2° Forum della Ricerca Sperimentale e Tecnologica

# Design and implementation of in-site visualization approaches in VisIVO

*Giuseppe Tudisco*, Fabio Vitello, Eva Sciacca, Ugo Becciani*
*Osservatorio Astrofisico di Catania*

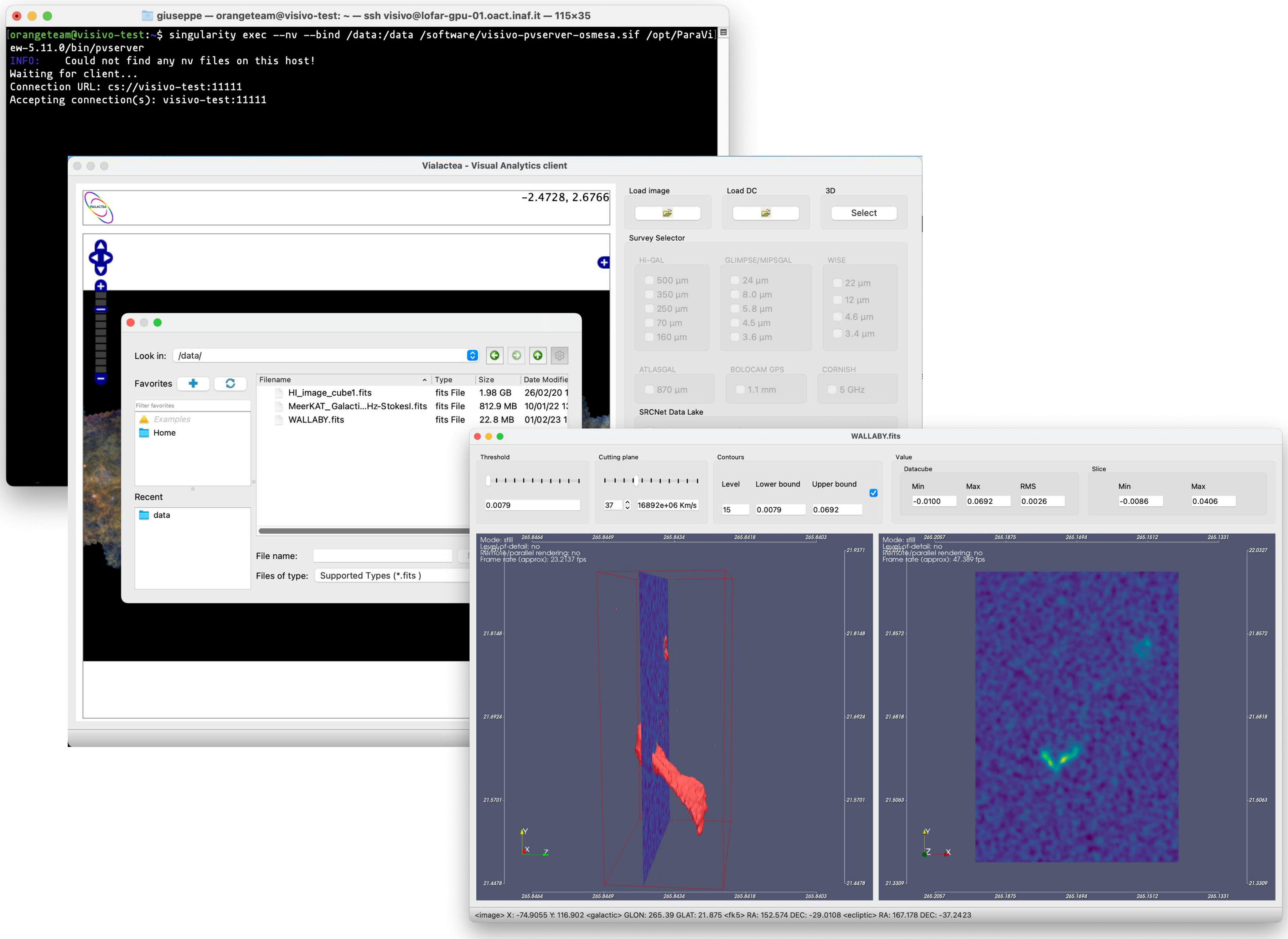
Next-generation ground-based facilities are expected to collect huge amounts of data that demand innovative techniques for data processing, storage, visualization and analysis. High precision and high resolution measurements allows researchers to probe the Universe in unprecedented detail. However, as data to be visualized may exceed by several orders of magnitude the typical storage and memory available in traditional personal computers, it becomes necessary to adopt appropriate approaches, such as in-site visualization, to limit and potentially avoid data transfers.

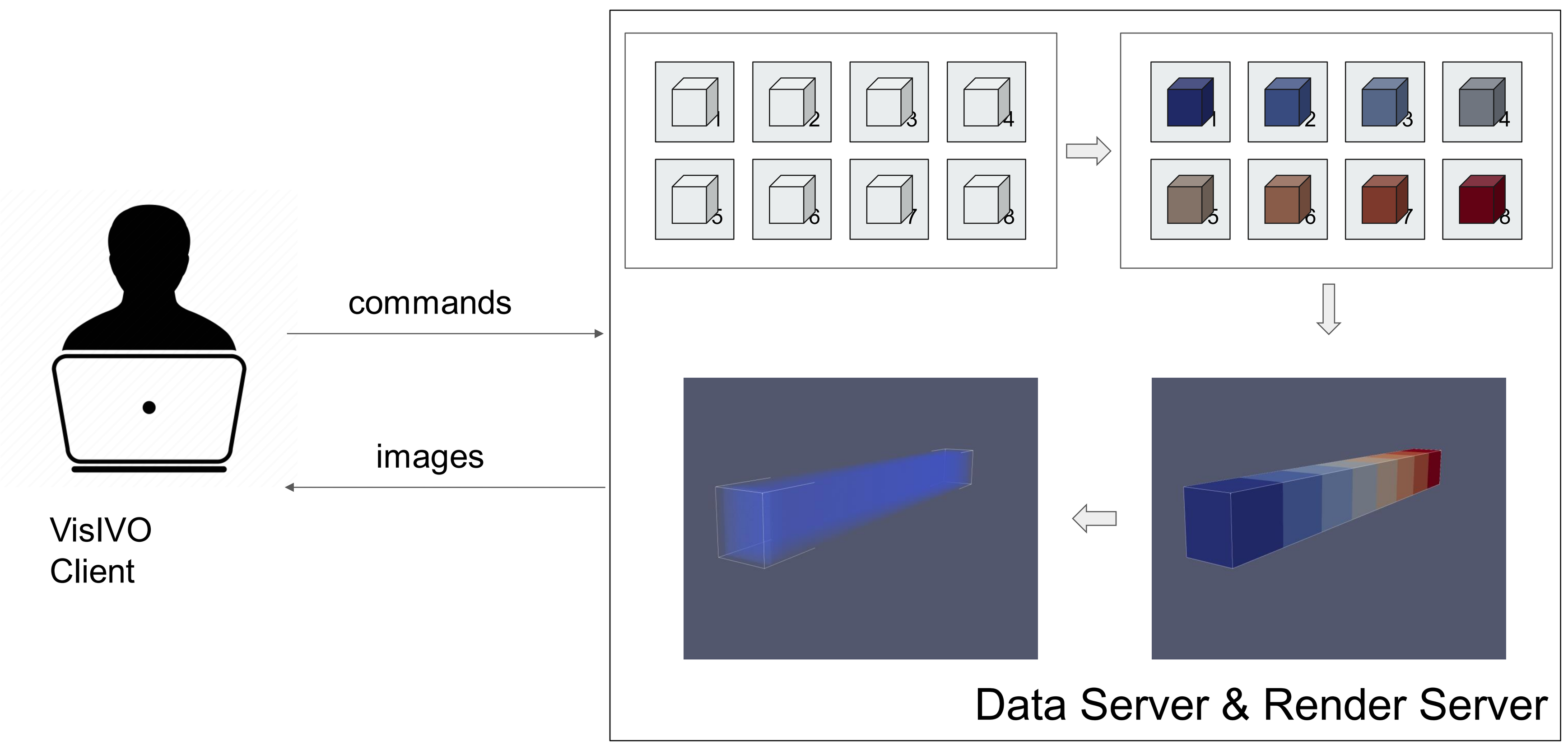
## Motivations

In view of the needs and challenges of the Square Kilometre Array (SKA) and the SKA Regional Centres (SRC), we have started to design and implement a remote visualization approach in VisIVO Visual Analytics. Remote visualization refers to any visualization where some or all of the data is on a different machine from the one used to look at images.

Motivations are

- Storage: Data expected to be visualized exceeds by several orders of magnitude the typical storage and memory available in traditional personal computers
- Bandwidth: Need to avoid data transfers as much as possible
- Computing: Personal computers do not have the processing power to handle such large amounts of data.





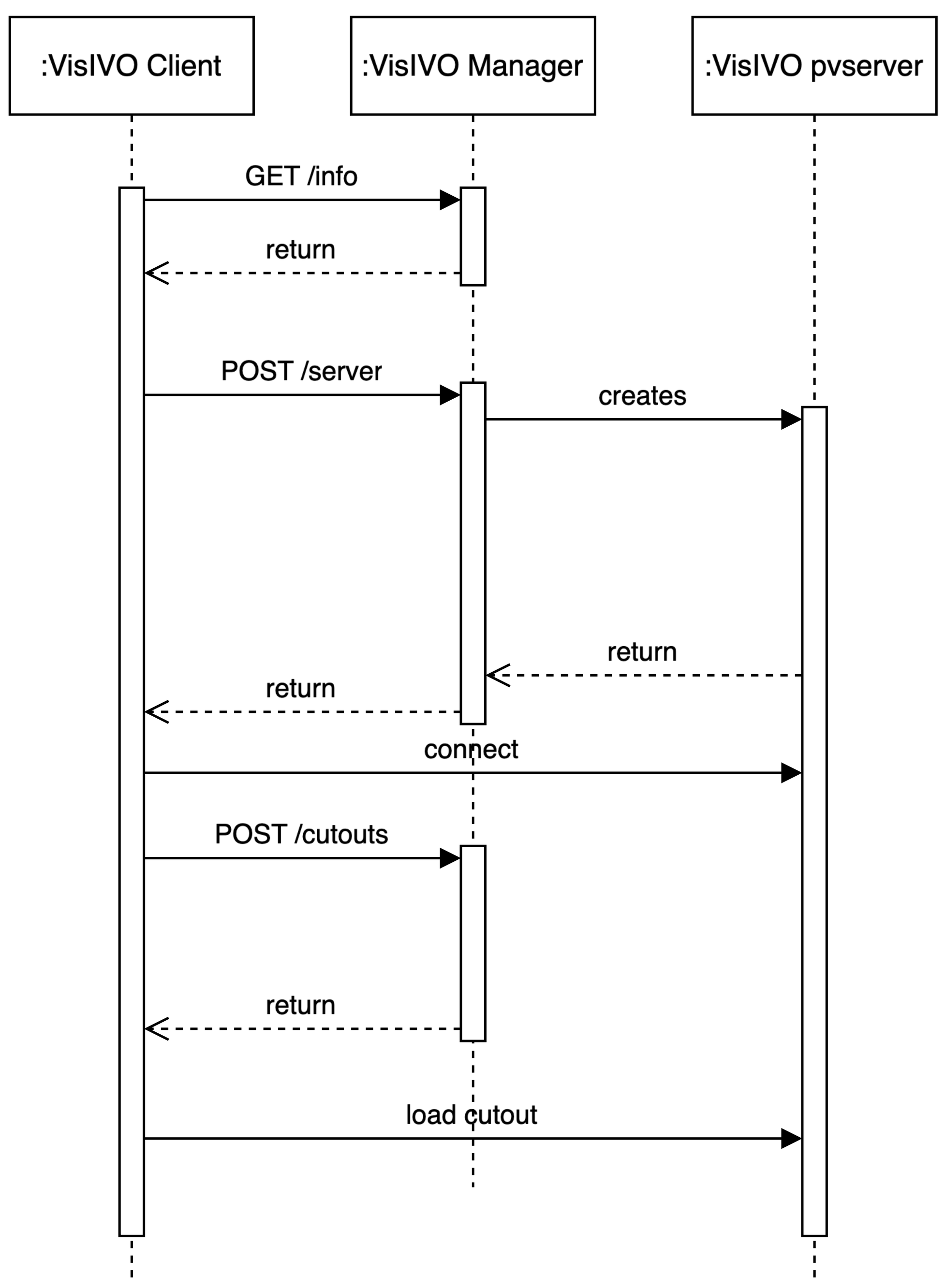

## Client-Server model

Our remote visualization approach consists of three main components: a client responsible for interfacing with the user, a data server that is responsible for reading and processing the data to be visualized, and a render server that is responsible for generating the geometries and images that are sent and displayed in the client. Data Server and Render Server may also co-exist in a single component.
This software architecture also allows the tool to achieve parallel visualization in addition to remote visualization. In this case, the server component runs on multiple nodes in an MPI cluster thus allowing the load of reading and analysis to be distributed among the cluster nodes.

| HTTP Method | Route | Description |
|---|---|---|
| GET | /info | Retrieves server information and tells clients if an MPI runtime has been detected. |
| POST | /server | Starts a new server instance. The client specifies the port and, if allowed, the number of MPI processors for parallel execution. |
| GET | /server | Get info on a running server instance such as start time, process status, and working directory. |
| GET | /logs | Retrieves the logs of the currently running server instance, providing insights into execution details and errors (if any). |

## Middleware services

To facilitate communication between the client and server in our client-server model, a middleware service that handles the lifecycle of server instances via HTTP requests is being developed. This service listens on a TCP port and provides critical functions for server management. For instance, through a HTTP POST request, clients can initiate a server instance by specifying the desired TCP port and, if applicable, request parallel execution via MPI. Additionally, the manager provides routes for retrieving execution logs or process information of ongoing server sessions, and executing tasks other than data rendering (querying SODA services, generating plots, etc).




***giuseppe.tudisco@inaf.it***