



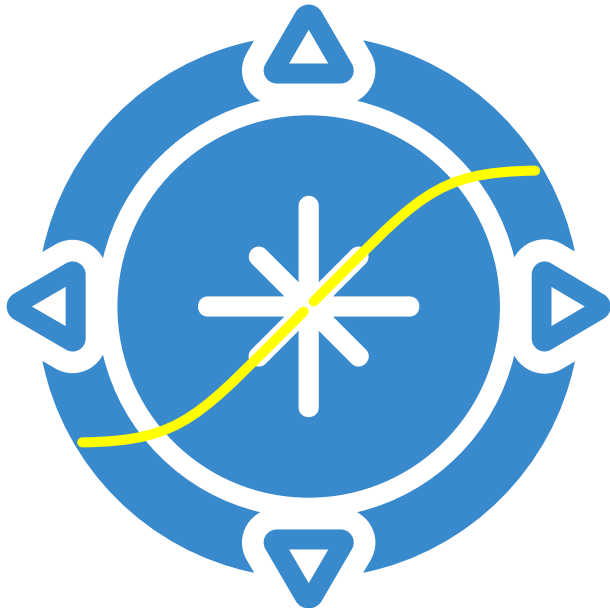
ami

Ecosystem

P.-A. Delsart, J. Fulachier, F. Lambert, J. Odier

Presentation outline

ATLAS Metadata Interface (AMI)
A generic ecosystem for scientific metadata



1

What is AMI?

2

AMI backend

3

AMI frontend

4

Metadata aggregation

5

AMI live demo

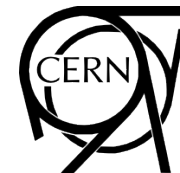
01

What is AMI ?



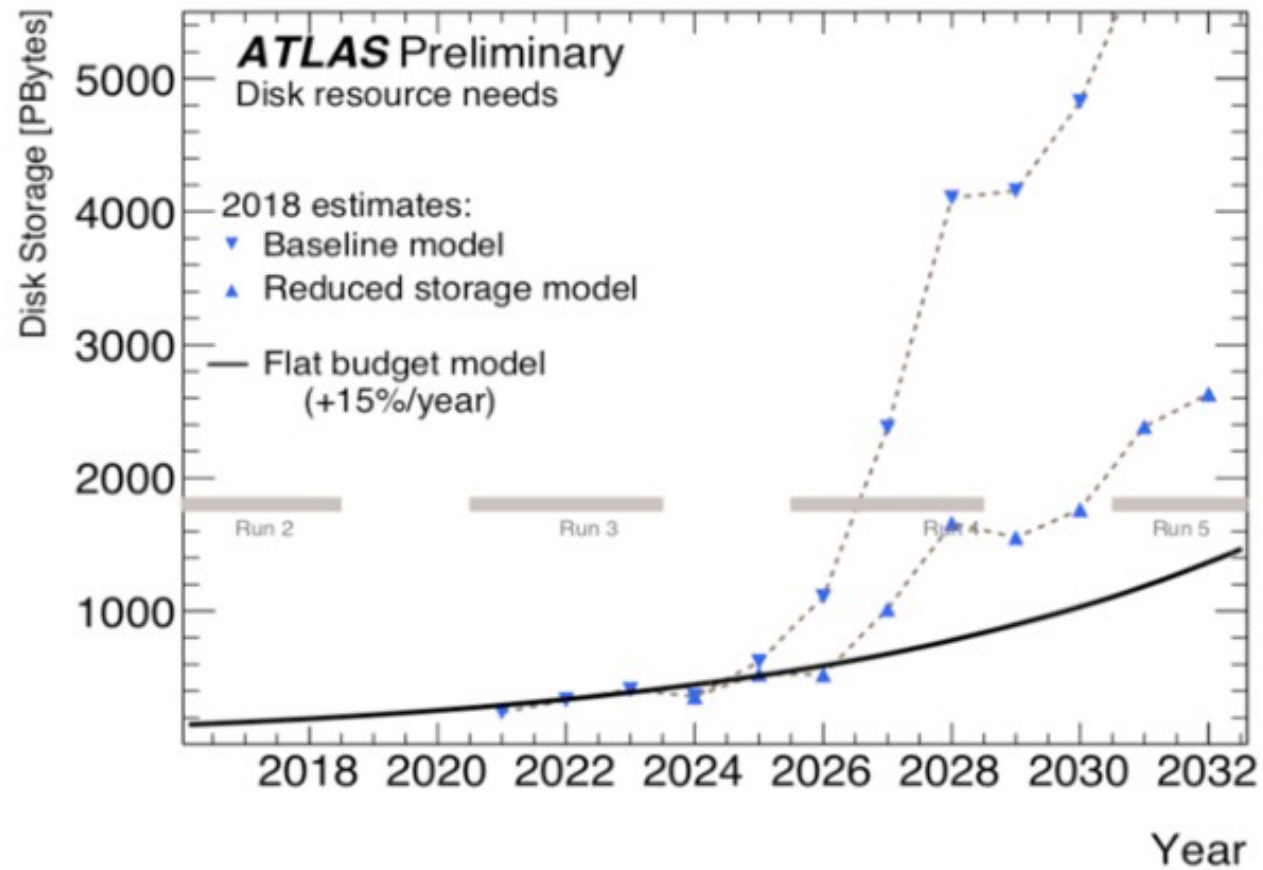
What is AMI?

- **AMI (ATLAS Metadata Interface) is a generic ecosystem dedicated to scientific metadata:**
 - Heterogenous datasource / Web service connectivity
 - AMI can connect to any existing DB or Web service
 - High level primitives for metadata extraction and processing
 - High level tools for performing data selection by metadata criteria
- **The ecosystem has development kits for:**
 - Developing server-side metadata-oriented subsystems (in JAVA)
 - Developing client-side metadata-oriented Web applications (in JS) and scripts (in Python, JS, ...)
- **AMI is designed to work in big-data context:**
 - Scalability, evolutivity and maintainability

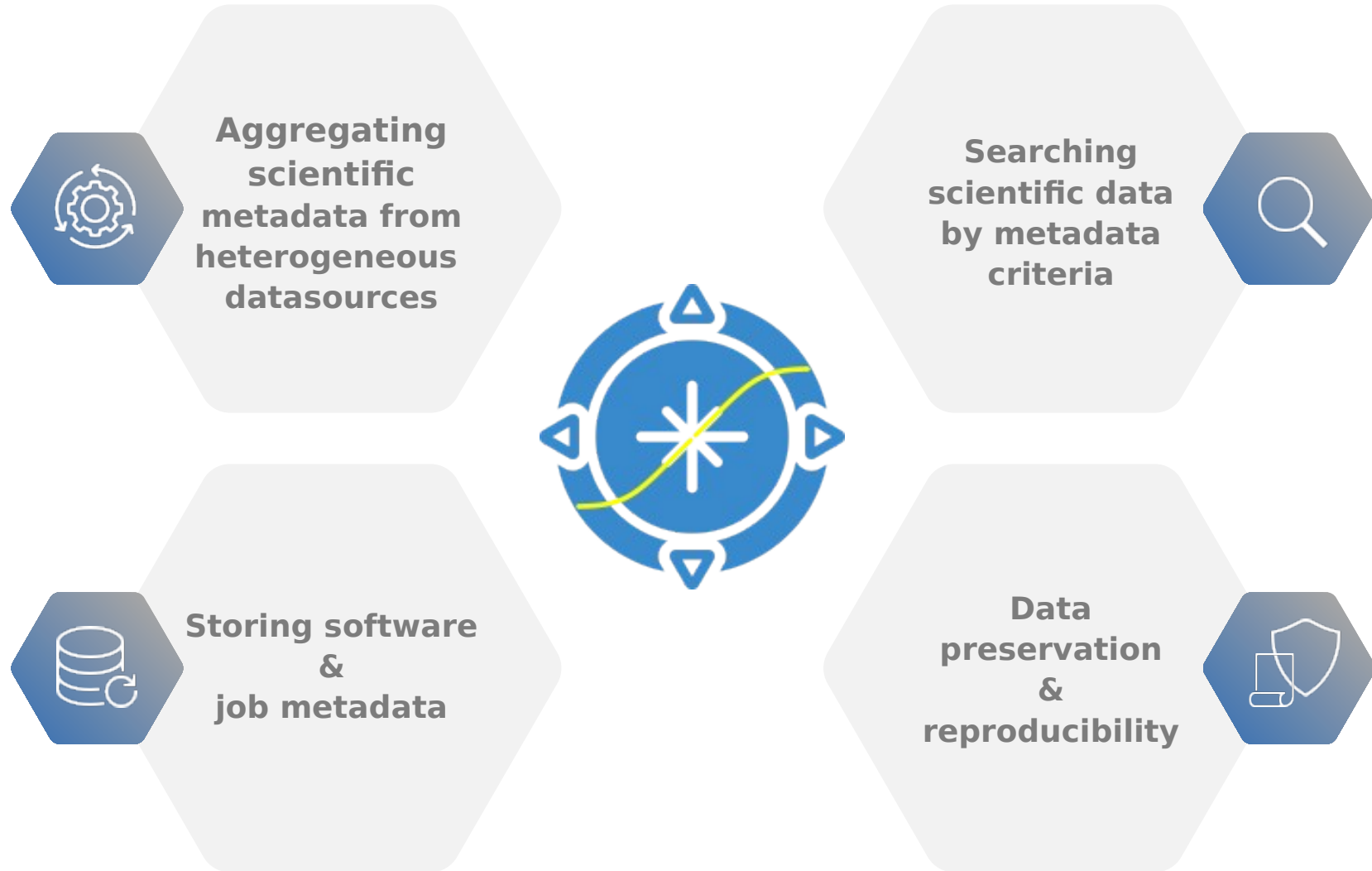


What is AMI?

- > 22 years of experience within the ATLAS collaboration at CERN
- AMI hosts metadata for $O(10^6)$ datasets and $O(10^9)$ files



Typical usage

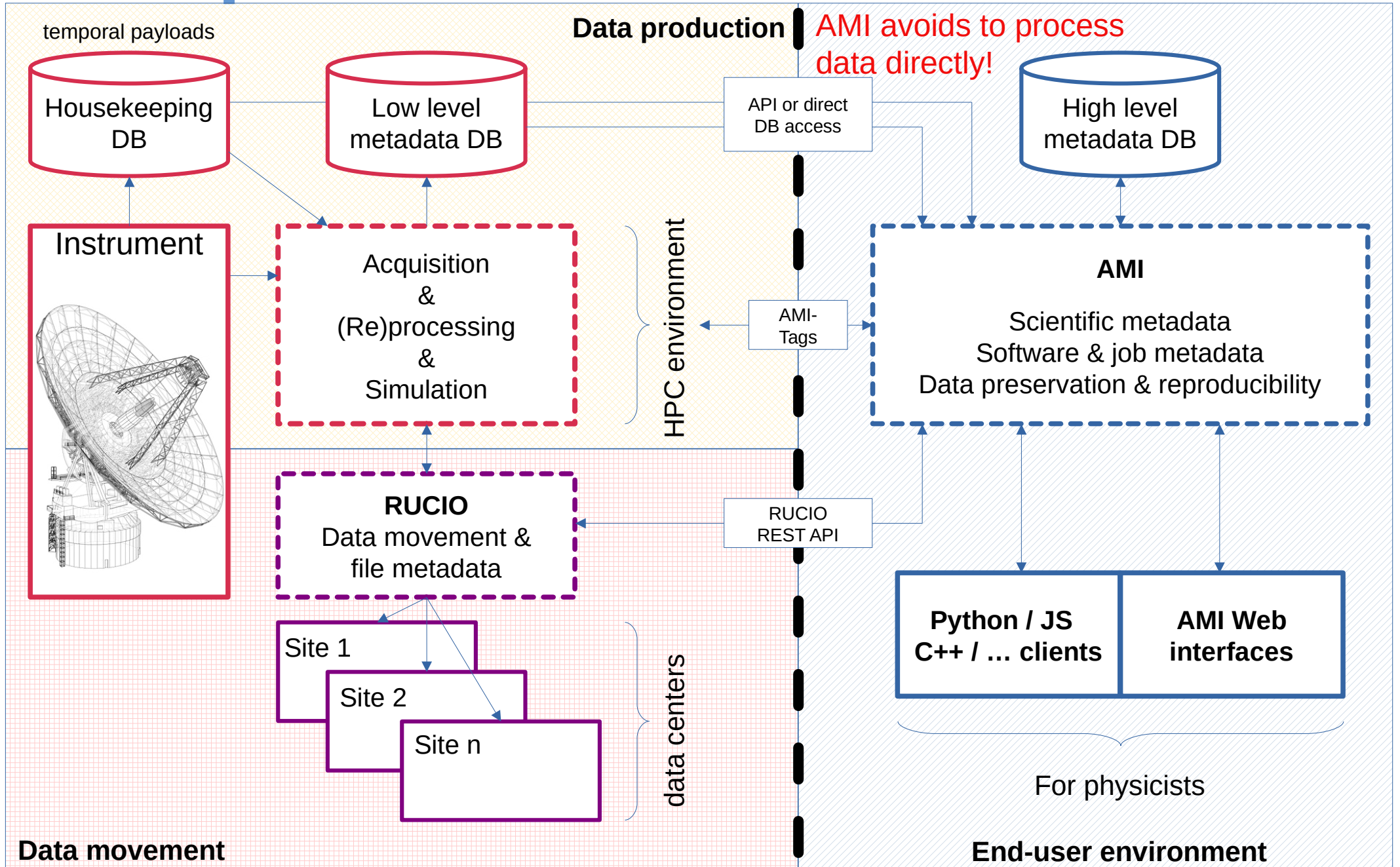


Typical usage

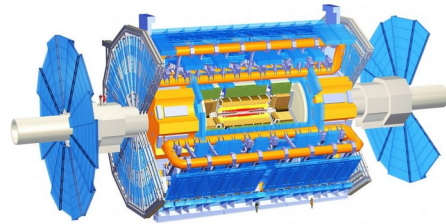
- **Aggregating scientific metadata from heterogeneous datasources:**
 - Condition / housekeeping metadata
 - Dataset and file metadata (from production and data movement systems)
 - End-user metadata (physics parameters, annotations, comments, ...)
- **Searching scientific data by metadata criteria:**
 - Via Web interfaces and Python / JS / C++ / Java / ... scripts
 - 👍 This is the most important feature for end-users / physicists
- **Defining metadata for softwares (version, parameters), grid jobs, etc...**
 - In ATLAS, each grid job is defined from an AMI-Tag (= set of software parameters)
 - Each dataset is associated to one or many AMI-Tags
- **Associating datasets and papers:**
 - Data preservation
 - Reproducibility

Example of workflow

What is AMI?



AMI Ecosystem



Detector

Initial data and metadata



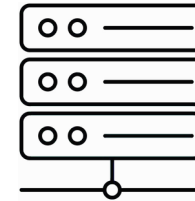
AMI TaskServer

Metadata aggregation, transformation and storing



AMI Backend

Web services & high level metadata features

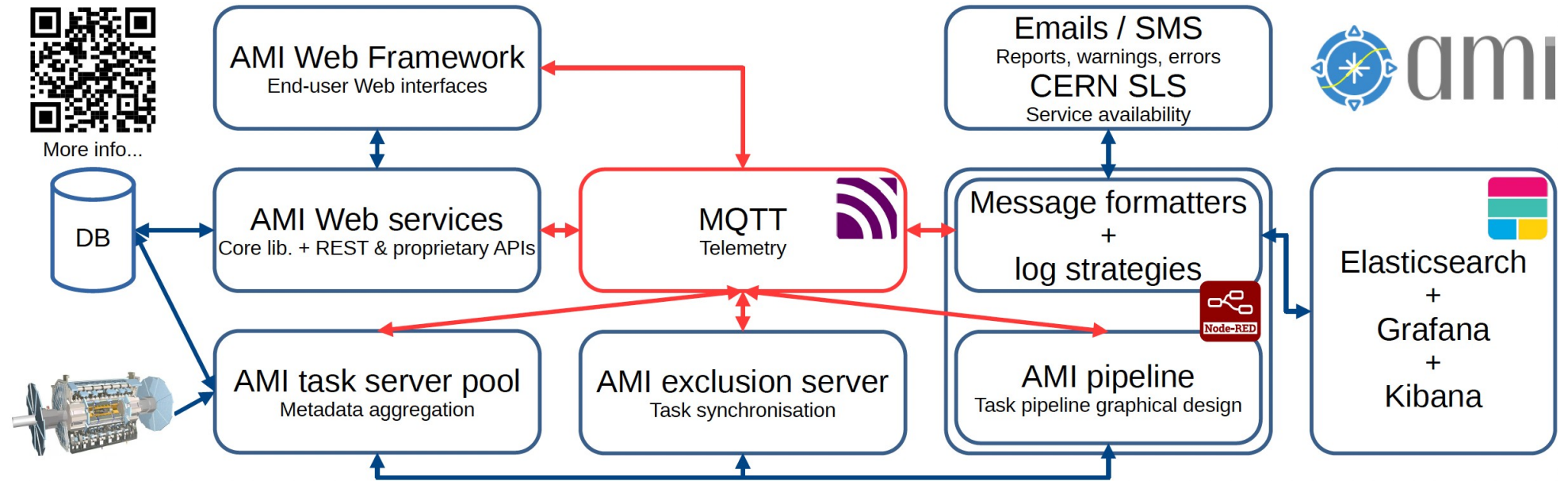


AMI Frontend

Web interfaces & REST, Python, JS, Java, C++, ... clients

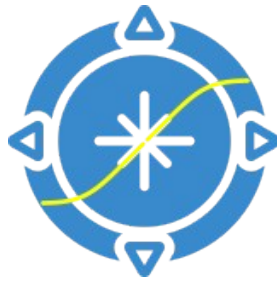


AMI Ecosystem - whole stack



What is AMI?

- **Images on DockerHub and DockerCompose demo**
 - Easy deployment in a Docker Compose or Kubernetes environment
 - CHEP 2023 paper: <https://cds.cern.ch/record/2868009/files/ATL-SOFT-PROC-2023-009.pdf>
- **Each sub-system of the AMI ecosystem can connect to an optional MQTT broker for:**
 - Monitoring purpose (cpu usage, ram usage, disk usage, ...)
 - Control purpose (reload, restart, ...)



AMI can immediately connect to any existing database or web service

AMI meets the needs of both small and large scientific experiments



02

AMI Backend

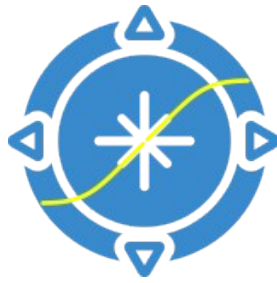


AMI Backend



- **AMI Backend**
 - Based on the AMI Java Core library
 - Control and monitoring capabilities via MQTT
 - Scalable Web service (REST and proprietary APIs) with authentications / authorizations
 - Heterogeneous datasource connectivity (as soon as a Java JDBC driver exists)
 - Command engine (= the way of talking with AMI)
 - Metadata queries (generic or more specific queries),
experiment-specific commands, service administration, ...
 - **Metadata Query Language (= MQL, see next slide)**
 - High-level primitives for data & metadata handling
- **See CHEP 2019 paper:**
 - https://www.epj-conferences.org/articles/epjconf/pdf/2019/19/epjconf_chep2018_05046.pdf

AMI Backend - MQL



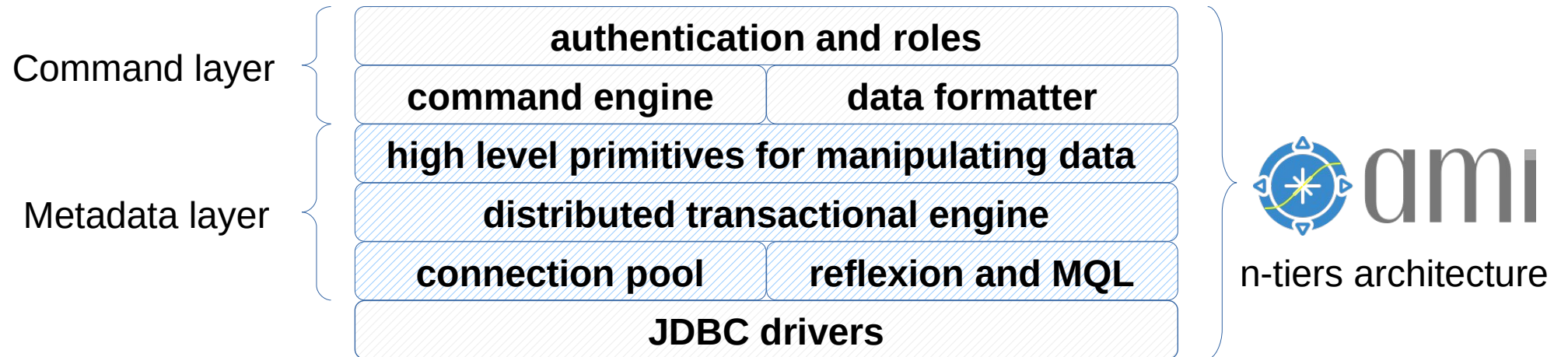
- **Metadata Query Language (MQL)**
 - Kind of SQL without FROM clause nor join
 - It makes it possible to build queries without knowing table relations
 - Joins are automatically generated from the AMI reflexion sub-system
 - MQL turns the database-oriented perspective into a metadata-oriented perspective.
- **See CHEP 2019 paper:**
 - https://.../epjconf_chep2020_04044.pdf

```
SELECT
  `Track`.*
WHERE
  [ `Playlist`.`Name` = '90's Music' ]
AND
  [ `Playlist`.`Name` = 'Heavy Metal Classic' ]
```

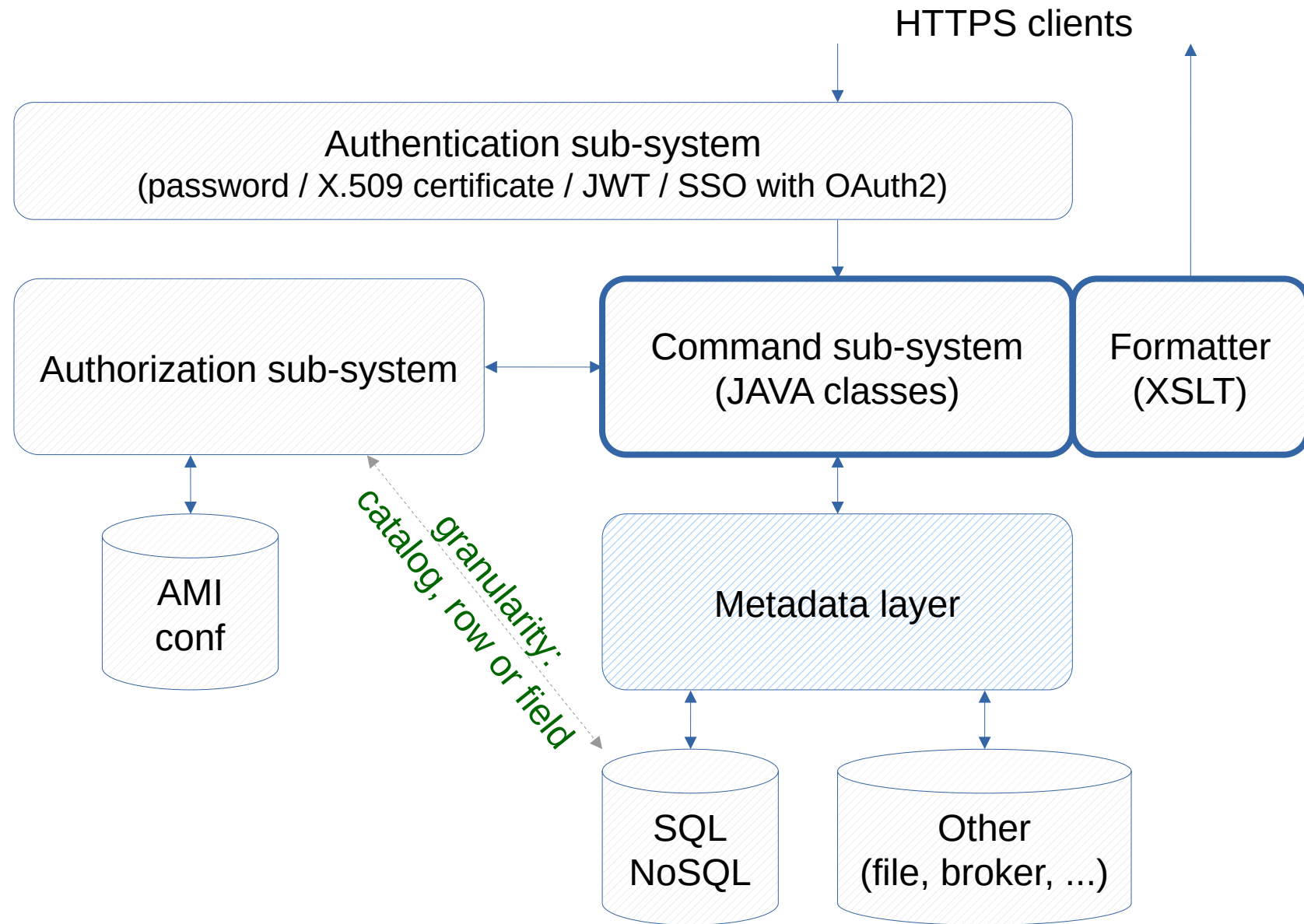
MQL to
SQL

```
SELECT
*
FROM
  `demo`.`Track`
WHERE
  (
    `demo`.`Track`.`TrackId` IN (
      SELECT
        `demo`.`Track`.`TrackId`
      FROM
        `demo`.`Playlist`,
        `demo`.`PlaylistTrack`,
        `demo`.`Track`
      WHERE
        (
          `demo`.`Playlist`.`Name` = '90's Music'
        )
        AND `demo`.`PlaylistTrack`.`TrackId` = `demo`.`Track`.`TrackId`
        AND `demo`.`PlaylistTrack`.`PlaylistId` = `demo`.`Playlist`.`PlaylistId`
      )
    )
  AND (
    `demo`.`Track`.`TrackId` IN (
      SELECT
        `demo`.`Track`.`TrackId`
      FROM
        `demo`.`Playlist`,
        `demo`.`PlaylistTrack`,
        `demo`.`Track`
      WHERE
        (
          `demo`.`Playlist`.`Name` = 'Heavy Metal Classic'
        )
        AND `demo`.`PlaylistTrack`.`TrackId` = `demo`.`Track`.`TrackId`
        AND `demo`.`PlaylistTrack`.`PlaylistId` = `demo`.`Playlist`.`PlaylistId`
      )
    )
  )
```

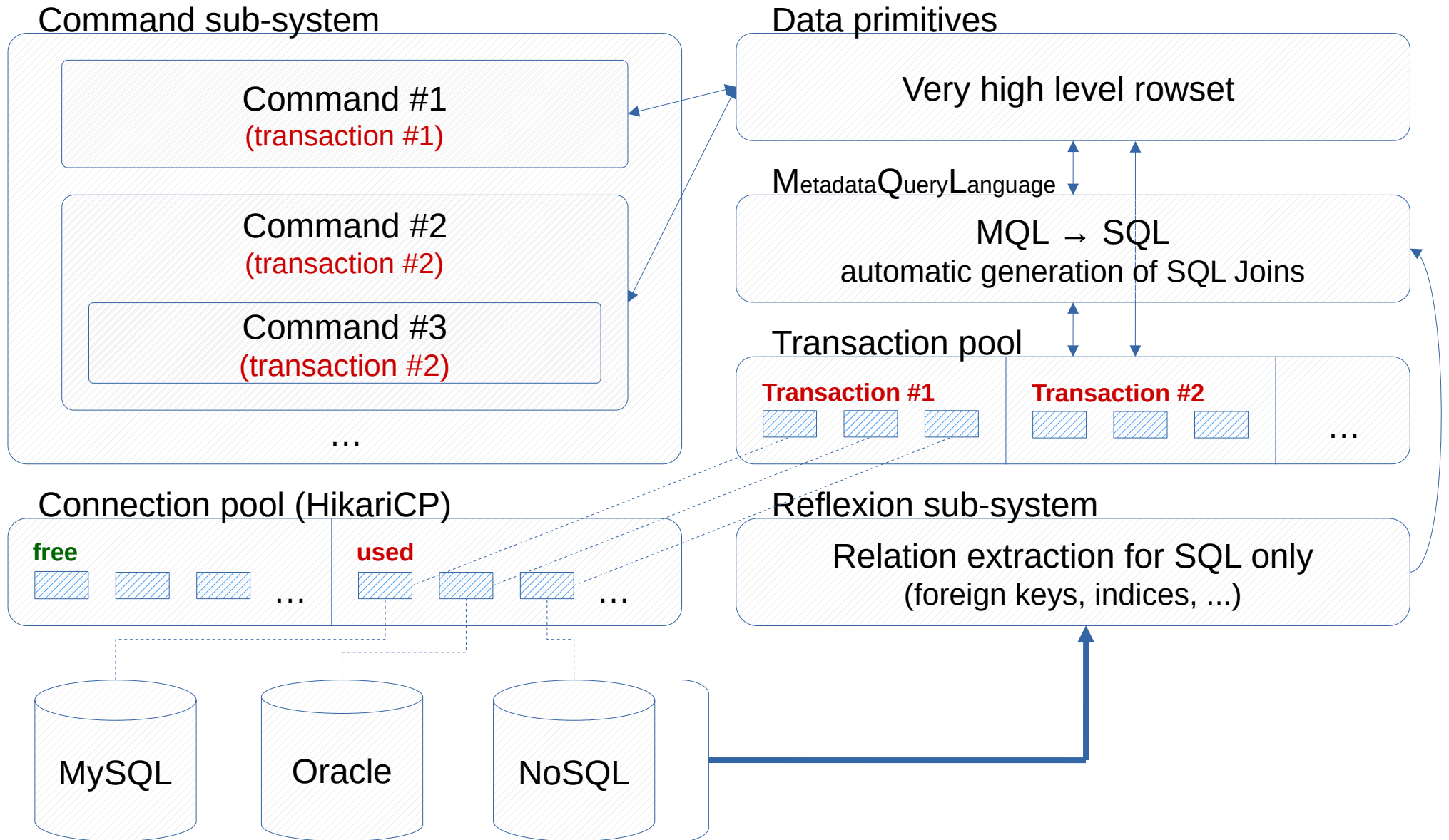
AMI Backend - overview



AMI Backend - command layer



AMI Backend - metadata layer

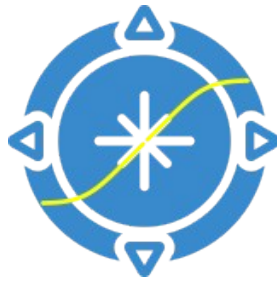


03

AMI Frontend

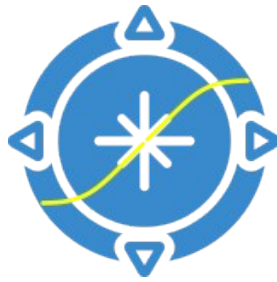


AMI Web Framework



- **AMI Frontend** (aka AMI Web Framework (AWF))
 - Based on modern technologies (Webpack, Bootstrap, TWIG)
 - **Optimized for developing metadata-oriented JS Web applications**
 - AMI provides a set of ready-to-use applications and controls (see next slide)
 - **AMI controls are embeddable in external websites such as WIKIs, confluence, ...**
 - Patterns MVC (with AMI-Twig) or MVVM (with Vue.js 3)
- **See CHEP 2019 paper:**
 - https://www.epj-conferences.org/articles/epjconf/pdf/2019/19/epjconf_chep2018_04004.pdf

AMI Web Framework



- **Applications are generally built by assembling controls**
- **Main available controls:**
 - Dialog boxes
 - Controls for annotating entities
 - Controls for searching (Simple Search, Criteria Search, ...)
 - Controls for displaying (Schema Viewer, Tab, Table, Element Info, ...)
- **Main available applications:**
 - Embedded CMS
 - AMI command interpreter
 - Admin Dashboard and Monitoring
 - Schema Viewer, Table Viewer, Simple Search, Criteria Search, Search Modeler, ...

Designing search interfaces (admin)

The screenshot displays the AMI Search Modeler admin interface. At the top, there is a navigation bar with the AMI logo and various menu items: Datasets, Files, SW Images, AMI-Tags, Nomenclature, Tools, Issue reporting, and search icons. On the right side of the navigation bar, there is a 'CC-IN2P3 website' badge, a star icon, a user profile 'jodier', and a 'Sign Out' button.

The main content area is titled 'Search Modeler' and is divided into two panels. The left panel, 'Search interfaces', lists various search interfaces with a 'goto' link for each. The right panel, 'Search interface modeler', allows for configuring a search interface. It includes fields for 'Group*' (Real data), 'Name*' (data21), 'Catalog*' (data21_001:real_data), and 'Entity*' (DATASET). There are also controls for 'Archived' (a toggle switch set to 'no / yes') and 'Options' (a dropdown menu). Below these fields is a table of search criteria.

Criteria (alias, catalog*, entity*, field*, type*)	Group	Entity	Field	Type	Options
AMI status	data21_001:real_data	DATASET	AMISTATUS	boolean	[Flag] [Trash]
Project	data21_001:real_data	DATASET	PROJECTNAME	text (few results)	[Flag] [Trash]
Run number	data21_001:real_data	DATASET	RUNNUMBER	text (many results)	[Flag] [Trash]
Stream	data21_001:real_data	DATASET	STREAMNAME	text (few results)	[Flag] [Trash]
Prod. Step	data21_001:real_data	DATASET	PRODSTEP	text (few results)	[Flag] [Trash]
Data type	data21_001:real_data	DATASET	DATATYPE	text (few results)	[Flag] [Trash]
AMI-Tag	data21_001:real_data	DATASET	VERSION	text (many results)	[Flag] [Trash]
Dataset name	data21_001:real_data	DATASET	LOGICALDATASETNAME	text (many results)	[Flag] [Trash]
Campaign	data21_001:real_data	CAMPAIGN	CAMPAIGNNAME	text (few results)	[Flag] [Trash]
Period	data21_001:real_data	DATASET	PERIOD	text (few results)	[Flag] [Trash]
Geometry	data21_001:real_data	DATASET	GEOMETRYVERSION	text (many results)	[Flag] [Trash]
Status	data21_001:real_data	DATASET	PRODSYSSTATUS	text (few results)	[Flag] [Trash]
ECM energy	data21_001:real_data	DATASET	ECMENERGY	number	[Flag] [Trash]
Superdataset	data21_001:real_data	SUPERDATASET	SUPERDATASETNAME	text (many results)	[Flag] [Trash]

Searching data by criteria

The screenshot shows the AMI Frontend search interface. At the top, there is a navigation bar with the AMI logo and various menu items: Datasets, Files, SW Images, AMI-Tags, Nomenclature, Tools, Issue reporting, and a search icon. On the right side of the navigation bar, there is a 'CC-IN2P3 website' link, a star icon, a user profile 'jodier', and a 'Sign Out' button.

Below the navigation bar, the page title is 'Metadata / Search'. There are three main categories of data: 'Real data' (with sub-items: physics container, data23, data22, data21, data20, data19, data18, data17, data16, data15, data14, data13, data12, data11, data10, data09), 'Simulated data' (with sub-items: mc23, mc21, mc20, mc16, mc15, mc14, mc11, mc12, mc10, mc09), and 'Validation data' (with sub-item: valid).

The search results are filtered to 'data21'. A search bar contains 'data21'. Below the search bar, there is a 'View Selection' dropdown and a status indicator: 'Number of selected items (DATASET): 10'. The search criteria are displayed as a logical expression: '(((Q1 and Q2) and Q3) and Q4) and Q5'.

The search criteria are defined by five filters (Q1-Q5):

- Q1: AMI status** (toggle on): ALL / VALID
- Q2: Data type** (toggle on): AOD
- Q3: AMI-Tag** (toggle on): f1100_m2066, f1110_m2066, f1111_m2066, f1124_m2066, f1135_m2066
- Q4: Stream** (toggle on): express_express, physics_Main
- Q5: Prod. Step** (toggle on): merge

Each filter has a 'Filter, % for wildcarding' input and an 'Apply' button. The filter counts are: Q1 (#1), Q2 (#1), Q3 (#10, limit: 10 -/+), Q4 (#2), and Q5 (#1).

On the left side of the search results, there is a vertical list of filterable attributes: AMI status, Project, Run number, Stream, Prod. Step, Data type, AMI-Tag, Dataset name, Campaign, Period, Geometry, Status, ECM energy, and Superdataset.

Search results

The screenshot shows the AMI Frontend interface. At the top, there is a navigation bar with the AMI logo and various menu items: Datasets, Files, SW Images, AMI-Tags, Nomenclature, Tools, Issue reporting, and a search icon. On the right, there are links for 'CERN website', a user profile for 'admin', and a 'Sign Out' button. Below the navigation bar, the page title is 'Admin / Admin Dashboard'. The main content area has tabs for 'Catalogs', 'Entities', 'Fields', and 'Foreign keys'. A search filter is applied: 'Select router_catalog by router_catalog.externalCatalog'. To the right, there is a '+ Add new router_catalog' button. Below the search area, there is a table with 10 columns: details, Id, externalCatalog, jdbcUrl, user, pass, json, and ar. The table contains 10 rows of data, each representing a catalog entry with its ID, name, JDBC URL, user, password, and JSON metadata.

details	Id	externalCatalog	jdbcUrl	user	pass	json	ar
	1	self Show/Edit catalog	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=...	ATLAS_AMI_ROUTER_W		{"router_authority":{"x":250,"y":370,"color":"#1494CC"},"router_...	
	23	dataSuper_001:real_data Show/Edit catalog	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=...	ATLAS_AMI_DATASUPER_01_W		{"contained_dataset":{"x":625,"y":500,"color":"#0066CC"},"datase...	
	25	data10_001:real_data Show/Edit catalog	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=...	ATLAS_AMI_DATA10_01_W		{"DATASET":{"x":590,"y":225,"color":"#0066CC"},"DATASET_COMMENT"...	
	26	mc10:production Show/Edit catalog	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=...	ATLAS_AMI_MC10_01_W		{"DATASET":{"x":650,"y":245,"color":"#0066CC"},"DATASET_COMMENT"...	
	717352	tasks Show/Edit catalog	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=...	ATLAS_AMI_TASKS_W		{"router_task":{"x":355,"y":60,"color":"#0066CC"},"router_task_s...	
	28	COMA:external Show/Edit catalog	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=...	ATLAS_TAGS_METADATA_AMI_W		{"coma_cb_gtags":{"x":605,"y":580,"color":"#0066CC"},"coma_cool...	
	29	data11_001:real_data Show/Edit catalog	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=...	ATLAS_AMI_DATA11_01_W		{"DATASET":{"x":590,"y":225,"color":"#0066CC"},"DATASET_COMMENT"...	
	30	mc11_001:production Show/Edit catalog	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=...	ATLAS_AMI_MC11_01_W		{"DATASET":{"x":650,"y":245,"color":"#0066CC"},"DATASET_COMMENT"...	
	31	mc12_001:production Show/Edit catalog	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=...	ATLAS_AMI_MC12_01_W		{"DATASET":{"x":650,"y":245,"color":"#0066CC"},"DATASET_COMMENT"...	

Low code interface for customization (admin)

mc16_001:production > DATASET > LOGICALDATASETNAME

Metadata

Rank: 1

Description: LOGICALDATASETNAME

Hidden Admin only Crypted

Primary JSON

Statable Groupable

User free field Automatic field « Created » field « Created by » field « Modified » field « Modified by » field

Display Query

1 @NULL

Web link script

```
1 import net.hep.ami.jdbc.WebLink;
2
3 webLink = new WebLink();
4
5 if(rowSet.isANameOrLabel("LOGICALDATASETNAME"))
6 {
7     webLink.newLinkProperties().setLabel("#hashtags").setCtrl("HASHTAG").setLocation(WebLink.Location.BODY).setParams("[\"\" + row.getValue("LOGICALDATASETNAME") +
8     webLink.newLinkProperties().setLabel("Rucio").setCtrl("rucioElementInfo").setLocation(WebLink.Location.CONTAINER).setParams("[\"\" + row.getValue("LOGICALDATASE
9     webLink.newLinkProperties().setLabel("Provenance").setCtrl("graph").setLocation(WebLink.Location.CONTAINER).setParams("[\"GetDatasetProvenance -logicalDatasetN
10 }
11
12 if(rowSet.isANameOrLabel("projectName"))
13 {
14     webLink.newLinkProperties().setLabel("Series").setCtrl("table").setLocation(WebLink.Location.CONTAINER).setParams("[\"BrowseQuery -catalog=\\\\"Atlas_Production
15 }
16
17 return webLink;
```

	mc16_valid.361034.Pythia8EvtGen_A2MSTW2008LO_minbias_inelastic_... #hashtags - Rucio - Provenance - Series	ALL DATA DELETED:VALID CHILDREN	HITS	e3581_s2931 Datasets - AMI-Tags	0 Files	0
--	---	---------------------------------	------	------------------------------------	------------	---

Details and linked entities

empty fields hidden / shown More... ▾
★

Metadata		Linked Entities	
LOGICALDATASETNAME	mc16_valid.361034.Pythia8EvtGen_A2MSTW2008LO_minbias_inelastic_L... #hashtags - Rucio - Provenance - Series	←	DATASET_COMMENT 0 record(s)
PRODSYSSTATUS	EVENTS PARTIALLY AVAILABLE	←	DATASET_EXTRA 1 record(s)
DATATYPE	LOG	←	DATASET_KEYWORDS 0 record(s)
VERSION	e3581_s2931 Datasets - AMI-Tags	≡	PHYSICSPARAMETERS 0 record(s)
NFILES	0 Files	←	DATASET_PROPERTY_BRIDGE 0 record(s)
TOTALEVENTS	0	←	FILES 0 record(s)
COMPLETION	99.0 %	←	JOBOPTIONS 0 record(s)
STATSALGORITHM	exclude_outliers	←	PRODSYS_TASK 1 record(s)
PROJECTNAME	mc16_valid Project	←	PHYSICSPARAMETERVALS_ALL 0 record(s)
PHYSICSSHORT	Pythia8EvtGen_A2MSTW2008LO_minbias_inelastic_low	←	EI_METADATA_STATES_ALL 1 record(s)
PHYSICISTRESPONSIBLE	UNKNOWN	←	HASHTAGS 0 record(s)
PRINCIPALPHYSICSGROUP	gen-user	←	CAMPAIGN 1 record(s)
DATASETNUMBER	361034	←	PHYSICSPARAMETERVALS 0 record(s)
GEOMETRYVERSION	ATLAS-R2-2016-00-00-00_VALIDATION		
CONDITIONSTAG	OFLCOND-MC16-SDR-03		
BEAMTYPE	collisions		
RELATIONALLOADED	0		
PRODUCTIONSTEP	simul		
REQUESTEDBY	ycoadou		
AMISTATUS	VALID		
CREATED	2016-06-14 05:13:24.189510		
LASTMODIFIED	2019-11-05 14:57:35.720279		
GID	267210643		

DB visualization and browsing

The screenshot displays the AMI Frontend interface for database visualization and browsing. The top navigation bar includes the AMI logo, a menu with 'Datasets', 'Files', 'SW Images', 'AMI-Tags', 'Nomenclature', 'Tools', 'Issue reporting', and search icons, a 'CC-IN2P3 website' badge, a user profile 'jodier', and a 'Sign Out' button.

The left sidebar contains three sections:

- Catalogs:** A dropdown menu showing 'dataSuper_001:real_data', with 'Open' and 'Save' buttons, and two 'Flush server caches' buttons (full and partial).
- Box color:** A color selection bar.
- Import / Export schema:** A 'Drop a file' area and 'Export' and 'Print' buttons.

The main area shows a database schema diagram on a grid background. The tables and their fields are:

- OAIDEL:** IDN (INT)
- PRODSYS_TASK:** IDENTIFIER (INT, PK), PRODSYSIDENTIFIER (INT), DATASETFK (INT), TASKSTATUS (TEXT), MODIFIEDBY (TEXT), CREATEDBY (TEXT), LASTMODIFIED (TIMESTAMP), CREATED (TIMESTAMP)
- DATASET:** IDENTIFIER (INT, PK), LOGICALDATASETNAME (TEXT), PROJECTNAME (TEXT), VERSION (TEXT), STREAMNAME (TEXT), NFILES (INT), TOTALEVENTS (INT), CREATIONCOMMENT (TEXT), PRODSTEP (TEXT), DATATYPE (TEXT), TRASHTRIGGER (TEXT), TRASHANNOTATION (TEXT), TRASHDATE (TIMESTAMP), TRASHEDBY (TEXT), AMISTATUS (TEXT), CREATEDBY (TEXT), MODIFIEDBY (TEXT), CREATED (TIMESTAMP), LASTMODIFIED (TIMESTAMP), RUNNUMBER (TEXT), REQUESTEDBY (TEXT), TOTALSIZE (INT), GID (INT)
- HASHTAGS:** HASHTAGID (INT, PK), GLOBALDATASETFK (INT), DATASETCATALOG (TEXT), DATASETNAME (TEXT), DATASETFK (INT), SCOPEFK (INT), FULLNAME (TEXT), SCOPE (TEXT), NAME (TEXT), COMMENT (TEXT), ARCHIVED (INT), CREATED (TIMESTAMP), CREATEDBY (TEXT), MODIFIED (TIMESTAMP), MODIFIEDBY (TEXT)
- DATASET_COMMENT:** IDENTIFIER (INT, PK), DATASETFK (INT), TEXT (TEXT), CREATEDBY (TEXT), MODIFIEDBY (TEXT), CREATED (TIMESTAMP)
- CONTAINED_DATASET:** IDENTIFIER (INT, PK), DATASETFK (INT), CONTAINED_DATASETN... (TEXT), AMIPROJECTNAME (TEXT), AMIPROCESSNAME (TEXT), CREATEDBY (TEXT), MODIFIEDBY (TEXT)

Relationships are indicated by lines connecting fields in different tables: OAIDEL.IDN to PRODSYS_TASK.PRODSYSIDENTIFIER, PRODSYS_TASK.DATASETFK to DATASET.IDENTIFIER, DATASET.IDENTIFIER to DATASET_COMMENT.DATASETFK, DATASET.IDENTIFIER to HASHTAGS.HASHTAGID, and DATASET.IDENTIFIER to CONTAINED_DATASET.DATASETFK.

Interacting with AMI from the Web

ami Datasets Files SW Images AMI-Tags Nomenclature Tools Issue reporting CERN website Sign Out

Tools / Command

SearchQuery ?

Command

```
SearchQuery -catalog="mc23_001:production" -entity="dataset" -mq="SELECT * WHERE totalEvents=100000 AND dataType='EVNT' LIMIT 2 OFFSET 0"
```

Output format

TEXT

Search Execute

XML, JSON, CSV, TEXT, ...

```
1 #AMI Result
2
3 Rowset: mc23_001:production
4 Sql: SELECT `ATLAS_AMI_MC23_01_W`.`DATASET`.`LOGICALDATASETNAME`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PRODSYSSTATUS`,
`ATLAS_AMI_MC23_01_W`.`DATASET`.`DATATYPE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`VERSION`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`NFILES`,
`ATLAS_AMI_MC23_01_W`.`DATASET`.`TOTALEVENTS`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`CROSSSECTION`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`GENFILTEFF`,
`ATLAS_AMI_MC23_01_W`.`DATASET`.`TOTALSIZE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`ECMENERGY`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`COMPLETION`,
`ATLAS_AMI_MC23_01_W`.`DATASET`.`STATSALGORITHM`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`IDENTIFIER`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PROJECTNAME`,
`ATLAS_AMI_MC23_01_W`.`DATASET`.`PHYSICSCOMMENT`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PHYSICSSHORT`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PHYSICISTRESPONSIBLE`,
`ATLAS_AMI_MC23_01_W`.`DATASET`.`PRINCIPALPHYSICSGROUP`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`DATASETNUMBER`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`GENERATORNAME`,
`ATLAS_AMI_MC23_01_W`.`DATASET`.`GENERATORNAME`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PDF`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`GEOMETRYVERSION`,
`ATLAS_AMI_MC23_01_W`.`DATASET`.`CONDITIONSTAG`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`BEAMTYPE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`CREATIONCOMMENT`,
`ATLAS_AMI_MC23_01_W`.`DATASET`.`RELATIONALLOADED`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PRODUCTIONSTEP`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PRODUCTIONHISTORY`,
`ATLAS_AMI_MC23_01_W`.`DATASET`.`TRANSFORMATIONPACKAGE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`ATLASRELEASE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`REQUESTEDBY`,
```

Interacting with AMI from a shell

```
lambert@fedora:~ — ssh flambert@lxplus.cern.ch
-bash-4.2$ ami -e atlas-replica-v2 cmd SearchQuery -entity="dataset" -catalog="mc23_001:production" -mql="SELECT * WHERE totalEvents=100000 AND dataType='EVNT' LIMIT 2 OFFSET 0"
#AMI Result

Rowset: mc23_001:production
Sql: SELECT `ATLAS_AMI_MC23_01_W`.`DATASET`.`LOGICALDATASETNAME`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PRODSYSSTATUS`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`DATATYPE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`VERSION`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`NFILES`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`TOTALEVENTS`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`CROSSSECTION`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`GENFILTEFF`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`TOTALSIZE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`ECMENERGY`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`COMPLETION`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`STATSALGORITHM`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`IDENTIFIER`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PROJECTNAME`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PHYSICSCOMMENT`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PHYSICSSHORT`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PHYSICISTRESPONSIBLE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PRINCIPALPHYSICSGROUP`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`DATASETNUMBER`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`GENERATORNAME`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`GENERATORNAME`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PDF`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`GEOMETRYVERSION`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`CONDITIONSTAG`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`BEAMTYPE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`CREATIONCOMMENT`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`RELATIONALLOADED`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PRODUCTIONSTEP`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PRODUCTIONHISTORY`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`TRANSFORMATIONPACKAGE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`ATLASRELEASE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`REQUESTEDBY`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PHYSICSPROCESS`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PHYSICSCATEGORY`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`PHYSICSSUBCATEGORY`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`JOBCONFIG`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`TRIGGERCONFIG`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`TRASHTRIGGER`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`TRASHANNOTATION`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`TRASHDATE`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`TRASHEDBY`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`AMISTATUS`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`CREATEDBY`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`MODIFIEDBY`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`CREATED`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`LASTMODIFIED`, `ATLAS_AMI_MC23_01_W`.`DATASET`.`GID` FROM `ATLAS_AMI_MC23_01_W`.`DATASET` WHERE (`ATLAS_AMI_MC23_01_W`.`DATASET`.`TOTALEVENTS` = 100000 AND `ATLAS_AMI_MC23_01_W`.`DATASET`.`DATATYPE` = 'EVNT') LIMIT 2 OFFSET 0
Mql: SELECT * WHERE totalEvents=100000 AND dataType='EVNT' LIMIT 2 OFFSET 0
```


04

Metadata aggregation

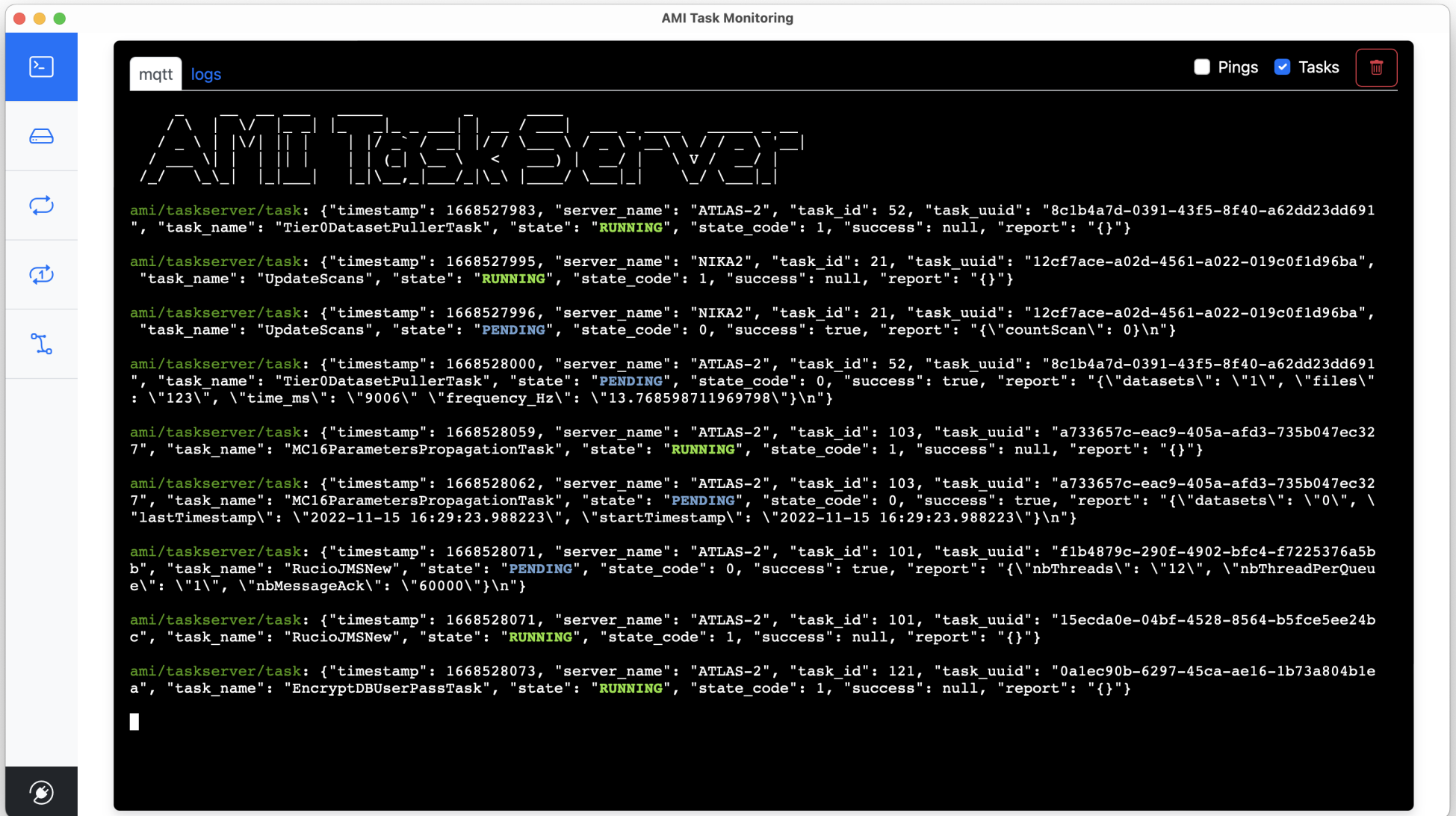


AMI Task Server



- **AMI Task server**
 - Extracting metadata from primary sources (pull mode)
 - (Re)Processing and storing metadata in AMI
- **It can run any kind of tasks (Shell, Python, JS, C++, Java, ...)**
 - Optionally benefits from the AMI Java Core library
- **Main features:**
 - Kind of super CRON
 - **The AMI Task Server is distributed**
 - Control and monitoring capabilities via MQTT
 - Mutual exclusion mechanism between tasks (with the AMI Exclusion Server)
 - Priority lottery scheduler for avoiding starvation (not real time)
 - **Pipelined tasks with execution report**

AMI Task Server



The screenshot shows a terminal window titled "AMI Task Monitoring" with a dark background. At the top, there are window control buttons (red, yellow, green) and a terminal icon. Below the title bar, there are tabs for "mqtt" and "logs", and a search bar. The main content is a log stream of JSON objects representing task events. The tasks shown include "Tier0DatasetPullerTask", "UpdateScans", "MC16ParametersPropagationTask", and "RucioJMSNew". The states of the tasks are "RUNNING", "PENDING", or "SUCCESS".

```
ami/taskserver/task: {"timestamp": 1668527983, "server_name": "ATLAS-2", "task_id": 52, "task_uuid": "8c1b4a7d-0391-43f5-8f40-a62dd23dd691", "task_name": "Tier0DatasetPullerTask", "state": "RUNNING", "state_code": 1, "success": null, "report": "{}"}
ami/taskserver/task: {"timestamp": 1668527995, "server_name": "NIKA2", "task_id": 21, "task_uuid": "12cf7ace-a02d-4561-a022-019c0f1d96ba", "task_name": "UpdateScans", "state": "RUNNING", "state_code": 1, "success": null, "report": "{}"}
ami/taskserver/task: {"timestamp": 1668527996, "server_name": "NIKA2", "task_id": 21, "task_uuid": "12cf7ace-a02d-4561-a022-019c0f1d96ba", "task_name": "UpdateScans", "state": "PENDING", "state_code": 0, "success": true, "report": "{\"countScan\": 0}\n"}
ami/taskserver/task: {"timestamp": 1668528000, "server_name": "ATLAS-2", "task_id": 52, "task_uuid": "8c1b4a7d-0391-43f5-8f40-a62dd23dd691", "task_name": "Tier0DatasetPullerTask", "state": "PENDING", "state_code": 0, "success": true, "report": "{\"datasets\": \"1\", \"files\": \"123\", \"time_ms\": \"9006\", \"frequency_Hz\": \"13.768598711969798\"}\n"}
ami/taskserver/task: {"timestamp": 1668528059, "server_name": "ATLAS-2", "task_id": 103, "task_uuid": "a733657c-eac9-405a-afd3-735b047ec327", "task_name": "MC16ParametersPropagationTask", "state": "RUNNING", "state_code": 1, "success": null, "report": "{}"}
ami/taskserver/task: {"timestamp": 1668528062, "server_name": "ATLAS-2", "task_id": 103, "task_uuid": "a733657c-eac9-405a-afd3-735b047ec327", "task_name": "MC16ParametersPropagationTask", "state": "PENDING", "state_code": 0, "success": true, "report": "{\"datasets\": \"0\", \"lastTimestamp\": \"2022-11-15 16:29:23.988223\", \"startTimestamp\": \"2022-11-15 16:29:23.988223\"}\n"}
ami/taskserver/task: {"timestamp": 1668528071, "server_name": "ATLAS-2", "task_id": 101, "task_uuid": "f1b4879c-290f-4902-bfc4-f7225376a5b", "task_name": "RucioJMSNew", "state": "PENDING", "state_code": 0, "success": true, "report": "{\"nbThreads\": \"12\", \"nbThreadPerQueue\": \"1\", \"nbMessageAck\": \"60000\"}\n"}
ami/taskserver/task: {"timestamp": 1668528071, "server_name": "ATLAS-2", "task_id": 101, "task_uuid": "15ecda0e-04bf-4528-8564-b5fce5ee24bc", "task_name": "RucioJMSNew", "state": "RUNNING", "state_code": 1, "success": null, "report": "{}"}
ami/taskserver/task: {"timestamp": 1668528073, "server_name": "ATLAS-2", "task_id": 121, "task_uuid": "0a1ec90b-6297-45ca-ae16-1b73a804b1ea", "task_name": "EncryptDBUserPassTask", "state": "RUNNING", "state_code": 1, "success": null, "report": "{}"}

```


AMI Task Server

The screenshot displays the 'AMI Task Monitoring' interface. At the top, it shows the task name 'Task RucioJMSNew' with a green status indicator. Below this, it provides the last start and stop dates (2022-11-15 17:01:11) and a 'Success' status. A row of buttons includes 'View server', 'Start task', 'Kill task', and 'Remove task'. The central part of the interface is a terminal window with tabs for 'mqtt', 'report', 'stdout', 'stderr', and 'logs'. The 'report' tab is active, showing a JSON object: `{"nbThreads": "12", "nbThreadPerQueue": "1", "nbMessageAck": "60000"}`. To the right of the terminal, there are several configuration fields: 'Server name' (ATLAS-2), 'Task name' (RucioJMSNew), 'Description' (Consume ActiveMQ rucio messages about ATLAS dataset and...), and 'Command' (a multi-line shell script). Below these are 'Priority [> 0]' (0) and 'Time step [s]' (1) dropdowns, an 'Exclusion locks' field, and a 'Task is unlocked' toggle switch (which is turned on). An 'Update' button is located at the bottom right of the configuration panel.

Pipelined tasks



- AMI provides a Node-RED-based interface for defining and monitoring pipelined tasks
- No single point of failure
- CHEP 2023 paper:
 - <https://cds.cern.ch/record/2867330/files/ATL-SOFT-PROC-2023-006.pdf>

The screenshot displays the AMI pipeline interface. The main workspace shows a Node-RED workflow with the following components:

- AMI Supervisor** (selected)
- Demo 1**
- Demo 2**

The workflow consists of the following nodes and connections:

- timestamp** (inject) node connected to **task1**.
- task1** (AMI task) node connected to **submission msg**, **starting msg**, **success msg**, and **error msg** (debug) nodes.
- task1** connected to **json** (parser) node.
- json** node connected to **task2**.
- task2** (AMI task) node connected to **msg.payload** (debug) node.

The **debug** console on the right shows the following log entries:

```
03/05/2022, 11:02:48 node: Submission msg
e1fb971f-3e8a-478c-b846-9b6c76e39cde : msg.payload :
Object
  > { AMIMessage: object }

03/05/2022, 11:02:51 node: 82cd536042459941
ami/taskserver/task : msg.payload : Object
  > object
    timestamp: 1651568571
    server_name: "DEM02"
    task_id: 13
    task_uuid: "eafa268d-143b-41c3-8084-8e8eecb108e0"
    task_name: "task-3d484818-0596-43cb"
    state: "FINISHED"
    success: true
    > report: "Hello jodier."
```

05

AMI live demo



AMI live demo

Run AMI on your laptop

<https://github.com/ami-team/AMIDemo>

<http://localhost:667/>

Conclusion



Conclusion

- **AMI is mature metadata ecosystem of more than 20 years of existence in the LHC context:**
 - } AMI and a data movement tool (for example Rucio) are complementary tools with different purposes
 - } AMI is easy to deploy and rescale (docker images), administrate (low code application design), and use
- **Benefits of using AMI in your experiment:**
 - } AMI is pluggable to any existing production system (databases and Web services)
 - ➔ AMI is not intrusive
 - } AMI aggregates low-level metadata into high-level metadata for physicists
 - Housekeeping, dataset and file, end-user, software, ... metadata
 - } AMI provides Web services, interfaces, and clients (Python, JS, C++, Java) for easily select data by metadata criteria
 - } AMI can associate data and papers for data preservation & reproducibility
- **Prerequisites: having well-defined low level metadata at data production level**

Thank You for your attention



LPSC - Grenoble



ami@lpsc.in2p3.fr
<https://ami-ecosystem.in2p3.fr/>



