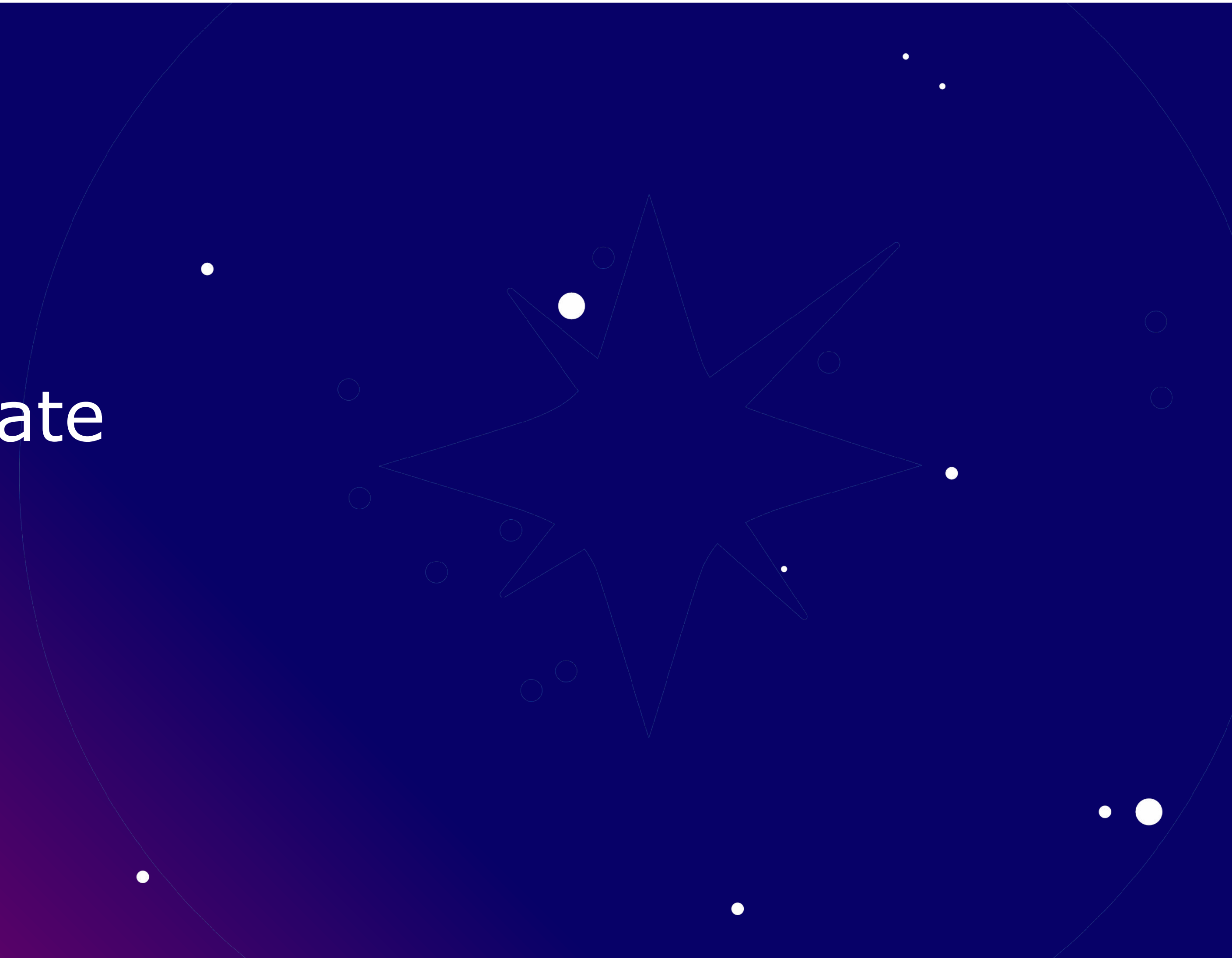


Rucio Metadata SIG Update



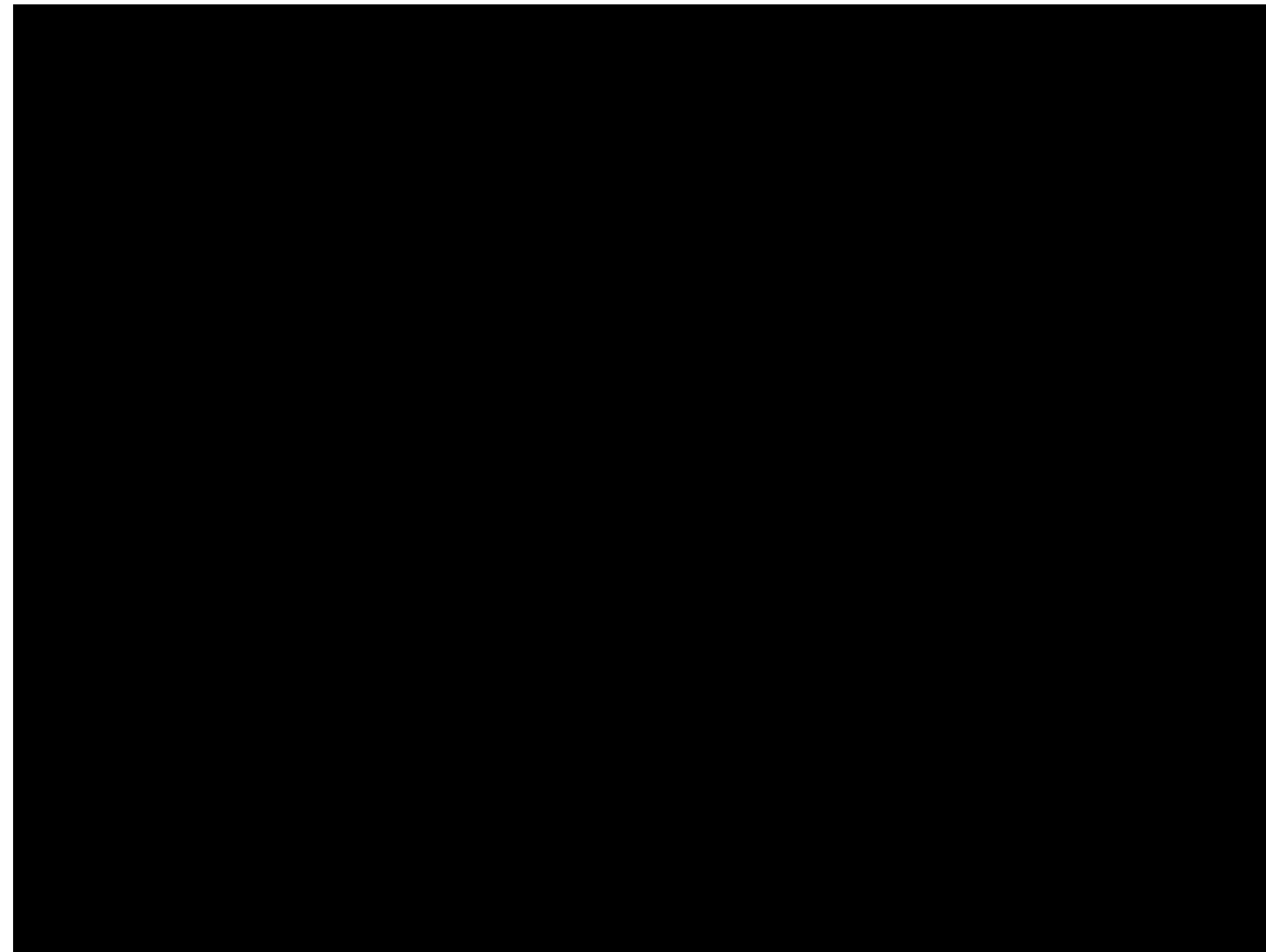
Work done between 1.26 to 1.28 (Filtering engine)

- First, a reminder - out of the box, Rucio has two distinct metadata stores:
 - A “base” metadata store that uses columns from the **dids** table; 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 the **did_meta** table
- Up until 1.27, there was limited support for metadata queries via adding a filter string to the `list-dids` and `list-dids-extended` functions:
 - Only the equality operator, and
 - Only the logical AND operator
- From 1.28 onwards, `list-dids` passes the filter string through the “filtering engine”. This engine supports an extended syntax including inequalities and wildcards, viz `=`, `!=`, `>=`, `<=`, `>`, `<`, `LIKE`, `NOT LIKE`, as well as the logical OR operator
 - Only applied to filtering on columns in the base metadata store



Work done between 1.26 to 1.28 (Filtering engine)

- From 1.28.0 onwards, `list-dids-extended` also passes the filter string through this same filtering engine
 - Works with the custom (json) metadata store



Work done between 1.26 to 1.28 (Plugins)

- Rucio uses a plugin system allowing one to write external metadata interfaces by:
 - Writing functions for how to get/set/delete metadata, and
 - Overriding the base list-dids method.
- From 1.28.0 onwards, a new metadata plugin to interface to an external mongodb collection was made available by specifying it in the rucio.cfg 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.

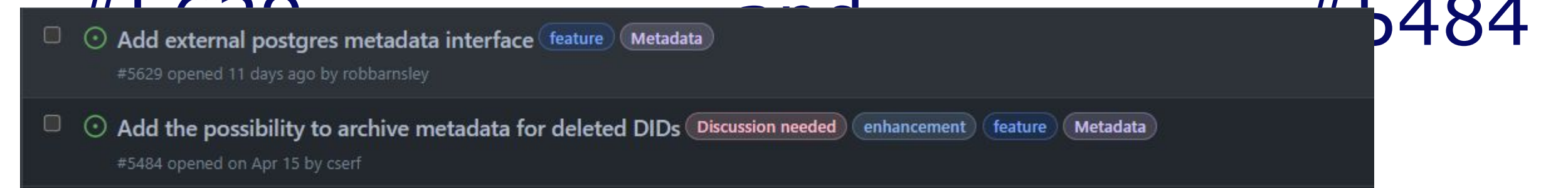
```
[metadata]
plugins = rucio.core.did_meta_plugins.postgres_meta.PostgresJSONDidMeta
postgres_service_host=postgres
postgres_service_port=5433
postgres_db=metadata
postgres_user=rucio
postgres_password=secret
postgres_db_schema=public
postgres_table=dids
```

- Can be tested in a dev environment by using the **storage-externalmetadata** compose file

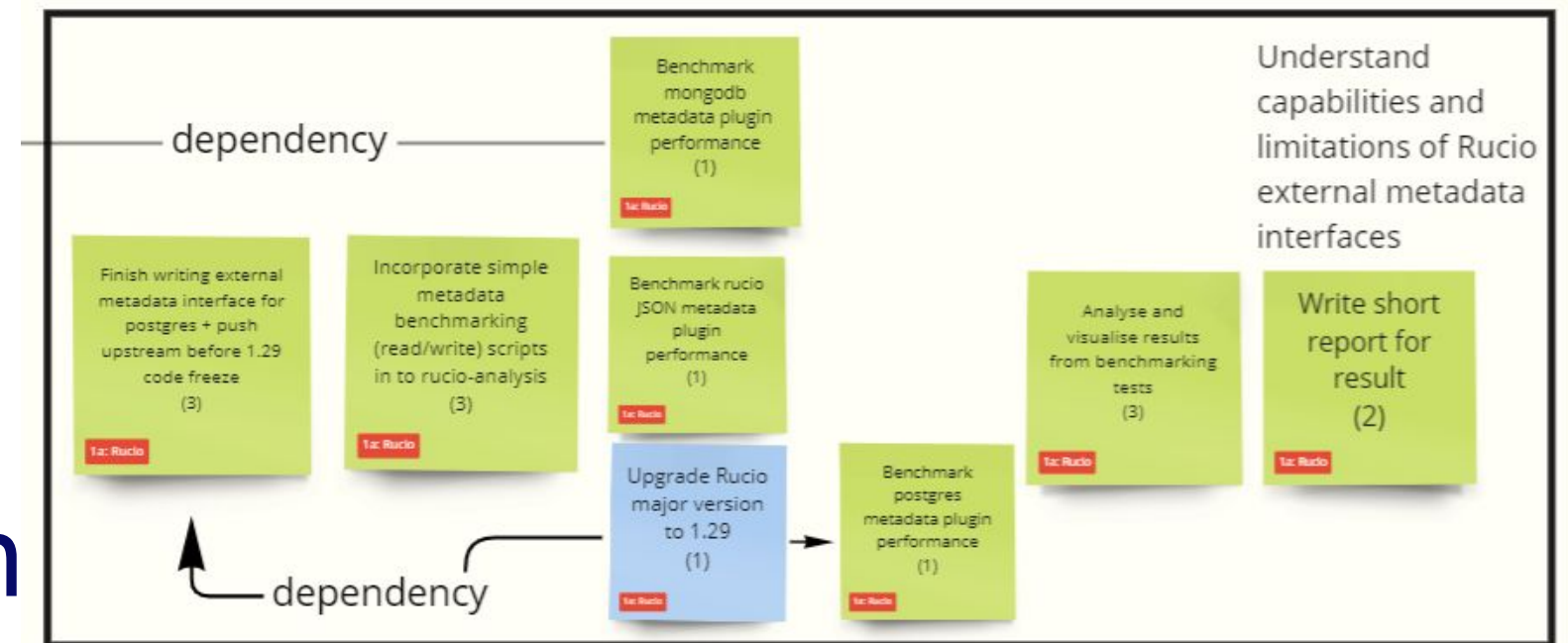


Future

- Currently only a couple of issues open (& being worked on) that directly touch metadata work:



- SKAO has plans to test e.g. how performant searching is, and how this scales with number of files and but when this is done is program roadmap. This work officially planned for this current increment, so is unlikely to happen in th



- Welcome contributions to this component
- #metadata on Rucio Slack

