SKA Regional Centres and SRC Data Management

Rohini Joshi, Rob Barnsley

Operations Data Scientist, SKAO

PUNCHLunch Seminar, DESY
June 30th 2022

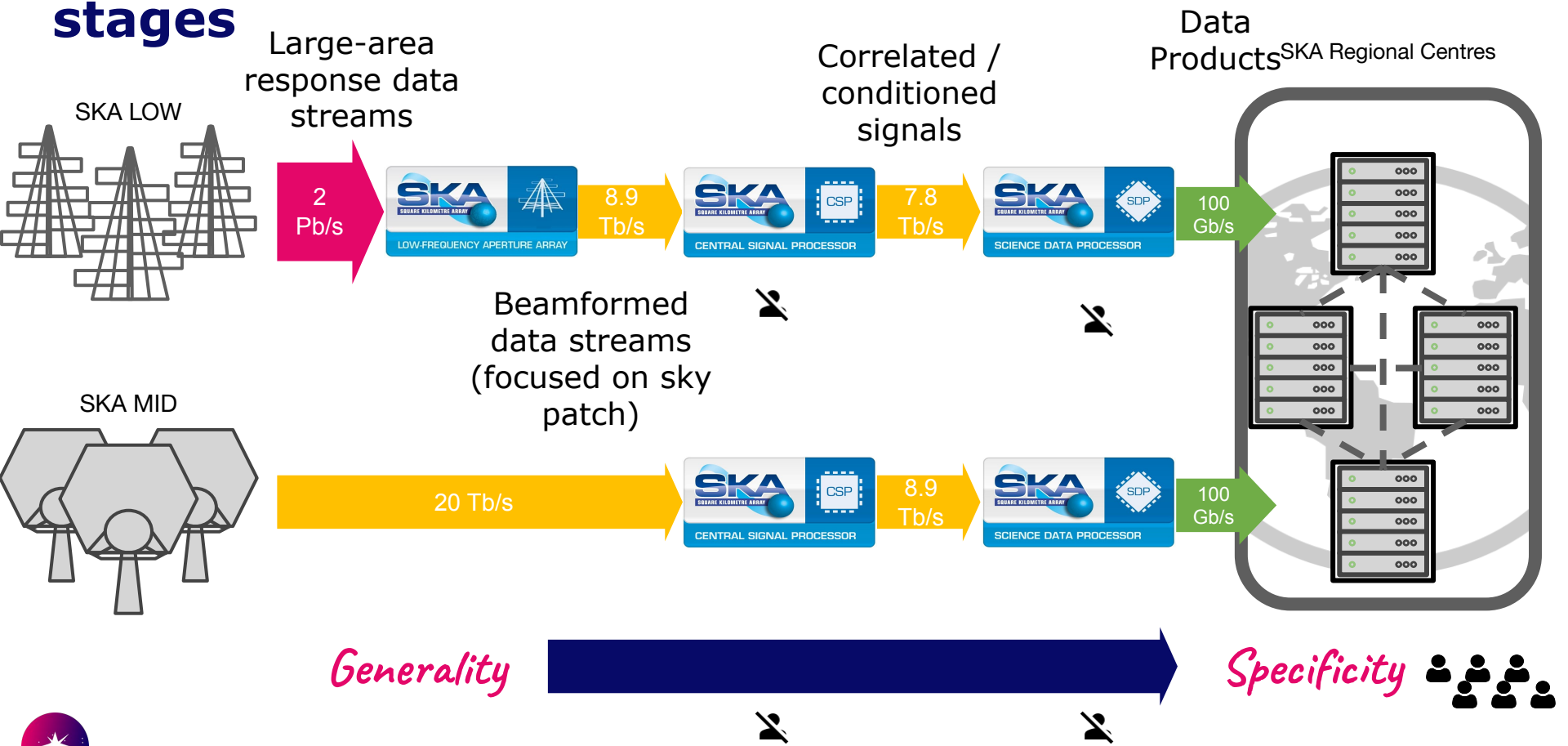
Overview

- SKAO and the SKA Regional Centres
- Data Reduction and the Role of SRCs
- SKAO Rucio prototype
- Rucio metadata

SKAO Mission

"The SKAO's mission is to build and operate cutting-edge radio telescopes to transform our understanding of the Universe, and deliver benefits to society through global collaboration and innovation."

SKA Regional Centres: SKAO data processing stages



How do users "control" data products?

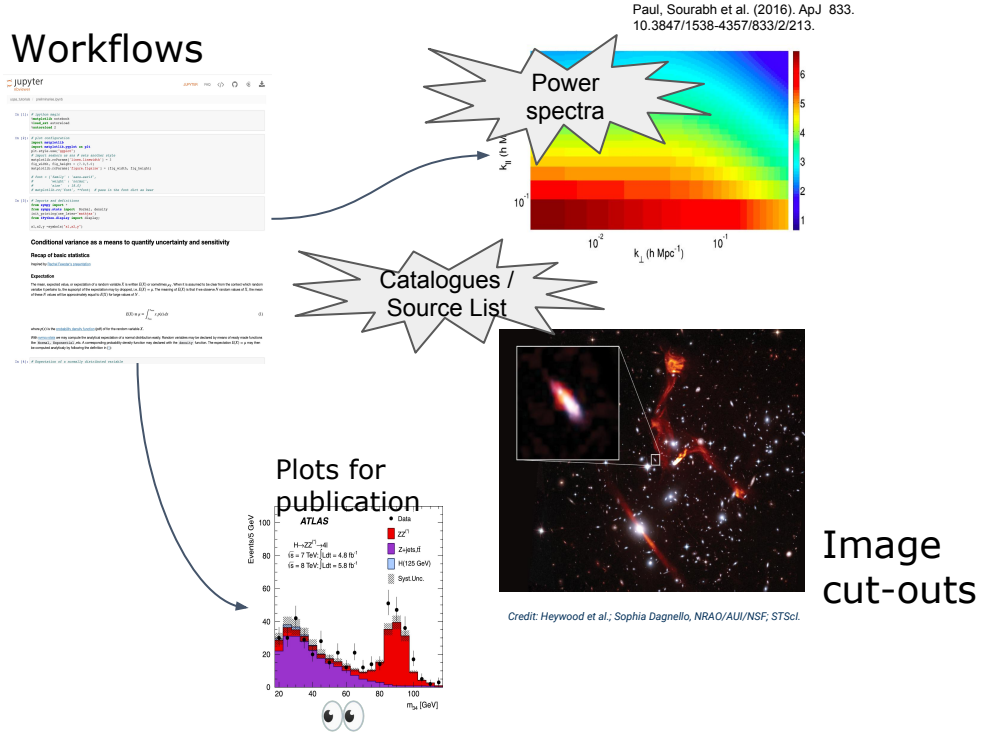
- Need to satisfy users whilst retaining control of SKAO resources
- Science Data Processor is fundamentally part of each SKA Telescope
- Users will specify required data products in Observing Proposals
- All user interaction with data products will be in SRCs

The Role of SRCs: Collaboration platform

SRCs will bridge the gap between the highly data intensive **pre-defined workflows generating SKA data products** in the SDP, and the **iterative flexible, user-led data analysis** required to produce scientific results

SRCs will provide collaborative tools backed up by **powerful compute and data management**

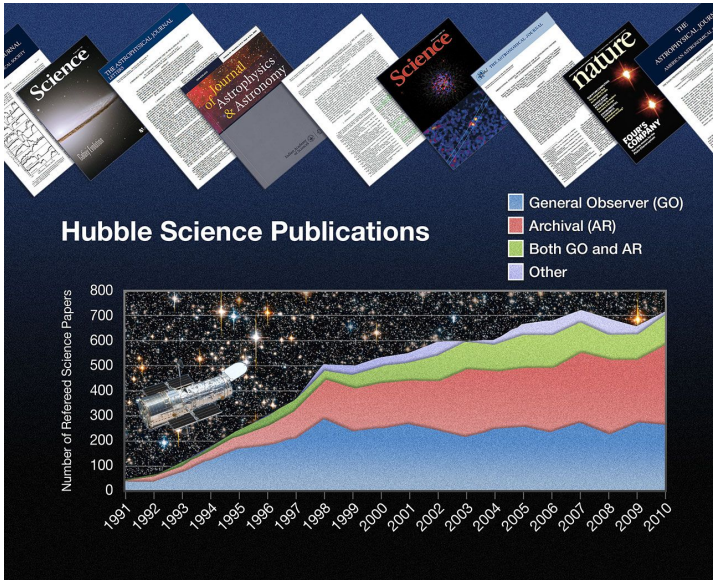
Users will not have access to the SDP or to Raw SKA data!



The Role of SRCs: Support data product (re-)use

Why

- All SKA Data Products will (in time) become public - this is likely to be the biggest science generator (see Hubble)
  - Build SKA science archive around IVOA standards
  - Ensure interoperability with other archives and other experiments

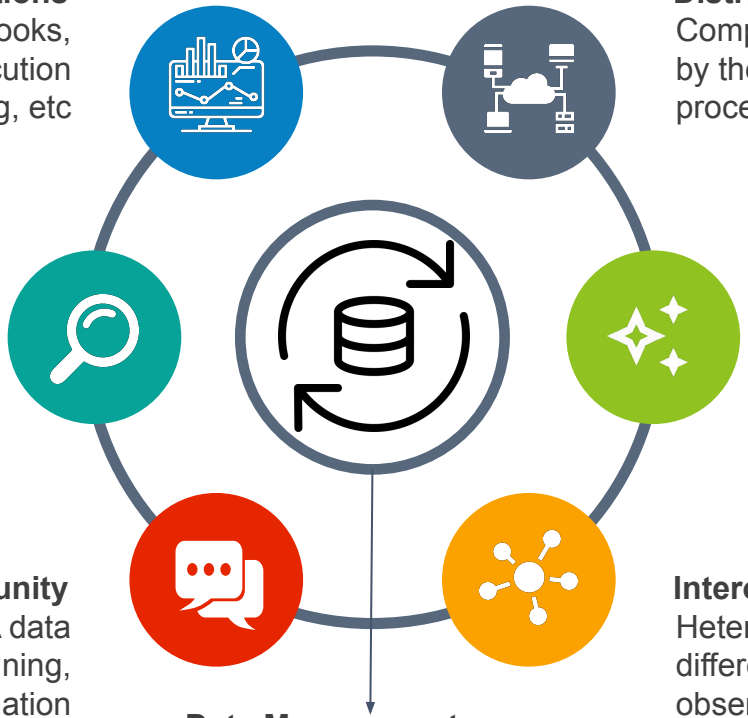


SKA Regional Centre Capabilities

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

SKA Regional Centers: Data management

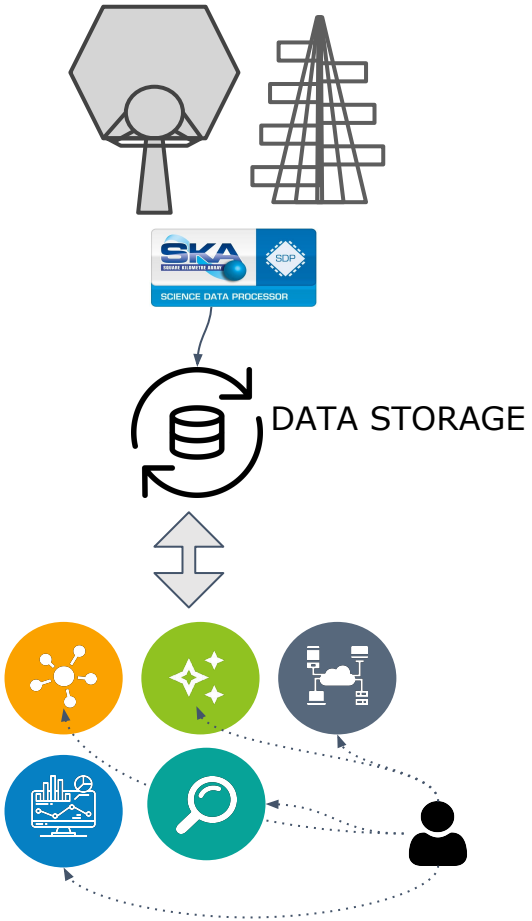
Storing SKAO data growing at up to 700 PBytes each year will be a challenge (plus user-generated data too).

Several million dollars per year in new data, for one copy

Global data management within SRCNet should enable best possible use to be made of available storage resources

Avoid (reduce) unnecessary duplication

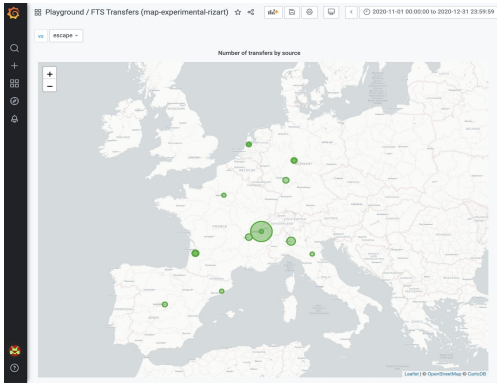
Support mirroring of popular data products to enhance user experience

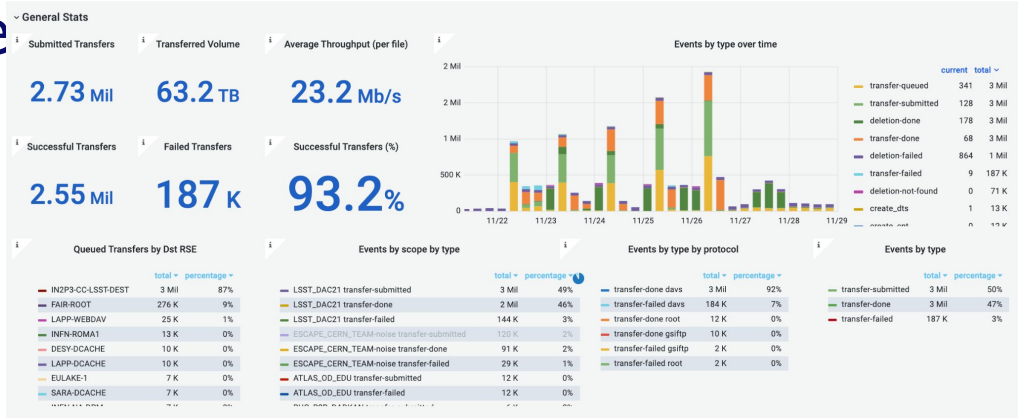


ESCAPE Data Management

ESCAPE WP2 DIOS collaboration - CERN as lead, but developing real interest from several Astro-Particle HEP Experiments

CTAO, KM3NET, LOFAR, SKAO, FAIR





SKAO Rucio prototype

Judicious re-use of existing stack from ESCAPE (eg FTS, storage, IAM)

Well suited to centralised Operations model for data management

Performed long-haul transfers, Rucio stress tests, subscriptions (via our automated test framework)

Aim to integrate storage from national SRC efforts to increase understanding and inform assessment

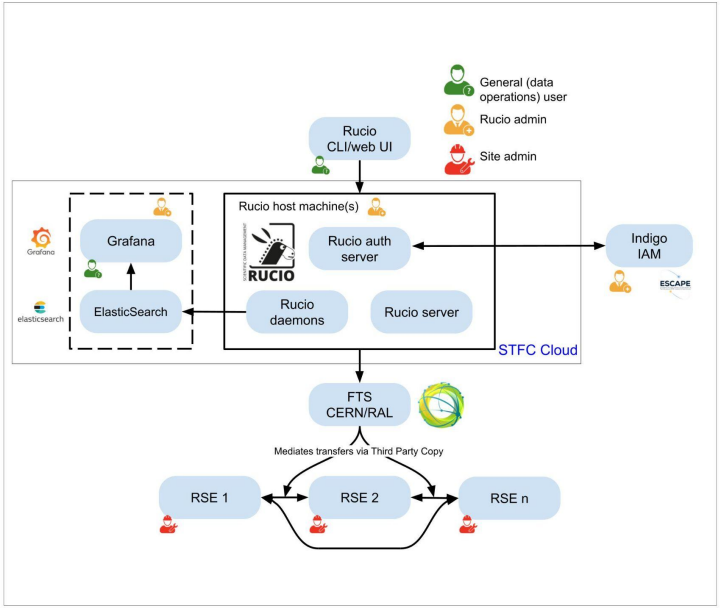
SKAO Rucio prototype

Kubernetes based deployment on STFC Cloud resources

Currently managing two instances as we attempt to move to token-based Rucio

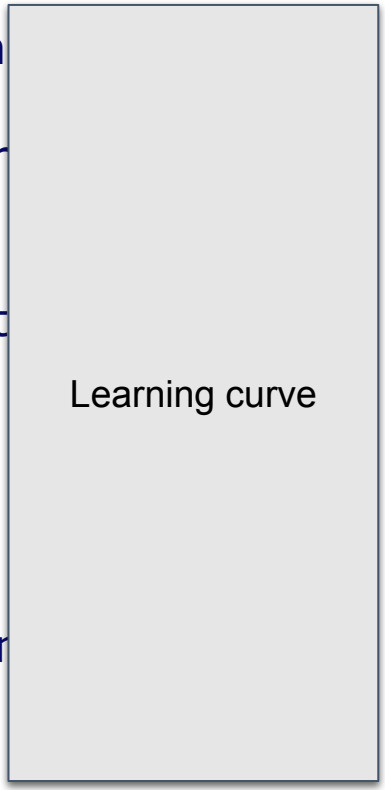
Storage endpoints include gridFTP, Dcache, StoRM-webdav, EOS

Deployment journey (ongoing) captured here: https://gitlab.com/ska-telescope/src/ska-rucio-prototype/



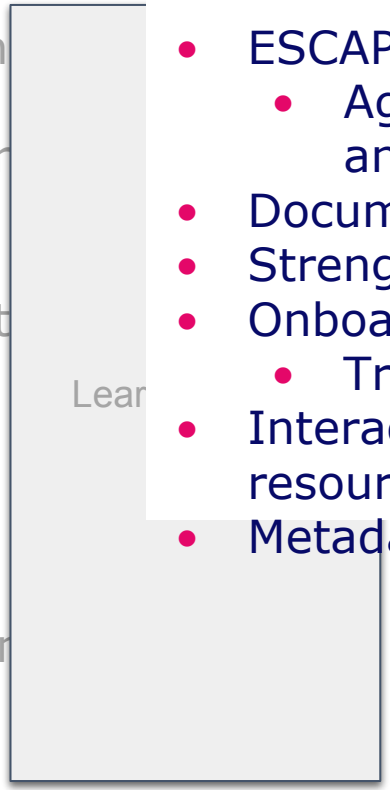
Areas for improvement

- Tokens and OIDC flows are in active state of development
  - Config needed at all layer the stack (storage, FTS, Rucio)
- Authentication and Authorizat
  - Embargoed data and fine grained auth
- Storage integration
  - Onboarding astronomy storage sites
  - Object storage integratior



Areas for improvement

- Tokens and OIDC flows are in active state of development
  - Config needed at all layer the stack (storage, FTS, Rucio)
- Authentication and Authorizat
  - Embargoed data and fine grained auth
- Storage integration
  - Onboarding astronomy storage sites
  - Object storage integratior



Looking ahead

- ESCAPE -> SRCSC
  - Agile processes for building and delivering the SRCNet
- Documentation
- Strengthening k8s infrastructure
- Onboarding new storage sites
  - Transfer optimisation
- Interaction with compute resources
- Metadata

Metadata: Recent work done (Filtering engine)

- For SKAO, metadata is a very important topic

- Out of the box, Rucio has two distinct metadata stores:
  - A **base** metadata store that uses columns from a database table (dids); these are fixed fields, often HEP specific or related to file metadata
  - A **custom** metadata store that stores key-value pairs in a json column in another database table (did_meta)

- Up until version 1.27, there was limited support for querying metadata. Listing files (via the `list-dids` and `list-dids-extended` functions) allowed filtering results by constructing a filter string containing only key-value equalities and the logical AND operator

- From 1.27 onwards, these functions pass the filter string through a new "filtering engine". This engine supports an extended syntax including inequalities and wildcards, viz =, !=, >=, <=, >, <, LIKE, NOT LIKE, the logical OR operator, as well as support for dates e.g.

```
[root@5348da215fc2 rucio]# rucio list-dids-extended test:* --filter "test_key_3 >= 1, test_key_3 <= 3; test_key_2 >= 2022-06-01"
```

Metadata: Recent work done (Plugins)

- Rucio uses a plugin system allowing one to write external metadata interfaces by overriding base functions for how to get/set/delete/query metadata
- From 1.28.0 onwards, a new metadata plugin to interface to an external mongodb collection. This can be used by specifying it in the rucio configuration file as e.g.

```
[metadata]
plugins = rucio.core.did_meta_plugins.mongo_meta.MongoDidMeta
mongo_service_host=mongo
mongo_service_port=27017
mongo_db=test_db
```

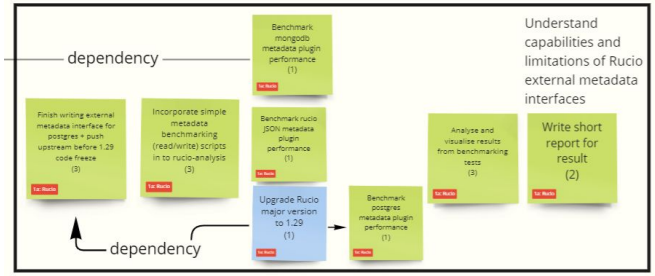
- From 1.29.? onwards, another metadata plugin to interface to an (external) postgres database will be available, e.g.

```
[metadata]
plugins = rucio.core.did_meta_plugins.postgres_meta.PostgresJSONDidMeta
postgres_service_host=postgres
postgres_service_port=5433
postgres_db=metadata
postgres_user=rucio
postgres_password=secret
postgres_db_schema=public
postgres_table=dids
```

- Adding an external RDBMS such as postgres may help with IVOA integrations, e.g. TAP access

Metadata: Future

- SKAO has plans to test e.g. how performant searching is, and how this scales with number of files and complexity of query, but when this is done is restricted by our project roadmap



- These tests will be made part of our testing framework, rucio-analysis
- Welcome suggestions to this component
- #metadata to follow the metadata SIG on Rucio Slack