ami

Scientific Metadata Ecosystem

Fabian Lambert, Jérôme Fulachier, Jérôme Odier, Maxime Jaume, Pierre-Antoine Delsart

What is AMI?

- AMI (ATLAS Metadata Interface) is a generic ecosystem for metadata:
  - Heterogenous datasource connectivity
  - Primitives for metadata extraction and processing
  - High level tools for selecting data by metadata criteria

- The ecosystem has development kits for:
  - Developing JAVA business objects (server-side)
  - Developing metadata-oriented Web applications (client-side)

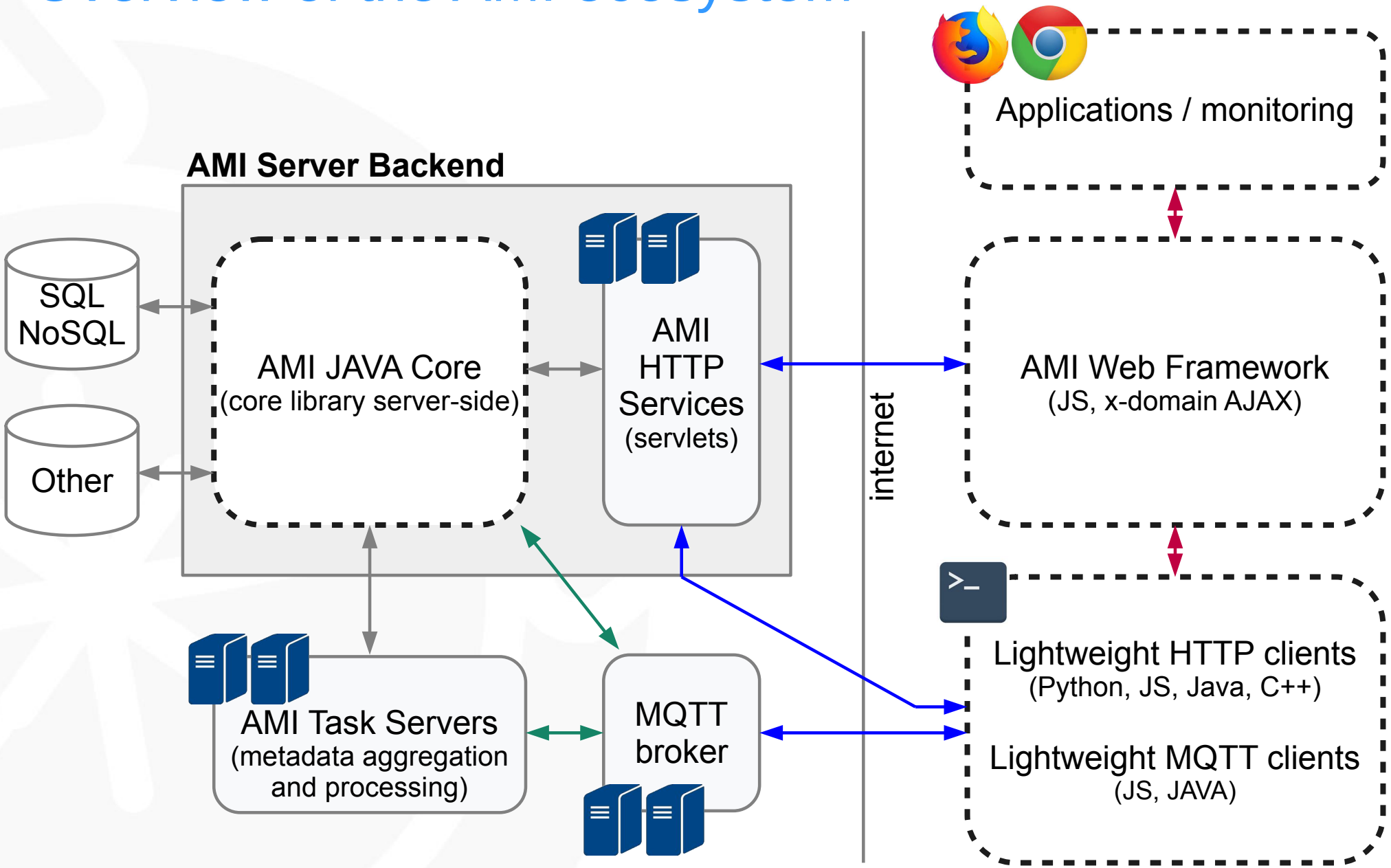
- AMI is designed for:
  - Scalability, evolutivity and maintainability

History of AMI

- AMI is mature ecosystem of more than 20 years of existence
- Originally developed for the ATLAS experiment at CERN
  - 2000 → Metadata bookkeeping for the Liquid Argon (LAr) calorimeter
  - 2001-2016 → ATLAS software release management (TagCollector)
  - 2006-ongoing → AMI provides the ATLAS official dataset discovery tool
  - 2015-ongoing → parameter definition for dataset processing (AMI-Tags)
  - …

  → Used by the ATLAS production system
- Development of AMI 2.X started in 2015 for:
  - Simplifying deployments
  - Avoiding subsystem entanglements
  - Improving maintainability and performance (to be ready for ATLAS Run 3)
- Since 2017-2018, AMI 2.X is used by:
  - ATLAS, NIKA2, n2EDM, SuperNemo, Rosetta Philae, ...
  - administration forms @ LPSC

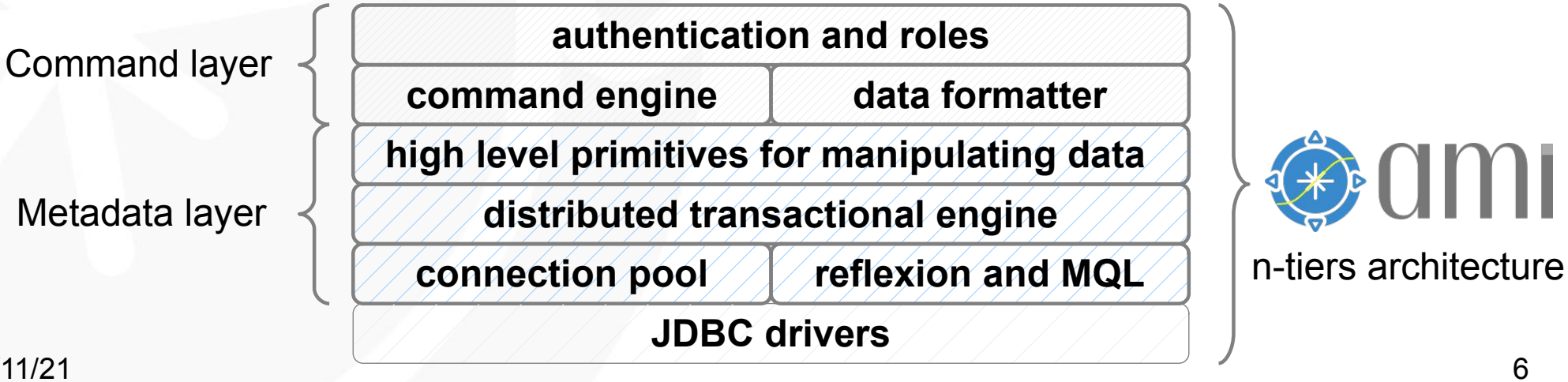
Overview of the AMI ecosystem



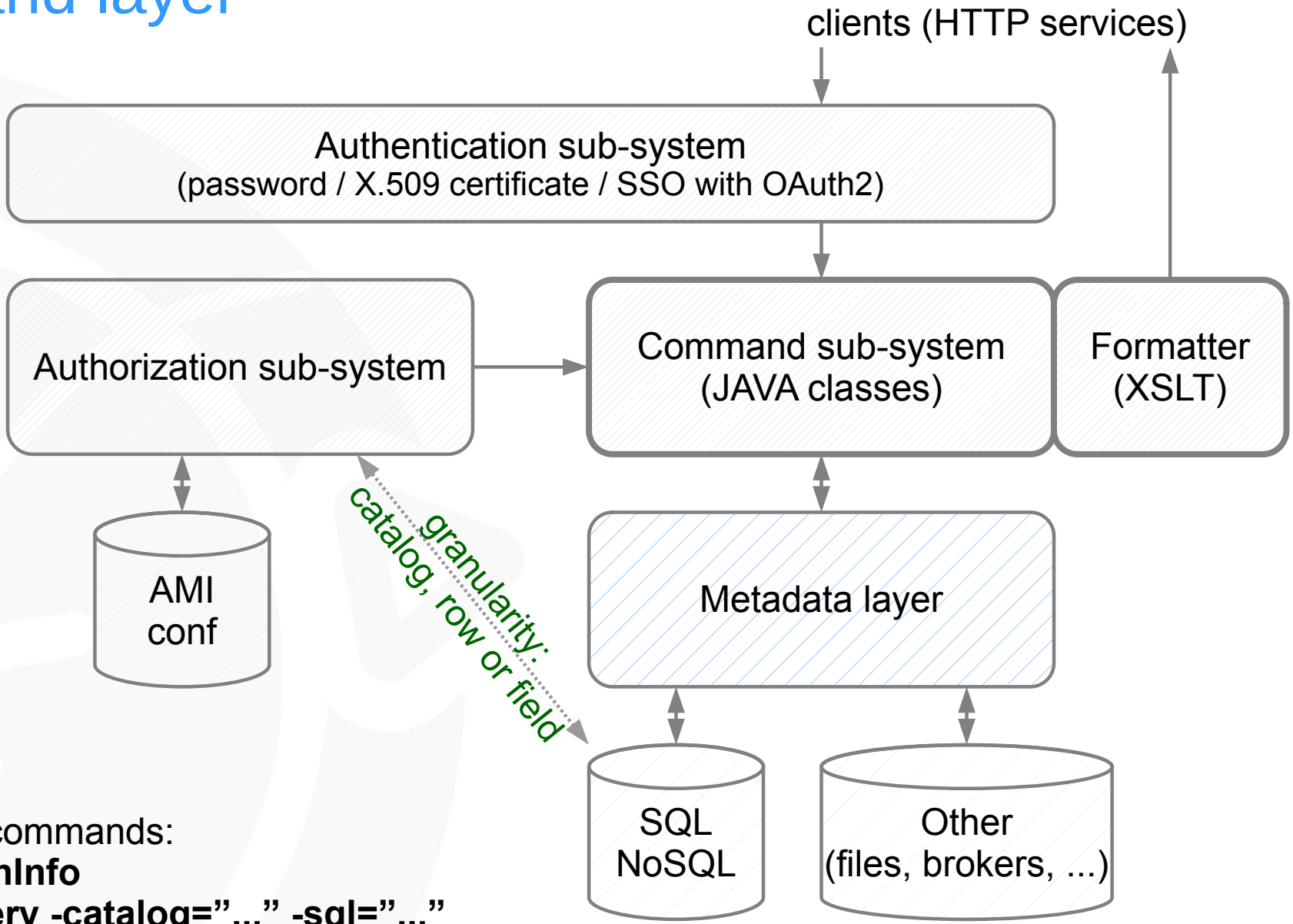
AMI JAVA Core

AMI JAVA Core features

- AMI JAVA Core is the central part of the AMI ecosystem
- Main features:
  - Authentication and authorizations
  - Command engine (~100 generic commands, ~500 ATLAS-specific commands)
    - Metadata queries (trivial [SQL, MQL] or more complex, read or write), experiment-specific commands, service administration, ...
  - High level primitives for manipulating data
    - DB rowsets, JSON documents, XML documents, remote access, ...
  - Metadata Query Language (MQL) and Structured Query Language (SQL)



Command layer



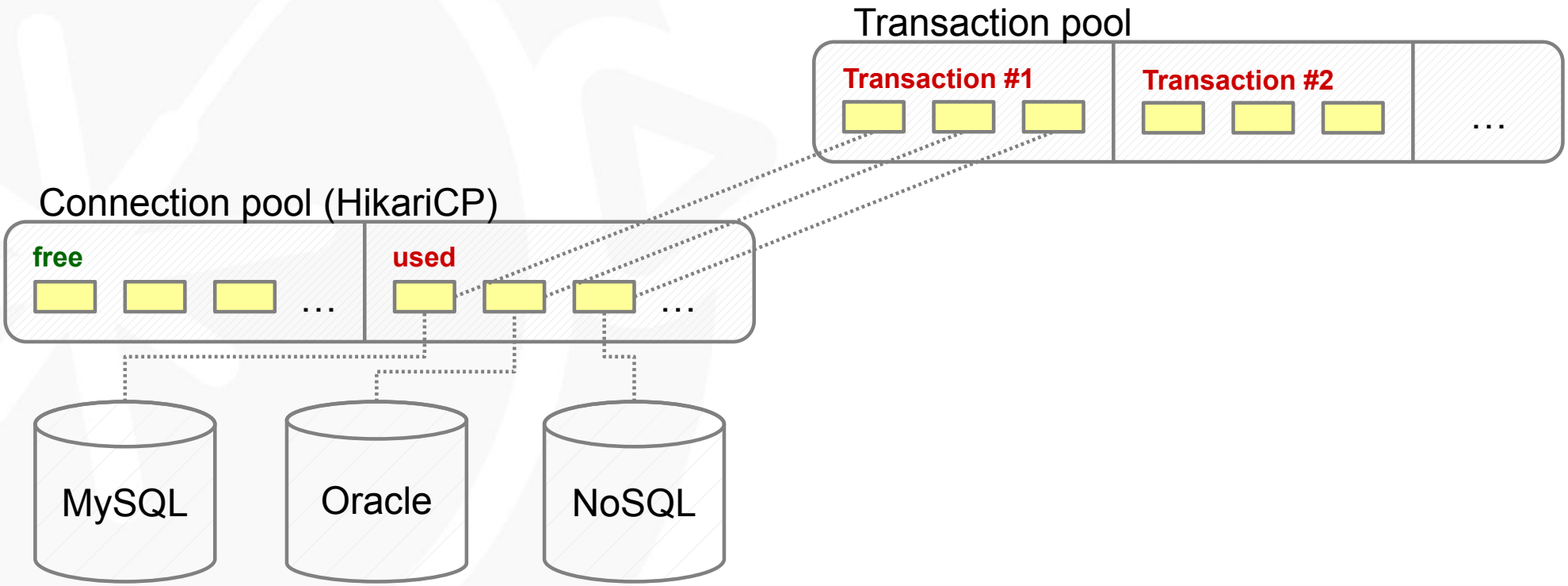
Example of commands:

GetSessionInfo

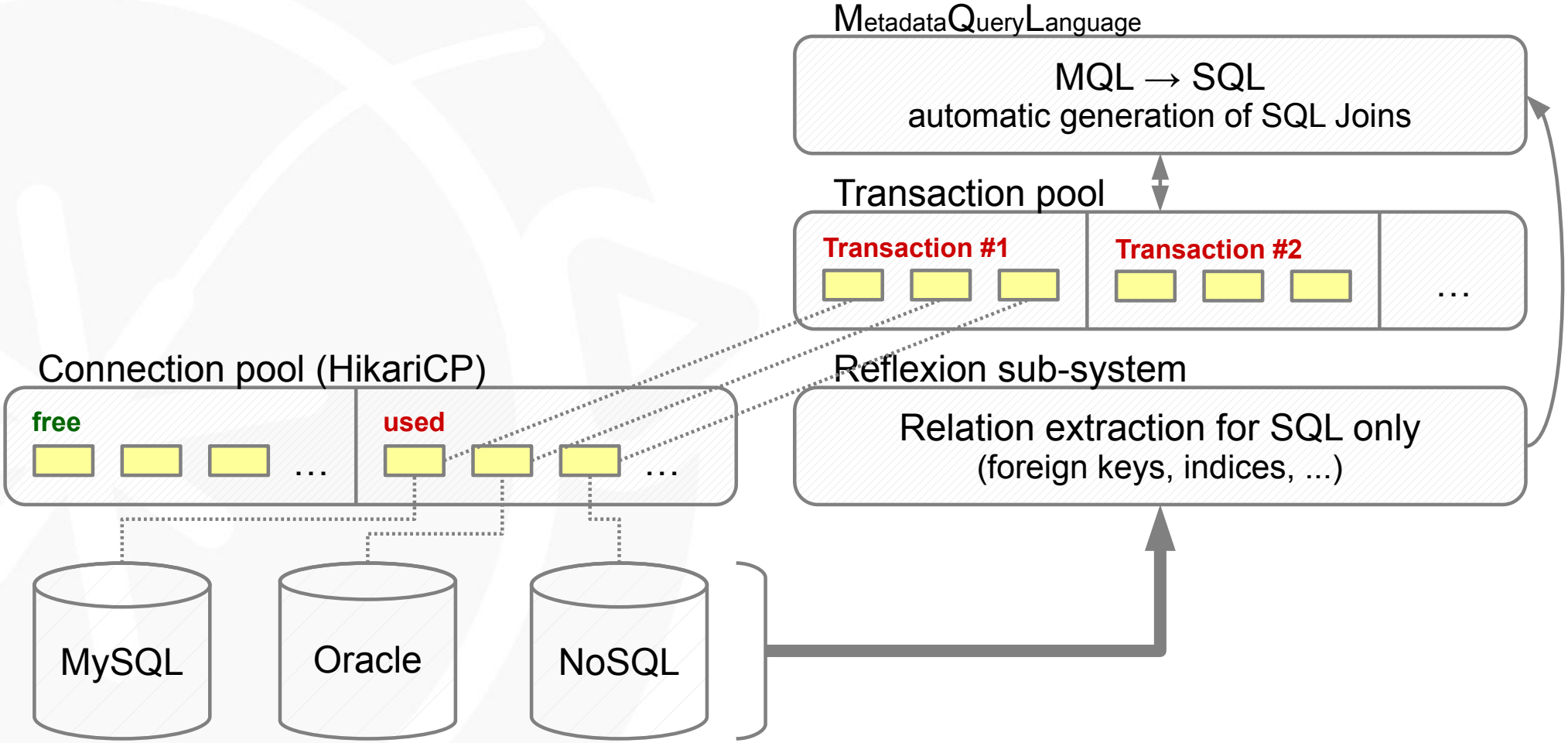
SearchQuery -catalog="..." -sql="..."

GetDatasetInfo -logicalDatasetName="..." (for ATLAS, getting detailed dataset info)

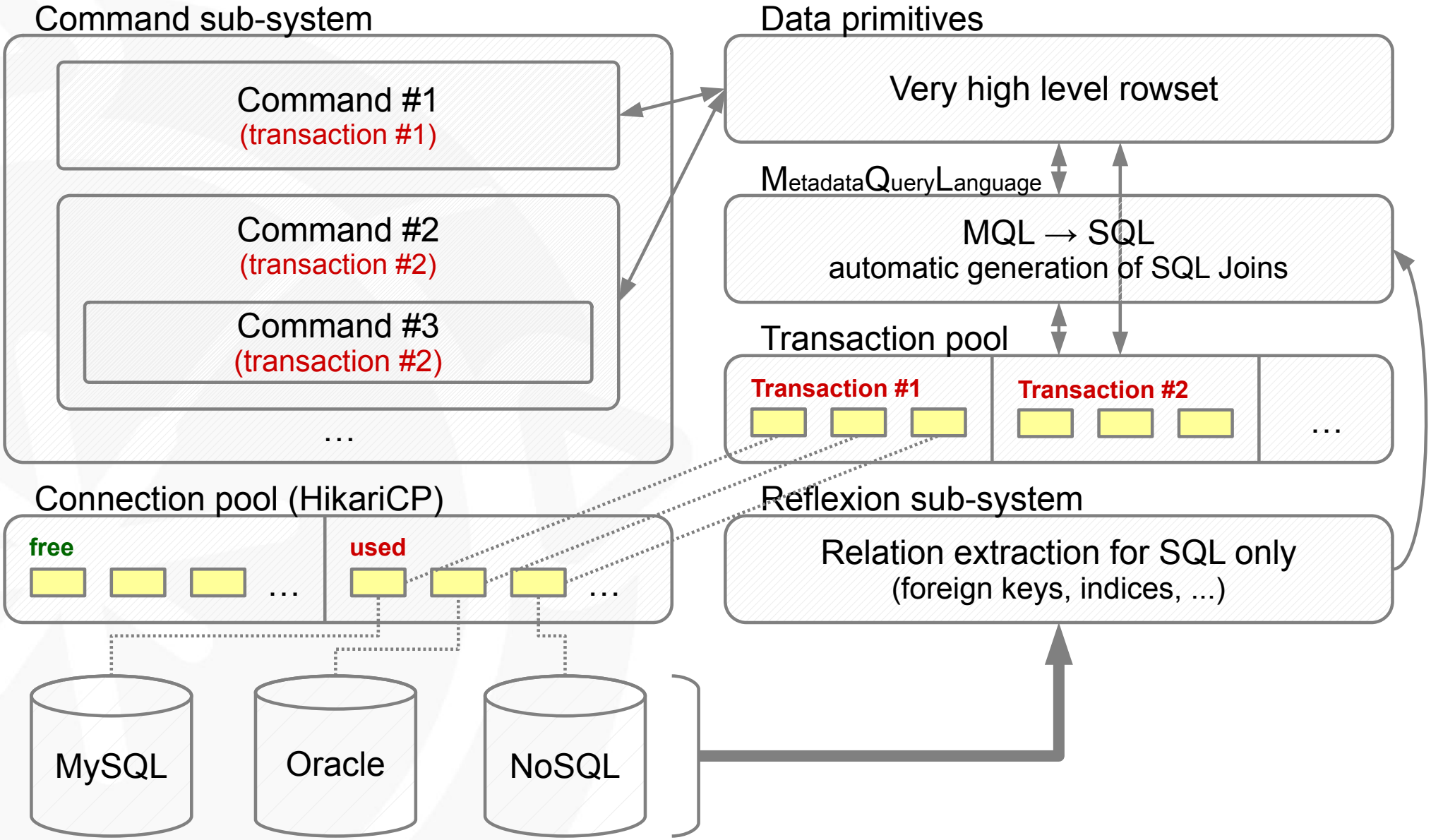
Metadata layer



Metadata layer



Metadata layer



Metadata Query Language (MQL)

- MQL is a kind of SQL without FROM clause nor join
- It makes it possible to build queries without (precisely) knowing relations
- Joins are automatically generated from the reflexion sub-system info
- MQL turns DB-oriented point of view to metadata-oriented point of view
- When there are cycles in relations, there is a dedicated syntax to apply path constraints

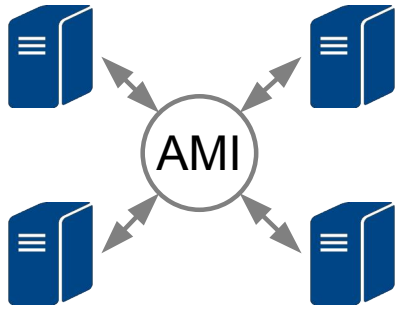
```
SELECT * WHERE (`AMISTATUS` = 'VALID')
               AND
               [`DATASET_KEYWORDS`.`KEYWORD` = 'stau']
               AND
               [`KEYWORD` = 'stop']
```

MQL to SQL

```
SELECT * FROM `ATLAS_AMI_MC16_02`.`DATASET`
WHERE
  (
    `ATLAS_AMI_MC16_02`.`DATASET`.`AMISTATUS` = 'VALID'
  )
  AND (
    `ATLAS_AMI_MC16_02`.`DATASET`.`IDENTIFIER` IN (
      SELECT
        `ATLAS_AMI_MC16_02`.`DATASET`.`IDENTIFIER`
      FROM
        `ATLAS_AMI_MC16_02`.`DATASET_KEYWORDS`,
        `ATLAS_AMI_MC16_02`.`DATASET`
      WHERE
        (
          `ATLAS_AMI_MC16_02`.`DATASET_KEYWORDS`.`KEYWORD` = 'stau'
        )
        AND `ATLAS_AMI_MC16_02`.`DATASET_KEYWORDS`.`DATASETFK` = `ATLAS_AMI_MC16_02`.`DATASET`.`IDENTIFIER`
    )
  )
  AND (
    `ATLAS_AMI_MC16_02`.`DATASET`.`IDENTIFIER` IN (
      SELECT
        `ATLAS_AMI_MC16_02`.`DATASET`.`IDENTIFIER`
      FROM
        `ATLAS_AMI_MC16_02`.`DATASET_KEYWORDS`,
        `ATLAS_AMI_MC16_02`.`DATASET`
      WHERE
        (
          `ATLAS_AMI_MC16_02`.`DATASET_KEYWORDS`.`KEYWORD` = 'stop'
        )
        AND `ATLAS_AMI_MC16_02`.`DATASET_KEYWORDS`.`DATASETFK` = `ATLAS_AMI_MC16_02`.`DATASET`.`IDENTIFIER`
    )
  )
```

AMI Task Server

AMI Task Server features



for instance
4 AMI Task Servers

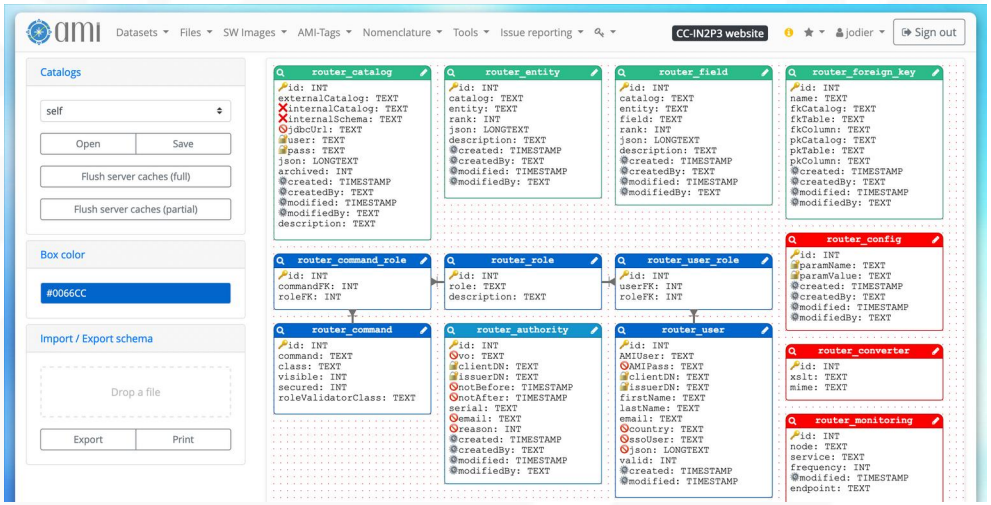
- The AMI Task Server is used for:
  - Extracting metadata from primary sources (pull mode)
  - (Re)Processing metadata
  - Storing metadata in AMI
- It can run any kind of tasks (shell, python, java, ...)
- When needed, it can benefit from the AMI Java Core library
- Main features:
  - Kind of super CRON
  - The AMI Task Server is distributed
  - Web interfaces and monitoring (via MQTT)
  - Mutual exclusion mechanism between tasks
  - Priority lottery scheduler for avoiding starvation (not real time)
  - One shot tasks

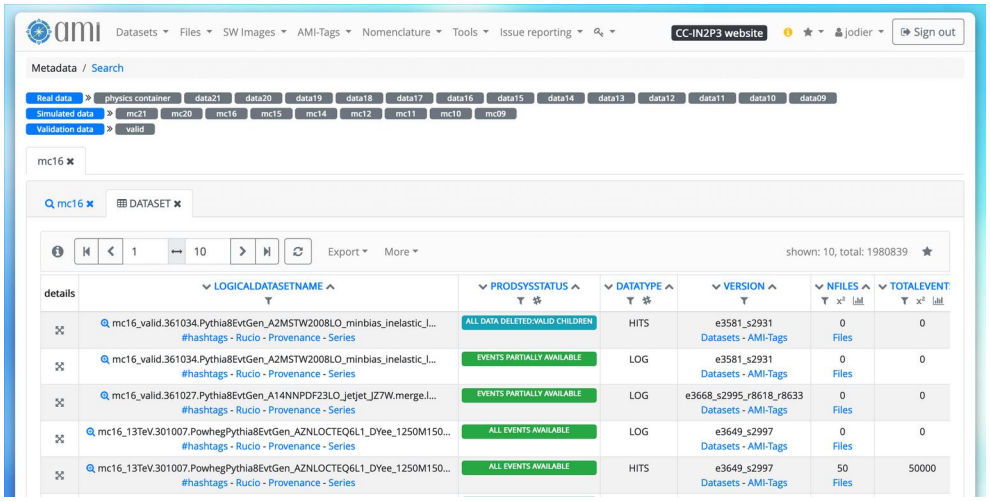
In ATLAS:
- Metadata extraction from Tier0 (real data)
- Metadata extraction from ProdSys (simulated data, reprocessing)
- ActiveMQ messages from RUCIO (data placement)
- ~50 other tasks

AMI Web Framework

AMI Web Framework (AWF)

- A Web framework for designing metadata-oriented applications
- AWF can be used without the AMI Server Backend
  - Server-side, libraries AMIMini{PHP,Python,JAVA} can easily bridge AWF to existing services
- AWF is based on standard technologies:
  - JS2020 (transpiled to JS5 using webpack and babel), CSS3, HTML5
  - JQuery, Twitter Bootstrap 4 and 5, AMI-Twig (MVC pattern, homemade JavaScript Twig template engine implementation), Vue.js 3 (MVVM pattern)





Features and patterns

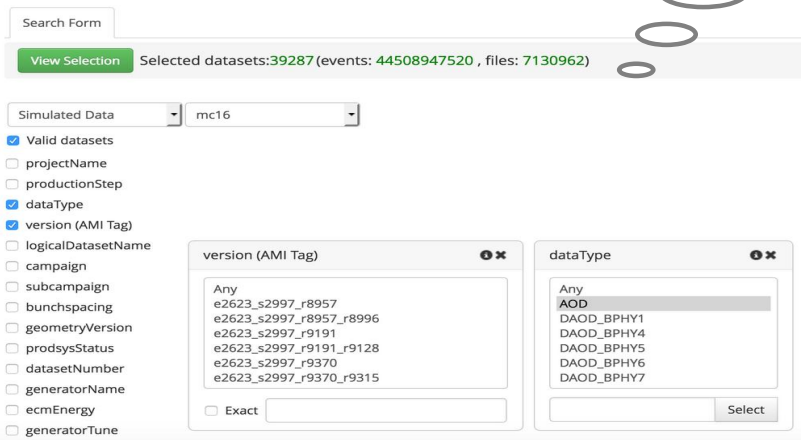
- Authentication & roles
- URL router, short URLs
- Sub-applications and reusable graphic controls (object paradigm)
- Centralized resource live cycle management (CSS, JS, JSON, xml, Twig files; AMI sub-applications; AMI controls)
- Wizards for generating sub-application and control skeletons
- Patterns:
  - MVC
    - Model → AMI commands
    - View → TWIG templates
    - Controller → classes ami.SubApp, ami.Control (JavaScript)
  - MVVM
    - based on Vue.js 3 (new!)

Default controls and applications

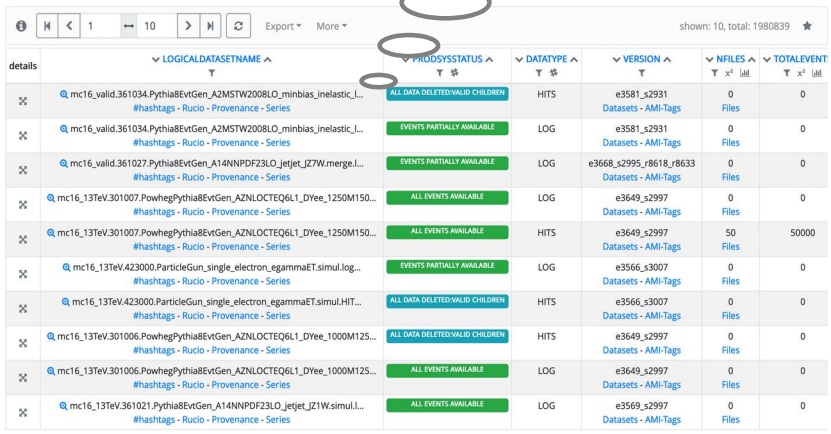
- Controls can be embedded in external Web pages such as wikis
- Applications are generally built by assembling controls
- Main available controls:
  - Dialog boxes
  - Controls for searching (Google-like Search, Criteria Search, …)
  - Controls for displaying (Schema Viewer, Tab, Table, Element Info, …)
  - Controls for annotating entities (WhiteBoard, …)
- Main available applications:
  - Embedded CMS
  - AMI command interpreter
  - Admin Dashboard and Monitoring
  - Schema Viewer, Table Viewer, Simple Search, Criteria Search, Search Modeler, …

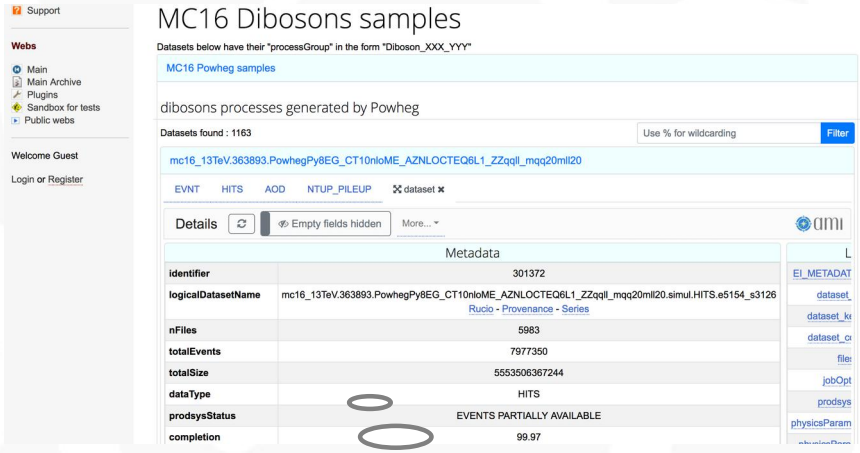
Screenshots

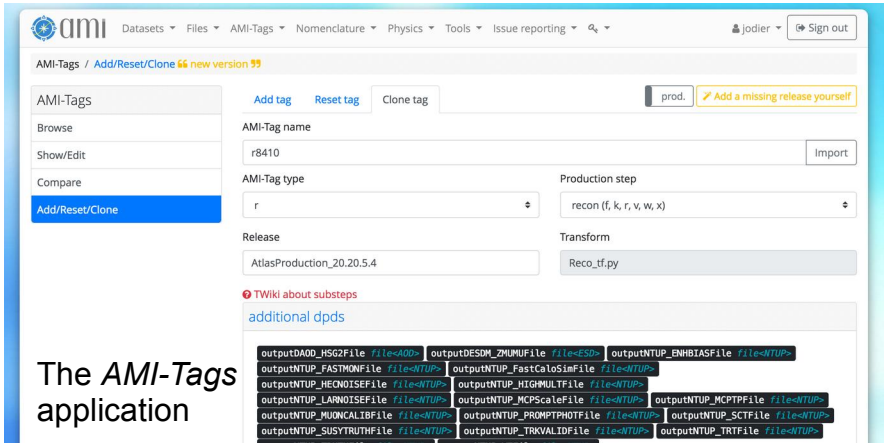
Searching ATLAS datasets by criteria



Displaying search results in AMI





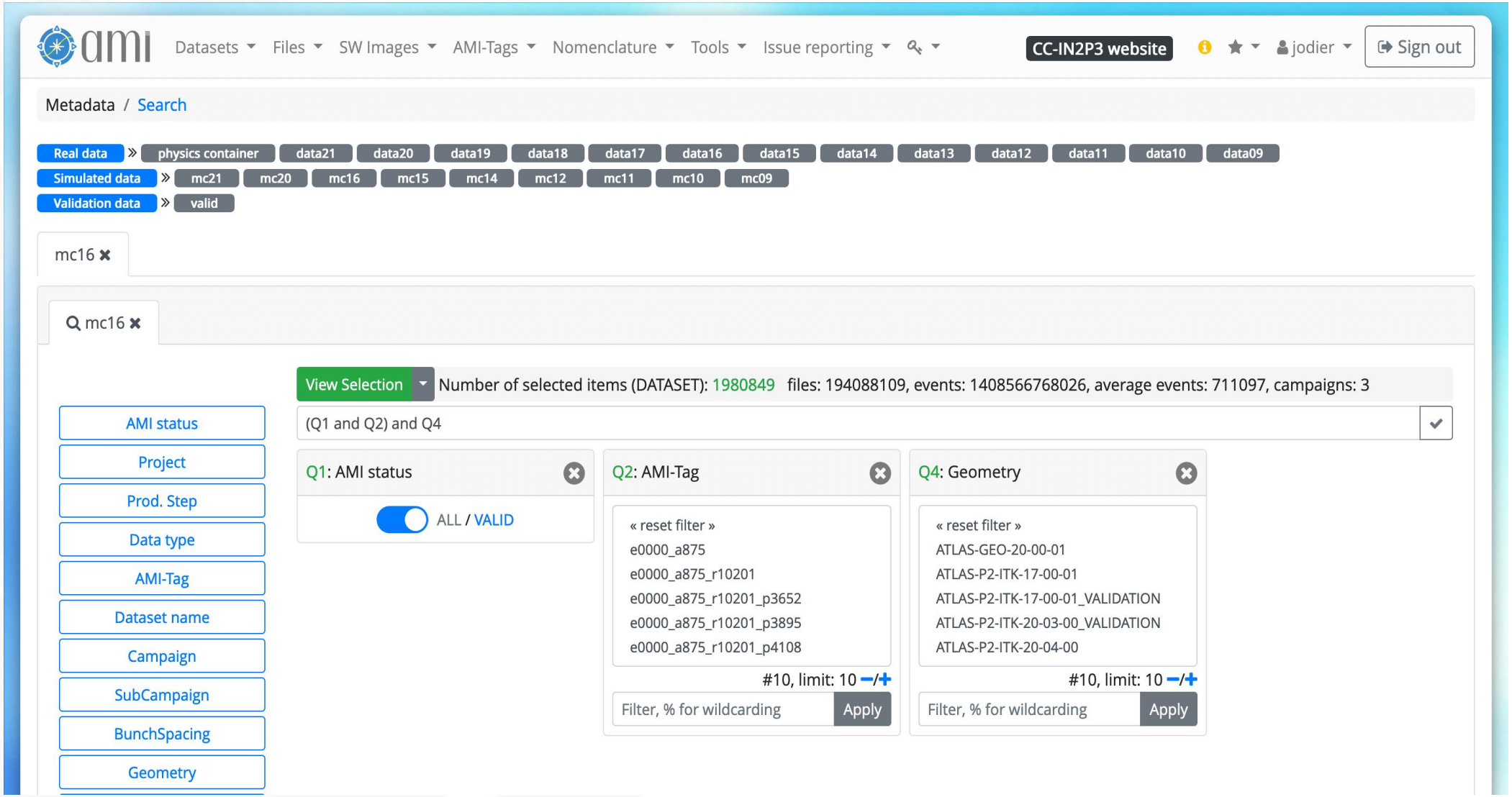


The *AMI-Tags* application

A control embedded in a wiki and connected to the central AMI service

This control executes the **GetDatasetInfo** command

Screenshots

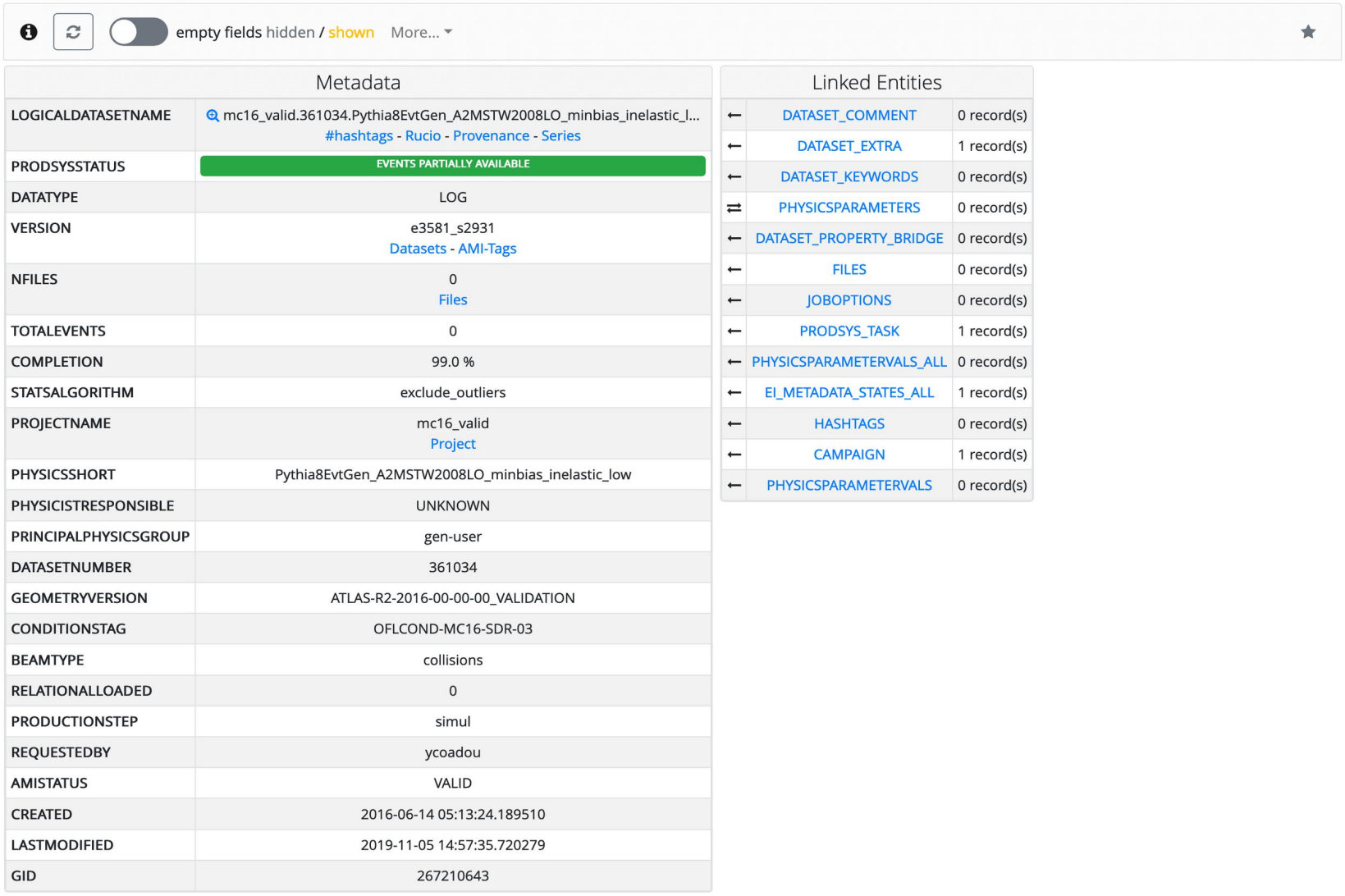


Screenshots

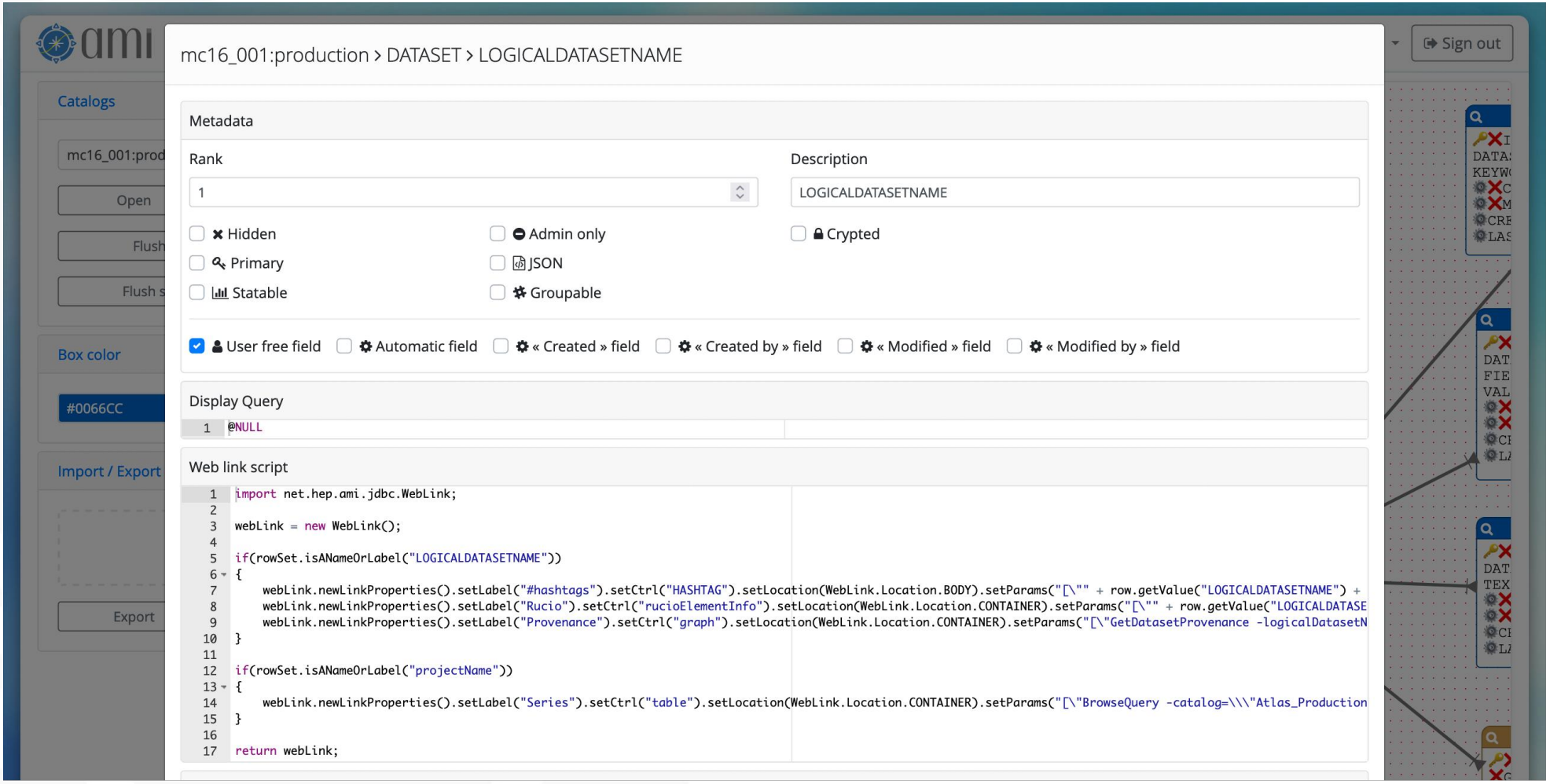
1 | 10 | Export | More | shown: 10, total: 1980839

details	LOGICALDATASETNAME	PRODSYSSTATUS	DATATYPE	VERSION	NFILES	TOTALEVENT
	mc16_valid.361034.Pythia8EvtGen_A2MSTW2008LO_minbias_inelastic_l... #hashtags - Rucio - Provenance - Series	ALL DATA DELETED:VALID CHILDREN	HITS	e3581_s2931 Datasets - AMI-Tags	0 Files	0
	mc16_valid.361034.Pythia8EvtGen_A2MSTW2008LO_minbias_inelastic_l... #hashtags - Rucio - Provenance - Series	EVENTS PARTIALLY AVAILABLE	LOG	e3581_s2931 Datasets - AMI-Tags	0 Files	0
	mc16_valid.361027.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ7W.merge.l... #hashtags - Rucio - Provenance - Series	EVENTS PARTIALLY AVAILABLE	LOG	e3668_s2995_r8618_r8633 Datasets - AMI-Tags	0 Files	0
	mc16_13TeV.301007.PowhegPythia8EvtGen_AZNLOCTEQ6L1_DYee_1250M150... #hashtags - Rucio - Provenance - Series	ALL EVENTS AVAILABLE	LOG	e3649_s2997 Datasets - AMI-Tags	0 Files	0
	mc16_13TeV.301007.PowhegPythia8EvtGen_AZNLOCTEQ6L1_DYee_1250M150... #hashtags - Rucio - Provenance - Series	ALL EVENTS AVAILABLE	HITS	e3649_s2997 Datasets - AMI-Tags	50 Files	50000
	mc16_13TeV.423000.ParticleGun_single_electron_egammaET.simul.log... #hashtags - Rucio - Provenance - Series	EVENTS PARTIALLY AVAILABLE	LOG	e3566_s3007 Datasets - AMI-Tags	0 Files	0
	mc16_13TeV.423000.ParticleGun_single_electron_egammaET.simul.HIT... #hashtags - Rucio - Provenance - Series	ALL DATA DELETED:VALID CHILDREN	HITS	e3566_s3007 Datasets - AMI-Tags	0 Files	0
	mc16_13TeV.301006.PowhegPythia8EvtGen_AZNLOCTEQ6L1_DYee_1000M125... #hashtags - Rucio - Provenance - Series	ALL DATA DELETED:VALID CHILDREN	HITS	e3649_s2997 Datasets - AMI-Tags	0 Files	0
	mc16_13TeV.301006.PowhegPythia8EvtGen_AZNLOCTEQ6L1_DYee_1000M125... #hashtags - Rucio - Provenance - Series	ALL EVENTS AVAILABLE	LOG	e3649_s2997 Datasets - AMI-Tags	0 Files	0
	mc16_13TeV.361021.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ1W.simul.l... #hashtags - Rucio - Provenance - Series	ALL EVENTS AVAILABLE	LOG	e3569_s2997 Datasets - AMI-Tags	0 Files	0

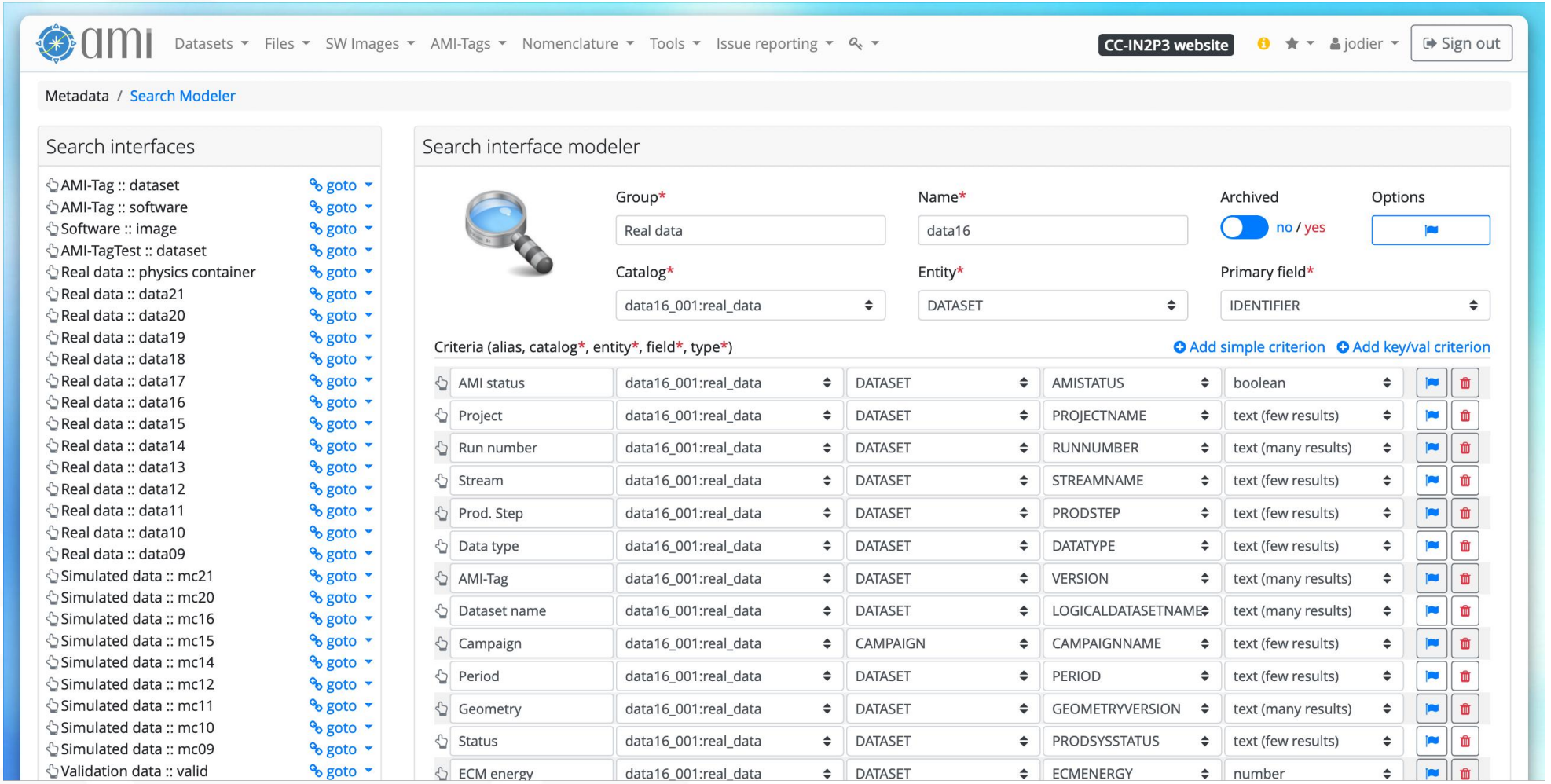
Screenshots



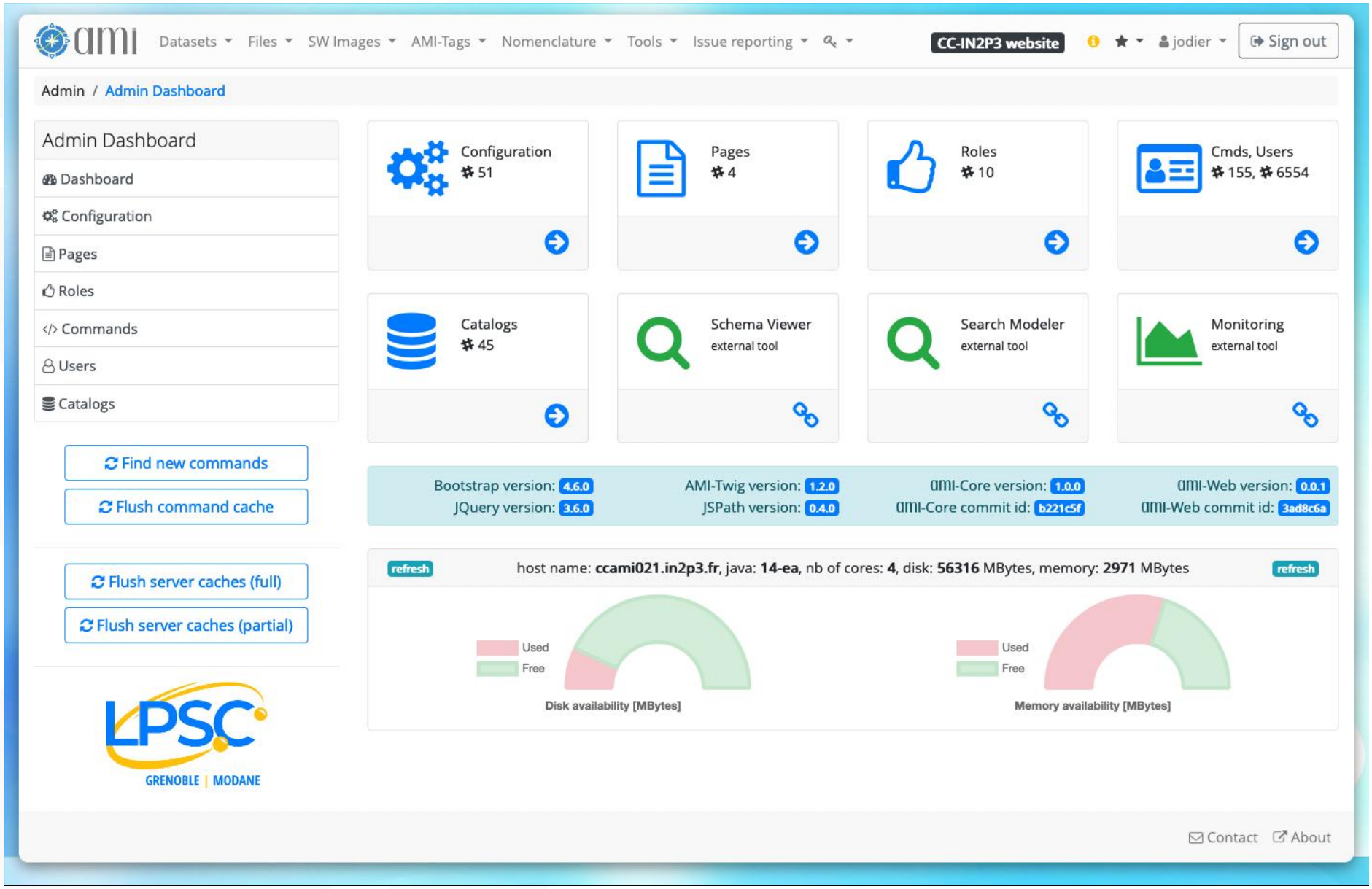
Screenshots



Screenshots



Screenshots



Conclusion

Conclusion

- AMI is mature metadata ecosystem of more than 20 years of existence
- Originally developed for the ATLAS experiment:
  - i) Official dataset discovery tool (millions of datasets, billions of files),

    ii) Used by the ATLAS production system (parameter definition for dataset processing), …
- AMI Java Core
  - High level server-side JAVA library for processing metadata

    i) High level primitives for manipulating metadata,

    ii) Metadata Query Language (MQL),

    iii) datasource connectivity.
- AMI Services + lightweight clients
  - AMI HTTP command service (proprietary), REST API, MQTT server control and monitoring
- AMI Task Server
  - Distributed system for extracting, processing and storing metadata
- AMI Web Framework
  - For developing metadata-oriented Web applications and graphic controls

Questions?