POSTOPUS

POST processing of OctoPUS data

SSU - Computational Science

MPSD

September 14, 2023

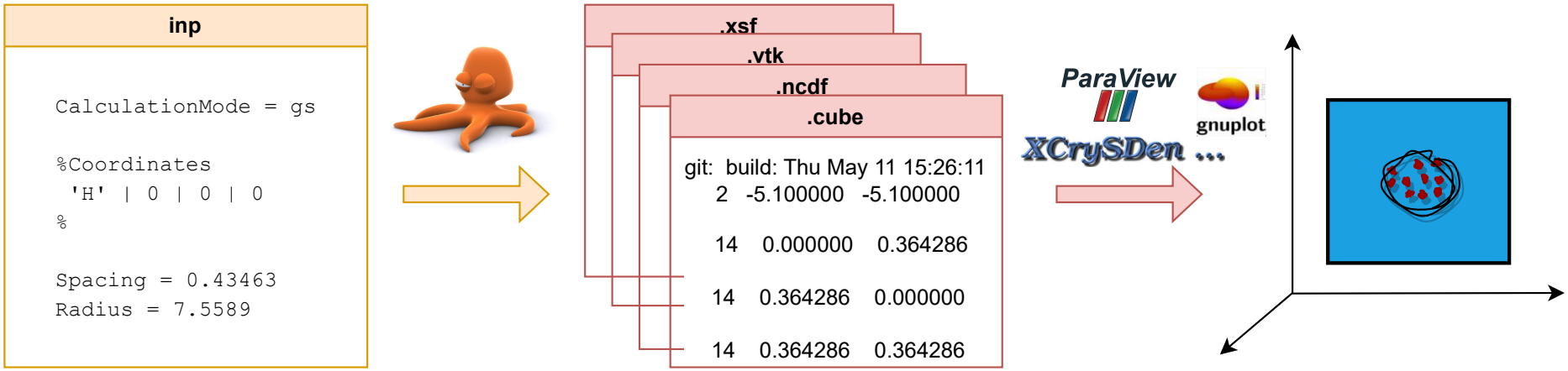
Developers

- Daniel Bremer,
- Kevin Yanes Garcia,
- Martin Lang,
- Ashwin Kumar Karnad,
- Fabian Gropp,
- Sebastian Ohlmann,
- Hans Fangohr

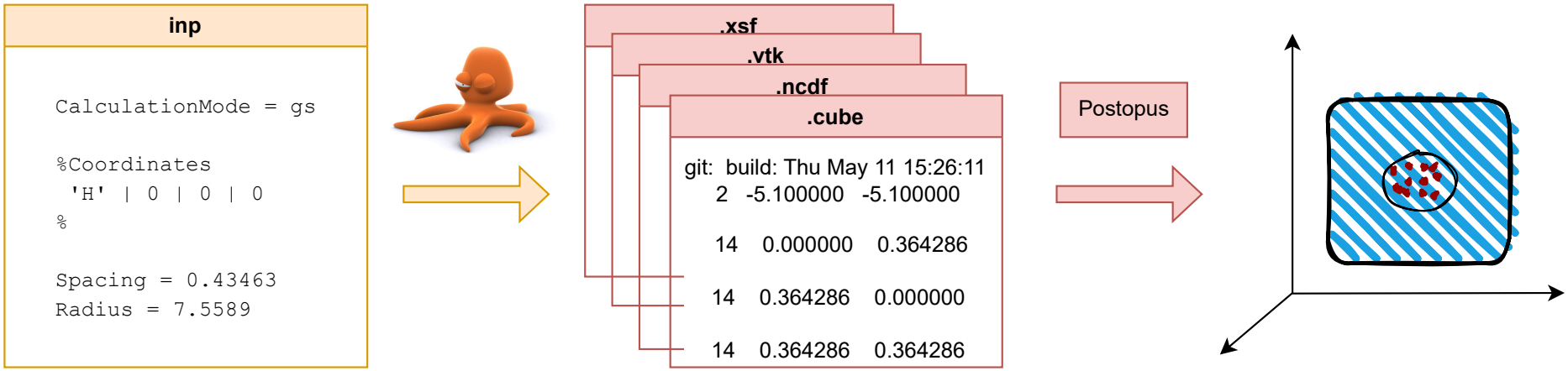
Motivation

- POST-processing of OctoPUS data must be:
  - Easy to use
  - Unified Workflow
  - Easily reproducible

Octopus workflow



Postopus — Integration of Postopus in Octopus workflow



Octopus project structure — Benzene example

```
$ cd benzene_example
$ tree

├── benzene.xyz ..............Geometry of the molecules (input)
└── inp ......................................Our input file
```

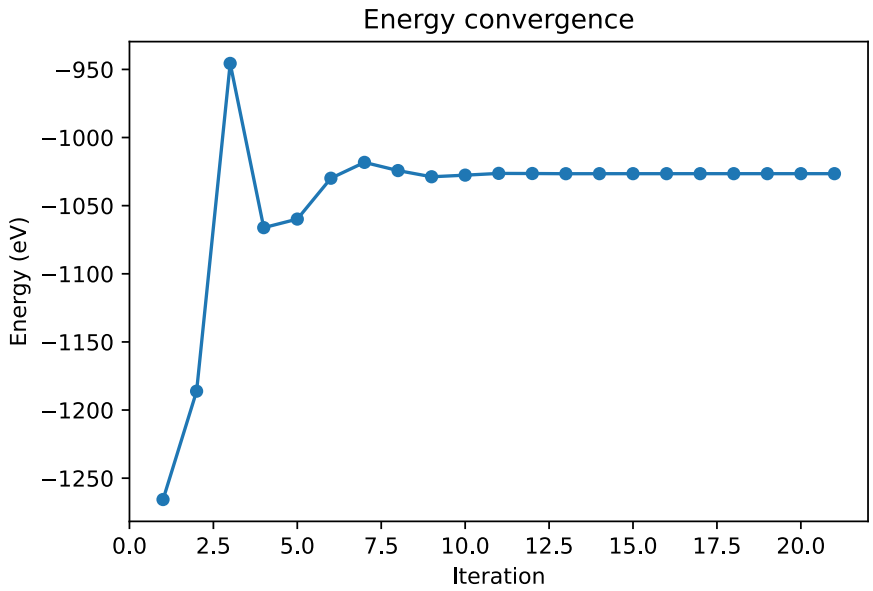
Octopus project structure — Benzene example

```
$ octopus 2>&1 | tee out_gs.log
$ tree
```

```
├── benzene.xyz ..............Geometry of the molecules (input)
├── exec
├── ...
│   └── parser.log ........Full set of variables used for the run
├── inp ..........................................Our input file
├── out_gs.log ...........................................Log file
├── output_iter ..............Output for each iteration (empty)
├── restart
│   └── gs
│       ├── 0000000001.obf  Checkpoint file to restart calculation
│       ├── 0000000002.obf
│       ├── ...
│       └── wfns
└── static
    ├── info
    ├── convergence
    ├── density.cube
    ├── ...
    └── wf-st0015.z=0
```

Postopus — Benzene Example — Convergence

In [4]:
```
from postopus import Run
run = Run(".")
convergence_df = run.default.scf.convergence # pd.Df
fig = convergence_df['energy'].plot(
    title='Energy convergence',
    marker='.',
    markersize=10,
)
fig.set(xlabel="Iteration",ylabel="Energy (eV)");
```



Postopus — Benzene Example — Density

In [4]:
```
from postopus import Run
run = Run(".")
density = run.default.scf.density # field object
xa = density.get_converged("cube") # xarray
s1 = xa.sel(z=0, method="nearest") # slicing
s1
```

Out[4]:

xarray.DataArray 'density' (**step**: 1, **x**: 95, **y**: 99)

```
array([[[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]]])
```

▾ Coordinates:

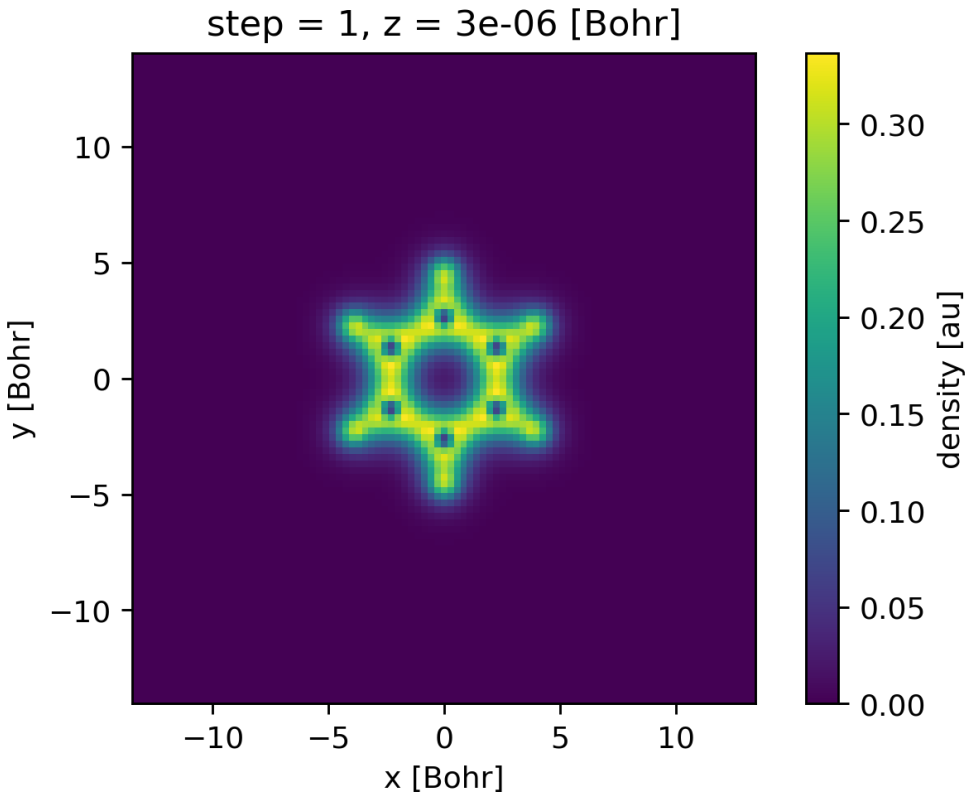
step	(step)	int64	1
x	(x)	float64	-13.32 -13.04 ... 13.04 13.32
y	(y)	float64	-13.89 -13.61 ... 13.61 13.89
z	()	float64	3e-06

▾ Attributes:

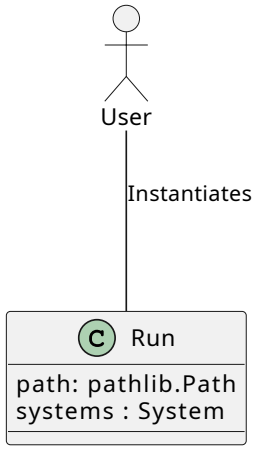
units :	au

Postopus — Benzene Example — Density

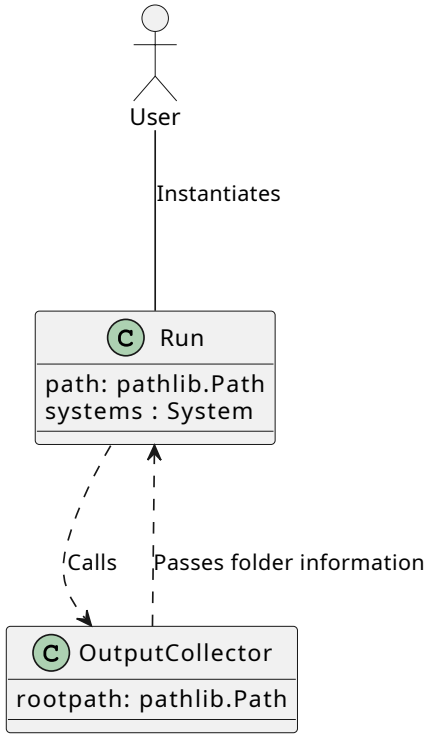
```
In [18]: s1.plot(x="x")
         plt.gca().set_aspect('equal')
```



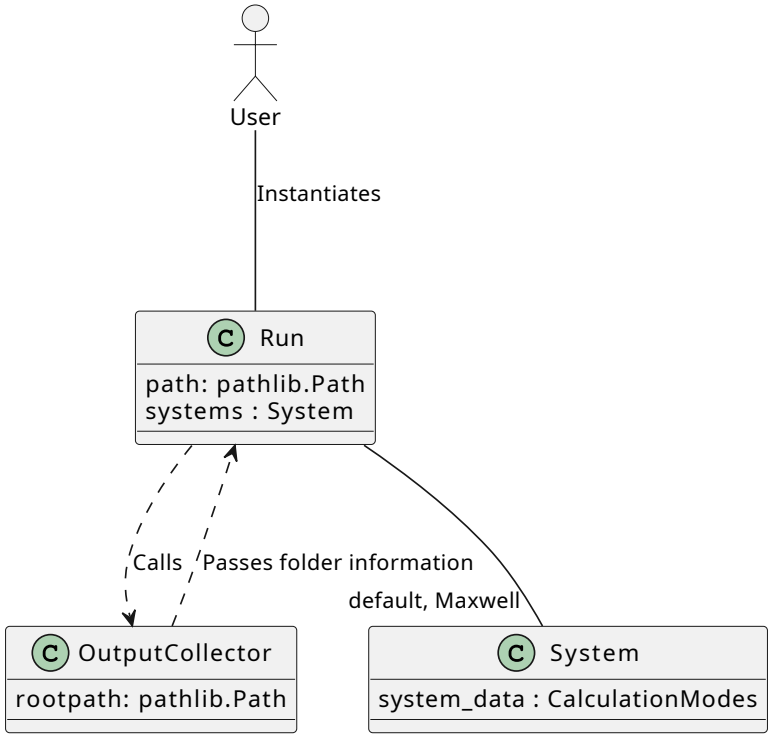
Postopus — Structure



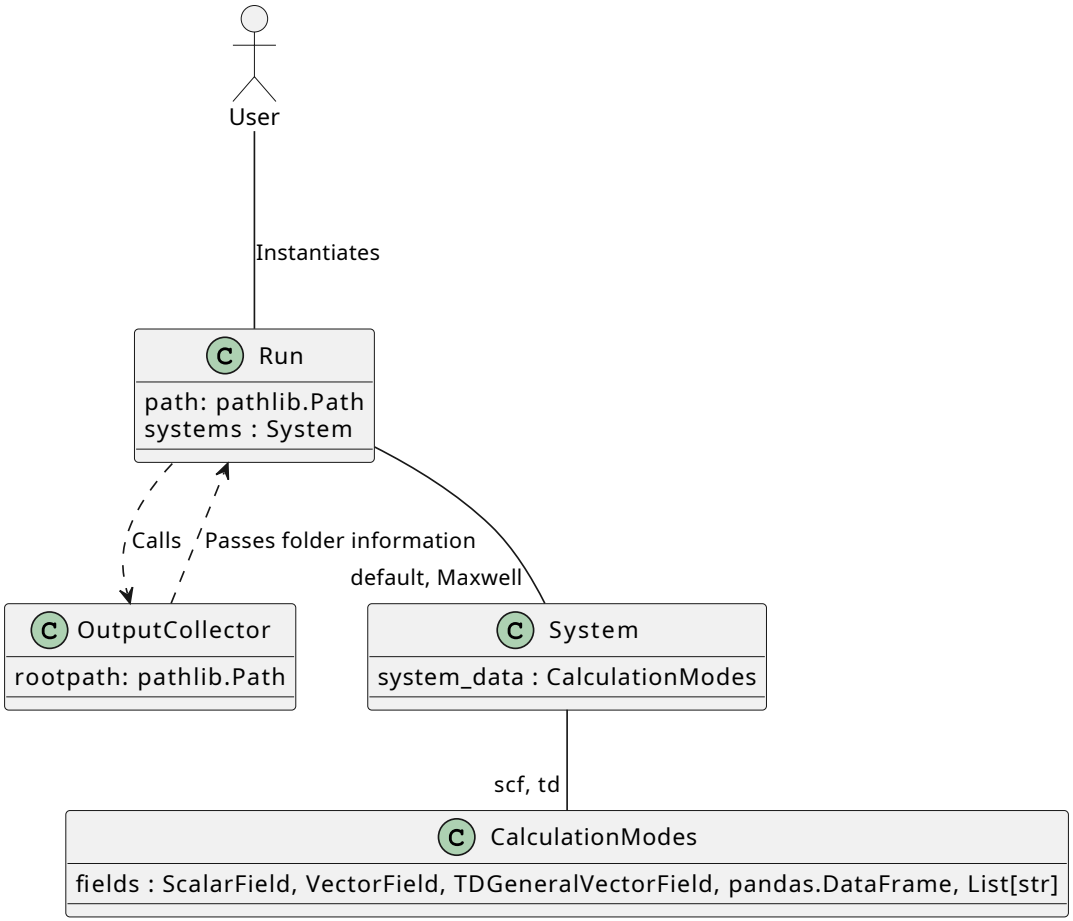
Postopus — Structure



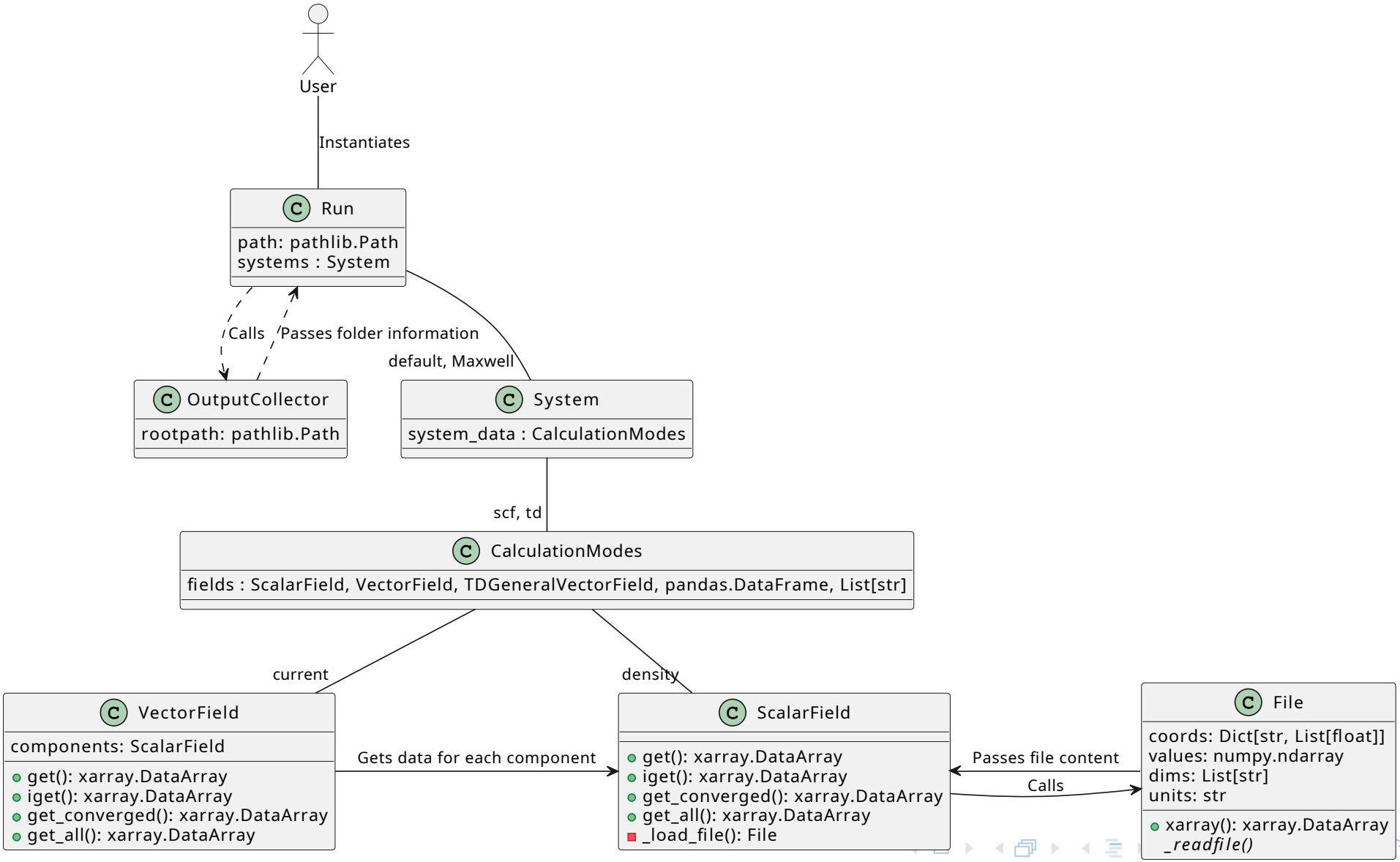
Postopus — Structure



Postopus — Structure



Postopus — Structure



Postopus — Installation

```
(venv) $ pip install "postopus[recommended]"                    *
```

* As of 14/09/23 you need to install the package *ase* from their master branch

```
pip install  git+https://gitlab.com/ase/ase.git@master
```

Hands on session

Launch the binder link in the postopus repository readme:
https://gitlab.com/octopus-code/postopus