SRC | Net

SKAO Regional Centre Network

SRCNet v0.1 Implementation Plan

Document Number	SRC-0000009
Classification:	UNRESTRICTED
Document type:	PLN
Revision	01
Date:	2024-06-11
Status:	RELEASED
Authors:	Salgado, Jesús; Joshi, Rohini; Sánchez Expósito, Susana; Parra, Manuel; Walder, James; Llopis, Pablo; Guo, Shaoguang; Morris, Dave; Taffoni, Giuliano; Gheller, Claudio; Ouellette, John; Kang, Hyunwoo; Gaudet, Séverin; An, Tao; Akahori, Takuya; Swinbank, John; Hess, Kelley Michelle; Oonk, Raymond; Kok, Tim;

le	Name	Designation	Affiliation	Signature	Date
Author	J. Salgado	SRCNet Software Architect	SKAO	Jesus Salgado	2024-06-24
Owner	J. Salgado	SRCNet Software Architect	SKAO	Jesus Salgado	2024-06-24
Approver	R. Bolton	SRCNet project Lead	SKAO	R. Bolton	2024-07-02
Released by	L. Ball	SKAO Director of Operations	SKAO	Lewis T Ball.	2024-06-26

TABLE OF CONTENTS

Status of the Document

This is an SRCNet released document developed by the SRCNet Architecture Forum members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time.

Acknowledgement to the Tiger Team members for the generation of the consolidated draft: Rohini Joshi, Susana Sánchez Expósito, Manuel Parra, James Walder, Pablo Llopis, Shaoguang Guo, Dave Morris, Giuliano Taffoni, John Ouellette, Séverin Gaudet, Hyunwoo Kang and Claudio Gheller.

1. Scope

This document outlines the deployment plan for the initial version (v0.1) of the SKA Regional Centre Network (SRCNet). SRCNet aims to establish a collaborative network of computing, storage and network resources distributed across geographically diverse international facilities participating in the Square Kilometre Array (SKA) project and aims to create a shared service supporting radio astronomers dealing with SKA data.

This document details the initial deployment scenario for v0.1, which prioritises participation by including all SRCs that expressed an interest in contributing resources. This approach fosters a sense of ownership and investment from the outset while ensuring broad representation from the international SKA community. However, it also presents challenges associated with managing a large, geographically dispersed network, particularly during the initial phase with evolving software and processes.

This document will be a guideline for the activities to achieve the implementation on the SRCNet v0.1 and execute the first set of tests confirming its successful operation. Its target audience is mainly technical staff at the SRC nodes and the SRCNet Architecture Forum members.

The following sections of this document elaborate on the chosen deployment scenario; analyse potential risks and mitigation strategies; an overview of the SRCs involved in SRCNet v0.1, including a description of their resources and relevant milestones for their integration into the SRCNet v0.1; and provide a detailed technical breakdown of the planned SRCNet architecture. This includes specifications for software stacks, service deployments across SRCNet nodes, and considerations for compute and storage resource configuration.

This document covers:

- Description of resources available per SRC for the SRCNet v0.1 production network
- Description of software stack elements and how these elements will be deployed on the SRCNet
- Description of intended operations and deployment procedures for SRCNet v0.1
- Description of nodes validation procedures and SRCNet v0.1 test campaigns
- Intended main milestones for the deployment of the network and operations

This document does not cover:

- Procurement and purchase orders from the SRCs
- Detailed operations plan
- Full list of functional and benchmarking tests to be run to validate the nodes and the SRCNet v0.1 network
- SRCNet governance details

1.1 Audience

The primary target audience for this document is engineers responsible for the deployment, operations, and maintenance of the SRCNet nodes. Additionally, engineers involved in the development of services and software elements for the SRCNet v0.1 software stack are also intended recipients of this information.

2. SRCNet v0.1 Timeline[1]

This section provides a concise overview of the key milestones achieved during the preparation and deployment of SRCNet v0.1. It also presents preliminary details concerning the test campaigns planned for this version. The subsequent sections of this document will furnish comprehensive information on the available resources, the software stack employed, the deployment methodologies utilised, and the operational procedures for managing this specific iteration of SRCNet.

Start Date	Deadline	Event	Description
3-May-2024	3-May-2024	Distributed Data Management Selection	Assessment and election between the two prototypes for the DDM
8-May-2024	30-Jun-2024	Selection of specific software for each of the compulsory services	
14-May-2024	30-May-2024	Implementation Plan Release	Release of the SRCNet v0.1 Implementation Plan including the identification of features to be implemented during next PIs
	30-Jun-2024	Operations Group Set-up	Initial operations group identification and set-up, establish Terms of Reference

1 https://docs.google.com/spreadsheets/d/1QxkX3XFXFprV4Pun6DThu9S62AYxcK82ghAaUknu8nM/

1-Jun-2024	1-Sep-2024	Compilation of infrastructure level tests	Compilation of infrastructure level tests (storage, network, compute exists)
15-May-2024	31-Aug-2024	Compilation of deployment procedures of compulsory services software	The software corresponding to the compulsory services will be compiled in a common software repository, which will be shared among the SRC nodes. It will contain enough documentation on how to install each of these services, including the deployment technique (see section 6). This repository will gather the source code / containers of the software developed within the SRCNet collaboration as well as links to external repositories for third party software
17-Jun-2024	1-Oct-2024	Execution of infrastructure level tests	These tests will provide useful information for the nodes on the status of their computing infrastructure. The aim is to identify potential bottlenecks that could be solved during the deployment and configuration of the hardware and computing services
	10-Jun-2024	Define ingestion nodes	Define requirements for SRCNet Ingestion Nodes

	24-Jun-2024	SRCNet Baseline Policies	Establishment of Baseline SRCNet Policies
1-Jun-2024	30-Sep-2024	Define the test campaigns	Data lifecycle tests are the closest tests to the behaviour expected by scientists, so they will provide good metrics of the behaviour of the SRCNet and, also, support the identification of bottlenecks. We will want to run these tests during SRCNet0.1 testing campaigns (2025) so need to build out these tests in PI23 and PI24.
1-Jun-2024	31-Aug-2024	Building service monitoring infrastructure	The common software repository will also include the source code / containers and/or links to external repositories documentation to deploy the monitoring services on the nodes in order
30-Jun-2024	1-Nov-2024	Deployment of the Service Monitoring tests	
1-Oct-2024	1-Nov-2024	SRCNet v0.1 Deployment on Nodes Started	Deployment of compulsory services on validated SRCNet nodes started
	1-Nov-2024	Operations Administration Portal	First version of the Operations Administration Portal
3-Jul-2024	30-11-2024	Execution of Integration testing	Ingestion and dissemination tests and data transfer tests as sites get onboarded. Test data (precursor data/synthetic test data) is ingested using the ingestion service, and replicated to sites using a set of sites.
1-Dec-2024	31-Dec-2024	SRCNet v0.1 Preliminary Data Ingestion	Ingestion of a subset relevant data sets to enable integration testing. SP-4256 will feed into this.
1-Nov-2024	31-Jan-2025	SRCNet v0.1 beta release	Release of beta versions and testing
31-Jan-2025	31-Jan-2025	SRCNet v0.1 Formal Release	Release of operational version

28-Feb-2025	31-Mar-2025	Execution of Data Transfer test Campaigns	Data ingestion campaigns to identify bottlenecks and data transfer dissemination performance.
1-Apr-2025	30-May-2025	Execution of Data lifecycle test Campaigns	SRCNet data lifecycle campaigns, including execution of workflows.

Table 1: SRCNet v0.1 events time line in tabular format. In blue, deployment activities. In green, testing phases. In yellow, milestones

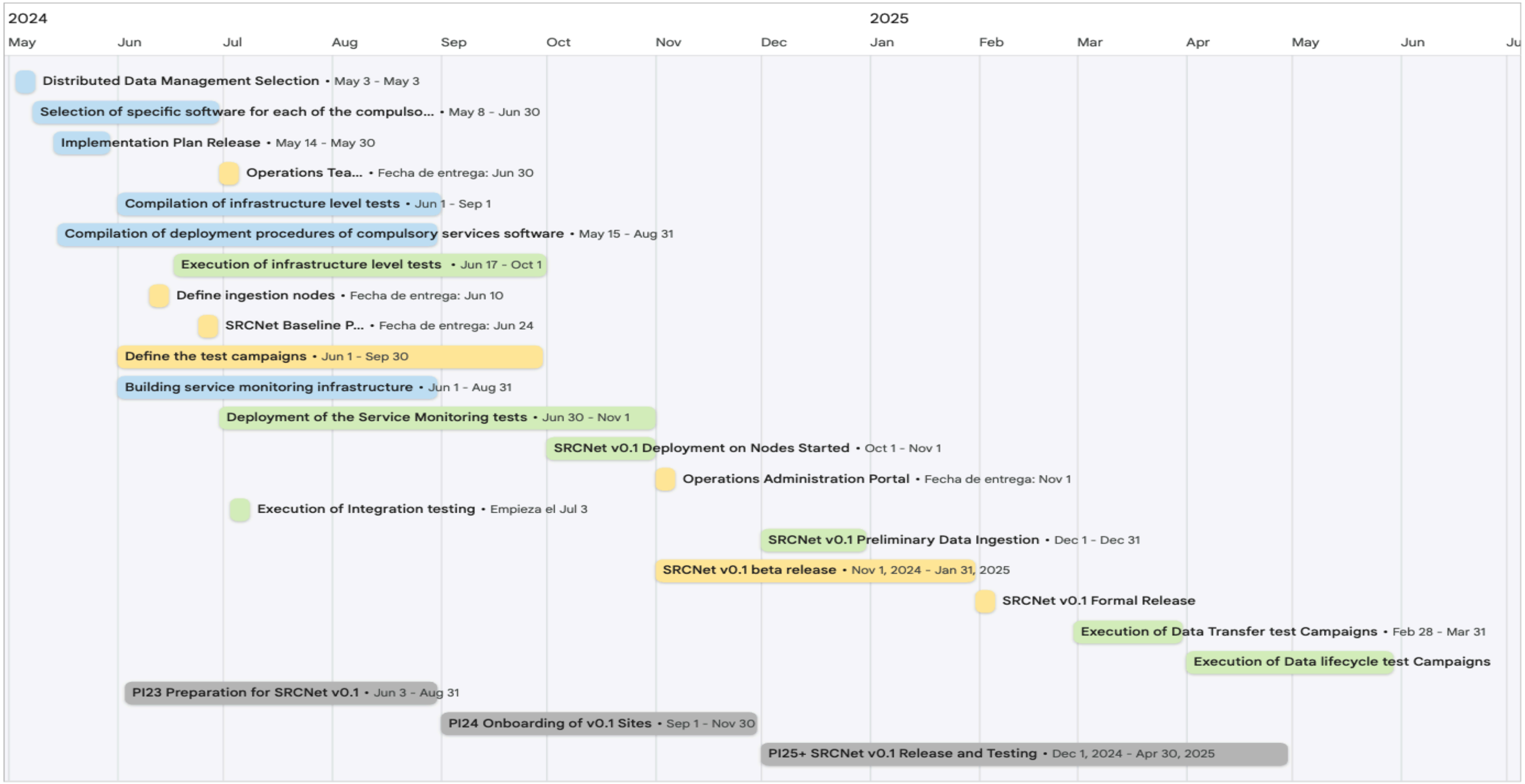

Figure 1: SRCNet v0.1 events timeline: In blue, deployment activities. In green, testing phases. In yellow, milestones.

3. Scenario

As per discussions at SRCNet partners-level, the deployment scenario to be used for version 0.1 will be that all the SRCs that want to provide resources for the SKA Regional Centre Network v0.1 will be part of the v0.1 operational version.

This scenario prioritises participation by including all countries that expressed an EoI (Expression of Interest) as operational nodes in the v0.1 version of the SRCNet.

This scenario includes certain benefits:

- **High Participation:** This approach fosters early involvement from all participating countries, fostering a sense of ownership and investment in the project.
- **Maximum Representation:** Including all interested nodes ensures a geographically diverse network with broad representation from the international SKA community.
- **Illumination of diversity:** It is likely that local operations (e.g., technical infrastructure, policies and security) will be different between nodes. Illuminating its impact on operations and collaboratively overcoming it will be crucial to establishing a well-managed, distributed service offering.

and some challenges to be mitigated:

- **Operational Overhead:** Managing and maintaining a large, geographically dispersed network can be complex, especially for a first version with evolving software and processes.
- **Resource Imbalance:** Countries with fewer resources may struggle to meet the operational demands of an immature version, potentially limiting their ability to contribute to development efforts.
- **Software Management:** Maintaining consistent software versions and ensuring compatibility across all SRCs with varying technical capabilities can be challenging.

In the table below, we present the possible contributions for SRCNet v0.1 from different SRCs, with their expressions of interest forms. These contributions will be an upper-level (not continuously maintained) and real requirements will depend on the testing phase (e.g. during performance tests).

	SP SRC	NL SRC	SW SRC	UK SRC	CH SRC	CN SRC	CA SRC	JP SRC	IT SRC	Total
Storage (PB)	0.500	0.100	0.300	4.000	0.400	1.000	1.200	0.651	0.300	8.451
Compute (PFLOPS)	0.010	0.010	0.011	0.175	0.014	0.175	0.040	0.022	0.100	0.557
Percentage Storage (%)	5.916	1.183	3.550	47.332	4.733	11.833	14.200	7.703	3.550	
Percentage Compute (%)	1.797	1.797	1.887	31.447	2.516	31.447	7.188	3.953	17.969	
Harmonisation Rate *	0.020	0.100	0.035	0.044	0.035	0.175	0.033	0.034	0.333	

* This ratio is calculated by dividing the Compute by the Storage. Higher is better

A description of all the SRCs involved in v0.1 and resources can be found in A.5 SKA Regional Centres Contributing to SRCNet v0.1

3.1 Scenario Risk Analysis

The following risks have been identified (see SRCPT-51 for more general risks):

Risk 1: Complexity of Network Management

- **Likelihood:** High. Managing a large, geographically diverse network inherently presents challenges.
- **Impact:** High. Complexity can lead to delays, inefficiencies, and potential security vulnerabilities.
- **Mitigation:** Implement clear communication and collaboration protocols through the setting up of an Operations Group, composed of members of different SRCs to divide the overhead of writing procedures, analysis of errors and reduction of complexity by sharing knowledge.

Risk 2: Unequal Contribution to Development due to Operations Overhead

- **Likelihood:** Moderate. Resource disparity among countries is a known factor.
- **Impact:** Moderate. Unequal contribution could limit the overall effectiveness of the network and hinder scientific progress.
- **Mitigation:** Develop a tiered participation model with different levels of commitment based on available resources. Implement capacity-building programs to support less-resourced countries. Validation of nodes and

network using benchmarking tests will mitigate possible bottlenecks in the network.

Risk 3: Software Incompatibility Issues

- **Likelihood:** Moderate. Rapid development cycles can lead to compatibility challenges.
- **Impact:** Moderate. Incompatibility issues can disrupt operations and hinder scientific collaboration.
- **Mitigation:** Implement robust version control procedures, including deployment techniques from a central software repository using methodologies as automated as possible. Develop clear guidelines and testing methodologies to ensure software compatibility across environments and configurations for software updates.

Scenario 1 offers inclusivity but presents significant operational challenges that will be mitigated by using communication, coordinated procedures and node and SRCNet validation tests. In case of problems found during deployment, coordinated decisions on network deployment and registration could be made by the different SRCNet governance bodies.

4. Policies

Below is a brief overview of the core expectations of policies surrounding security, data, and resources. These Policies will expect to be refined and evolve as the SRCNet matures.

4.1 Data Governance

Data Governance policies can be classified by:

- Data Classification: SKA Data utilised during SRCNet v0.1 is expected to be either public or non-sensitive and non-custodial. The utilisation of data from precursor and pathfinder projects may entail specific usage conditions. If utilised, policy adherence to these constraints is mandatory.
- Data Access Control: Sites must adhere to SRCNet policies concerning the restriction of access to authorised Users and Clients.
- Data Retention: Data used for v0.1 is considered non-unique or non-custodial, and hence somewhat ephemeral. While a fully resilient network is not required, SRC Nodes are expected to provision redundancy measures (e.g., RAID replication, Erasure Coding) to minimise operational disruptions. Retention and replication policies for individual datasets will be determined by the SRC Operations Group based on type, necessity, and test campaigns and set-up through the DDM. Procedures will be established

to inform the Operations Group in case of data loss at a Site (e.g., due to failed disks).

- Data Encryption: Currently, no expectation of encrypting data (either at source, or at the infrastructure of an SRC Node) is expected. Transport of data (particularly over WAN connections) will use encrypted connections (e.g. ssl: https), and especially the transport of authorisation information (e.g. Bearer Tokens) should always be encrypted.
- Data Localization: The SRC Operations Group will manage resource allocation across SRC Nodes. Allocation refinement within an SRC Node will be discussed between the Node representative and the Operations Group.

4.2 Security

An initial set of Security Policies are in the process of being developed. These Policies are intended to be a minimal set that Sites must adopt for SRCNet v0.1 and will develop as the needs and requirements change. The Security Policies broadly follow the guidelines from the Policy Development Kit and will cover:

- Infrastructure Security:

The necessary policies regulating the activities of SRCNet Participants related to the security of the SRCNet Infrastructure in order to minimise the likelihood of impact from security incidents on the SRCNet Infrastructure, its Participants and the wider Research community.

- Acceptable Use and Conditions of Use:

Defines the rules and conditions that govern the access and use of the resources and services provided for, and by, the Participants in SRCNet.

- Service Operations Security:

The set of expectations of the behaviour between those providing services and the operators of those services to establish a sufficient level of trust between all Participants in the Infrastructure to enable reliable and secure Infrastructure operation.

- Incident Response Procedures:

The set of procedures to be followed in the event or suspicion of a security related incident.

The set of security policies for SRCNet and their implementations are currently in preparation and under discussion in SRCNet Objective (SPO-3131), and through Features (to be worked on throughout PI22–24, e.g. (SP-4077), (SP-4135).

Following agreement within the SRCNet community, sites are expected to adopt these policies. These policies will need to be aligned with local policies so possible conflicts will be studied in a case by case approach. For SRCNet v0.1, compliance with these policies will be crucial factors in determining inclusion into the active set of SRCNet v0.1 Nodes.

4.3 Resource Allocation and Management

Resource provisioning by each SRC Node will adhere to their Expression of Interest (EoI) commitments. Not all resources may be continuously available for SRC usage, necessitating scaling as required. The Operations Group will coordinate with each Node to plan and communicate the need for additional resources. SRC Nodes will provision these resources accordingly, with Compute and Storage management defined by the Operations Group to ensure smooth general operations and the support of anticipated challenge campaigns. Metrics reporting allocation and usage will be collected and made available within SRCNet.

4.4 Compliance and Regulatory

In addition to mandatory SKA and SRCNet Policies, Nodes may need to comply with additional laws and regulations at local, national, and interna. Nodes will ensure adherence to these regulations.

4.5 Service Level Agreements

While no enforced SLAs are expected for v0.1, sites are expected to adhere to requirements specified in the Expression of Interest (EoI). Metrics quantifying readiness levels (e.g., availability and reliability metrics) will be developed. Such metrics, coming from local and global benchmarks, network status monitors between SRCs, local/global monitoring services, etc, may be used to assess suitability to be included (or removed) from the active nodes in the SRCNet. Nodes are expected to respond to raised issues within 3 working days and collaborate closely with Operations and relevant development Teams towards timely resolutions.

5. Software Stack

This section details the software stack that underpins the SRCNet platform. The software stack refers to the various software subsystems that work together to enable the functionality of SRCNet. These subsystems can be broadly categorised into local services running on individual nodes, optional services that can be deployed on nodes, and global services that run centrally. The following sections detail the specific software components that make up each category.

5.1 Orchestrator Meta-Service

Every node will have a meta-service orchestrator for managing the SRC services in their local node. For v0.1, it would be required to have a software containers orchestrator. The default implementation provided by the SRCNet ART will be a Kubernetes compliant orchestrator. For this software stack, the SRCNet ART will provide support on the analysis or errors, deployment techniques and node configuration, to guarantee an easy learning curve for the nodes maintenance by the SRCNet global Engineering Operations group. Also, integration with the common GitLab software repository will be properly documented and guaranteed.

Certain SRCNet nodes would like to use other kind of orchestrators due to their present infrastructure, so the requirement for them will be to provide the proper documentation and compatibility on the services execution.

The selection of Kubernetes as default meta-service orchestrator is because Kubernetes offers numerous benefits like Container Orchestration, Scalability, High Availability, Self-Healing, Portability, Resource Efficiency and Extensibility.

Another important aspect is that Kubernetes uses a declarative approach to configuration, allowing you to define the desired state of your applications and infrastructure using YAML or JSON manifests. This simplifies configuration management, version control, and automation, making it easier to deploy and maintain complex applications at scale (i.e GitOps & CI/CD integration). That will simplify the deployment of different versions of the services in the different SRCs of SRCNet v0.1, managed by local operation team members.

A common software repository, including containerised versions of the services to be deployed in different nodes, will be available. This common software repository will be a central GitLab repository, with collaborative software developed by all the SRCNet ART teams.

GitLab serves as a centralised repository for storing code, configurations, and container images. This centralisation streamlines collaboration among developers, ensuring that everyone works from the same codebase and has access to the latest versions of the software components. It also provides an authoritative operational software source for the SRCNet nodes.

GitLab offers:

1. Containerisation support
2. Kubernetes Integration
3. DevOps Collaboration by CI/CD pipelines

4. Scalability and Flexibility after integration with Kubernetes
5. Version Control and Auditing

See Deployment Technique section for a more detailed description on the integration.

5.2 Local Compulsory Services

Every node should have the following services running locally:

Common data-related services (discover data, access data, etc)	Rucio prototype (as RSE) Data Management API (aka permissions interface client) configured with SKAO IAM A&A and GMS
Local data parsing services (to allow remote invocation) and ***visualisation services***	IVOA SODA server and future extensions for other operations Visualisation tools servers including: VisIVO server, CARTA server and YAFits server (under the Data Management API, aka permissions interface client)
Containerised Visualisation tools able to run at SRCNet node	CARTA, Aladin, VisIVO containers from the common software repository (Containers to be invoked under Science Platform permissions layer. Data Access, under the Permissions API interface)
Interactive analysis interface (notebook interface)	JupyterHub or compatible notebooks interface (configured with SKAO IAM A&A or GMS)

Monitoring Services	Services that allow the global monitoring of the SRCNet (e.g. perfSONAR for network performance) This covers the required local deployment that would allow a global System Administrator Portal by gathering metrics from all the SRCs Complete list of services and tests to be used for monitoring to be defined by the Operations Group
Interfaces and integration with global services	See below for IAM, Site Capabilities, etc

5.3 Local Optional Services

Every node could have the following services running locally:

Science Platform Gateway (webserver interface)	SRCNet Gateway (configured with SKAO IAM A&A)
Science Platform	CANFAR (configured with SKAO IAM A&A and GMS)
User Areas based on VOSpace	Cavern (configured with SKAO IAM A&A and GMS)
Cloud-Based Software Deployment Platform	Azimuth

HPC and Batch Compute	SLURM clusters
Software distribution	CVMFS client (with CERN, Compute Canada, and EESSI CVMFS projects configured)
Additional services	Services like Slurm clusters could be deployed and registered into the Site Capabilities Service These services could not be fully integrated into the SRCNet, e.g. not single sign on required for this version

5.4 Global Services

In addition to this, there will be certain global services running centrally. Due to the stateful nature of these services, there will likely be only one instance of them running at a time (additional instances for failover/load balancing might be necessary but outside the scope of this document).

These global services include:

Common software repository	*GitLab* [2]
Identity and Access Management service	Indigo IAM, Keycloak (a solution that supports the OpenID Connect, OIDC standards), GMS
Permissions Service[3]	Service that abstract and combine information from IAM and Rucio Data Identifiers information

[2] https://gitlab.com/ska-telescope/

[3] https://gitlab.com/ska-telescope/src/src-service-apis/ska-src-permissions-api

Science metadata services (publishing and discovery)	An IVOA TAP service with a dedicated science metadata database (PostgreSQL) for data discovery A publishing service for persisting observational metadata (CRUD) and supporting mirroring to local nodes. A IVOA Datalink service to map observations to data products. Optionally deploy a TAP service, a DataLink service and mirrored read-only database to local nodes.
Services metadata-related services (data registration, discovery and access of services of the SRCNet)	Site capabilities API - this allows for the discoverability of information regarding a site, including but not limited to computing, storage and services available
Distributed data management service	Rucio server instance to manage and orchestrate async transfers between the storage endpoints.
Service to manage Rucio data transfers	File Transfer Service (FTS) to conduct and execute transfers
At least one data registration (ingestion) prototype node	Using the ingestion prototype server, which includes data import into the SRCNet data lake and a metadata ingestion service. These services will be deployed in the ingestion nodes, usually close to pipelines although for SRCNet v0.1 the exact location will be determined depending on resources availability. Requirements to be compiled in SP-4227
Operations portal/dashboard	A central Grafana-based set of dashboards to assess health of the

	SRCNodes, the local compulsory services, collect network metrics and generally compile monitoring metrics. The Portal will be used for v0.1 testing and the test campaigns. It will include points of contact for the SRCNodes.

6. Deployment Technique

We distinguish between the deployment of SRCNet services on one hand, and SRCNet software on the other hand. Services include components such as backend or frontend APIs, as well as components that users interact with directly (such as a Jupyter notebook service, or the Science Platform). Software refers to the packaging and distribution setup (e.g., containers) that power these services.

6.1 SRCNet service deployment

Since the SRCNet v0.1 is assumed to run on Kubernetes, we leverage a Kubernetes-based GitOps solution provided by the community. We use a **GitOps** approach for the following reasons and benefits:

- Allows the **declarative definition of infrastructure services** and their configuration, following the **Infrastructure as Code** (**IaC**) paradigm (as opposed to the use of shell scripts that make the muting of infrastructure code error prone and less generalisable).
- Allows a **methodical approach** for the definition, deployment, and upgrading of services.
- Since the **code is version-controlled**, roll-backs become trivial (application state such as database schemas permitting).
- Increases the **reusability and shareability** of infrastructure service setup and configuration across the SRCNet.
- Offers **git** as a well-known interface for the **continuous deployment** of SRCNet infra, delegating the "plumbing" aspect of synchronising the state between the git repository and the infrastructure to the GitOps controller, which is provided and maintained by the open source community.

Optionally, sites can choose to add CI capabilities to the repositories before merging changes.

A procedural description of how to follow the GitOps recommendations of the SRCNet at SRC level will be produced as per feature SP-4249

For the previously mentioned additional services, e.g., HPC systems, a different approach may be followed and best practices for doing so will be shared between nodes offering such services using similar approaches as discussed above (e.g., common software repository to share software)

6.1.1 Infrastructure as Code structure and governance

The simplest and recommended approach would be for each site to have their own GitOps deployment, as well as a git repository which contains the site's IaC definition and configurations of the services to be deployed.

If certain sites want to share the same git repository (for instance because they have a federated Kubernetes deployment, or simply due to an established trust and collaboration agreement), they would be able to implement a repository directory structure, as well as relevant security and infrastructure management policies that suit these needs.

As for the organisation of the GitOps repository(s), there is no specific structure required, and the way it is organised is flexible. Given that each SRC can have specific values for concrete services, for example, one approach would be:

> …
>
> SRCNet/espSRC/science-platform/{production|development}
>
> SRCNet/espSRC/jupyterhub/{production|development}
>
> SRCNet/espSRC/carta/{production|development}
>
> …
>
> SRCNet/chSRC/science-platform/{production|development}

In the case where there's a dedicated production and a development cluster. Within each cluster, git branches can be mapped as environments to specific services or features in development.

6.1.2 Secret management

SRC nodes will be responsible for managing and storing the secrets of the services that are deployed (i.e. in IAM, `client_id`, `client_secret`, etc.). For this purpose, SRCs have to install a secret management tool like *HashiCorp Vault*[4] or use *Kubernetes Secrets*[5], and configure the integration with the GitOps operator. The integration between the secret management tool and the GitOps operator will achieve the following goals:

- Keeping the secrets separate from the Git repositories, enabling shareability of IaC repositories.
- Achieve easy rotation of secrets: when a secret is updated at the source, the change is reflected in the deployment.

[4] https://www.vaultproject.io/

[5] https://kubernetes.io/docs/concepts/configuration/secret/

Both at the SRC node level and at the SRCNet global management level, it will be necessary to monitor the status of deployments for all services. FluxCD and ArgoCD could for example provide these capabilities.

6.2. Software Deployment

This concerns only the software that runs the SRCNet services. We expect that this software will come from multiple sources, as in some cases it will be SRCNet specific software, while in other cases it will be software provided by other communities.

Since the bulk, if not all, of SRCNet v0.1 services will be container-based, we expect these to live in SRCNet container registries (such as Harbor or Gitlab). GitOps configurations can simply reference these to be pulled by Kubernetes.

Of course this opens up the possibility to apply CI/CD pipelines to build the software automatically, especially for software that is owned and developed by SRCNet teams. In this case, we can use existing Gitlab infrastructure to develop pipelines that do basic sanity checks before merging any changes to the repo (CI), and even build and push new container versions to the relevant registries automatically (CD).

6.2.1 Software Versioning

Container services should follow the SemVer 2.0.0 versioning scheme when applying version tags, to allow for consistent versioning across SRCNet.

Following SemVer is simple, while enabling GitOps configurations to bump the container version automatically and yet conservatively using the range notation: Specifying a desired version `~1.2.3` entails that any version in the range `>= 1.2.3 to < 1.3.0` will be pulled. Similarly, the notation `^1.2.3` entails any version in the range `>= 1.2.3 to < 2.0.0`.

7. Services Per Node

Double check implies that this service is already running at this SRCNet node in some way. Single tick implies that the SRCNet node can have this service running for v0.1

Service Category	Service Name	SP SRC	NL SRC	SW SRC	UK SRC	CH SRC	CN SRC	CA SRC	JP SRC	IT SRC	SKAO	SKA-MID	SKA-LOW
Metaservices	Orchestrator Meta-Service	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓		
Metaservices	Common Software Repository										✓✓		

Service Category	Service Name	SP SRC	NL SRC	SW SRC	UK SRC	CH SRC	CN SRC	CA SRC	JP SRC	IT SRC	SKAO	SKA-MID	SKA-LOW
Compulsory Services	Local Data-related services: Rucio Storage Element (RSE)	✓✓	✓✓	✓	✓	✓✓	✓✓	✓	✓✓	✓	✓✓		
Compulsory Services	Local Data Management API[6] (aka permissions interface client)	✓✓	✓	✓	✓	✓✓	✓	✓	✓	✓	✓✓		
Compulsory Services	Parsing Local to data for visualisation of remote data (extensions of IVOA SODA services) exposed by Data Management API	✓✓	✓	✓	✓	✓✓	✓✓	✓	✓	✓	✓✓		
Compulsory Services	Containerised visualisation tools to visualise local data (same as Science Platform instances)	✓✓	✓	✓	✓	✓	✓✓	✓✓		✓			
Compulsory Services	Monitoring services	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compulsory Services	JupyterHub (or compatible)	✓✓	✓✓	✓✓	✓✓	✓	✓✓	✓✓	✓✓	✓✓	✓✓		

[6] https://gitlab.com/ska-telescope/src/src-service-apis/ska-src-data-management-api

Service Category	Service Name	SP SRC	NL SRC	SW SRC	UK SRC	CH SRC	CN SRC	CA SRC	JP SRC	IT SRC	SKAO	SKA-MID	SKA-LOW
Optional Services	Science Platform Presentation Layer (SRCNet Gateway)		✓✓				✓						
Optional Services	Science Platform (CANFAR)	✓✓			✓	✓	✓✓	✓✓		✓			
Optional Services	Additional services (Slurm clusters)	✓✓	✓✓		✓✓		✓✓			✓			
Optional Services	Additional services (Azimuth)				✓✓								
Optional Services	User Areas based on VOSpace (Cavern Service)	✓✓		✓✓	✓		✓	✓✓					

Service Category	**Service Name**	**SP SRC**	**NL SRC**	**SW SRC**	**UK SRC**	**CH SRC**	**CN SRC**	**CA SRC**	**JP SRC**	**IT SRC**	**SKAO**	**SKA-MID**	**SKA-LOW**
Global Services	Identity & Access Management (IAM)				✓✓		✓✓						
Global Services	Permissions Service										✓✓		
Global Services	IVOA Group Management Service (GMS)				✓✓								
Global Services	Science Metadata Service										✓✓		
Global Services	Data Locations Service										✓✓		
Global Services (Optional)	Science Metadata Mirroring Service	✓		✓		✓		✓					
Global Services	Site Capabilities Service										✓✓		
Global Services	Rucio			✓							✓✓		
Global Services	FTS			✓	✓✓								
Global Services	perfSONAR Mesh scheduler				✓✓								
Global Services	Data registration (ingestion) prototype					✓					✓✓	✓	✓
Global Services	Operations Portal and Dashboards										✓		

8. Storage Configuration

8.1 Bulk storage

The heterogeneous set of storage provisioned across the Nodes in SRCNet v0.1 must be configured for interoperability, adhering to the Policies described in Policies, above. Aside from the architecture and implementation at the storage layer itself, access must be made available to Compute resources within the Node, and for data ingress and egress between the other SRC Nodes and Telescope sites. It is required that all Nodes comply with the APIs, the access protocols and associated AuthN/Z, as defined by SRCNet.

An outcome of the downselect process of the DDM assessment is that Rucio (including the necessary ancillary services and dependencies, e.g. FTS) will be used for SRCNet v0.1. This requires Sites to provision Storage Endpoints able to interoperate with the SRCNet Rucio instance and transfer data using compatible protocols

The Compute section describes the requirements of storage local to the Compute resources 9.1 Level 1 - Compute platforms.

8.1.1 Bulk storage requirements

For SRCnet v0.1 it is assumed that sites will provision bulk storage as a filesystem (POSIX-based). Alternatives, as provisioning bulk storage as an Object Store should be discussed within SRCNet via Operations Group. For the scratch storage close to the compute, this is expected to be POSIX-like filesystem based.

Nearline storage (e.g. tape or other archival technology) is not required during v0.1. Online storage may be provisioned via SSD and HDD (having the ability to test against these different levels would be advantageous). The transfer of bulk data between sites will likely be facilitated through a Data Transfer Node (DTN). The set of requirements for bulk storage (and DTN) is given as follows:

MUST

- Allow internet connectivity to other SRC Sites (likely via a DTN).
- Allow data ingress or egress via either push or pull mechanisms (i.e. as the passive or active end of the connection) with other Sites, initiated by any appropriately authorised client.
- Provision the necessary software and infrastructure to support the creation of a Rucio Storage Element (RSE) (defined below).
- Secure client access with encrypted endpoints (e.g. HTTPS via TLS), with protocols, ciphers, and hosts certificates of sufficient assurance, as defined by the Operations Group.

- Follow SRCNet agreed policy procedures to provide assurance of data access control of all data; this includes (but not limited to): adopting agreed systems for Authentication and Authorisation for data access; following site security and incident response procedures, etc.
- Expose detailed metrics concerning external connectivity with the central SRCNet Operations Group (subject to security and privacy protection).
- Allow tests to be run to characterise Site performance and determine indicator metrics such as storage Availability and Reliability.
- Allow access to the local compute resources (via correctly authorised systems)

SHOULD

- Ensure a basic level of resilience to day-to-day operational routine hardware failures (e.g. failed hard-drives).
- Provide details of the available storage volume available for SRCNet after overheads for resilience and operational needs are accounted for (which is overwise assumed to be the EoI commited storage amount).
- Document their storage configurations to facilitate knowledge transfer so to collectively gain from the experiences of SRCNet v0.1.

8.2 Storage Endpoint (SE) configuration

A number of existing Storage Endpoint technologies that are interoperable with the Rucio ecosystem are available, namely: StoRM, XRootD, EOS, dCache.

Assuming the WebDAV will be the primary bulk data transport protocol, the SE will need to support the methods: GET, PUT, POST, DELETE, HEAD (including to request checksums digest) , PROPFIND, MKCOL, COPY (for third-party-copy requests) and MOVE. For certain storage implementations only a subset of these might be implemented and Rucio should be configured appropriately (after ensuring such reduced capability is supported). For storage supporting S3 an appropriate endpoint should be configured.

Other transfer protocols may be used for testing and development purposes. The SE will be configured to meet the authorisation requirements of SRCNet.

Rucio Storage Endpoints (RSE):

In Rucio, the RSE is a logical representation of a storage location. A site may have several RSEs, depending on need. Each RSE might share the same SE, but describe different (and distinct) locations on disk (or other medium).

For each RSE that is configured, the site will provide the set of protocols, hostnames, port, and basepath to the rucio-managed storage area.

8.3 Networking (WAN)

For efficient, performant and secure data ingress and egress at each Node, each Node (and site) should provide Storage Endpoints (SE) to connect SRCNet storage for external transfers, which will provide the services and network capabilities required for each Site. Here, the SE is defined as the necessary software, host, port and protocols needed to support data transfer. A Data Transfer Node (DTN) is defined as the physical (or perhaps virtualised) hardware and host configuration configured to support high-throughput data transfers.

Sites should also consider the scenario in the roadmap where Virtual Routing and Forwarding (VRF) is deployed across SRCNet. If a VRF is provisioned, sites will make a best effort to join the VRF at the earliest opportunity available to them.

The set of requirements for v0.1, (and with preparations towards v0.2 and beyond) is as follows:

MUST

- Allow internet connectivity (TCP, possibly UDP) between all SRCNet Sites (including the TWO SKAO telescope sites if required in v0.1).
- Provide at least 10Gbps (both ingress and egress) internet connectivity at each Site. This might be shared between other Site users.
- Provide the capability of at least 1 Gbps sustained (continuously for periods of Test Campaigns) ingestion from each of Mid and Low telescope sites (possibly emulated by access to other hosts in AU and SA SRC, or by using an alternate SRC Node).
- Provide the capability of at least 1 Gbps sustained (continuously) both to and from any other Node in the SRCNet. This therefore requires that each node is able to provide such capability, and there are no bandwidth limitations within the global network to restrict such a requirement (which is an external factor for v0.1). This also assumes that while each link (site to site transfer) may be able to achieve 1 Gbps, having concurrent site to site transfers, the per-link throughput may be less than 1 Gbps, while achieving at least 1 Gbps combined.
- Allow internet connectivity for the required SRCNet services (e.g. AAI servers, Data cataloguing and transport tooling, metadata querying); this includes those services that run at the Site, and other SKA services run at other locations.
    - In particular, access from the SRCNet Rucio instance, FTS instances used for SRCNet, and other SRCNet Site SEs must be allowed, as well as tooling for any related services.
- Provision network performance tooling (perfSONAR), with metrics collected and propagated back to SRCNet monitoring to enable the characterisation

of (external) Site network performance and determine indicator metrics such as Availability and Reliability.

SHOULD

- Provide the capability for at least 1 Gbps sustained (continuously) throughput of data transfers to and from other Nodes in the SRCNet v0.1.
- Provide capability for IPv6 and ancillary necessary components (e.g. ICMPv6, allowing for MTU path discovery).
- Enable Jumbo Frame support (MTU 9000), supported along the full network path.
- To apply network-based tunings following the recommendations of the NREN forum WG (based on the tunings available from https://fasterdata.es.net/). NREN forum WG recommendations will be circulated within the NREN forum shortly.

9. Computing Configuration

It is important to emphasise that hardware heterogeneity and different software stack flavours to run the services is foreseen on the SRC data centres as long as hardware QoS and software compatibility are fulfilled (although these heterogenous flavours will be gradually supported by the SRCNet Operations Group). Any kind of computing solutions can be expected, from bare metal, to more sophisticated cloud services, from standard X86 CPUs to ARM+GPU multi-node architectures.

Interoperability layers (IVOA-like) that can access various computer services are therefore required. This is the goal of the Computing Service API, for example, which although prototype, partial implementations might be tested already on v0.1, is not anticipated to be released before v0.2.

In order to verify the functionality and the capabilities of progressively available computing platforms, test cases are provided, addressing use cases that allows to assess:

1. the basic functionalities and capabilities of available computing platforms (Compute Tests, see 10.2.2);
2. the usability of the computing platforms for the specific science supported by the SRC nodes and the interoperability among different nodes (Compute Tests, see 10.2.2, Integration Tests, see 10.4.2)

The use cases fall into two categories.

- Level 1 use cases address advanced users who are familiar with the underlying technologies and imply direct access to the platforms.
- Level 2 services are for general community use (e.g. end-users/scientists using analysis interfaces). A limited set of level 2 services are compulsory for SRCNet v0.1 like Jupyter notebooks and visualisation. Level 2 services

using federated execution (moving software to the proper execution node either due data proximity or computing resources availability) will be formally integrated for v0.2 [AD2]

9.1 Level 1 - Compute platforms

As stated above, no specific restriction can be imposed on the computing platform implementation. We expect for v0.1 that Level1 use cases will be available for technical users who are already familiar with the deployed platform technologies, as for instance:

- Slurm, for managing batch and interactive jobs on HPC systems (https://slurm.schedmd.com/),
- Kubernetes, deploying and managing containerized solutions (https://kubernetes.io/),
- OpenStack, for Virtual Machines based services (etc. who will prefer to have direct access to the platform with minimal obstacles.

Several other options are available.

The Level 1 tests will be run directly on the nodes by the local site administrators or operators. To run the tests they will need to be able to access sufficient resources to run the target test cases close to or co-located with the test data imported at each site.

9.2 Level 2 - Compute services

Level 2 use cases will be for non-technical users who don't want to deal with the details of how the tasks are achieved. They can be run through high-level tools, like Jupyter notebooks, hiding the complexity of the underlying access and management layers (although complete hiding may not always be achievable for certain tasks).

There are no plans to provide support for non-technical end-user access in SRCNet v0.1, although integration of ongoing work could be allocated as optional. Level 2 compute services will be formally integrated for v0.2. See [AD2]

9.3 Impact on Local Network Policies

This section clarifies the impact of local network access policies, defined by individual site administrators, on different deployment scenarios:

- **Internal Tests:** Local access policies won't affect tests run internally by site administrators because they have full control over the network environment.
- **Public Service Deployments (Static):** Static deployments of public services, like JupyterHub, are also managed by site administrators. So, they can configure the network access policies (firewalls, public IP assignment) to ensure proper access.

- **Dynamic Services:** Dynamic services, as visualization parsing services, invoked by user actions, might be impacted by local access policies. These user-launched services might require specific ports or internet access which could be restricted by local network rules. Connectivity rules of these services should be coordinated with local site administrators.

10. SRCNet v0.1 Tests

This section details the various kinds of tests required for a participating SRCNet site. These tests are crucial for SRCNet node sites to give visibility of the resources they have pledged, and build a joint understanding of the SRCNet. They will help bootstrap the work of the SRCNet Operations Group and will likely be conducted and managed by the same team.

We want to run tests to know what resources are available at a site (see Sect. 10.1), what services are running at a site (see Sect. 10.2), and whether these services can work in harmony within the SRCNet (see Sect. 10.3). For each of these categories and the corresponding tests, we should be able to demonstrate the results openly, in a way that is largely uniform across sites and makes the results easy to comprehend e.g., in the form of Grafana dashboards.

These tests will be packaged appropriately (e.g., containerised) and easily reproducible on a candidate SRCNet v0.1 site without violating any of the local (or global) policies. It is understood that by running a common suite of tests across heterogeneous hardware infrastructure, this may not provide the most accurate picture of the resources. However, the goal of these tests is to establish a baseline (not peak) level of performance in preparation for v0.2, establish the Operations Group procedures, and build trust in the SRCNet community.

See A.4 for a detailed description of the workload classification. For v0.1, we have benchmarks, systems workloads, and of course science workflows.

Category	Outline	Current status	Related features	Results
Infrastructure tests	What storage/compute/network resources are available at a site?	Storage: Tests under development	SP-4216 SP-4293	Metrics and storage operations to be defined
		Compute: Tests ready	SP-4288	STARS metric
		Network: perfSONAR tests	SP-4082 SP-4356	Grafana-based dashboard showing bandwidth and

		ready, operational schedule to be defined	SP-4301	latency metrics.
Service monitoring	Are the local compulsory services and global services running and healthy?	To be defined	SP-4326 SP-4217	Grafana-based dashboards to give visibility of what services are running at a site in real-time, and historical information of the same.
Test campaigns	Are all these services working in harmony within the SRCNet?	Test Data	SP-4132 SP-4229 SP-4299	Test data identification and ingestion complete to ingestion nodes.
		Integration tests	SP-4256 SP-4274 SP-4229	Grafana-based dashboard showing data ingestion, replication, and data transfer performance in conjunction with perfSONAR dashboard. Tests added to the relevant gitlab repositories (src-workloads or rucio-task-manager).
		Test Campaigns	SP-4228	

10.1 Infrastructure level tests

This section outlines the various tests designed to assess the performance and capabilities of the SRCNet infrastructure. The tests cover three primary areas: storage, compute, and network.

By implementing these tests, we gain valuable insights into the infrastructure's health, performance, and potential bottlenecks. This information is essential for maintaining a robust and efficient system that supports the scientific challenges of the SRCNet community.

10.1.1 Storage tests

There are plans for CASA measurement set tests, on a single node, for serial read, serial write, parallel read and parallel write operations[7]. These CASA tests will

[7] https://jira.skatelescope.org/browse/SP-4293 https://jira.skatelescope.org/browse/SP-4216

make use of a representative set of visibilities and image cubes, likely using a combination of publicly available LOFAR data and simulated SKA data sets.

All relevant storage tiers should be tested, both the online bulk storage (i.e. data lake) and any available faster storage tiers.

The initial version of those tests can be found here:

https://confluence.skatelescope.org/display/SRCSC/Benchmarking

SP-4216 These tests need to be made reusable, and easily reproducible by all. The resources for running the tests (scripts, container images, documentation) should be available on Gitlab.

10.1.2 Compute tests

A classification of the compute workflow tests to be provided can be found in Workload Classification.

The compute tests aimed at Level 1 services to evaluate the capabilities and performance of a single SRC node on scientific meaningful use cases and on distributed resources. These tests are run by technical experts and defined as benchmark workloads in A.3. Scientifically meaningful use cases and complex workflow over local and distributed computing resources using Level 2 services are covered in 10.3.2 and 10.3.3 (System tests and scientific workflows as defined in A.4).

Compute tests will be run routinely by local Operations Group members to keep the local and global infrastructure under control and keep track of possible updates and/or issues.

These tests exploit the SRC Workloads test suite under construction (https://gitlab.com/ska-telescope/src/src-workloads) that can run specific tasks and capture runtime information to build a STARS metric. They are described here. They aim at verifying:

1. The possibility of building and/or running a test application on a given SRC node
2. The computing capabilities of such node
3. The current performance of the node

The code for the testing will be provided as:

1. Source code and libraries to be built on the node with available compilers, distributed through git repository

2. Docker container to be downloaded from a centralised repository or provided through CVMFS
3. Possibly Easybuild and/or Spack recipes for automatic configuration and building

The benchmark workloads will be run to establish baseline performance on a single node across the proposed SRCNet v0.1 nodes. The set of representative workloads will be expanded over time. These workflows, and others to be compiled, will be close to real science use cases to be executed by users to benchmark the nodes with realistic scenarios.

Enabling features in the SRCNet ART in PI22: SP-4166

10.1.3 Network tests

Having a monitored understanding of the network will facilitate debugging of potential issues and help characterise the various network flows to aid the development of network architecture and identify bottlenecks, etc. Work is currently ongoing (SP-4082) to provision perfSONAR across SRC Sites and to establish a mesh of network based tests (SP-4301). The performed tests will comprise of routing based tests, latency and tests of packet loss, as well as the capability to run throughput level tests. Most tests will run at regular intervals (automated), and return data for a common dashboard (e.g MadDash or Grafana). More intensive tests (e.g. throughput) may need to be scheduled and announced via the Operations Group. Sites will be expected to apply the configuration necessary to join the SRCNet mesh, however the tests themselves are expected to be automated and centrally managed.

10.1.4 Results

As sites may have certain unique considerations or modifications to executing these tests, they should document their test setup and execution details in addition to the results obtained.

Storage tests: Sites to share metrics for performing storage operations on an identified set of datasets.

Compute tests: STARS metric for a subset of workloads in the src-workloads repository.

Network tests: Automated, central tests perfSONAR gathering latency and bandwidth metrics into a Grafana-based dashboard.

10.2 Service monitoring

Services monitoring involves the systematic collection and analysis of data to assess the performance, availability and health of services at both the local SRC and global levels. This includes compulsory local services and global services as defined in the Software Stack in this document.

10.2.1 Description

To enable a centralised dashboard view on the local and global services running at the SRCNodes, we need a data source containing information from these services.

There can be several ways of doing this

- In its simplest form, this could be events from services being pushed into a central DB serving as a Prometheus data source that is used to build dashboard views.
- It could also mean sites running an metrics/events exporter service that is probed by/scraped by the central monitoring service. In either approach (push vs pull, activeMQ, Kafka etc), an initial format for the corresponding events needs to be established ().
- The most loosely coupled way of doing this would be for sites to run their own monitoring stack, generating dashboard views off of a common Grafana dashboard template.

Relevant features are SP-4326 and SP-4217.

10.2.2 Results

Dashboards (e.g., Grafana-based) should be available to give visibility of what services are running at a site in real-time, and historical information of the same including when they started running, downtime etc.

10.3 SRCNet v0.1 Test campaigns

This section describes the test campaigns designed to evaluate the functionality and integration of the SRCNet v0.1. These campaigns aim to verify that SRCNode services operate harmoniously within the network. The focus lies on assessing the system's robustness, scalability, and performance under diverse operational scenarios.

This section covers:

- Data to be ingested into SRCNet v0.1
- Integration tests that involve multiple SRCNet components and sites
    - Ingestion and Dissemination Tests, using the ingestion interface and executing data dissemination and data registration
- Campaigns or execution of coordinated tests within a specific timeframe, involving all v0.1 sites to exercise the pledged resources.

10.3.1 SRCNet v0.1 Test Data

Selected data to be included in v0.1 will have associated possible use cases (scientific and technical) that could be tested during the test campaigns (see SP-4132). These data will include:

- Precursor/Pathfinders data
  - Ingesting precursor data enables the validation and refinement of data processing pipelines, ensuring their readiness for handling real SKA data and the execution of complete science use cases. Public data available at the different participating v0.1 members should be identified.
  - Pathfinder data, specifically LOFAR data, would be valuable. [AD4] mentions several datasets, notably including LoTSS DR2 which is primarily stored at SURF in the Netherlands.
- SKA test data:
  - Synthetic datasets generated by SDP (Science Data Processor) pipelines, mirroring the scale and format expected from real SKA observations. By simulating SKA data, the network can assess the efficacy of data management protocols and software solutions under conditions close to actual operations.
  - Simulated test data can be generated using SKA telescope config using the Karabo pipeline. It makes use of OSKAR and RASCIL packages underneath. Example notebook here shows generating visibilities from sources in the GLEAM survey with ASKAP config, and making dirty images of the same. More documentation here: https://i4ds.github.io/Karabo-Pipeline/examples/examples.html. See feature SP-4229.
- Science Data challenges data: Data already analysed in a locally controlled scenario using scientific workflows will provide information on the overhead provided by the SRCNet distributed data and resources

For each data type, the following aspects need to be documented:

- Availability: Source or generation mechanism of the data.
- Contact Point: Responsible individual or entity for data inquiries and coordination.
- Size/Type: Magnitude and format specifications of the dataset.
- Science Use Cases/Software: Specific scientific objectives and requisite software tools for analysis.

These test datasets will be instrumental in evaluating the robustness, scalability, and performance of the SRCNet infrastructure across various operational scenarios, including data management, scalability, stress tests, and more.

10.3.2 Integration tests

These tests will involve several components of the SRCNet and more than one SRCSite thus testing integration across these components and sites. They will be run multiple times, ideally regularly.

User workflow tests
Description For SRCNet v0.1 there will be an initial set of workflows that can execute at all candidate SRCNet nodes using public precursor data or synthetic test data. These workflows will be packaged (e.g., containerised for easy portability) and properly documented, so executing them at the SRCs would be as straightforward as possible. These workflows are publicly available within this Gitlab repository and categorised per SRCNet version. For v0.1, this repository contains test workflows intended as basic science use cases as defined by the tasks here. These workflows, and others to be compiled, will be close to real science use cases to be executed by SRCNet Operations users (L1 tests and limited available L2 tests) to assess the integration of the system with realistic scenarios. [8] Sites can propose their own end-to-end tests provided the corresponding tasks/workflows are added to the common repository. In the case of interactive workflows like using VisIVO or CARTA to visualise a range of image cube sizes, involving human verification for acceptable performance, these should be described and documented in detail. The set of workflows to be used as node validation tests should be available at least two months before the formal release into the common software repository.
Results Sites running these integration tests to document the tasks/workflows execution from the repo via Jupyter Notebooks. Additional details of the test execution include but not limited to local environments, resources available, sites involved, and data preparation to be documented on Confluence. Tests to be shared/demoed within the ART. Any new tasks/workflows to be added to the common repository with the corresponding documentation for the same on Gitlab or Confluence as appropriate.

Ingestion and Dissemination tests
Description Test data (precursor data/synthetic test data) is ingested using the ingestion service, and replicated to a set of sites.
Results Dashboard views showing data ingestion, replication and the corresponding data transfer performance.

10.3.3 Test Campaigns

These refer to orchestrated tests within a particular identified and agreed time window that involve all the participating SRCNet v0.1 sites to exercise the resources pledged.

Data lifecycle
Description These should come from/be documented in the workloads repo just like the integration tests. The main difference between these tests and the integration tests will be scale and level of participation (all v0.1 sites to participate). These will be end-to-end data lifecycle tests starting from data ingestion, data replication to multiple sites, data processing via Jupyter notebooks (and data visualisation) and uploading of new data products. See feature SP-4228

[8] There is a clear miss around further pipelines that manipulate images (e.g. source finding) and pipelines that require GPUs. We expect to add these for version 0.2.

Results   Documented set of 'live' tests conducted during the identified high-availability window describing the data lifecycle being demonstrated, sites participating, data used (type/volume), workflow chosen, and description of any new data ingested. The Operations Group will likely be consolidating this information in preparation for the test campaigns. Where applicable demos of the tests conducted, and dashboard views showing data ingestion, and replication.

Data transfer
Description   Exercise the network and storage with full network mesh tests conducted with data of various file sizes using the rucio task manager.
Results   Dashboard views showing data replication and the corresponding data transfer performance.

A References

A.1 Applicable documents

The following documents apply to the extent stated herein. In the event of a conflict between the contents of the applicable documents and this document, **the applicable documents** shall take precedence.

[AD1] Salgado, J.; Wicenec, A.; Goliath, S.; Joshi, R.; Swinbank, J.; Bolton, R.; Webster, B.; Oonk, J.; Grainge, K.; Sánchez, S.; Parra, M.; Dack, T.; Hardcastle, M.; Barbosa, D.; Llopis, P.; Fabbro, S.; Beswick, R.; Villote, J.P.; Breen, S.; Yates, J.; Grange, Y.; Gaudet, S.; An, T.; Possenti, A.; Darriba, L.; Holanda, V.; Mendoza, M.; Galluzzi, V.; Svedberg, T.; Lee-Waddell, K.; Vitlacil, D.; Pandey, V.N.; Akahori, T.; Chisholm, L.; Horton, M.; Watson, R.; (2023)

SRC-0000001 SKA Regional Centre Network (SRCNet) Software Architecture

https://confluence.skatelescope.org/download/attachments/74711238/SRCNet%20Software%20Architecture.pdf?version=1&modificationDate=1693571258908&api=v2

[AD2] Salgado, J.; Bolton, R.; Swinbank, J.; Joshi, R.; Sánchez, S.; Villote, J.P.; Gaudet, S.; Yates, J.; Barbosa, D.; Taffoni, G.; Frank, B.; van Haarlem, M.; Breen S; Conway, J.; Akahori, T.; Yates, J.; Tolley, E.E.; Wadadekar, Y.; Lee-Waddell, K; de Boer, J. (2023)

SRC-0000002 SRCNet Top-Level Roadmap

https://confluence.skatelescope.org/download/attachments/74711238/SRCNet%20Top%20Level%20Roadmap.pdf?version=2&modificationDate=1693571286081&api=v2

[AD3] Garbutt, John; Walder, James; Salgado, Jesús; Joshi, Rohini (2024)

SRCNet v0.1 Node Requirements

https://docs.google.com/document/d/1PZ4Il_RgIs2rtR0XawoAa0Q3FXycI4-4yhmjbbrDzLw/

[AD4] Clarke A.; Franzen T.; Breen S.; Bolton R. (2023)

SRCNet Use Cases

https://docs.google.com/document/d/12Tlg438xfZahNusCfAdqzm6u6XqUZQ8HdhtgI5zr5rQ/edit

A.2 Reference documents

The following documents are referenced in this document. In the event of conflict between the contents of the referenced documents and this document, **this document** shall take precedence.

AARC Consortium Partners;AppInt members;Nicolas Liampotis. (2019, 4 30). *Deliverable DJRA1.4: Evolution of the AARC Blueprint Architecture*. AARC2-DJRA1.4_v2-FINAL. https://aarc-project.eu/wp-content/uploads/2019/05/AARC2-DJRA1.4_v2-FINAL.pdf

A.3 Jira Features

Link	Description
SP-4227	Define requirements for SRCNet Ingestion Nodes
SP-4132	Identify test data collections for v0.1
SRCPT-51	Risk Register / Tracking
SP-2641	Groundwork for SRC Policy Framework
SP-4135	Establishment of Baseline SRCNet Policies
SP-4216	Existing storage tests to be made more usable by adding them to GitLab
SP-4166	Use STARS to obtain benchmark scores in the SRC workloads Gitlab repo.
SP-4082	perfSONAR network monitoring between multiple SRC sites
SP-4301	Definition of perfSONAR configuration for automated sets of tests
SP-4326	Service monitoring v0.1 infrastructure

SP-4217	First version of the Operations Administration Portal
SP-4249	Document procedures on the deployment of compulsory services for SRCNet v0.1 (and error handling), following GitOps, to be used by the Operations Group and ART members.
SP-4229	Generate simulated test data with Karabo
SP-4256	Ingestion and Dissemination tests Test data (precursor data/synthetic test data) is ingested using the ingestion service, and replicated to sites using a set of sites.
SP-4228	Data lifecycle tests
SP-4257	Data transfer tests These will be one category of tests driving the test Campaigns for v0.1. The data transfer tests will exercise the network and storage with full network mesh tests conducted with data of various file sizes using a tool such as the task manager (or something similar).

A.4 Workload Classification

#	Name	Data dependency	Purpose	Type of test output	Status for v0.1
1.a.	Micro-benchmark workload	None. Cloning the git repository is enough. Data inputs are small or non-existent.	Measure the performance of computing infrastructure (may include compute, storage, or networking). Produces a performance metric (runtime, latency,	Benchmark (score).	Out of scope. Currently, no SRCNet-specific micro-benchmarks (1.a.) exist, since they

			throughput, STARS,..). Micro-benchmarks consist of a single task, are minimalistic by nature (equivalent to just the most costly loop/kernel of an application), Designed to stress a specific part of a system. Can be easily shared outside of SRCNet (e.g. hardware vendors) due to the minimalistic nature and reduced data dependency, and therefore often do not take very long to run (both in effort and runtime). The advantage of having representative SRCNet workloads distilled into micro-benchmarks (e.g. by extracting the main kernels from each application) would be having a smaller application that requires less input data and runs in less time, but still stresses the system in the exact same way that the original application did.		are costly to develop (CASA and IDG projects do have some micro-benchmarks that could be used for this purpose).
1.b.	Benchmark workload	Data are relatively small, but need to be downloaded separately after cloning	Measure the performance of computing infrastructure (may include compute, storage, or networking). Produces	Benchmark (score)	See 10.1.2

		the benchmark repository. Data can be downloaded anonymously (without the need for any special credentials or account).	a performance metric (runtime, latency, throughput, STARS, ..). Benchmarks consist of a single task (**not a workflow**, but closer to a single full program rather than just the kernel/loop). Benchmarks can be shared outside of SRCNet, but requires additional steps to run due to requiring input data downloads. Benchmarks often take more time and effort to run than micro-benchmarks.		
2.	System test	Data are obtained from the Data Lake automatically. May require specific credentials or accounts to run.	This workload is designed to run by combining different systems together (for instance: download inputs from data lake, submit job to batch system). They are meant for system integration and system validation testing.	Validation in the form of demos, and dashboard views demonstrating the test runs. (While it could also measure runtime score, benchmarks or micro-benchmarks are typically more useful for measuring performance)	See 10.3
3.	Scientific workflow	Data are obtained	A whole workflow (often a combination	Validation in the form of	See 10.3.2

		from the Data Lake automatically. May require specific credentials or accounts to run.	of workload tasks) that represents a full scientific use case.	demos, and dashboard views demonstrating the test runs. (While it could also measure runtime score, benchmarks or micro-benchmarks are typically more useful for measuring performance)	
4.	Interactive scientific workflow	Data are staged from the data lake, may require specific account credentials and staging requirements may depend on the application using the data.	This is a end to end system test that tests the integration of multiple components of the SRCNet, specifically for interactive workflows such as visualisation.	User feedback	See 10.3.2
5.	Continuous monitoring	None. If any data required for this type of test, it is relatively small, synthetic and generated on demand.	Monitoring of service status across sites in a central way to capture the health of the SRCNet system as a whole.	Dashboard views indicating service health across different sites.	See 10.2

A.5 SKA Regional Centres Contributing to SRCNet v0.1

This section provides an overview of the SKA Regional Centres (SRCs) involved in SRCNet v0.1. Each SRC subsection will include a brief description of the centre, a table outlining the resources they contribute to the network, and a table highlighting relevant milestones for their involvement in SRCNet v0.1. The subsections are presented alphabetically by SRC name.

A.5.1 Canada SRC

A.5.1.1 Overview

The Canadian SRC has partnered with the Digital Research Alliance of Canada (the 'Alliance') to provide the infrastructure for SRCNet v0.1 and beyond. The resources being purchased will go to expanding the current compute and storage clusters that the CANFAR Science Platform and storage systems run on. SRCNet v0.1 resources will be available as part of these already existing systems, with quotas and allocations to manage the resource usage.

The first expansion to these resources was ordered in March 2024, with expected availability in September 2024; the second expansion is expected to be ordered in June 2024 with availability in December 2024. Storage resources will be added to the existing Alliance Ceph cluster (providing CephFS and Ceph Object Storage) and allocated as outlined in the CanSRC EoI and the table below. Compute resources, composed of 40 nodes each with two AMD EPYC 7513 32-core CPUs and 512GB of RAM (theoretical peak ~9.8TFlops/node), will be added to the existing kubernetes cluster on which the existing CANFAR Science Platform runs. Access to the compute resources for SRCNet v0.1 use will be made through this platform.

Networking TBD

A.5.1.2 Resources available for SRCNet v0.1

Resource	Value	Expected Availability
Storage	1.2 PB: 0.6 PB for RSE; 0.6 PB for cavern If there is a demonstrated need, we ***may*** provide additional capacity on those systems, up to 4.6 PB.	December 2024
Procurement	March 2024 June 2024	September 2024 December 2024

Compute	0.04 PFLOPS/273 physical cores through the CANFAR Science Platform If there is a demonstrated need, we may provide up to 0.11 PFLOPS/702 physical cores.	September 2024
Procurement	March 2024	September 2024
Network	100 Gb	Available now
Procurement	Nothing planned	

A.5.1.3 Relevant local milestones

Milestone	Date
Procurement Deadline	See above.
Hardware Upgrade	
Node Ready for Validation	
Node Integrated in SRCNet v0.1	
Limited availability period	

A.5.2 China SRC

A.5.2.1 Overview

ChinaSRC is a cutting-edge computing platform that employs a novel hybrid architecture. This setup combines general-purpose Intel x86 CPUs, GPUs, and also ARM processors, providing unparalleled flexibility and efficiency for data processing tasks.

Specifically, the prototype features 35 Intel x86 CPU nodes with 2240 cores (maximum 128 cores per node), 12 ARM CPU nodes with 1152 cores (96 cores per node), and 4 GPU nodes housing 16 Nvidia V100 and 8 A100 GPUs, which offer approximately 800 TFlops of computing power. Collectively, the CPU nodes contribute about 196 TFlops in total[9].

Regarding storage, the prototype boasts approximately 9PB of distributed storage capacity, facilitated by a high-speed internal connection network of 100-200Gb/s for data exchange between compute and storage nodes. There are several node which can support up to 4TB of memory maximum, with up to 36GB per core, ensuring efficient processing of large-scale data sets.

Additionally, the prototype is connected to a transcontinental internet link of 10 Gbps, and maintains a 200 Mbps connection with other SKA regional centre nodes, ensuring smooth collaboration in global research efforts.

A.5.2.2 Resources available for SRCNet v0.1

Resource	**Value**	**Expected Availability**
Storage	*2PB*	*Finalise resource allocation details*
	1PB	*Available now*
	Negotiate and finalise contracts with vendors	*None*
Compute	*Initial allocation defined*	*Available now*
	Additional resources planned for future upgrades	*None*
Network	*Baseline connectivity established (10Gbps)*	*Available now*
	High-bandwidth upgrades planned for peak data periods (100Gbps)	*01/2026*

9 https://shaoska-user-guide.readthedocs.io/

A.5.2.3 Relevant local milestones

Milestone	Date
Procurement Deadline	*See above*
Hardware Upgrade	
Node Ready for Validation	
Node Integrated in SRCNet v0.1	
Limited availability period	

A.5.3 Italy SRC

A.5.3.1 Overview

INAF will implement and manage the Italian SKA Regional Centre node. The facility will run in the new Technopole site in Bologna and it will be supported in terms of system management and operations by CINECA, the National Italian Supercomputing Centre. It will be integrated in the CINECA ecosystems in the perspective of scaling in capacity and computing power. In view of that, the Italian SRC will have to discuss with CINECA's about agreed policies.

The facility will be acquired in 2024 and deployed during the first quarter of 2025. The procurement is already on-going.

In parallel with the hardware, INAF is also planning to recruit a significant number of FTE, dedicated to the setup and management of the facility and the related software stack.

A.5.3.2 Resources available for SRCNet v0.1

Resource	Value	Expected Availability
Storage	**0.3 PB** will be available on disk for version 0.1. Additional 1.2 PB on tape available. Upgrade to 2 PB on line disk (S3) and 5 PB tape storage is expected by the end of 2025	*03/2025*
	Negotiate and finalise contracts with vendors	*09/2025*

Compute	During the duration of SRCNet v0.1, the Italian SRC will have access to **0.1 PFLOPS** of computing power (CPU+GPUs) through the TIER3 infrastructure of INAF, integrated with CINECA. There is a possibility of "bursting" onto other systems thanks to the integration with the CINECA supercomputing environment for specific tests to identify in advance, providing further flexibility and scalability.	*03/2025*
	Negotiate and finalise contracts with vendors	*09/2025*
Network	10 Gb/s is the available network bandwidth, that will be upgraded to 100 Gb/s by the end of 2025	*Available*
	High-bandwidth upgrades: 100 Gb/s	*12/2025*

A.5.3.3 Relevant Local Milestones

Milestone	**Date**
Procurement Deadline	*06/2024*
Hardware Upgrade	*11/2024*
Node Ready for Validation	*01/2025*
Node Integrated in SRCNet v0.1	*03/2025*
Limited availability period	*None in 2025*

A.5.2.4 Japan SRC

A.5.4.1 Overview

SKA1 Promotion Group (SKAJ) of National Astronomical Observatory of Japan (NAOJ) will implement and manage the JPSRC node v0.1. JPSRC shall commit to achieve our following v0.1 implementation, if that is asked by the SRCNet.

Following the same percentage for the SKA1 construction, JPSRC shall target a 2% share for the SRCNet construction. JPSRC recognizes that this 2% share is projected as 14 TFlops computing resources, 420 TB storage resources, and 1.3 FTE human resources, based on the SRCNet Top-Level Roadmap document. JPSRC also understands that there are several other requirements that are specified in the v0.1 Expression of Interest form.

One may question how much JPSRC's 2% contribution of resources to the SRCNet v0.1 is critical for SRCNet v0.1 technology demonstration. There are broadly two answers to the question, the geometrical location and the unique HPC solutions, summarised below.

Geometrical location

One of the key features to be tested with SRCNet v0.1 is the distributed data management. It means that, using the adopted mechanism, we can transfer the data and can manage the data in a distributed manner. The SRCNet is a global network linked to the different geometrical locations in this globe; Australia, South Africa, India, Canada, Europe, and East Asia. Through the SRCNet v0.1 implementation, we have to verify as soon as possible that the adopted mechanism works appropriately and efficiently in the long-haul networks among the locations.

At the eastern end of the SRCNet, i.e. in East Asia, China is a primary SRC partner with Korea and Japan committing a small fraction (1-2%) of the resources to the SRCNet. Therefore, it is natural that JPSRC represents a secondary node in this location, but JPSRC (and KRSRC) should make sure by itself the link to/from East Asia without China in terms of mitigating the single-point-failure. Specifically, data transfer from/to South Africa (SKA1 MID) through Indian Ocean (or Atlantic-Arctic Oceans) should be carefully assessed because of no dedicated link, while the other major long-haul links are known to be used in other Japanese academic programs.

Unique HPC solution

JPSRC's software stacks will be like those adopted in other nodes, while JPSRC will adopt two unique hardware/middleware solutions in order to improve the efficiency of operation. Those are the "gfarm file system" and the "SX-Aurora TSUBASA". JPSRC plays a unique role in demonstrating these solutions in the SRCNet v0.1 and beyond. Those solutions can be easily deployed in various systems (say, 1 FTE-week), so that the demonstrations are also beneficial for other SRC nodes.

The gfarm file system is a distributed parallel fault tolerant file system. Gfarm has been developed in the lattice QCD community in Japan and adopted in the common-use data storage for Fugaku, one of the fastest computers in the world, managed by the high performance computing infrastructure (HPCI) in Japan. Gfarm is free in use and surprisingly easy to deploy, so that it is more accessible by other, particularly small, nodes compared to other expensive paid systems.

The SX-Aurora TSUBASA is a PCIe-connected vector engine (VE) processor made by NEC Cooperation. The VE core of TSUBASA has roughly 10 times faster peak performance than a CPU core. It is thus powerful for workflows that can safely work only with a single process/thread. The vectorization is efficiently made by the NEC compiler automatically. Therefore, it does not require massive overheads and human resources for reprogramming needed to obtain an efficient use of the accelerator such as GPU.

A.5.2.4.2 Resources available for SRCNet v0.1

JPSRC shall implement 6 servers for the SRCNet v0.1. Table summarises the specifications of the workstations. All the servers will be under the control of OpenStack and are usable as virtual machines.

ID	CPU	CPU	Accelerator	DRAM	SSD	HDD	Network
	# of cores	FP64 TFlops	FP64 Tflops	GB	TB	TB	Gbps
p001	4 x 3.6 GHz Xeon W-2223	0.46	5.13 NVIDIA Quadro GP100 16GB	128 DDR4-2933 ECC	0.25 M.2/NVMe	24	1.0 → 10*
p003	20 x 2.5 GHz Xeon Gold 6210U	1.6	N/A	128 DDR4-2933 ECC	1.0 M.2/NVMe	N/A	1.0 → 10*
p005	64 x 2.0 GHz AMD EPYC 7713	2.05	2.58 2 x NVIDIA RTX A6000 48 GB GDDR6, PCIe4.0	1024 DDR4-3200 ECC	1.0 M.2/NVMe	162	1.0 → 10*
p006	8 x 2.1 GHz Intel Xeon Silver 4208	0.54	2.15 NEC SX-Aurora TSUBASA 10CE 8 cores, 24 GB	96 DDR4-2400 ECC	N/A	8	1.0 → 10*
p007	24 x 3.2 GHz AMD EPYC 74F3	1.23	N/A	512 DDR4-3200 ECC	1.0 SSD	220 +220*	100
*p008	4 x 3.6 GHz Xeon W-2223	0.46	5.13 NVIDIA Quadro GP100 16GB	128 DDR4-2933 ECC	0.25 M.2/NVMe	17	1.0 → 10*
Switch	N/A	N/A	N/A	N/A	N/A	N/A	Out 10 In 2.5 → 10*
Router	N/A	N/A	N/A	N/A	N/A	N/A	Out 10 In 10
Total	132	6.42 +0.46	9.86 +5.13	1888 +128	3.25 +0.25	414 +237	1.0 → 10*

The "*" means that they will be implemented in January 2025.

Storage

JPSRC shall provide SSD and HDD storages which will be permanently online. Those drives are SATA3-connected, giving 6 Gbps in theory. The HDD storage will be unified as a single file system by gfarm. The capacity of the bulk HDDs is 414 TB in total. But they are distributed in several servers with the bandwidth of 1

Gbps now and 10 Gbps by January 2025. There is a plan to add more 220 TB HDDs on the main file server by the end of January 2025. Then, the file server alone will meet the v0.1 requirement for storage. There is a plan to add one more server with a 17 TB HDD. Therefore, a total 661 TB will be available by the end of January 2025.

Computing

The 6 servers shall provide CPU, VE, and GPU for computing. The theoretical peak performance in total is 6.42 Tflops by CPU (128 cores), and 9.86 Tflops by the accelerators. There is a plan to add one more server with 0.46 Tflops by CPU (4 cores) and 5.13 Tflops by GPU. Therefore, a total 21.87 Tflops will be available by the end of January 2025.

The 6 servers shall have DRAM of 1.88 TB in total, i.e. 14.7 GB per core on average. Most of them are DDR4-2933 or above. There is a plan to add one more server with a 128 GB DRAM. Therefore, a total 2.08 TB will be available by the end of January 2025.

Network

JPSRC will utilise Japan's high speed academic network, SINET6, as a backbone network. The SINET6 provides a 400 Gbps bandwidth for domestic institutes and universities. The SINET6 provides a 100 Gbps bandwidth to/from Amsterdam, which will be doubled (200 Gbps) in the near future. The SINET6 also provides a 100 Gbps bandwidth to/from Singapore. JPSRC will also utilise the JGA network as well. The JGA provides a 100 Gbps bandwidth to/from Sydney via Guam. JPSRC, however, recognises that the use of those academic networks is temporal. The SRCNet must have a dedicated bandwidth for the intercontinental (submarine cable) network that should be arranged by SKAO.

NAOJ connects to the SINET6. However, the current bandwidth inside the building in which the JPSRC servers will be is limited to 10 Gbps. Based on the limitation, the core firewall router of JPSRC used for SRCNet v0.1 supports up to 10 Gbps. There is a possibility to place the JPSRC facilities at other buildings or institutes in order to meet the requirement of the 100 Gbps link in the future.

The network switch supports 10 Gbps to/from the router but supports up to 2.5 Gbps to/from each JPSRC server. This is because most of the network interface cards (NICs) of the servers support only 1 Gbps so far. There is a plan to support 10 Gbps bandwidth by replacing the switch and all of the NICs by the end of January 2025.

Resource	**Value**	**Expected Availability**
Storage	**414 TB** (220 TB for RSE)	*10-12/2024*
	A further 237 TB. *Total **651 TB** (440 TB for RSE).*	*01/2025*

Compute	**16.28 Tflops** (CPU 6.42, Accelerator 9.86)	*10-12/2024*
	*A further 0.46 TFlops CPU and 5.13 TFlops GPU. Total **21.87 TFlops** (CPU 6.88, Accelerator 14.99)*	*01/2025*
Network	Server NICs: **1 Gbps** Core router/hub: 10 Gbps Backbore: >100 Gbps SINET6	*Available now*
	Server NICs: **10 Gbps** Core router/hub: 10 Gbps Backbore: >100 Gbps SINET6	*01/2025*

A.5.4.3 Relevant Local Milestones

JPSRC intends to complete the v0.1 implementation by the end of January 2025, where major functionalities will be met by December 2024. JPSRC will host the Science Data Challenge (SDC) 3b and 4 in 2024–2025, so some resources described above will be occasionally shared with the use for the SDCs.

The delay in software implementation is a risk that JPSRC is facing. To mitigate the risk, JPSRC is considering implementing the system that follows as closely as possible the mini-SRCNet demonstrator developed by the Red team and tested by Coral and Lavender teams.

Milestone	**Date**
Procurement Deadline	*09/2024*
Hardware Upgrade	*10/2024 - 01/2025*
Node Ready for Validation	*10/2024 - 01/2025*
Node Integrated in SRCNet v0.1	*10/2024 - 01/2025*
Limited availability period	*SDC3b, SDC4*

A.5.5 Netherlands SRC

A.5.5.1 Overview

NLSRC is currently a collaboration between ASTRON, the Netherlands Institute for Radio Astronomy, and SURF, the Dutch national e-infrastructure provider and NReN for education and research. ASTRON brings many decades of experience in designing, constructing, and operating radio telescopes — including LOFAR and WSRT/Apertif — and their supporting data infrastructures, while SURF provides professionally-managed infrastructure, including hundreds of thousands of CPU cores, hundreds of PB of storage, and high-bandwidth networking, all based on 100% renewable energy. In the NLSRC context, ASTRON will provide scientific leadership and software development effort, while SURF brings infrastructure and integration expertise.

NLSRC currently operates a small cluster in the SURF data centre. We anticipate scaling this up to meet the demands of the SRC Network. The scale of the SURF facilities give us some flexibility regarding the details of exactly what, when and how infrastructure will be deployed.

A.5.5.2 Resources available for SRCNet v0.1

Resource	Value	Expected Availability
Storage	100 TB bulk storage and 50 TB scratch storage available for the duration of SRCNet v0.1. Additional storage will be available in support of specific tests, identified in advance, on an as-needed basis; given the resources available at SURF, we expect to be able to offer some flexibility here.	*Start of 2025*
Compute	0.01 PFLOPS available continuously for the duration of SRCNet v0.1. We expect to provision this as ~250 CPU cores, but can potentially allocate GPUs if requested. We may also be able to support “bursting” onto other systems in support of specific tests identified in advance.	*Available now*
Network	*Baseline connectivity established (10 Gbps available for testing)*	*Available now*

A.5.6 Spanish SRC

A.5.6.1 Overview

The espSRC is the Spanish prototype of a SKA Regional Centre. The espSRC is a cloud computing platform based on OpenStack (version Antelope 2023.1) and Ceph (version Quincy 17.2.7). It currently consists of a cluster of 6 computing nodes (hypervisors), 3 control nodes, 3 monitoring nodes, and 8 storage nodes. Each hypervisor is equipped with 40 CPU cores, with one offering a memory/core ratio of 25GB, while the remainder offer 9.4GB per core. The espSRC raw storage capacity is 1.3PB and effective storage ~650PB. All cluster nodes are interconnected with a 100 Gbps Ethernet network for data and computing. The espSRC platform shares the 10 Gbps link of our Institute to RedIRIS-NOVA, the Spanish academic network.

This infrastructure provides the Spanish community with computing and storage resources to support their scientific studies, especially those involving SKA precursors /pathfinders data. It also supports the SRCNet prototyping activities and several SRCNet prototypes have been deployed on the espSRC, highlighting the SKA Rucio Data Lake, the Mini-SRCNet demonstrator and the ESAP Science Platform.

The espSRC is currently carrying out a procurement process to acquire new hardware to expand its computing and storage resources. This expansion will consist of ~1PB of additional raw storage capacity and ~384 CPU cores (12GB memory/core). These additional resources will be integrated into the espSRC OpenStack/Ceph platform. This expansion will allow espSRC to provide support to the Spanish community and the SRCNet prototyping activities as well as to fulfil Spain's allocated computing and storage resources as outlined in the SRCNet Roadmap document for the v0.1 phase.

While the new hardware is acquired and installed, the espSRC can allocate resources for the SRCNet v0.1 using its current infrastructure.

A.5.6.2 Resources available for SRCNet v0.1

Resource	**Value**	**Expected Availability**
Storage	~1 Petabyte raw storage capacity (we will apply 2 factor redundancy, so we will be able to host ~0.5PB of data) provisioned using SSD and mechanical disks	*Target date: December/2024*
Compute	0.01 PFLOPS (provisioned with CPUs)	*Target date: December/2024*
Network	10 Gbps link to RedIRIS	*Available now*

A.5.6.3 Relevant local milestones

Milestone	**Date**
Procurement Deadline	*Target date: November/2024*

Hardware Upgrade	*Target date: December/2024*
Node Ready for Validation	*Some tests could be executed on our current infrastructure before the hardware expansion expected by December/2024.*
Node Integrated in SRCNet v0.1	*Target date: January / 2025*
Limited availability period	*August. During the hardware and software upgrade expected in December 2024*

A.5.7 Sweden SRC

A.5.7.1 Overview

The Swedish SKA Regional Center (SWE-SRC) is based at Onsala Space Observatory (OSO), Chalmers University of Technology. OSO is the national radio astronomy infrastructure for Sweden, and Chalmers is the administrative body for Sweden's representation in the SKA. At OSO we host several existing radio astronomy facilities (Onsala 20 m, Onsala 25 m, Onsala Twin Telescopes, a LOFAR international station, VLBI, Geodetic VLBI) and have experience running a center in support of large international telescope facilities through the Nordic ALMA Regional Center (ARC).

SWE-SRC is a close collaboration between OSO and Chalmers e-Commons HPC center. Broadly speaking personnel at OSO provide scientific leadership and software development, while e-Commons provides infrastructure and integration expertise. In practice, we have scientific expertise embedded at e-Commons leading our local SRC node development and deployment; as well as infrastructure and network experience at OSO, for example, in support of ongoing telescope operations.

Sweden has an expected 1.5% share in the SKA (with the exact share of SRCNet resources provided as we approach the full operations phase aimed to equal the assessed overall Swedish SKA share at that point) with the resources that will be available for v0.1 by Dec 2024 being in line with that 1.5% level. In addition, we have already acquired funding to support expansion of resources. Given current cost estimates, we believe that procurements made in 2024 for additional compute, and in 2025 for additional storage will prepare us to be able to deliver resources beyond the requirements for v0.2 in advance of the timeline in the SRCNet Roadmap.

A.5.7.2 Resources available for SRCNet v0.1

Resource	Value	Expected Availability
Storage	**300 TB** online storage. **30 TB** scratch storage.	*Available now.*
	Expected upgrade to 2.5 PB online storage and 250 TB scratch storage.	*Upgrade bulk storage target date: before 01/2026. Scratch storage: before 01/2025.*
Compute	**10.5 TFLOPS**	*Available now.*
	Expected upgrade to 85 TFLOPS.	*Upgrade target date: before 01/2025.*
Network	**100 Gbps**, shared network.	*Available now.*
	Upgrade is possible, but to what and when needs to be established first, through demonstrated needs.	

A.5.7.3 Relevant Local Milestones

Milestone	Date
Procurement Deadline	*Deadline for vendors is 24/05/2024; one month evaluation; decision & order before 07/2024*
Hardware Upgrade	*See table 3.2.7.1*
Node Ready for Validation	*Target date: 10/2024*
Node Integrated in SRCNet v0.1	*Target date: 12/2024*
Limited availability period	*Most of July; 2-3 weeks around the end of Dec.*

A.5.8 Switzerland SRC

A.5.8.1 Overview

The Switzerland SRC's hardware resources will largely be made available by the supercomputing centre CSCS, in Lugano. While infrastructure level operations will be handled by engineers onsite, application and service level management will be done by people distributed in other SKACH institutes (including but not limited to EPFL, and FHNW).

With current bridging funding, in January 2025 we can provide resources in line with Switzerland's 2% share in the SRCNetv0.1.

We will apply for competitive funding for compute and storage of CHSRC in 2024. In the likely event of a positive decision, we expect the funding to be available summer 2025. Until this funding is awarded, we only have limited bridging funds to provide compute and storage for the CHSRC.

A.5.8.2 Resources available for SRCNet v0.1

Resource	**Value**	**Expected Availability**
Storage	**100 TB**	*Available now*
	400 TB.	*Jan 2025*
Compute	Currently, there are sufficient compute resources to run SRCNet services via kubernetes, or Openstack VMs as applicable.	*Discussions ongoing with CSCS to provide compute nodes on the new hardware infrastructure.*
	14 TFlops (approximately 3 nodes at CSCS) *The new infrastructure consists of diskless AMD nodes, 2x64=128 cores (256 with hyperthreaded) and 256/512 GB RAM.*	*Jan 2025*
Network	One link of 400 Gbps connection and 2 links of 100 Gbps to the internet, shared. IPv6 capable.	*Available now*
	The what and when of network connectivity upgrade needs to be established first, through demonstrated needs.	

A.5.8.3 Relevant Local Milestones

Milestone	Date
Procurement Deadline	*No further procurement at CSCS at this stage. Positive outcomes of funding rounds in 2025 will help expand on resources at CSCS.*
Hardware Upgrade	*We will apply for competitive funds Dec 24, and hope to hear of a positive decision in summer 2025.*
Node Ready for Validation	*Jan 2025*
Node Integrated in SRCNet v0.1	*Jan 2025*
Limited availability period	-

A.5.9 UK SRC

A.5.9.1 Overview

UKSRC will deploy the majority of resources for SRCNet v0.1 to the Rutherford Appleton Laboratory (RAL), a STFC hosted national facility. The bulk of the compute will be added to the OpenStack based STFC cloud. This provision is part of the IRIS e-Infrastructure community (https://www.iris.ac.uk/), which is a cooperative federated digital research infrastructure community . Currently, bulk storage is provided via development storage, and is available for SRCNet (already incorporated into the existing Rucio testbed). Dedicated storage for UKSRC is in the procurement process and will be available prior to SRCNet v0.1. This storage will be deployed to the same configuration management environment as existing petascale ceph datastores at RAL. The UKSRC ultimately expects to distribute its storage and compute resources across a number of physical locations, exploiting federated access and the heterogeneity of the available resources. As such, UKSRC may wish to bring additional sites into SRCNet v0.1 at an appropriate time and with agreement of the SRCNet Operations Group.

A.5.9.2 Resources available for SRCNet v0.1

Resource	Value	Expected Availability
Storage	In excess of 100TB is available now via development storage.	Now

	4PB of dedicated SRC usable storage (POSIX, HDD, cephFS , Erasure Coded) is in procurement.	Operationally available Jan 2025
Compute	Access to sufficient IRIS resources for work necessary to prepare for v0.1 in OpenStack Cloud infrastructure at RAL is currently available.	Available now
	Additional computing resources, procured by UKSRC, and available within the IRIS community, up to 0.175 PFlops, will be ramped up in early 2025. Significant compute resources required prior to this date available by arrangement.	Apr 2025
Network	The RAL Facility has 400Gb/s connectivity across all users. UKSRC DTN hosts have 100GbE interfaces installed. Compute nodes have at least 25GbE interfaces.	Available now
	UKSRC will continue to update its NREN about expected future bandwidth requirements, supporting these requests with clear, demonstrable usage along the planned roadmap.	Continuing

A.5.9.3 Relevant local milestones

Milestone	**Date**
Procurement Deadline	For storage, the procurement process is estimated to be completed by June 2024. There are currently extended lead times for network switches and deployment is contingent on this. For Compute, this will follow the established procurement cycles within the STFC Cloud infrastructure.
Hardware Upgrade	ETA for 4PB, commissioned and available: Dec 24.
Node Ready for Validation	Dec 24
Node Integrated in SRCNet v0.1	Jan 25
Limited availability period	Mid October 24; 3 day at risk period for electrical testing works.

A.5.10 KR SRC

A.5.10.1 Overview

KRSRC consists of KASI, KISTI, UNIST, and other Univs. KASI will be assignee of Korea SKA project including SRC. KISTI has high abilities on network connectivity. UNIST is supporting SKA project aggressively. Official process on SKA is not constructed, but attendance on SRCNet v0.1 is possible. SRCNet v0.1 on KRSRC will be interim-SRCNet to fulfill a share of SRCNet v0.1. UNIST will undertake preparation of resources, and KASI will support installation of SRCNet on the resources. H/W is prepared mostly, by switching other resources to SRCNet v0.1. There will be not critical modifications, but external network is absent on the resource. UNIST has 10G external inside for other usages. It is on positive discussion linking with the SRCNet v0.1.

Note: This contribution was proposed after the initial EoI gathering

A.5.10.2 Resources available for SRCNet v0.1 (preliminary-work ongoing)

Resource	Value	Expected Availability
Storage	**277 TB** (197TB for RSE) 10*8TB for user (HDD, CephFS) +40TB(not included, for K3S)	*9/2024* *H/W prepared*
Compute	**11.26+alpha TFLOPs**	*9/2024* *H/W prepared*
Network	**10gbps, internal** **1gbps, external** -> 10gbps (on discussion)	*11/2024*

LIST OF ABBREVIATIONS

AA	Array Assembly
API	Application Programming Interface
AAI	Authentication and Authorization Infrastructure
ART	Agile Release Train
CASA	Common Astronomy Software Applications for Radio Astronomy
CI/CD	Continuous Integration and Continuous Delivery
DTN	Data Transfer Node
ESO	European Southern Observatory
FPGA	Field-Programmable Gate Array
GMS	IVOA Group Membership Service
GPU	Graphics Processing Unit
IAM	Identity and Access Management
IaC	Infrastructure as Code
IOTA	Infrastructure as Code Testing Appliance
JIRA	Issue and project tracking system
NREN	National Research and Education Network
OIDC	OpenID Connect
OSKAR	Open SKA Key-Value Store
POSIX	Portable Operating System Interface
Prometheus	Open-source monitoring system
RASCIL	Radio Astronomy Software CALibration Interface Library
SDP	Science Data Processor
SemVer	Semantic Versioning
SKA	Square Kilometre Array

SP	Science Platform
SSD	Solid-state drive
SRC	SKA Regional Centre
SRCNet	SKA Regional Centres Network
STARS	Science Test And Run System
VRF	Virtual Routing and Forwarding
WAN	Wide Area Network

POLICY APPROVAL AND OWNERSHIP DETAILS

POLICY APPROVAL AND OWNERSHIP DETAILS	
Impact Assesment:	Yes/No
Date of Review:	YYYY-MM-DD
Document Location:	[Path link]
Compliance Checks:	Yes/No

DOCUMENT HISTORY

Revision	**Date Of Issue**	**Comments**
01	2024-06-11	First release of the document