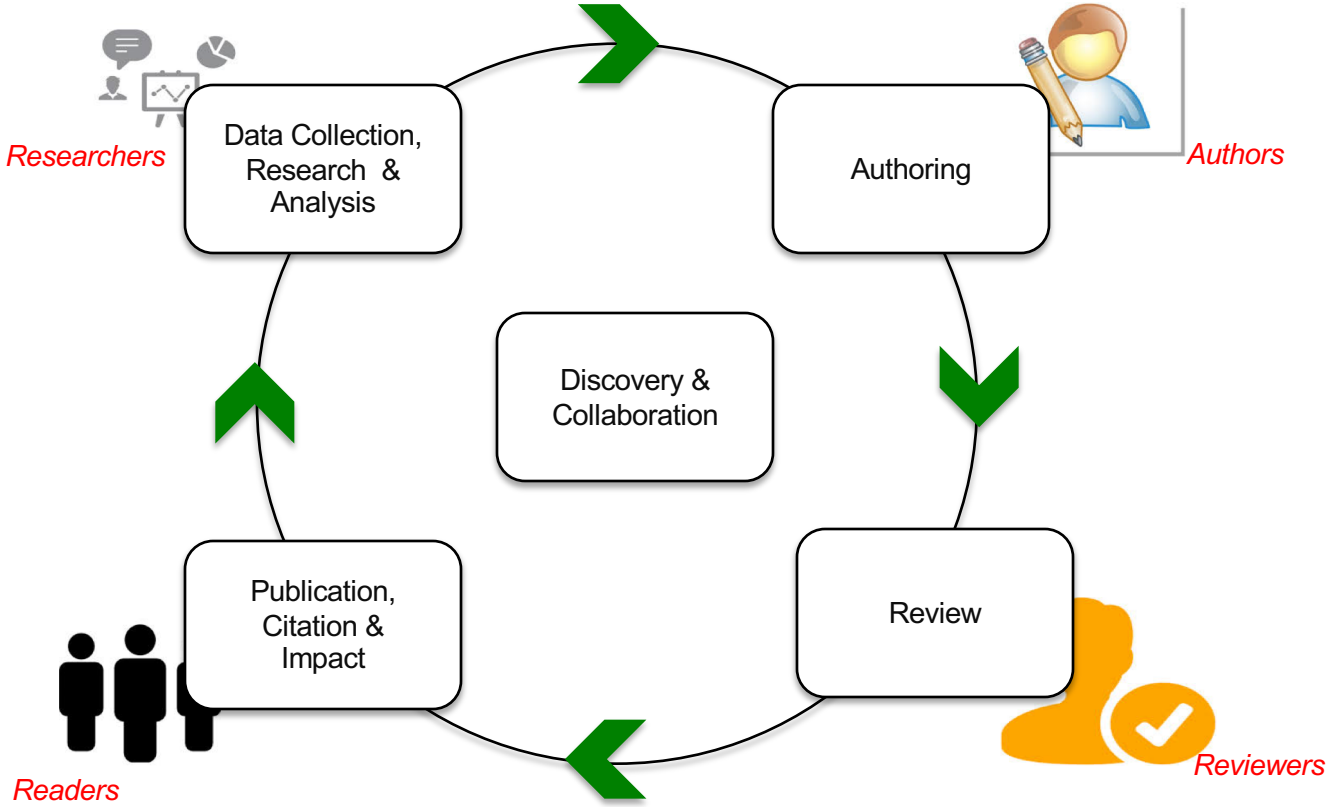
Semantically-Organized Containers for Reproducible Research

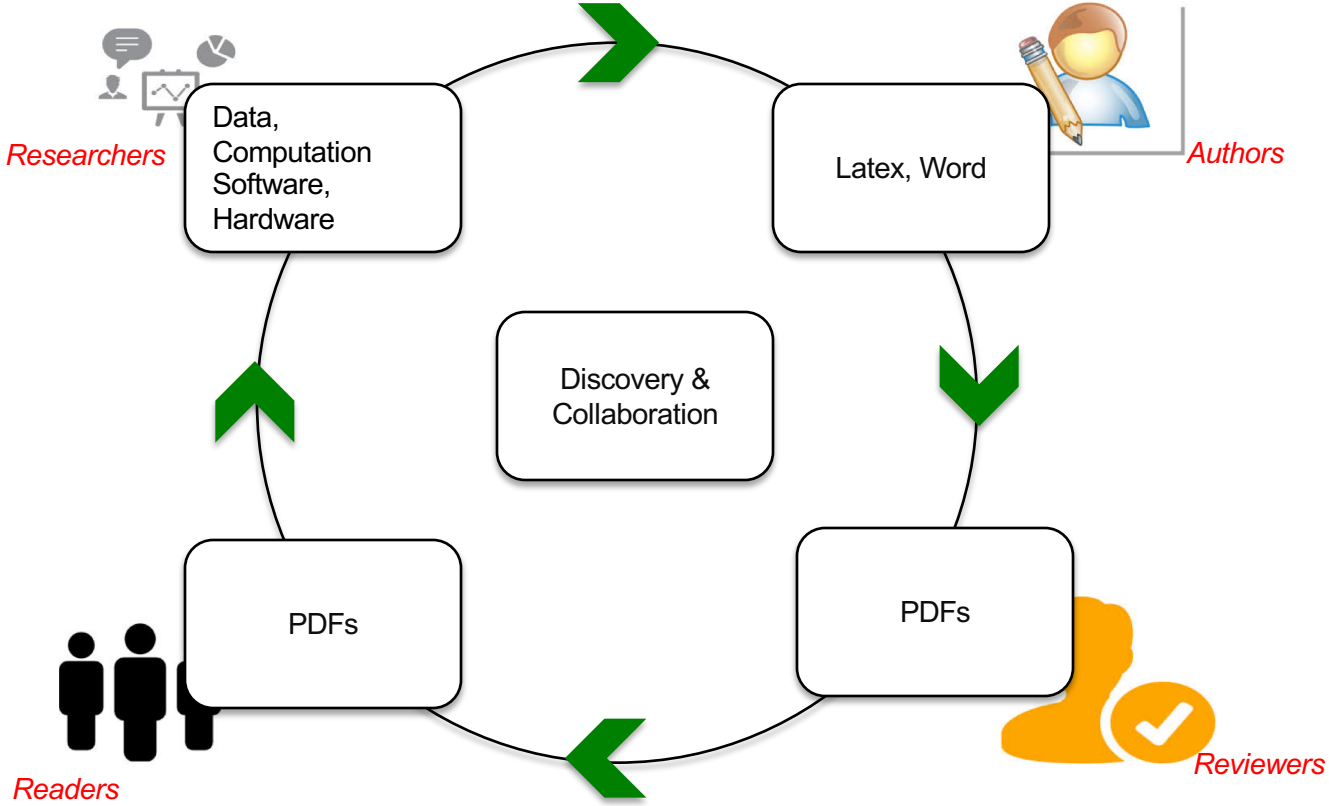
Andrew Youngdahl, Zhihao Yuan, Dai Hai Ton That, Tanu Malik
DePaul University
Chicago, IL

Ivo Jimenez, Carlos Maltzahn
UC Santa Cruz
Santa Cruz, CA

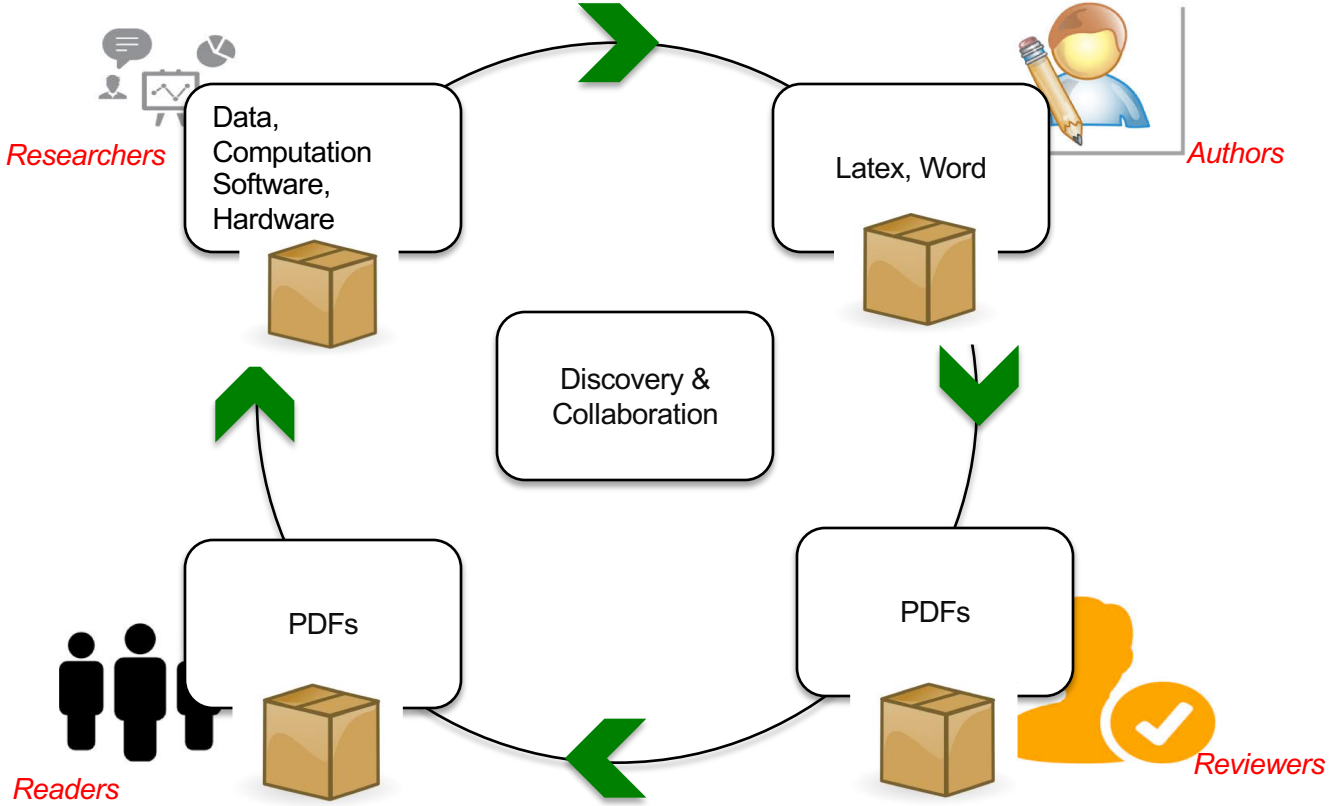
Scientific Publication Cycle



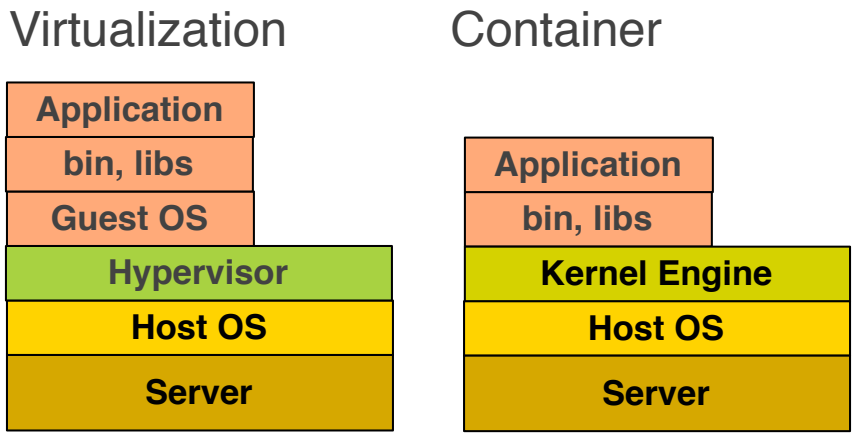
Scientific Publication Handoffs



Scientific Publication with Computational Handoffs



Containers



- Containers are lightweight alternative to virtualization.

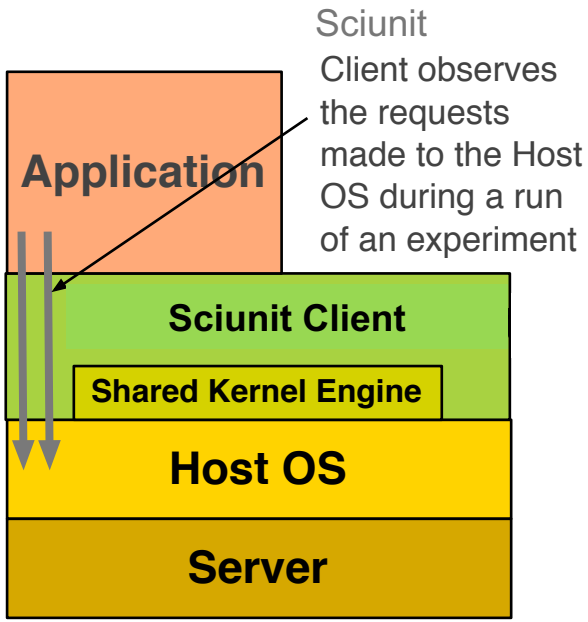
Containers for Reproducible Research

- Programmatic capture
  - Contain items that are irrelevant to the applications
  - E.g. various shell utilities and library versions unknown to user
- Rigid composition
  - No automatic way to distinguish input files from outputs to OS files or have any organization
- Large image sizes
  - Duplicate user software, environment and datasets arising from multiple runs
- Inefficient execution
  - Use layered file systems which can take a performance hit

Sciunit: Reusable Research Objects. In IEEE eScience '17

Sciunit: Automatic Containerization of Executions

http://sciunit.run



Sciunit

1. Create

2. Share

3. Run

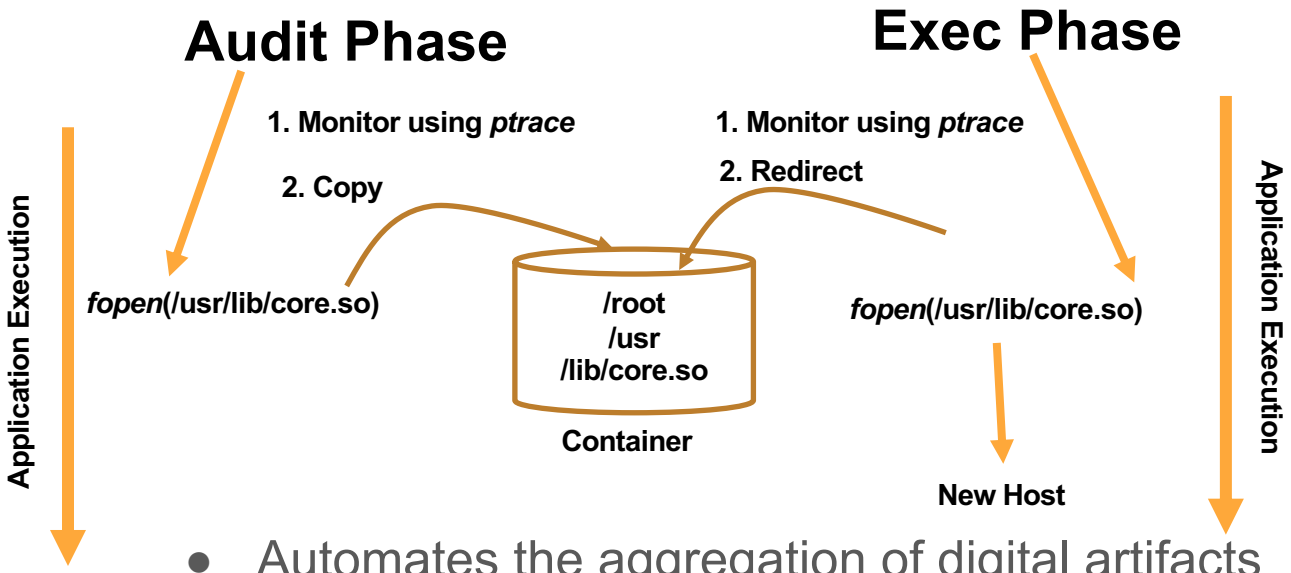




Globus RDP

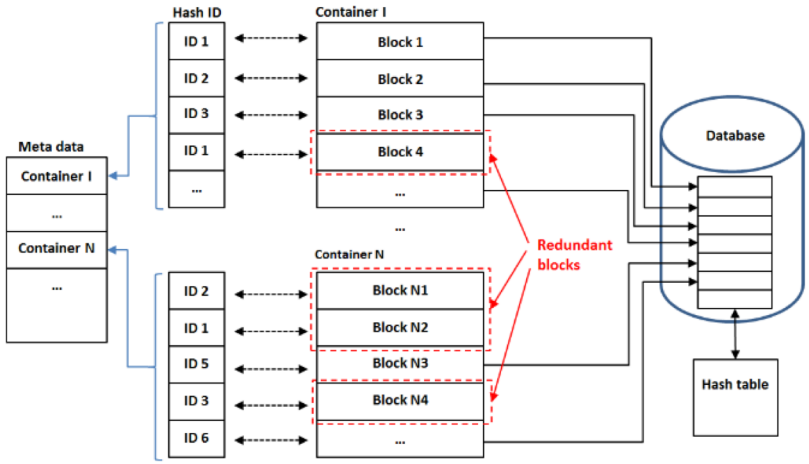


Application Virtualization



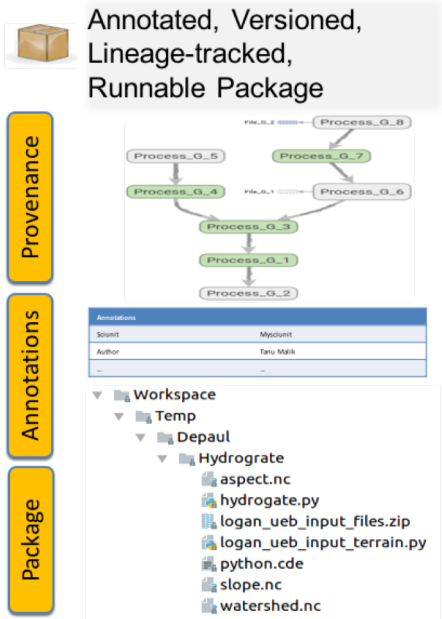
- Automates the aggregation of digital artifacts
  - Makes the computation portable and verifiable across environments.

Sciunit Storage



Deduplicated Container Storage

Sciunit


Annotated, Versioned,
Lineage-tracked,
Runnable Package
Provenance
Annotations
Package
Process_G_8
Process_G_5
Process_G_7
Process_G_4
Process_G_6
Process_G_3
Process_G_1
Process_G_2
Annotations
Sciunit
Mysciunit
Author
Tanu Malik
Workspace
Temp
Depaul
Hydrograte
aspect.nc
hydrogate.py
logan_ueb_input_files.zip
logan_ueb_input_terrain.py
python.cde
slope.nc
watershed.nc

Sample Interaction

```
1. > sciunit create FIE
2. > sciunit exec ./FIE.sh ./DATA/weather_201710.Rds
     0. Download…
     1. Calculate violation matrix…
     2. Calculate heat map…
     3. Generate model data with ./DATA/weather_201710.Rds…
     4. Apply random forest model…
     5. Evaluation…
3. > sciunit list
     e1 Dec 4 12:44 ./FIE.sh ./DATA/weather_201710.Rds
4. > sciunit show
     id: e1
     sciunit: FIE
     command: ./FIE.sh ./DATA/weather_201710.Rds
     size: 306.6 MB
     started: 2017-12-04 12:44
5. > sciunit push
     …
     Title for the new article: FIE
     new: 306.6 MB [01:05, 4.72MB/s]
6. > sciunit copy
     mSLLTj#
```

Host Computer

```
1. > sciunit repeat e1
     …
     0. Download…
     1. Calculate violation matrix…
     2. Calculate heat map…
     3. Generate model data with ./DATA/weather_201710.Rds…
     4. Apply random forest model…
     5. Evaluation…
     …
2. > sciunit repeat e1 <27050>
     …
     3. Generate model data with ./DATA/weather_201710.Rds…
     …
3. > sciunit given '/tmp/weather_201801.Rds' e1 %
     …
     0. Download…
     1. Calculate violation matrix…
     2. Calculate heat map…
     3. Generate model data with /tmp/weather_201801.Rds…
     4. Apply random forest model…
     5. Evaluation…
     …
```

Target Computer

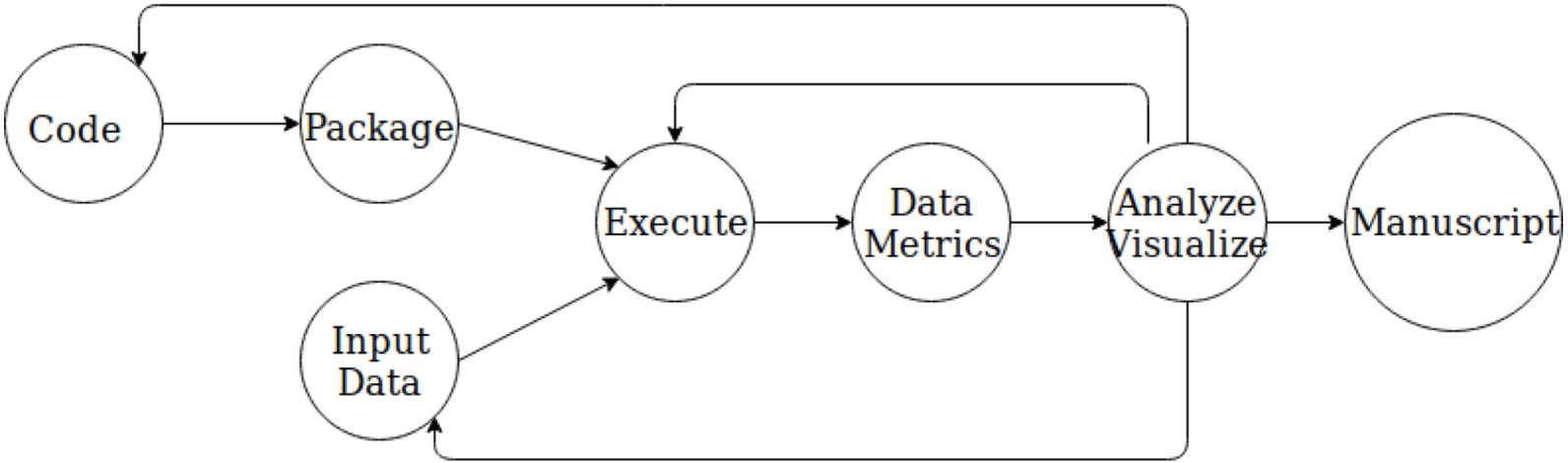
Containers for Reproducible Research

- ~~Programmatic capture~~
  - Captures necessary and sufficient program dependencies using *ptrace*
- Rigid composition
  - No automatic way to organize the container content in meaningful ways
- ~~Large image sizes~~
  - Uses content de-duplication to disambiguate at block level
- ~~Inefficient execution~~
  - Use *ptrace* for re-execution and native file systems for re-execution.

Popper: Incentivizing organization

- Convention to provide versioning, structure, clarity, and repeatability to a workflow
- Build *pipelines* which utilize common DevOps tools (Docker, CI, and validation tools) to conduct computational experiments
- Pipeline initialized as a git repository with metadata and scripts which produce the entirety of a research work

Steps covered by Popper flow

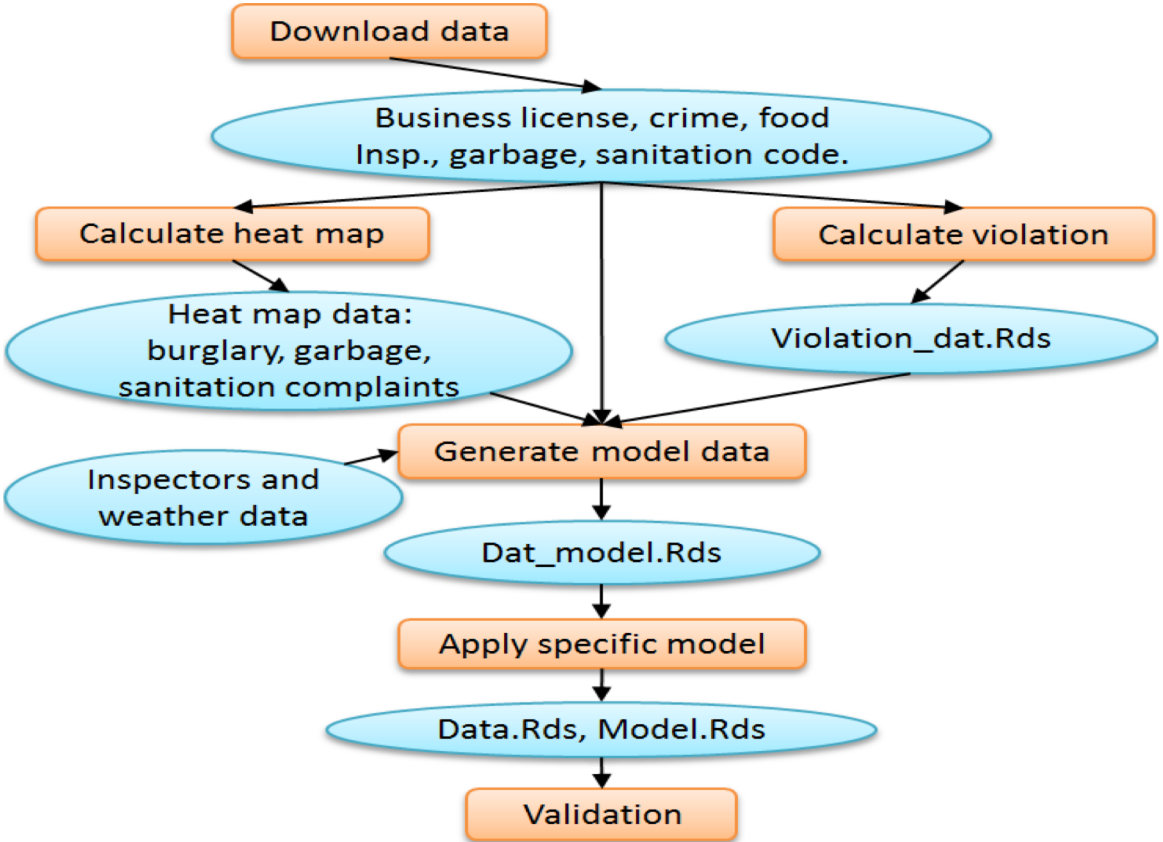


Adapted from Ivo Jimenez, Michael Sevilla, Noah Watkins, Carlos Maltzahn, Jay Lofstead, Kathryn Mohror, Andrea Arpaci-Dusseau and Remzi Arpaci-Dusseau. *The Popper Convention: Making Reproducible Systems Evaluation Practical*. In 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 1561–70, 2017.

Sciunit-Popper integration

- Should the convention be followed within a container or should the containers be part of a convention

Food Inspection Evaluation -- An example workflow

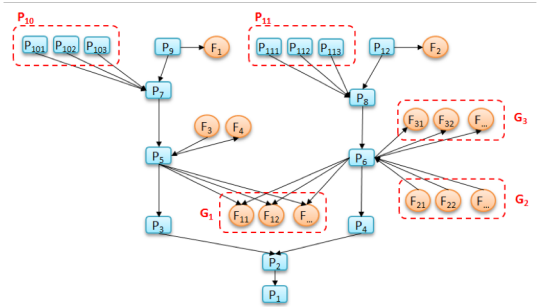


Provenance

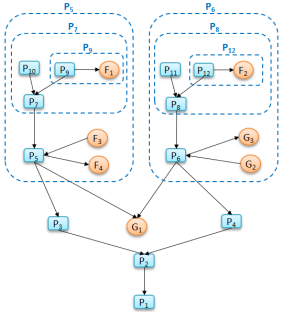
```
1507596280 10770 CLOSE /usr/lib/python2.7/site-packages/chardet-3.0.4-py2.7.egg
1507596280 10770 READ /usr/lib/python2.7/site-packages/ipaddress-1.0.18-py2.7.egg
1507596280 10770 CLOSE /usr/lib/python2.7/site-packages/ipaddress-1.0.18-py2.7.egg
1507596280 10770 CLOSE /usr/lib64/python2.7/site.py
1507596280 10770 READ /usr/lib/locale/locale-archive
1507596280 10770 CLOSE /usr/lib/locale/locale-archive
1507596280 10770 READ /usr/lib64/python2.7/encodings/__init__.py
1507596280 10770 READ /usr/lib64/python2.7/encodings/__init__.pyc
1507596280 10770 CLOSE /usr/lib64/python2.7/encodings/__init__.pyc
1507596280 10770 READ /usr/lib64/python2.7/codecs.py
1507596280 10770 READ /usr/lib64/python2.7/codecs.pyc
1507596280 10770 CLOSE /usr/lib64/python2.7/codecs.pyc
1507596280 10770 CLOSE /usr/lib64/python2.7/codecs.py
1507596280 10770 READ /usr/lib64/python2.7/encodings/aliases.py
1507596280 10770 READ /usr/lib64/python2.7/encodings/aliases.pyc
1507596280 10770 CLOSE /usr/lib64/python2.7/encodings/aliases.pyc
1507596280 10770 CLOSE /usr/lib64/python2.7/encodings/aliases.py
1507596280 10770 CLOSE /usr/lib64/python2.7/encodings/__init__.py
1507596280 10770 READ /usr/lib64/python2.7/encodings/utf_8.py
1507596280 10770 READ /usr/lib64/python2.7/encodings/utf_8.pyc
1507596280 10770 CLOSE /usr/lib64/python2.7/encodings/utf_8.pyc
1507596280 10770 CLOSE /usr/lib64/python2.7/encodings/utf_8.py
1507596280 10770 READ /home/gfils/pydelty3.py
1507596280 10770 CLOSE /home/gfils/pydelty3.py
1507596280 10770 READ /home/gfils/pydelty3.py
1507596280 10770 CLOSE /home/gfils/pydelty3.py
1507596280 10770 SPAWN 10771
1507596280 10770 EXECVE 10771 /bin/sh /home/gfils ["sh", "-c", "rm tmp.*"]
1507596280 10771 EXECVE2 10770
1507596280 10770 MEM 136056832
1507596280 10770 MEM 136056832
1507596280 10771 MEM 1409024
1507596280 10771 READ /etc/ld.so.cache
1507596280 10771 CLOSE /etc/ld.so.cache
1507596280 10771 READ /lib64/libtinfo.so.6
```

Part Of A Normal (Verbose) Provenance Log

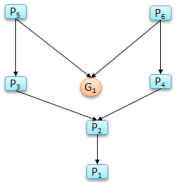
Small Section Of Graph Built From Normal Provenance Log



Full Graph

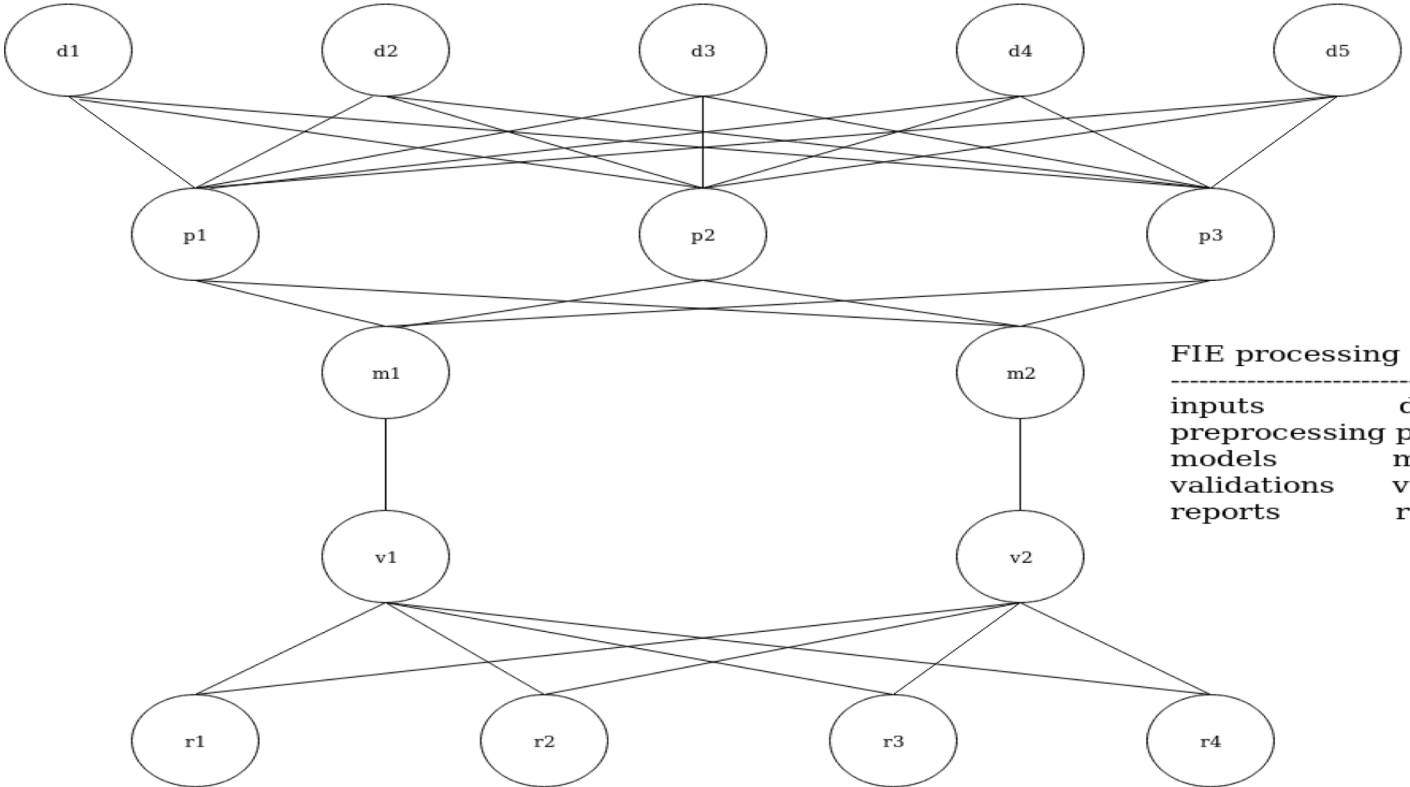


Similarity Applied



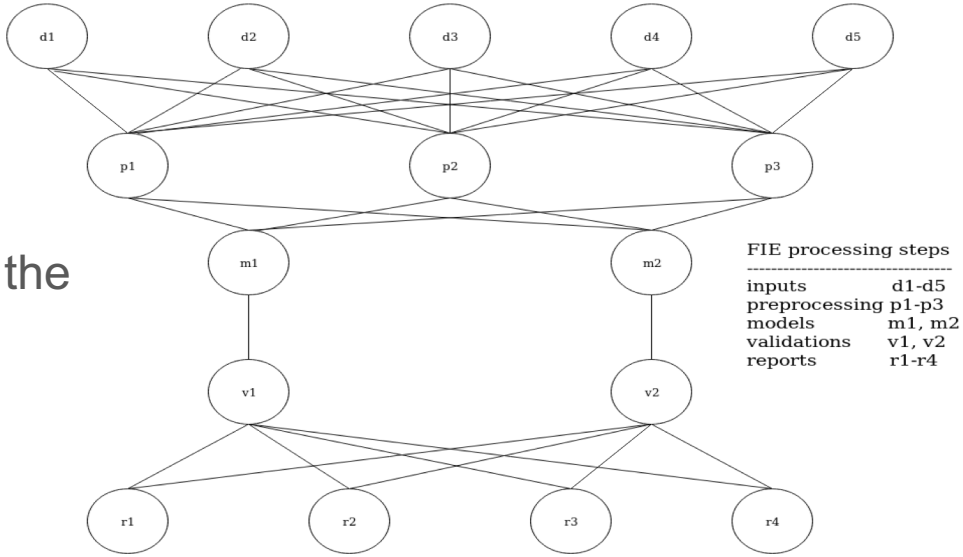
Packability Applied

Our workflow another way -- Each circle represents an input or processing step

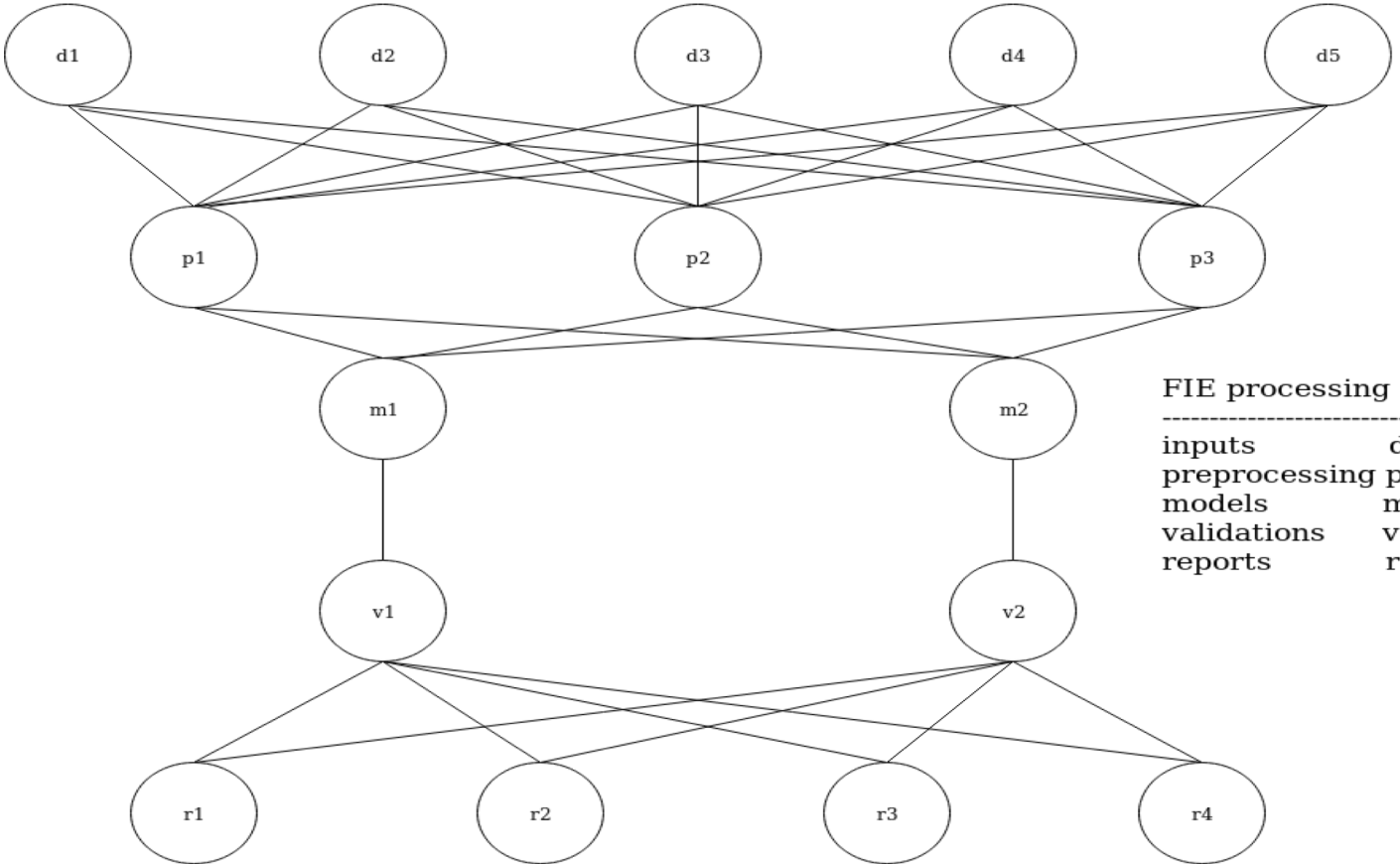


There are many possible combinations of groupings

- One container contains all
- A container for each step
- Or perhaps a container for the top two rows, a left branch container, a right branch container, and then a container to merge and produce results.



There are many possible combinations of groupings



FIE processing steps

inputs	d1-d5
preprocessing	p1-p3
models	m1, m2
validations	v1, v2
reports	r1-r4

Picking the “right” set of groupings

- We don’t know of any objective measure to evaluate grouping granularity.
- Given any particular set of groupings there will be a shortcoming -- less ability to reuse parts, increased risk of portability failure, etc.
- Grouping according to likelihood of reuse of the group seems a good rule of thumb. A tradeoff between transparency/reusability and portability.
- Could be a time consuming task for an author.

Sciunit and Popper

Browse: http://sciunit.run

Contribute: https://bitbucket.org/geotrust/sciunit-cli

Issues: pr@sciunit.run

National Science Foundation grants ICER-1639759,

ICER-1661918, ICER-1440327, ICER-1343816

POPPER: http://falsifiable.us/

NSF: 1450488

Sciunit-Popper: https://github.com/popperized/sciPopper