Rucio Metadata Catalog for ePIC

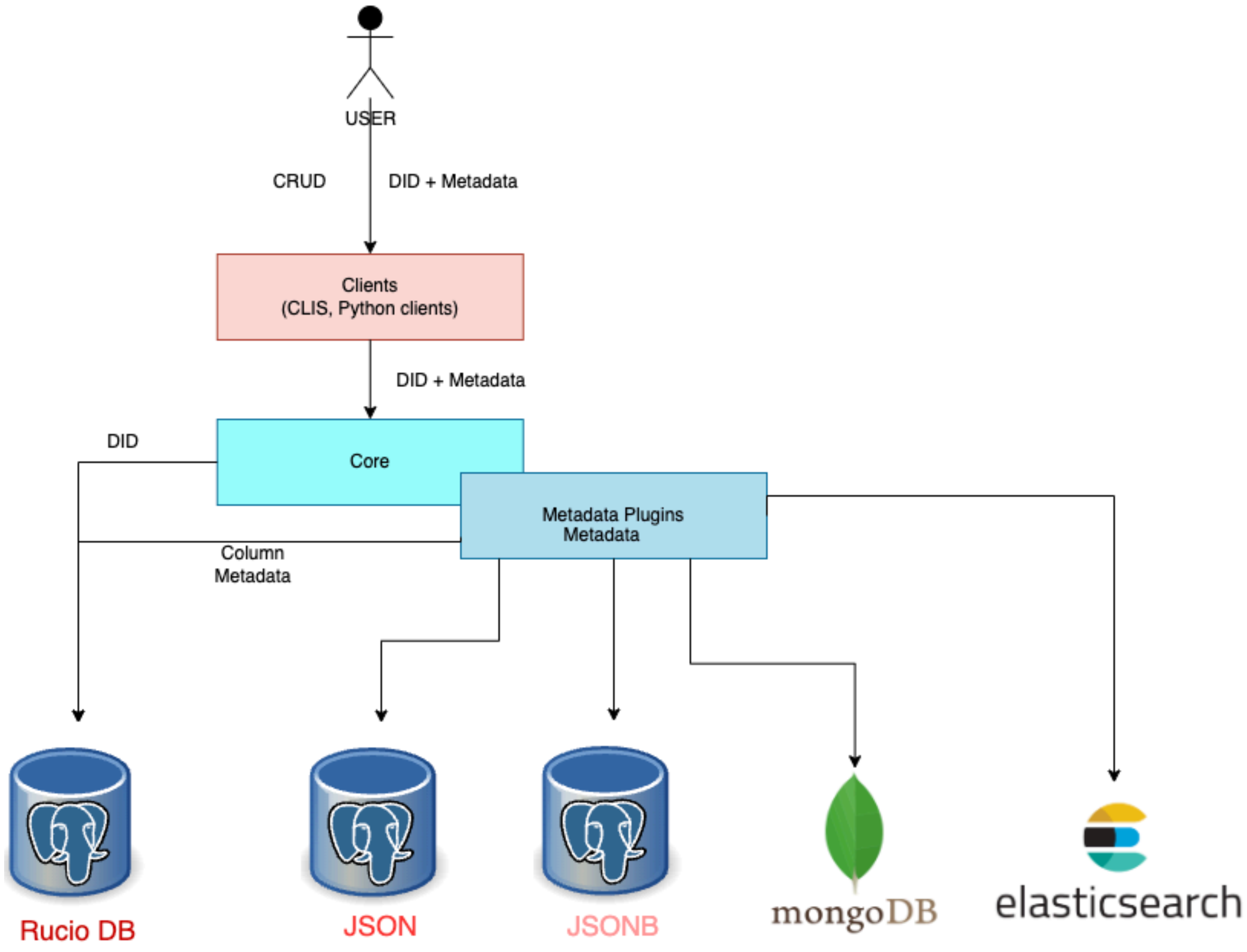
Anil Panta

2025

Jefferson Lab

Metadata in Rucio

- Rucio Provides metadata catalog attached to each DIDs.
- These metadata are provided through plugins
- Available Plugins:
  - Column Metadata
  - Postgres JSON
  - Postgres JSONB
  - MongoDB
  - ElasticSearch

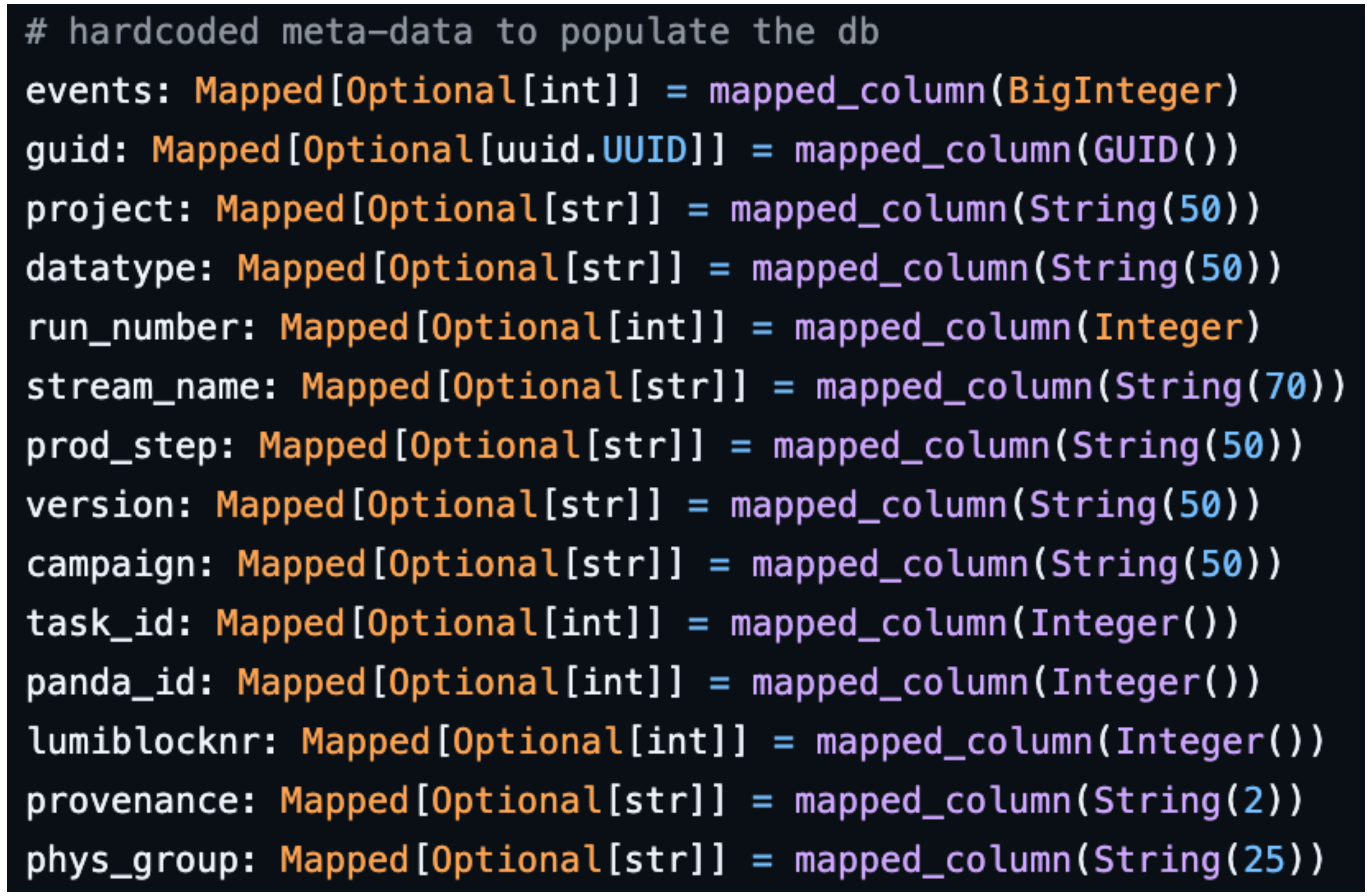


Column Metadata

- Few Columns in main DID table.
- Planned for removal from rucio

- **Pros:** None relevant (soon to be legacy only).
- **Cons:**
  - Only few columns
  - Hard-coded schema.
  - Not generalizable beyond ATLAS.

- **Assessment:**
  - ❌ Legacy, planned for removal. Can be discarded.



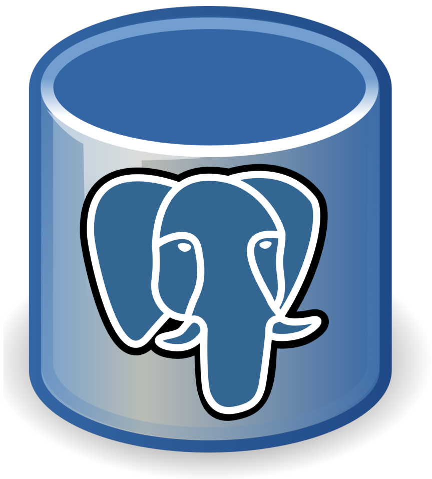
```
# hardcoded meta-data to populate the db
events: Mapped[Optional[int]] = mapped_column(BigInteger)
guid: Mapped[Optional[uuid.UUID]] = mapped_column(GUID())
project: Mapped[Optional[str]] = mapped_column(String(50))
datatype: Mapped[Optional[str]] = mapped_column(String(50))
run_number: Mapped[Optional[int]] = mapped_column(Integer)
stream_name: Mapped[Optional[str]] = mapped_column(String(70))
prod_step: Mapped[Optional[str]] = mapped_column(String(50))
version: Mapped[Optional[str]] = mapped_column(String(50))
campaign: Mapped[Optional[str]] = mapped_column(String(50))
task_id: Mapped[Optional[int]] = mapped_column(Integer())
panda_id: Mapped[Optional[int]] = mapped_column(Integer())
lumiblocknr: Mapped[Optional[int]] = mapped_column(Integer())
provenance: Mapped[Optional[str]] = mapped_column(String(2))
phys_group: Mapped[Optional[str]] = mapped_column(String(25))
```

Postgres JSON

- Metadata stored as raw JSON -> Text
- **Pros**:
  - Flexible, schema-less.
  - Preserves formatting and order (good if exact JSON input matters).
  - Easy to store “as-is” metadata payloads.

- **Cons:**
  - No indexing support → full table scan for queries.
  - Very inefficient for large datasets.
  - Only good for storing/displaying, not searching.

- **Assessment:** ⚠️ Useful for minimal cases, not scalable for search.



JSON

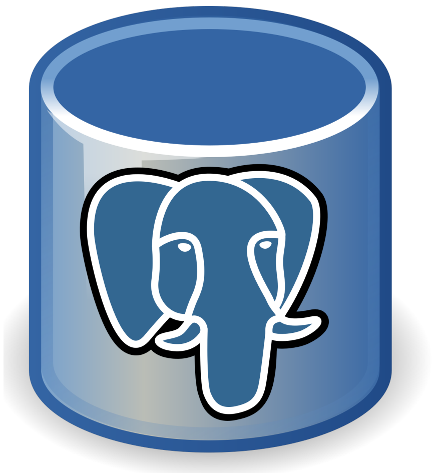
Postgres JSONB

- Binary JSON storage with indexing support.
- Pros:
  - Optimized for retrieval and querying.
  - Supports GIN/BTREE indexes → efficient metadata search.
  - Already familiar tech (Postgres is Rucio’s main DB).
- **Cons:**
  - Slightly higher write cost (binary conversion).
  - Indexing strategy must be explicitly designed and maintained.
- **Assessment:**
  - ✅ Strong balance of flexibility + search performance.

JSONB

Postgres JSONB

- Binary JSON storage with indexing support.
- Pros:
  - Optimized for retrieval and querying.
  - Supports GIN/BTREE indexes → efficient metadata search.
  - Already familiar tech (Postgres is in Rucio’s main DB).
- **Cons:**
  - Slightly higher write cost (binary conversion).
  - Indexing strategy must be explicitly designed and maintained.
- **Assessment:**
  - ✅ Strong balance of flexibility + search performance.
  - DaFab project is invested in this. Talk: DaFab_D.Xenakis.pdf



JSONB

MongoDB

- Document database backend for schema-less JSON-like data.
- Pros:
  - Natively supports flexible, dynamic documents.
  - Powerful querying & indexing (nested keys, arrays, ranges).
  - Horizontal scaling, proven for metadata-heavy workloads.
- **Cons:**
  - Requires separate MongoDB infrastructure.
  - Operational expertise needed.
- **Assessment:**
  - ✅ Production-ready for communities with complex, evolving metadata models.

ElasticSearch

- Search engine backend for structured + full-text metadata queries. Very new in Rucio.
- Pros:
    - Excellent for full-text and complex search.
    - Scales horizontally for very large metadata volumes.
    - Flexible JSON storage + powerful indexing.
- **Cons:**
    - Experimental in Rucio (no production deployments yet).
    - Separate infra + expertise required.
    - Eventual consistency (not strictly transactional).
- **Assessment:**
    - 🚀 Promising for large-scale metadata search & analytics.
    - Needs production adoption experience before wide rollout.

elasticsearch

Needs and Testing : Postgres JSONB

- **Needs**:
  - Json Schema Validation. Simple schema validation feature PR made: PR:8096

- **Testing**:
  - Feature PR madeDelete metadata bug. Bug Fix in PR: PR:8096
  - Smoke test okay including above 2 PRs.
  - Trying out GIN indexing.
    - Some performance testing ongoing.
    - Tracking the performance and optimization in : Issues:8101