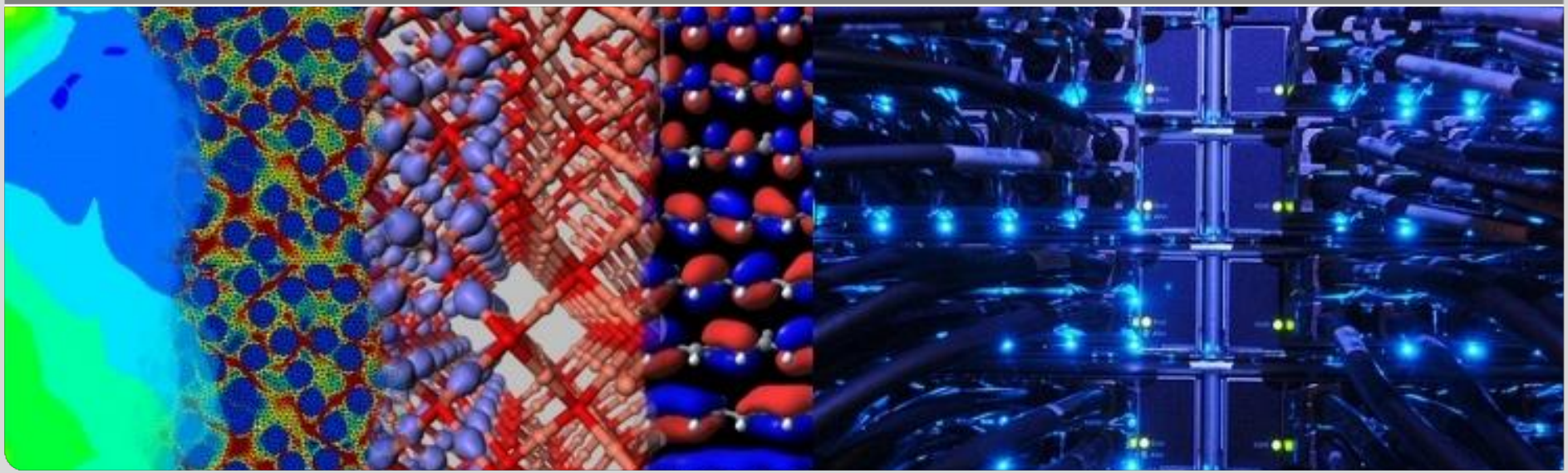
Karlsruhe Institute of Technology

College on Multiscale Computational Modeling of Materials for Energy Applications: Tutorial

Ivan Kondov

STEINBUCH CENTRE FOR COMPUTING - SCC



Acknowledgements

Thanks to all college organizers!

The first tutorial has been partially funded by the 7th Framework Programme of the European Commission within the Research Infrastructures with grant agreement number RI-261594, project MMM@HPC.

Thanks to all partners in the project who have contributed to this tutorial!

Outline

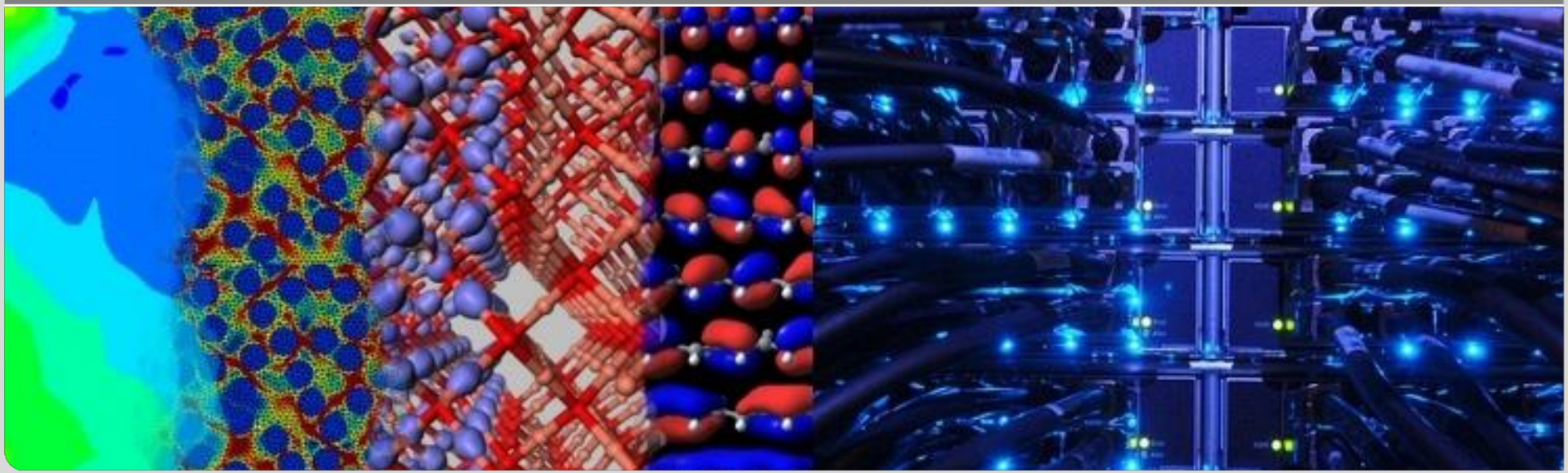
- Tutorial 1: Using MMM@HPC
  - Exercise 1: Morphology in polymer electronics
  - Exercise 2: Morphology and donor-acceptor pairs in small-molecule semiconductor
- Tutorial 2: Using Fireworks and ASE
  - Introduction
  - Exercise 1: SingleTask
  - Exercise 2: ForeachTask
  - Exercise 3: Charge transfer in dimers: sequential workflow
  - Exercise 4: Charge transport in disordered structures
    - Sequential workflow
    - Parallel workflow
  - Exercise 5: Adding data analysis: Extending an existing workflow

Karlsruhe Institute of Technology

Tutorial 1
Using the MMM@HPC Framework

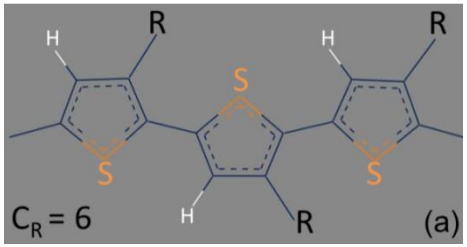
Ivan Kondov

STEINBUCH CENTRE FOR COMPUTING - SCC



Charge transport in polymer-based devices

Task: Link between morphology protocols and charge transport protocols



***P3HT**: organic semiconducting polymer*

MOPHOLOGY

1 Generation of fully equilibrated atomistic structures and dynamics of model materials (Rr-P3HT, PQT)

amorphous

crystalline

CHARGE TRANSPORT

2 Computation of microscopic quantities, simulation of charge transport properties according to the physical model

hopping

band-like

Outcome: Charge transport rates inside crystalline and amorphous domains, in the interfacial area, and charge carrier mobilities through the entire material.

Polymer: Morphology workflow

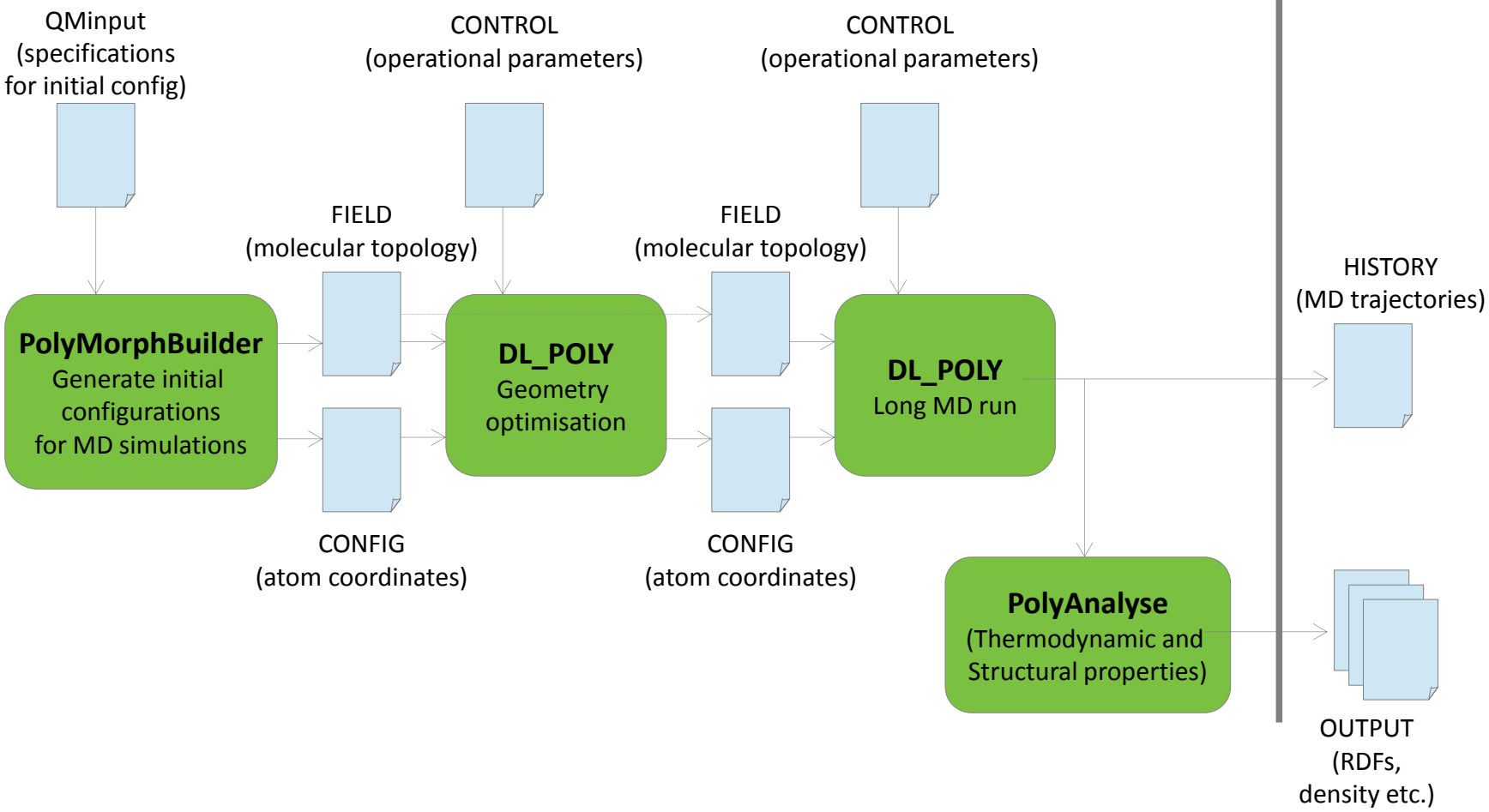
1. PolyMorphBuilder (building the initial morphology)
   - Initially combined with MAPS2 software platform for creating connectivity and assigning the corresponding force field.
   - Produces the input files for the MD simulation packages LAMMPS and DL_POLY and also for Monte Carlo home-made FORTRAN code.

2. DL_POLY (or LAMMPS)
   - Pre-equilibrated (geometry optimized)
   - Long all-atom Molecular Dynamics (and Monte Carlo) simulations using DL_POLY (or LAMMPS)

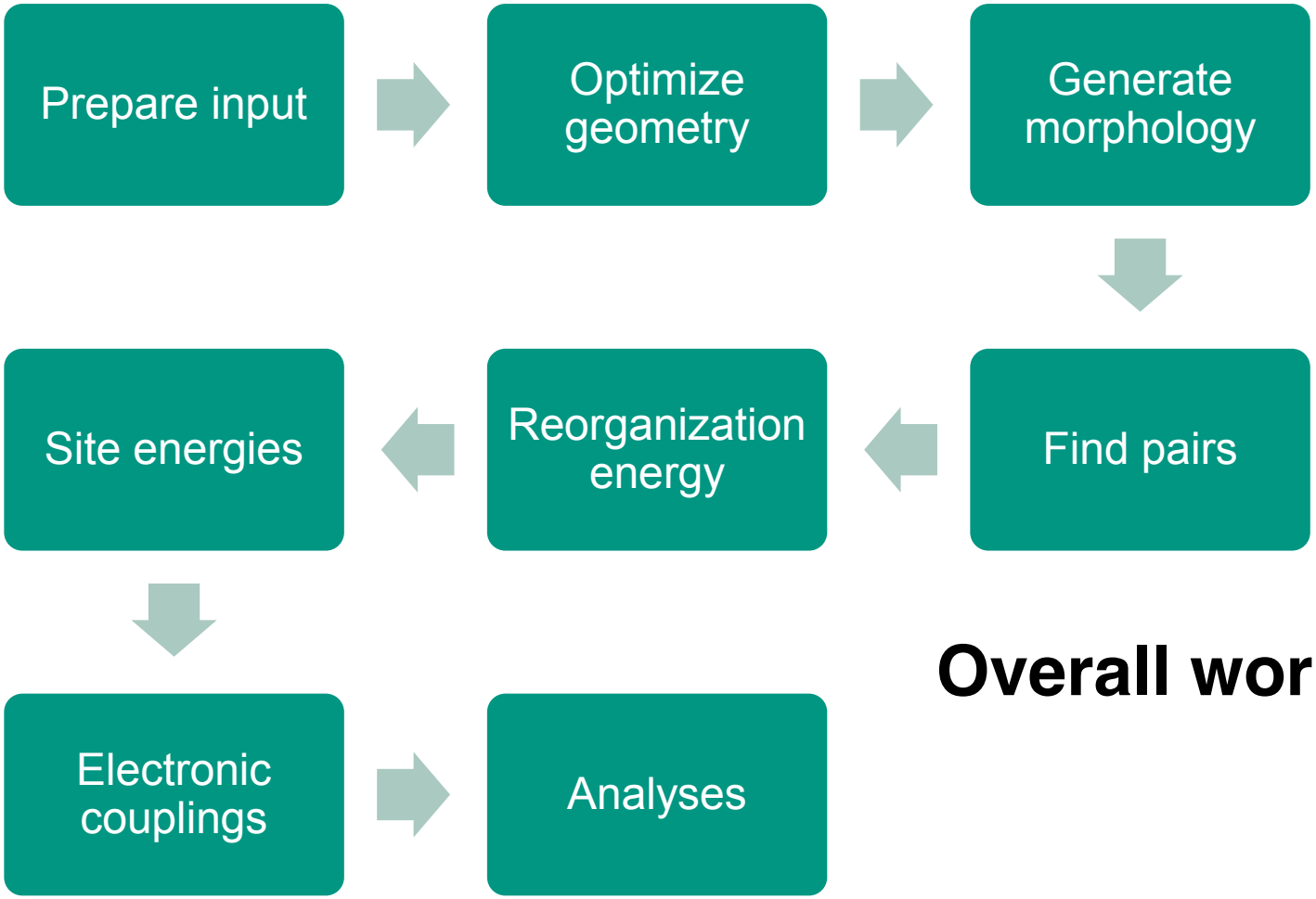
3. PolyAnalyse (home-made FORTRAN code)
   - Trajectories from step 2 are used for analysis to predict structural properties of interest.

Polymer: Morphology workflow

Part A – Morphology generation: data interchange

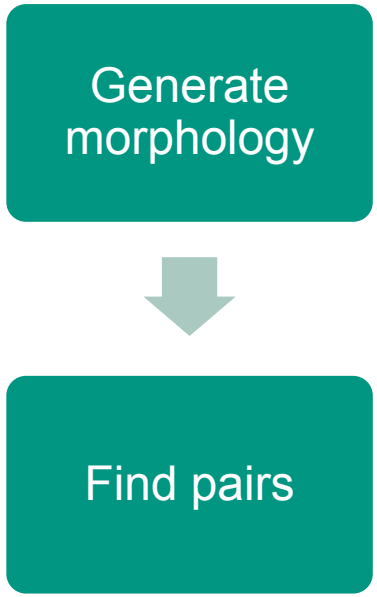


Exercise 2: Organic electronics workflow



Overall workflow

Exercise 2: Organic electronics workflow



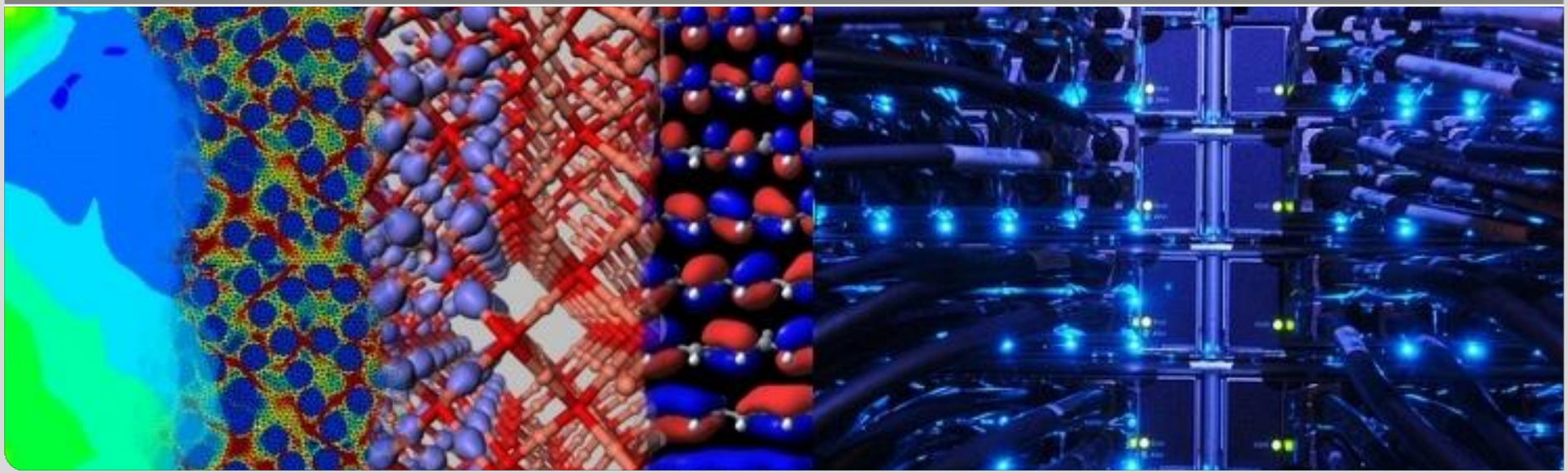
On the Live CD

Karlsruhe Institute of Technology

Tutorial 2
Using FireWorks and ASE

Ivan Kondov

STEINBUCH CENTRE FOR COMPUTING - SCC



FireWorks general

- Define, manage and execute workflows
- Open source – a modified BSD license
- Website https://pythonhosted.org/FireWorks
- Used in the Materials Project http://www.materialsproject.org/

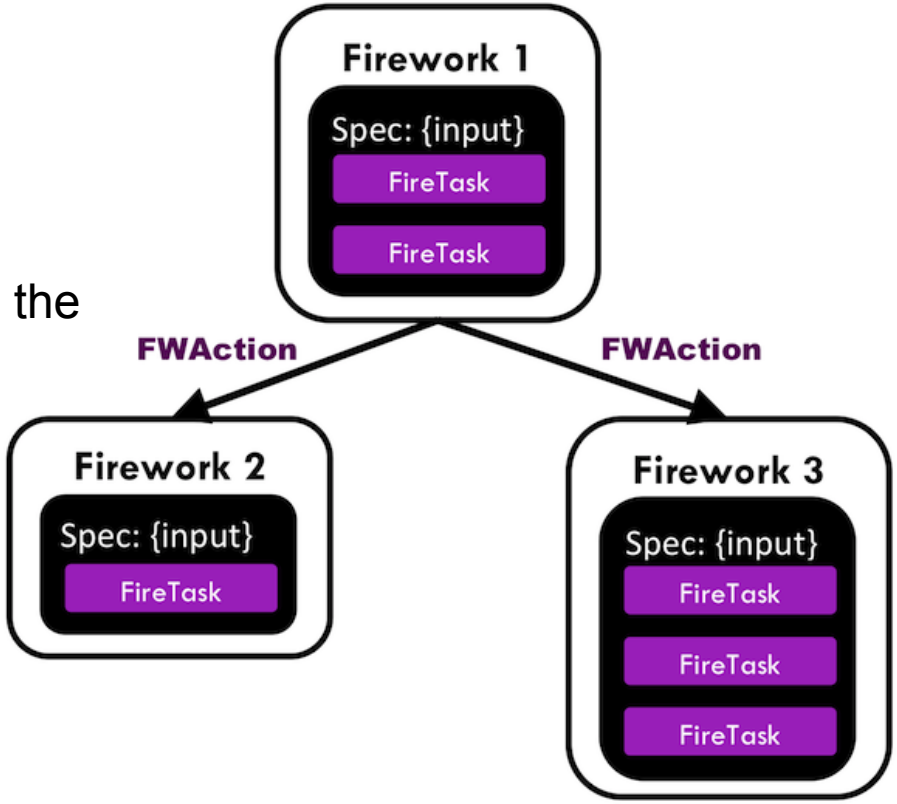
A. Jain et al., Concurrency and Computation: Practice and Experience 27, 5037 (2015)

FireWorks basics

- FireTask: an *atomic* computing job: one code, one script, one function

- FireWork:
  - Contains one or more FireTasks executed in a sequence
  - FireTasks in one FireWork share the same working directory
  - Includes bootstrap information: **spec**

- Workflow
  - A Directed Acyclic Graph (DAG) of FireWorks
  - **FWAction**



Source: https://pythonhosted.org/FireWorks

FireTasks

Karlsruhe Institute of Technology

Built-in FireTasks

- ScriptTask
- FileWriteTask, FileDeleteTask, FileTransferTask, CompressDirTask, ArchiveDirTask
- TemplateWriterTask
- PyTask

Custom FireTasks

- SingleTask
- ForeachTask
- JoinDictTask
- JoinListTask
- … write your own FireTasks!

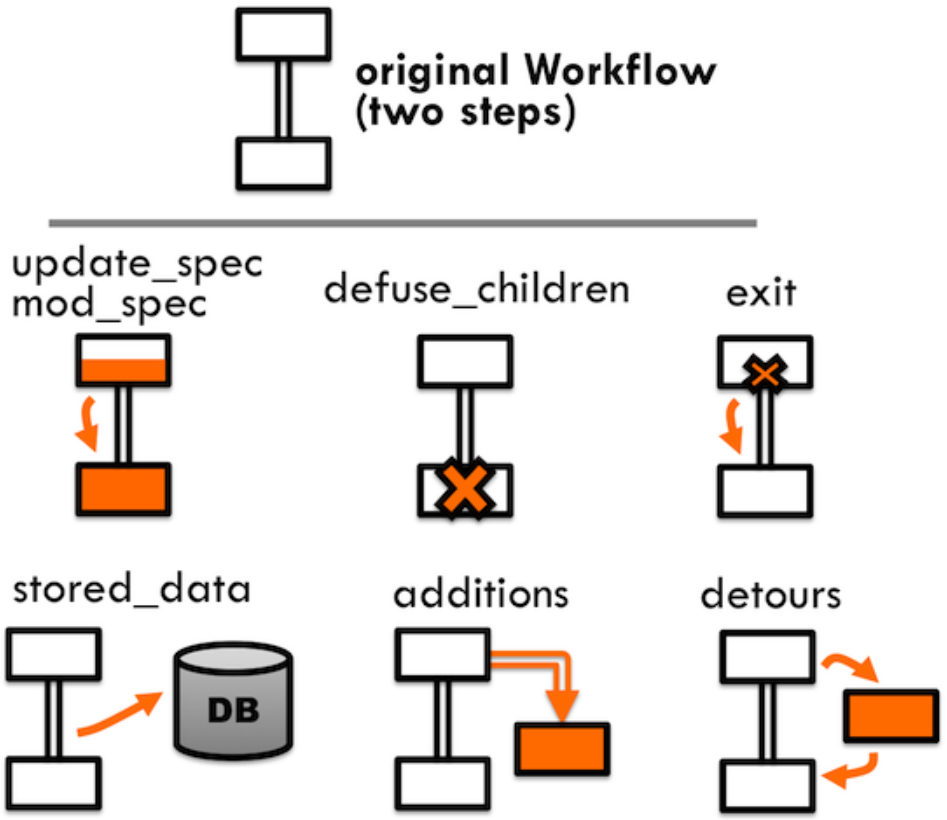
FWAction object

Returned by

- FireTasks
- Python functions called by PyTask

Purpose:

- Manage dataflow
- Dynamically change workflow



Source: https://pythonhosted.org/FireWorks

Using FireWorks (basic)

Create

- No workflow language
- No workflow editor available
- Write YAML or JSON manually
- Use Python scripts

Add

- LaunchPad: storage for all FireWorks objects
- Using command line:
  `lpad add`
- Use Python scripts

Launch

- FireWorker: computing resource
- Rocket launcher:
  `rlaunch`
- Multiple rockets launcher:
  `mlaunch`
- Execution via a batch system:
  `qlaunch`
- Use Python scripts

Query

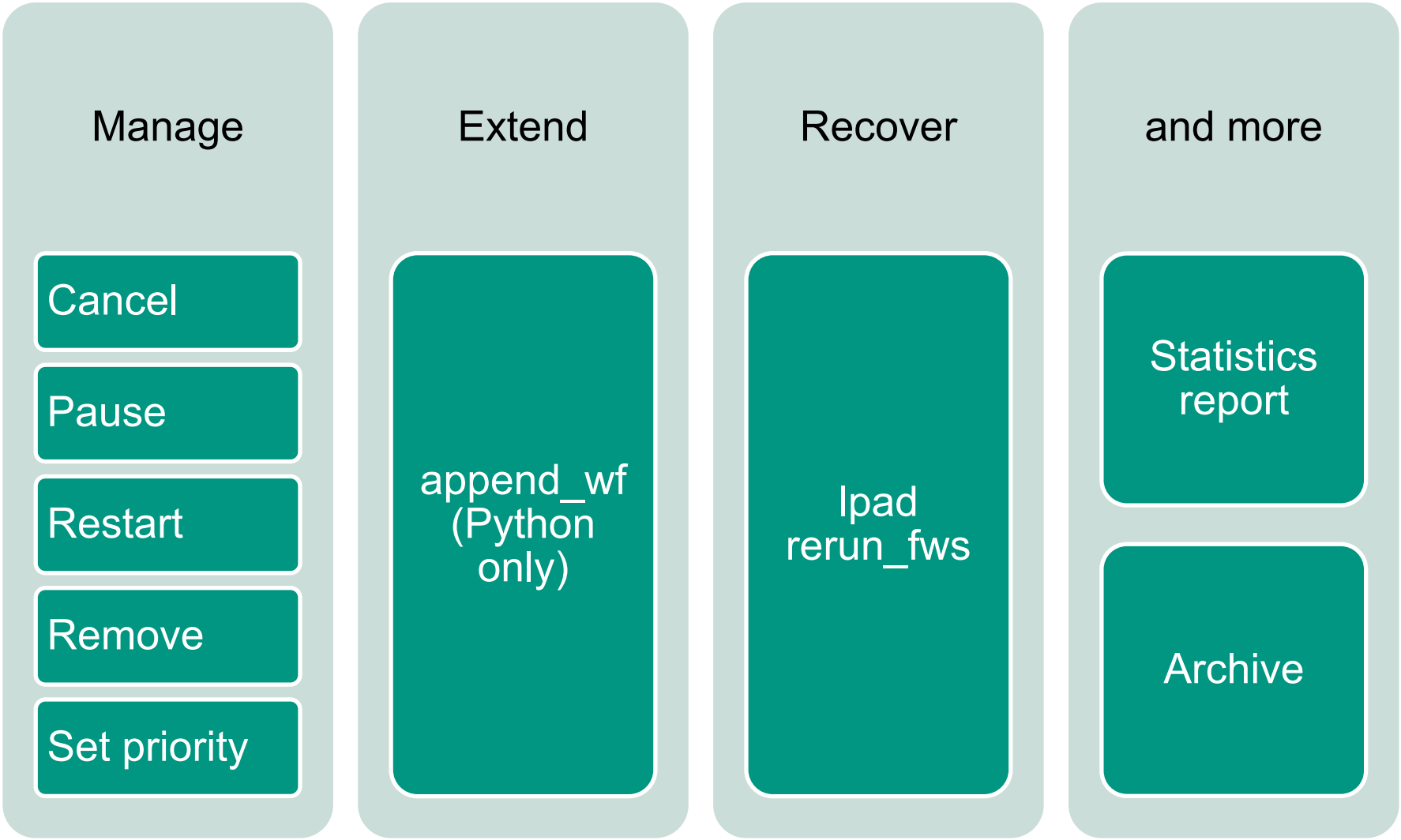
- Command line:

  ```
  lpad
  get_fws
  ```

  ```
  lpad
  get_wflows
  ```

- Use tracker
- Use web GUI
  `lpad webgui`
- Use Python scripts

Using FireWorks (advanced)



Using FireWorks productively

- Use FireWorks with batch systems
- Configure security
- Tune performance
- Consult the FireWorks documentation

Atomic Simulation Environment (ASE)

- Provides code integration infrastructure with calculator objects
- NWChem / Turbomole calculators in this tutorial
- Atoms class used for atomic structure data layout
- Dictionaries and lists (JSON serializable) used for all other data
- No data model
- Packages (in directory ictp-tutorial-fireworks/lib)
  - nwchem.py
    - basic wrapper, pre- and post-processor methods
  - transport.py
    - transport core methods
  - fw_task_functions_seq.py
    - reusable python functions needed for sequential workflows (one data entity)
  - fw_task_functions_par.py
    - reusable python functions needed for parallel workflows (multiple data entities)

wiki.fysik.dtu.dk/ase

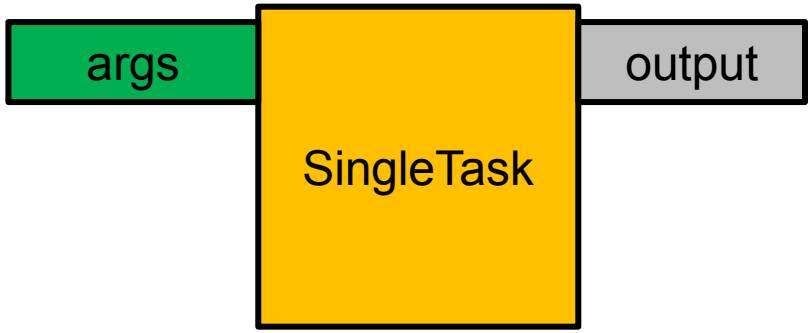
S. R. Bahn and K. W. Jacobsen, Comput. Sci. Eng. 4, 56 (2002)

SingleTask

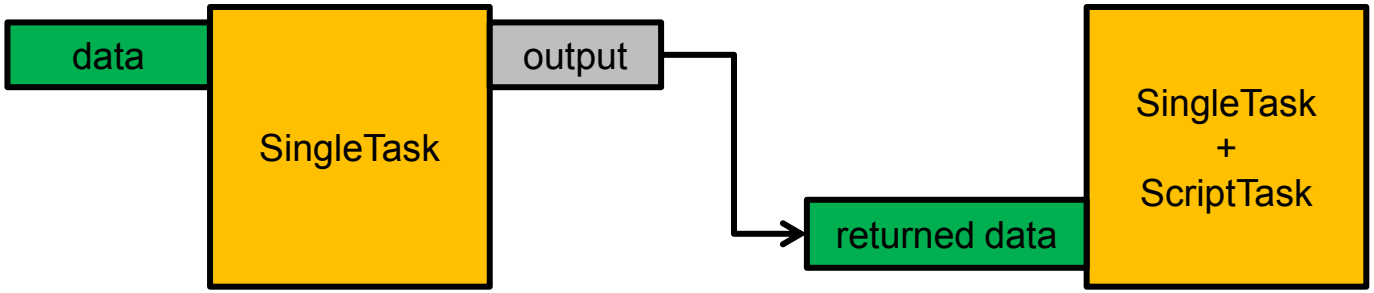
- Purpose: passes a list of data entities as positional arguments to a Python function and forwards the returned objects to the next FireWork via FWAction.

- Mandatory arguments:
  - ‘function’: the name of the Python function

- Optional arguments (must be existing spec keys):
  - ‘args’: a string or list of names of data fields in the spec
  - ‘output’: a string or a list of names of fields for output



Exercise 1: SingleTask



Python:

```
cd exercise_1
cp ../demo/singletask_demo.py .
python singletask_demo.py

lpad reset
lpad add workflow.yaml
lpad get_fws -d all
rlaunch singleshot
rlaunch -s singleshot
rlaunch -s rapidfire
```

JSON/YAML:

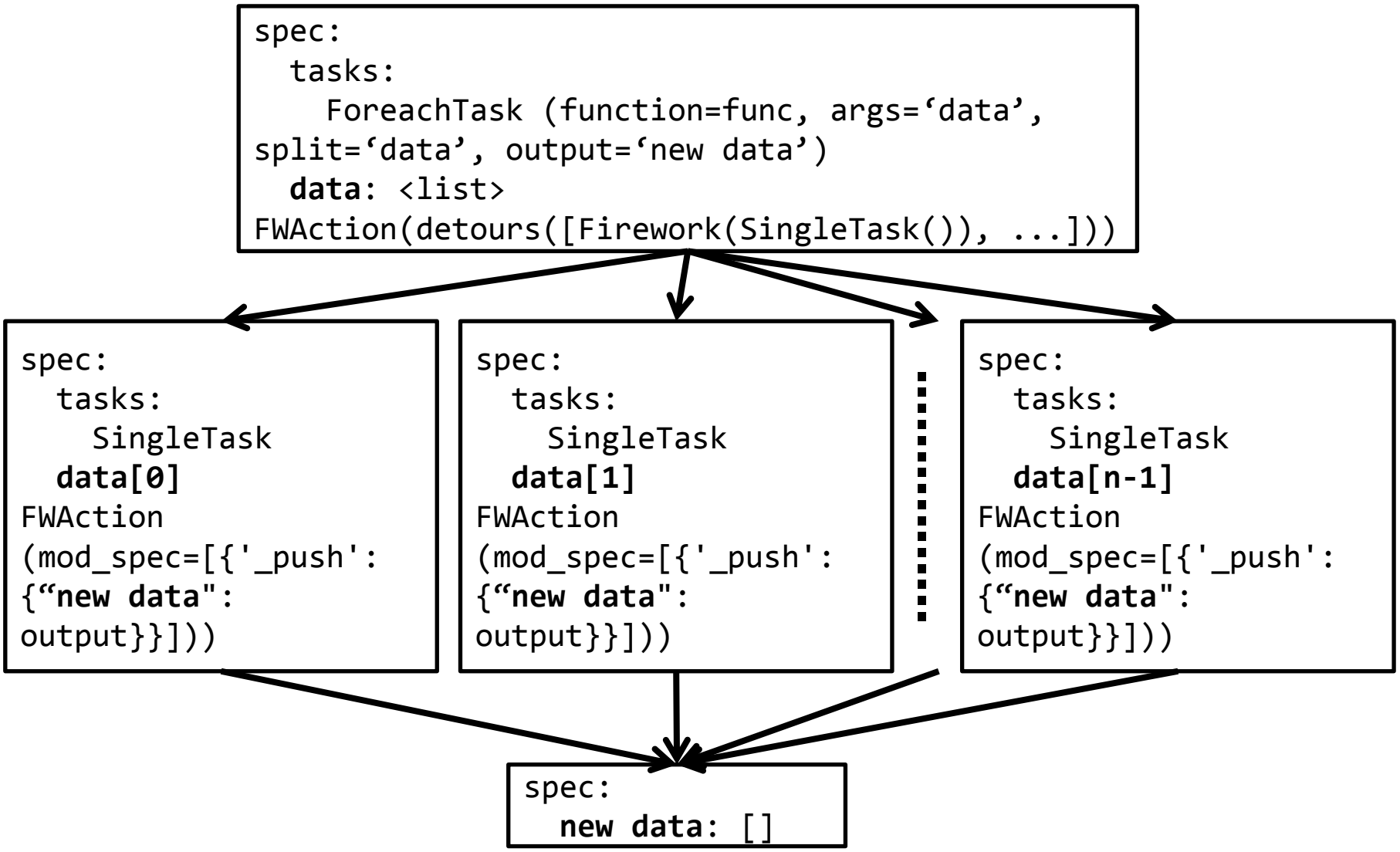
```
cd exercise_1
cp ../demo/singletask_demo.yaml .
cp ../demo/singletask_demo.json .
gedit singletask_demo.[json|yaml]

lpad reset
lpad add workflow.[json|yaml]
lpad get_fws –d all
rlaunch singleshot
rlaunch -s singleshot
rlaunch -s rapidfire
```

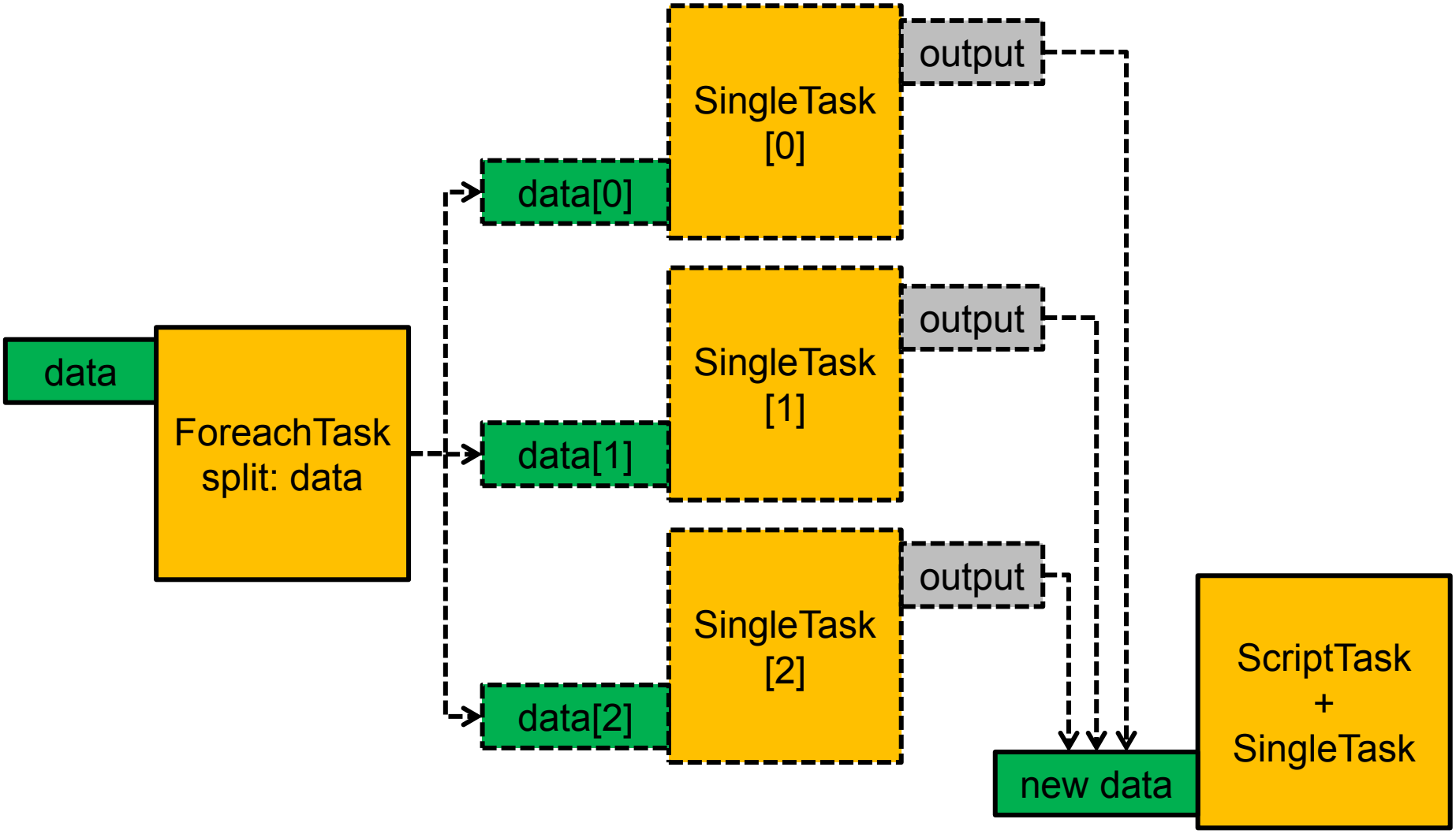
ForeachTask

- Purpose: realize a fork-join workflow model passing data through the spec
- Mandatory arguments:
  - ‘function’ is the name of the function
  - ‘split’ must be name of a list
  - ‘args’ must contain at least the ‘split’ argument
- Optional arguments:
  - ‘output’
- Arguments ‘args’ and ‘split’ must refer to existing spec keys

ForeachTask



ForeachTask



Exercise 2

Python:

```
python .py

lpad reset
lpad add workflow.yaml
lpad get_fws -d all
rlaunch singleshot
rlaunch -s singleshot
rlaunch -s rapidfire
```

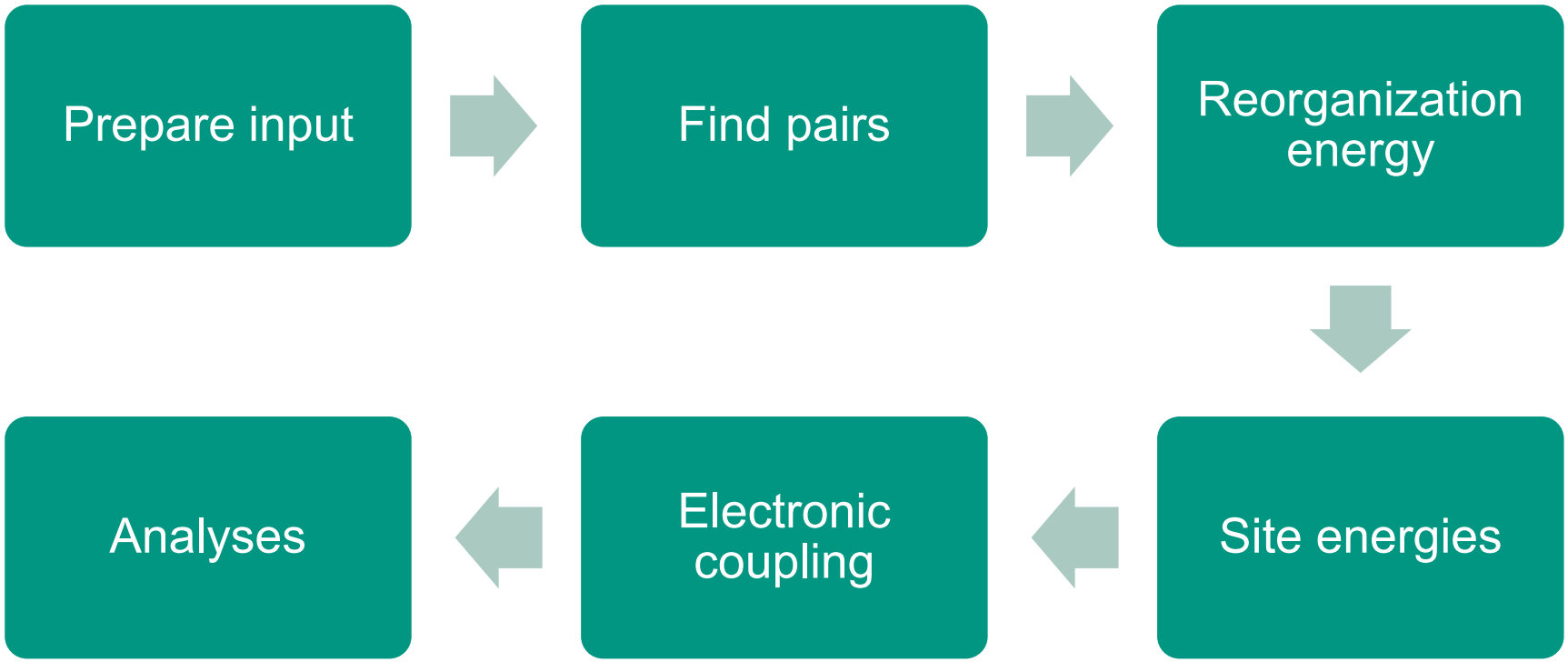
JSON/YAML:

```
gedit workflow.[json|yaml]

lpad reset
lpad add workflow.[json|yaml]
lpad get_fws –d all
rlaunch singleshot
rlaunch -s singleshot
rlaunch -s rapidfire
```

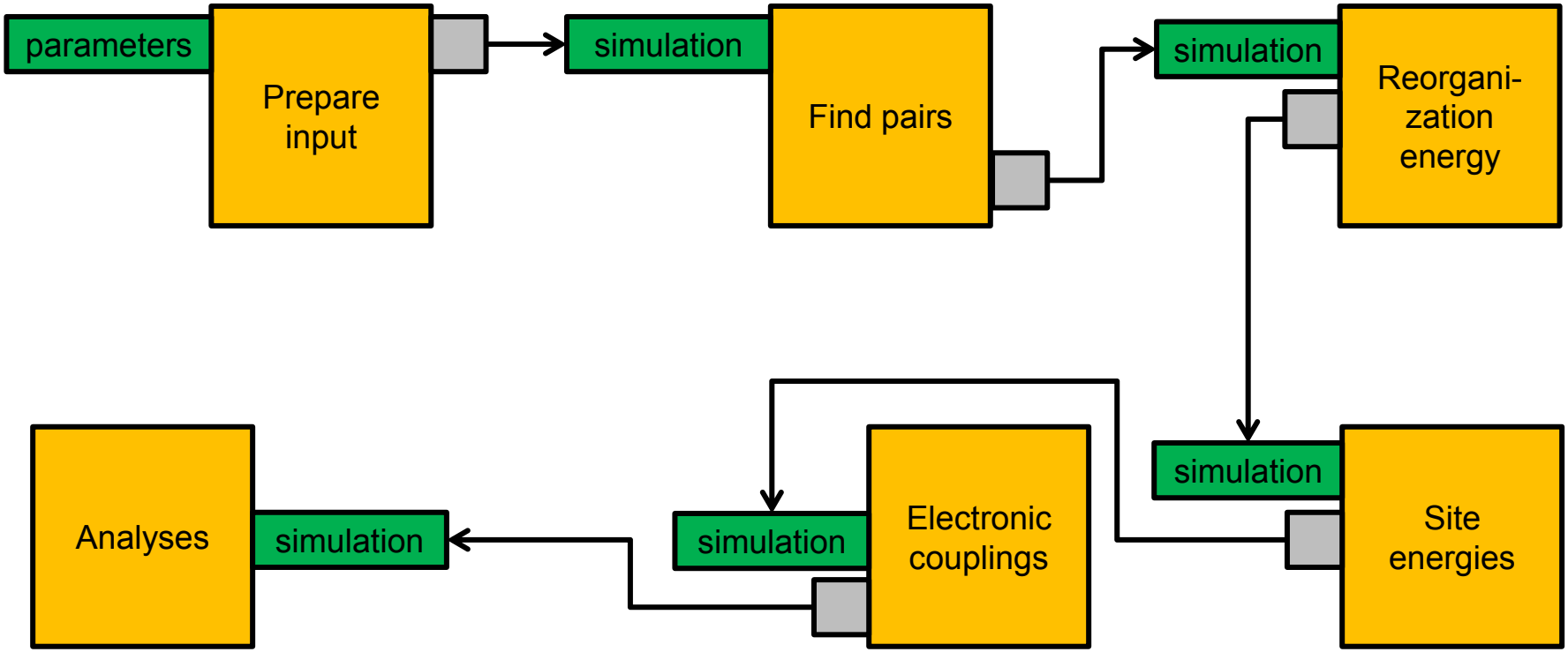
Charge transfer in dimers

- Control flow



Charge transfer in dimers

- Sequential dataflow



Exercise 3

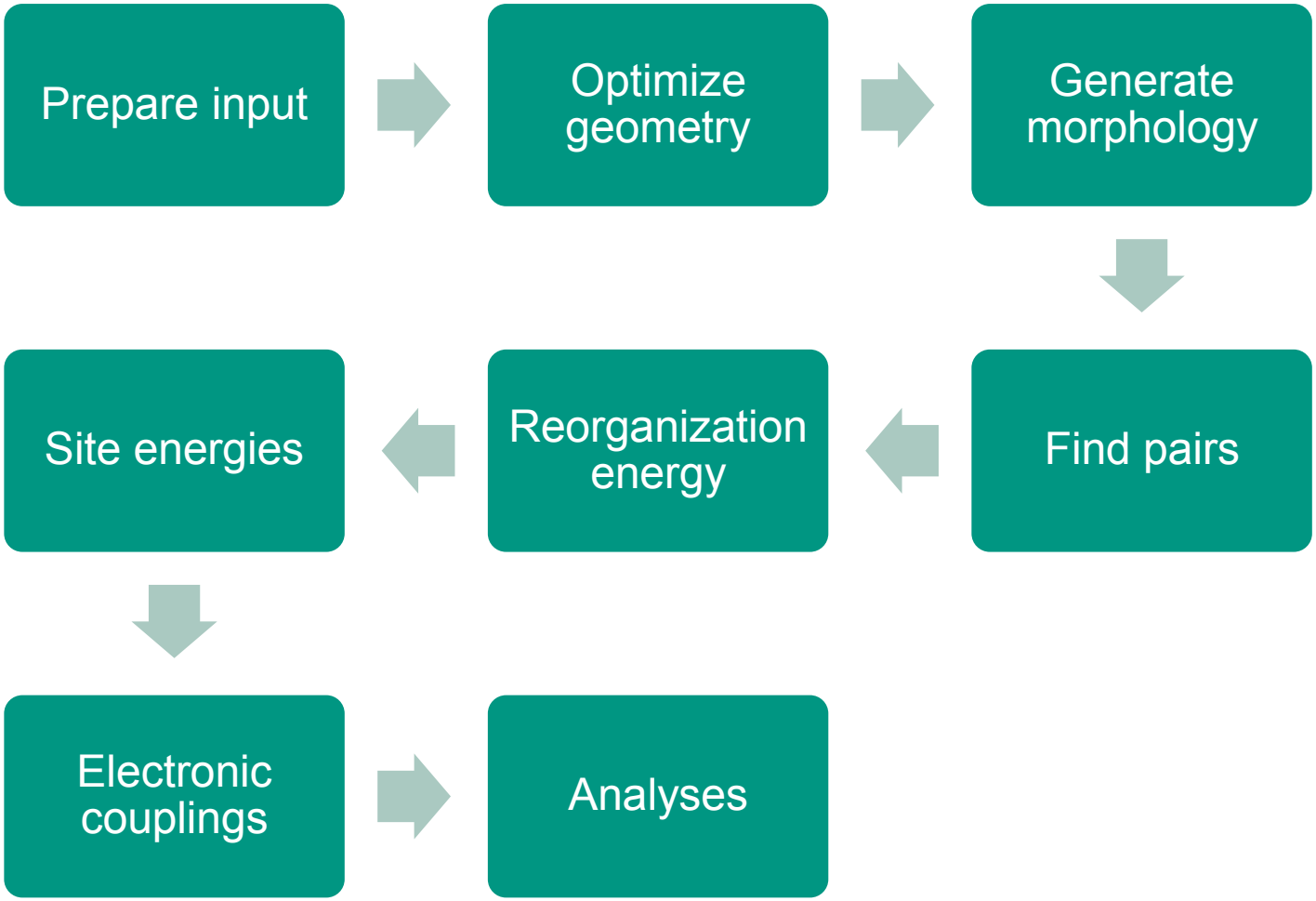
- For different dimers, for both A^+D and D^-A, calculate:
  - Internal reorganization energy λ
  - Electronic coupling V
  - Marcus rate, k

Dimer	V, eV	λ (A^+D), eV	λ (D^-A), eV	k (A^+D), s^{-1}	k (D^-A) , s^{-1}
Formic acid					
T-shaped benzene					
Parallel-displaced benzene					
H-bonded uracil					
Uracil stack					
Alq_3					

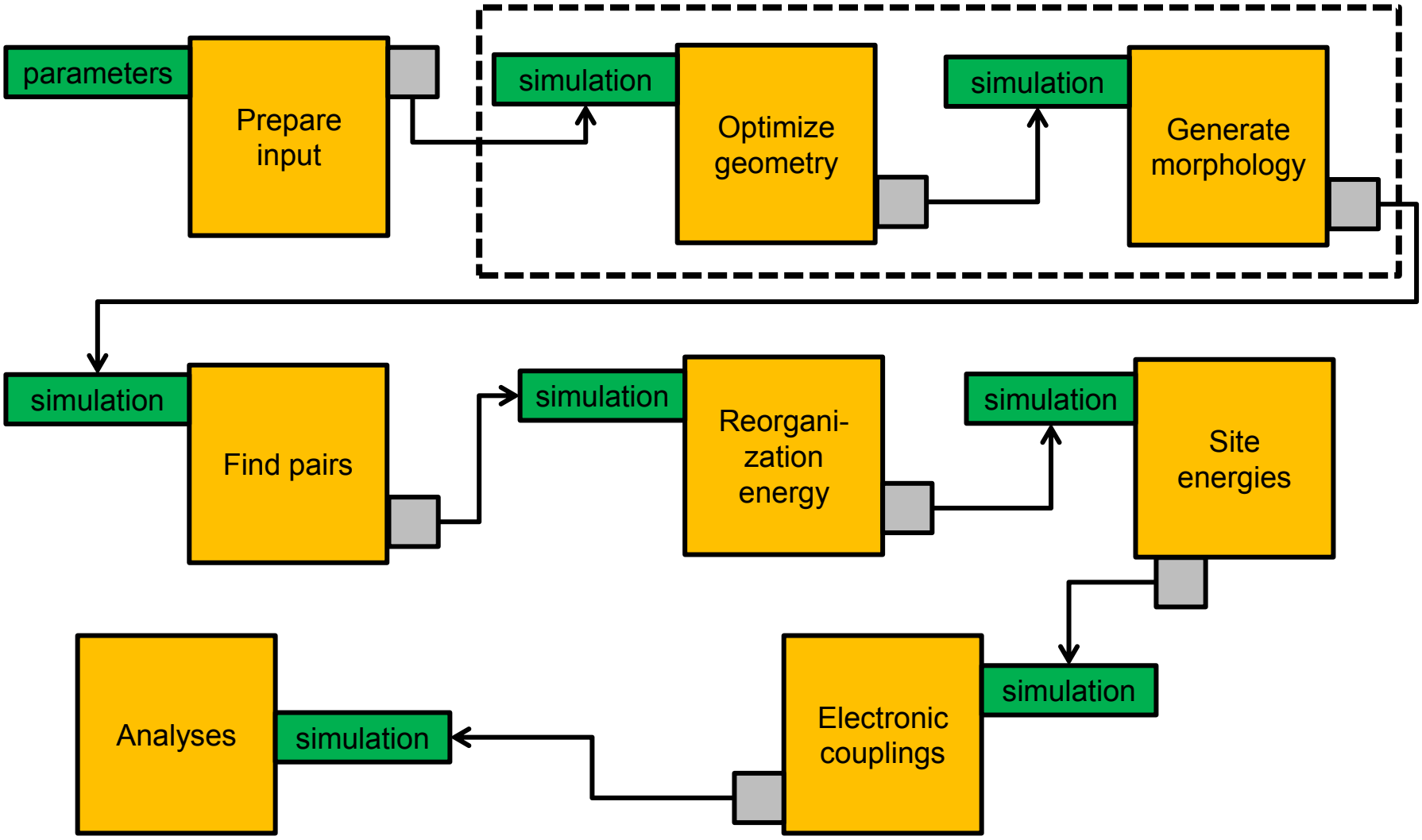
- Optional task: Repeat the calculation with the def2-SV(P) basis set.

Charge transport in amorphous structures

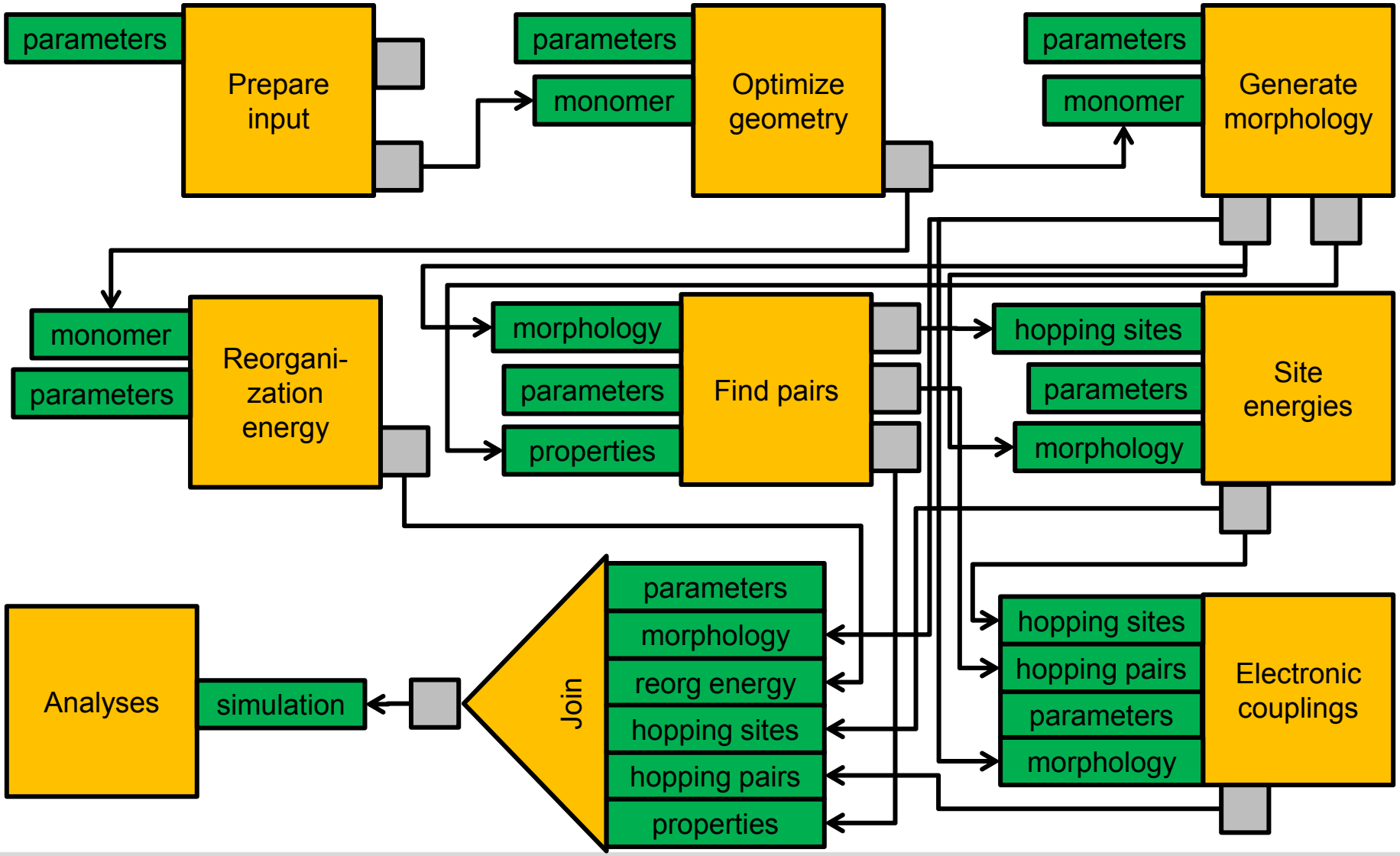
Sequential workflow



Charge transfer in amorphous structures



Charge transport in amorphous structures

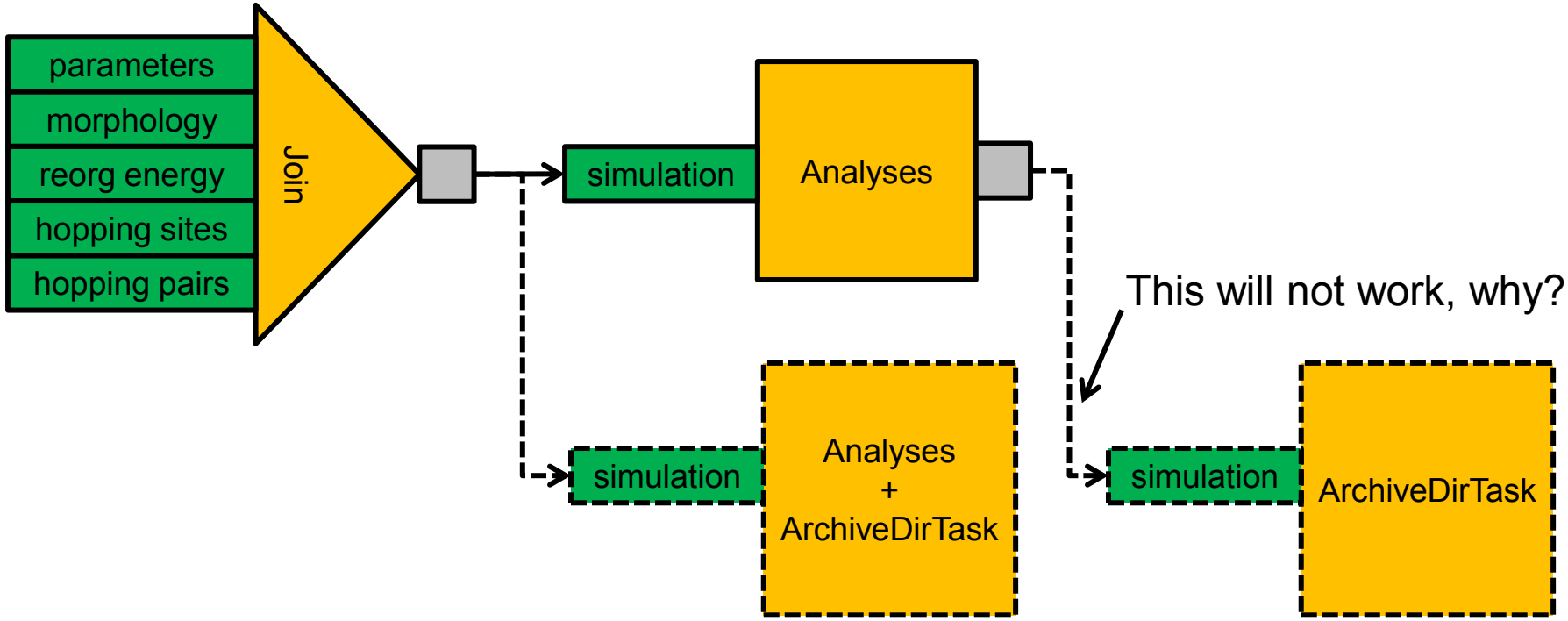


Exercise 4

- For Alq3 generate a morphology in a box with dimensions 40x40x40 Å.
- In the case of zero electric field and low carrier concentration, calculate:
  - Energy disorder σ
  - Mean electronic coupling $\langle V^2 r^2 \rangle$
  - Hole mobility μ^+

Appending fireworks / workflow

- Late extension in order to
  - Reuse simulation results in an other workflow
  - Repeat parts of the simulation with other parameters
  - Analyze / visualize / download results



Exercise 5

- Extend the completed workflow from Exercise 4 to visualize and download the results.

This work is licensed under an Attribution-NonCommercial-NoDerivatives 4.0 International Creative Commons License
http://creativecommons.org/licenses/by-nc-nd/4.0/

Copyright © 2016 Karlsruhe Institute of Technology (KIT)