Karlsruher Institut für Technologie

15th HPC Café

07.05.2026

Agenda HPC Café – 07.05.2026

Topics for today:

- **Domain-specific language for materials modeling and simulation (Dr. Ivan Kondov, SCC)**
- Discussion and open floor

As usual this format is meant to be interactive.
Don’t hesitate to ask questions!

Karlsruher Institut für Technologie

Domain-specific language for materials modeling and simulation

Ivan Kondov, SCC

Karlsruher Institut für Technologie

Time for Discussion

Next HPC Café

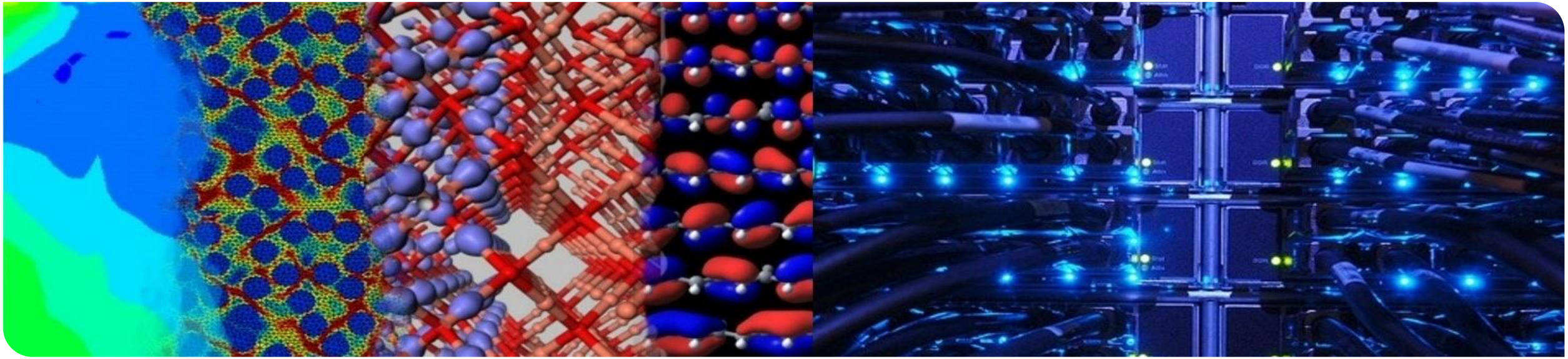
- Next HPC Café planned for **9 June 2026, 10 am**
  - Note: different timeslot due to the public holiday Corpus Christi
- Reminder: regular timeslot for the HPC Café
  - Every **first Thursday of the month 10:00 am**

- As always, open to hear ideas for topics and talks, from …
  - Yourself
  - Colleagues and collaborators
  - Also just expressions for “topics of interest”

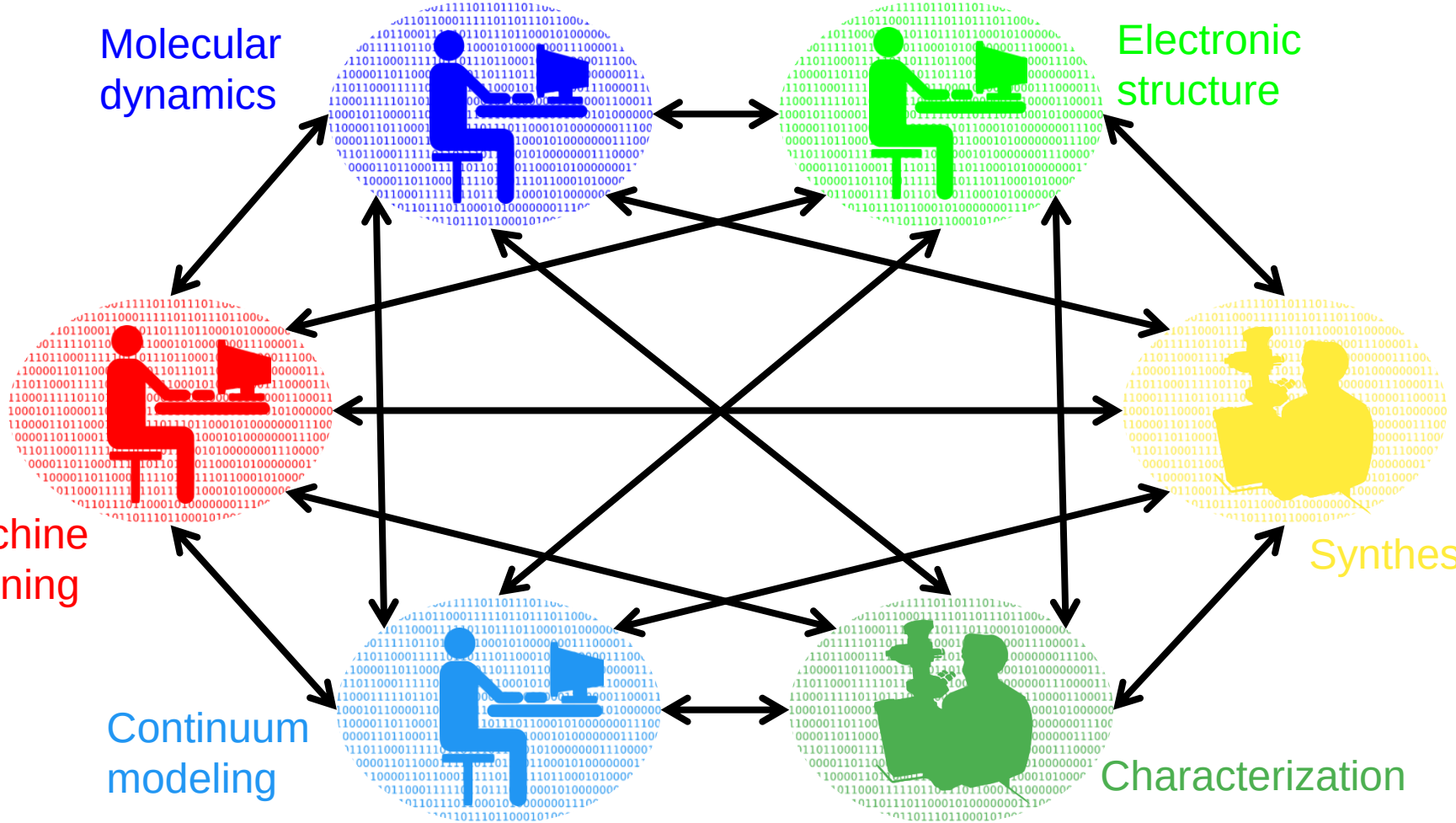
Karlsruhe Institute of Technology

Domain-specific language for materials modeling

Ivan Kondov



Materials research: traditional approach



Domain-specific code/data

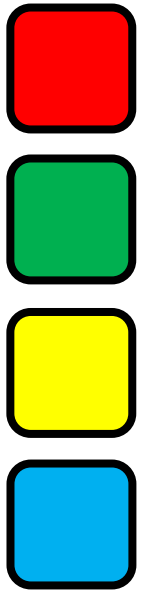
Domain expert

Lacking:

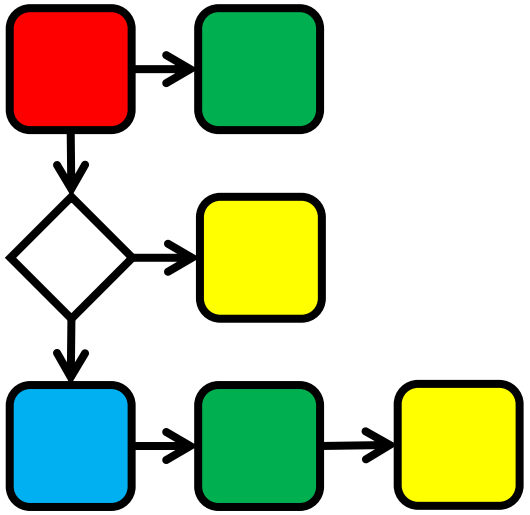
- Reusability
- Reproducibility
- Scalability
- Productivity

Current approach: scientific workflows

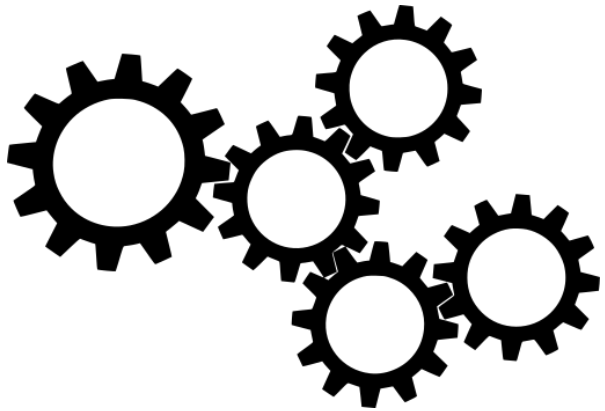
Good



Better



Best



Best? Really?
Open issues:

- Usability
- Productivity
- Reusability
- Willingness to use

Adapters

Workflows

Workflow management systems (WMS)

Outline

- Lessons learned
- Our approach: Domain Specific Language
- Domain specific languages textS and textM
- Use case:
  Virtual screening of OER/ORR catalysts
- Current, recent developments
- Outlook

Lessons learned

Usability	• Scientific workflows concept too complex • Too complex Python APIs and CLI tools • Database queries - yet another domain
Productivity	• Time-consuming “boilerplate” code • Tedious work to vary parameters • Clumsy API, CLI and GUI
Reusability	• Too specific Python functions • Lack of data typing • Simple operations not in the workflow

Partial solutions

- wfGenes: CLI and GUI
  https://wfgenes.gitlab.io/wfgenes

  M. Roozmeh, I. Kondov, In “Workflows in support of large-scale science (WORKS)”, IEEE; pp 9–16 (2020).
  M. Roozmeh, I. Kondov, LNCS 13353, 69 (2022).

- VRE Middleware: API and GUI
  https://vre-middleware.readthedocs.io

The full solution

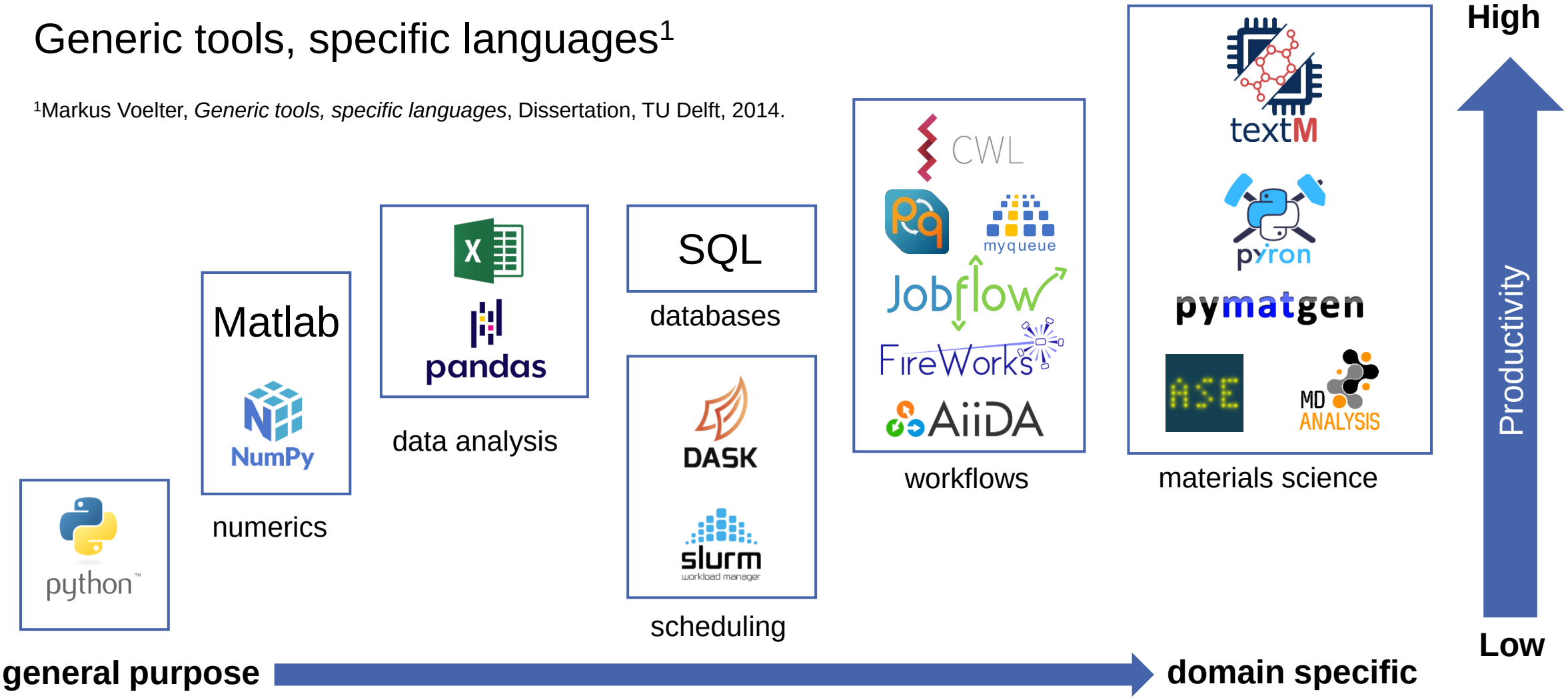
- Domain-specific language
  https://vre-language.readthedocs.io

  I. Kondov, R. Cortés Mejía, M. Müller, N. Pfisterer, S. Sreenivasan, LNCS 15909, 221 (2025).
  https://doi.org/10.1007/978-3-031-97564-6_18

How to improve (re)usability and productivity?

Generic tools, specific languages[1]

[1]Markus Voelter, *Generic tools, specific languages*, Dissertation, TU Delft, 2014.



What is a domain-specific language (DSL)?

- Domain-specific concepts and abstractions
- More expressive in its domain

Examples

GPLs	DSLs
• C/C++ • Fortran • Python • Java • Julia • …	• Matlab • Excel • SQL • Regex • PostScript • … • textS, textM

- Python

```
len_1 = 2.0
units_1 = 'meter'
len_2 = 1.0
units_2 = 'centimeter'
diff_1 = len_1 - len_2  # no error but wrong
diff_2 = 2 - len_1  # no error but wrong
assert units_1 == units_2  # python(!) error
```

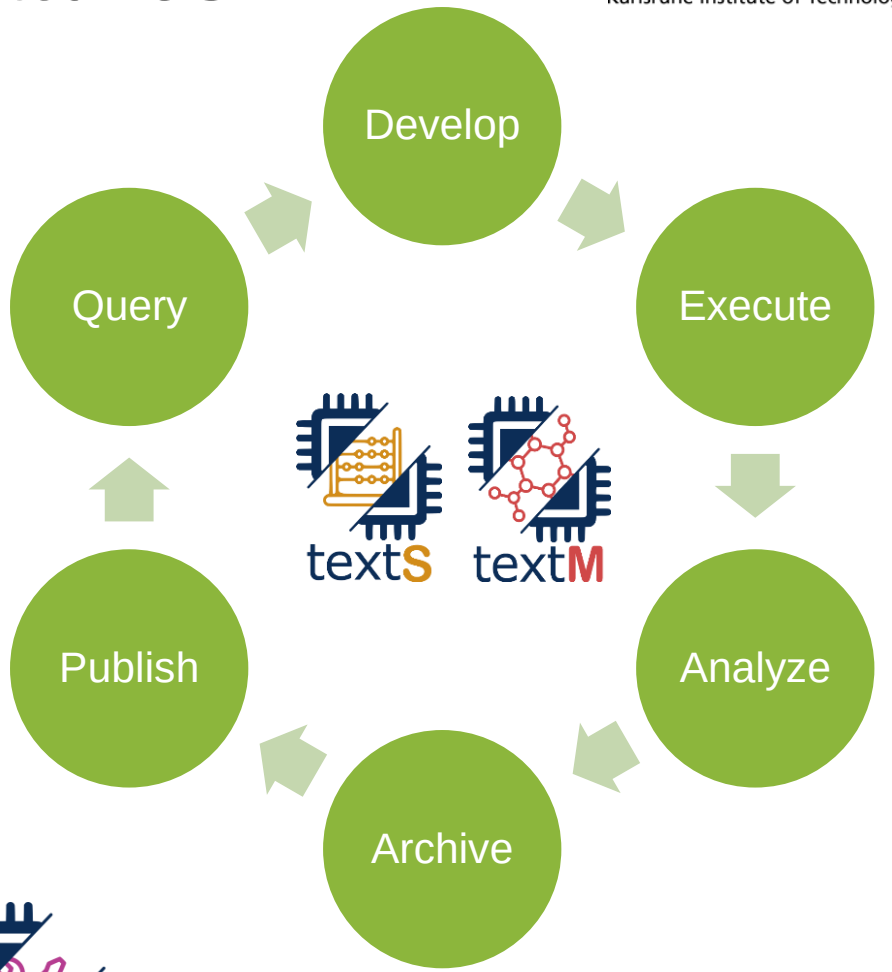
- Better: a custom Python class or the **pint** package

- Even better: native units support. In textS:

```
len_1 = 2.0 [m]
len_2 = 1.0 [cm]
diff_1 = len_1 - len_2  # unit conversion
diff_2 = 2 + len_1  # domain-specific error
```

textS/textM: Basic concepts and features

- Cover full life cycle
- Transparent database, batch system and data staging
- Frontends: console, Jupyter
- Supporting tools: parser and editor from grammar
- Reuse domain-specific libs: ASE, Pint, pandas, NumPy …



FireWorks mongoDB NumPy pandas

HoreKa jupyterhub KIT | NHR National High-Performance Computing Center

I. Kondov, R. Cortés Mejía, M. Müller, N. Pfisterer, S. Sreenivasan, LNCS 15909, 221 (2025). https://doi.org/10.1007/978-3-031-97564-6_18

Scientific Computing Language: textS

Karlsruhe Institute of Technology

Key concepts
- Declarative
- Pure functions
- Expressions
- Static typing

Basic types
- Integer
- Float
- Complex
- Boolean
- String

Data structures
- Tuple
- Series
- Table
- Array

Numerics
- Physical units
- Physical constants
- Optional uncertainties
- Library of functions

Evaluation
- Instant, batch
- On demand
- Caching
- Lazy evaluation
- Duplicates detection
- Checkpointing and restart
- Resource annotations

Reusability
- Bulk processing
- Sub-model reuse
- Grammar versioning
- Schema versioning

Datastore
- Storage: GridFS and file system
- Formats: JSON, YAML, HDF5, custom
- Migration, replicas, registry
- Bitstream preservation
- Persistent identifiers

Jupyter kernel
- Magics
- Syntax highlighting
- Autocompletion
- Interactive plots

textS

https://vre-language.readthedocs.io

Atomic and molecular modeling language: textM

Types

- Property
- Structure (~ASE atoms)
- Calculator (~ASE Calculator)
- Constraint (~ASE Constraint)
- Algorithm (~ASE)
- Trajectory (~ASE)
- Species (chemistry)
- Reaction (chemistry)

Calculators

- EMT
- VASP
- TURBOMOLE
- …

> 20 Algorithms

- Structure builders and build tools
- Local minimum optimization
- Molecular dynamics
- Density of states
- Band structure
- Equation of state
- Vibrations
- …

Constraints

- FixedLine
- FixedPlane
- FixedAtoms
- …

https://vre-language.readthedocs.io

How to start

- Read the quick-start guide
- Run some examples in the examples folder
- Run the ~1500 regression tests
- Read the comprehensive documentation

https://vre-language.readthedocs.io

Support

virtmat-tools@lists.kit.edu

Use case: Oxygen reduction reaction

M, MO_x, MO_xH_y systems

Thermodynamic overpotential:

$$\eta_{ORR} = U_0 - U_{\max} \qquad U_0 = 1.23\ \text{V (RHE)}$$

Critical potential:

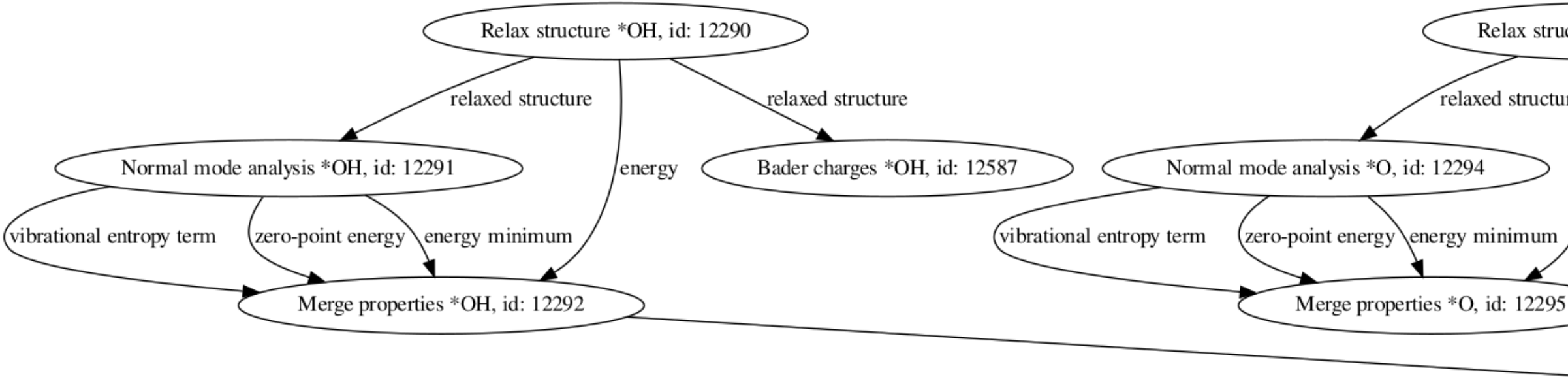
$$U_{\max} = \sup\{U:\ \Delta G^{(i)}(U) \geq 0\ \forall i \in [1,2,3,4]\}$$

Free energies:

$$\Delta G^{(i)}(U) \approx \Delta E^{(i)} + \Delta ZPE^{(i)} - T\Delta S^{(i)} - z_i U$$

J. K. Nørskov et al., J. Phys. Chem. B 108, 17886 (2004).

Workflow implementation



M. Salmanion, I. Kondov, M. Vandichel, P. Aleshkevych, M. M. Najafpour, Inorg. Chem. **61**, 2292 (2022).
N. Akbari, I. Kondov, M. Vandichel, P. Aleshkevych, M. M. Najafpour, Inorg. Chem. **60**, 5682 (2021).
M. Vandichel, K. Laasonen, I. Kondov, Top. Catal. **63**, 833 (2020).
P. Faubert, I. Kondov, D. Qazzazie, O. Yurchenko, C. Müller, MRS Communications 8, 160 (2018).
I. Kondov, P. Faubert, C. Müller, Electrochimica Acta 236, 260 (2017).

All data and workflows available in the Materials Cloud https://archive.materialscloud.org/.

TextM: Electronic structure properties

```
geom_surface_O = Structure from file 'geometry_files/pt_slab-o.cif'  # only the *O species shown
view structure (geom_surface_O)

c_surf_O = constr(9, 28)
fix_surf_O = FixedAtoms where c_surf_O
c_nrm_O = constr(27, 28)
nrm_surf_O = FixedAtoms where c_nrm_O
constr(n, m) = map((x: x < n), range(0, m, 1))

calc_rlx = Calculator vasp >= 5.4.4 ((algo: 'Fast'), (ediff: 1e-06) [eV], (ediffg: -0.01) [eV/angstrom], (encut: 400.0) [eV],
                                     (ibrion: 2), (icharg: 2), (ismear: 0), (ispin: 2), (istart: 0), (kpts: [5, 5, 1]),
                                     (sigma: 0.1) [eV], (xc: 'PBE')),
                                    task: local minimum
calc_nrm = Calculator vasp >= 5.4.4 ((algo: 'Fast'), (ediff: 1e-07) [eV], (encut: 400.0) [eV], (ibrion: 5), (icharg: 2),
                                     (ismear: 0), (ispin: 2), (nfree: 2), (nsw: 1), (potim: 0.005) [angstrom],
                                     (istart: 0), (kpts: [5, 5, 1]), (sigma: 0.1) [eV], (xc: 'PBE')),
                                    task: normal modes

props_O = Property energy ((structure: geom_surface_O), (calculator: calc_rlx), (constraints: (fix_surf_O))
          on 76 cores for 4.0 [hours]
nrm_O = Property vibrational_energies, energy_minimum ((structure: props_O.output_structure), (calculator: calc_nrm),
                                                       (constraints: (nrm_surf_O)) on 76 cores for 4.0 [hours]
view structure (props_O.output_structure)
```

I. Kondov, R. Cortés Mejía, M. Müller, N. Pfisterer, S. Sreenivasan, LNCS 15909, 221 (2025). https://doi.org/10.1007/978-3-031-97564-6_18

TextM: Properties of the chemical species

```
# use exp, log from numpy
from virtmat.functions use exp, log
zpe_f(vib_ener) = 0.5 * sum(vib_ener)
vib_entr_f(vib_energies) = kB * sum(map(entr, vib_energies, map(exp_hob, vib_energies)))
beta = 1.0 / (kB * temperature)
exp_hob(ene) = exp(-ene * beta)
entr(ene, eh) = ene * beta * eh / (1.0 - eh) - log(1.0 - eh)
temperature = 298.15 [K]
kB = 1 [boltzmann_constant]

MO = Species MO (
     (energy: props_O.energy[0]),
     (zpe: zpe_f(nrm_O.vibrational_energies[0])),
     (entropy: vib_entr_f(nrm_O.vibrational_energies[0])),
     (temperature: 298.15) [K]
)

print(MO)
```

I. Kondov, R. Cortés Mejía, M. Müller, N. Pfisterer, S. Sreenivasan, LNCS 15909, 221 (2025). https://doi.org/10.1007/978-3-031-97564-6_18

TextM: Electrochemical properties

```
orr1 = Reaction M + O2 + 0.5 H2 = MOOH
orr2 = Reaction MOOH + 0.5 H2 = MO + H2O
orr3 = Reaction MO + 0.5 H2 = MOH
orr4 = Reaction MOH + 0.5 H2 = M + H2O
orr = (reactions: orr1, orr2, orr3, orr4)

orr_free_energies = map((r: r.free_energy[0]), orr)
U_max = min(map((g: -g / (n * e)), orr_free_energies))
orr_overpotential = U_0 - U_max
min(s) = reduce((x, y: if(x < y, x, y)), s)
U_0 = 1.23 [V] # standard reversible potential
n = 1  # number of exchanged electrons, all ORR/OER elementary reactions are one-electron
e = 1 [elementary_charge]
print('U_max:', U_max, 'ORR_overpotential:', orr_overpotential)
```

FireWorks: 1256 lines of code (Python + YAML)
TextM: 244 lines of code

Complete use-case code: https://gitlab.kit.edu/kit/virtmat-tools/oxycat-use-case

I. Kondov, R. Cortés Mejía, M. Müller, N. Pfisterer, S. Sreenivasan, LNCS 15909, 221 (2025). https://doi.org/10.1007/978-3-031-97564-6_18

Recent and current developments

- Projects:
  - JL VMD: VirtMat 2 (2021-2023), VirtMat 3 (2024-2026)
  - Helmholtz IVF Science Serve: VM-FIRE, with FZJ Tier-1/Tier-0 (2026)
- Deployment
  - Two computing projects (7+3 Mio core-hours) on HoreKa (Tier-2 system)
- VRE Language nominated for Helmholtz Software Award 2026

Outlook

- Deployment
  - Further Tier-2 systems via NHR
  - On Tier-1 and Tier-0 systems: FZJ partner in VM-FIRE
- Features
  - Uncertainties in series and arrays
  - Uncertainty propagation: Quasi Monte Carlo, Polynomial Chaos Expansion
  - FAIR compliance: persistent identifiers
  - New calculators: Elk, LAMMPS, CP2K
- Usability
  - Datastore: bitstream preservation, replicas and registry
- Sustainability, maintainability
  - Generalization of Algorithm and Calculator into textS
  - Package modularization
  - Facilitate textS and textM extensions via plugins
  - Add further DSLs

Thank you!

Team (current and former):

Allen Benny
Sijia Huang
Rodrigo Cortés Mejía
Marvin Müller
Nikolai Pfisterer
Thomas Schraivogel
Sruthy Sreenivasan

Collaborations:

Claas Müller (U Freiburg)
Mohammad Mahdi Najafpour (IASBS, Zanjan)
Kushal Ramakrishna (HZDR, Dresden)
Godehard Sutmann (FZJ, Jülich)
Matthias Vandichel (U Limerick)
Wolfgang Wenzel (KIT)

Legal notice

This presentation may be used by the participants of HPC Café only and may not be further distributed.