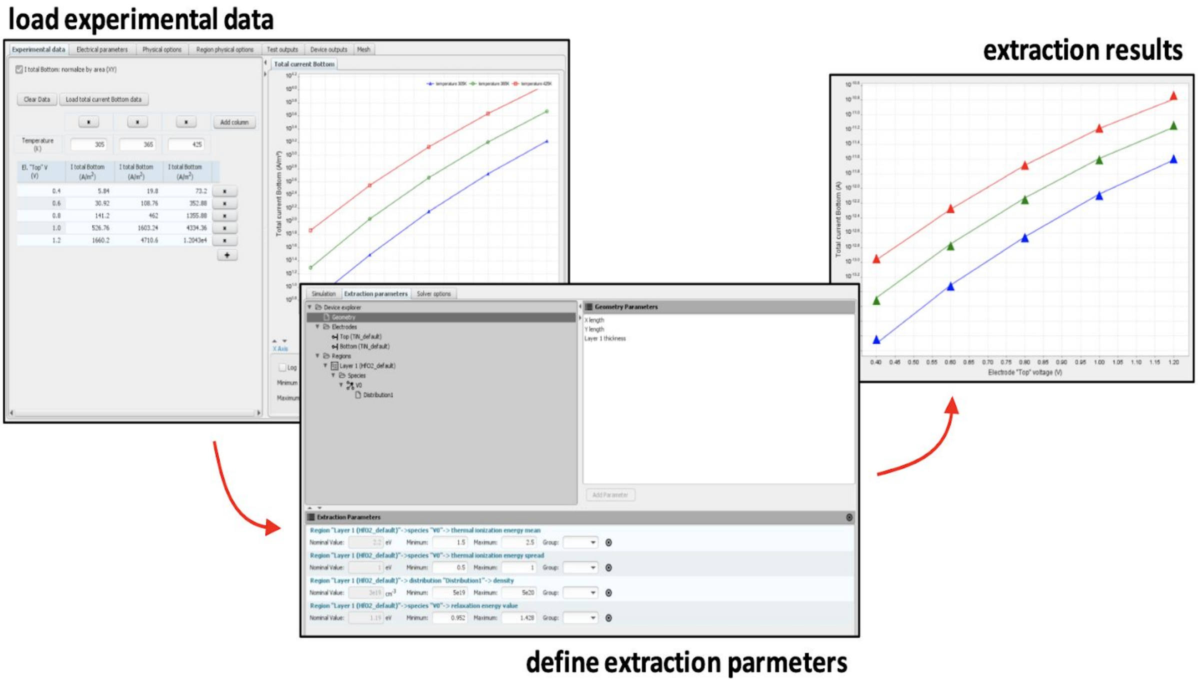

***Figure 13** Ginestra™ DDT.*

Output

Values of the materials and traps properties that allow to reproduce the experimental characteristics.

Simulation data input (Fig. 14)

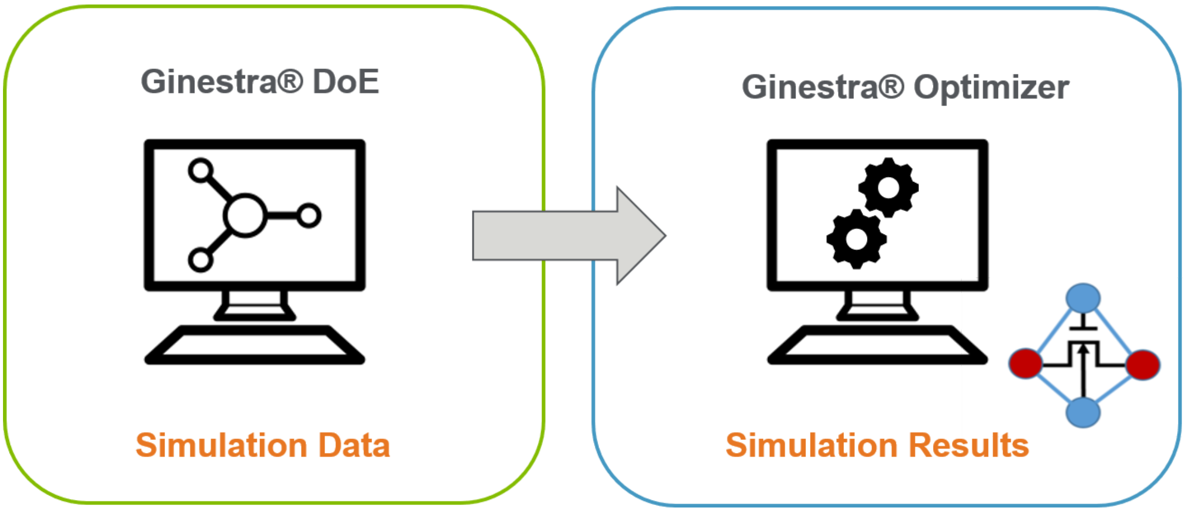

Figure 14 *D2M First Block simulation data input.*

Input

Simulation data

Process Description

Ginestra™ DoE, Ginestra™ Optimizer

Output

Value of the materials and traps properties that are required to obtain the simulated characteristics.

3.5.2 Second Block: INTERSECT plugin for Ginestra™ application.

The plugin queries the AiiDA database via the REST-API looking for the materials that match the outputs of the first block.

Query may be refined according to specific filters: candidate materials list, variability range for the parameter, concatenation of multiple conditions (e.g. the material should satisfy both the band-gap and the dielectric constant values), as shown in Fig. 15.

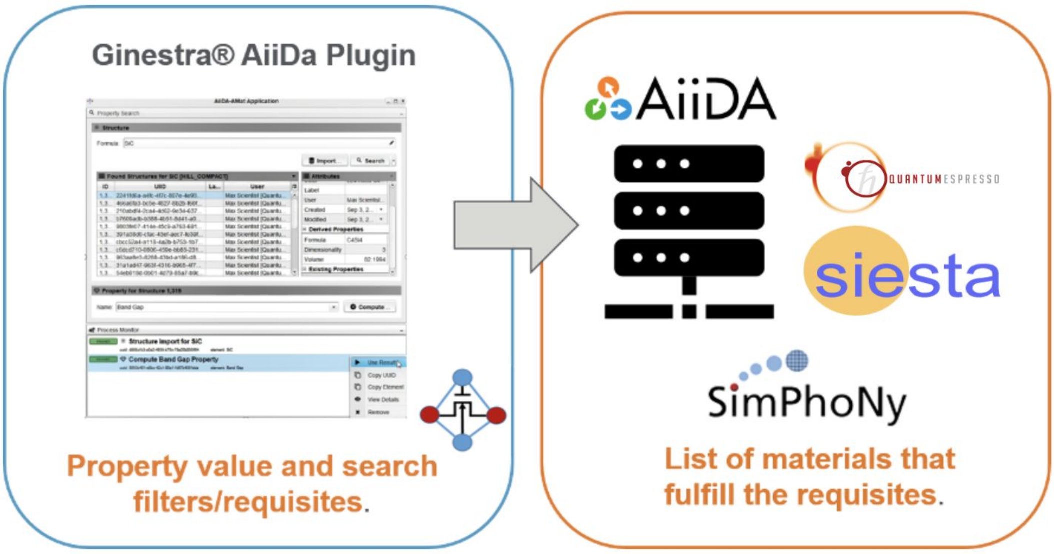

Figure 15 *INTERSECT plugin for Ginestra™ application.*

Input

Property value and search filters/requisites.

Process Description

Query to a specific endpoint of the AiiDA-POST REST-API.

Output

List of materials that fulfill the requisites and/or of the atomic defects matching the extracted material traps properties.

In case a list of candidate materials is provided (eventually obtained from selecting the best candidates in the AiiDA database on the basis of a preliminary query), the search is restricted to those materials for which the desired property has already been computed.

If a specific “computation” field is explicitly enabled in the query, the property computation workflow is contextually submitted for all the materials in the list having no associated value to that property.

4 Conclusions

The deliverable shows the IM2D iHub improvement for the M2D and D2M pathways. After depicting the general interconnection framework, we have described the AiiDA workflows interconnection (explained in details in D2.3) for the M2D workflow identified features (refer to D1.1):

- Band Gap;
- Band Structure;
- Effective Mass;
- Relaxed Energy;
- Relaxed Structure;
- Dielectric Constant;
- Formation Energies.

Moreover, we have described the iHub functionalities, such as the structure search and structure import (both COD in CIF version), and the property inspection and property computation. Finally, regarding the D2M, we have described its general concept and the overall framework. D2M proof of concept was successfully validated by the OTS structure (D3.2). In the next few months, we are going to implement the D2M workflow and complete the remaining M2D workflows, such as NEB – activation energy for diffusion.

References

[1] INTERSECT project www.intersect-project.eu

[2] Giannozzi P., et al. Quantum ESPRESSO: a modular and open-source software project for quantum simulations of materials. J. Phys. Cond. Matt. 21, 395502 (2009). Quantum ESPRESSO URL: www.quantum-espresso.org

[3] Soler J. M., et al. The SIESTA method for ab initio order-N materials simulation. J. Phys. Cond.Matt. 14, 2745 (2002). SIESTA URL http://departments.icmab.es/leem/siesta

[4] Larcher L., et al. A simulation framework for modeling charge transport and degradation in high-k stacks, J. Comput. Electron.12, 658–665 (2013). Ginestra™ URL: www.mdlsoft.com

[5] Adler, J., et al. Visualization in the integrated SimPhoNy multiscale simulation framework, Comp. Phys. Comm. 231, 45-61 (2018). SimPhoNy URL: www.simphony-project.eu

[6] Pizzi G., et al. AiiDA: automated interactive infrastructure and database for computational science. Comp. Mat. Sci. 111, 218 – 230 (2016). AiiDA URL: http://www.aiida.net

ACRONYMS

API - Application Programming Interface

CIF - Crystallographic Information File

COD - Crystallography Open Database

D2M - Device-to-material

DFT - Density Functional Theory

EMMO - European Materials Modelling Ontology

GUI - Graphical User Interface

iHub - Interoperability Hub

IM2D - Interoperable Materials-To-Device

JSON - JavaScript Object Notation.

M2D - Material-to-device

QE - Quantum Espresso

REST - Representational State Transfer

UUID - Universally Unique Identifier