SRC Science Analysis Platform Vision

Cimpan, Das, Fabbro, Grange, Hardcastle, Horton, Sharma, Swinbank, Webster
To be filled in at a later stage

*corresponding author

Abstract. This document describes the vision for the Square Kilometer Array (SKA) Regional Centre Science Analysis Platform. It is intended to set the broad terms of reference for the platform and to provide guidance for both development teams and other stakeholders.

Version: **0.8**

Contents

1 Introduction and Context 3
1.1 Purpose and Status of this Document . . . 3
1.2 What is a Science Analysis Platform? . . . 3
1.3 Platform Aims and Objectives . . . 3
1.4 The design of the platform . . . 5
1.5 Accessibility and Inclusion . . . 5

2 Back-end Features and Services 6
2.1 Compute services . . . 6
2.2 Archive and distributed data . . . 6
2.3 User file and database services . . . 6
2.4 Authentication and Authorization . . . 7

3 User-facing Features and Services 7
3.1 Main User Interface . . . 7
3.2 Data Querying and Discovery . . . 8
3.3 Notebook Interface . . . 8
3.4 Software environments . . . 9
3.5 Web APIs . . . 9
3.6 Workflow Management . . . 10
3.7 Resource Management . . . 11
3.8 Groups . . . 11
3.9 Software repository . . . 11

4 Connections/dependencies on other SRC development/prototyping teams 12

5 Summary 13

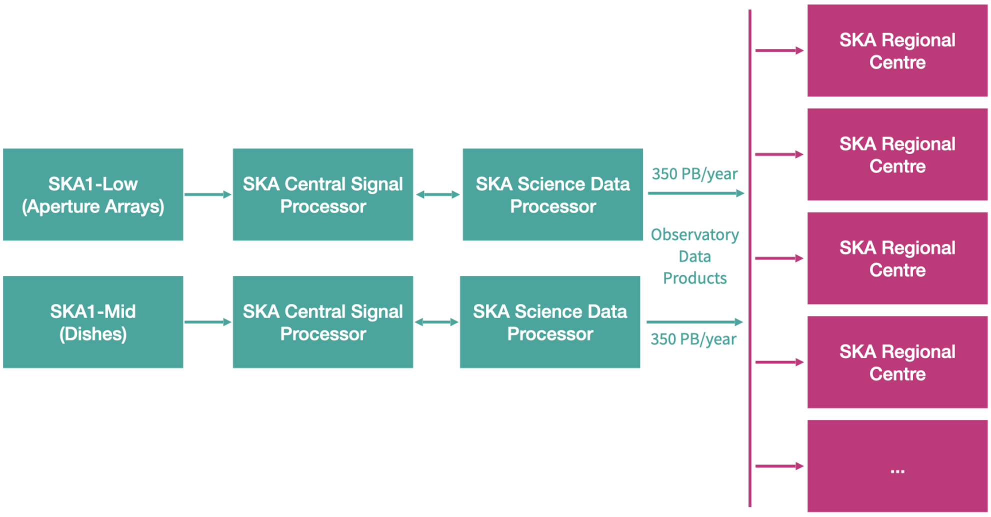
SKA1-Low (Aperture Arrays)
SKA Central Signal Processor
SKA Science Data Processor
350 PB/year
Observatory Data Products
SKA1-Mid (Dishes)
SKA Central Signal Processor
SKA Science Data Processor
350 PB/year
SKA Regional Centre
SKA Regional Centre
SKA Regional Centre
SKA Regional Centre
...

Figure 1: The relationship between SRCNet and the Square Kilometer Array Observatory (SKAO).

1 Introduction and Context

1.1 Purpose and Status of this Document

This document describes the community vision for a Science Analysis Platform intended to be delivered across the SKA Regional Centre Network (SRCNet). This platform will provide a consistent interface between SKA users and their data, while offering enough flexibility for individuals to customise the platform to meet their own needs.

As a vision statement, this document sets out the broad terms of reference for the platform. It is not a set of detailed requirements or specifications, and nor is it a design document. Rather, it sketches a high-level view of the necessary platform functionality and identifies key interfaces with other SRCNet technologies and areas of development. It is intended to serve as a reference for both development teams and other stakeholders.

The document has its origin in discussions undertaken within the responsible SRCNet prototyping team ("Team Tangerine") during summer 2022. It is not the result of a formal requirements gathering or use-case solicitation exercise; rather, it draws on the expertise of team members, inputs from other members of the SKA Regional Centre (SRC) development community, existing requirements on Confluence and a wide-ranging literature review covering the philosophies and practicalities of many existing science platforms.

This is expected to be a *living document*: it will be updated over time to reflect lessons learned during development and refined understanding of user needs.

1.2 What is a Science Analysis Platform?

A science platform provides scientists with a computer environment that permits the collaborative handling of large and diverse datasets and allows them to access large-scale computing resources that they may not have locally available. Science platforms are often linked to a specific project, instrument or method of analysis and are typically designed to provide consistency for all users while providing access to appropriate tools and data. Thus, a science platform must provide features that are relevant to the user base and avoid any pitfalls that will make it difficult for scientists to complete their work.

1.3 Platform Aims and Objectives

The SKAO will generate around 700 PB/year of science-ready Observatory Data Products (ODPs), including — amongst others — image cubes, uv-grids, calibrated visibilities, and pulsar timing solutions [1]. However, these products will not directly be made available to the scientific community. Instead, as illustrated in Figure 1, ODPs will be provided to a worldwide network of SRCs [2]. The SRC network then assumes responsibility for the following user-facing functionality[3]:

- Data logistics, including making data available to end users;
- Data processing, including providing computational and storage resources and appropriate software environments to enable end users to interact with the data and produce Advanced Data Products (ADPs);
- Data archiving and curation, including data discovery and re-use;
- User support.

The platform will be the primary mechanism by which end users interact with the SRC network. As such, it will provide users with a unified place where they can come to search for and discover SKA data (using the functionality described in §3.2), to perform interactive data analysis tasks (§§3.3 & 3.4), and to schedule and manage long-running or complex workflows (§3.6). The platform will also provide capabilities for users to collaborate, sharing and publishing both their analysis workflows and their results in the form of ADPs, as described in §3.7. All of this functionality will be available both through a modern and accessible user interface (§3.1) and through an API which will facilitate access from user scripts and other automated tooling (§3.5).

The Platform will be designed and built to comply with the high-level SRC system requirements, and to follow the SRCNet Architectural Principles[1]. In particular, we highlight the following:

- The platform should provide accessible and straightforward interfaces, where possible adopting paradigms which are already well established within the astronomical community, to ensure that SKA data is available to the widest possible user base;
- The platform should provide expert users with maximal flexibility in deploying sophisticated custom tools and provide them direct access to low-level data products;
- The platform should provide appropriate interfaces to the lower-level services and tools deployed across the SRC network;
- The platform should provide seamless access to available services regardless of the physical location of the user and/or of the SRC node providing a particular service;
- The platform should follow Findability, Accessibility, Interoperability, and Reusability (FAIR) principles by ensuring that all ADPs are generated through reproducible workflows and are accompanied by appropriate metadata describing their provenance;
- The platform should provide abstractions so that users are appropriately insulated from the details of the underlying SRCNet infrastructure;
- The platform should only provide users with access to services and/or data products to which those users are entitled.

[1] https://confluence.skatelescope.org/display/SRCSC/SRCNet+Architecture+Principles

1.4 *The design of the platform*

At the time of writing, we expect that the set of interfaces constituting the SKA Science Platform will be primarily web-based, with both interactive and programmatic access, but the design should also be capable of responding to new standards as they evolve. The design and operation of the SKA Science Platform will provide the following five attributes:

- *consistency*: the user experience should not be dependent on location or the available infrastructure;
- *scalability*: a resource-intensive analysis must finish on a manageable timescale;
- *reproducibility*: any analysis produced on the platform can be reproduced at any later time;
- *usability*: interfaces should be usable by a user base with a wide range of skills across a wide variety of local compute resources and accepted standards;
- *reliability*: interfaces should be fully functioning at all times and should provide transparency on a user's resource usage.

Reliability and scalability are dependent on the underlying components of the SRC network and compute infrastructure. The overall science platform experience will pass on information about the state of this infrastructure to its users. Implementing all five of these attributes *simultaneously* will be challenging but is necessary since doing so will ultimately mean that the science platform is valued as the first choice analysis platform by SKA scientists.

We have also developed guiding principles for the design of the platform. Specifically we emphasize the following principles:

- *highly collaborative*: The platform will allow the sharing of workflows, data and code in order to bring teams together;
- *end-to-end*: Scientists will be able to submit and refine proposals, perform (collaborative) analysis, and publish their data products and analysis code in a transparent way (e.g. with DOIs) that follows FAIR [4] principles, thus being supported through all stages of the lifecycle of a scientific project;
- *accessible*: The platform's interfaces will be configurable to provide maximum accessibility to the diverse scientific user base (see §1.5).

1.5 *Accessibility and Inclusion*

Proper consideration of accessibility issues are crucial to the design of the platform and its user interface. This has two aspects:

1. The platform must facilitate individual accessibility requirements by making sure everyone can use, adjust and configure the user interface to suit their own needs. Examples may include choice of interaction languages, colour filters, screen readers, text scaling among others. Since our expectation is that scientific users will be interacting with SKA data almost exclusively through this platform (see §1.3), we need, to the extent possible, to allow the user to have as much control over the web interface presented to them as they would expect to have over a well-written local application. As legislation and best practice will vary throughout the network, it will be essential to keep up-to-date with evolving web accessibility guidelines[2], relevant laws and community feedback.

[2] e.g., `https://www.w3.org/TR/wcag-3.0/`

2. The platform must be be usable to astronomers from a wide variety of backgrounds, specialisms, and career stages, and must therefore present an interface that is accessible to all of those groups, again potentially through the use of configurations that allow the system to be customized to meet the needs of both novice and highly experienced users.

In order to meet these needs the users will need to feel supported, directed towards the right resources, and helped, inspired and motivated by collaborating closely with them through user support, workshops, documentation and user-friendly applications.

2 Back-end Features and Services

In this section we briefly describe the back-end features required for an SRC node in order to provide the context for the user-facing aspects that will then be discussed in Section 3.

2.1 Compute services

Analysis through the SRCs will be distributed over different regional, national or supra-national compute infrastructures. Computing requirements are varied and include distributed processing of very large datasets for generation of ADPs or even re-running of the SKAO's Science Data Processor (SDP) pipeline, as well as user-driven batch processing or interactive processing via, for example, a notebook. The responsibility for resource management will necessarily be shared between the SRC and the local resource providers, with the overall SRC allocation being predetermined by the governance policies of the different countries participating. In principle, the size of the computing resource allocation to an SRC should be determined by SKA science requirements. Resource allocations to individual users will then be the responsibility of the SRC node. An important feature of the distributed nature of the SKA archive is that it will in many cases be more efficient to generate an ADP on the SRC node that is local to the data, rather than the one that is local to the user: such remote execution of standard workflows will need to be transparent to the user but will also need to be accounted to the user to ensure fairness of resource allocation.

2.2 Archive and distributed data

As noted above, SRCs will need to provide an archive that stores the ODPs, together with the functionality needed to allow users to query the archive for ODP, retrieve them for further processing and store generated ADP. In addition, the generation of an ADP will typically require 'staging' an ODP from the archive to faster distributed storage that permits local processing, for example to make cutouts or frequency-average a large data cube. Although this process will largely be abstracted away from the point of view of the end user, the fast storage required consumes resources on a per-user basis which will need to be tracked by any resource allocation system.

2.3 User file and database services

Users will be provided with a persistent, personal Portable Operating System Interface (POSIX)-like file system, to which they can upload and download files at will, to allow them to storeADPs for further processing and visualization as well as to hold code and additional uploaded data for analysis. Metadata including processing logs will be stored along with the data for all processing operations and should record information on software and resources used and any other parameters to ensure reproducibility of ADP generation. The platform will give access to a set of discovery services to make metadata and analytics easily findable.

The file system will allow fine-grained sharing of files or directories for collaboration purposes, making use of the Authentication and Authorization Infrastructure (AAI) service (§2.4).

Users will also have access to database services, which will come in two forms:

1. Centrally managed databases will be created and made available for the released data products, detected sources etc. depending on various science cases. The users will be able to run queries through Table Access Protocol (TAP), Astronomical Data Query Language (ADQL) etc. providing both synchronous and asynchronous query modes depending on datasets. The TAP-like queries will support uploading the table to user space from multiple inputs like URLs, files, tables from a job, and tables from an astropy table. These are implemented in astroquery[3] and CADC user database[4], and can be built on.

2. Users can generate their own databases through similar TAP, ADQL-like queries, and can store structured data relevant to their science cases.

Again, database rights should be sharable between groups of users.

2.4 *Authentication and Authorization*

The SRC network will provide AAI services which will provide mechanisms for verifying user identity ("authentication") and establishing the services and data that each user has the right to access ("authorization")[5].

The platform will integrate with this AAI system. In particular:

- Access to the platform itself will be subject to appropriate authorization;
- All services and data that are offered to or accessed by users through the platform will be subject to appropriate authorization;
- platform-provided user interfaces and APIs will provide seamless interoperability with the AAI system.

In general, the platform will not offer users access to services or data to which they do not have rights; where appropriate, opportunities for acquiring elevated privileges may be indicated.

Ultimate responsibility for controlling the access to data or services is the responsibility of the system which hosts the data or service. For example, the underlying storage system, not the platform, is the final arbiter of whether a given user can access a particular data element. However, the platform assumes responsibility for forwarding user credentials to appropriate systems and presenting the results of data or service access operations to platform users.

3 User-facing Features and Services

3.1 *Main User Interface*

The main user interface for the platform will be a web page (the 'portal') providing access to the functionality of the SRC node in a set of panels that reflect different modes of access to the node. The user will sign on to the portal on the front page using single sign-on criteria (§2.4) and will then be given access to the full user interface. The user interface will be consistent across different SRC nodes, although in limited

[3] https://astroquery.readthedocs.io/en/latest/utils/tap.html

[4] https://ws.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/youcat/

[5] https://jira.skatelescope.org/browse/SRC-147

circumstances SRC nodes may make minor modifications to the portal to make local resources available.

Many users will simply want to access either their own data or data from the archive, e.g. by obtaining previews of images or catalogs or downloading reduced-volume datasets directly to their own computer. Simple data-querying and discovery (§3.2) will be the default panel, allowing users who do not wish to carry out more sophisticated analysis to go straight to the data. Once a dataset has been selected, users will need to be able to visualize the data (images, spectra, cubes, catalogs, light curves etc) either internally to the platform or by spawning an external application. Other panels will provide access to a notebook for interactive analysis (§3.3) and the ability to run Virtual Machines (VMs), containers, or distributed jobs (§3.4) as well as constructing more complex workflows (§3.6). The user will have access to information covering their resource usage, either as a single user or a member of one or more groups (§3.7).

Many elements of the user interface of the portal will be configurable by the user to support (1) the user's particular accessibility needs (Section 1.5) and (2) the user's ability to simplify access to the tools that they are most likely to need.

3.2 Data Querying and Discovery

In order to maximize the science potential of the data held within the SRCNet archives, users will be provided with a web-based search engine. Any web-based queries entered by the user will call an underlying API that will process the data and provide the results. The API will be publicly available so that users can execute archive searches and examine the results within other environments such as a notebook.

The archive search will allow users to search both the ODPs and publicly available ADPs as well as compatible non-SKA archives. Since the amount of data output by the SKA is expected to be enormous, it is important that users are able to perform targeted, restrictive searches of the archive to enable them to quickly focus on the data of interest without having to potentially download or stage and search through large amounts of unnecessary data. Users will be able to create simple searches using web forms as well as entering more advanced ADQL-type queries.

Search results will be presented to users in tabular form with the ability to access the associated data product. In the case of image data users will be able to access either the whole file or make a cutout. Cutouts can be made in both spatial and/or frequency domains.

In order for the search algorithms to work correctly, each archived data product will require associated metadata. Therefore, whenever an ADP becomes publicly available it will be stored in the archive along with the associated metadata, which will include links to the code used to generate it (§3.9). Once data is submitted to the archive, it will not be possible for users to alter the archival contents, although it will be possible to upload a new version of the data to the archive.

3.3 Notebook Interface

The notebook interface enables the user to do science in their own web browser by creating and running notebooks, defined broadly as structured interactive environments combining executable code, text and visualization; at the time of writing the predominant notebook type in the astronomy community is the Jupyter notebook, but the platform will need to be capable of evolving to meet new community standards.

Notebooks will run 'next to the data' [5] within the science platform and will offer user environments with preinstalled packages that provide the functionality required by a standard scientific user, including the generation of standard ADPs from ODPs as well as the ability to customize the environment by installing new packages. The results of data queries or discovery (§3.2) will be available within the notebook environment, and notebooks will be persistent for each user. Thus the user experience will be similar to that of running a notebook on their own local compute resources but they will have transparent access to the much larger compute resources provided by the SRC.

Sharing of notebooks will be a convenient way for users to collaborate on analysis tasks in a reproducible way and this will be linked to a software repository (§3.9) to allow for well documented and clear software sharing.

3.4 Software environments

The platform will provide, through its web interface, the possibility for researchers to launch, provision, manage, build, and share customized environments that include complete software dependencies for running complex applications [6]. Technologies for providing these at present include VMs and containers in systems like Docker or Singularity, which will run on top of the computing resources provided to the SRC node (§2.1). Researchers should be able to run code in a specific environment, extend it and resubmit it as a new environment and also create snapshots of a running instance of an environment for reproducibility. Software environments used as part of a scientific analysis should be stored in the software repository (§3.9) and should be linked to the metadata of datasets that they generate. Notebooks (§3.3) are likely to be implemented as a specialized version of a software environment. More generic environments will be accessible through web console or web-based graphical user interfaces, and the option of opening an Secure Shell (SSH) connection will be provided.

We envisage that the platform will be based on a cloud computing paradigm and will support users in launching instances of virtual machines preconfigured with a specific operating system and software. This then requires the platform to provide the users with resources to develop cloud-based virtual machine images, and allowing the users to have their own individual space for testing and development.

The benefits of providing [7] an environment focused on container and cloud native development services for researchers are:

- The possibility for researchers to launch virtual machine images preconfigured with software suites, work environments, and community-contributed software, whether designed by the user themselves or taken from the pre-existing software repository;
- Providing users with already installed specific versions of different packages/software and containing established astronomy tools such as DS9, HEASoft, CASA, etc.
- Available statistical reporting for tracking user resources like memory usage, total CPU hours, number of instances and applications launched by the user, and manageable resource allocation (§3.7)
- Connecting to more powerful distributed computing environments (e.g. through an on-demand Slurm or Kubernetes cluster)
- Integrating with other infrastructure components through API services (§3.5).

3.5 Web APIs

It is important that APIs are publicly available to interact with the system. These APIs will serve two purposes:

- They enable remote discovery of, and access to, all data products, including access to a user's proprietary and private data. This will allow the results to be combined with data from other facilities.

- They enable users to remotely access all low-level services and functionality provided by the SRC. This will allow users to create new tools built on top of the SRC network.

Machine-accessible web APIs should handle the access to databases, images, and other files which will be exposed to the public internet. These will make remote data access easy. However, we will go beyond this and follow the LSST approach [8] of making all low-level platform functionality available through these APIs with suitable authorization. The interfaces provided by e.g. notebooks (§3.3) will then internally be wrappers around APIs.

Accessing the SKA data through Virtual Observatory interfaces will be crucial. This will allow the discoverability of SKA data products from within the Virtual Observatory. It will also enable the easy use of widely utilized tools such as DS9 and CASA by the end-users, which will make it easy to access the SKA data, and shortening the path to science. It will also allow these tools to be used in commissioning, easing the way for scientists new to the SKA environment to access the data and make meaningful contributions to this time-compressed activity.

Initially, we will implement the TAP[6] for accessing data in tabular form, the Simple Image Access Protocol (SIAP)[7] to provide capabilities for discovery, description access and retrieval of datasets, the Simple Spectral Access Protocol (SSAP)[8] for discovery and access to spectral data and Server-side Operations for Data Access (SODA)[9] for low-level data access or server side data processing. Users will also be able to use the ADQL[10] when posting queries to the APIs.

3.6 Workflow Management

The science platform will facilitate workflow management with a tool that will allow users to specify individual workflow steps and combine them to form a larger workflow which can be stored in the software repository (§3.9) and re-used by others. It will be possible to combine workflow steps sequentially as well as using logical operators. Each workflow step will consist of one or more steps that can be taken using other parts of the user interface, such as code within a notebook, a call to one of the pre-defined APIs, a call to a piece of software (e.g. CASA) that is pre-installed within an existing defined software environment, or a call to another workflow, including SDP and/or publicly available workflows.

After a workflow is defined, there may be options to run the workflow either in real-time or as a background process that can be scheduled when the system is quiet, with the user being notified upon completion/failure. This will be dependent on the underlying architecture of the local SRC node. Using the workflow management tool to define workflows will offer the following advantages:

- For any workflow, the system can automatically generate a list of parallelisable steps that will need to be executed;

- It is anticipated that the size of the ODPs will mean it will normally be significantly more efficient to transfer the code for an individual workflow step to the centre containing the data and execute it there before transferring the results back. However, this may not always be the case, so that for each individual step the system can assess whether to transfer the code or the data between centres;

[6] https://www.ivoa.net/documents/TAP/

[7] https://www.ivoa.net/documents/SIA/

[8] https://www.ivoa.net/documents/SSA/

[9] https://www.ivoa.net/documents/SODA/20170517/REC-SODA-1.0.html

[10] https://www.ivoa.net/documents/ADQL/20180112/PR-ADQL-2.1-20180112.html

- Use of background processing to schedule larger workflows when the system is quiet will improve the overall usage of limited resources.

Each workflow will have access to: (1) all ODPs and ADPs for that user; (2) workflows made available by the SRC and (3) the user's file storage area, among others.

3.7 Resource Management

Users will necessarily have allocations of compute and storage resources on the computing infrastructure underpinning the SRC node, as discussed in §2.1 and §2.3. These allocations will be set by local policy, but the resources allocated and used should be visible in a consistent way across SRC nodes. Ideally, users with resource allocations across multiple SRC nodes will be able to see all allocations from all nodes.

A user is likely to use their resource allocation in two ways:

1. They may make a high-level data discovery/processing request which requires resources that will be determined by the system. For example, the staging of an image cube ODP from the archive to nodes capable of performing a reduction operation will consume storage; the actual reduction operation, such as averaging in frequency space or extracting a sub-cube, will consume CPU resources, most likely on numerous compute nodes that the user does not normally interact with directly.

2. They may carry out further operations on the ADPs that are the results of the previous step: for example, a user may request that the sub-cube from the previous step be moved to their private, persistent POSIX-like work space (§2.3), which would consume their storage allocation, and spin up a VM (e.g. by opening a notebook) to carry out further operations on it using standard tools, consuming CPU and also allocations from a VM pool.

Resource utilization information will need to make the user's current consumption of these different types of resources clear to them and allow them to take steps to reduce their usage if necessary. For example, the user will need to be able to see 'their' current staged ODPs and take the decision to remove some of them from hot storage to free up capacity for another task. They will need to be able to see and if necessary terminate background operations that they have initiated, such as staging ODPs or processing them into ADPs. If they have multiple VMs running, they will need to see their overall resource allocation as well as the CPU consumption on each VM.

3.8 Groups

To allow collaborative working, users must be able to set up, join and leave groups of other users dynamically (e.g. through an invitation to collaborate system similar to GitHub's) and to allow different levels of access to their personal workspace, their holdings in the software repository and their proprietary ODPs and ADPs to group members. Current group memberships and the resources that they have been granted access to will be under user control.

3.9 Software repository

Code written for scientific analysis and the 'software environment' used to generate it (§3.4) must be preserved essentially as part of the metadata of the ADPs or user-generated datasets that it produces. Therefore, all such code and environment descriptions (which may internally be VM or container build instructions) must be held in a software repository which is integrated with the platform and transparently keeps track of user changes. The repository will be searchable and queryable in the same way as the data archive, and will be shared across SRC nodes.

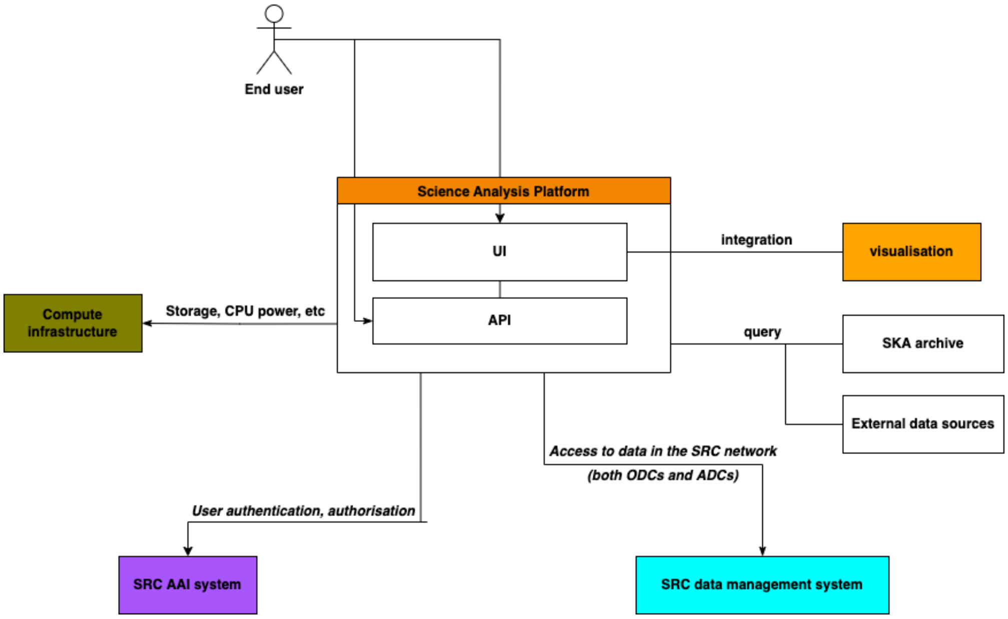
End user
Science Analysis Platform
UI
API
integration
visualisation
query
SKA archive
External data sources
Compute infrastructure
Storage, CPU power, etc
Access to data in the SRC network (both ODCs and ADCs)
User authentication, authorisation
SRC AAI system
SRC data management system

Figure 2: Dependencies between the science analysis platform and other components of the SRC network.

4 Connections/dependencies on other SRC development/prototyping teams

The science analysis platform component will be the primary interaction point between the SRC system and its users. This means it is a component that has connections with or dependencies on all other components being developed in the SRC context. In this section we give an overview of those dependencies by linking the components as discussed in this document to the other components that are going to be part of the SRC network.

The relations are visualized in Fig. 2. The top-left (olive-colored) block represents the compute hardware, but also the method of brokering between resource requests and available resources, and to move the compute jobs to the appropriate system. This component is linked to the Notebook, which is the primary interface between users and the compute hardware, as described in Sec. 3.3.

The AAI system, which makes it possible for the SRC network system as a whole to provide users with access to data and compute resources as described in Sec. 2.4, is represented by the bottom-left (purple) block. The link with the data management system should provide both query functionality and functionality to make data products (both ADPs and ODPs) to be copied to locations close to the compute hardware, as well as re-upload end results system (ADPs).

The orange block on the right-hand side shows the visualization system. Visualization is one of the key components of the SRC system. The visualization will integrate with the query functionality, so that image results can be displayed when querying, or to display tables of data found in a query, see Sec. 3.2. Visualization will also need to be integrated in the notebook component (described in Sec. 3.3) so that end products can be visually inspected by users.

5 Summary

The volume of the data collected by SKAOs will be in petabyte scale. Hence, the science platforms used for SKAOs should support data mining and machine learning, and they need to be developed based on advanced technologies for large data handling with the support and active involvement of the astronomy community.

The computing services of the SRCss will include regional, national or supra-national compute infrastructures. The amount of computing resource allocation to anSRC should be determined bySKA science requirements. SRCss will also need to provide an archival ODP storage, which should allow users to query the archive for ODP, retrieve them for further processing and store generated ADP. Users will be provided with a personal POSIX-like file system, to upload and download files and to store ADPs for further processing and visualization as well as to hold code and additional uploaded data for analysis. Centrally managed databases will be created and made available for the released data products, detected sources etc. Users will also be able to generate their own databases through similar TAP, ADQL-like queries, and can store them. The SRC network will provide AAI services for authentication and authorization. However, the platform will not offer users access to services or data to which they do not have rights.

The main user interface for the platform will be provided by a web page which will allow access to the functionality of the SRC node in a set of panels. Users will be able to sign on to the portal on the front page to access the full user interface. Many elements of the user interface of the portal will be configurable by the user. Users will be provided with a web-based search engine to query data using an underlying API that will process the data and provide the results. The archive search will allow users to search both the ODPs and publicly available ADPs as well as compatible non-SKA archives. Users will be provided a notebook interface which will allow them to do science in their own web browser by creating and running the notebooks. Sharing the notebooks will be a convenient way for users to collaborate with other and this will be linked to a software repository. The platform will provide, through its web interface, the possibility for researchers to launch, provision, manage, build, and share customized environments that include complete software dependencies. These will be provided in the form of VMs and containers in systems like Docker or Singularity, which will run on top of the computing resources provided to the SRC nodes. Users can use their resource allocation either to make a high-level data discovery/processing request or to carry out further operations on the ADPs that are the results of the previous step. Users should have clear knowledge of resource utilization information. In order to collaborate with others, users must be able to set up, join and leave groups of other users dynamically. All the code written for scientific analysis and the ‘software environment’ used must be preserved essentially as part of the metadata.

List of Abbreviations

AAI Authentication and Authorization Infrastructure. 7, 12, 13

ADP Advanced Data Product. 4, 6, 8, 11–13

ADQL Astronomical Data Query Language. 7, 8, 10, 13

API Application Programming Interface. 4, 7–10, 13

FAIR Findability, Accessibility, Interoperability, and Reusability. 4, 5

ODP Observatory Data Product. 3, 4, 6, 8, 10–13

POSIX Portable Operating System Interface. 6, 11, 13

SDP Science Data Processor. 6, 10

SIAP Simple Image Access Protocol. 10

SKA Square Kilometer Array. 1, 3–6, 8, 10, 13

SKAO Square Kilometer Array Observatory. 3, 6, 13

SODA Server-side Operations for Data Access. 10

SRC SKA Regional Centre. 3–13

SRCNet SKA Regional Centre Network. 3, 4, 8

SSAP Simple Spectral Access Protocol. 10

SSH Secure Shell. 9

TAP Table Access Protocol. 7, 10, 13

VM Virtual Machine. 8, 9, 11, 13

References

[1] Shari Breen, Rosie Bolton, and Antonio Chrysostomou. SKAO Science Data Products: A Summary. Technical Report SKA-TEL-SKO-0001818, SKAO, 2021.

[2] A. Chrysostomou and the SRCCG. SKA Regional Centres: Background and Framework. Technical Report SKA-TEL-SKO-0000706, SKAO, 2017.

[3] Peter Quinn, Michiel van Haarlem, Tao An, Domingos Barbosa, Rosie Bolton, Antonio Chrysostomou, John Conway, Séverin Gaudet, Hans-Rainer Klöckner, Andrea Possenti, Simon Ratcliffe, Anna Scaife, Lourdes Verdes-Montenegro, Jean-Pierre Vilotte, and Yogesh Wadadekar. SKA Regional Centres: A White Paper. https://confluence.skatelescope.org/display/SRCSC/SRCSC+White+Paper+V1.0, May 2020.

[4] Mark D. Wilkinson, Michel Dumontier, Ijsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 'T Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3:160018, March 2016.

[5] Rubin Science Platform Notebook Aspect documentation. https://nb.lsst.io/.

[6] Nirav Merchant, Eric Lyons, Stephen Goff, Matthew Vaughn, Doreen Ware, David Micklos, and Parker Antin. The iplant collaborative: cyberinfrastructure for enabling data to discovery for the life sciences. *PLoS biology*, 14(1):e1002342, 2016.

[7] Atmosphere. https://cyverse.org/atmosphere.

[8] M. Jurić, D. Ciardi, G.P. Dubois-Felsmann, and L.P. Guy. LSST Science Platform Vision Document. https://lse-319.lsst.io/, 2019.