

# Execution interface prototype

*Yan Grange*

On behalf of Team Tangerine, SRCNet

*IVOA Northern-fall Interop. 21 May 2024 (Sydney and remote)*

# The SKA Regional Centres Capabilities of the SRC Network

## Science Enabling Applications

Analysis Tools, Notebooks,  
Workflows execution  
Machine Learning, etc

## Distributed Data Processing

Computing capabilities provided  
by the SRCNet to allow  
data processing

## Data Discovery

Discovery of SKA data from the SRCNet,  
local or remote, transparently to the  
user

## Visualization

Advanced visualizers for  
SKA data and data from  
other observatories

## Support to Science Community

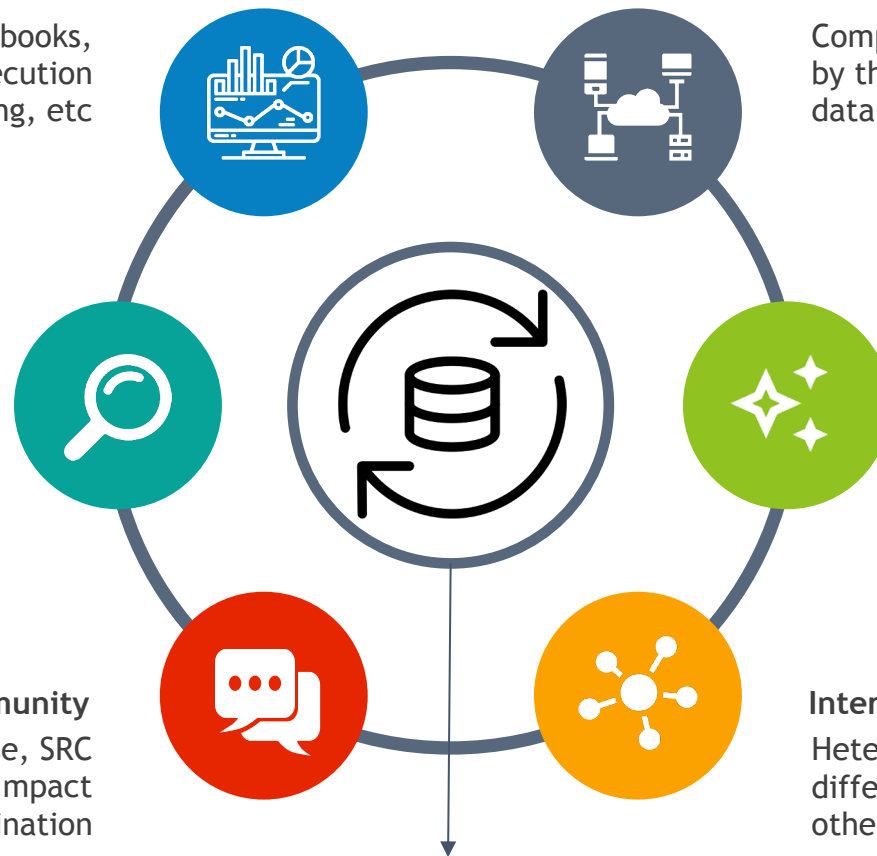
Support community on SKA data use, SRC  
services use, Training, Project Impact  
Dissemination

## Interoperability

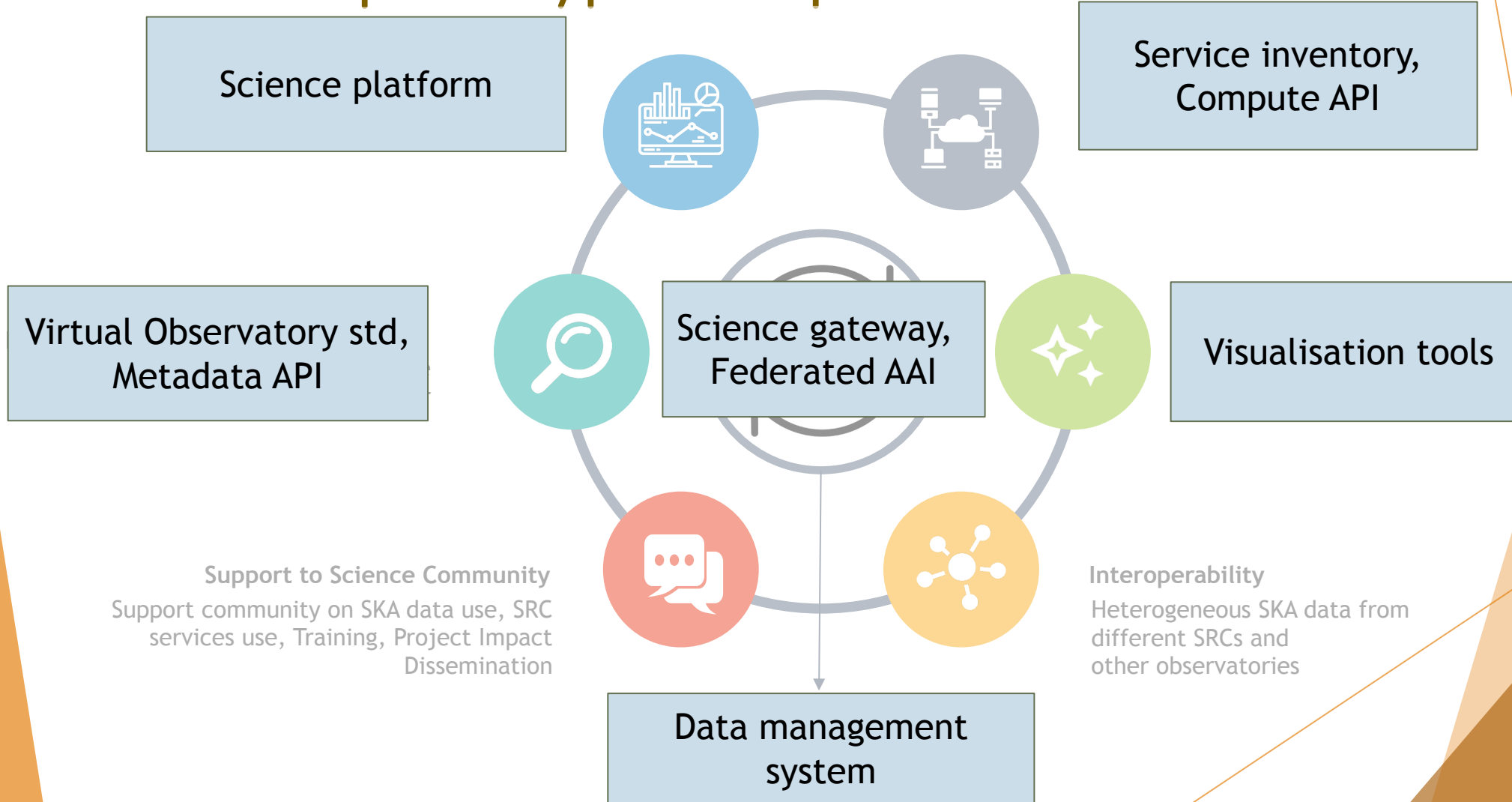
Heterogeneous SKA data from  
different SRCs and  
other observatories

## Data Management

Dissemination of Data to SRCs and  
Distributed Data Storage



# The SKA Regional Centres SRCNet prototype components

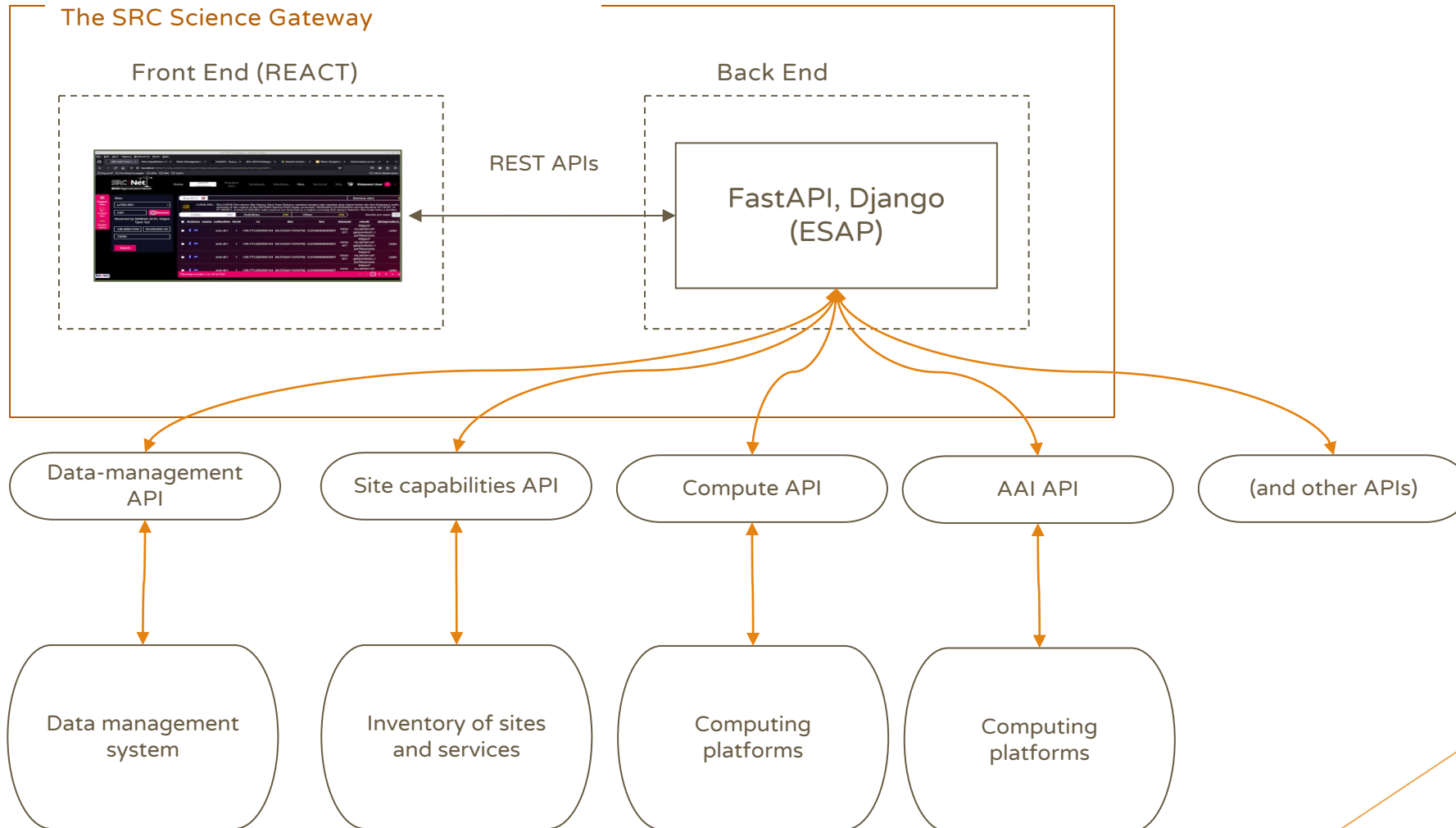


# The science gateway

The screenshot shows the SRC Net science gateway website. The header includes the SRC Net logo and navigation links: Home, Search catalogue, Search compute resources, Notebook, and Visualise data. A 'Log in' button with a lock icon and a language selector for 'English' with a UK flag are also present. A left sidebar contains three project selection buttons: Project One, Project Two, and Project Three. The main content area features a large SRC Net logo and the text 'SKA Regional Centre Network'. On the right, a 'Service Status' panel lists several APIs, each with a green status indicator:

Service Status	
IAM	●
APIs	
Authentication API	●
Site-capabilities API	●
Data-management API	●
Gateway-backend API	●
Permissions API	●

# The science gateway (2)



# Initial use case

- ▶ User login to gateway
  - ▶ NB AAI system uses tokens with limited scope, so each step below starts with exchanging for a token with specific access.
- ▶ User queries DM system for data
- ▶ User queries service catalogue for services they want to use for processing
  - ▶ E.g. Jupyter lab, SLURM, CANFAR, ...
- ▶ User provisions processing system, and launches processing on end point from query above  
*(implementation is currently a demonstrator only)*
- ▶ User downloads output data products

# Compute provisioning and execution: initial reference API



- ▶ */query*
- ▶ */provision*
- ▶ */{provision\_id}/submit*
- ▶ */{provision\_id}/status*

## Query json example:

```
{
  "data_location": "democity",
  "data_size": 60000,
  "output_data_size": 8000,
  "memory": 196,
  "cpu_cores": 16,
  "runtime": 72,
  "gpu_model": "K40",
  "deadline": "2024-04-27T22:10:13.584915"
}
```

## Submit json example:

```
{
  "container": "astroimaging/sourcefinder:3.4",
  "dataset": "https://webdav.data.skao.int/3811823/dataproduct.fits",
  "params": {},
  "contact_email": "jdoe@astro.demouniversity.org"
}
```

**Query** ▾

**POST** Query for general compute availability.

**Submit** ▾

**PUT** Query for general compute availability and provision resources.

**PUT** Submit job for the provision.

**GET** Status information for a submitted job.

# Integration of compute API

- ▶ Current flow contains a lot of user steps
  - ▶ Find compute system, reserve resources, submit job, follow-up on job
- ▶ Also based on a lot of simplifying assumptions
  - ▶ There is always processing near (a cthe data
  - ▶ Users know exactly what infra they need
  - ▶ Users know exactly what software they need
- ▶ Integration with the ExecutionBroker standard would solve some of those.



## Next steps for the prototyping

- ▶ Integrate / combine the compute API with the ExecutionBroker standard
- ▶ Gain awareness of lessons learned from existing workflow management systems
- ▶ Integrate the EB-based API with the science gateway
- ▶ Integrate the EB-based API with backend processing systems