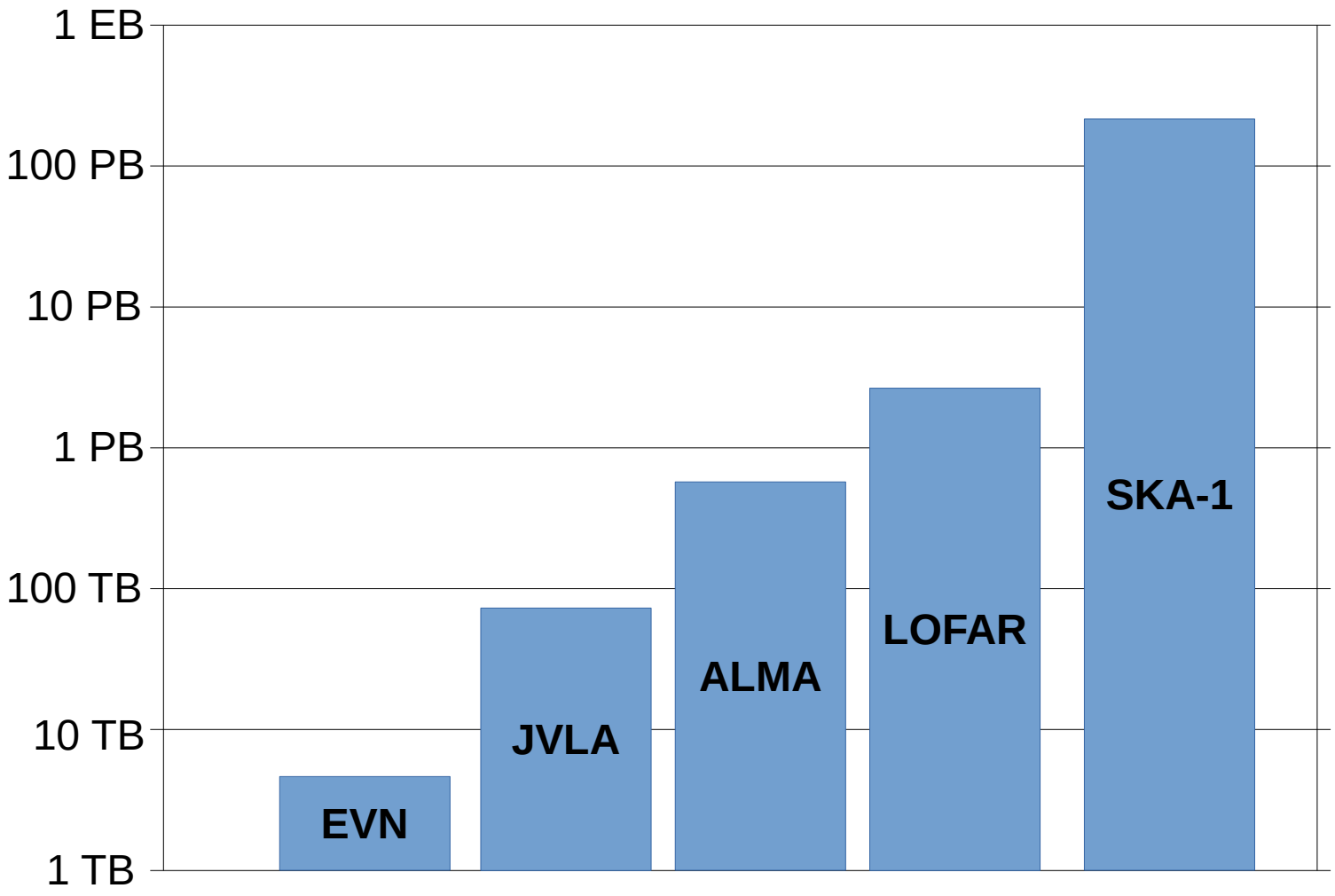
Efficient remote interactive pipelines using CASA and Jupyter

Aard Keimpema *(keimpema@jive.eu)*

JIVE
Joint Institute for VLBI ERIC

H2020-Astronomy ESFRI and Research Infrastructure Cluster (Grant Agreement number: 653477).

Yearly archivable data



Remote pipelines

- Data reduction is done where the data is stored
- Data reduction is based on pipelines
- Remote interactive pipelines based on Jupyter notebooks
- Major usability enhancement, increased accountability
- Successor to IPython, CASA is based on IPython
- Not limited to python, bindings to 40+ languages exist

vla-cont-demo - Mozilla Firefox

vla-cont-demo

jupyter.jive.nl/user/j4wycZFsiIeb/notebooks/vla-cont-demo.ipynb

jupyter vla-cont-demo (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted Casa

Run Code

The notebook interface to CASA was developed by Aard Keimpema at the Joint Institute for VLBI ERIC (JIVE). Installation in the sandbox interface (based on tmpnb) was done by Tammo Jan Dijkema at ASTRON. This work is a part of the ASTERICS project, Astronomy ESFRI & Research Infrastructure Cluster ASTERICS - 653477.

This notebook interface was built on top of CASA 5.3.0. Adaptions were made to run it on a newer version of Python than the one distributed with CASA. Future development of this notebook, or upgrading the underlying CASA version, is currently being planned.

In [1]:
```
casa['build']['version']
```

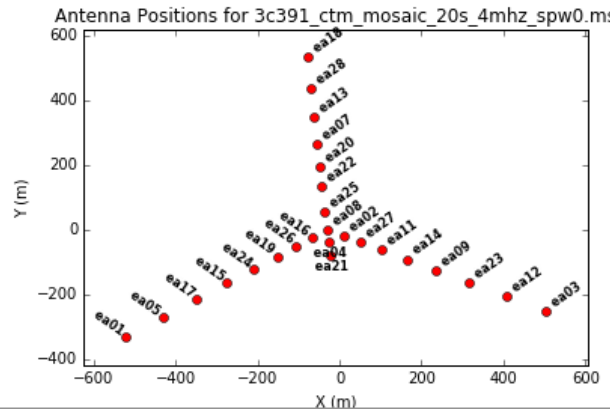
Out[1]:
```
'5.3.0-143  '
```

http://jupyter.jive.nl

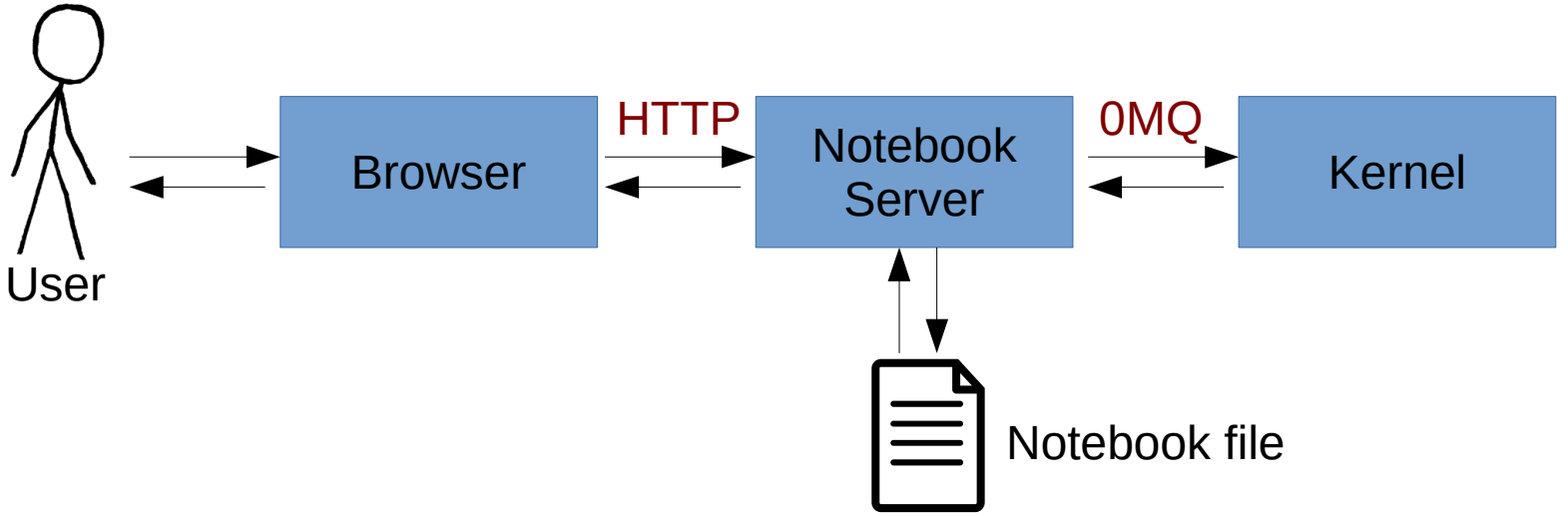
To get a sense of the array, as well as identify an antenna for later use in calibration, we use the task **plotants**.

In [3]:
```
plotants(vis='3c391_ctm_mosaic_20s_4mhz_spw0.ms',
         figfile='plotants_3c391_antenna_layout.png')
clearstat()  # This removes the table lock generated by plotants in script mode
```

```
Number of points being plotted: 26
```

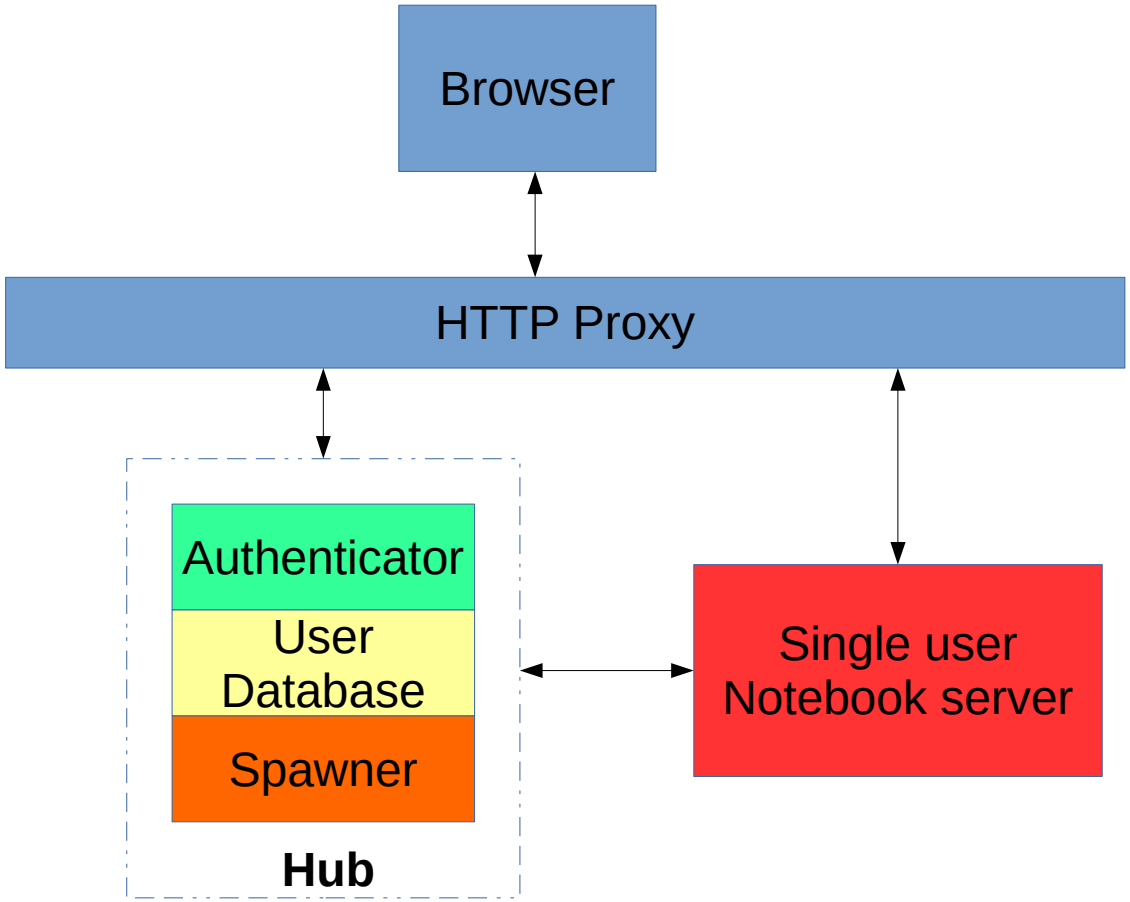


Jupyter Architecture



- Notebook server knows nothing about target language
- All language specifics are in the kernel
  - *Wrapper-kernel*: written in python, easiest to implement
  - *Native-kernel*: written in target language, much more work

Jupyter hub

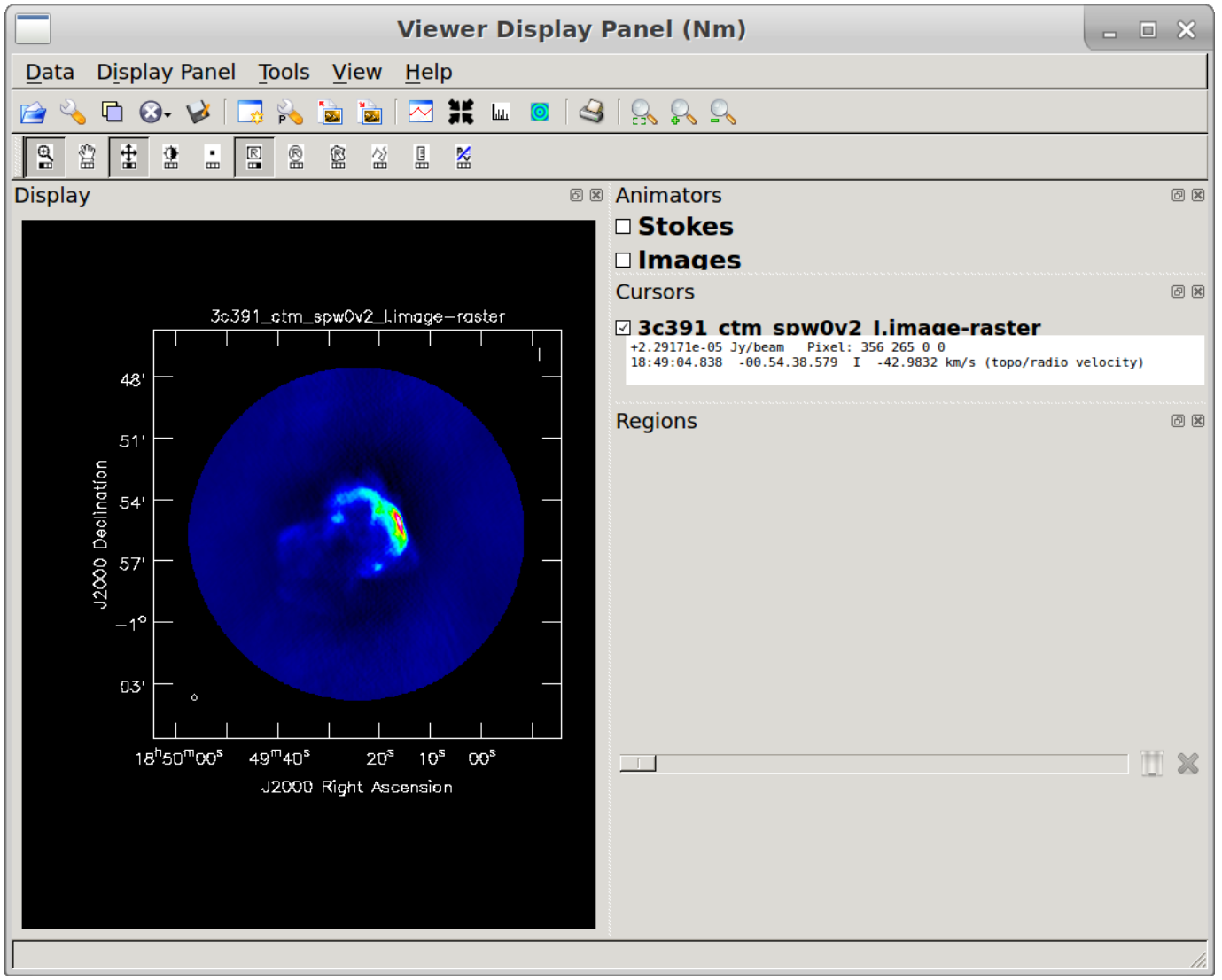


Multi-user front-end for Jupyter

CASA Jupyter kernel

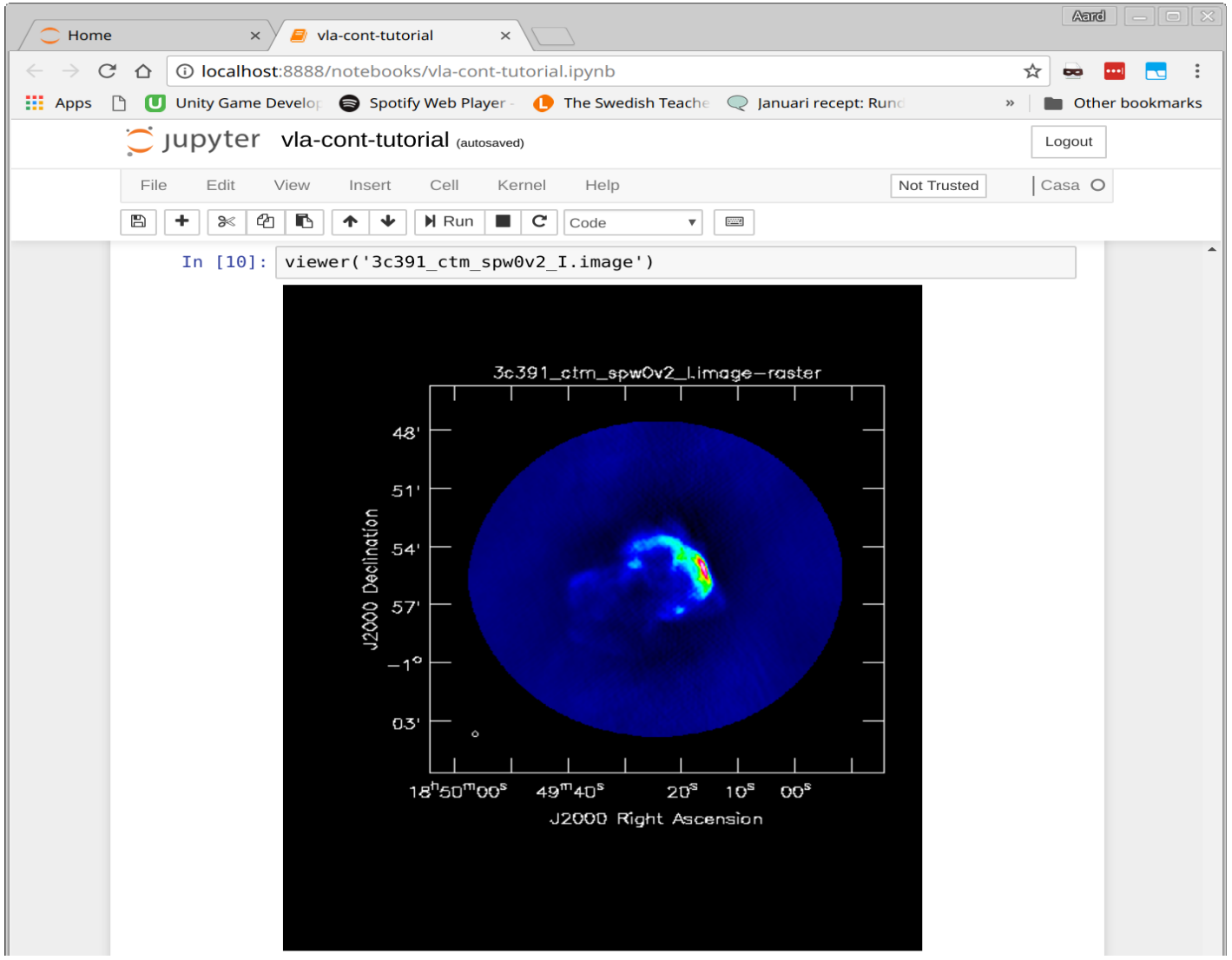
- Based on the generic python wrapper kernel
- Initialization:
  - Load needed python packages: casacore, casa tasks, matplotlib, ….
  - Setup environment: Config, logging, dbus, etc..
- CASA has python bindings for all tasks
- Many tasks open a C++ coded GUI, these are wrapped so that output goes to notebook.
- *https://github.com/aardk/jupyter-casa*

Example: casaviewer



viewer('3c391_ctm_spw0v2_I.image')

Example: casaviewer

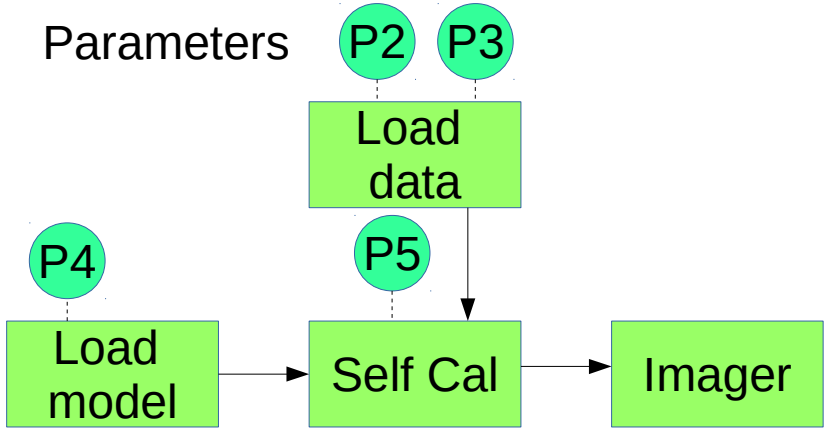


viewer('3c391_ctm_spw0v2_I.image', **gui = False,**
outformat = 'png', outfile = viewer_temp.png)

CASA for Jupyter

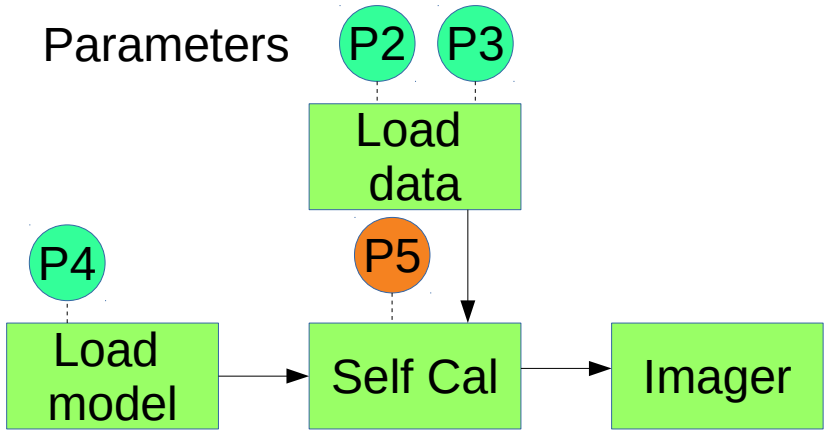
- NRAO CASA distribution is entirely self-contained, it is essentially a linux distribution
- Too old for Jupyter, many conflicting packages, e.g. Matplotlib, IPython, ….
- We created a custom build of CASA using based on latest Python
- Distributed as **Docker** and **Singularity** containers.
  - *docker pull penngwyn/jupytercasa*
  - *singularity pull shub://aardk/jupyter-casa*

Minimal Re-computation for Exploratory Data Analysis in Astronomy



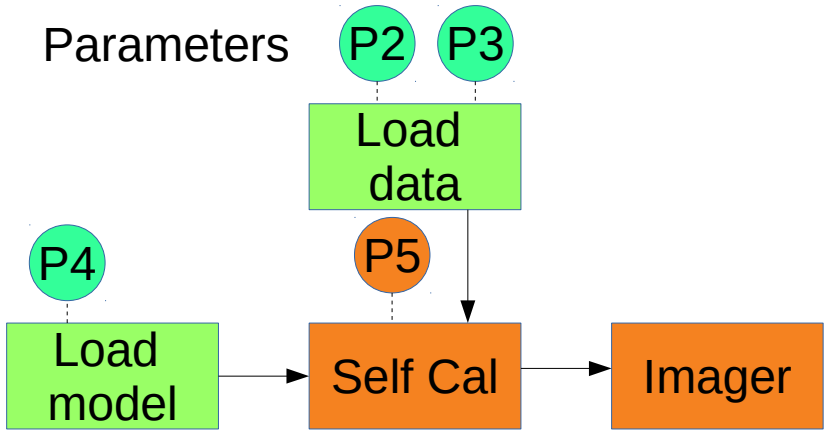
- Astronomy and Computing, **25**, 133, 2018 (arXiv:1711.06124)
  - U. Cambridge: Bojang Nikolic, JIVE: Des Small, and Mark Kettenis
- Generate dependency graphs
- Cache results efficiently using ZFS copy-on-write

Minimal Re-computation for Exploratory Data Analysis in Astronomy



- Astronomy and Computing, **25**, 133, 2018 (arXiv:1711.06124)
  - U. Cambridge: Bojang Nikolic, JIVE: Des Small, and Mark Kettenis
- Reliable re-computation of pipeline results
- Cache results efficiently using ZFS copy-on-write

Minimal Re-computation for Exploratory Data Analysis in Astronomy



- Astronomy and Computing, **25**, 133, 2018 (arXiv:1711.06124)
  - U. Cambridge: Bojang Nikolic, JIVE: Des Small, and Mark Kettenis
- Reliable re-computation of pipeline results
- Cache results efficiently using ZFS copy-on-write

Conclusions

- We have implemented Jupyter kernel for CASA suitable for pipelines
- We provide both Docker and Singularity images for easy deployment
- Minimal re-computation framework enables efficient reliable iterative pipelines.
- *https://github.com/aardk/jupyter-casa*
- *http://jupyter.jive.nl/*

Acknowledgement

- H2020-Astronomy ESFRI and Research Infrastructure Cluster (Grant Agreement number: 653477).