CCP CompMedChem

Technical Dive

Tim Dudgeon
tdudgeon@informaticsmatters.com

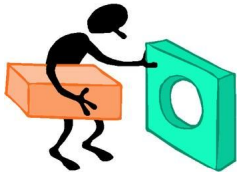
Topics

- Squonk Computational Notebook
- OpenRiskNet Project
- Containerisation
- Pipelines - a concrete example

Squonk Computational Notebook

Informatics Matters Ltd.

- Simple to use
- Open platform integrating multiple toolkits
- Support the entire workflow process
- Provide Traceability & Reproducibility
- Facilitate collaboration
- Deploy to cloud on in-house



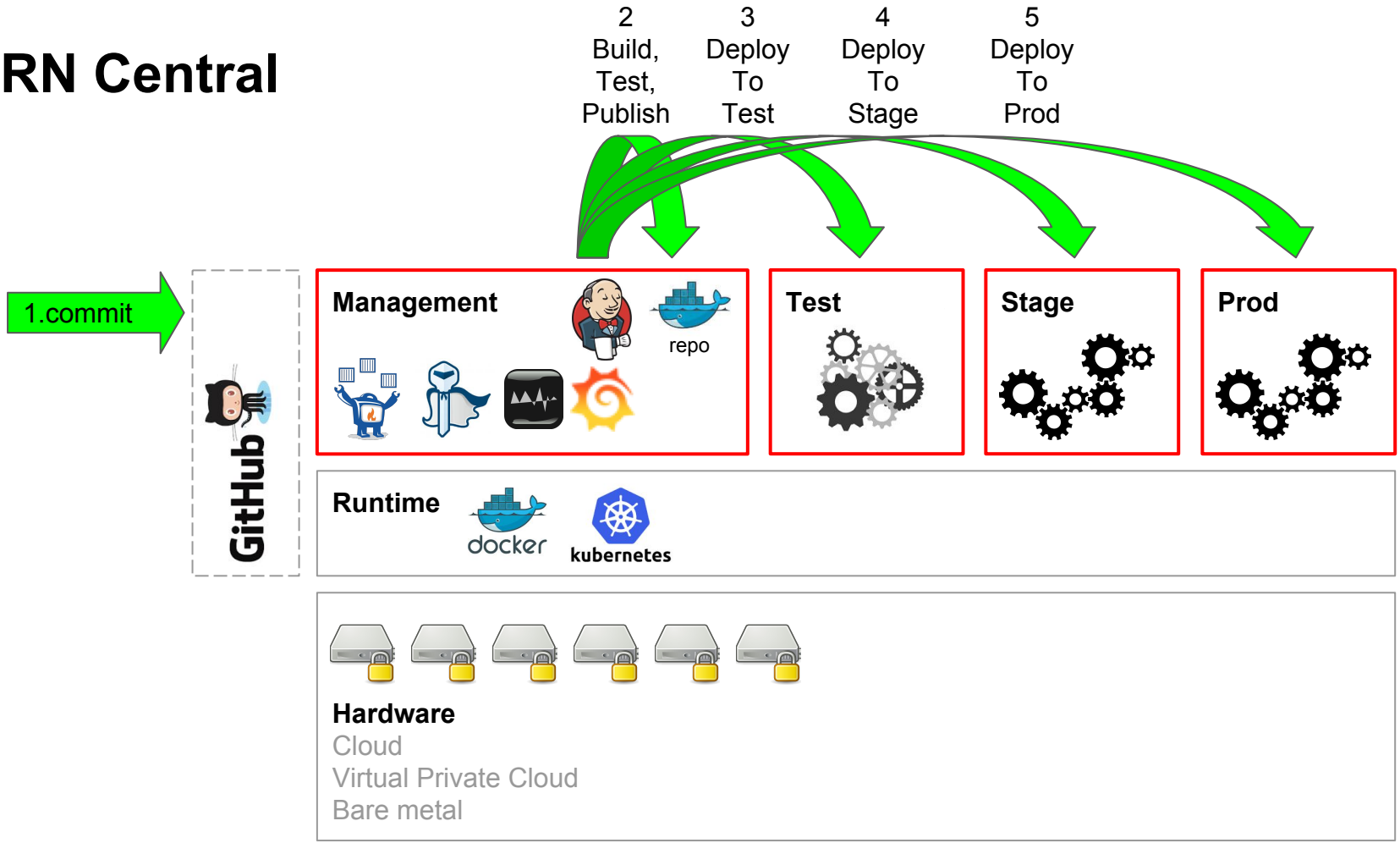
OpenRiskNet

- Horizon 2020 e-infrastructure project
- €3M budget over 3 years
- 11 partners, incl. Informatics Matters
- Started Dec 2016
- Focus is to provide e-infrastructure to support chemical safety assessment
- Target audience is pharma, biotech, agrochemicals, cosmetics, chemical industry, nano materials, environmental safety assessment, regulators

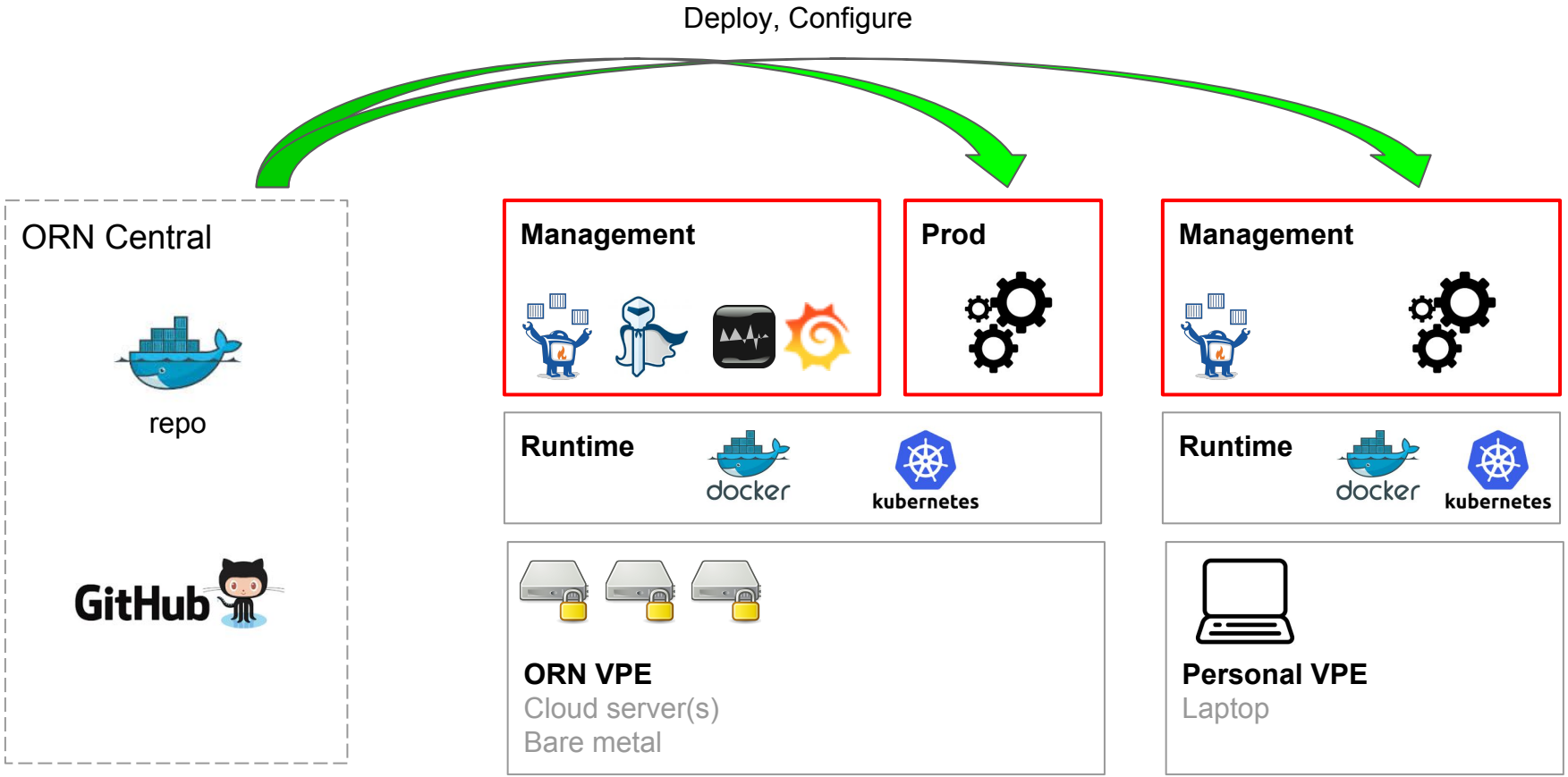
Benefits of partnering with ORN

- Avoid need to create infrastructure
- Complementary functionality
  - Access to predictive models
  - Access to common datasources
  - Generate predictive models
- Broaden the user base

ORN Central


2
Build,
Test,
Publish
3
Deploy
To
Test
4
Deploy
To
Stage
5
Deploy
To
Prod
1.commit
GitHub
Management
repo
Test
Stage
Prod
Runtime
docker
kubernetes
Hardware
Cloud
Virtual Private Cloud
Bare metal

Standalone ORN VPE



What this will provide

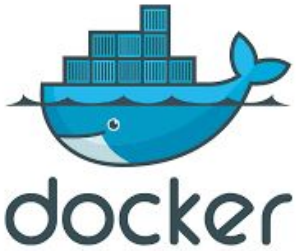
- Infrastructure to build and test software
- Public repositories for the built artifacts
- End-user environment to try out the tools
- Ability to compare tools and describe best practice
- An environment that can be deployed to your own infrastructure

Overall Approach

- Open Source where possible ...
- … but providing a place for commercial tools where appropriate …
- … allowing a more sustainable future
- Package as Docker containers

Why Containers

- Package up your code and all its dependencies in a simple to execute black box
- Avoid dependency hell
- Fast, lean and scalable
- Provide simple route to execution
- Future-proof your application - containers are taking over the world!



Examples

Aim to provide “upstream” projects that can be re-used in multiple software systems

Example 1: Javascript visualisation tools

Example 2: Pipelines project

1: Javascript visualisation tools

An OS project containing a range of Javascript components for use in webapps that provide a consistent API and UI

- Charts/plots (D3), Tables, Trees, 3D structure display, Chemical sketchers ...

We had interest in this from some companies, but the funding route didn’t work out - can we make it happen here?

2: Pipelines project

An OS project containing a range of “bite sized” components designed to allow to build execution pipelines

Contributions from Informatics Matters and XChem

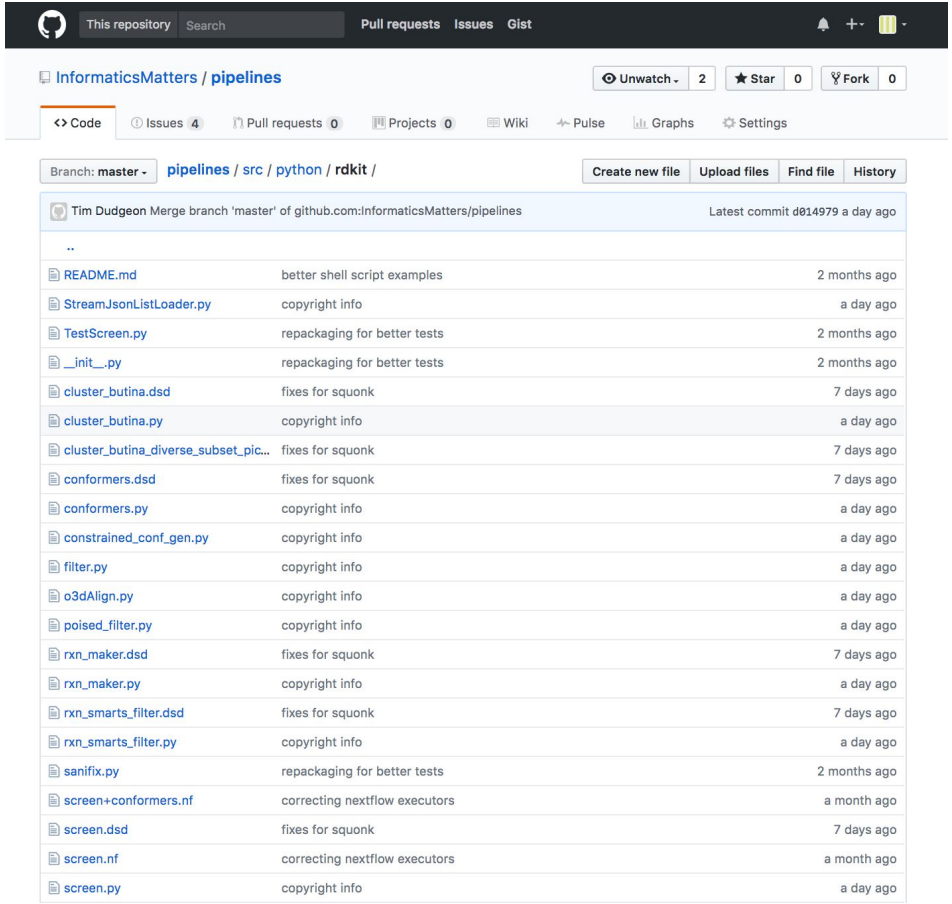
Based around the Unix pipes concept

Currently mostly Python and RDKit based, but will be extended into other toolkits and languages

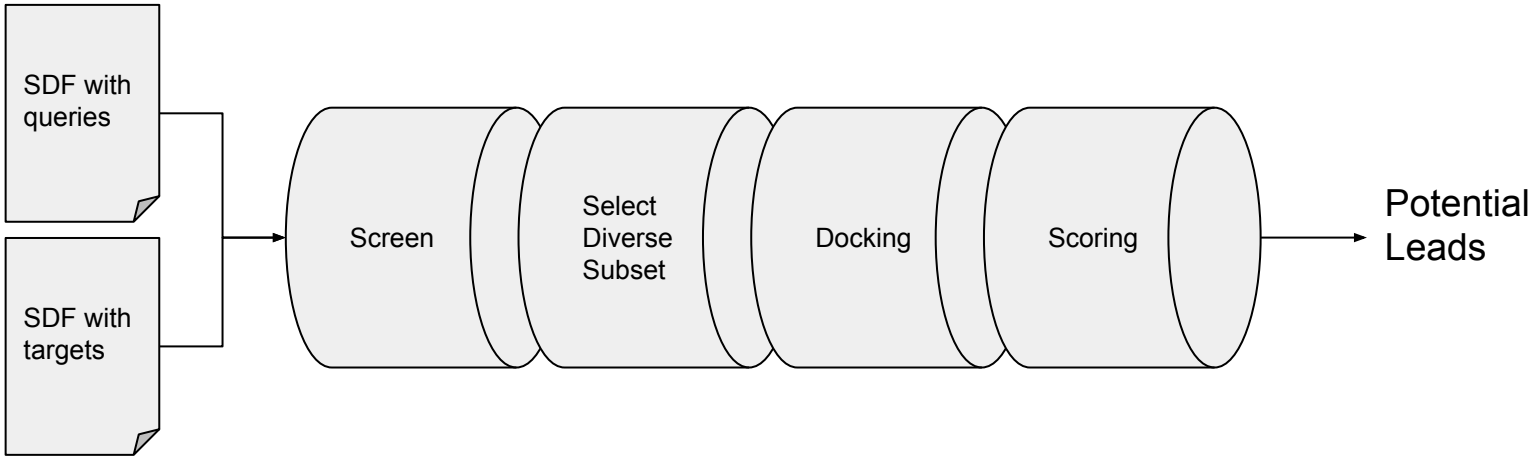
https://github.com/InformaticsMatters/pipelines

Apache 2.0 license

~10 useful components at present



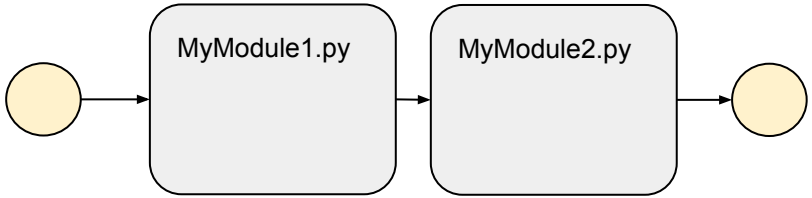
Bite sized components



Do one thing, do it well
Plan for re-use
Consistent approach to IO
Typically 50-200 lines of code

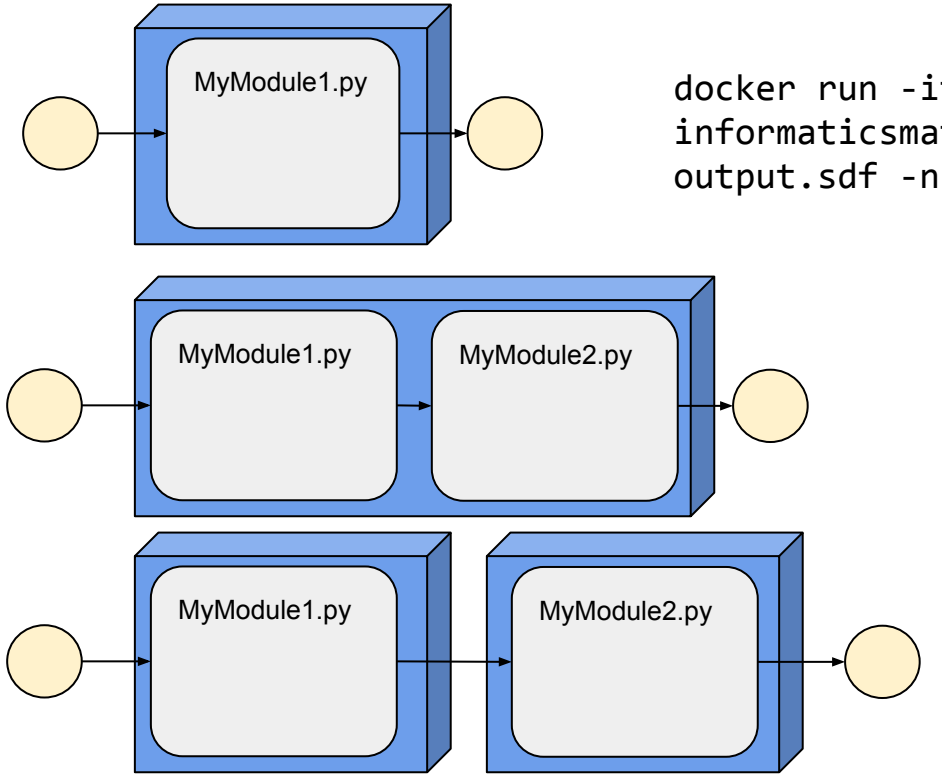
Direct use


```
MyModule1.py -i input.sdf -o output.sdf -n 5
```



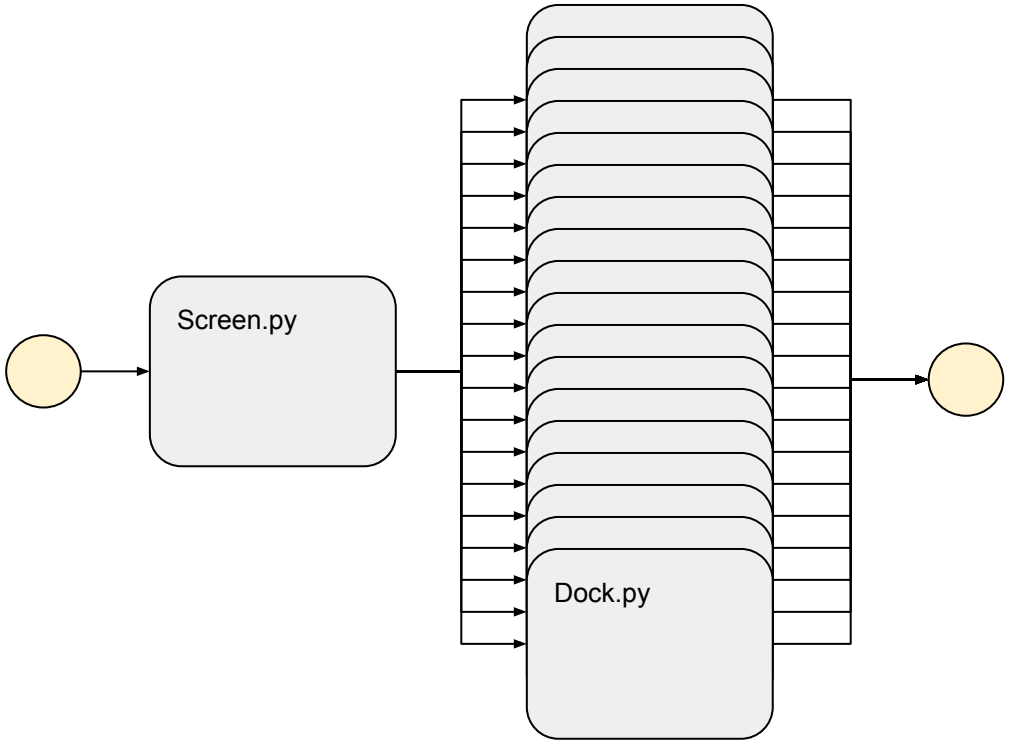

```
MyModule1.py -i input.sdf -of sdf -n 5
  | MyModule2.py -if sdf -o output.sdf
```

Use in Docker



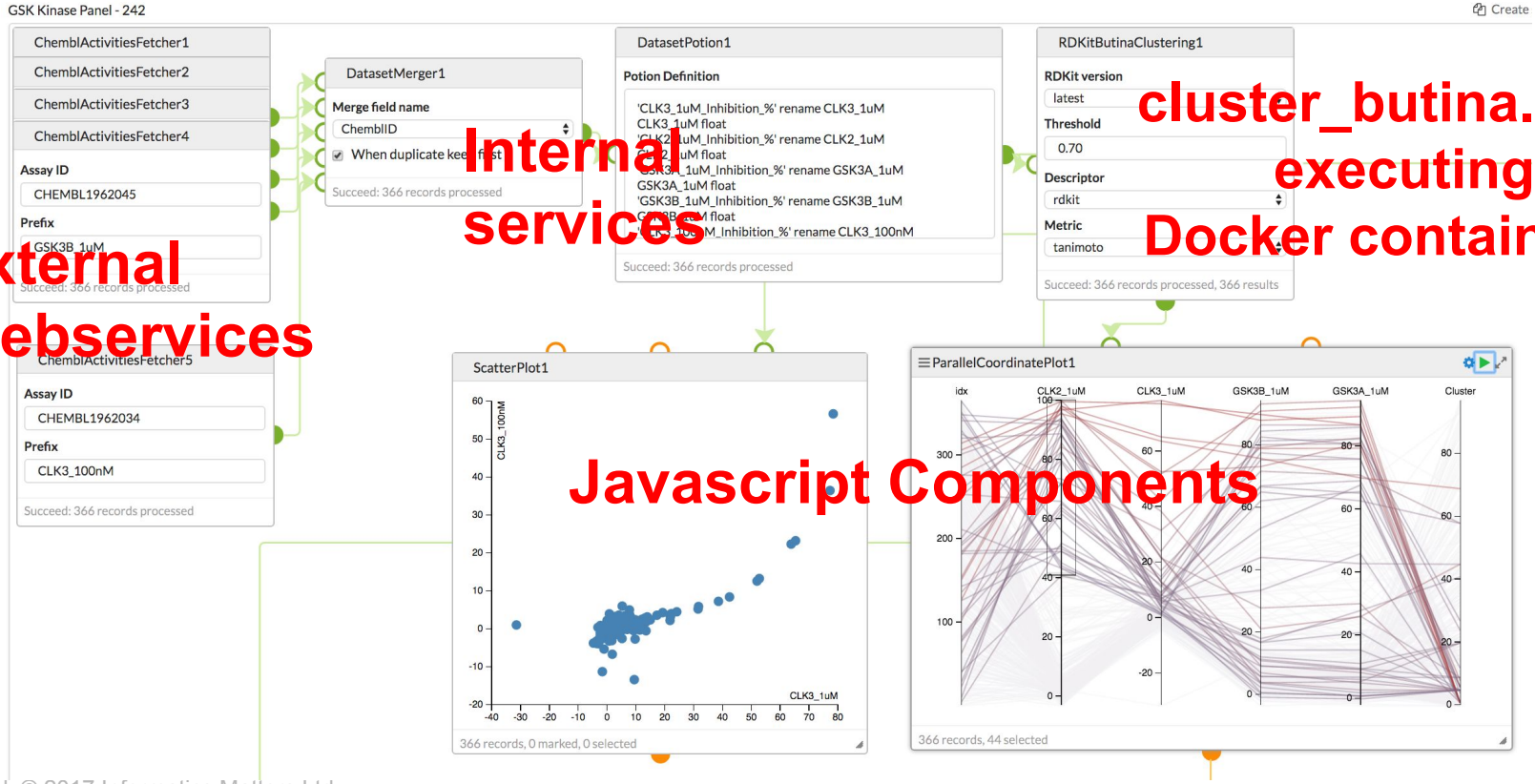

```
docker run -it --rm -v $(PWD):/work
informaticsmatters/rdkit MyModule1.py -i input.sdf -o
output.sdf -n 5
```

Use in Nextflow

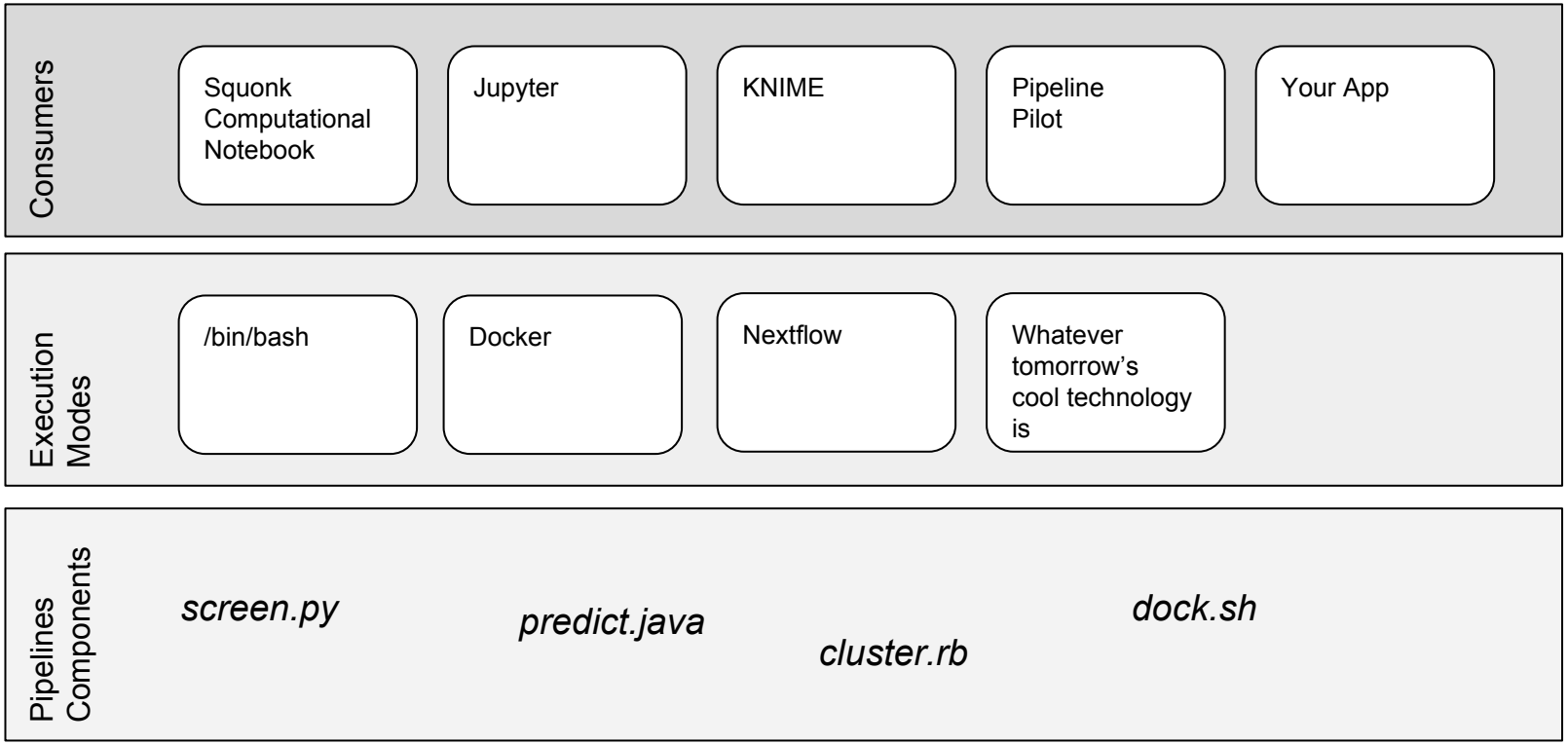



```
nextflow run screen.nf
--input mols.sdf ...
```

Squonk Example



Broader Usage



Questions

- Will this infrastructure be useful to you?
- Will these types of projects be of use to you?
- What else might be of use?
- What should be the priorities?
- What might you contribute to and in what way?