SRCNet - Visualization Tools

Ugo Becciani

and The SRC - IT members of Orange Team

SRCNet Prototypes

- 1.a Data products replication, distribution, and synchronization across multiple locations (Rucio)
- 1.b Data products replication, distribution, and synchronization across multiple locations (CADC)
- 2. Federated Authentication and Authorization (AA) API
- 3. Data processing Notebooks
- **4. Visualization of SKA data with high volume of users and high amount of data**
- 5. Distribution of software, tools and services

Prototype 4 - Visualization of SKA data with high volume of users and high amount of data

Instantiation and deployment of *visualisation services* for a high user demand and data volume environment, using *flexible local computational resources*. This includes storage sharing, data access, federated authentication, and integration with workload manager.

Three main elements

Visualisation Tools

Design, development and deploy of *standalone* applications and *web-based* components to visualise and analyse SKA data *connected to the SRCNet services*

Visualisation Services

Implementation of *server layer services* for efficient visualisation on the user layer including data discovery, efficient data access, data extraction services, etc

Connected to other elements

Computing Services, Data Logistics (coupling with prototype 1), Computing Infrastructure, Data Infrastructure

Orange Team

Started working on Prototype 4: Visualization in PI15 (June 2022)

Five main acitivities

- Contributing to the **definition** of visualization **use cases** for SRCNet
- **Visualization Tools review** (dependencies, interfaces)
- **Collection of data products** and data formats from precursors and pathfinders
- **Adapting Visualization Tools to address use cases** (WG6) and work with SRC architecture and its data lake
- **Testing** and **deployment** of visualization tools and data access services **into SRC nodes**

ITA SRC	FR SRC	SA SRC
F. Vitello PO **G. Tudisco SM** R. Butora V. Galluzzi A. Lorenzani G. Marotta M. Molinaro E. Sciacca A. Zanichelli	M. Allen T. Boch F. Bonnarel P. Salome B. Cecconi A. Loh	K. Kirkham

SRCNet Visualization Use Cases

Acticvity. As a user visualising data, I want to be able to:

- **visualise image and catalogue data in an interactive way**, including overlaying other wavelength images and source lists;
- **visualise spectra and time series**, either through beamformed observations or at locations in an image cube;
- **split-out subsets of datasets** *by selecting sky areas in images or freq/time sections in a spectrum or time series plot*;
- **fit profiles to sources or emission lines**;
- **perform mathematical operations per pixel** for analysis including polarisation, spectral index, emission lines or temperatures;
- **create publication quality plots** such that it is reproducible by accessing the code (python, matplotlib);

https://confluence.skatelescope.org/display/SRCSC/%5BSP-2642%5D+Visualization+Tool+Review

Visualization Tools Review

Activity. Tools review

Aladin (Desktop & Lite)

Carta

VisIVO

Yafits

https://confluence.skatelescope.org/display/SRCSC/%5BSP-2642%5D+Visualization+Tool+Review

Visualization Tools Review

Aladin (Desktop & Lite)

Carta

VisIVO

Yafits

Aladin Desktop is Java Application

Aladin Lite is a web browser facility intended to display all **2D HiPS** (**Hi**erarchical **P**rogressive **S**urveys) and overlay **FITS** images, catalogs, etc... Version 3 is currently in beta test.
Aladin Lite allow remote rendering / GPU

https://confluence.skatelescope.org/display/SRCSC/%5BSP-2642%5D+Visualization+Tool+Review

Visualization Tools Review

Aladin (Desktop & Lite)

Carta

VisIVO

Yafits

CARTA adopts a client-server architecture

Remote rendering / GPU / Container

https://confluence.skatelescope.org/display/SRCSC/%5BSP-2642%5D+Visualization+Tool+Review

Visualization Tools Review

Aladin (Desktop & Lite)

Carta

VisIVO

Yafits

VisIVO software architecture consists of a desktop client and a backend. The backend can be executed locally or remotely and it can be distributed over several nodes, using MPI to allow parallel and distributed rendering.

GPU or CPU (osMesa) rendering
Remote visualization
Containerized Backend (Docker and Singularity support)
Backend can be executed in multiple nodes through a job scheduler (e.g. SLURM)

https://confluence.skatelescope.org/display/SRCSC/%5BSP-2642%5D+Visualization+Tool+Review

Visualization Tools Review

Aladin (Desktop & Lite)

Carta

VisIVO

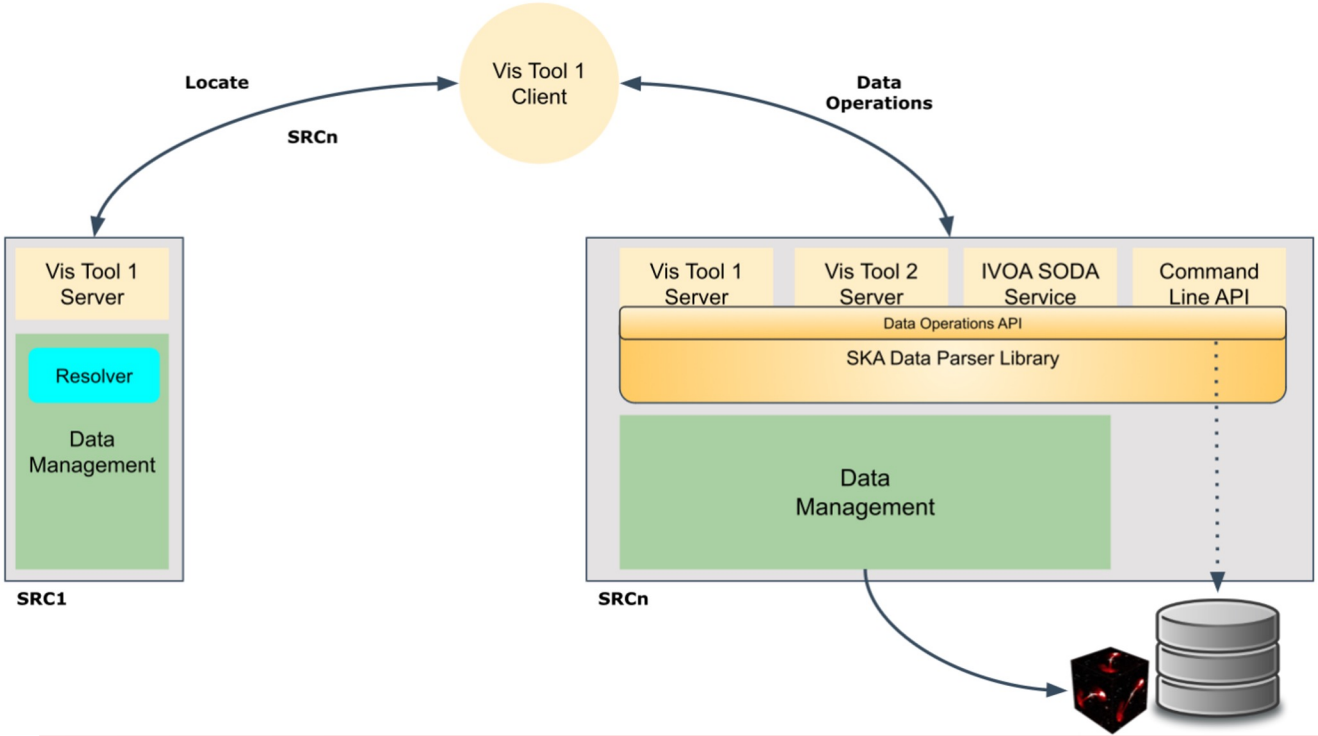
Yafits

Yafits is composed of several docker containers handling the different aspects of the application (data access, web server). Data are mounted in a docker container as a volume.
They are made available through endpoints provided by a web server and displayed in a web interface

GPU / Container

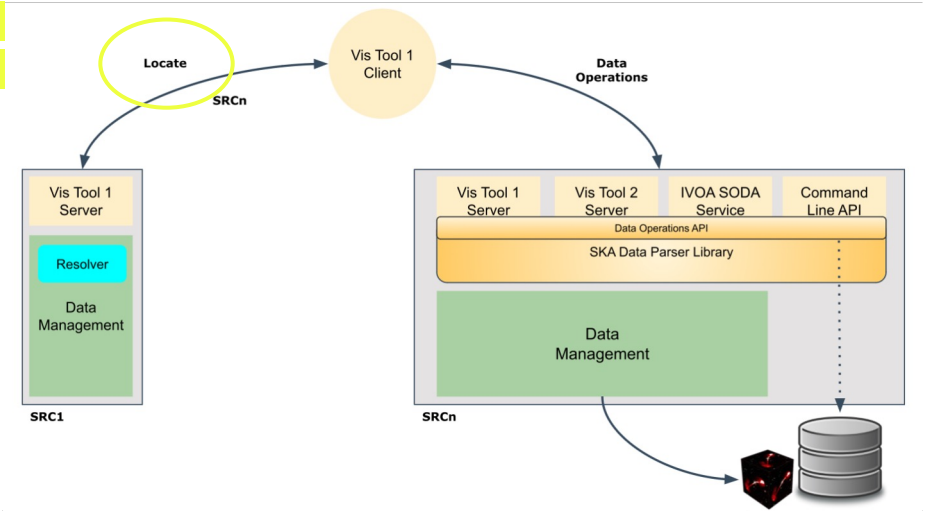
https://confluence.skatelescope.org/display/SRCSC/%5BSP-2642%5D+Visualization+Tool+Review

Visualization services architecture



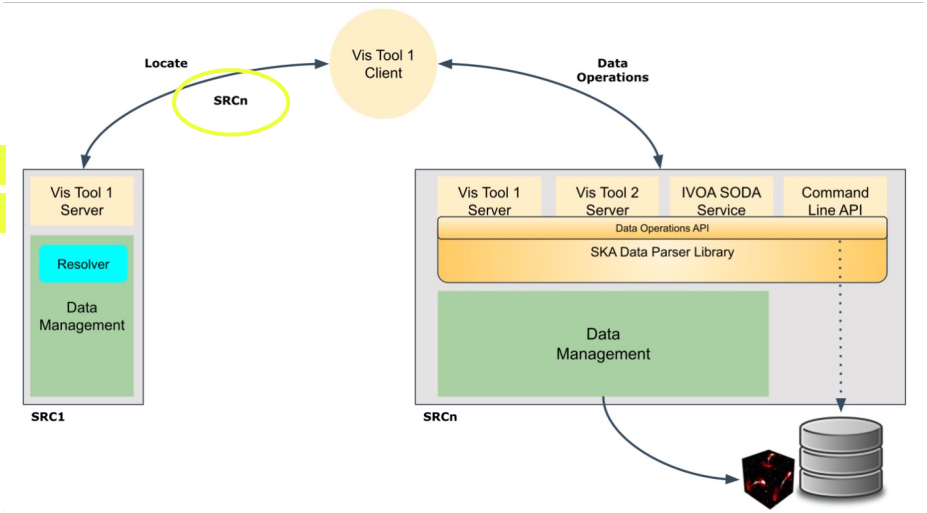
Visualization services architecture

- **Discovery Service will be used to locate the server that contains the requested data entity.**



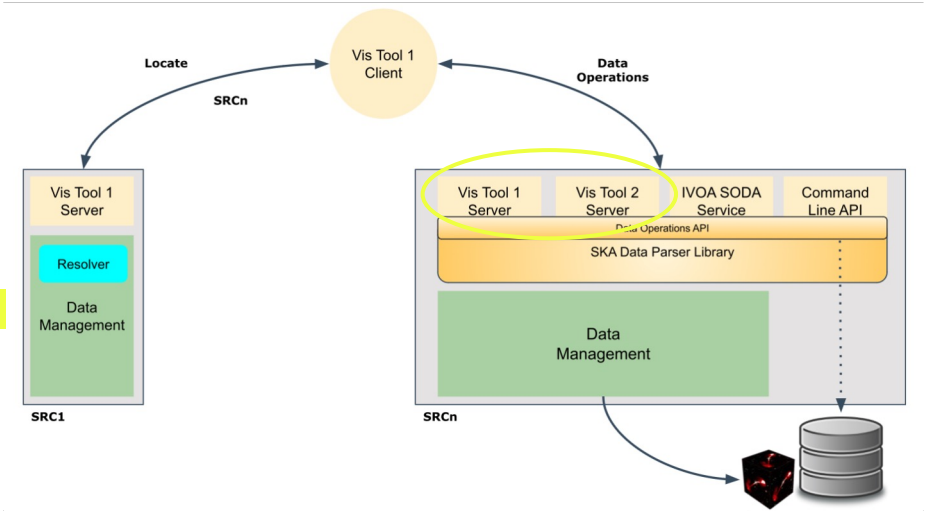
Visualization services architecture

- Discovery Service will be used to locate the server that contains the requested data entity.
- **The visualization tool will connect to the selected remote server to start the visualization.**



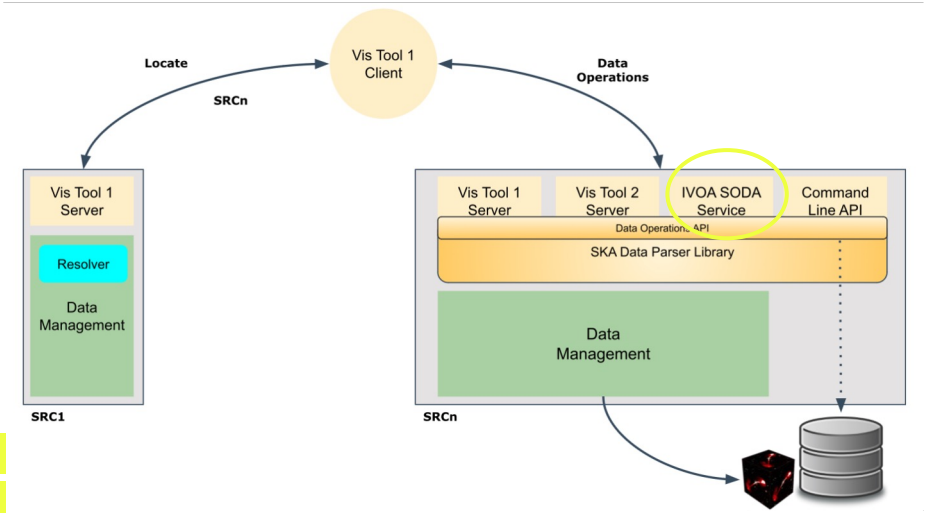
Visualization services architecture

- Discovery Service will be used to locate the server that contains the requested data entity.
- The visualization tool will connect to the selected remote server to start the visualization.
- **Visualization tool servers are deployed in all the SRCs.**



Visualization services architecture

- Discovery Service will be used to locate the server that contains the requested data entity.
- The visualization tool will connect to the selected remote server to start the visualization.
- Visualization tool servers are deployed in all the SRCs with data files to be visualized.
- **Performing data operation with the SODA service will help prevent latency**

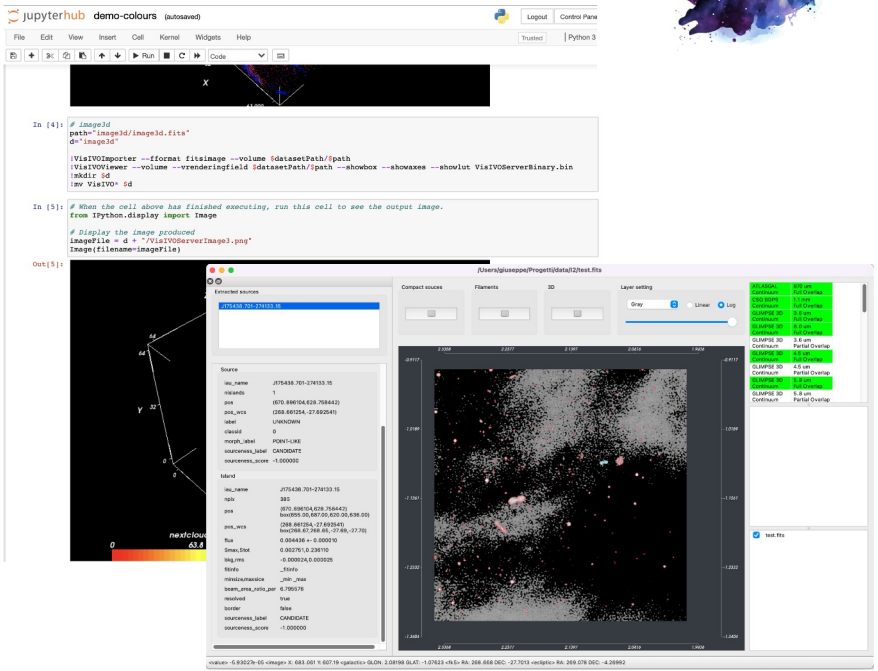


VisIVO Framework

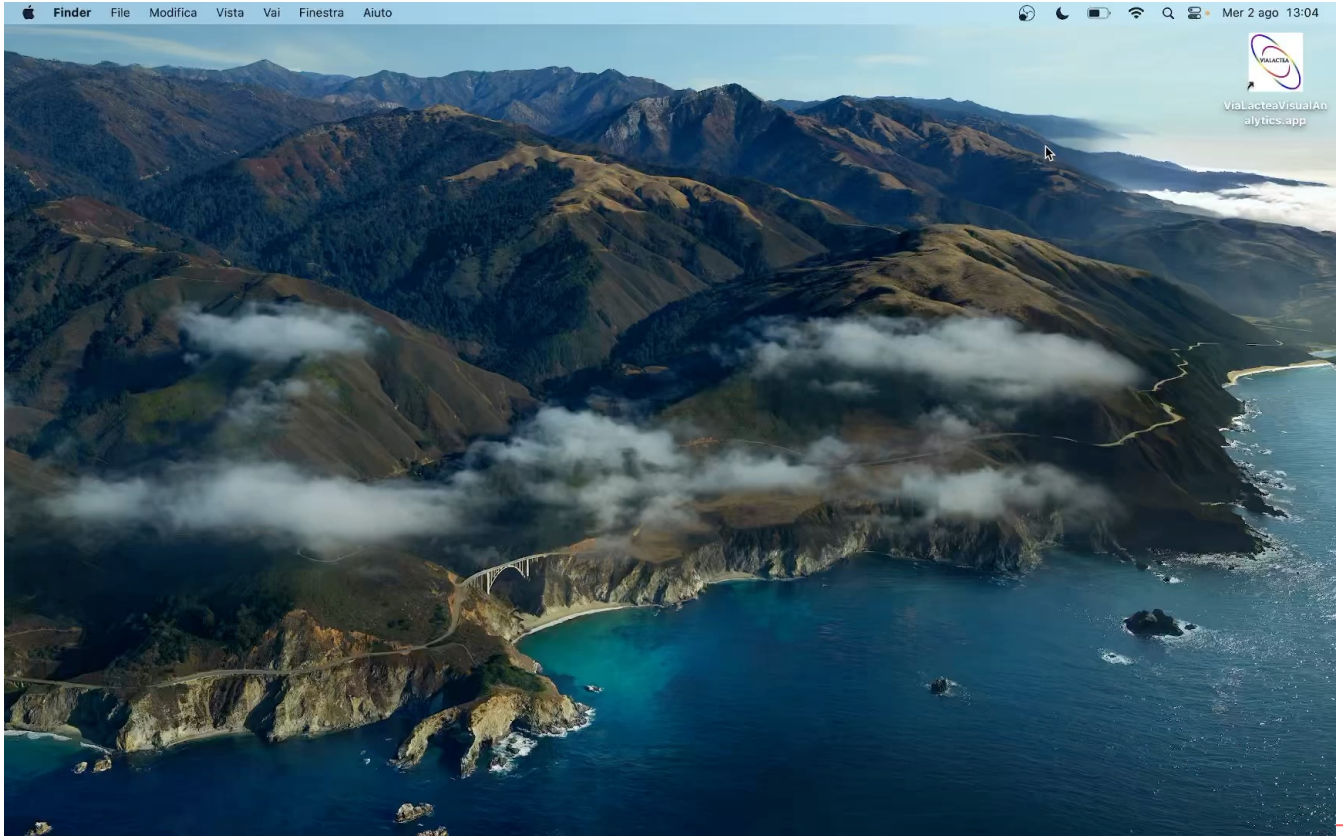
The VisIVO Framework (Visualization Interface for the Virtual Observatory) is a **set of tools and services for multi-dimensional data analysis**, maintained by Astrophysical Observatory of Catania.

VisIVO Framework consists of:

- **VisIVO Server**, a platform for high performance visualization.
- **VisIVO Gateway**, VisIVO Server on a Jupyter Notebook.
- **VisIVO ViaLactea Visual Analytics**, provides a visual analytic environment to analyze the correlation between different data, for example 2D intensity images with 3D molecular spectral cubes.



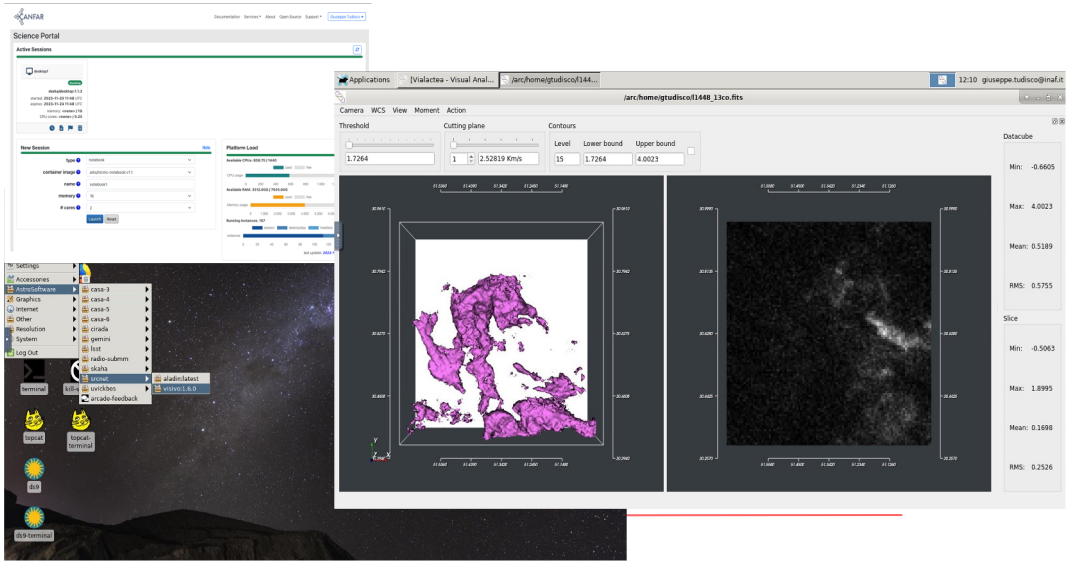
VisIVO SODA integration



Adapting VisIVO for the SRCs

Explored two approaches:

1. Containerized tool for science platforms (e.g. Canfar)
2. Client-server based application

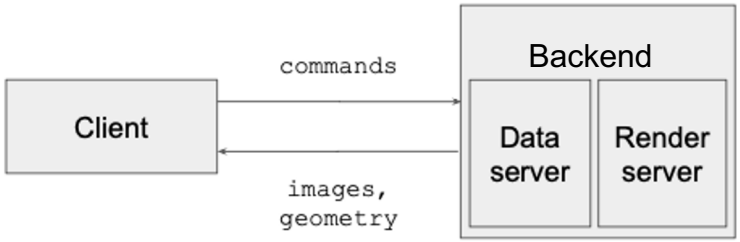


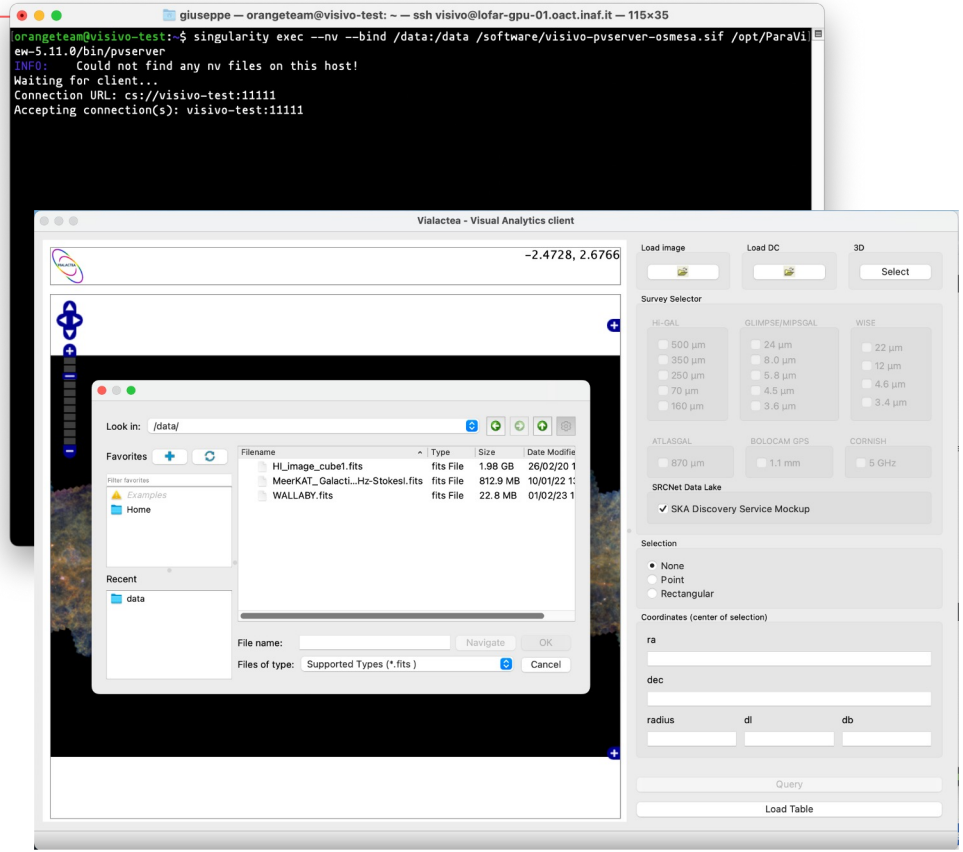
Client-server based application

Based on Kitware - Paraview framework

Three main logical components:

- a **client**, responsible for the user interface,
- a **data server**, to read and process data sets to create final geometric models,
- a **render server**, which renders that final geometry.





Parallel Visualization

- **Processing the data in parallel,** simultaneously using multiple workers. Workers can be different processes running on a multicore machine or on several nodes of a cluster.

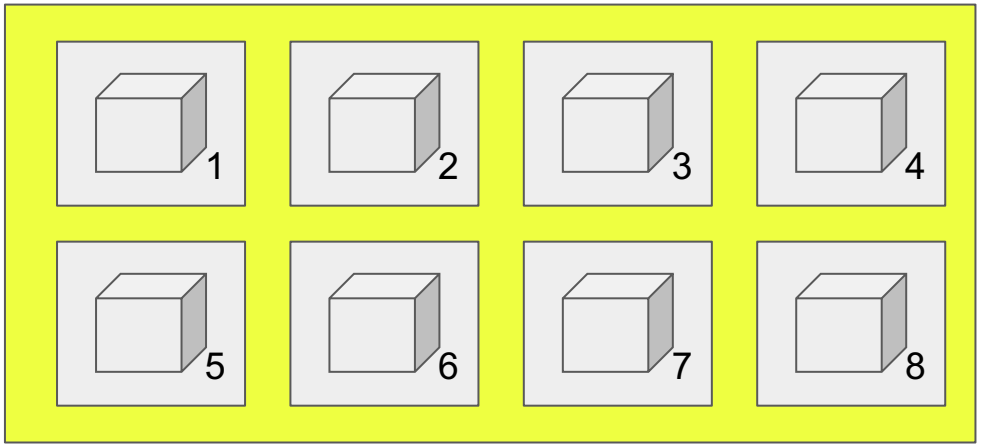
- The data server and the render server in this case are **a set of processes that communicate with MPI.**

- Each render server receives geometry from data servers in order **to render a portion** of the requested dataset.

Parallel Visualization

Steps:

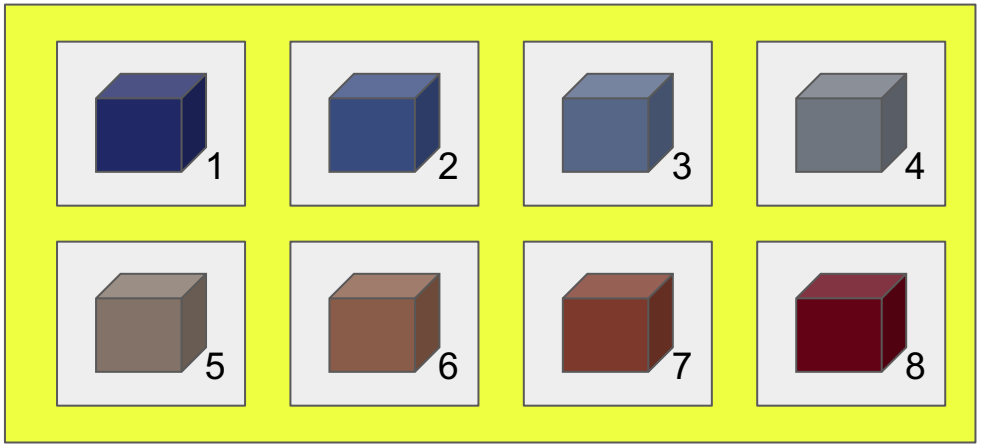
1. **Partitioning input**



Parallel Visualization

Steps:

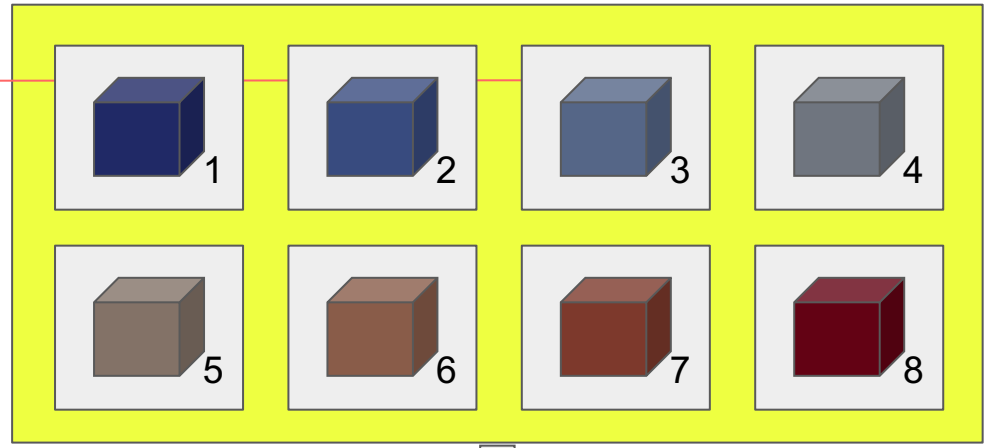
1. Partitioning input
2. **Each worker works on its own chunk of data and produces partial results**

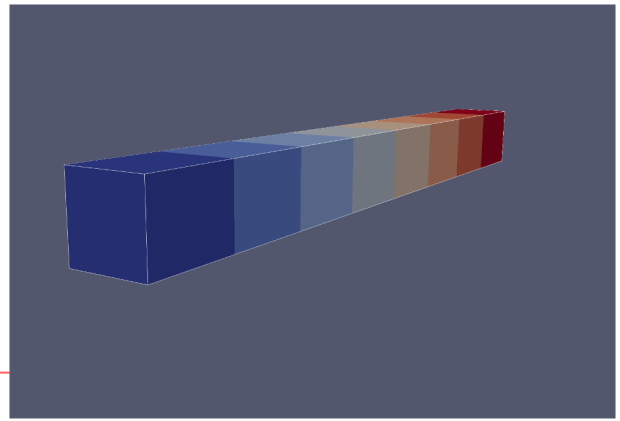


Parallel Visualization

Steps:

1. Partitioning input
2. Each worker works on its own chunk of data and produces partial results
3. **Construct the final output**

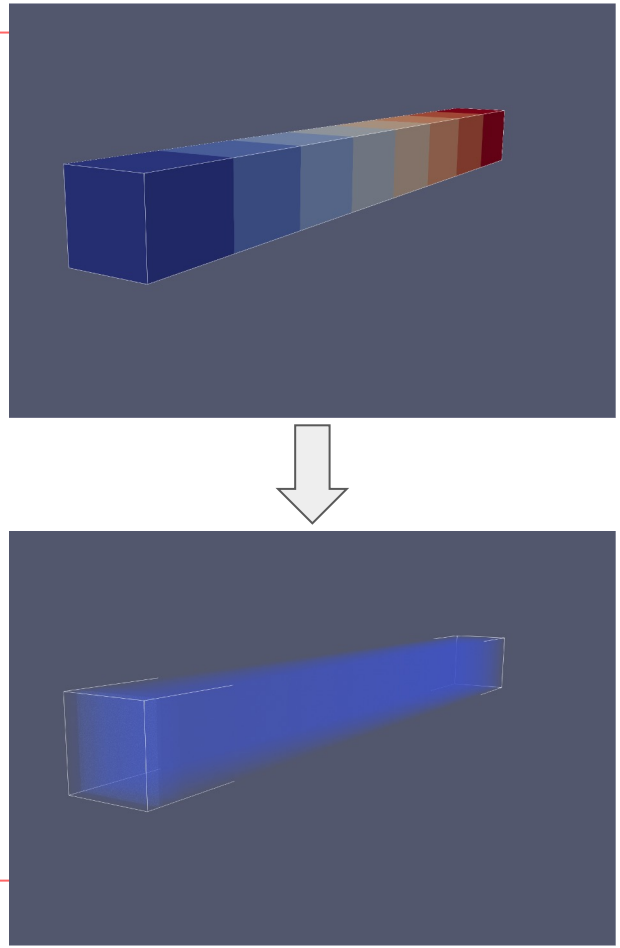




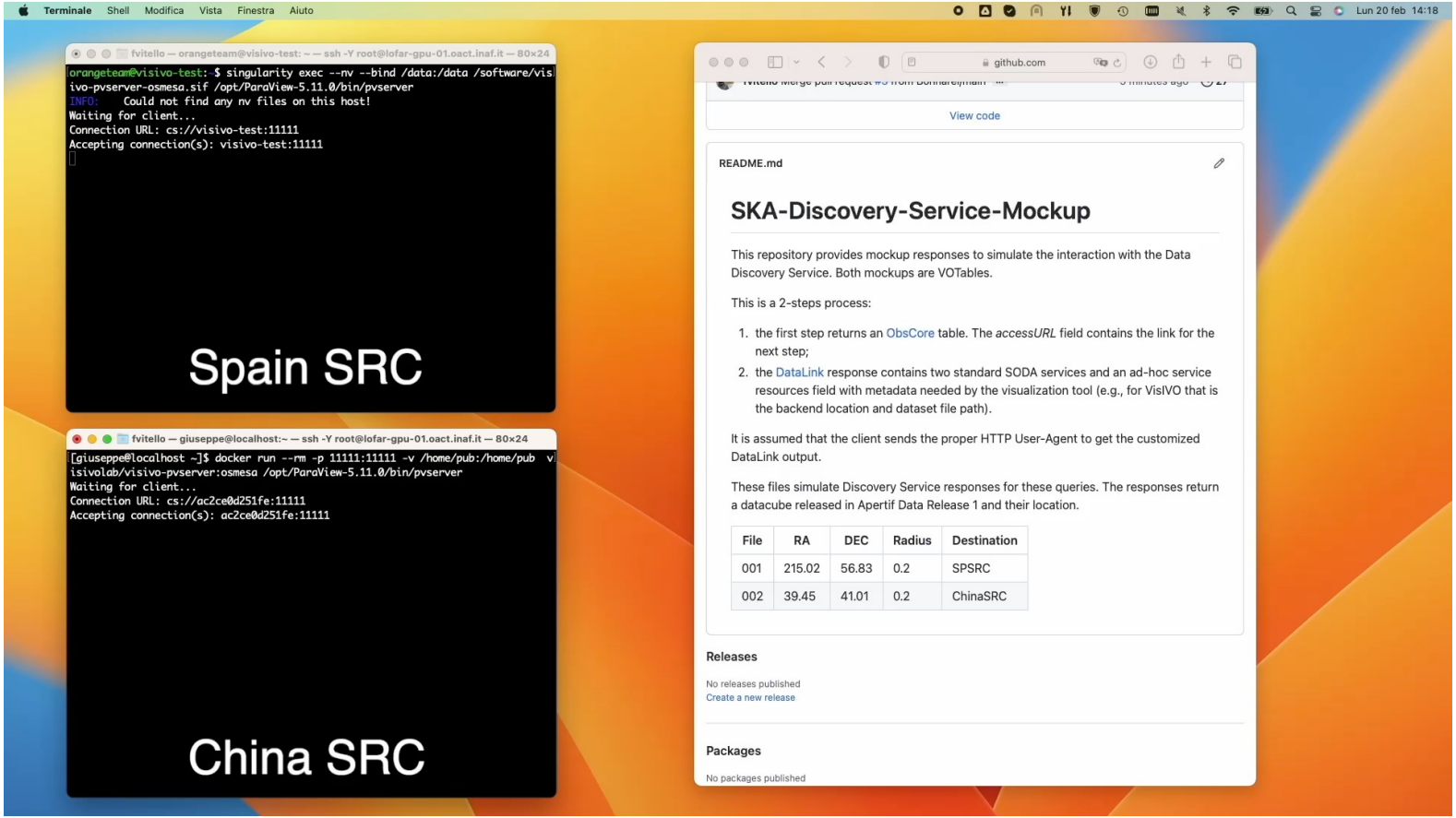
Parallel Visualization

Steps:

1. Partitioning input
2. Each worker works on its own chunk of data and produces partial results
3. Construct the final output
4. **Send the result to the client**



Parallel Visualization



Conclusions

- Orange team is contributing to different **Epics**:
  - **Mini SRC Demonstrator:** An end-to-end multi-SRC node demonstrator from data management to execution planning with integrated A&A and interoperability across all sites.
  - **Data Lake Integration:** Improve Rucio integration with data discovery tools with IVOA metadata integration, compute platforms, and visualisation tools.
  - **Computing API:** Develop a client-server API to submit/execute operations on the SRC federated computing resources and for the related monitoring. This EPIC will also cover the retrieval of information on available/suitable computing services
- **VisIVO @ Italian National Center for HPC, Big Data and Quantum Computing:**
  - VisIVO is one of the **Key Science Project** proposed within the Spoke 3;
  - VisIVO Technologies are employed in the **Innovation Project** HaMMon (UnipolSai+Sogei companies) for the development of methodologies to be used for the prevention and quantification of the effect of extreme natural events on Italian territory (flood and extreme climate events, earthquakes and fires)