Science Gateway/e-science platforms Requirements and Recommendations

F. Cavallaro, F. Vitello, E. Sciacca, A.Costa
INAF oact

H2020 AENEAS Outputs

Work Package 5 focus on the production of **recommendations** for the design of **user interfaces** for **data processing, reprocessing, analysis and visualization** for the ESRC.

1. Evaluation of tools currently in use in Radio-astronomy.
   - Packages such as CASA or AIPS or software used in LOFAR, ASKAP, ecc.
   - Tools for data processing, data imaging, visualization
   - Tools for post-processing (e.g. source finding)
2. Evaluation of the user survey results (D5.1) and gap analysis (D5.2)
   - “Local” computing ⇒ “Distributed” computing
3. Production of **recommendations** for the design of **user interfaces** that should facilitate the distributed processing foreseen for ESDC ⇒ *Science Gateway*

Current User Interaction Models and tools

Radio astronomy packages

- CASA
- Miriad
- AIPS
- LOFAR software stack
- MeerKAT pipelines
- ASKAPsoft
- Python tools
- …

Data Post-Processing

- Source extraction tools (point-like sources and extended ones)
- Cross matching tools

Data Processing (SDP?)

1. Flag the bandpass and flux calibrator;
2. Find the solution for the bandpass and flux calibration;
3. Apply the solution on the secondary calibrator (the gain calibrator);
4. Flag the gain calibrator;
5. Find the gain solution;
6. Apply the calibration on the data;
7. Flag the data.
8. (Optional) Direction-dependent Calibration

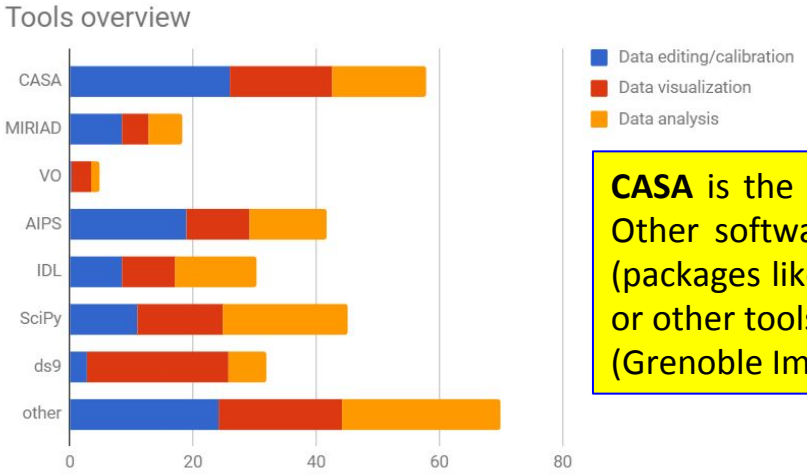
Data Imaging

- *tclean* task
- WSClean
- mfclean
- …

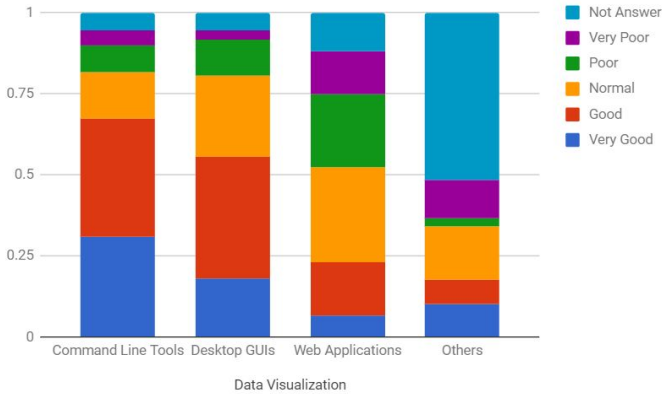
Data Visualization

- SAOImage DS9
- Aladin, TOPCAT, VisIVO
- Karma
- Casaviewer
- Python tools

Actual Interaction Models

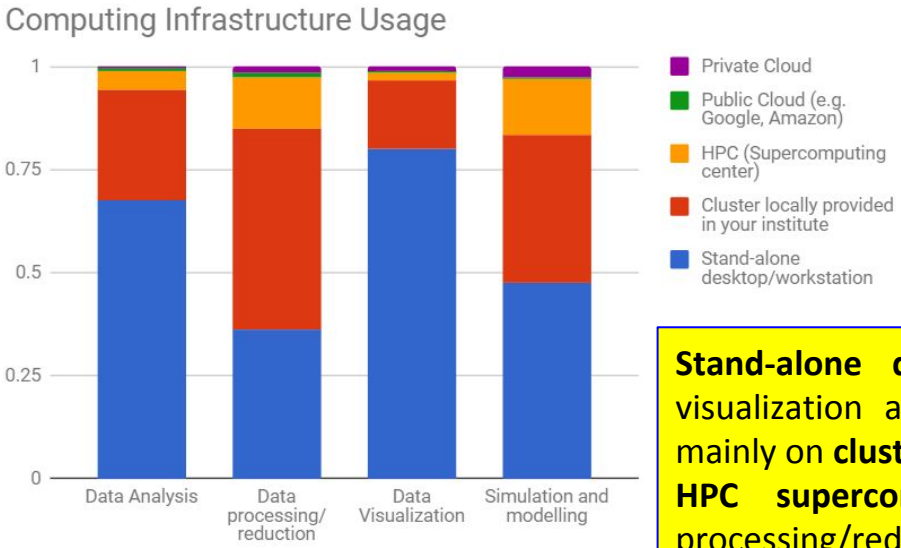


CASA is the most commonly used tool followed by **SciPy** and **AIPS**. Other software may include **user codes** or **libraries** e.g. in **Python** (packages like ParselTongue and libraries such as AstroPy) or Matlab, or other tools such as the **LOFAR data reduction** environment,**GILDAS** (Grenoble Image and Line Data Analysis Software) or **Difmap**.



Command line tools are the most commonly used mainly for all the data processing tasks. **Desktop GUIs** are mainly preferred for data visualization and analysis and **web applications** are most commonly used for data visualization with respect to the other processing activities. Other interaction models may include **scripting** and **batch interfaces** or **text-based user interfaces**.

Actual Interaction Models



Stand-alone desktop/workstation are mainly preferred for data visualization and analysis while data reduction tasks are processed mainly on **clusters locally provided** by the hosting research institutes.
HPC supercomputing centers are mainly exploited for data processing/reduction and simulation and modeling while **Cloud infrastructure** (public or private) have currently very limited usage.

User Interaction Model
from local to distributed analysis

Survey results show that users run their tools on **local resources** that are under their control.

Remote and distributed resources requires web interfaces that should be designed to facilitate the access to the computing infrastructures and give the flexibility to include command-line tools and smooth integration with local desktop environments.

The SKA REGIONAL CENTRE REQUIREMENTS document (SKA-TEL-SKO-0000735) defines a distributed framework, provided by the SKA Regional Center Alliance, for accessing the science data products. In particular, it states:

"Access to SKA science data products, as well as the tools and processing power necessary to fully exploit the science potential of those products, is provided via a Science Gateway. Access to science data products is irrespective of a SKA user's geographical location, or whether their local region or country hosts an SRC".

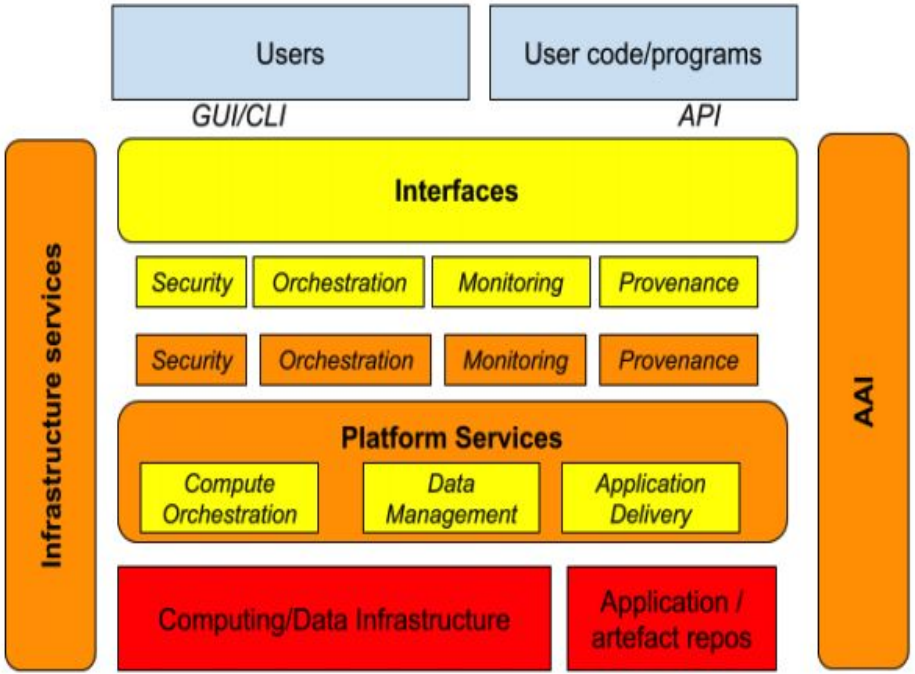
User Interface Recommendation

The main recommendation we make is to expose the ESDC processing system using a **Science Gateway**.

The term “Science Gateway (SG)” describes a whole class of *interfaces to scientific processing and underlying computing infrastructures*.

A science gateway provides users access to complex processing facilities. Generally this access is offered via a web interface that can be accessed via a browser. Depending on the implementation, a science gateway could expose GUI tools, command line interfaces, batch interfaces, API access etc.

Science Gateway features

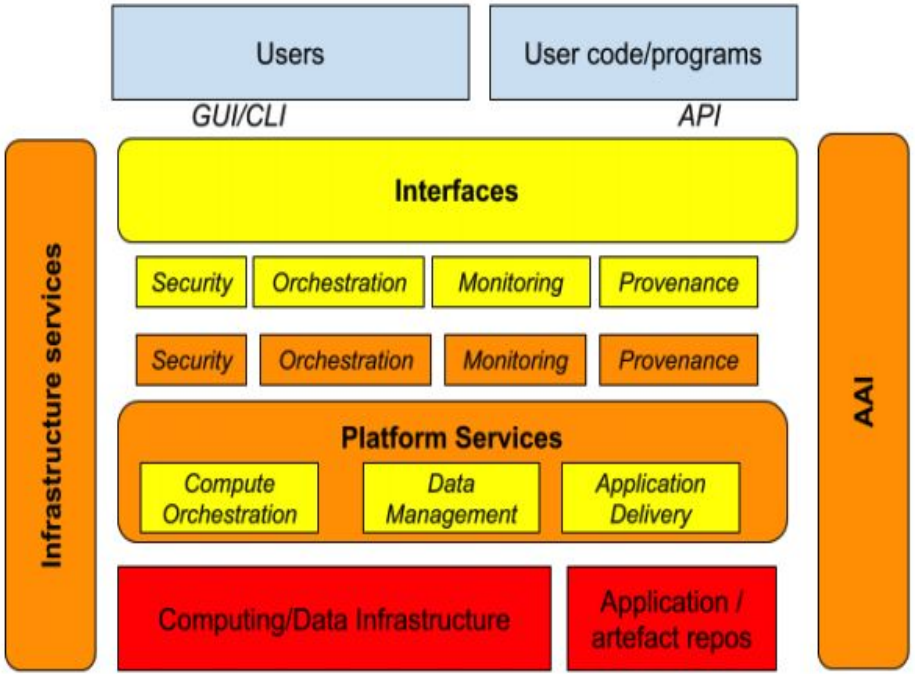


Computational Orchestration provides the scheduling, executing, and managing of various pipelines applied to a pool of resources to perform data analysis, processing or visualization offering transparent, scalable and interoperable computing management.

Data Management to support interfaces with archive related tasks such as acquisition, storage, retrieval, transport, organization, replication, curation, integration, and aggregation of data and metadata.

Application Delivery to support interfaces with application and/or artefact repositories including repositories of software codes (e.g. GitHub), containers that are able to package the software and all its dependencies (e.g. Docker Hub), pipelines and scientific workflows.

Science Gateway features



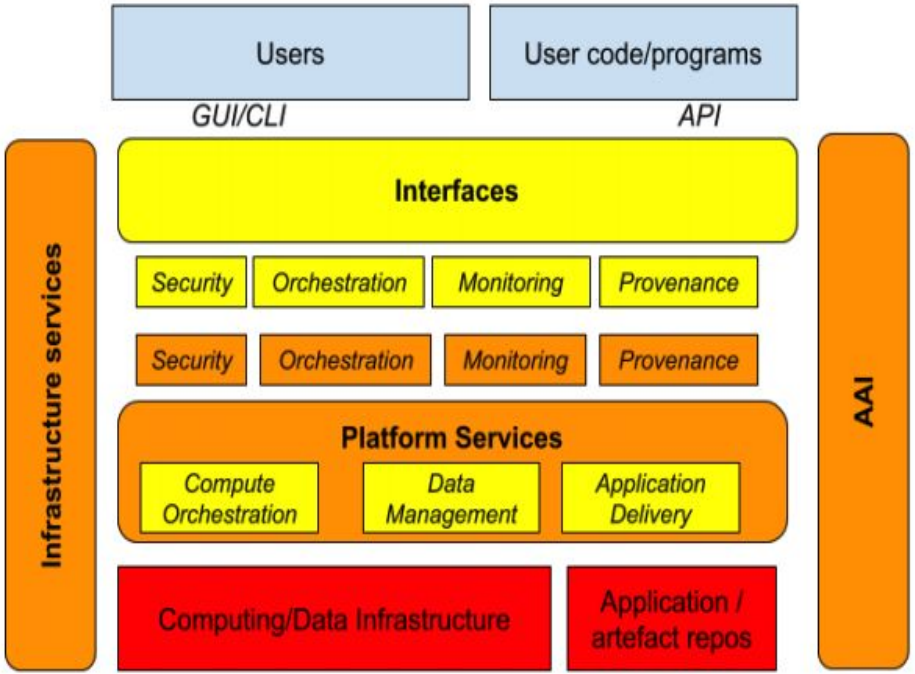
Orchestration to integrate and coordinate data, computing, and community management to instante and set up applications. Such functions are typically provided by a Workflow Management System (WMS).

Security to enforce authenticated and authorized access to data, applications, computing resources, communities, information, and SG functions.

Monitoring to collect and store the status of resources and different parts of the system, keeping track of the events and actions performed via the SG by users and automated processes.

Provenance to collect and provide lineage information about the actions performed by the SG and its users.

Science Gateway features



Users typically interact through ***Web-based Graphical User Interfaces*** **(GUIs)**. Others may prefer ***Command-Line Interfaces*** **(CLIs)** that offer more control and from which it is easier to automate repetitive tasks.

External applications can utilize **Application Programming Interfaces (APIs)**. Additionally, CLIs can also be used in scripts and programs to automate tasks or integrate with other systems.

Evaluation of SG technologies

	gUSE/ WS-PGRADE	Galaxy	HubZero	Catania SG Framework	Apache Airavata	Agave Platform
GUI/CLI	Yes	Yes	Yes	Yes	No	No
API	Partial	Yes	Yes	Partial	Partial	Yes
Coordination	Yes	Yes	Yes	Partial	Yes	Yes
Security	Partial	Partial	Partial	Yes	Partial	Yes
Monitoring	Partial	Yes	Partial	Yes	Partial	Yes
Provenance	No	Yes	Yes	No	Partial	Partial
Data Management	Yes	Yes	Yes	Yes	Partial	Yes
Computing Management	Yes	Yes	Yes	Yes	Yes	Yes
Community Management	Yes	Yes	Yes	Yes	Partial	Partial

Web-based science gateway frameworks include: gUSE/WS-PGRADE, Galaxy, HubZero and the Catania Science Gateway Framework.

APIs and libraries based science gateway frameworks include: Apache Airavata and the Agave Platform, which aim at reducing the effort on the developer side while enabling to apply novel user interface technologies and frameworks.

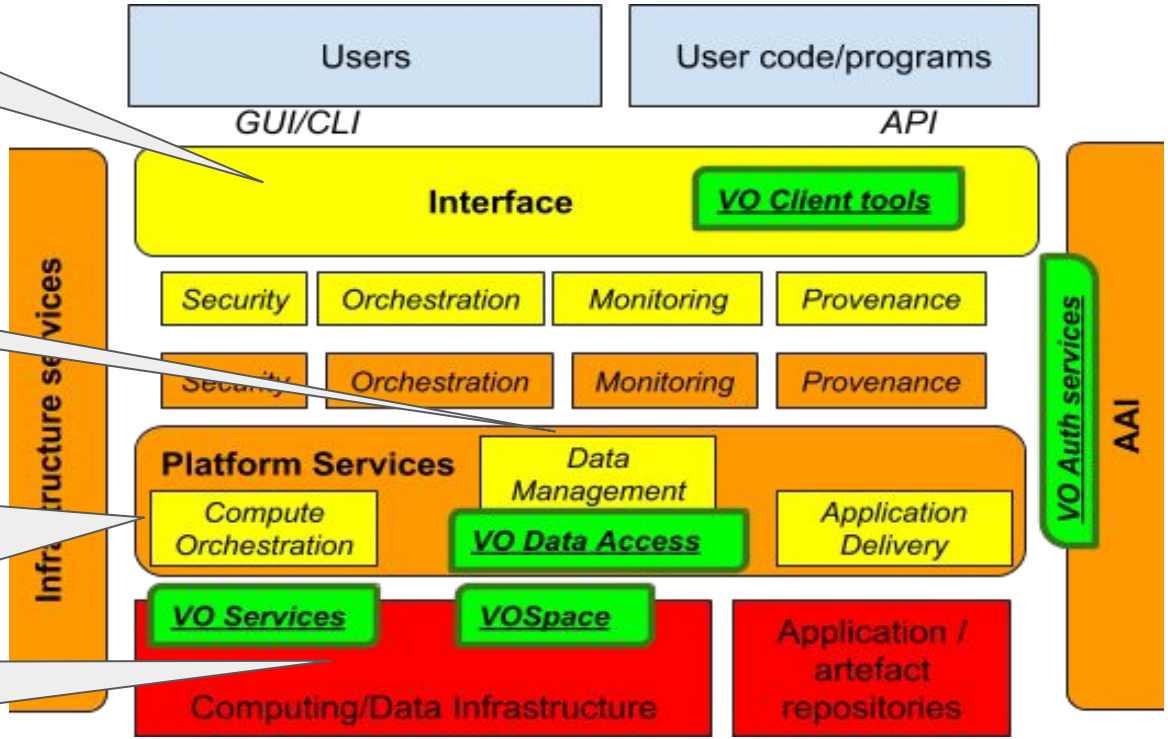
Science Gateway and VO interoperability

Interface: VO services and tools as web applications and also batch scripting mode of employ.

Data Management: VO data access services and protocols.

Orchestration: VO services in WMS. As an example *AstroTaverna* integrates VO web services such as STIL ToolSet, Aladin VO software and SAMP-compliant astronomy tools.

Resource Management: VOSpace provides a common interface to an underlying heterogeneity of distributed storage implementations.



References

AENEAS web page https://www.aeneas2020.eu

AENEAS D5.1 **Survey report** [link]

AENEAS D5.2 **Gap analysis** [link]

AENEAS D5.4 **Recommendations on the design of user interfaces for data processing, re-processing, analysis and visualization for the ESDC** [link]

AENEAS D5.5 **Applicability of VO framework** [link]