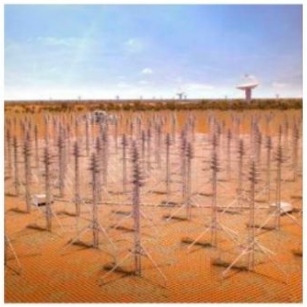
Road to SRCNet v0.1

SKACH Spring Meeting

Carolina Lindqvist, Pablo Llopis, Rohini Joshi

What is SRCNet?



SKAO

130PF Supercomputer

SKA1-Low (Aperture Arrays) → SKA Central Signal Processor ↔ SKA Science Data Processor → 350 PB/year

SKA1-Mid (Dishes) → SKA Central Signal Processor ↔ SKA Science Data Processor → 350 PB/year

Observatory Data Products

SRC Science Analysis Platform Vision Document de Boer, et al (2023)

SRC | Net

SKAO Regional Centre Network

SKA Regional Centre

SKA Regional Centre

SKA Regional Centre

SKA Regional Centre

...

SRC Science Gateway

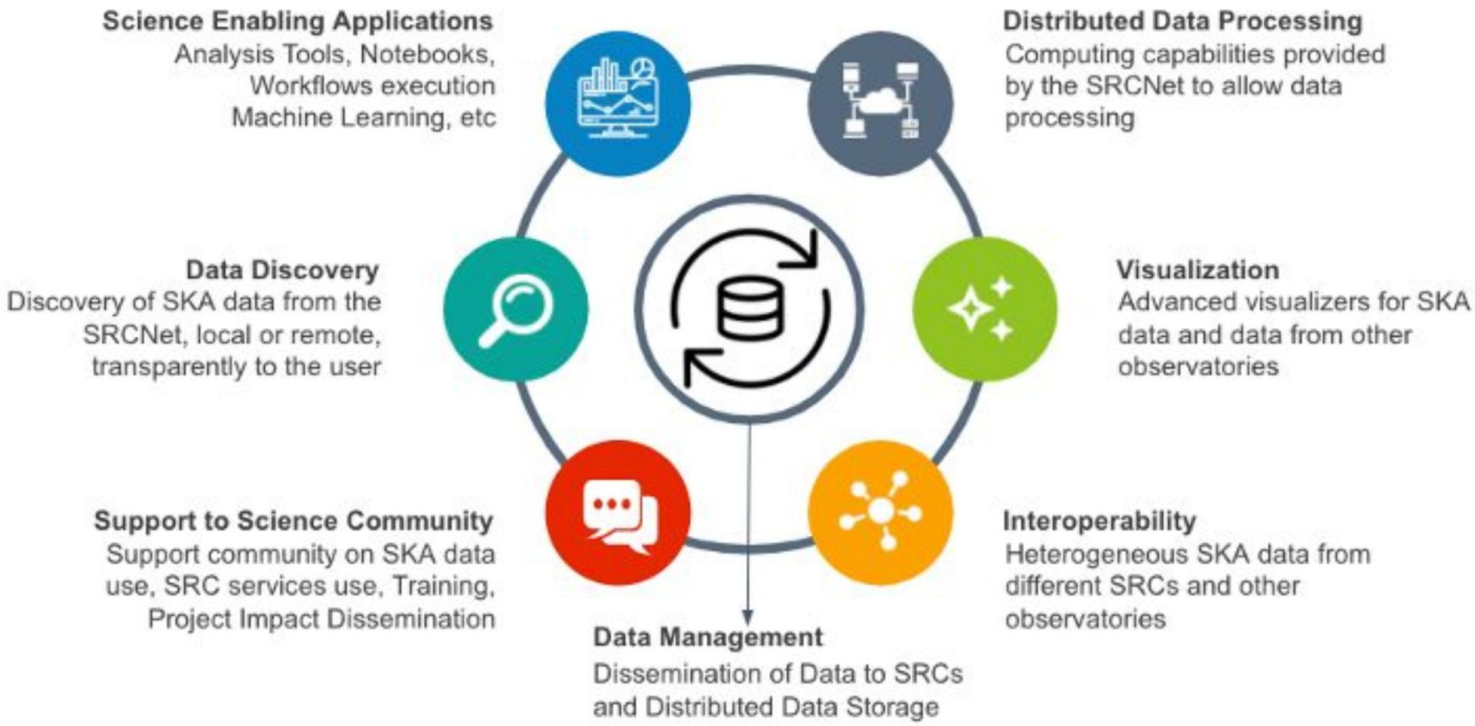
User

User

User

User

Why is SRCNet critical for the scientific user community?



SRCNet v0.1 Milestones and Objectives

v0.1 is the first SRCNet release!

Objectives:

Create a shared network of computing, storage, and network resources distributed across various international facilities

Initial deployment scenario:

- Include all SRCs that expressed an interest in contributing resources
- Provide an implementation plan and a comprehensive guide for deployment activities

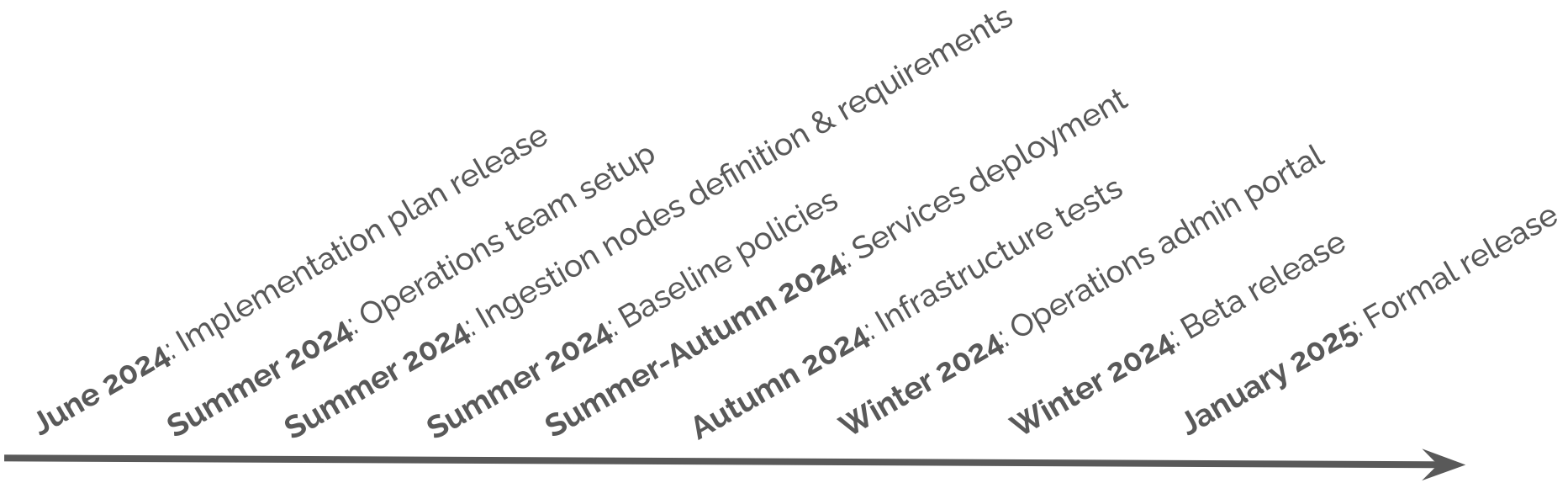
SRCNet v0.1: Target Audience

Engineers responsible for **deployment**, **benchmarking**, and **maintenance** of SRCNet infrastructure.

Engineering staff involved in setup and **management** of **computing**, **storage**, **network** resources.

Engineers involved in the **development** of software services for the **SRCNet v0.1 stack**.

Timeline and Key Milestones



Participating SRCs and resources

	SP SRC	NL SRC	SW SRC	UK SRC	CH SRC	CN SRC	CA SRC	JP SRC	IT SRC	KR SRC	Total
Storage (PB)	0.500	0.100	0.300	4.000	0.400	1.000	1.200	0.651	0.300	0.270	8.711
Compute (PFLOPS)	0.010	0.010	0.011	0.175	0.014	0.175	0.040	0.022	0.100	0.010	0.567
Percentage Storage (%)	5.740	1.148	3.329	45.919	4.592	11.480	13.776	7.473	3.444	3.100	
Percentage Compute (%)	1.765	1.765	1.853	30.891	2.471	30.891	7.061	3.883	17.652	1.765	
Harmonisation Rate *	0.020	0.100	0.036	0.044	0.035	0.175	0.033	0.034	0.333	0.037	

Software Stack and Services

Auth APIs
SKA-IAM
GMS

Data Management
Rucio
FTS
Data Ingestion

Federated Execution
Execution Broker
Software distribution (Registry, CVMFS)
Permissions API
Site Capabilities AP

Monitoring Services
Operations portal, dashboard

SKA GitLabl

Storage
Rucio Storage Element

Compute
Science Platform
Visualisation services
IVOA SODA
Notebooks
HPC
Container registry
Orchestrator

Monitoring
Internal services monitoring, dashboard

Deployment Strategy

Infrastructure as Code

GitOps

Risks and Mitigations

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 Team**, composed of members of different SRCs to divide the overhead of writing procedures, analysis of errors and reduction of complexity by sharing knowledge.

Risks and Mitigations

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.

Risks and Mitigations

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

Beyond v0.1

- Opening access to scientific users.
- Compute APIs (federated) especially single point of entry to SRCNet.
- Nearline storage.
- Accounting.
- Scaling up functionality first, scaling up resources second.

Q & A


srcnet
storage
will
data
resources
services
compute
node
nodes
src
service
tests
test
v0.1
local
sites
global
available
network
infrastructure
software
access
integration
milestones
policies
gbps
operations
science
expected
ska
performance
management
repository
support
relevant
may
tflops
site
dashboard
validation
procurement
description
campaigns
team
development
ingestion
transfer
requirements
deployment
hardware
provide
specific
jpsrc
configuration
common
distributed
additional
level
system
rucio