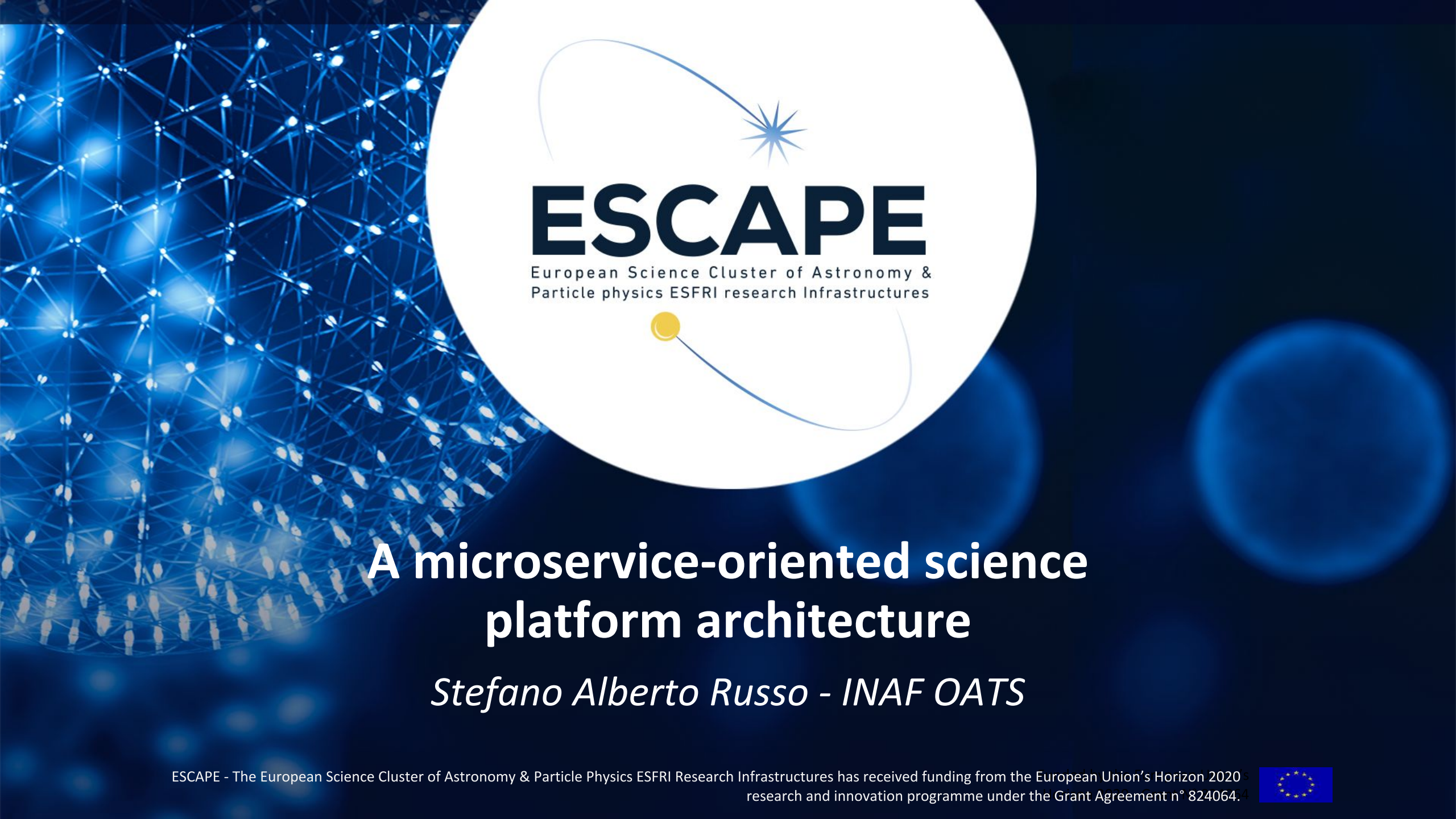


A microservice-oriented science platform architecture

Stefano Alberto Russo - INAF OATS

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

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.

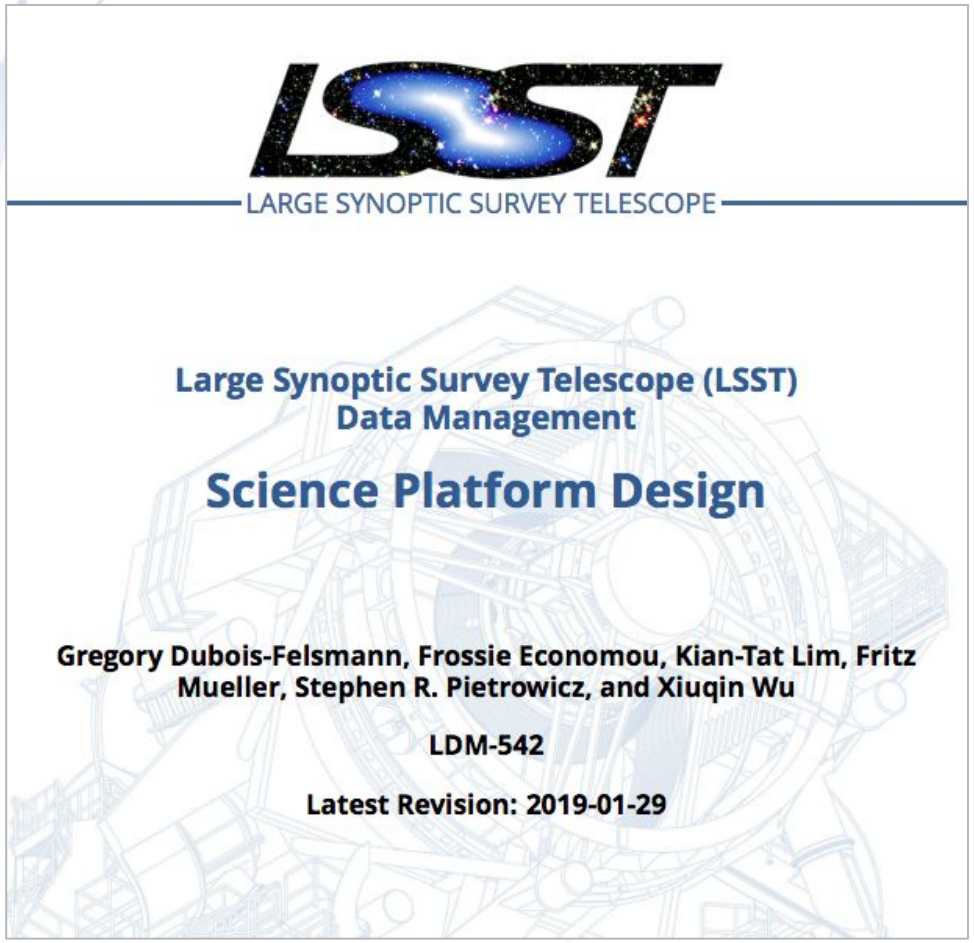
- Other approaches or definitions may be also feasible (e.g. full interactive desktop access).

Funded by the European Union's Horizon 2020 - Grant N° 824064

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



Funded by the European Union’s Horizon 2020 - Grant N° 824064

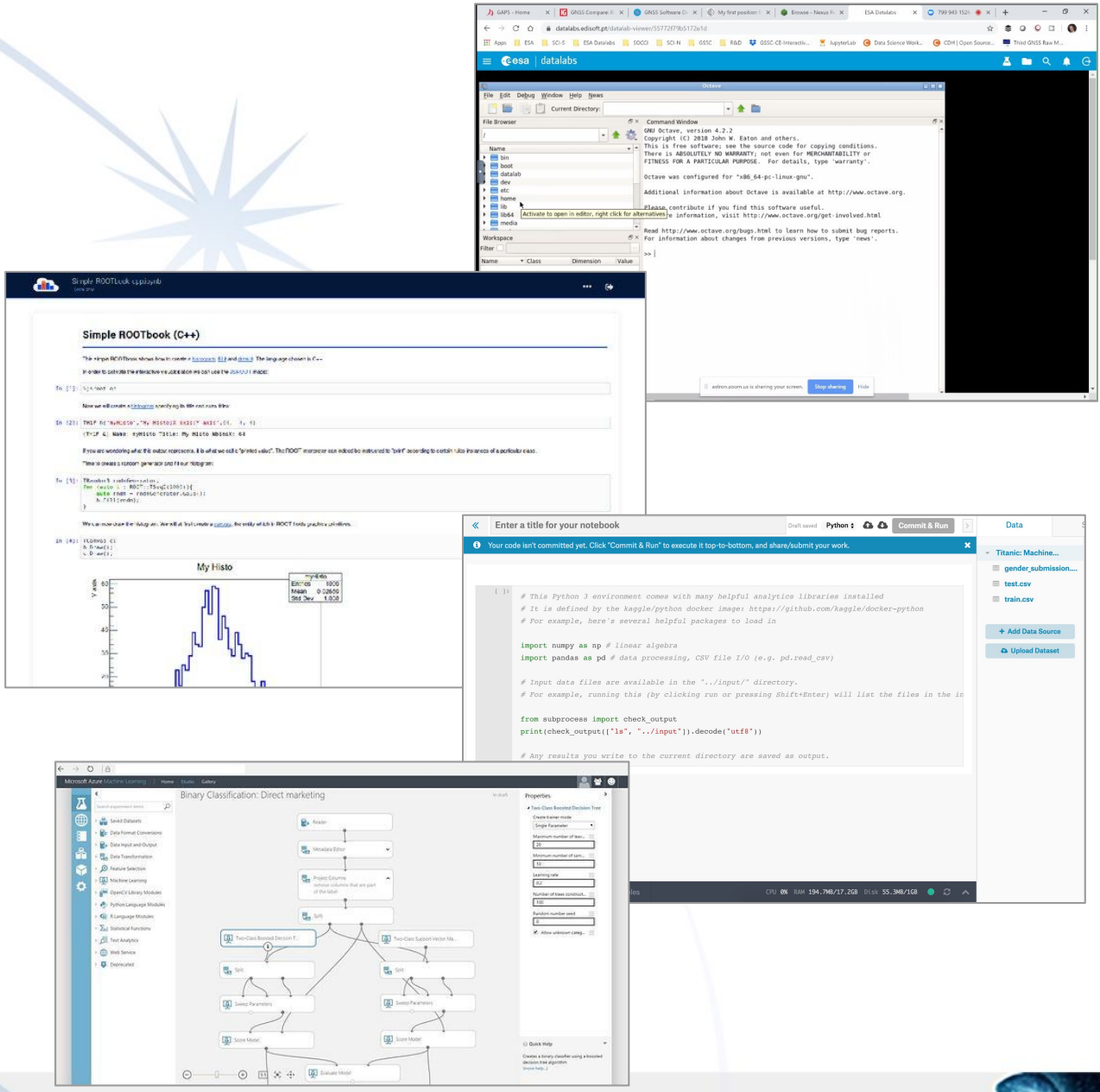
Some examples

Public Research/ Academia:

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

Private Research / Industry:

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



Funded by the European Union's Horizon 2020 - Grant N° 824064

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

Funded by the European Union's Horizon 2020 - Grant N° 824064

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?

Funded by the European Union’s Horizon 2020 - Grant N° 824064

Microservice interfaces

Microservices are interacted with using interfaces.

Examples:

- REST APIs
- HTTP
- SSH
- RPC

Funded by the European Union's
Horizon 2020 - Grant N° 824064

Microservices and containers

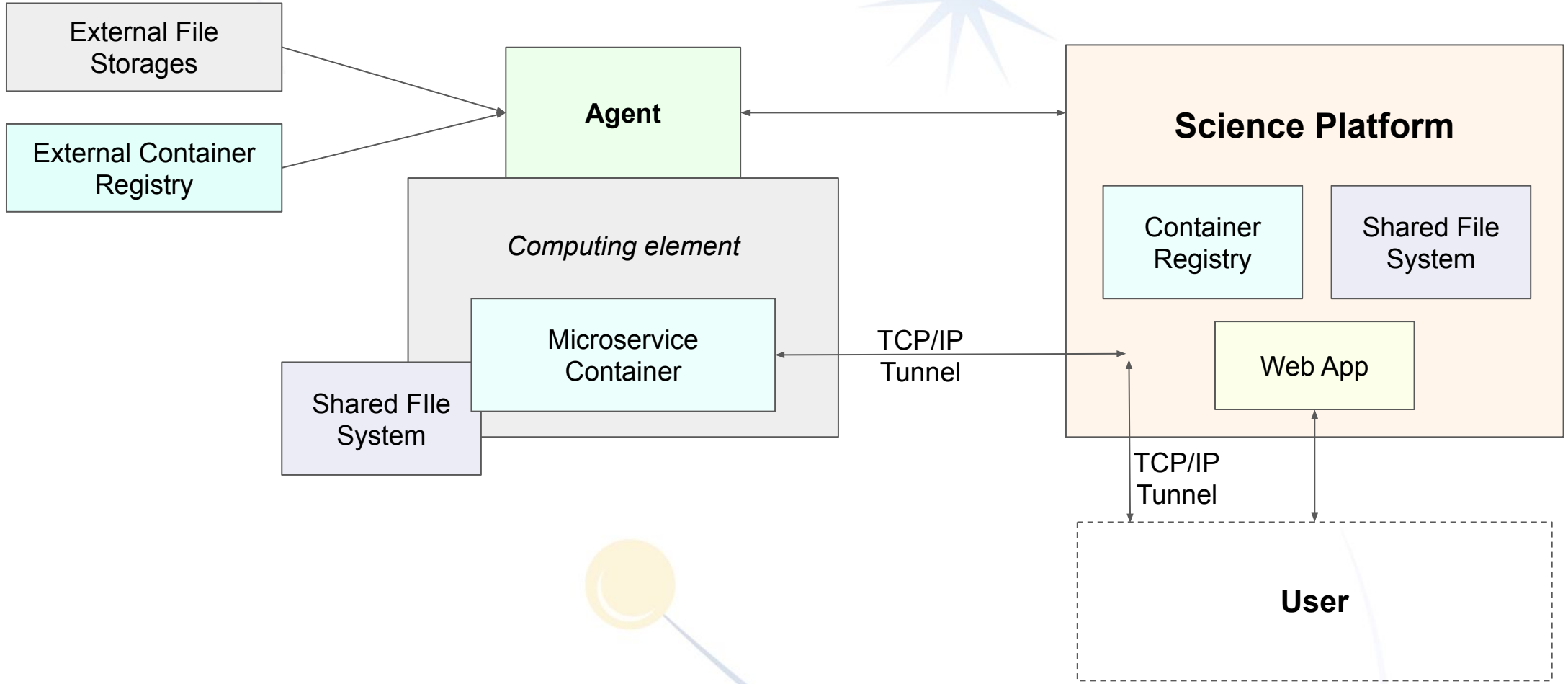
- Microservices fit naturally in the containerisation approach:
  → each microservice is a container.

- Once a microservice is containerised:
  - completely unaware about the underlying computing infrastructure
  - only the interface is relevant.

- We consider only TCP/IP based interfaces.

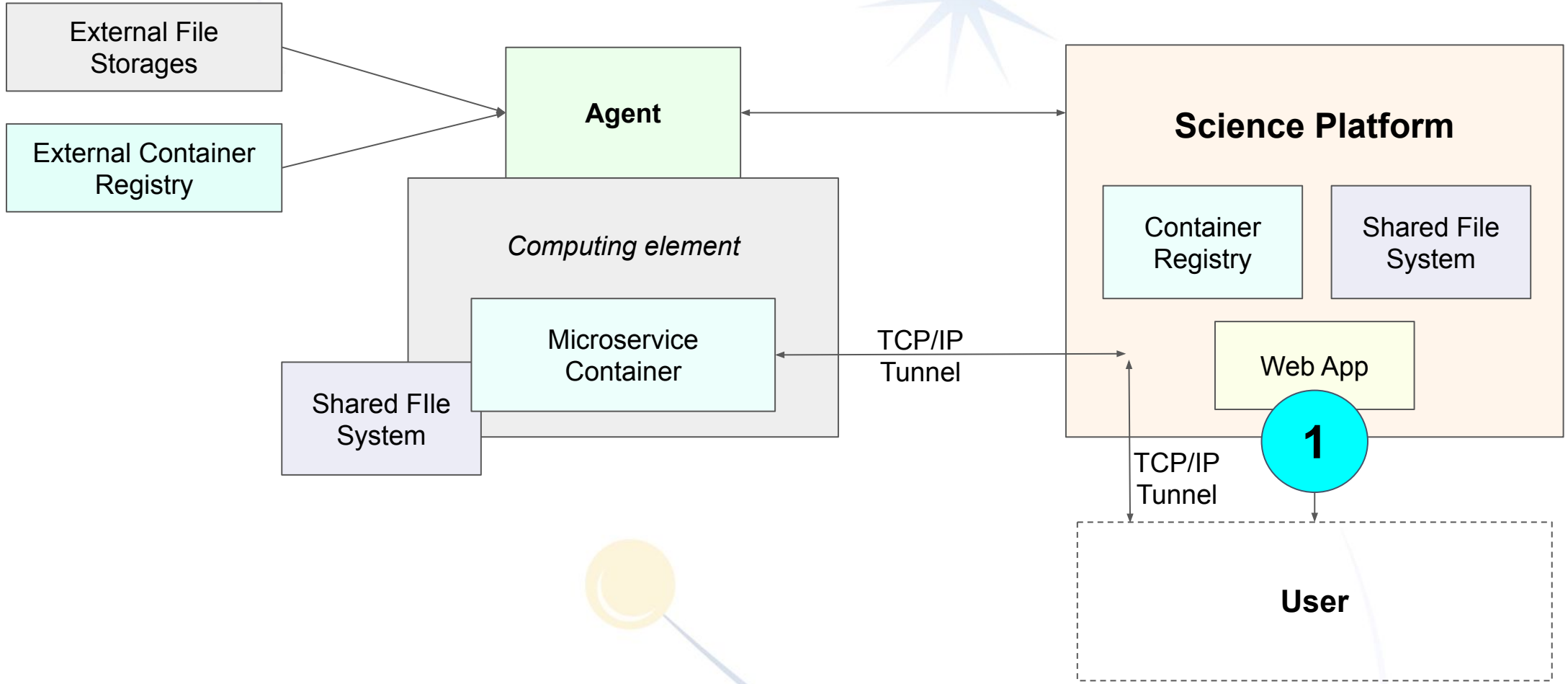
Funded by the European Union's
Horizon 2020 - Grant N° 824064

A Science Platform architecture



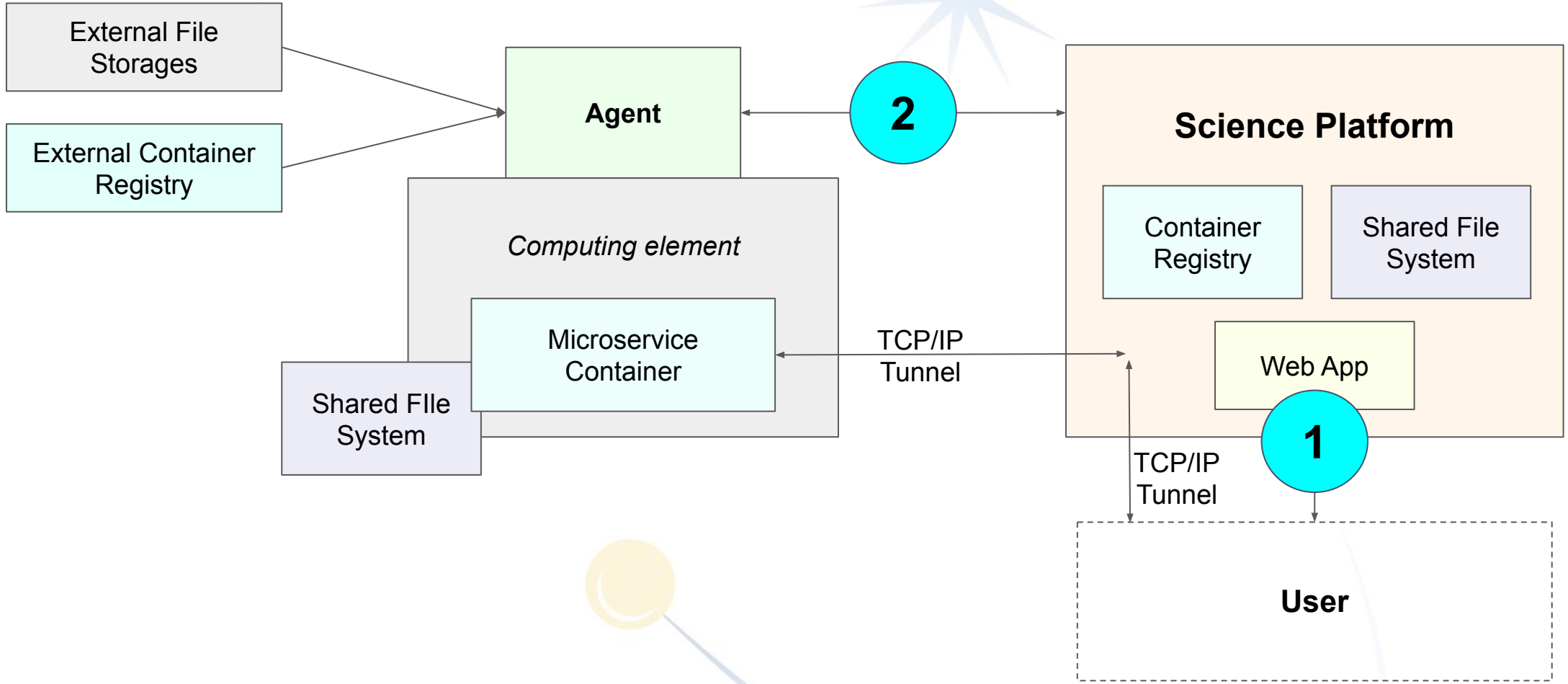
Funded by the European Union's Horizon 2020 - Grant N° 824064

A Science Platform architecture



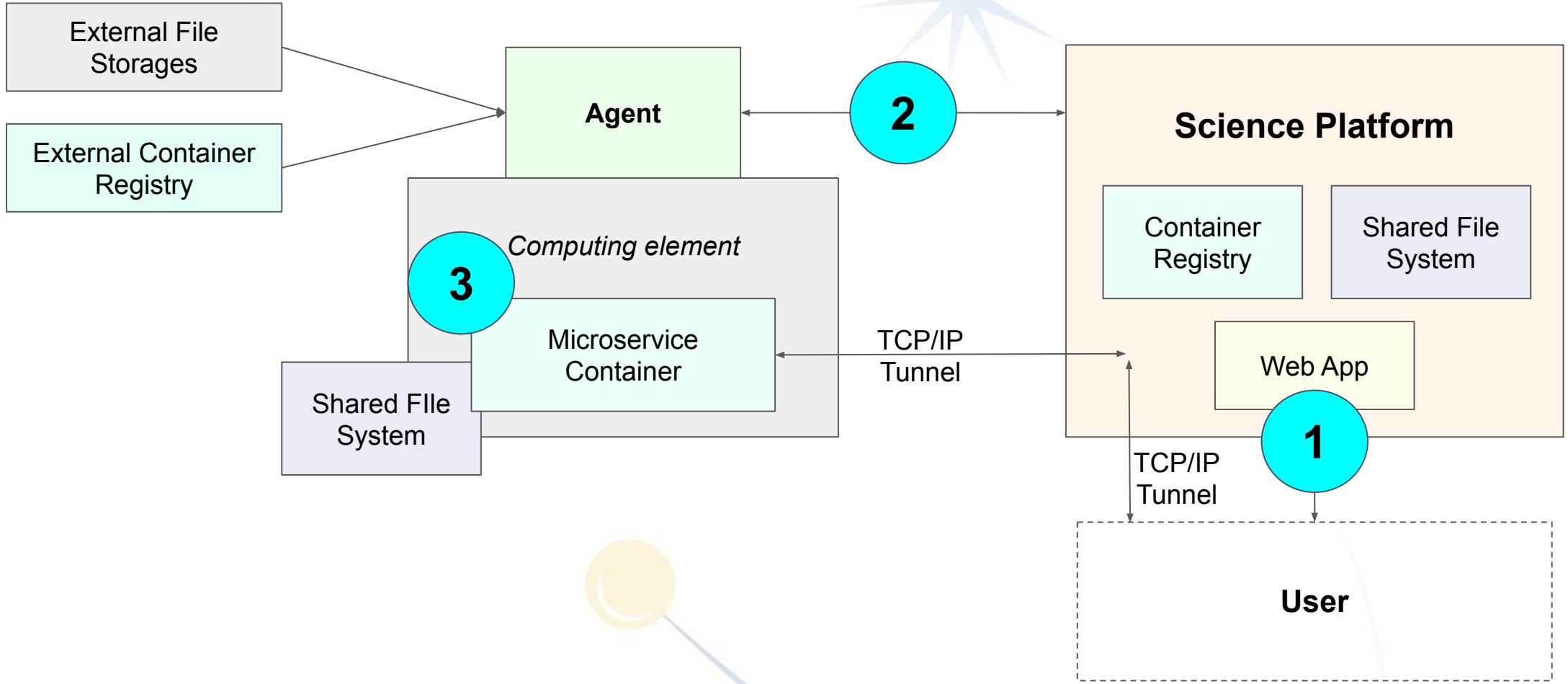
Funded by the European Union's Horizon 2020 - Grant N° 824064

A Science Platform architecture



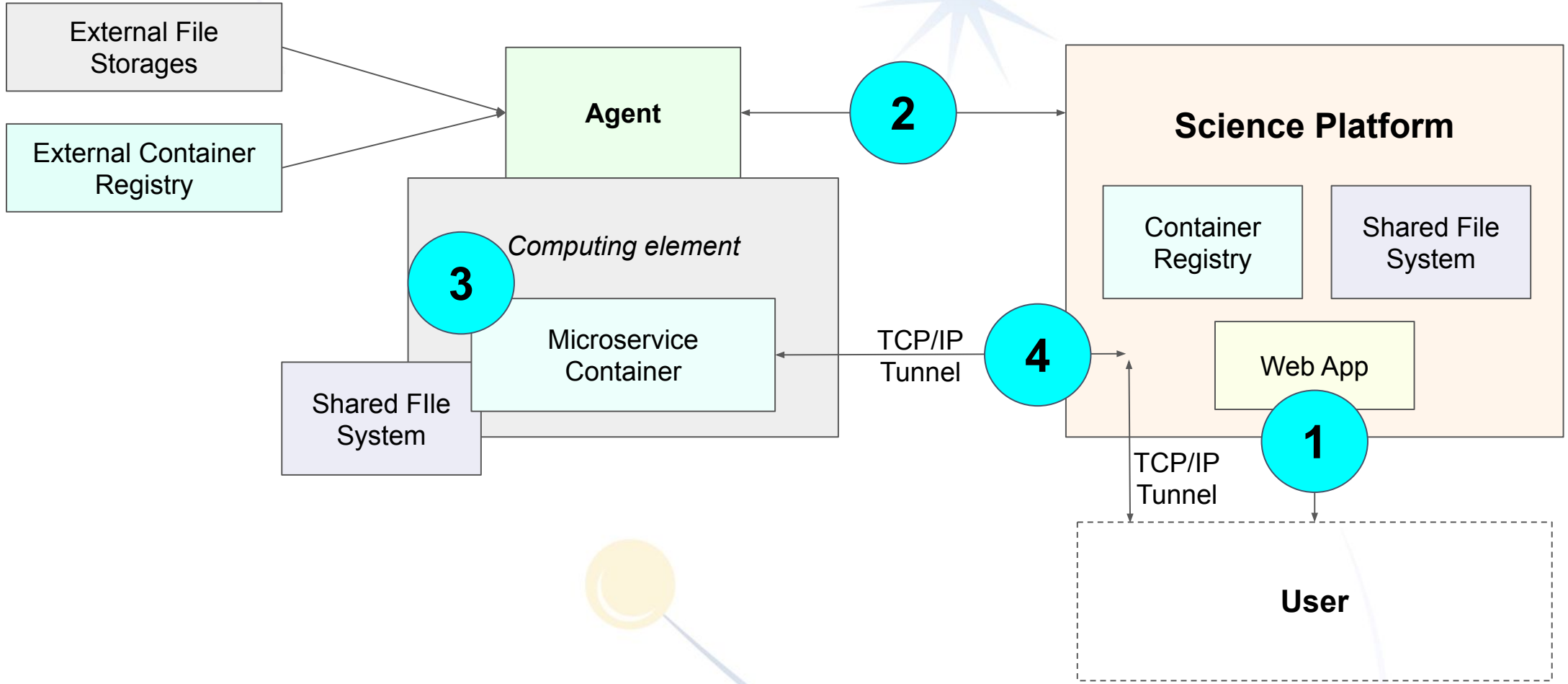
Funded by the European Union's Horizon 2020 - Grant N° 824064

A Science Platform architecture



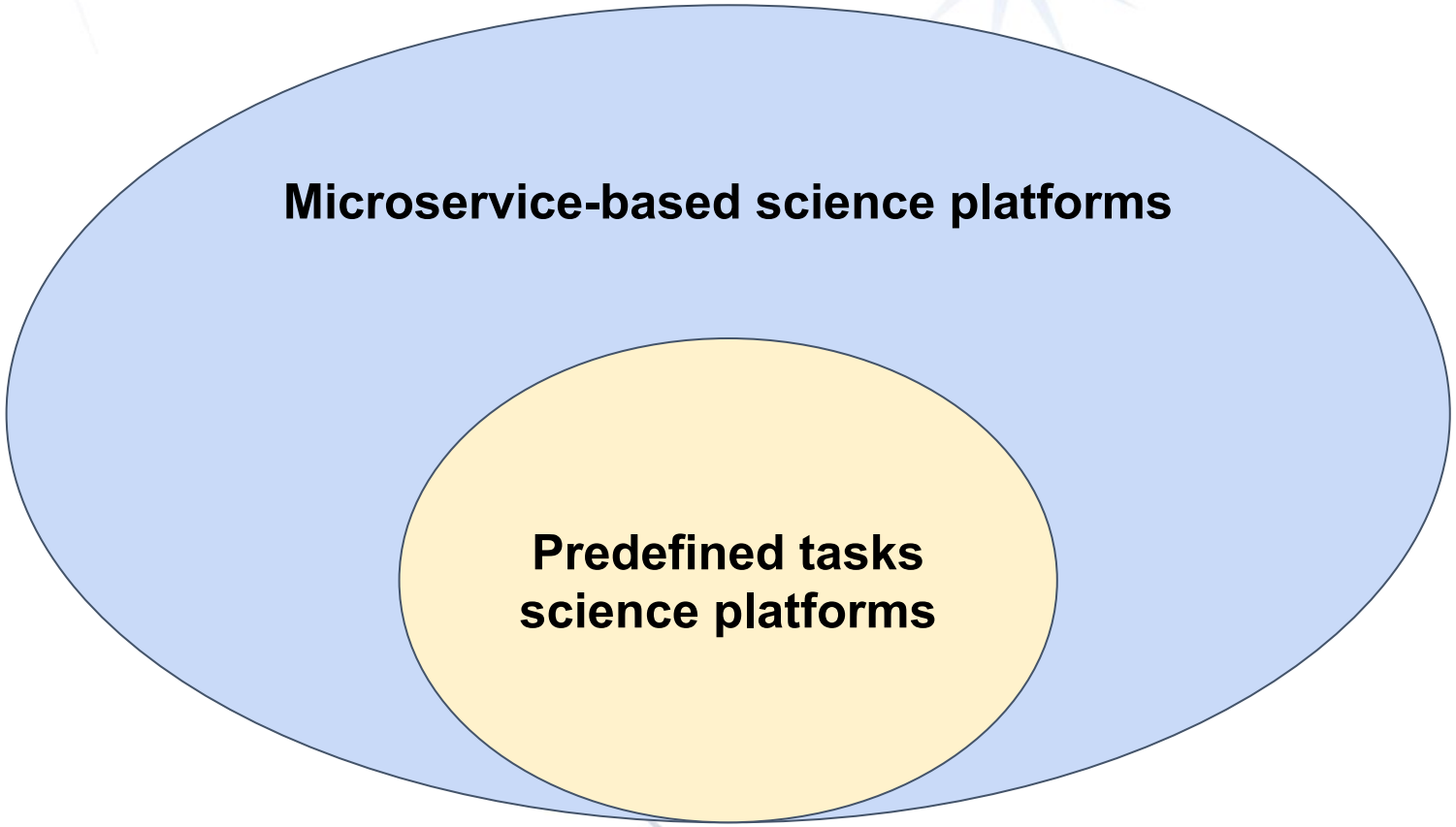
Funded by the European Union's Horizon 2020 - Grant N° 824064

A Science Platform architecture



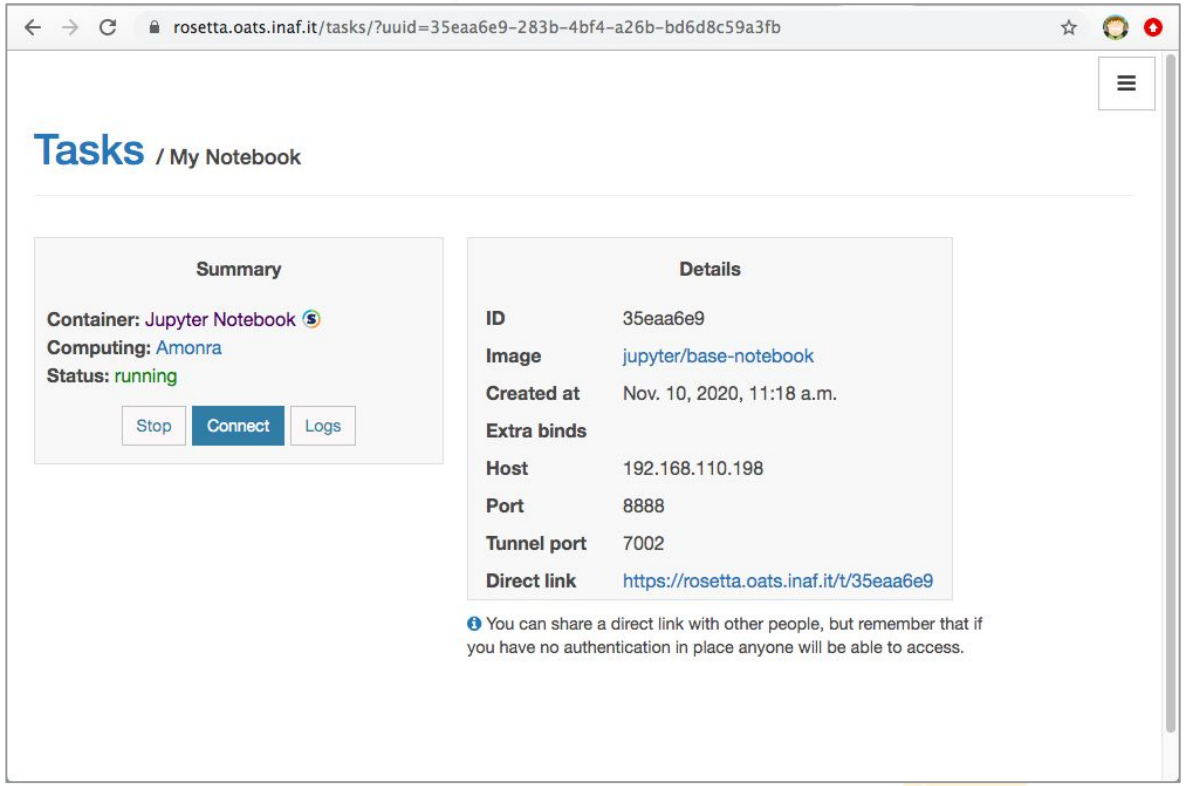
Funded by the European Union's Horizon 2020 - Grant N° 824064

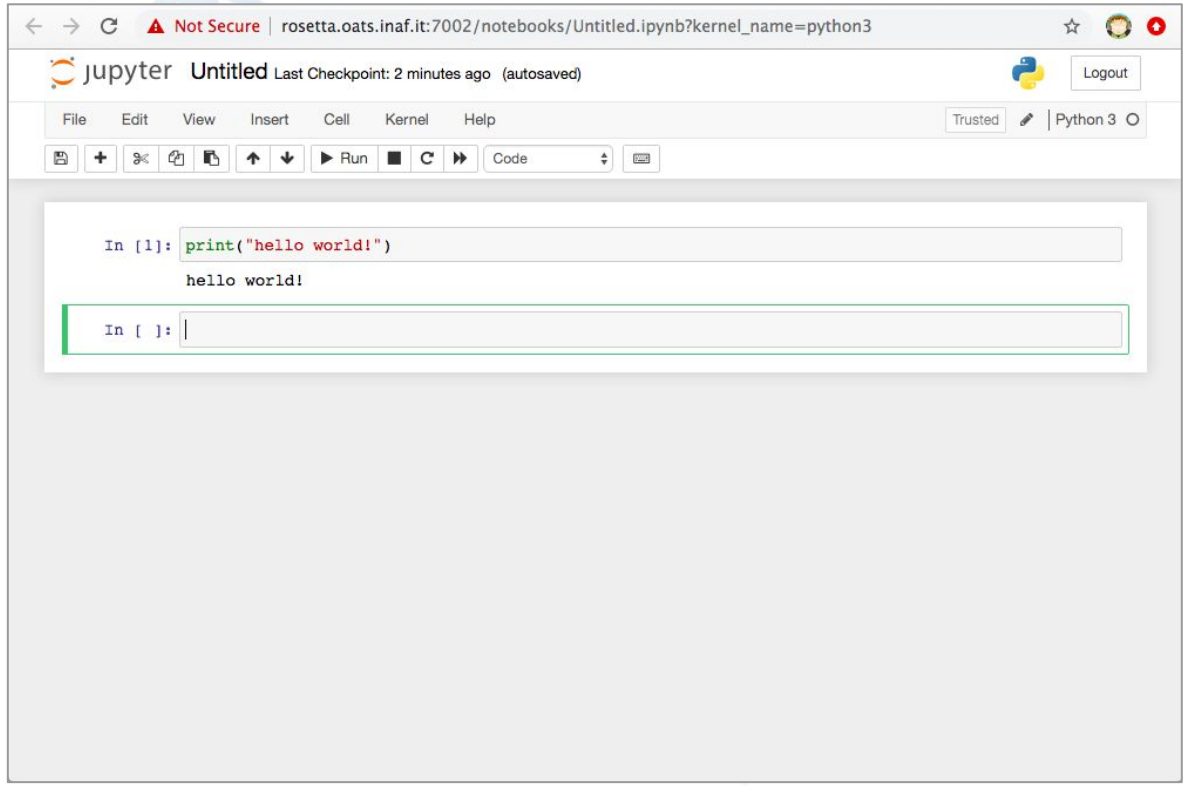
Capabilities of this approach



Funded by the European Union's
Horizon 2020 - Grant N° 824064

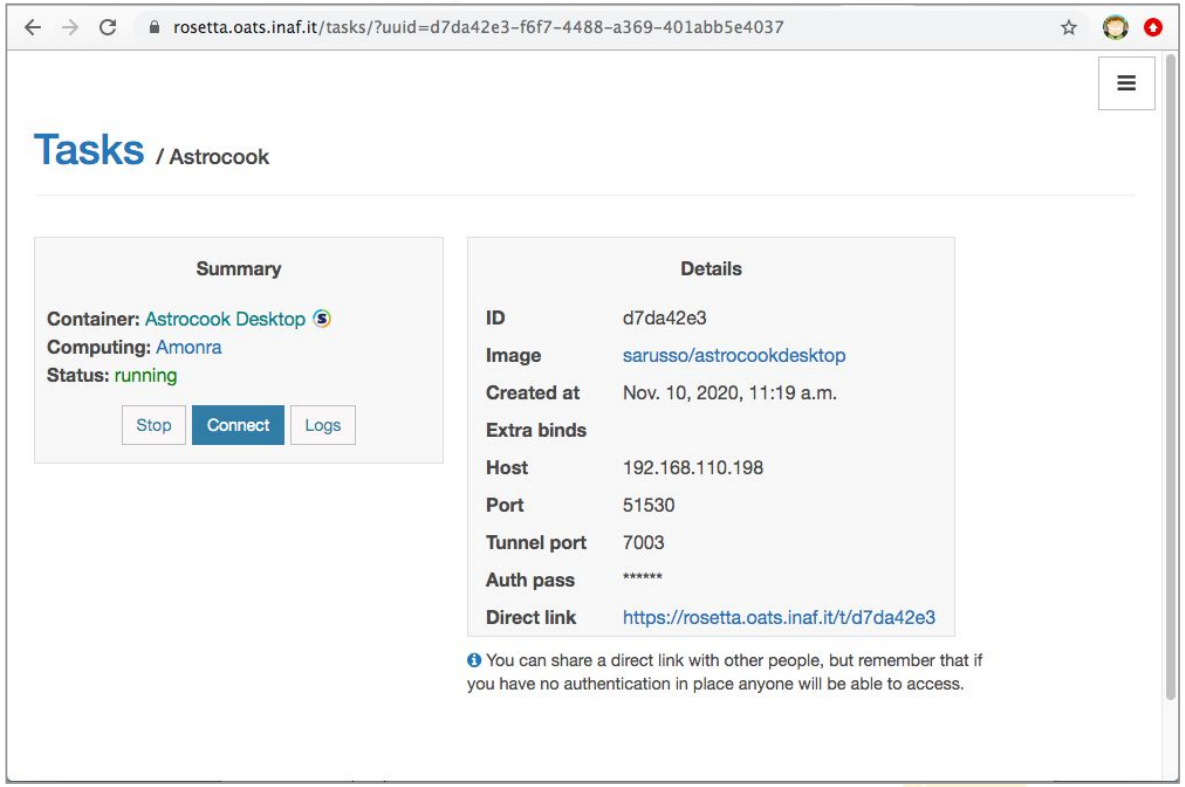
Our prototype at INAF

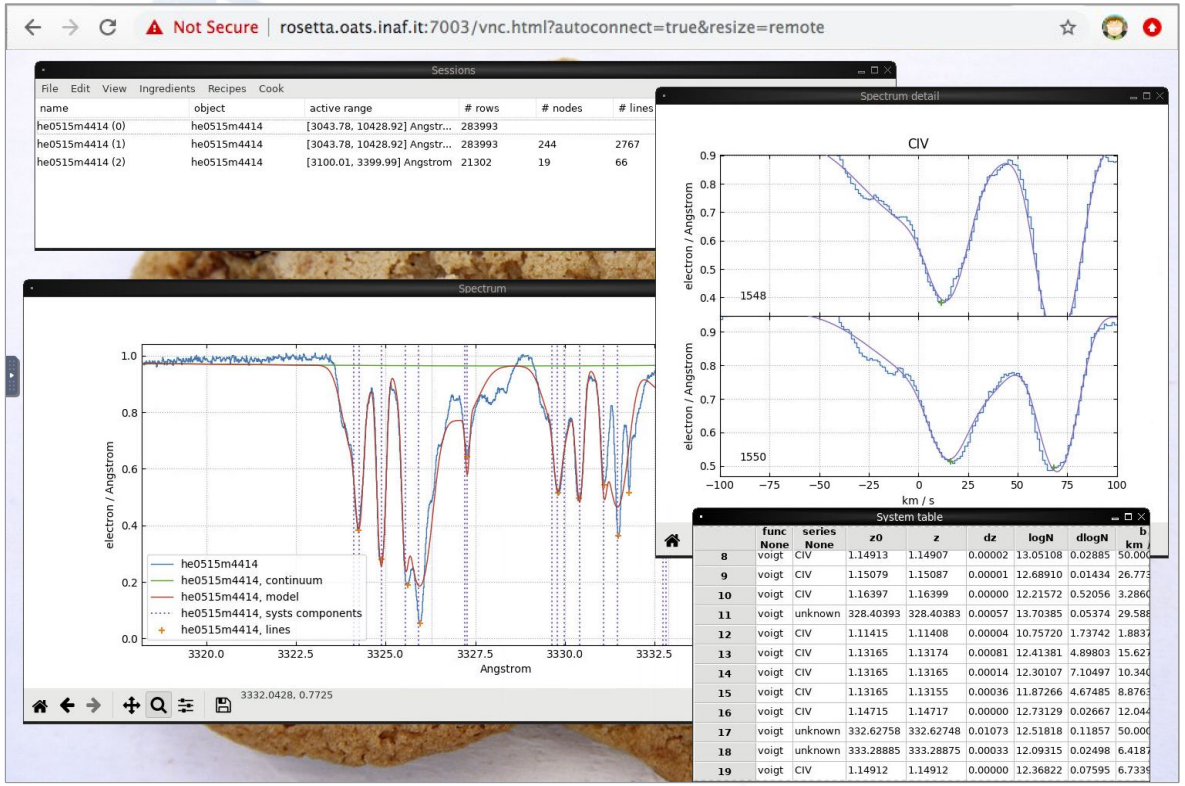




Funded by the European Union's Horizon 2020 - Grant N° 824064

Our prototype at INAF





Funded by the European Union's Horizon 2020 - Grant N° 824064

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*

Funded by the European Union's
Horizon 2020 - Grant N° 824064

THANKS!

Questions?

stefano.russo@inaf.it

Funded by the European Union's
Horizon 2020 - Grant N° 824064