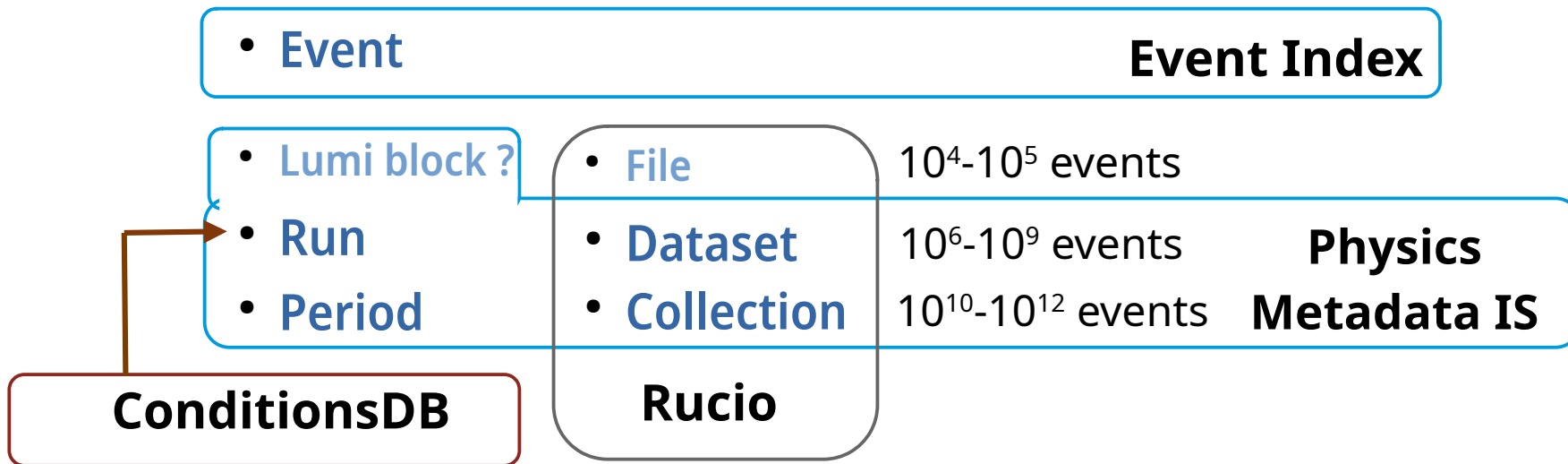


SPD Metadata

Current status

- Metadata is “data about data”
- Metadata can be classified by levels of granularity



- Physicists deal with datasets, runs and events
- Metadata IS provide tools to look information about them

- **Contain information about**
 - Datasets and data samples,
 - Provenance chains of processed data with links to production task configurations,
 - Online filter and luminosity information for runs
 - Cross-sections and configurations used for simulations
 - Software versions used to process the data
- **Should collect a great part of its information from other IS's and provide links to data there**
- **As MC data generation started, development of this IS is actual**
- **Conventions for the runs and dataset naming have to be defined, as well as for software and MC configuration versioning**

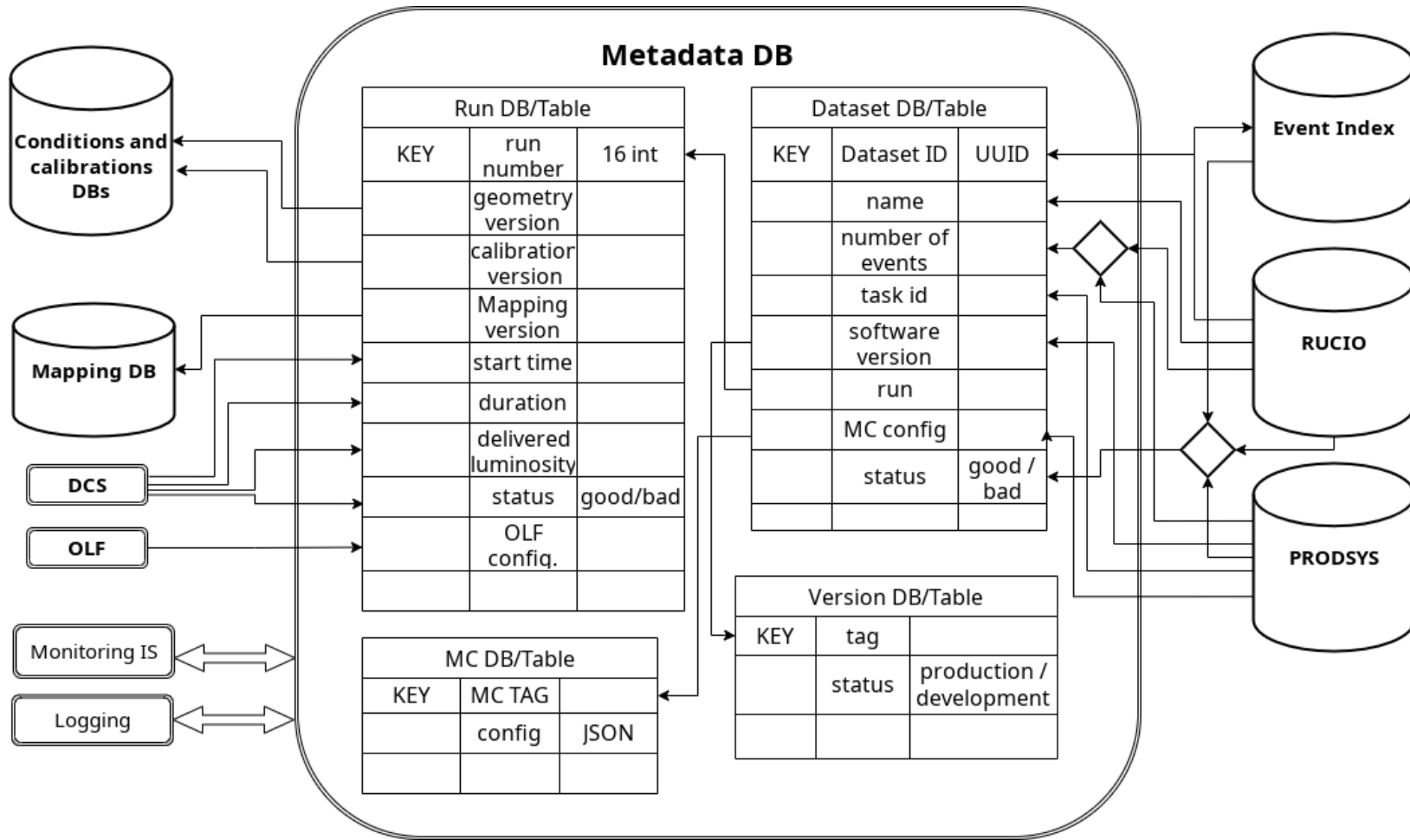
- **ATLAS Metadata Interface (AMI) is a generic ecosystem for metadata bookkeeping system used by the ATLAS experiment**
- **AMI is mainly used by the collaboration for three purposes**
 - to catalogue the datasets available for analysis
 - the Dataset discovery tool
 - for the management of the various releases of the ATLAS software
 - the Tag Collector
 - to record the processing of all the official ATLAS datasets
 - the AMI-Tags interface
- **The framework comes with a set of tools that can be used to quickly design metadata-oriented applications**
- **Provides facilities for aggregating and searching data with easy-to-use web interfaces and lightweight clients.**

- **The server, written in JAVA, provides**
 - a single HTTP endpoint for all clients (web applications, APIs),
 - a command interpreter with a highly configurable authorization system,
 - a single sign-on for applications supporting all standard authentication methods,
 - a distributed transaction engine,
 - connection pooling for data sources, a
 - Metadata Query Language (MQL) to SQL generator
 - a task server, a component equivalent to a distributed cron that can perform metadata aggregation on heterogeneous datasources
- **The web framework (JavaScript)**
 - provides an application and sub-application Model-View-Controller (MVC) pattern
 - Has a customizable search engine modeler and advanced data displaying tools
- **AMI Clients:**
 - The main client provided by the AMI team is a Python client called pyAMI

- The “command” layer is the standard AMI entry point.
 - Querying AMI consists in sending an AMI command to AMI Java Core
 - “GetDatasetInfo” (for getting the metadata associated with a given dataset)
 - “ListDataset” (for listing datasets by metadata criteria)
- The “metadata” layer provides business logic which hides the details of the underlying data sources and optimizes performance
- Metadata Query Language (MQL) is a Domain-Specific Language (DSL) for querying RDMS
 - Permits performing queries without (precisely) knowing relations between tables
 - MQL only deals with metadata entity names while SQL uses a catalog / table / field paradigm
 - MQL queries are internally converted to optimize SQL queries

- Using AMI one can search for datasets belonging to the specific run or range of runs, specifying version, data format and other criteria.
- Detailed Information on every dataset can be retrieved, including number of events, production task, configuration, etc...
 - Some information are being linked from other databases
 - AMI does not know about files or dataset location. Use Rucio!
 - Dataset provenance can be tracked
- Users can look through software versions or “tags”
 - Information like base release, deployment date, stsus

- Information from AMI can be used by the users, analysis and production tasks and other information systems
- **Event Index:**
 - Looking for new datasets to index
 - Deciding if the dataset is worth indexing:
 - Status in AMI and production system (should be valid and complete)
 - Tag and software release (exclude problematic)
 - Event number (exclude empty)
 - Date (should not be too old)
 - Provenance (exclude coming from dataset of unwanted kind)
 - etc...
 - A special table has been created in AMI for use with Event Index



- May have similar components as AMI, and something else
 - DB back-end (PostgreSQL)
 - Application layer for accessing data by users and applications
 - API for data exchange
 - RESTful
 - Python, C++ when needed
 - Caching proxy
 - WEB GUI and command line interfaces
 - Daemon for collecting data from other information systems
 - Validation daemon, that checks data consistency between systems
 - **Data WILL be inconsistent. This is USUAL and NORMAL way of operation.**
 - Components on demand

BACKUP

- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Nisi tincidunt eget nullam non.
- Quis hendrerit dolor magna eget est lorem ipsum dolor sit:
 - Volutpat odio facilisis mauris sit amet massa.
 - Commodo odio aenean sed adipiscing diam donec adipiscing tristique
 - Mi eget mauris pharetra et. Non tellus orci ac auctor augue. Elit at imperdiet dui accumsan sit
- Ornare arcu dui vivamus arcu felis. Egestas integer eget aliquet nibh praesent. In hac habitasse platea dictumst quisque sagittis purus. Pulvinar elementum integer enim neque volutpat a