



# A microservice-oriented science platform architecture

*Stefano Alberto Russo - INAF OATS*



# Introduction

## What is a Science Platform?

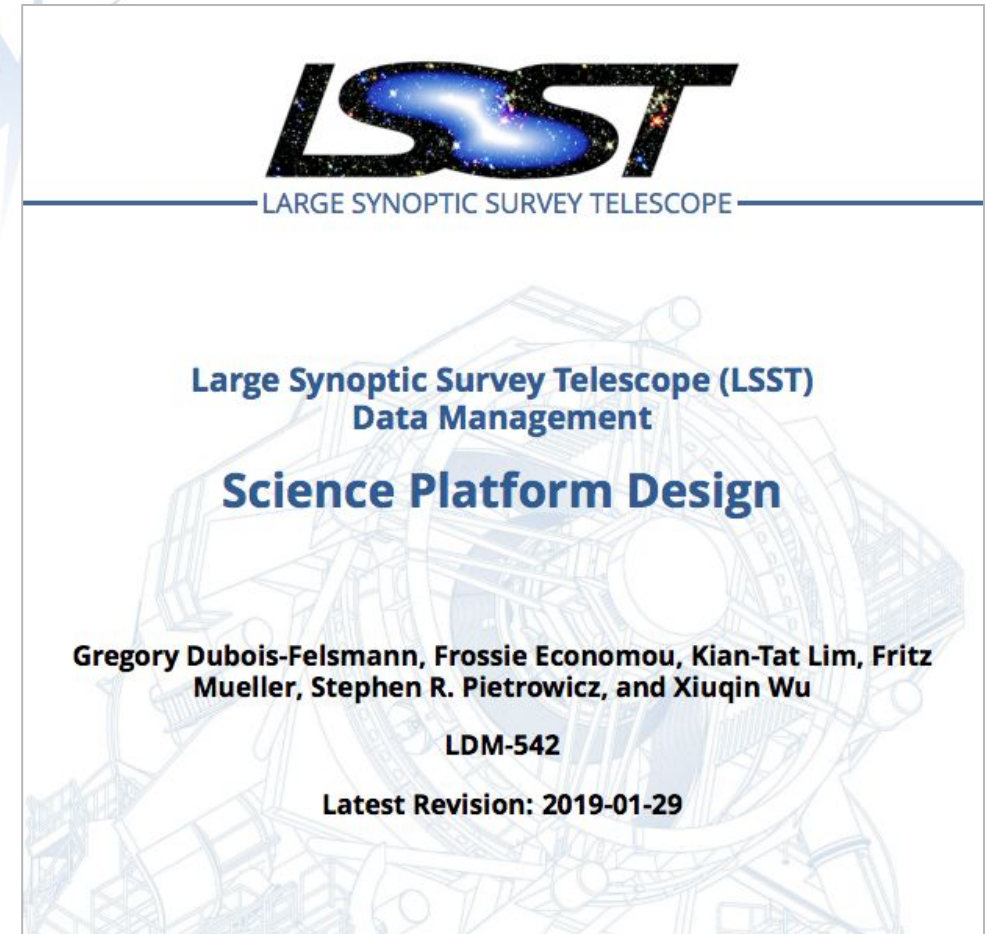
- An environment designed to offer users a smoother experience when interacting with data and computing resources  
→ *possibly bringing the user to the data and not vice-versa*
- We usually assume a web-based environment based on Jupyter notebooks or similar, and software containerisation.
- Other approaches or definitions may be also feasible (e.g. full interactive desktop access).



## LSST Science Platform definition

"a web Portal, designed to provide essential data access and visualization services through a simple-to-use website, a Notebook environment, that will provide a Jupyter Notebook-like interface, based on JupyterLab, enabling next-to-the-data analysis"

Juric et al. 2017





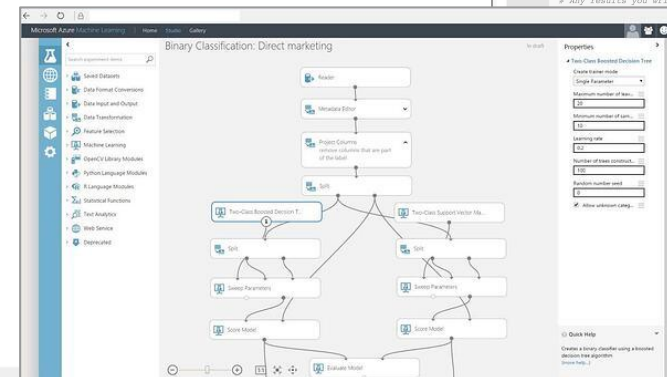
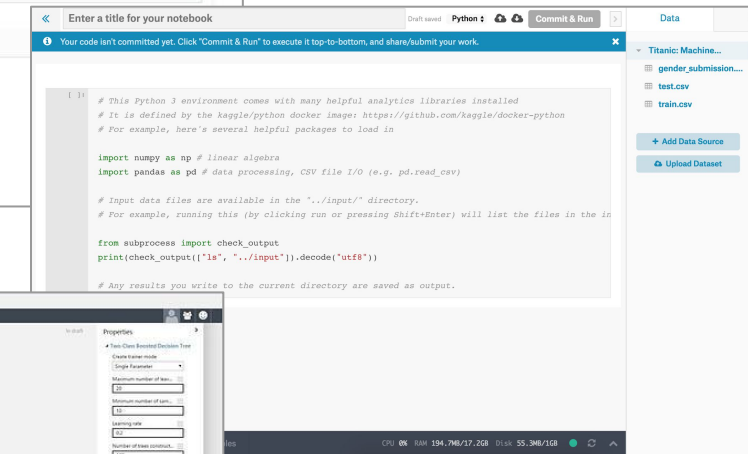
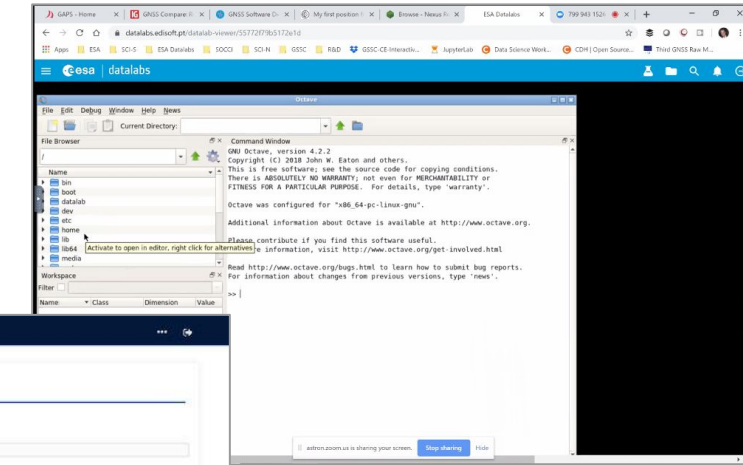
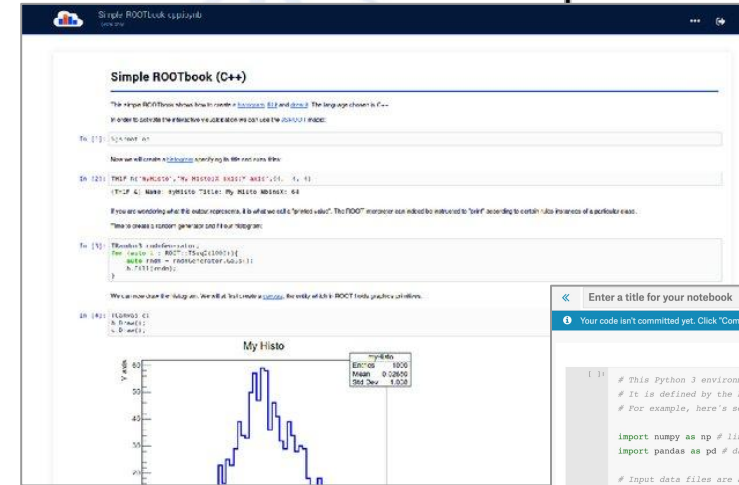
# Some examples

## Public Research/ Academia:

- CERN SWAN
- ESA Datalabs
- LSST Science Platform
- ...

## Private Research / Industry:

- Google Colabs
- Kaggle Kernels
- Azure Machine Learning Studio
- ...



# Limitations

**There are some strong limitations with the standard approach:**

1. A user cannot run in an environment not supported by the platform  
→ from a different Python version to a different Linux distribution.
2. A user cannot change the interface for using the Science Platform  
→ a web-based notebook (Jupyter or similar) interface makes it impossible to run native GUI applications (common in Astrophysics).
3. It is required to support all the (yet limited) softwares and versions  
→ from an administrative perspective this is a huge overhead



# Microservices

Microservices are independent and self-contained units that perform a given and well-defined task.

→ From just summing two numbers to running a neural network.

They are commonly used for the building blocks of the various Science Platforms cited before, in terms of “devops”.

*how about using them even for the user tasks?*



# Microservice interfaces

Microservices are interacted with using interfaces.

Examples:

- REST APIs
- HTTP
- SSH
- RPC



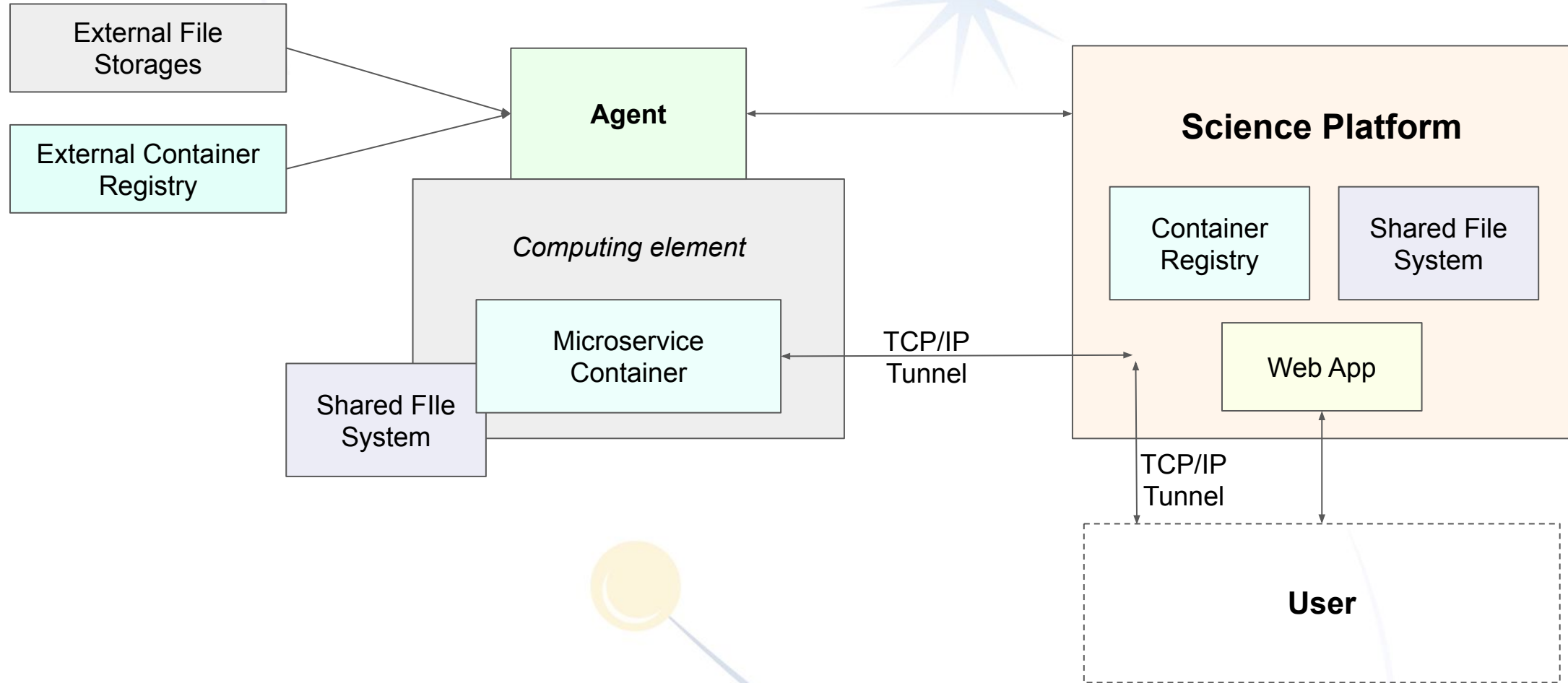
# Microservices and containers

- Microservices fit naturally in the containerisation approach:  
→ each microservice is a container.
- Once a microservice is containerised:
  - completely unaware about the underlying computing infrastructure
  - only the interface is relevant.
- We consider only TCP/IP based interfaces.

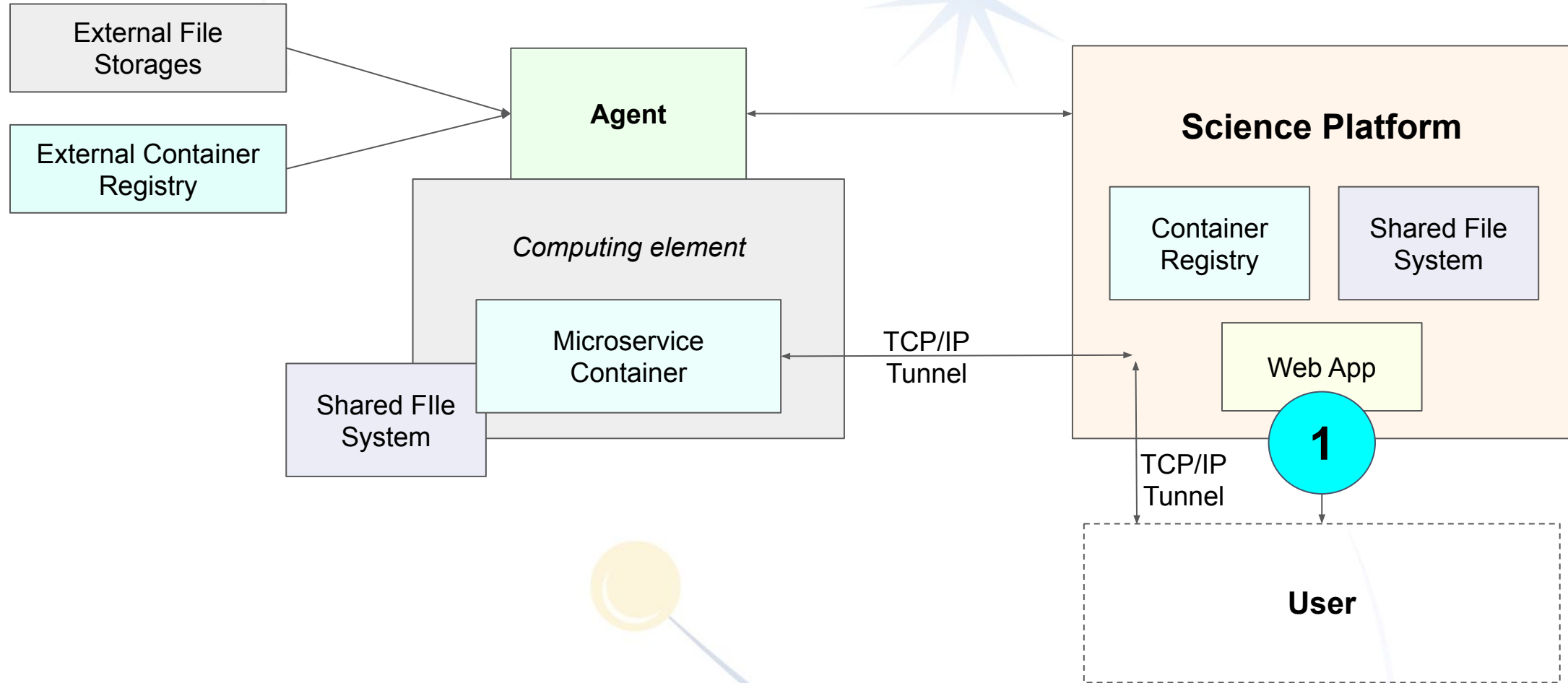




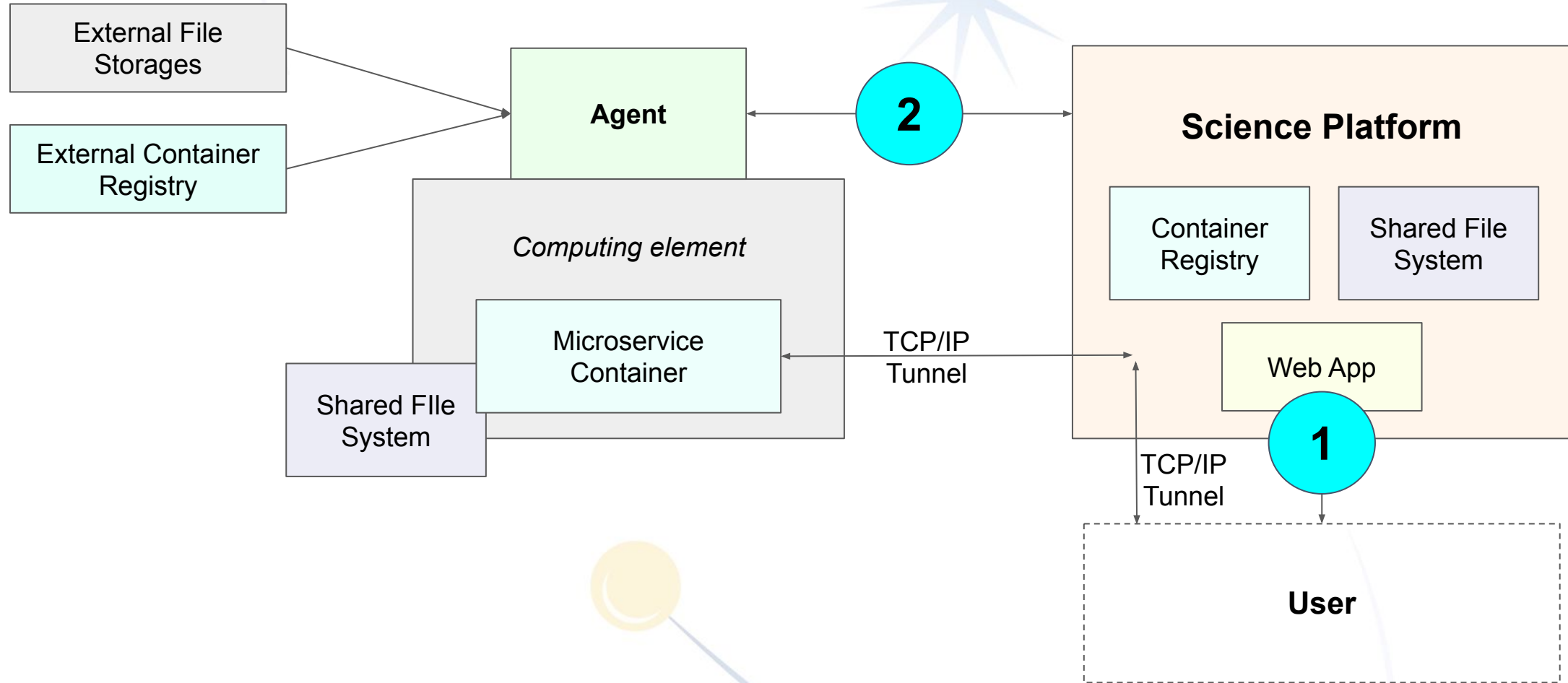
# A Science Platform architecture



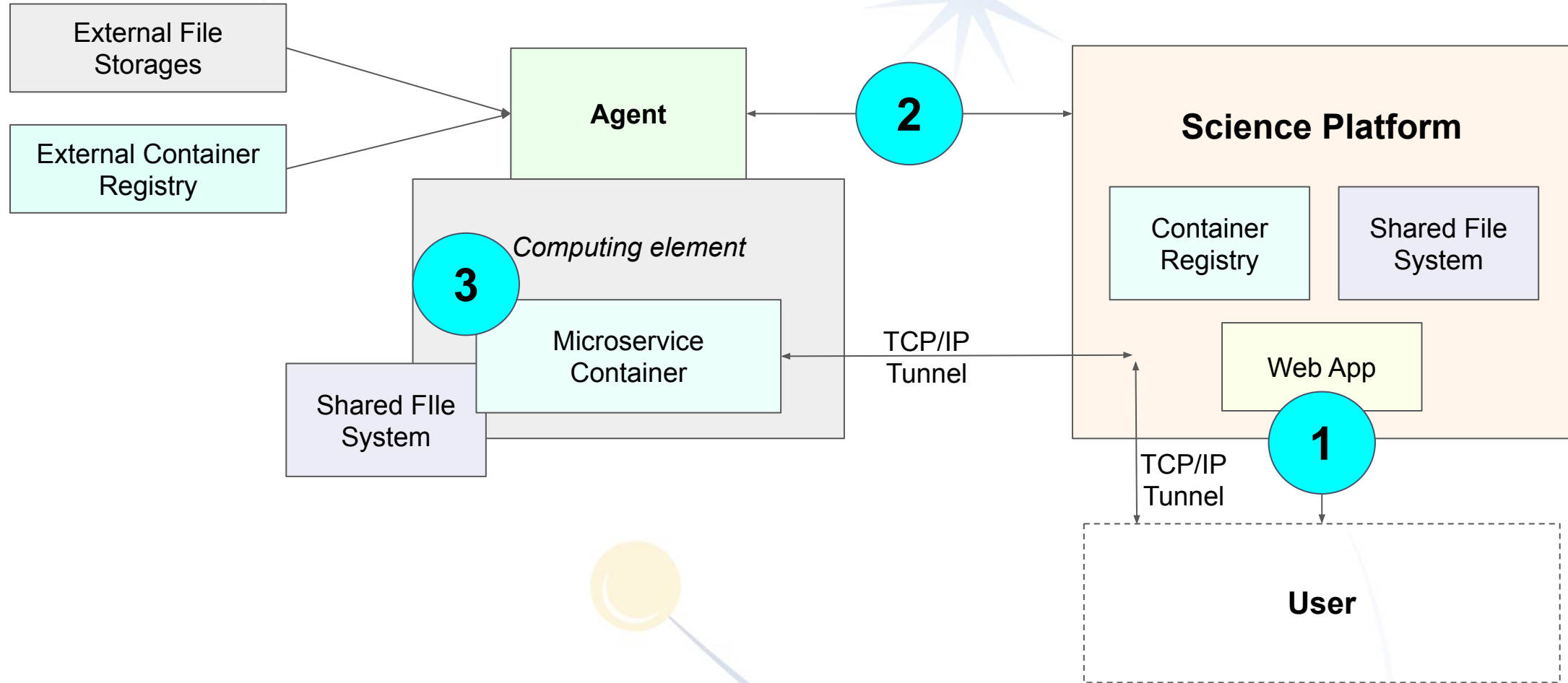
# A Science Platform architecture



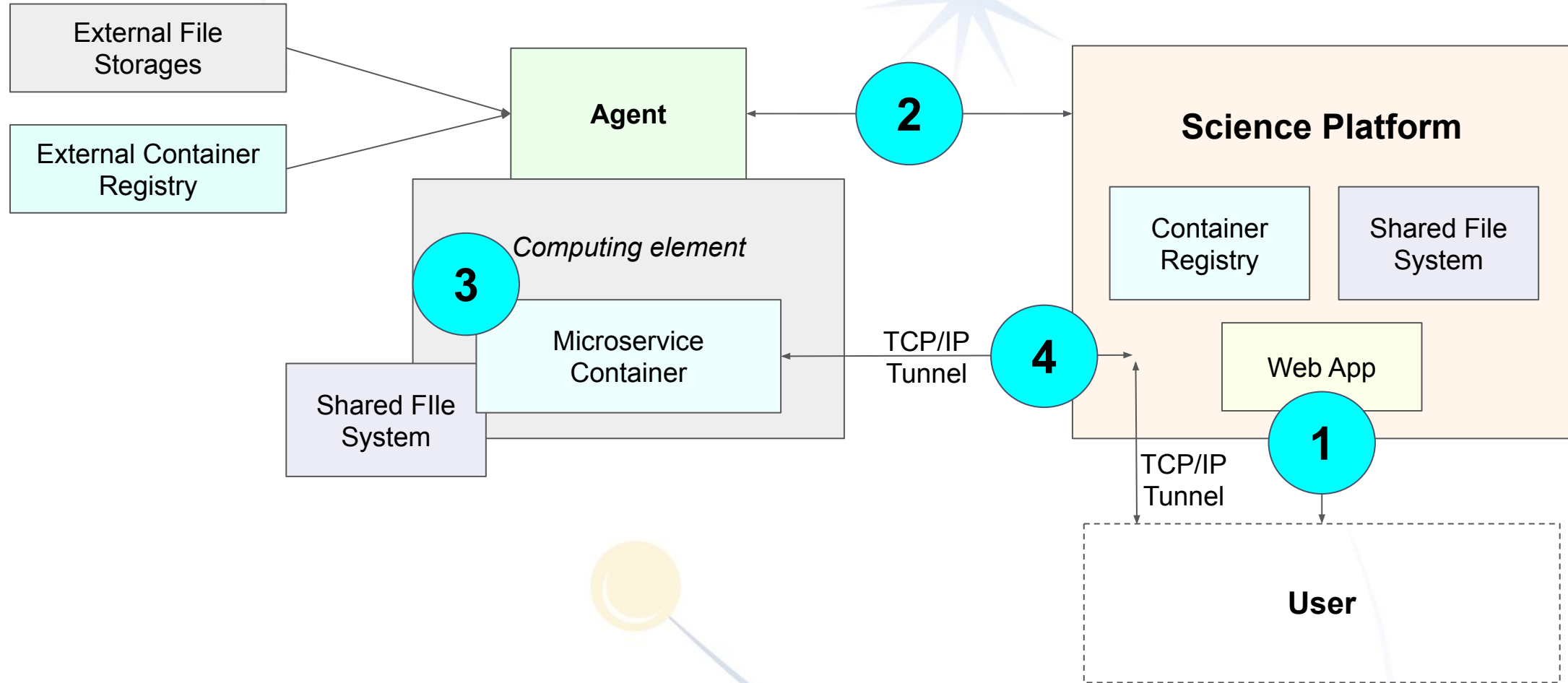
# A Science Platform architecture



# A Science Platform architecture

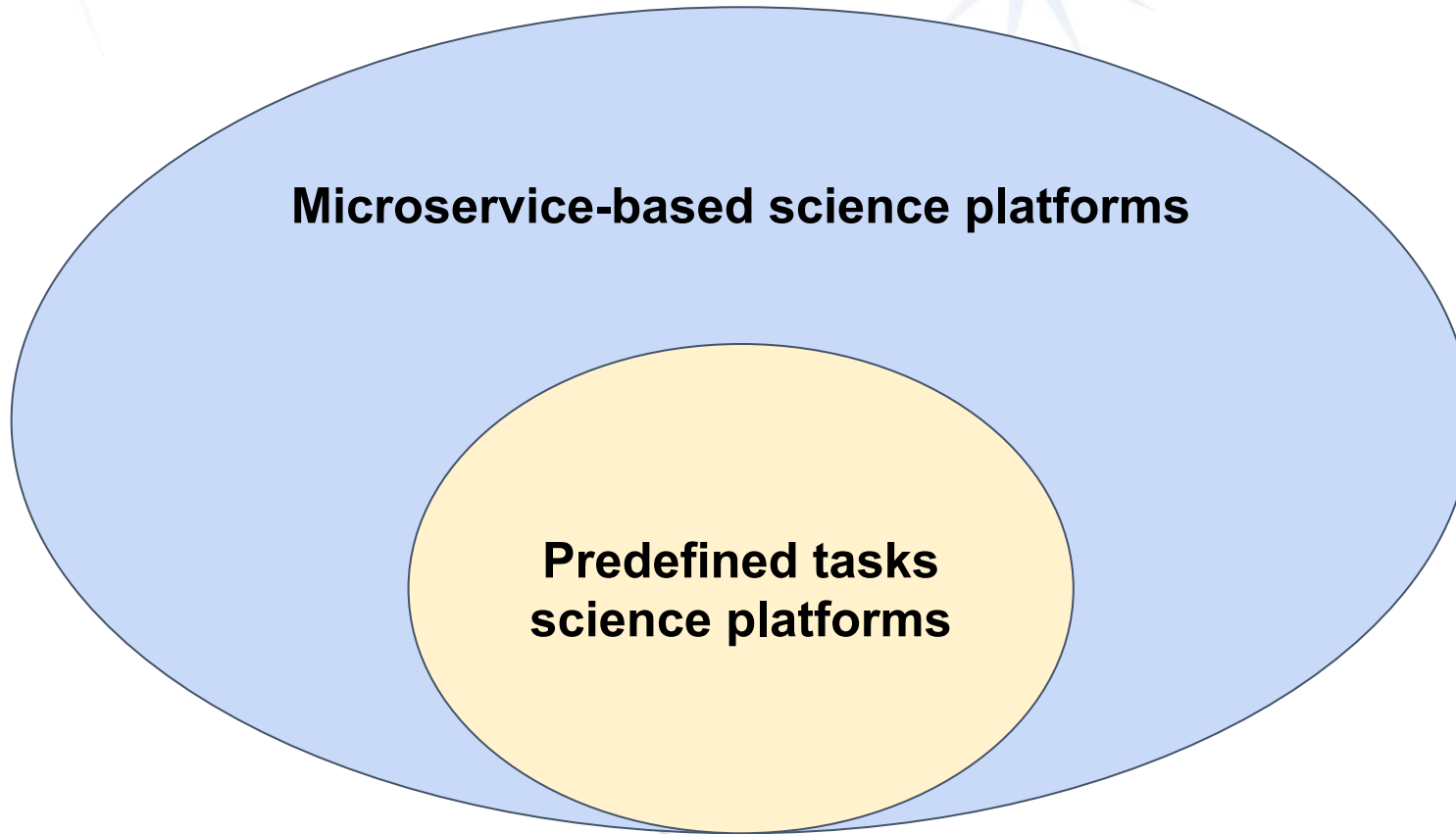


# A Science Platform architecture





# Capabilities of this approach



# Our prototype at INAF

rosetta.oats.inaf.it/tasks/?uuid=35eaa6e9-283b-4bf4-a26b-bd6d8c59a3fb

## Tasks / My Notebook

### Summary

Container: Jupyter Notebook

Computing: Amonra

Status: running

Stop Connect Logs

### Details

ID	35eaa6e9
Image	jupyter/base-notebook
Created at	Nov. 10, 2020, 11:18 a.m.
Extra binds	
Host	192.168.110.198
Port	8888
Tunnel port	7002
Direct link	<a href="https://rosetta.oats.inaf.it/t/35eaa6e9">https://rosetta.oats.inaf.it/t/35eaa6e9</a>

! You can share a direct link with other people, but remember that if you have no authentication in place anyone will be able to access.

rosetta.oats.inaf.it:7002/notebooks/Untitled.ipynb?kernel\_name=python3

jupyter Untitled Last Checkpoint: 2 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help Trusted Python 3

```

In [1]: print("hello world!")
        hello world!

In [ ]:
  
```



# Our prototype at INAF

rosetta.oats.inaf.it/tasks/?uuid=d7da42e3-f6f7-4488-a369-401abb5e4037

## Tasks / Astrocook

### Summary

Container: [Astrocook Desktop](#)

Computing: [Amonra](#)

Status: running

[Stop](#)
[Connect](#)
[Logs](#)

### Details

**ID** d7da42e3

**Image** [sarusso/astrocookdesktop](#)

**Created at** Nov. 10, 2020, 11:19 a.m.

**Extra binds**

**Host** 192.168.110.198

**Port** 51530

**Tunnel port** 7003

**Auth pass** \*\*\*\*\*

**Direct link** <https://rosetta.oats.inaf.it/t/d7da42e3>

! You can share a direct link with other people, but remember that if you have no authentication in place anyone will be able to access.

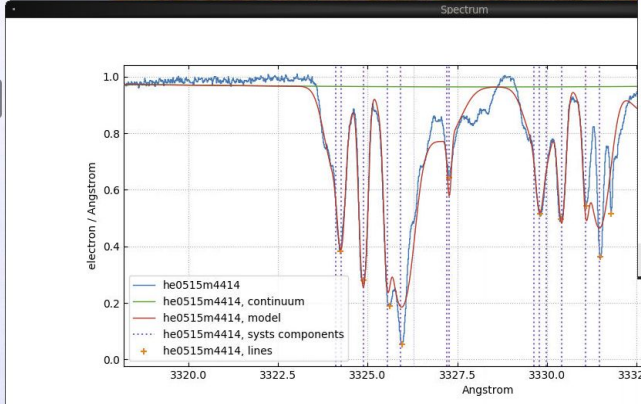
rosetta.oats.inaf.it:7003/vnc.html?autoconnect=true&resize=remote

Not Secure

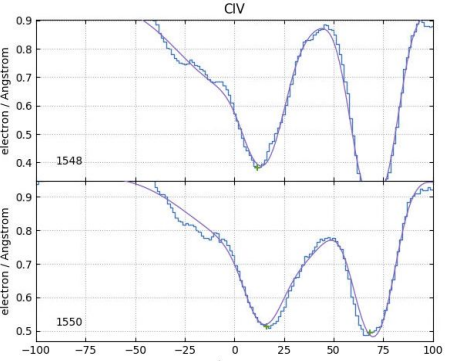
Sessions

name	object	active range	# rows	# nodes	# lines
he0515m4414 (0)	he0515m4414	[3043.78, 10428.92] Angstr...	283993		
he0515m4414 (1)	he0515m4414	[3043.78, 10428.92] Angstr...	283993	244	2767
he0515m4414 (2)	he0515m4414	[3100.01, 3399.99] Angstrom	21302	19	66

Spectrum



Spectrum detail



System table

	func	series	None	z0	z	dz	logN	dlogN	b
8	None	None	None	1.14913	1.14907	0.00002	13.05108	0.02885	50.000
9	voigt	CIV	1.15079	1.15087	0.00001	12.68910	0.01434	26.773	
10	voigt	CIV	1.16397	1.16399	0.00000	12.21572	0.52056	3.2866	
11	voigt	unknown	328.40393	328.40383	0.00057	13.70385	0.05374	29.588	
12	voigt	CIV	1.11415	1.11408	0.00004	10.75720	1.73742	1.8837	
13	voigt	CIV	1.13165	1.13174	0.00081	12.41381	4.89803	15.627	
14	voigt	CIV	1.13165	1.13165	0.00014	12.30107	7.10497	10.340	
15	voigt	CIV	1.13165	1.13155	0.00036	11.87266	4.67485	8.8763	
16	voigt	CIV	1.14715	1.14717	0.00000	12.73129	0.02667	12.044	
17	voigt	unknown	332.62758	332.62748	0.01073	12.51818	0.11857	50.000	
18	voigt	unknown	333.28885	333.28875	0.00033	12.09315	0.02498	6.4187	
19	voigt	CIV	1.14912	1.14912	0.00000	12.36822	0.07595	6.7335	



# Wrapping up

- Capabilities of this approach are a super-set of the standard approach
- Users can run whatever software they want  
→ *remote desktops, web applications, even databases...*
- Containers make it safe to run custom software
- Huge improvements in terms of reproducibility  
→ *users have control over what container version to run*



# THANKS!

Questions?

[stefano.russo@inaf.it](mailto:stefano.russo@inaf.it)

