

BMBF Ausschreibung »MaterialDigital«

Anknüpfung und Zusammenarbeit mit der Plattform MaterialDigital – **Workflows** (T. Hickel, MPIE)

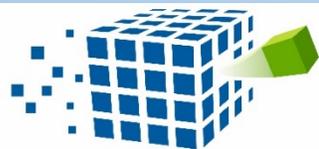
GEFÖRDERT VOM



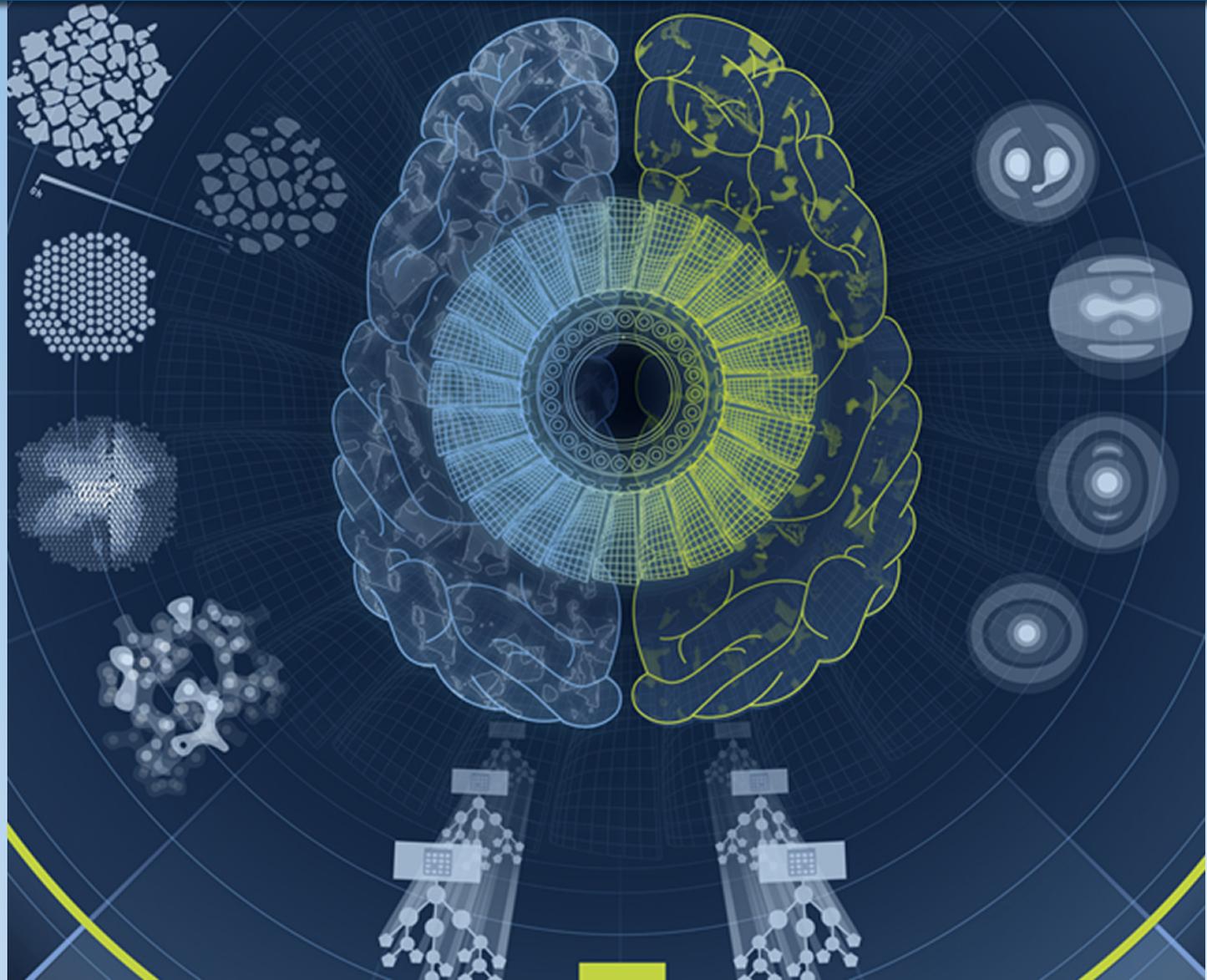
Bundesministerium
für Bildung
und Forschung



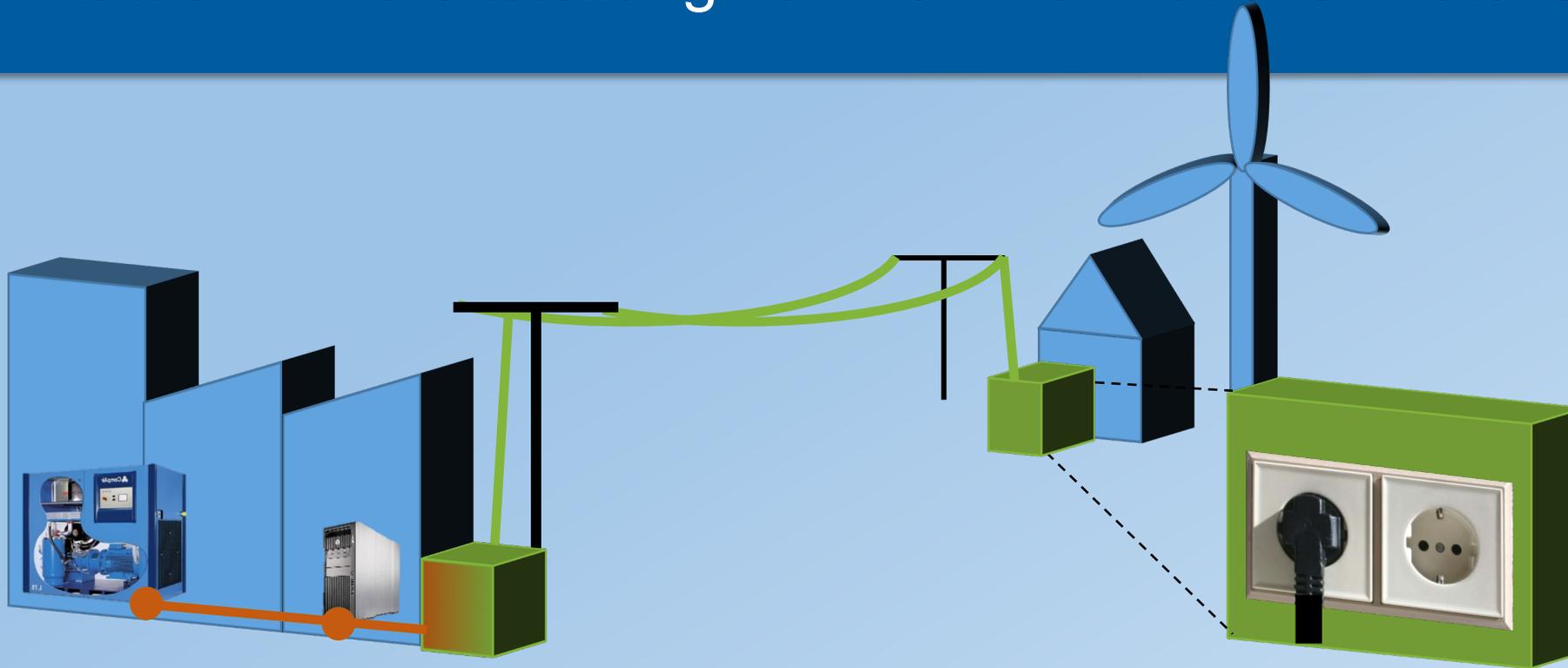
MAX-PLANCK-INSTITUT FÜR
EISENFORSCHUNG GMBH



MATERIALDIGITAL



Plattform: Bereitstellung von Kommunikations-Protokoll



Standardisiertes Stromnetz

- AC mit 50 Hz
- 220 V
- Schuko

für Energiekonsument und -erzeuger (Wechselrichter)



- Standards zwischen Kunden = Standards im Unternehmen
- Alternative Standards innerhalb von Geräten (Niedrigspannung)
- Individuelle Standards erschweren Nutzung etablierter Geräte

Ähnlicher Standards für den Umgang mit Daten und Workflows erforderlich

Kommunikations-Protokoll für Workflows



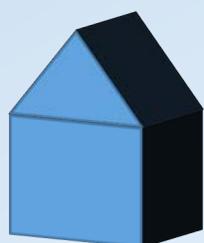
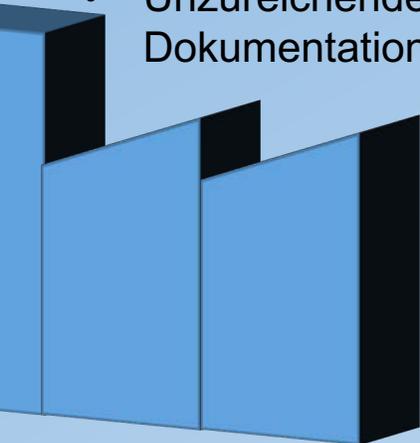
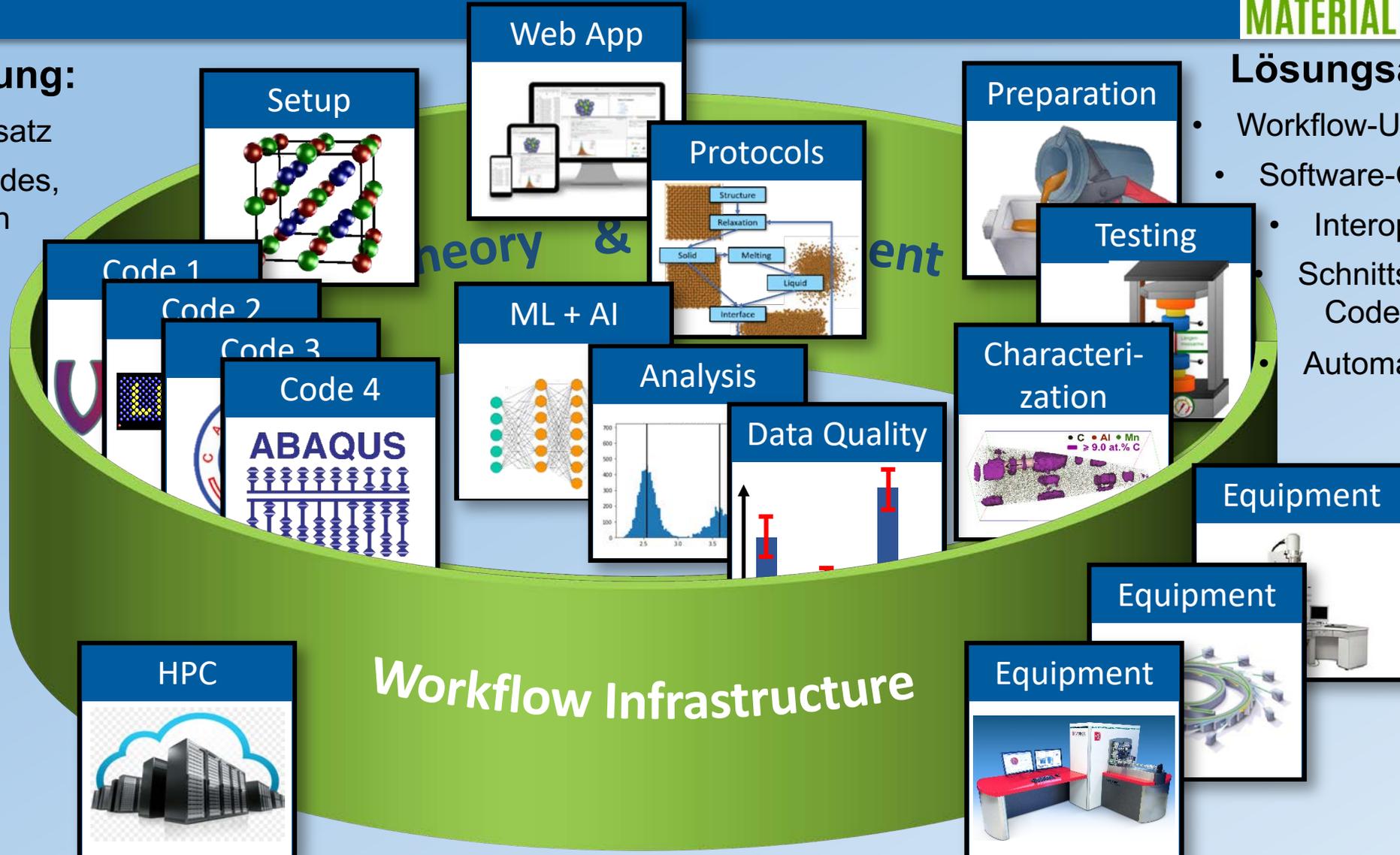
MATERIALD1G1TAL

Herausforderung:

- Multiskalen-Ansatz
- Vielzahl der Codes, Geräte-Treibern
- Inkompatible Codes
- Inkompatible Datenformate
- Unzureichende Dokumentation

Lösungsansatz:

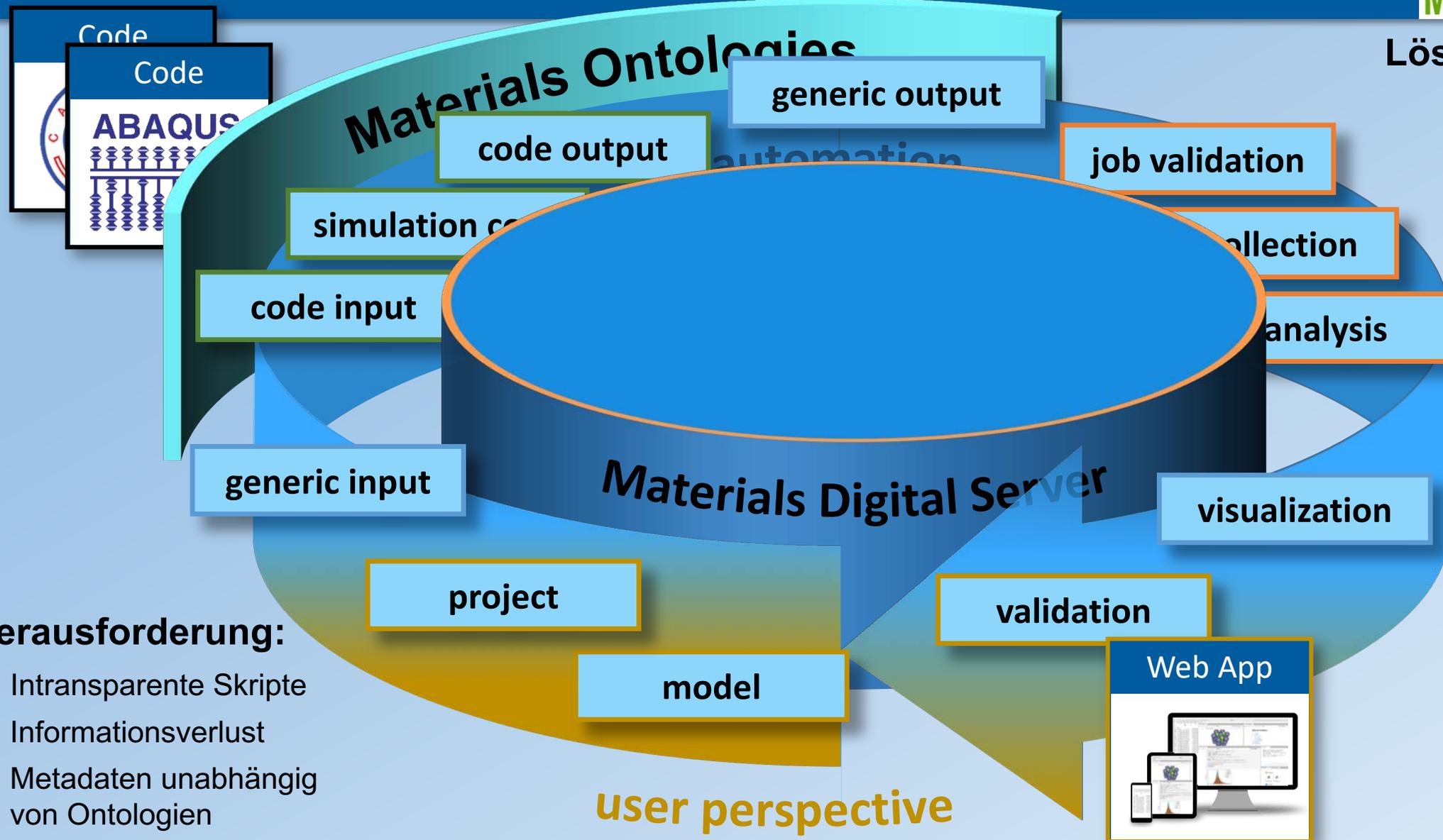
- Workflow-Umgebung
- Software-Container
- Interoperabilität
- Schnittstellen für Codes, Treiber
- Automatisierung



Generische Struktur von Workflows



MATERIALD1G1TAL



Lösungsansätze:

- Automatisierte, standardisierte Workflows
- Vereinheitlichte Serverstruktur
- Transparentes Web-Interface
- Verknüpfung Workflow und Ontologie

→ Vortrag Niebel

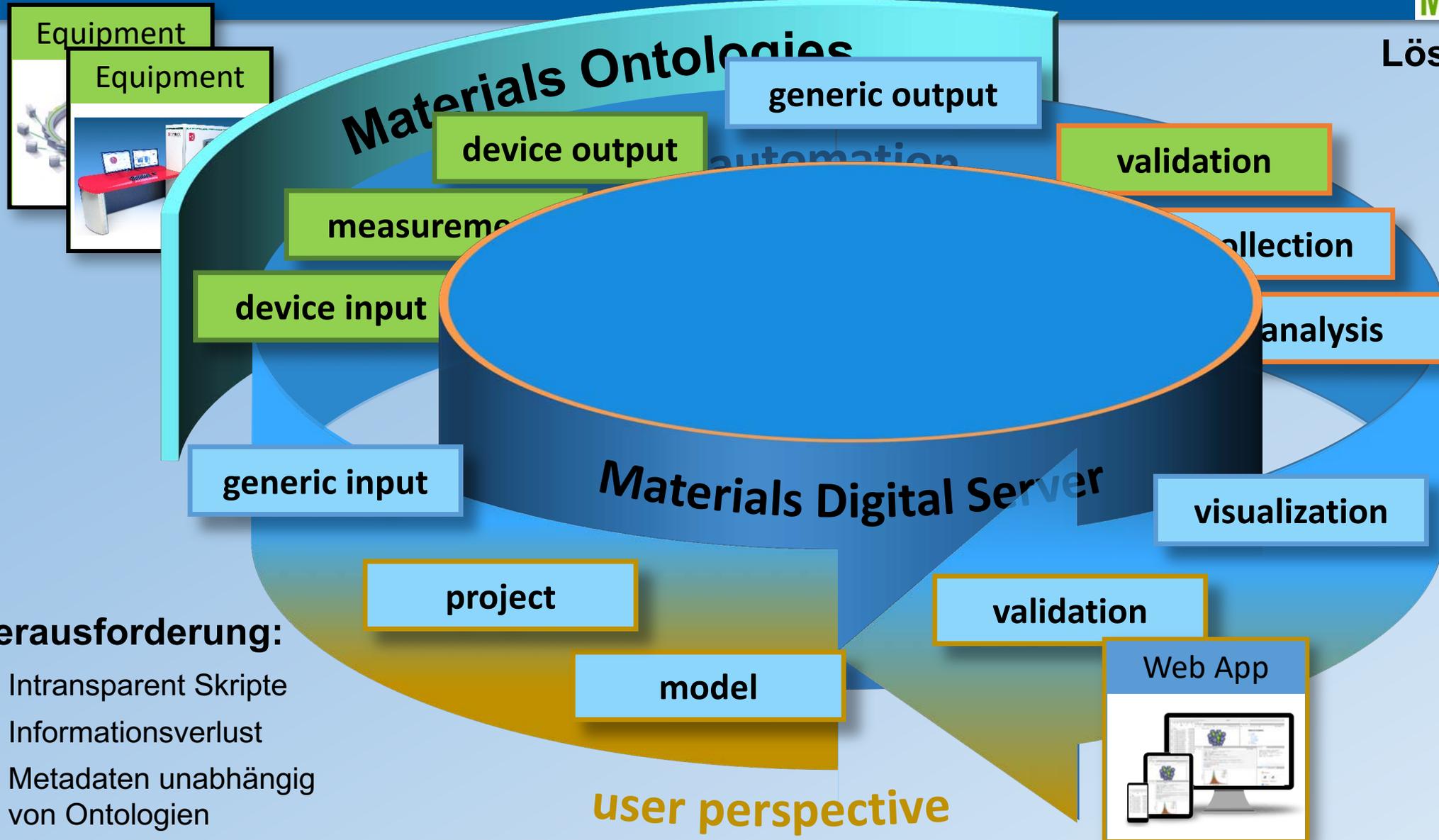
Herausforderung:

- Intransparente Skripte
- Informationsverlust
- Metadaten unabhängig von Ontologien

Generische Struktur von Workflows



MATERIALD1G1TAL



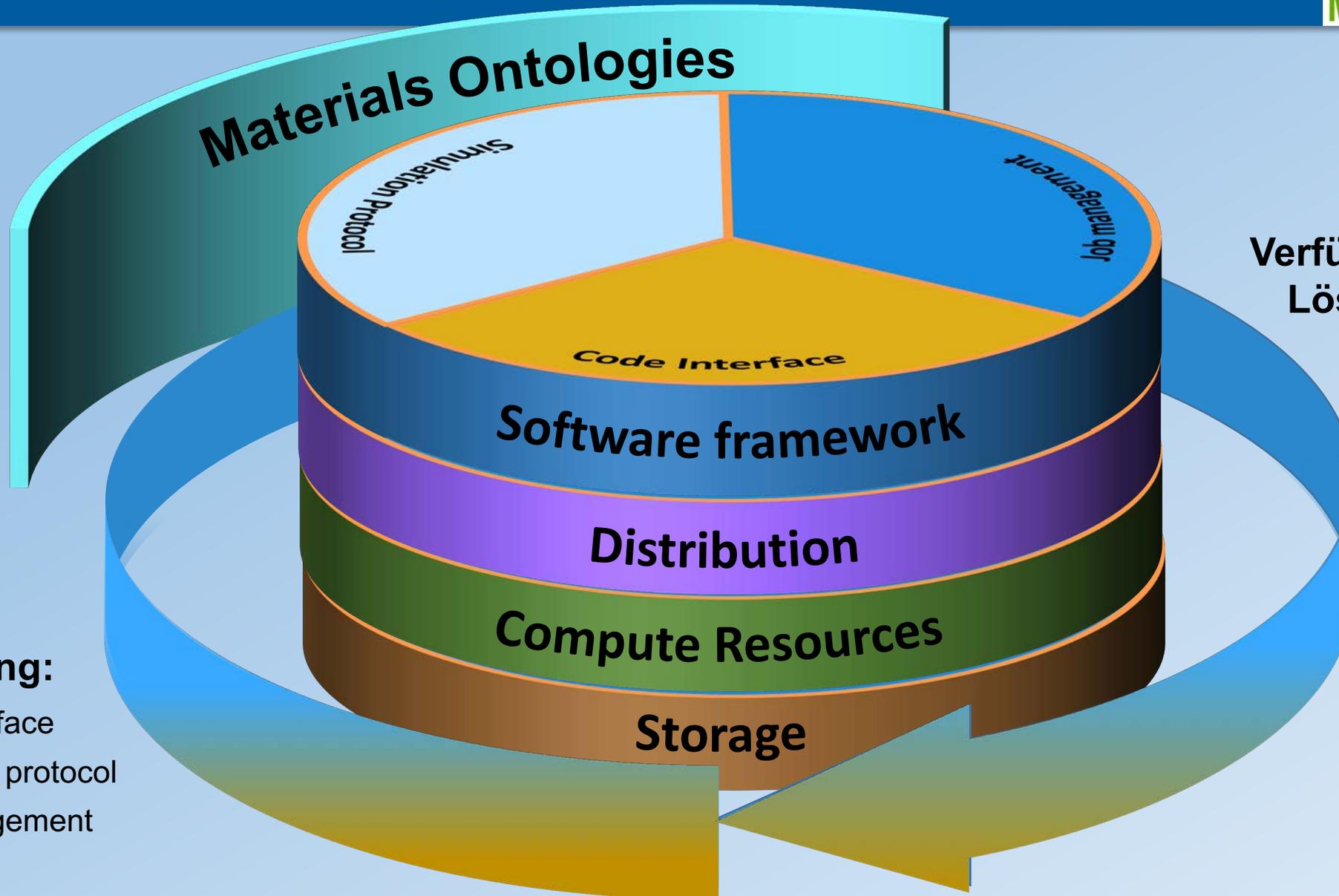
Lösungsansätze:

- Automatisierte, standardisierte Workflows
- Vereinheitlichte Serverstruktur
- Transparentes Web-Interface
- Verknüpfung Workflow und Ontologie

→ Vortrag Niebel

Herausforderung:

- Intransparent Skripte
- Informationsverlust
- Metadaten unabhängig von Ontologien



Verfügbare Workflow-Lösungen (Auswahl):



Realisierung:

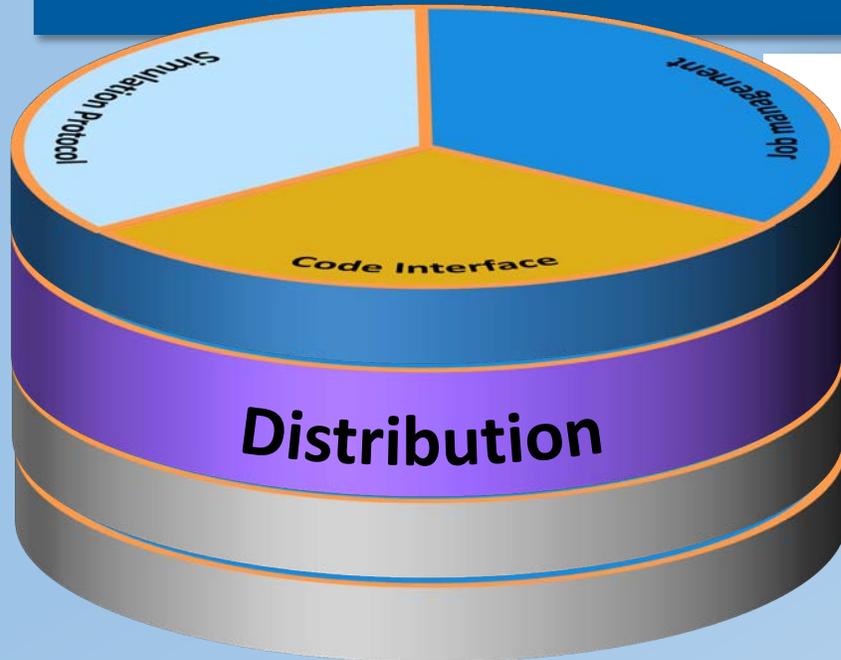
- Code interface
- Simulation protocol
- Job management

Distribution – Anknüpfung an die Plattform



MATERIALD1G1TAL

<https://www.materialdigital.de>



Bundesanzeiger

Herausgegeben vom
Bundesministerium der Justiz
und für Verbraucherschutz
www.bundesanzeiger.de

Bekanntmachung

Veröffentlicht am Freitag, 20. September 2019
BAnz AT 20.09.2019 B4
Seite 1 von 7

**Bundesministerium
für Bildung und Forschung**

Bekanntmachung
zur Förderung von Zuwendungen von Vorhaben
im Rahmen der Initiative zur Digitalisierung der Materialforschung in Deutschland (*MaterialDigital*)

Vom 20. August 2019

BMBF: „Es sollen im „**App-Store**“ der Innovationsplattform Modelle entstehen, [um] komplexe Workflows über die zugrundeliegende Architektur auf forschungsrelevante „echte“ Daten anzuwenden.“

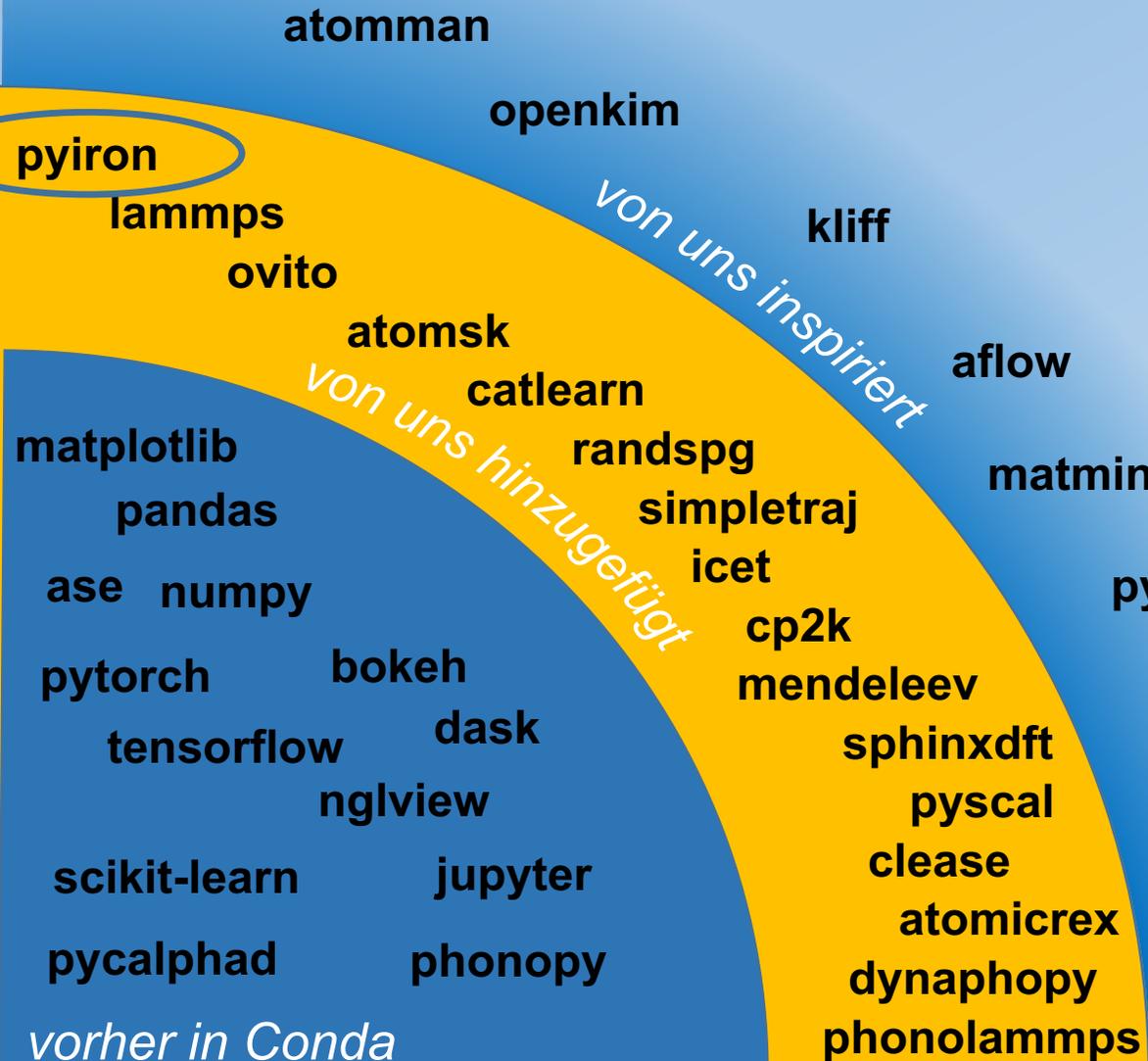
Konsequenz für Plattform:

Ein Server zum Teilen und Verwalten von Tools wird eingerichtet. Die Workflowumgebungen werden im **Conda Community Channel** zum Download bereitgestellt, um eine reibungslose Installation auf verschiedenen Computersystemen zu ermöglichen. Die Tools werden entsprechend ihrer Abhängigkeiten integriert.

Konsequenz für Zuwendungsempfänger:

- Software-Lösungen müssen als **Conda-Paket** zur Verfügung gestellt werden.
- Damit diese in den App-Store integriert werden können, muss es sich um **open-access** Software handeln.
- Für jedes Tool muss es ein **einziges Shell-Script** geben, das aus dem Input den Output generiert.

Verbreitung über Conda



Einfache Installation wissenschaftlicher Software:

```
conda install -c conda-forge pyiron
```



Vorteile:

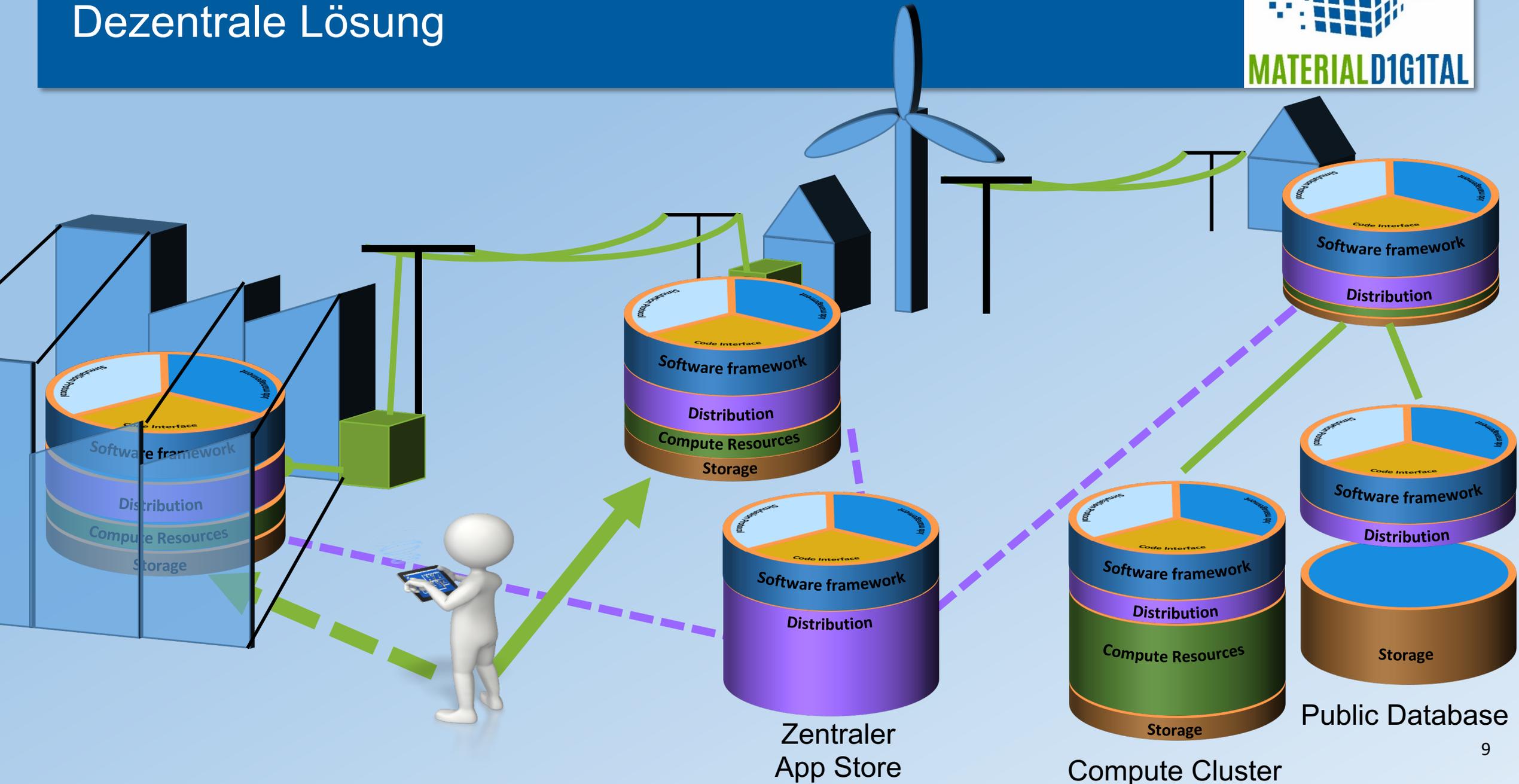
- Channel technische Realisierung eines AppStore
- Gelinkt zu definierter Python-Version
- Abhängigkeit von Paketen geklärt
- Unabhängigkeit von Betriebssystem

pyiron:

- Integrated Development Environment für Workflows und Apps



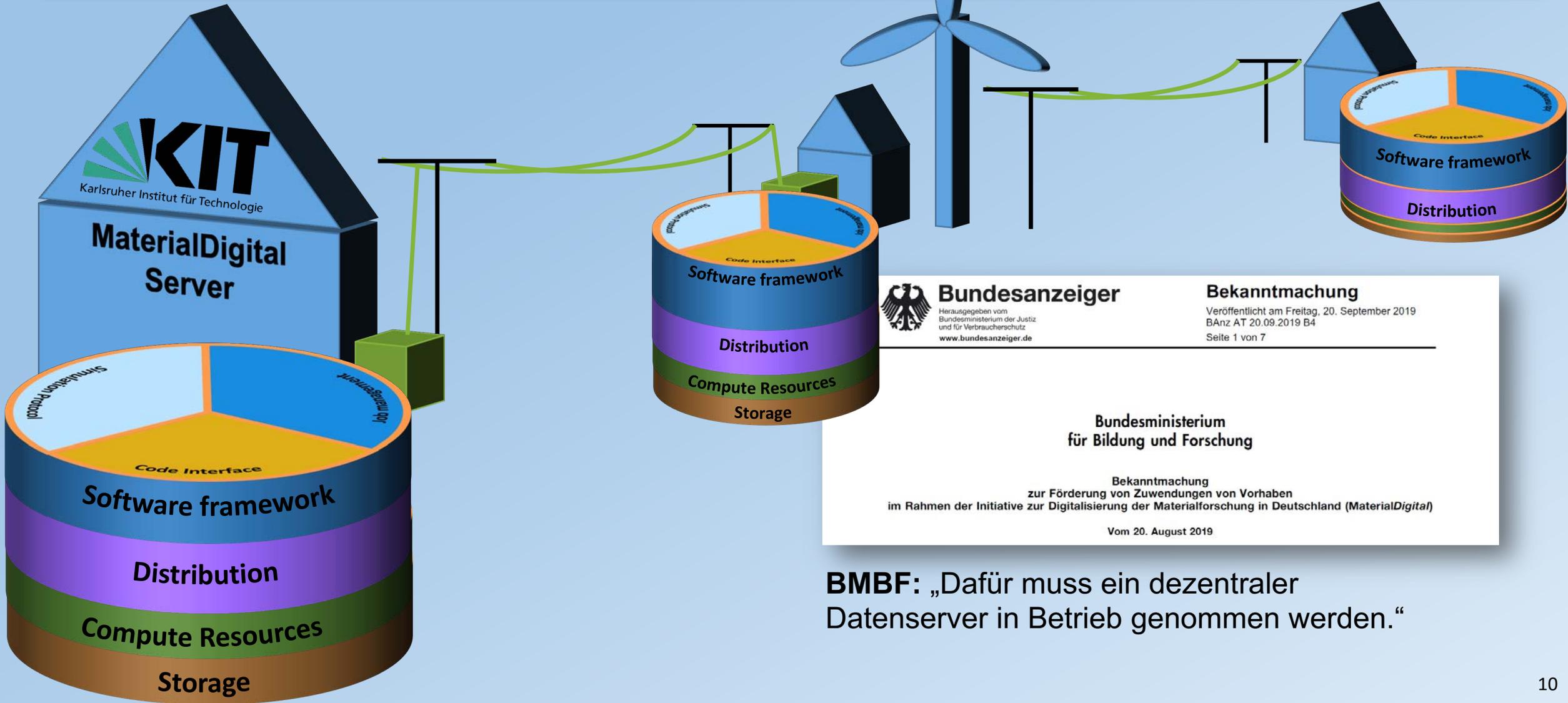
Dezentrale Lösung



Dezentrale Lösung



MATERIALD1G1TAL



Bundesanzeiger

Herausgegeben vom
Bundesministerium der Justiz
und für Verbraucherschutz
www.bundesanzeiger.de

Bekanntmachung

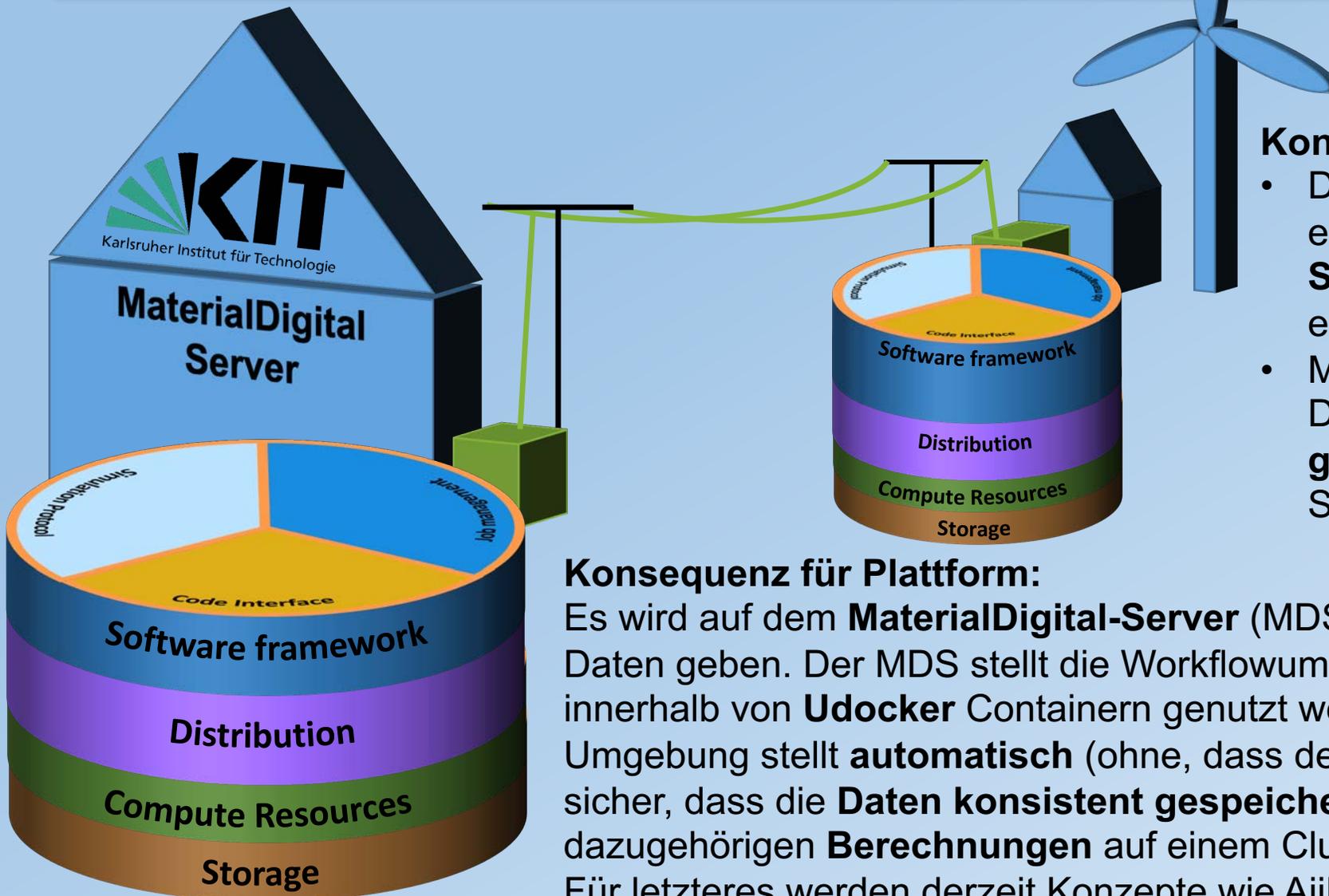
Veröffentlicht am Freitag, 20. September 2019
BAnz AT 20.09.2019 B4
Seite 1 von 7

**Bundesministerium
für Bildung und Forschung**

Bekanntmachung
zur Förderung von Zuwendungen von Vorhaben
im Rahmen der Initiative zur Digitalisierung der Materialforschung in Deutschland (*MaterialDigital*)

Vom 20. August 2019

BMBF: „Dafür muss ein dezentraler
Datenserver in Betrieb genommen werden.“



Konsequenz für Zuwendungsempfänger:

- Der Zuwendungsempfänger muss einen eigenen **Server zur Erzeugung und Speicherung** eigener Simulationsdaten einrichten oder dafür den MDS nutzen.
- Mittelfristig wird angestrebt, dass lokale Datenserver als Softwarestack vom **MDS geklont** werden können, dann also die Softwarestruktur des MDS aufweisen.

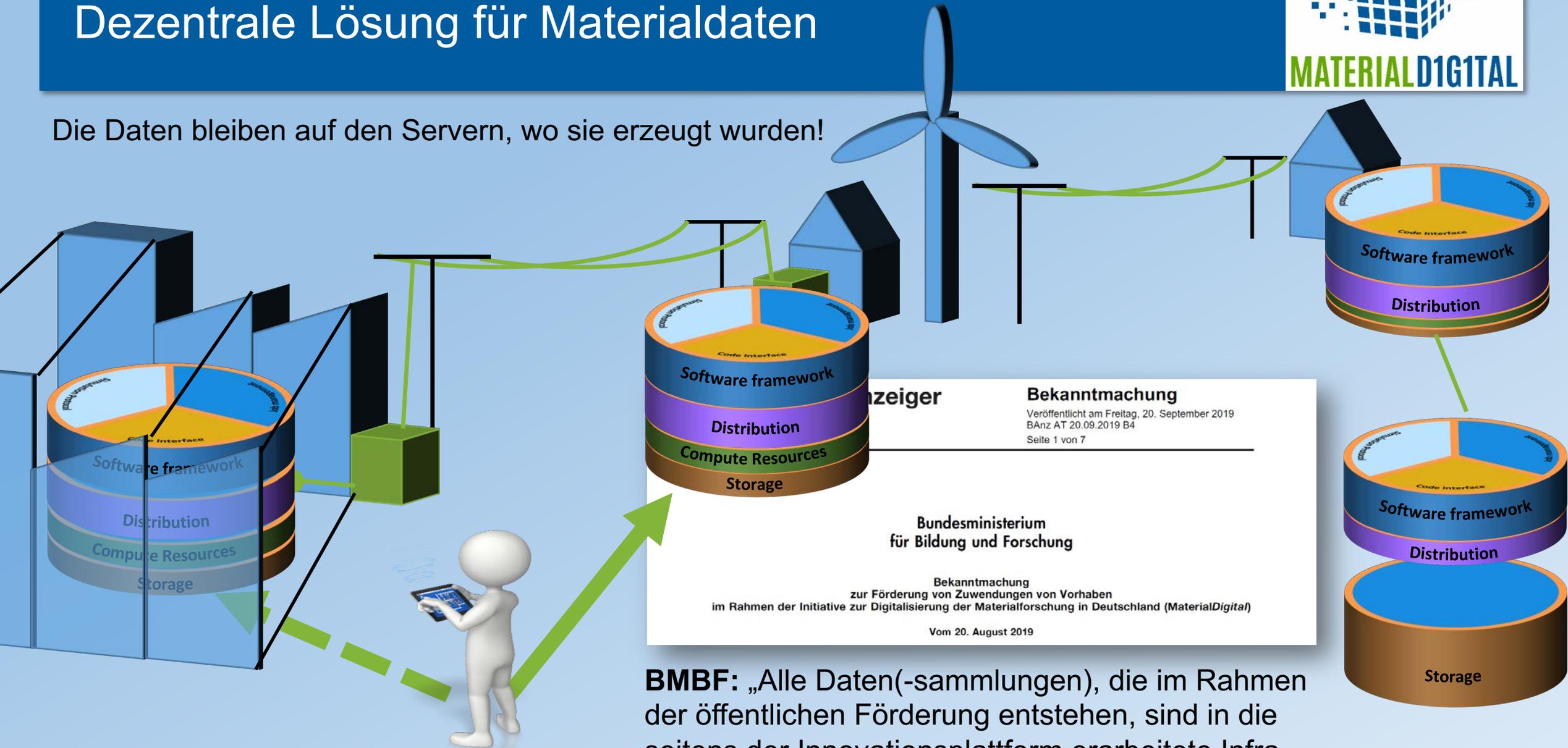
Konsequenz für Plattform:

Es wird auf dem **MaterialDigital-Server** (MDS) ausreichend Speicherplatz für Daten geben. Der MDS stellt die Workflowumgebungen zur Verfügung, die innerhalb von **Udocker** Containern genutzt werden können. Die Verwendung der Umgebung stellt **automatisch** (ohne, dass der Anwender Details kennen muss) sicher, dass die **Daten konsistent gespeichert** werden und dass die dazugehörigen **Berechnungen** auf einem Cluster ausgeführt werden. Für letzteres werden derzeit Konzepte wie AiiDA und SLURM unterstützt.

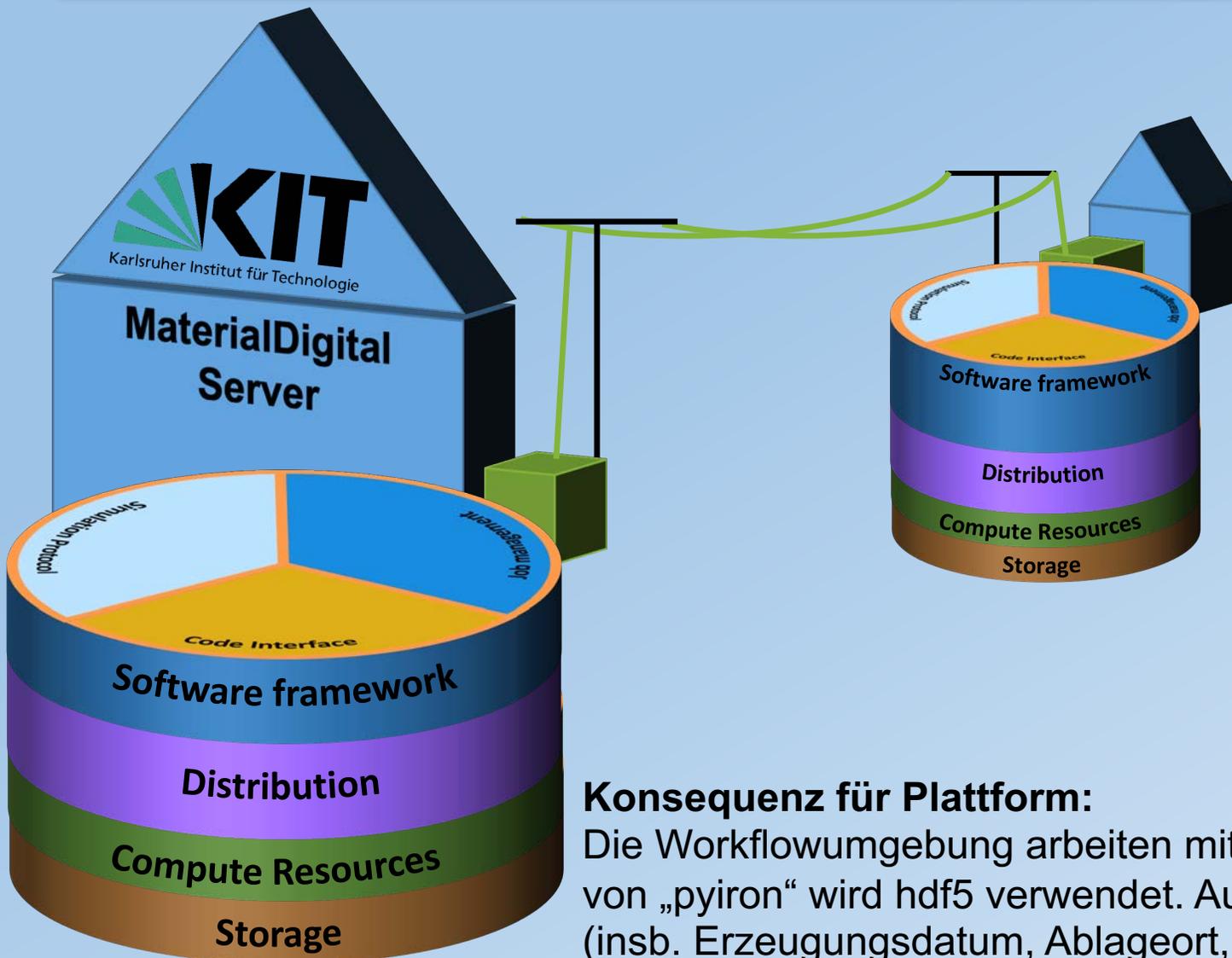
Dezentrale Lösung für Materialdaten



Die Daten bleiben auf den Servern, wo sie erzeugt wurden!



BMBF: „Alle Daten(-sammlungen), die im Rahmen der öffentlichen Förderung entstehen, sind in die seitens der Innovationsplattform erarbeitete Infrastruktur für verteilte Materialdaten einzupflegen.“



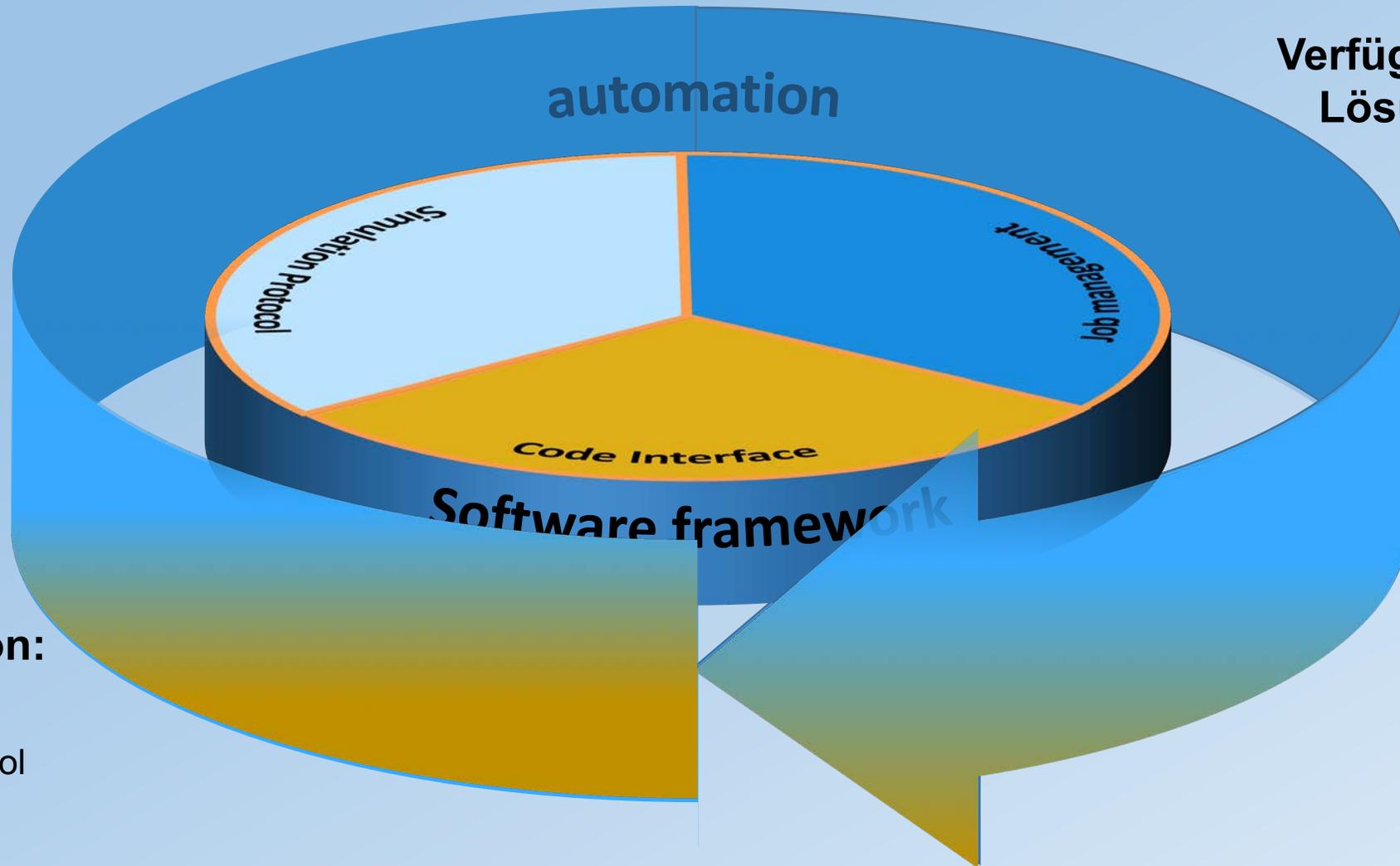
Konsequenz für Zuwendungsempfänger:

- Um die Einheitlichkeit sicherzustellen, müssen alle Daten (Input und Output von Tools) als **Python Dictionaries** oder alternativen **strukturierten Dateiformaten** wie hdf5, JSON verfügbar sein.
- Die Daten müssen **vollständig** in diesen Formaten vorliegen (d.h. alle Informationen müssen aus den Daten erzeugbar sein), um in die Plattform integrierbar zu sein.
- Die Daten müssen an eine **generische Ontologie gekoppelt** sein.
- Die **Metadaten** (insb. Erzeugungsdatum, Ablageort, Zugriffsrechte) werden an die Plattform zur Ablage auf den MDS übermittelt.

Konsequenz für Plattform:

Die Workflowumgebung arbeiten mit klar **strukturierten Datenformaten**. Im Falle von „pyiron“ wird hdf5 verwendet. Auf dem MDS werden die generischen **Metadaten** (insb. Erzeugungsdatum, Ablageort, Zugriffsrechte) der erzeugten Daten abgelegt.

Generische Struktur von Workflows



Verfügbare Workflow-Lösungen (Auswahl):



Realisierung von:

- Code interface
- Simulation protocol
- Job management

Workflows – Anknüpfung an die Plattform

<https://www.materialdigital.de>



MATERIALD1G1TAL



Bundesanzeiger

Herausgegeben vom
Bundesministerium der Justiz
und für Verbraucherschutz
www.bundesanzeiger.de

Bekanntmachung

Veröffentlicht am Freitag, 20. September 2019
BAnz AT 20.09.2019 B4
Seite 1 von 7

Bundesministerium
für Bildung und Forschung

Bekanntmachung
zur Förderung von Zuwendungen von Vorhaben
im Rahmen der Initiative zur Digitalisierung der Materialforschung in Deutschland (*MaterialDigital*)

Vom 20. August 2019

BMBF: „Etablierung von digitalen Workflows im Sinne des dezentralen Daten- oder Simulationskonzepts durch aktive Agenten innerhalb der **Software-Umgebung der Innovationsplattform.**“

Konsequenz für Plattform:

Es werden die Software-Umgebungen „**pyiron**“ und „**SimStack**“ zur Verfügung gestellt. Diese dienen der Programmierung und Ausführung von Simulationsprotokollen.



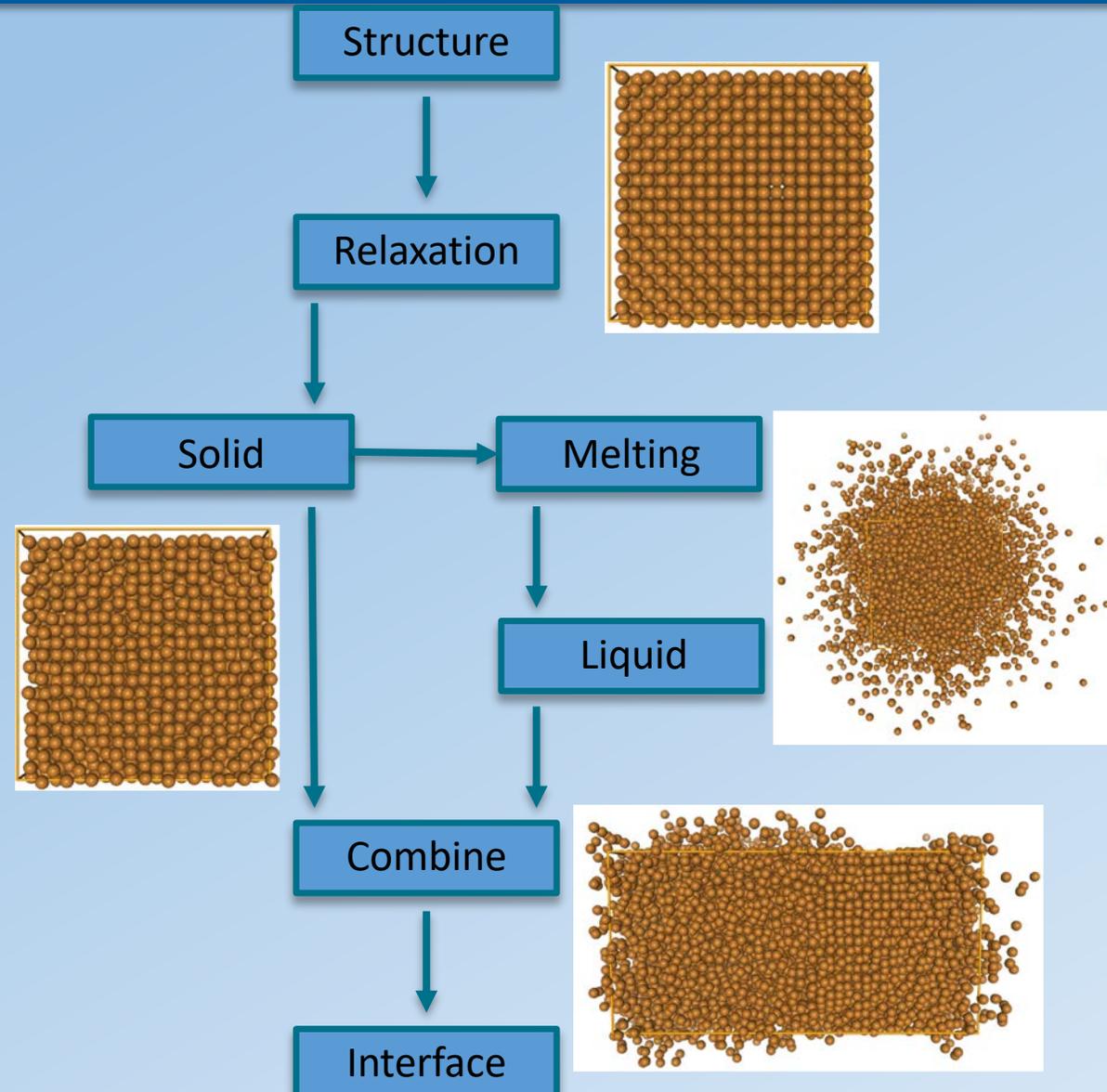
Konsequenz für Zuwendungsempfänger :

- Für jede Software-Lösung müssen die **Voraussetzungen** geschaffen werden, damit sie auf einer der Workflowumgebungen „pyiron“ oder „SimStack“ lauffähig ist.
- Es wird empfohlen, dass dies über ein **Python-Interface** (idealerweise Jupyter) erfolgt.
- Es wird nicht ausgeschlossen, dass die hier geschaffenen aktiven Agenten von **weiteren Serverumgebungen** außerhalb der Plattform zusätzlich unterstützt werden.

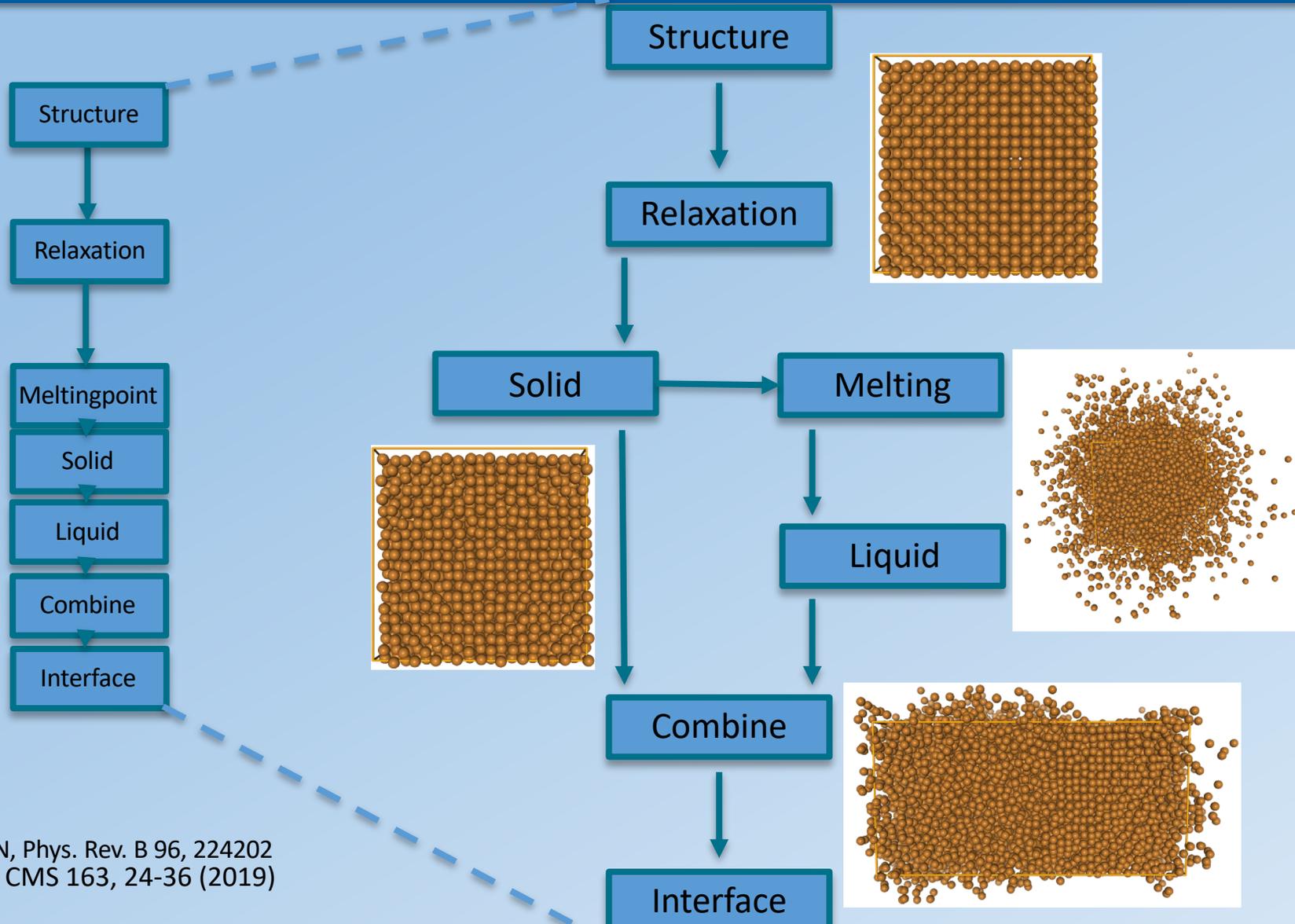
pyiron: Beispiel Schmelzpunkt-Berechnung

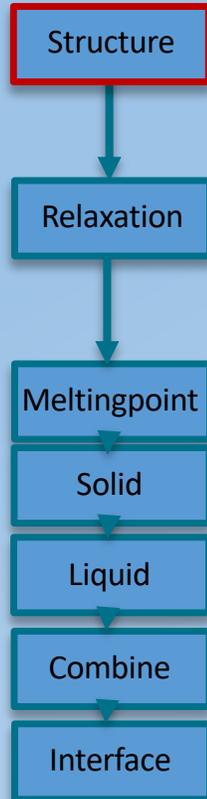


→ Simulationsprotokoll



pyiron: Beispiel Schmelzpunkt-Berechnung





Implement TOR-TILD

1. Import and create new project

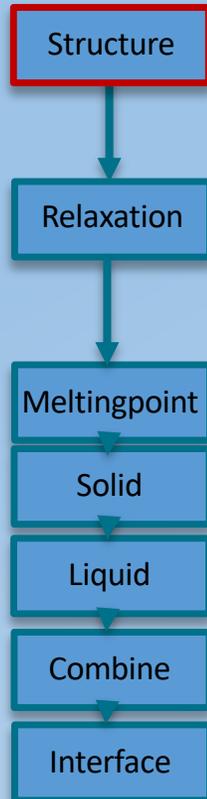
```
import numpy as np
import matplotlib.pyplot as plt
```

```
from pyiron import Project

pr = Project('TOR-TILD')
```

Jupyter Notebook

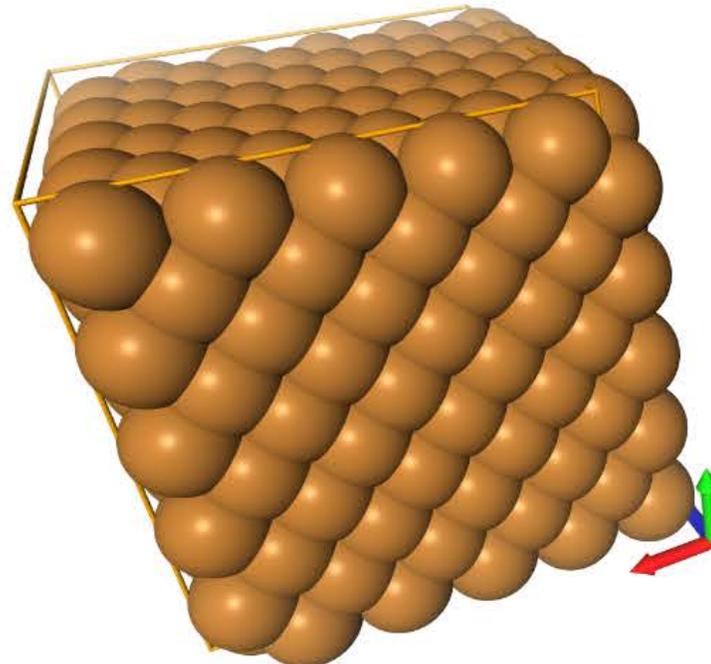
- Web-Interface
- Einfacher Einstieg
- Autocompletion
- Strukturierung mit Überschriften, Kommentaren, etc.



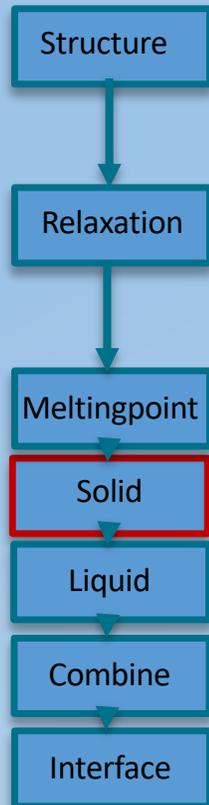
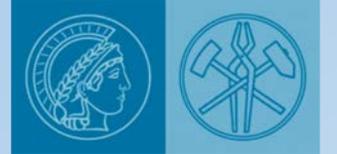
2. Setup structure

```
Cu_bulk = pr.create_ase_bulk('Cu', cubic=True).repeat(5)
```

```
Cu_bulk.plot3d(particle_size=2)
```



- Einbindung externer Bibliotheken



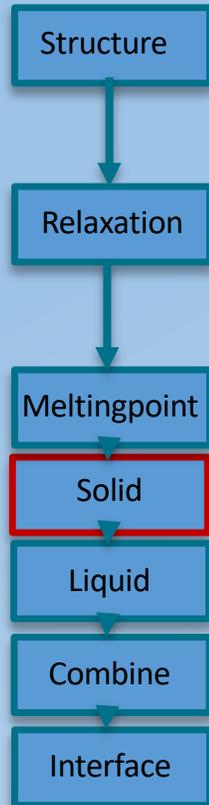
3. NPT Solid

```
job = pr.create_job(pr.job_type.Lammps, job_name='npt')  
job.structure = Cu_bulk  
job.list_potentials()
```

```
['CuAl_lammps_eam',  
'Cu_lammps_eam',  
'Cu_lammps_2_eam',  
'ZrCu_lammps_eam',  
'MgCu_lammps_eam',  
'ZrCuAg_lammps_eam',  
'ZrCuAl_lammps_eam',  
'CuZr_3_eam_fs',  
'Cu_eam_fs',  
'cu_ag_ymwu_eam_alloy',  
'CuAg_eam_alloy',  
'Mendelev_Cu2_2012_eam_fs',  
'Cu01_eam_alloy']
```

➤ Verfügbare Informationen abrufbar

pyiron: Beispiel Schmelzpunkt-Berechnung



```
job.potential = 'CuAl_lammps_eam'  
job.calc_md(temperature=600, pressure=0, n_ionic_steps=1000)  
job.run()
```

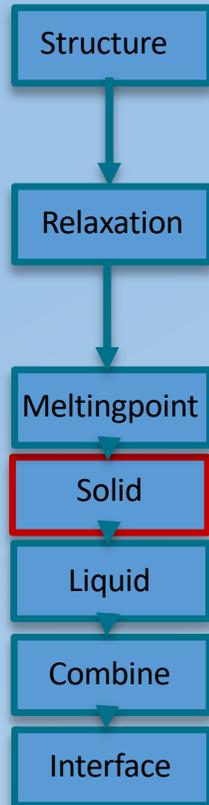
The job npt was saved and received the ID: 2427225

```
job.input.potential
```

	Comment	Parameter	Value
0		pair_style	eam/alloy
1		pair_coeff	** CuAl_lammps.eam Cu Al

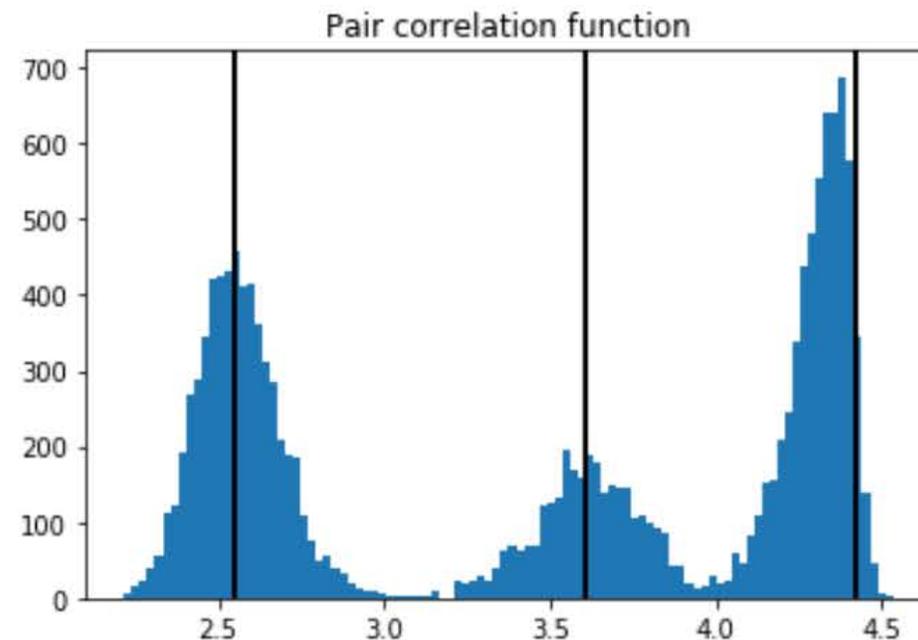
➤ Verfügbare Informationen abrufbar

pyiron: Beispiel Schmelzpunkt-Berechnung



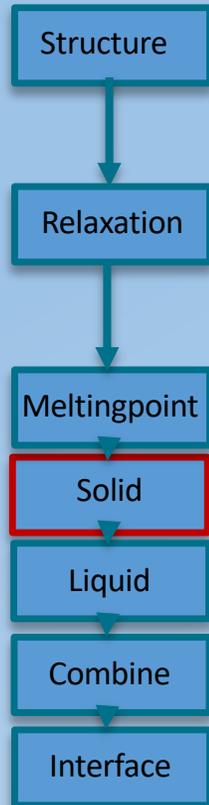
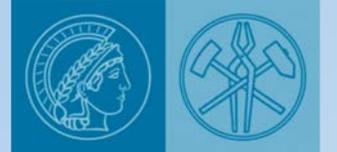
```
initial_struct = job.get_structure(0)
nbrs = initial_struct.get_neighbors(num_neighbors=30)
for d in nbrs.distances[0]:
    plt.axvline(d, color='k')

final_struct = job.get_structure(-1)
nbrs = final_struct.get_neighbors(num_neighbors=30)
plt.hist(nbrs.distances.flatten(), bins=100)
plt.title('Pair correlation function');
```



➤ Schnelle Analyse von Daten

pyiron: Beispiel Schmelzpunkt-Berechnung

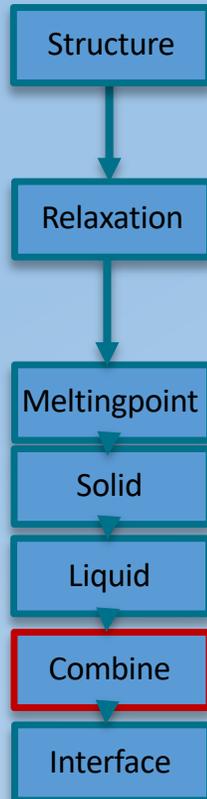


```
job.potential = 'CuAl_lammps_eam'  
job.calc_md(temperature=600, pressure=0, n_ionic_steps=1000)  
job.server.queue_view  
# job.run()
```

	maximum cores	minimum cores	run time limit
impi_hy*	1280	40	259200
impi_hydra	4240	20	259200
impi_hydra_cmfe.*	1280	40	259200
impi_hydra_small	40	1	604800

```
job.potential = 'CuAl_lammps_eam'  
job.calc_md(temperature=600, pressure=0, n_ionic_steps=1000000)  
job.server.core = 40  
job.server.queue = 'impi_hydra_small'  
job.run()
```

- Leichte Übertragung (Upscale) auf HPC



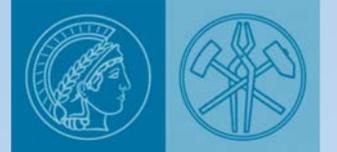
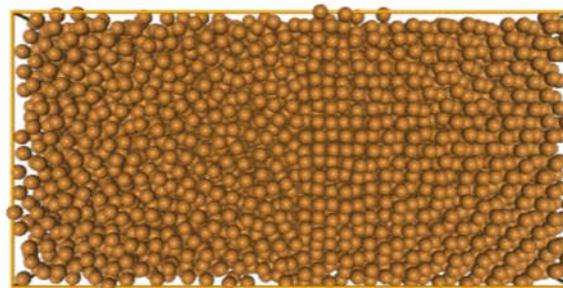
Meltingpoint:
1179.7 +/- 10K

9 Combine

```
[58]: def combine_interface(basis_solid, basis_liquid):
    basis_solid = ham_npt_solid.get_final_structure()
    basis_solid.center_coordinates_in_unit_cell()
    basis_solid.set_absolute()
    basis_liquid = ham_npt_liquid_low.get_final_structure()
    basis_liquid.center_coordinates_in_unit_cell()
    basis_liquid.set_absolute()
    basis_solid_cut = basis_solid.copy()
    x, y, z = basis_solid_cut.pos_xyz()
    atoms_l = basis_solid_cut[x>=3.0]
    atoms_r = basis_solid_cut[x<3.0]
    atoms_r.positions[:,0] += basis_solid.cell[0,0]
    basis_combined = atoms_l + atoms_r
    x_pos, y_pos, z_pos = zip(*basis_combined.positions)
    basis_combined.positions[:,0] -= np.min(x_pos)
    x_pos, y_pos, z_pos = zip(*basis_combined.positions)
    basis_combined.positions[:,0] += (basis_combined.cell[0,0] - np.max(x_pos))/2
    basis_solid_copy = basis_combined.copy()
    basis_liquid_copy = basis_liquid.copy()
    basis_solid_copy.cell[0,0] = basis_solid.cell[0,0] + basis_liquid.cell[0,0]
    basis_liquid_copy.cell[0,0] = basis_solid.cell[0,0] + basis_liquid.cell[0,0]
    basis_liquid_copy.positions[:,0] += basis_solid.cell[0,0]
    return basis_liquid_copy + basis_solid_copy

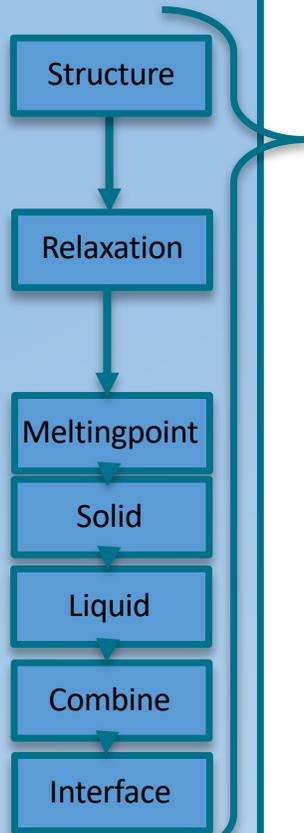
[59]: basis = combine_interface(basis_solid=ham_npt_solid.get_final_structure(), basis_liquid=ham_npt_liquid

[60]: basis.plot3d()
```



➤ Typische Umsetzung von Protokollen

pyiron: Beispiel Schmelzpunkt-Berechnung



Meltingpoint:
1179.7 +/- 10K

```
from pyiron.project import Project

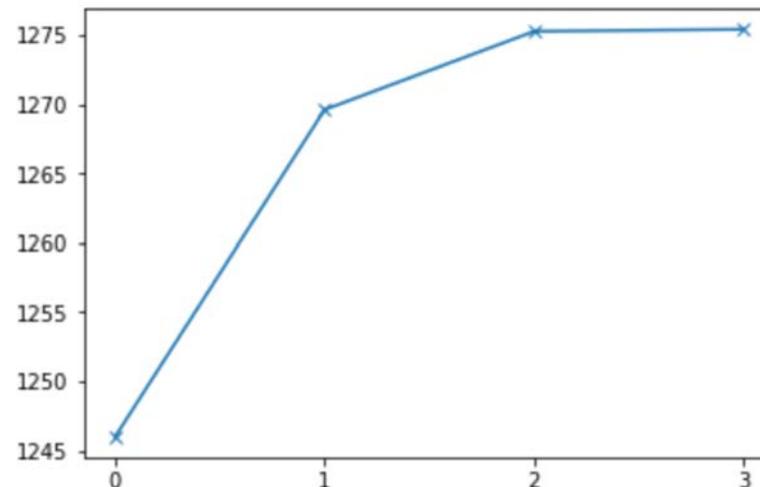
r = Project()

pr.remove_jobs(recursive=True)

for i in range(5):
    job = pr.create_job(pr.job_type.ScriptJob, 'melting_' + str(i))
    job.script_path = 'meltingpoint.ipynb'
    job.server.queue = 'impi_hydra_small'
    job.server.run_time = 200000
    job.server.cores = 10
    job.run()
```

```
The job melting_0 was saved and received the ID: 116338
Queue system id: 1364219
The job melting_1 was saved and received the ID: 116340
Queue system id: 1364220
The job melting_2 was saved and received the ID: 116341
Queue system id: 1364221
The job melting_3 was saved and received the ID: 116342
Queue system id: 1364222
```

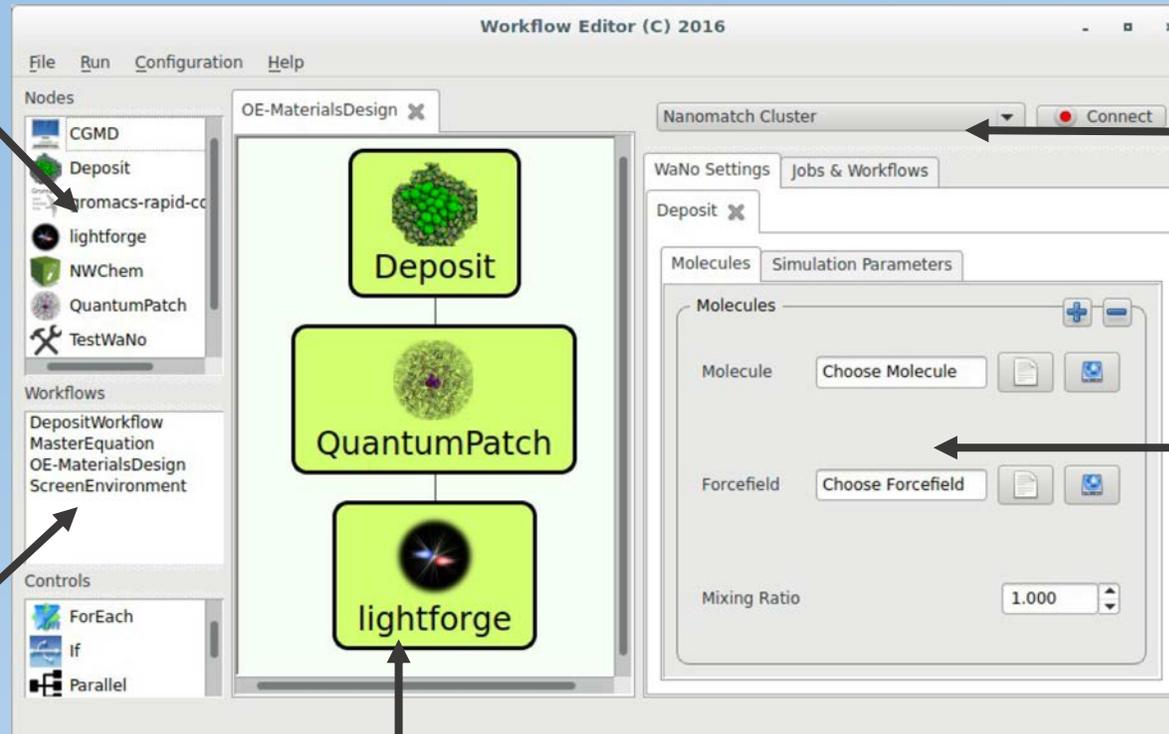
Melting point: 1278 +/- 10K



- Komplette Notebooks wie einzelne Jobs behandeln
- Statistische Analysen

Translation enabled by SimStack

Embedded Scientific modules = „WaNos“



Connect to remote computational resources

Define input files and parameters for each module

Saved workflows for reproducible multiscale simulations

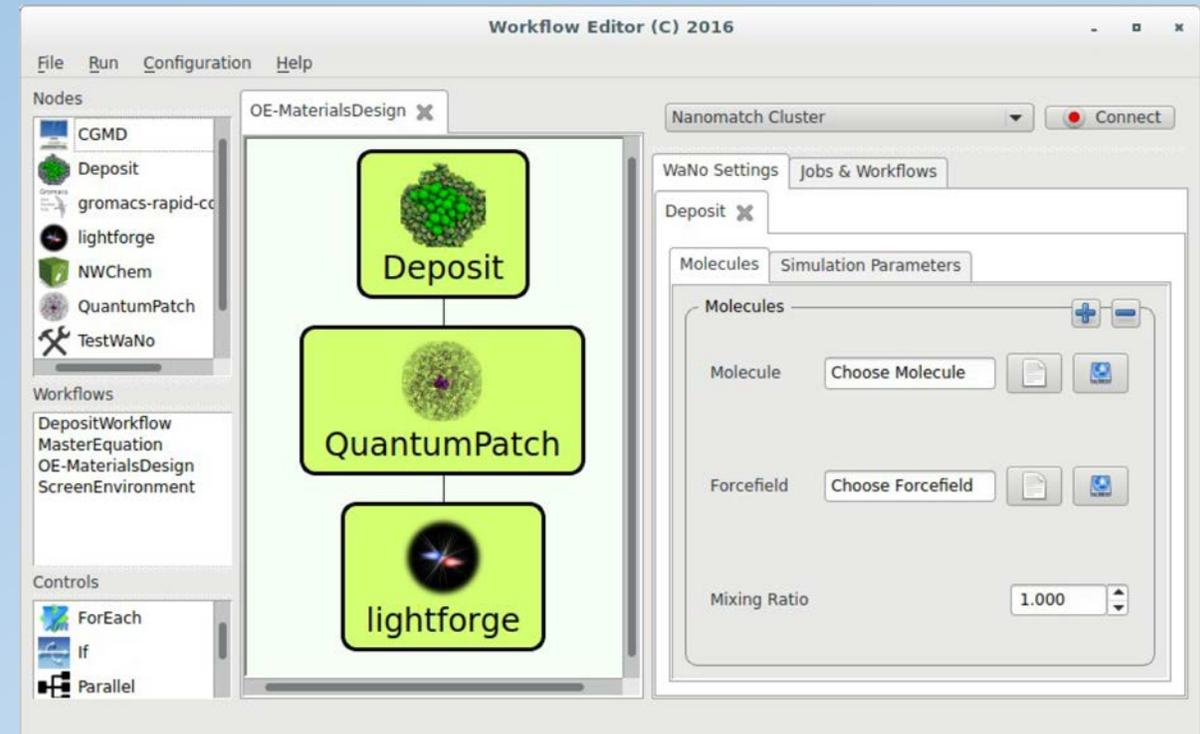
Construct a workflow by drag & drop

Translation enabled by SimStack

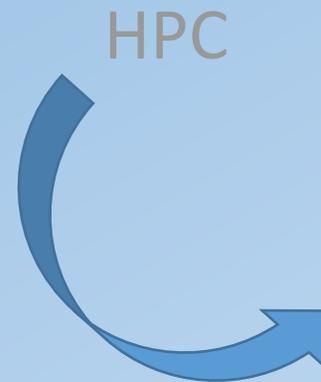
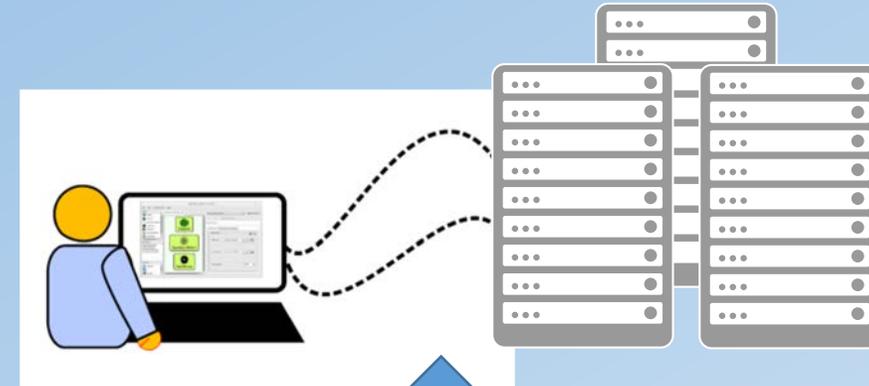
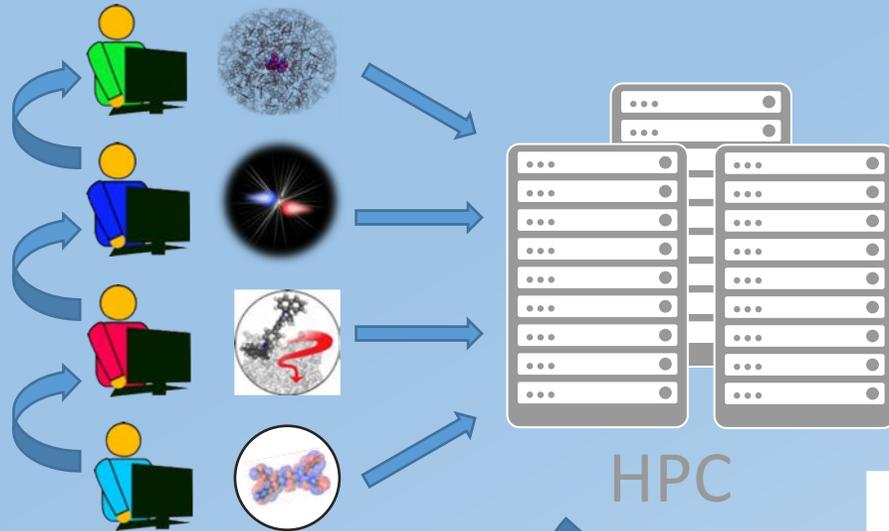


... a generic workflow platform conquering complexity

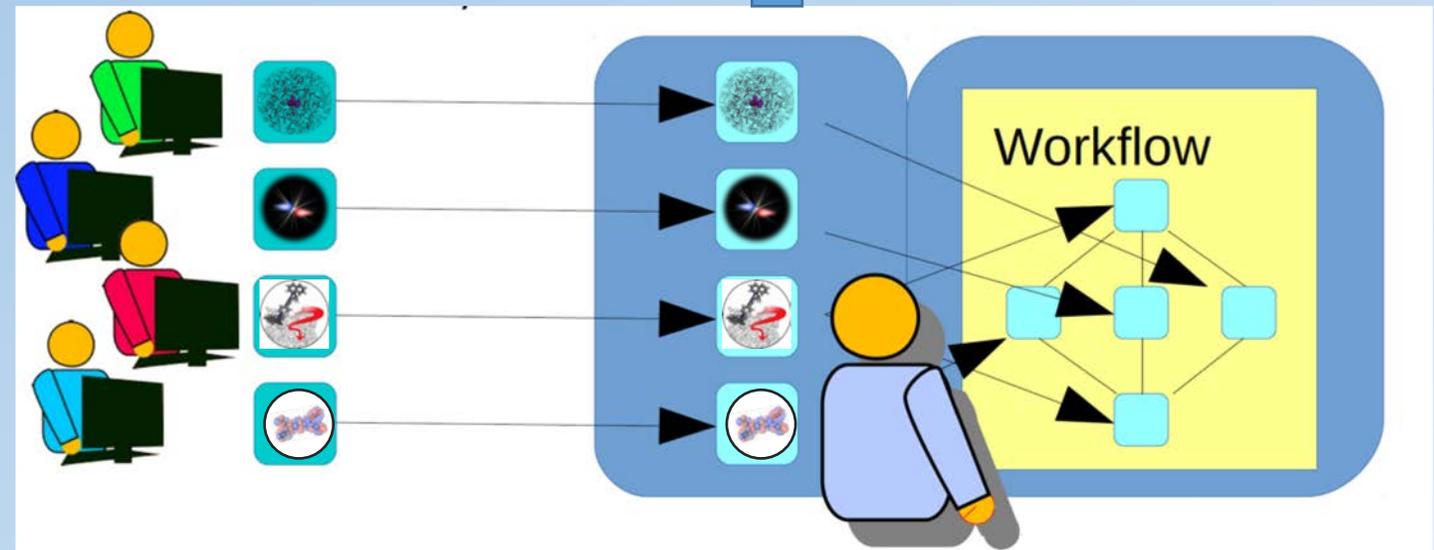
- Open to arbitrary software modules
- Rapid prototyping: 30min to include new modules, 1 h to construct functional workflows
- Maximal reusability and scalability
- Module interoperability:
 - Fully automated, file based data transfer between modules
 - Schema based data transfer in development
 - Compatible with OWL ontologies (e.g. EMMO, once developed)



The multiscale simulation workflow for Organic Electronics



Translation by experts into Reusable workflow elements



Incorporation of Modules (Workflow Active Nodes - WaNos)



```
<WaNoTemplate>
  <WaNoRoot name="Your Wano Name Here">
    </WaNoRoot>
    <WaNoExecCommand>./ExecutionScript.sh</WaNoExecCommand>
    <WaNoInputFiles>
      <WaNoInputFile logical_filename="ExecutionScript.sh">ExecutionScript.sh</WaNoInputFile>
      <WaNoInputFile logical_filename="MandatoryInputFile.dat">MandatoryInputFile</WaNoInputFile>
    </WaNoInputFiles>
    <WaNoOutputFiles>
      <WaNoOutputFile>MandatoryOutputFileName1.dat</WaNoOutputFile>
      <WaNoOutputFile>MandatoryOutputFileName2.dat</WaNoOutputFile>
    </WaNoOutputFiles>
  </WaNoTemplate>
```

Input parameters here

Command for program execution

Mandatory input files

File output expected by other WaNos

Incorporation of Modules (Workflow Active Nodes - WaNos)

```
<WaNoTemplate>
<WaNoRoot name="Random Number Generator">

  <WaNoFloat name="This is a float field">0.0</WaNoFloat>

  <WaNoInt name="This is an int field">1</WaNoInt>

  <WaNoChoice name="Choice of options">
    <Entry id="0" chosen="True">Standard option</Entry>
    <Entry id="1">Second option</Entry>
    <Entry id="2">Third option</Entry>
  </WaNoChoice>

  <WaNoDropDown name="maybe dropdown is better...">
    <Entry id="0" chosen="True">Standard option</Entry>
    <Entry id="1">Second option</Entry>
    <Entry id="2">Third option</Entry>
  </WaNoDropDown>

  <WaNoString name="String">"Writing letters..."</WaNoString>

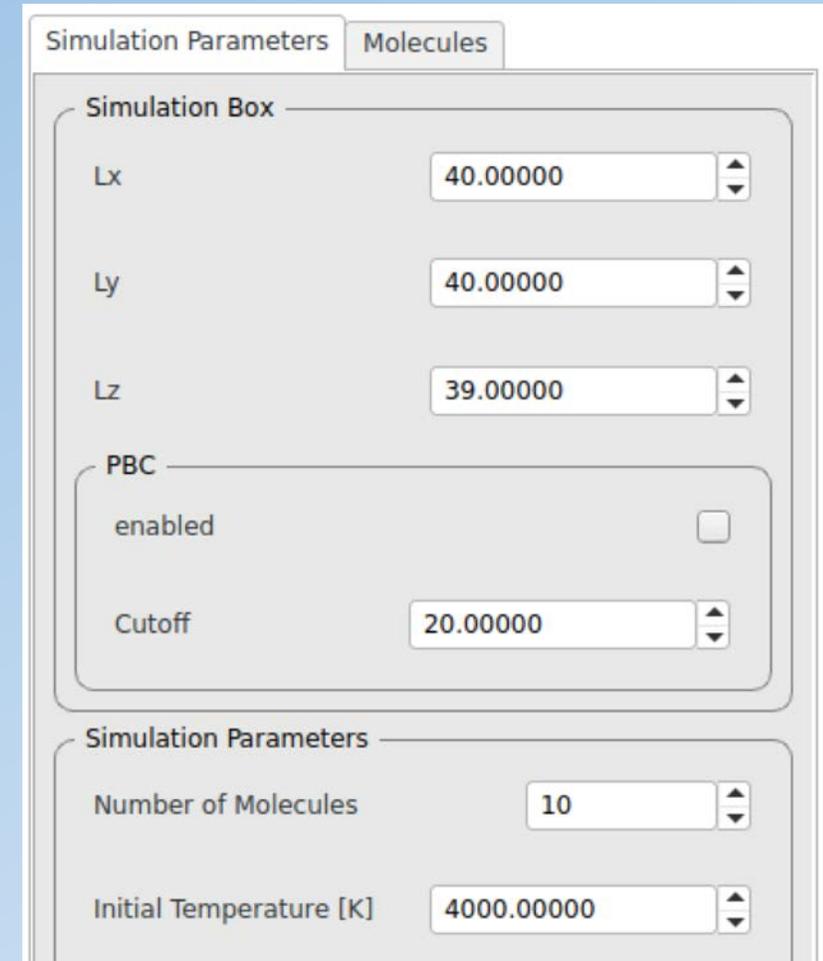
  <WaNoBool name="I like beer">True</WaNoBool>

  <WaNoBox name="Group your parameters in a box">
    <WaNoBool name="Neglect box content">False</WaNoBool>
    <WaNoInt name="Another optional int">3</WaNoInt>
    <WaNoDropDown name="And another drop down">
      <Entry id="0" chosen="True">Standard option</Entry>
      <Entry id="1">Second option</Entry>
      <Entry id="2">Third option</Entry>
    </WaNoDropDown>
  </WaNoBox>
</WaNoTemplate>
```



Deposit

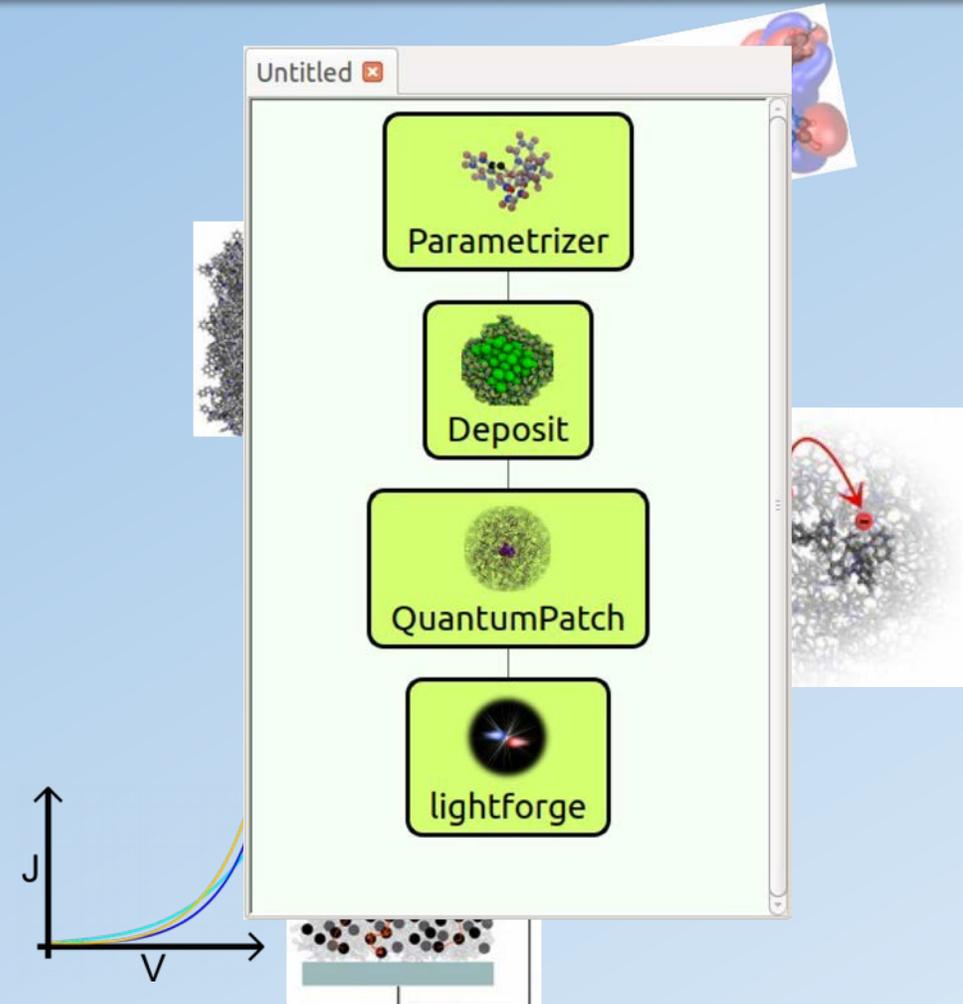
- Deposit is a Monte-Carlo tool to generate organic thin-film morphologies
 - Complex input language (roughly 80 parameters up front)
 - Minimum number of input files: 2
 - Minimum number of parameters, which you have to actually know: 7

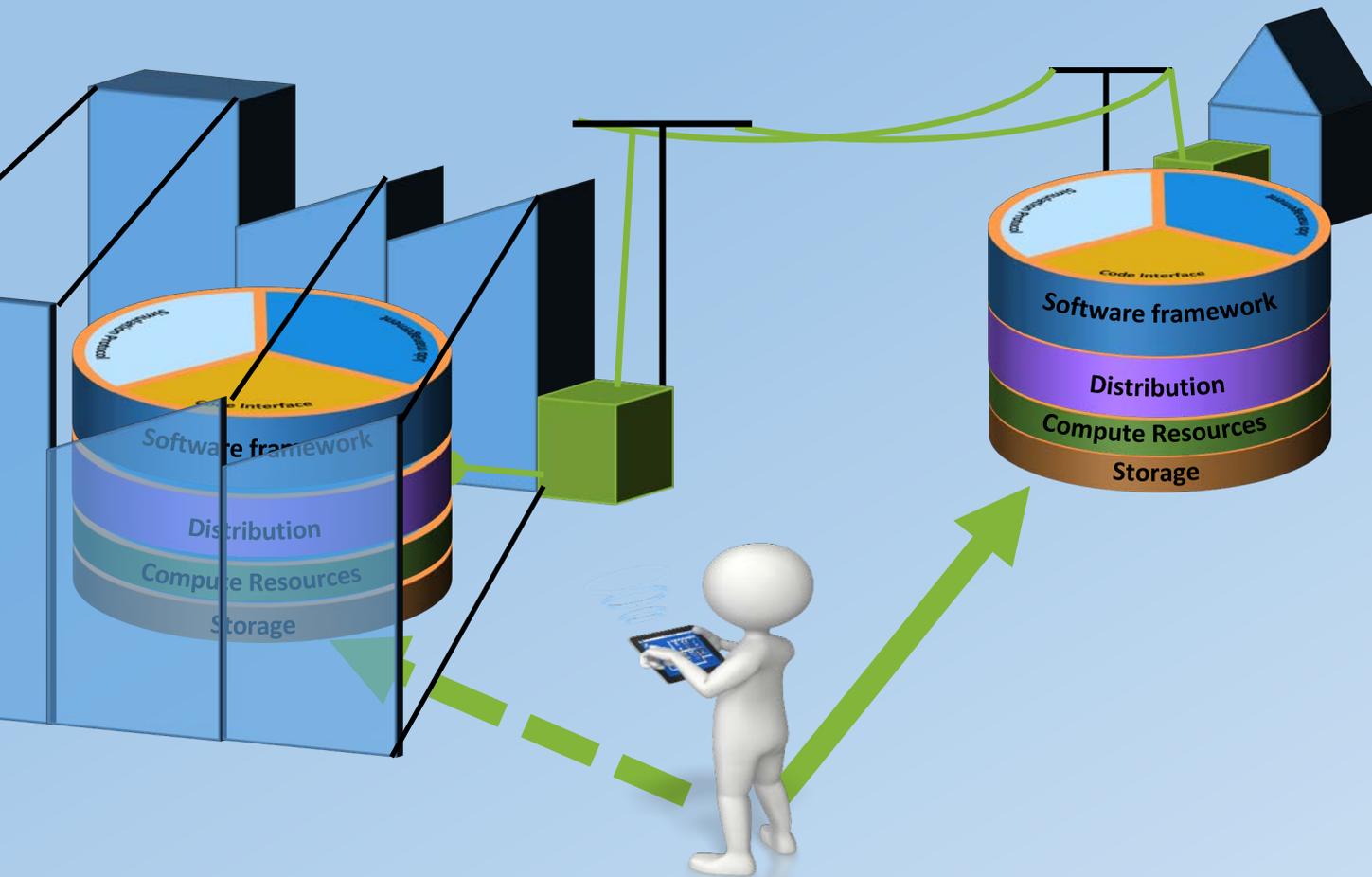


The screenshot shows the SimStack interface with two tabs: "Simulation Parameters" and "Molecules". The "Simulation Parameters" tab is active and displays the following settings:

- Simulation Box:**
 - Lx: 40.00000
 - Ly: 40.00000
 - Lz: 39.00000
- PBC:**
 - enabled:
 - Cutoff: 20.00000
- Simulation Parameters:**
 - Number of Molecules: 10
 - Initial Temperature [K]: 4000.00000

- 1. Single molecule parametrization (QM)
 - Geometry optimization
 - Customized force-fields
- 2. Generation of atomistic morphologies
 - Molecules parametrized on quantum mechanical level
 - Simulation of physical vapor deposition
- 3. Calculation of charge hopping rates
 - Full quantum mechanical electronic structure analysis
 - Electronic couplings, reorganization and orbital energies
- 4. Charge transport simulations
 - Time resolved charge carrier/exciton dynamics
 - IVs, IQEs, carrier balance, quenching, ...





- **Ziel:** Gemeinsame Entwicklung einer Plattform für den strukturierten, dezentralen Umgang mit Daten und Workflows, die für die Industrie geeignet ist
- **Anknüpfung:** Tools können in die Plattform eingepflegt werden, wenn sie ein
 - Python interface haben,
 - als Conda-Paket vorliegen und
 - einen strukturierten Input/Output erzeugen
- **Software-Umgebungen:** Derzeit werden pyiron und SimStack angeboten
- Die Plattform unterstützt die Anknüpfung an **Ontologien** → *Vortrag Niebel*