

Jupyter Notebooks in NAF

Short overview for NUC

(A lot of material shamelessly stolen from Todd Tannenbaum & Johannes Reppin !)

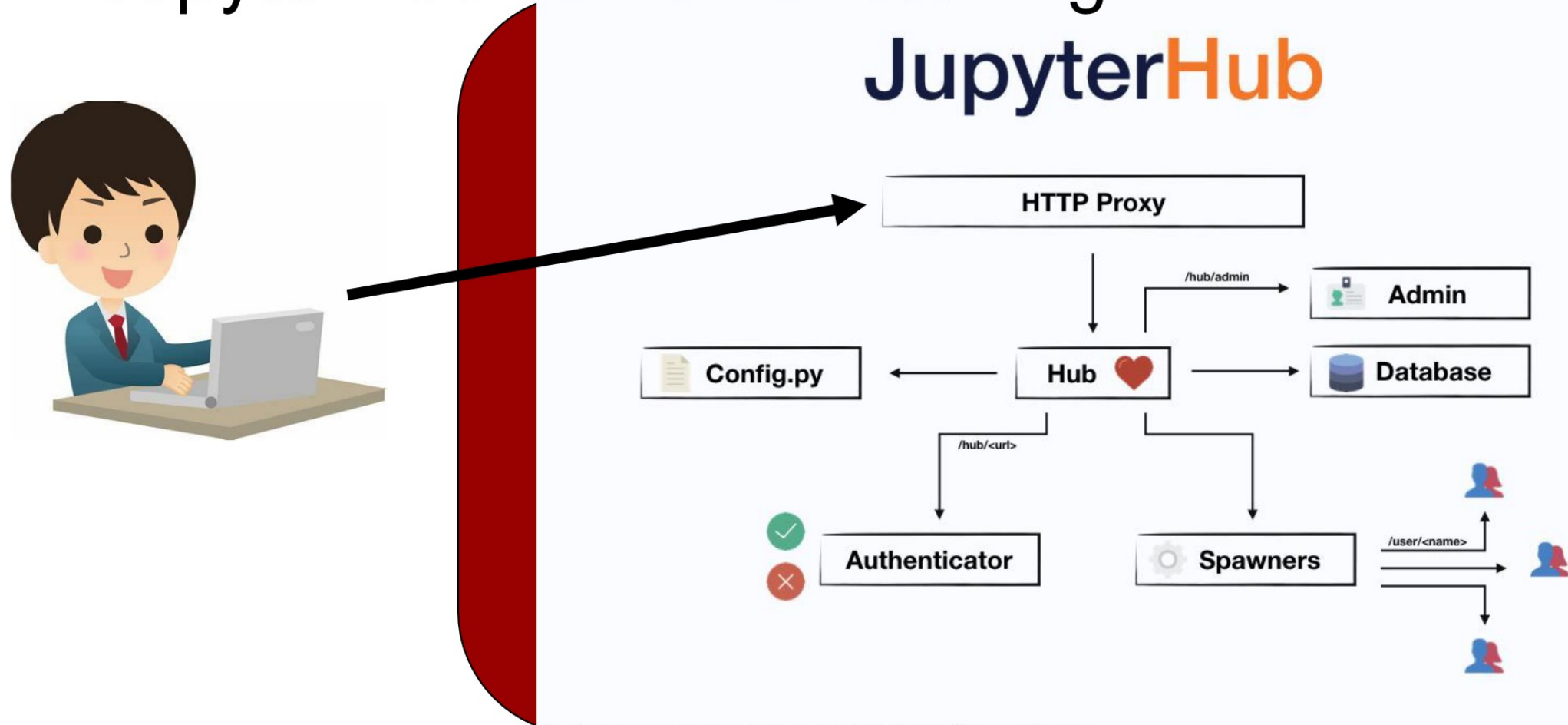
Beyer, Christoph
HH, 18-01-2021

Daily business

- SL6 decommissioning well under way, only very few remnants.
- Migration of old WGS hardware to VMs progressing slowly. Please report, if current setup not working ... and please report in a timely manner, so that we can investigate.
- Commissioning of end-of-2020-purchases: Delay due to working restrictions in the computing center. Currently making a plan, and prioritization. Will inform you about progress.
- Roughly 40 WN s for GRID and 20 WN s for NAF in the queue
- Parts of DUST will go out of warranty in Mai. Planning purchase. (same technology, different vendor)

Jupyter Notebooks – an interactive scientific environment

- › Point your browser at a URL where a JupyterHub server is listening



All icons were obtained on Flaticon (<https://www.flaticon.com/packs/essential-collection>)

Jupyter Notebooks

The image displays three overlapping Jupyter Notebook windows. The background window shows the 'Welcome to the Jupyter Notebook Server' page. The middle window shows a 'WARNING' message: 'Don't rely on this server'. The foreground window is titled 'Lorenz Differential Equations (autosaved)' and contains the following content:

Exploring the Lorenz System

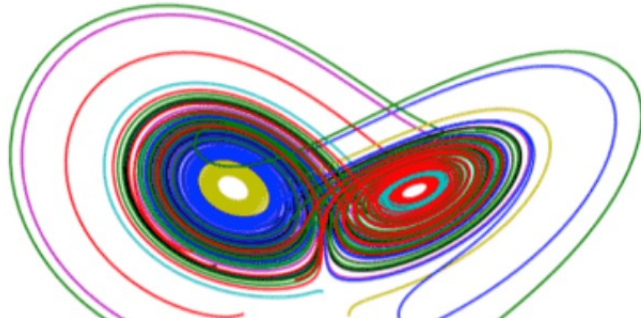
In this Notebook we explore the [Lorenz system](#) of differential equations:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

In [7]: `interact(Lorenz, N=fixed(10), angle=(0.,360.),
sigma=(0.0,50.0), beta=(0.,5), rho=(0.0,50.0))`

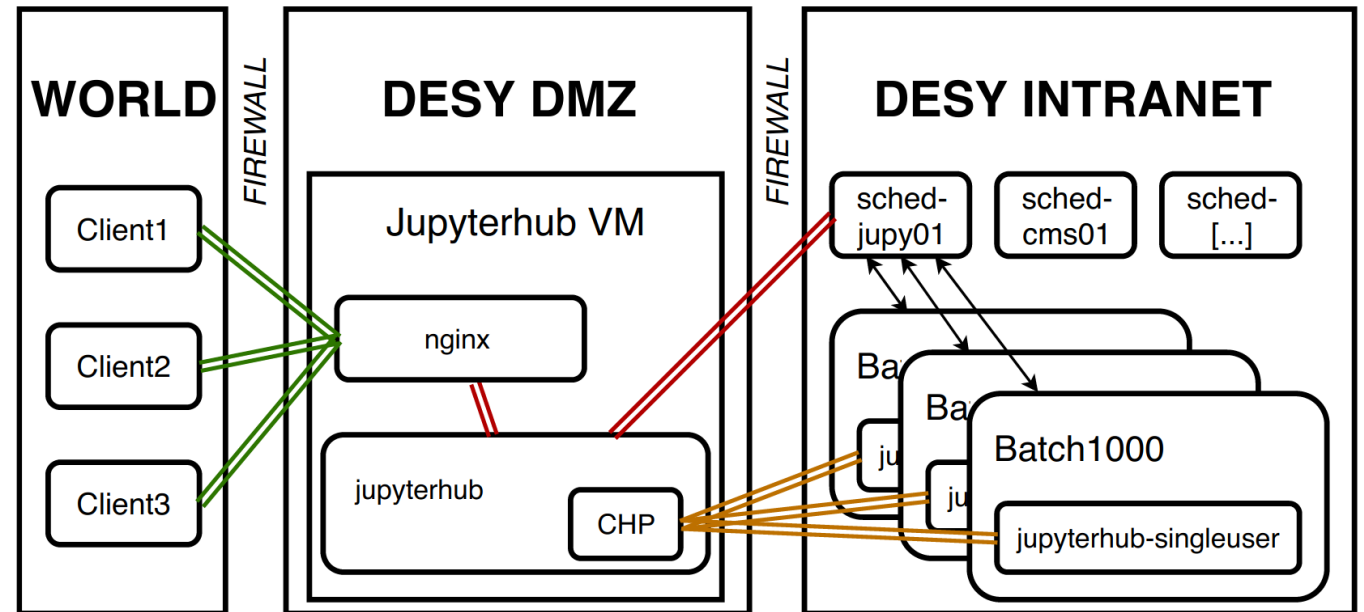
angle 308.2
max_time 12
 σ 10
 β 2.6
 ρ 28



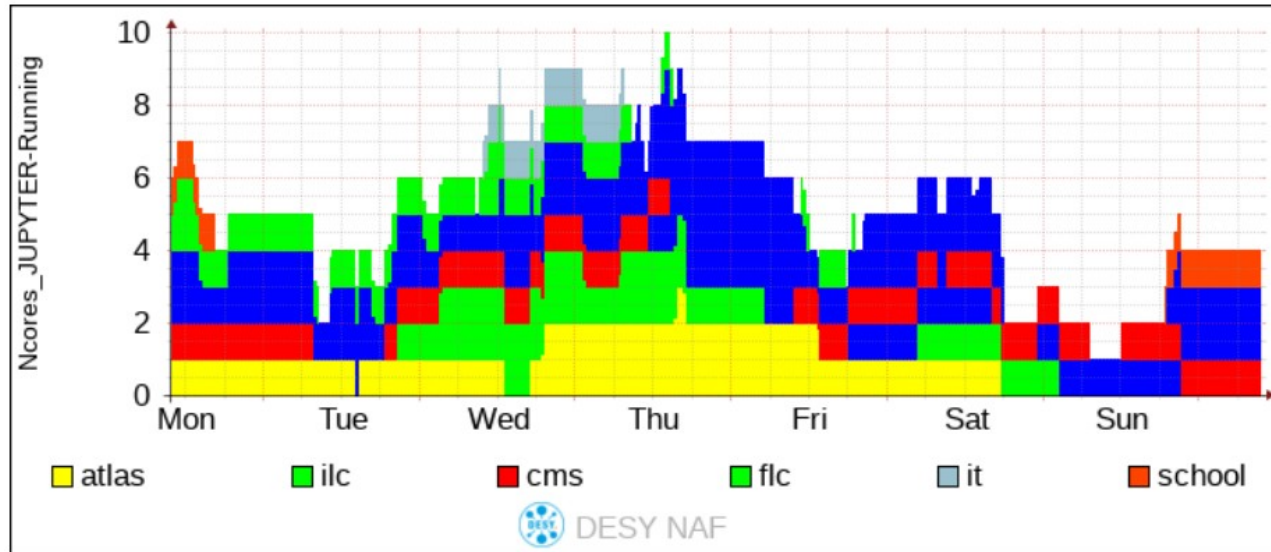
Current setup for the NAF

HTC computing with a notebook

- Access through webinterface worldwide (jupyterhub)
- Currently one slot reserved per host (= 300 slots)
- Cpus & memory used by other jobs when unclaimed
- Current slot setup: 1 core 1GB memory (3 GB memory usable)



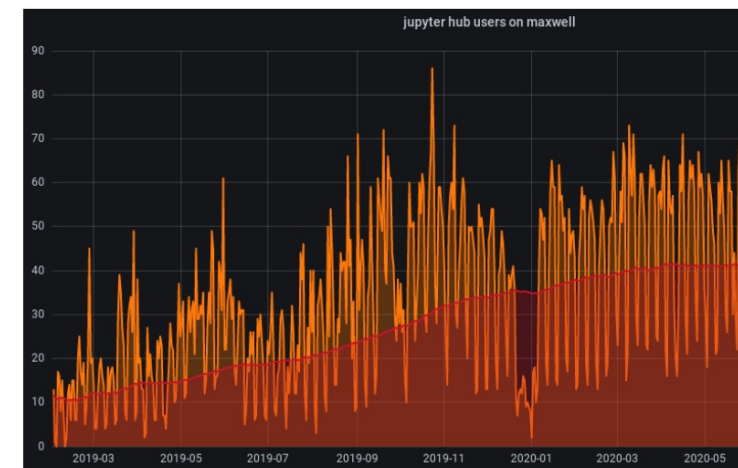
Python notebooks current usage



- Around 10 to 20 notebooks at a time
- Peak usage during CERN beamline for schools etc.
- Around 300 different users
- 1.500 accounts with NAF access

Notebook starts since summer last year by VO

belle	##### (2820)
atlas	##### (594)
cms	##### (1439)
flc	##### (562)
pier	# (181)
school	##### (475)
uni	## (227)



Using Python map (builtin function)

```
# Describe work
```

```
def double(x):  
    return 2 * x
```

```
# Do work
```

```
doubled = map(double, range(10))
```

```
# Use results!
```

```
print(list(doubled))
```

```
# [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

Limits/drawbacks

- What if this function takes minutes/hours ?
- What if you had hundreds or thousands of inputs ?
- Interactive behaviour = sitting in the waiting room

Using HtMap

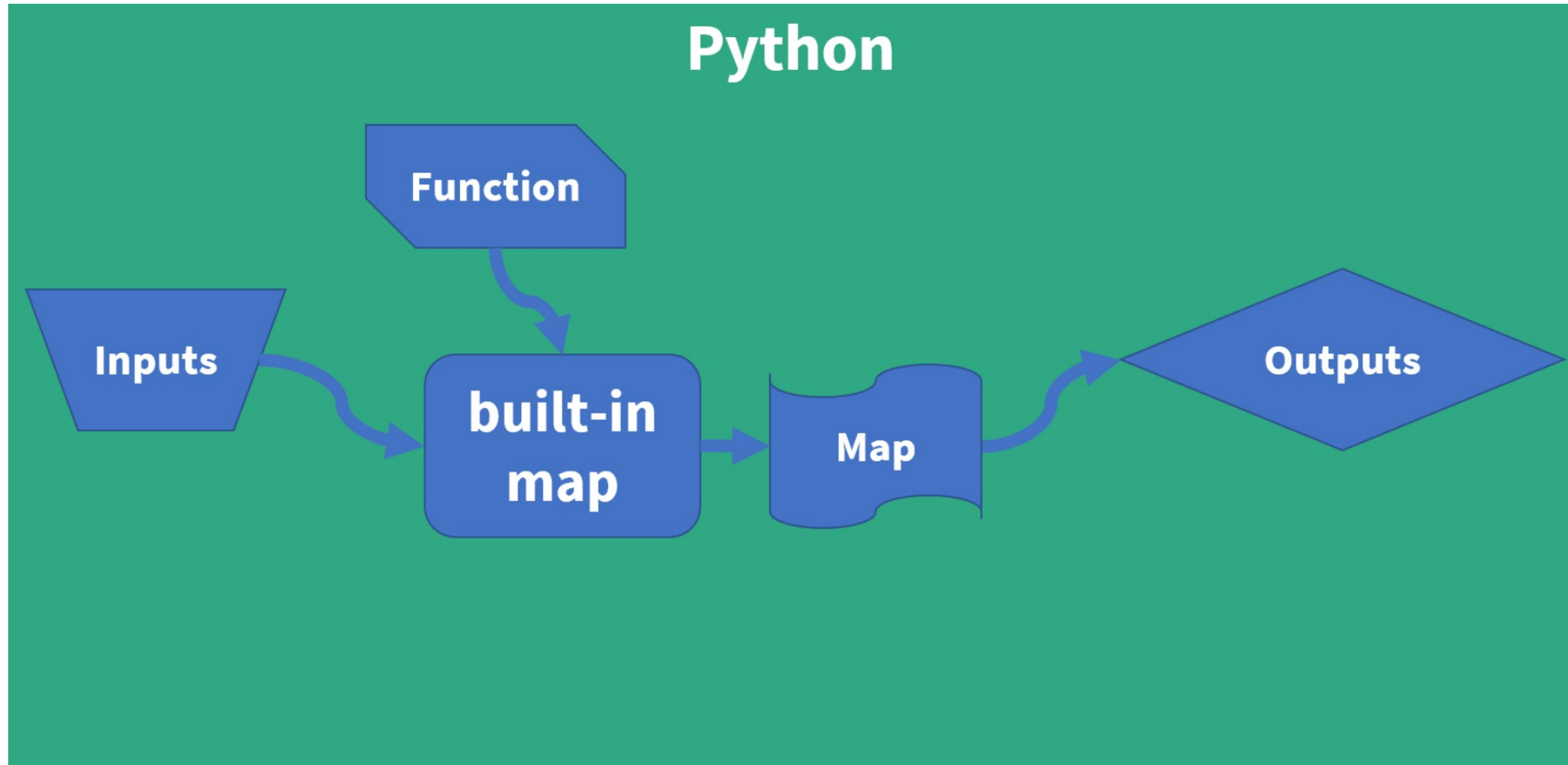
```
import htmap

# Describe work
def double(x):
    return 2 * x

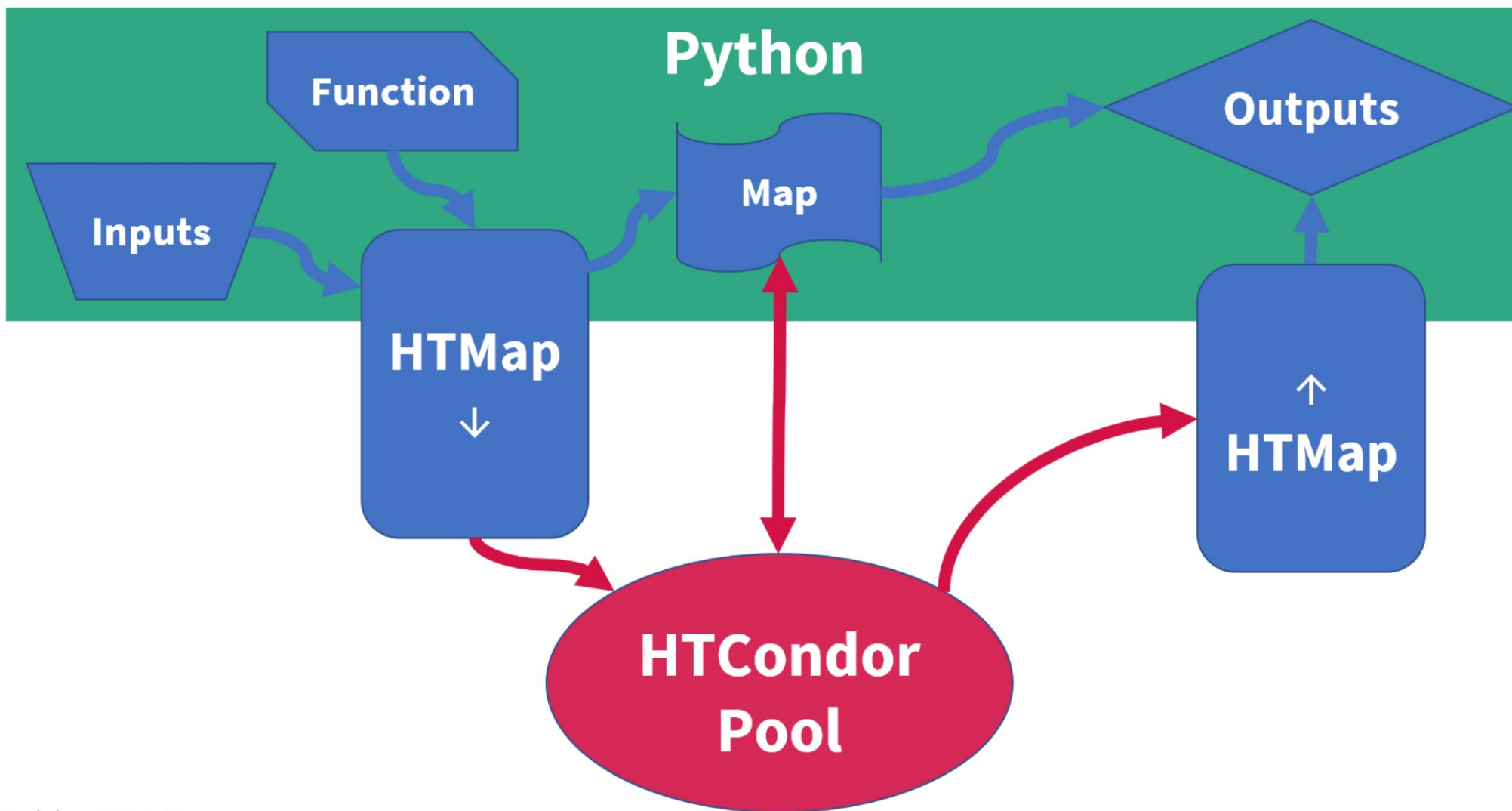
# Do work
doubled = htmap.map(double, range(10))

# Use results!
print(list(doubled)) # [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```


Python Map



HTMap



Conclusion

Notebooks with HTMAP

- Bring python based scientific computing to where the big boys play
- Notebook can be seen as a python based WGS
- Local resources shallow but access to compute farm **when needed and as long as needed**
- Minimum harm for others, as ressources dynamically used
- This even more important as Jupyter Notebook user literally are not even aware that you can 'turn off' a notebook !!!
- Following the spirit of HTC computing by implementation
- Of course: the 'high speed train' looks boring to some if you offer a 'ferrari' as an alternative at the same time ;)
- HTMap and Python Bindings will get to work without additional effort once the pool runs a higher version of Condor

Python bindings & outlook

Used by HTMap internally → Python Bindings

- Collector, sched and negotiator as Python objects
- Import htcondor; import classadd
- Can be used to submit jobs into the pool from the notebook

Outlook

- Both HTMap & Python Bindings working in the test cluster with KRB support
- No backporting and some things still to sort out due to different other changed behaviors
- Update strategy a bit clumsy at the moment
- Will be available before summer
- Users do see the need to break out of the notebook but nobody explicitly asked for HTMap so far
- Some tried to use python bindings and failed