ami

Scientific Metadata Ecosystem

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

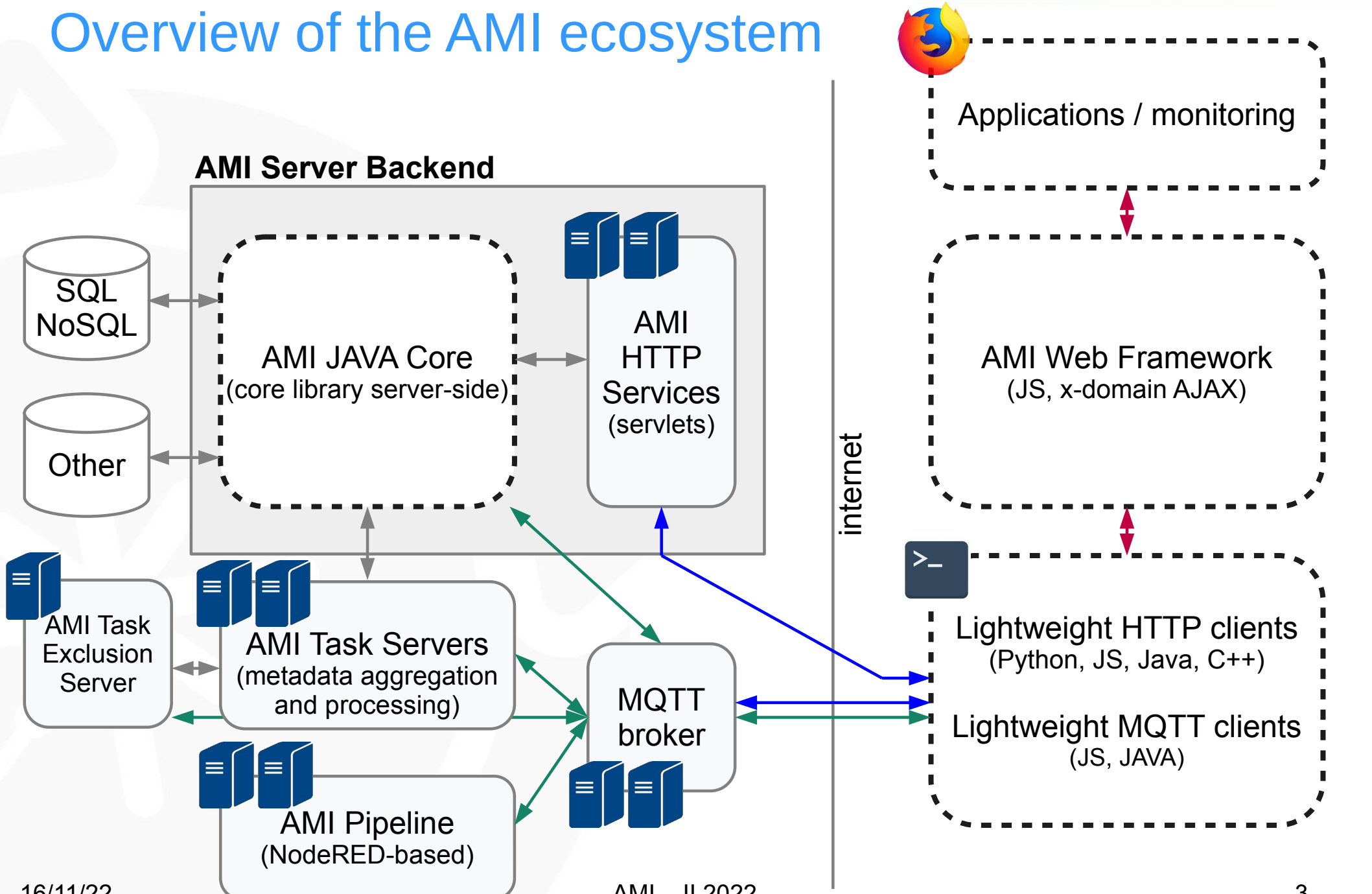
What is AMI?

- 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

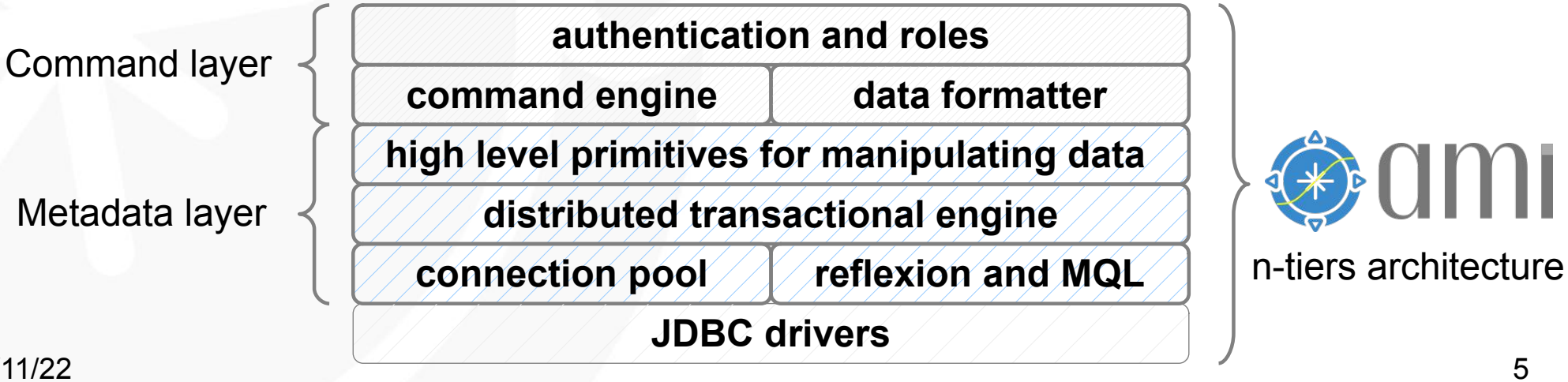
Overview of the AMI ecosystem



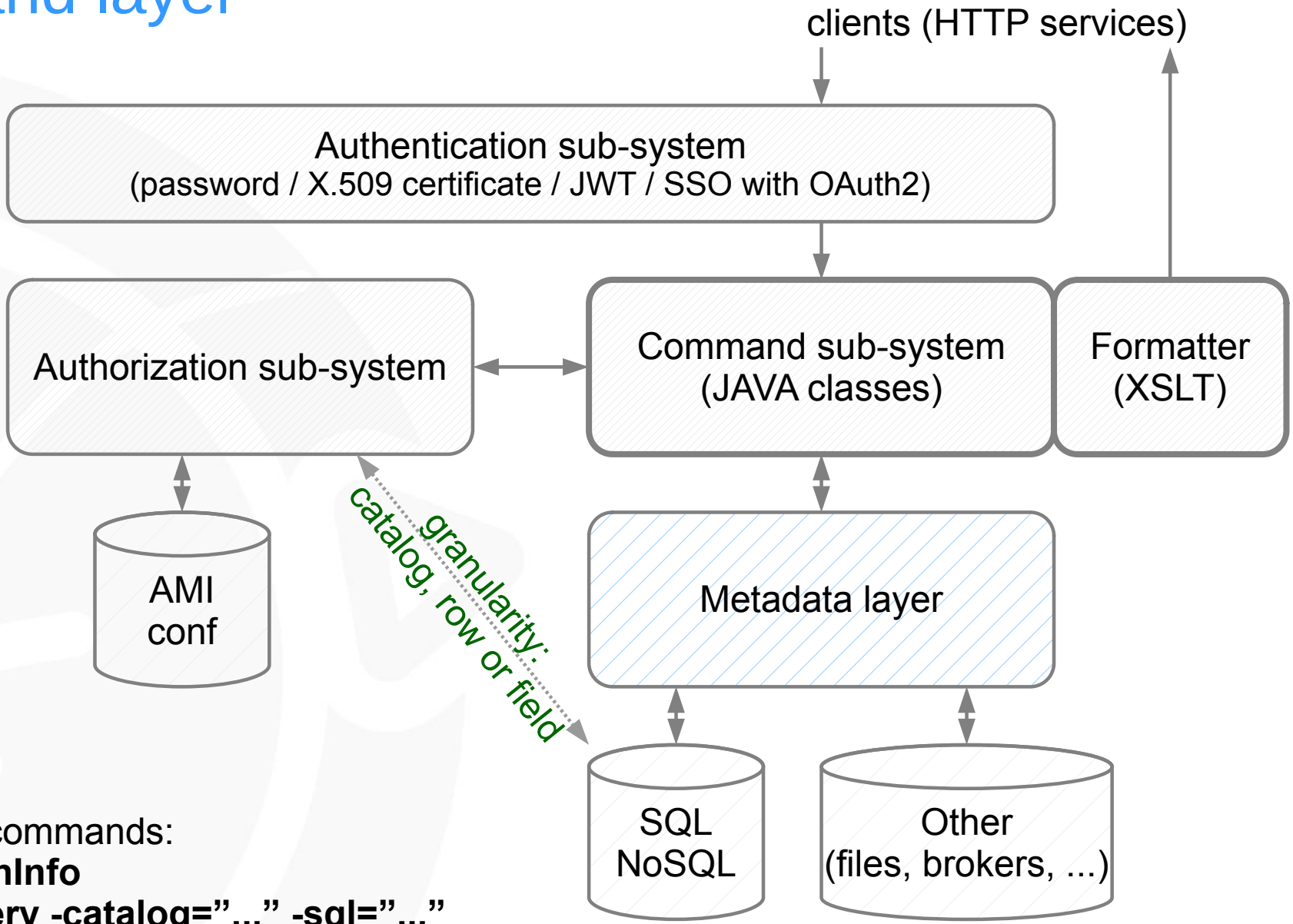
AMI JAVA Core

AMI JAVA Core features

- AMI JAVA Core is the central part of the AMI ecosystem
- Main features:
  - Authentication (SSO, OAuth2) and authorizations
  - Command engine (~100 generic commands)
    - Metadata queries (trivial [SQL, MQL] or more complex, read or write), experiment-specific commands, service administration, …
  - Metadata Query Language (MQL) and Structured Query Language (SQL)
  - High level primitives for manipulating data
    - DB rowsets, JSON documents, XML documents, remote access, ...



Command layer



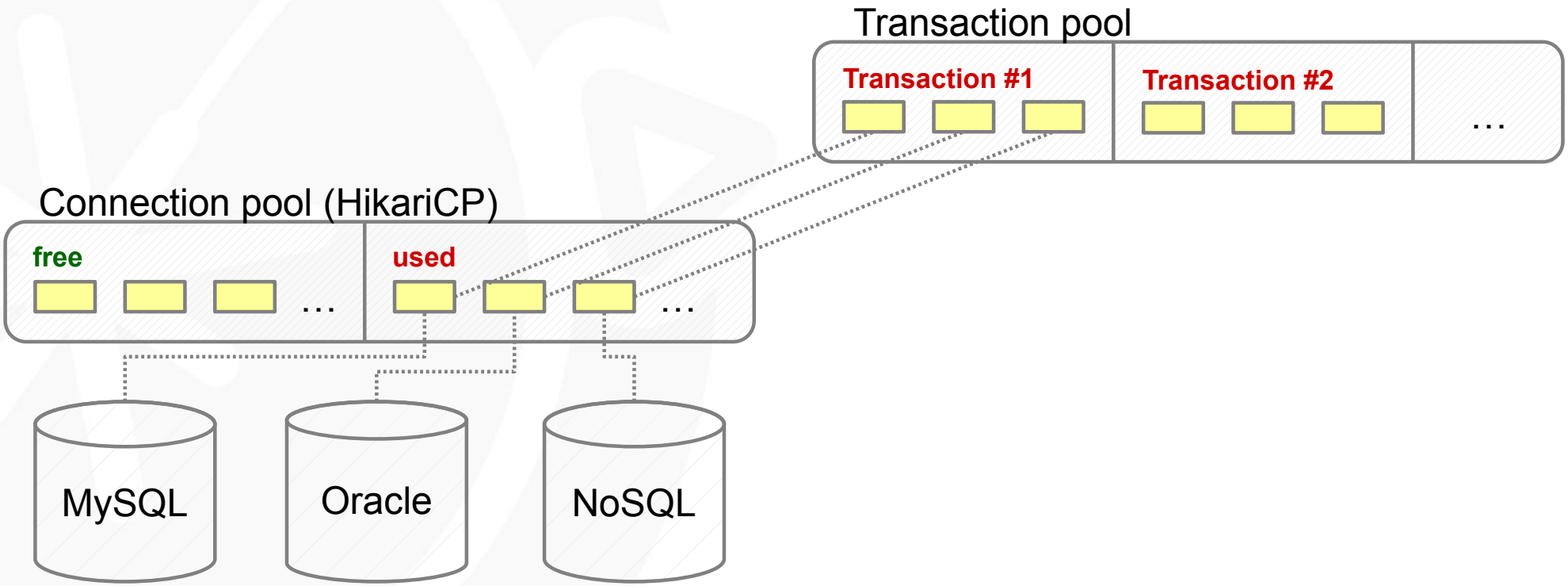
Example of commands:

GetSessionInfo

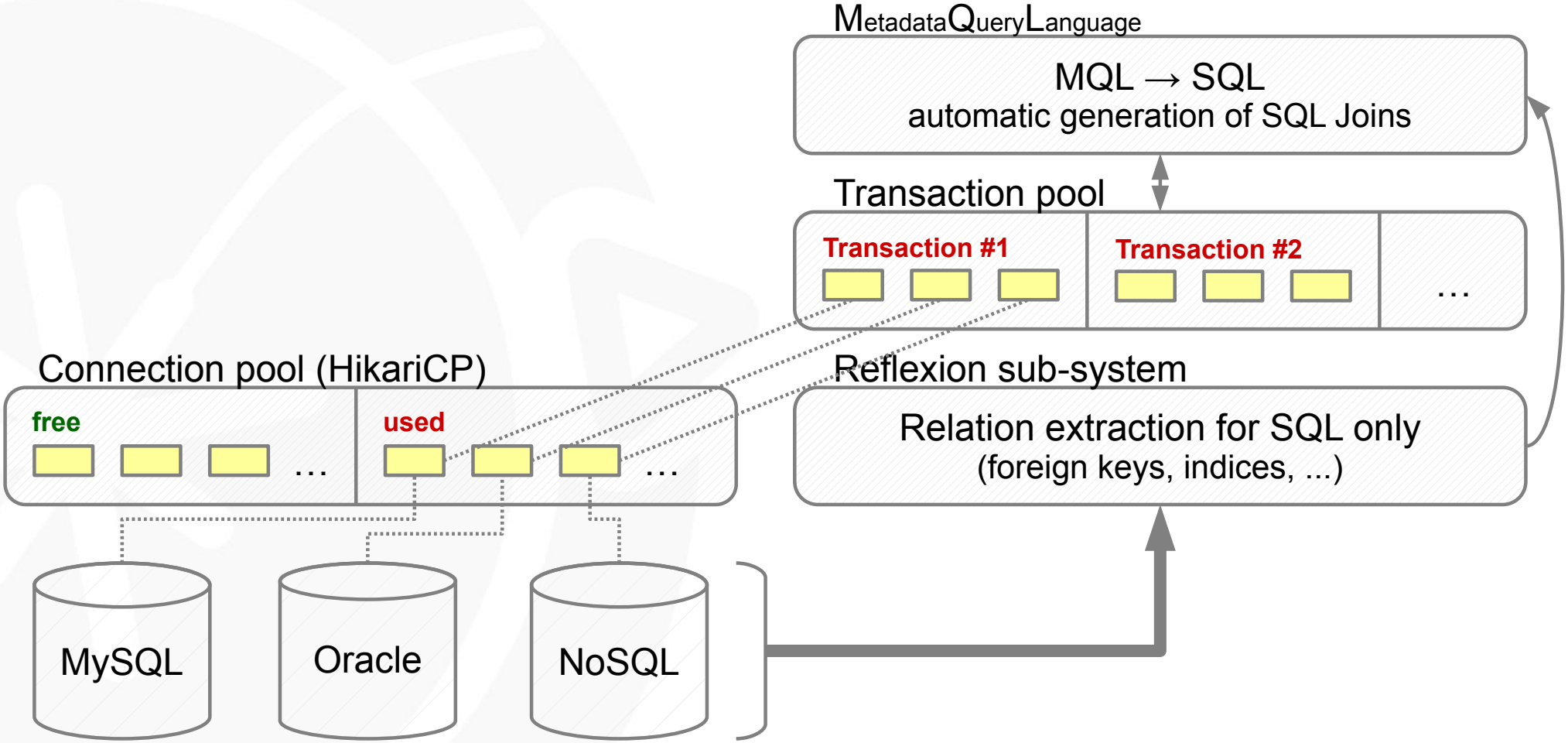
SearchQuery -catalog="..." -sql="..."

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

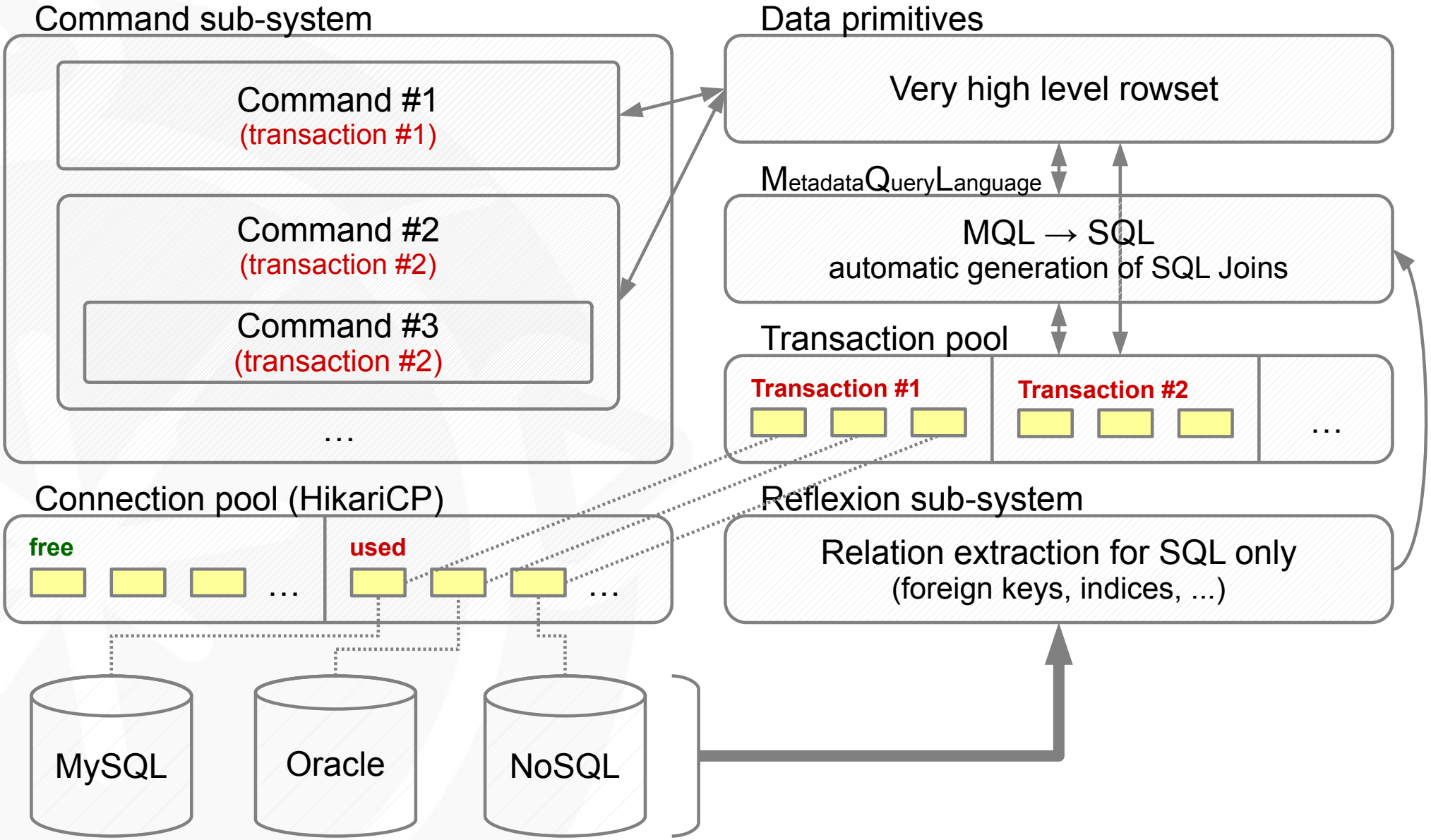
Metadata layer



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`
    )
  )
```

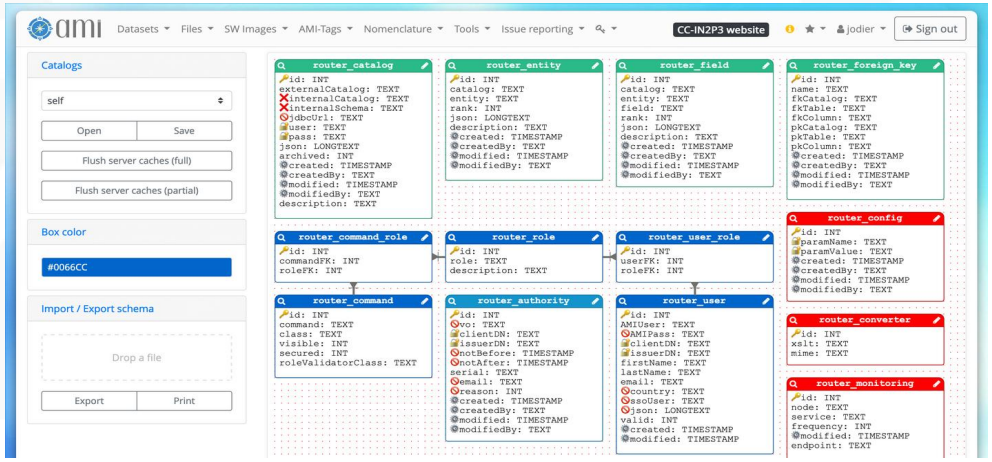
MQTT

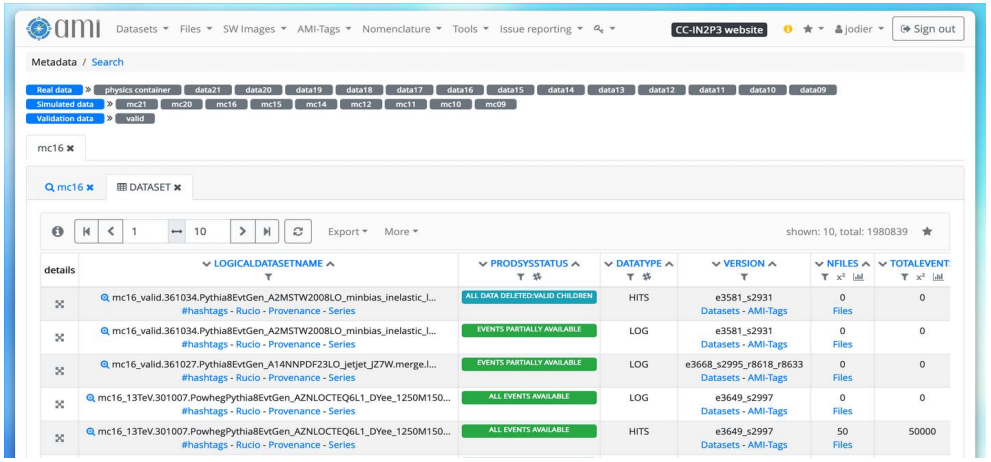
- For about 1 year, each sub-system of the AMI ecosystem can connect to an optional MQTT broker for:
  - Monitoring purpose (cpu usage, ram usage, disk usage, ...)
  - Sending AMI commands and / or providing custom AMI commands
- We provide:
  - A library to develop MQTT-based services (for exemple the AMI Task Servers, ...)
  - A standalone JavaScript client and a Vue.js 3-based micro-framework to develop Web applications
- Eclipse Mosquitto with JWT token authentication
  - https://github.com/odier-io/mosquitto-ip-jwt-auth
- Websockets with HTTPS handshake

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 (JS5 bundles with using Webpack and Babel), CSS3, HTML5
  - Bootstrap 5, AMI-Twig (MVC pattern, JS Twig template engine implementation), Vue.js 3 (MVVM pattern) natively or with the AMI MQTT micro-framework
- Nginx-based image on Docker Hub





Features and patterns

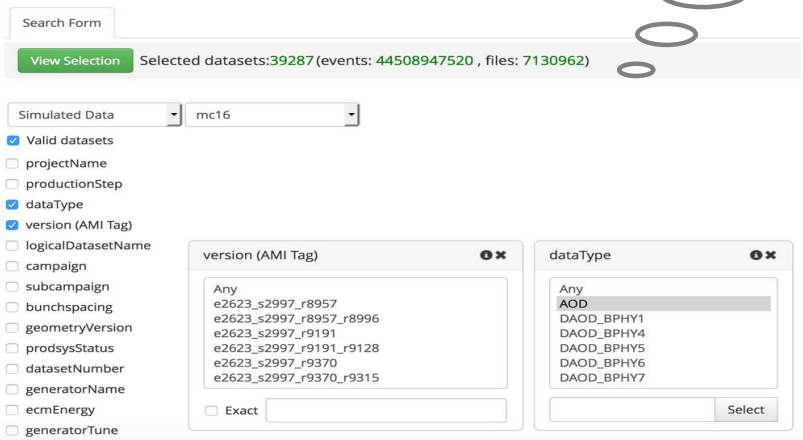
- Authentication & roles
- URL router, short URLs
- Sub-applications and reusable graphic controls (object paradigm)
- Centralized resource live cycle management (CSS, JS, JSON, xml, Twig files; AMI sub-applications; AMI controls)
- Wizards for generating sub-application and control skeletons
- Patterns: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

Default controls and applications

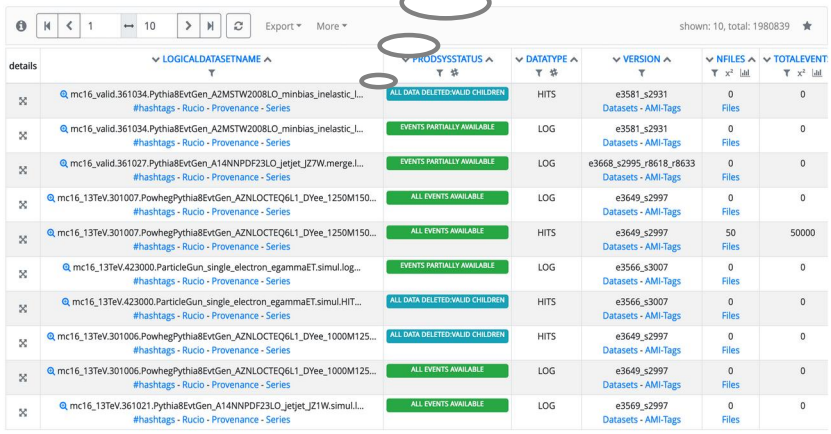
- Controls can be embedded in external Web pages such as wikis
- Applications are generally built by assembling controls
- Main available controls:
  - Dialog boxes
  - Controls for searching (Google-like Search, Criteria Search, …)
  - Controls for displaying (Schema Viewer, Tab, Table, Element Info, …)
  - 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, …

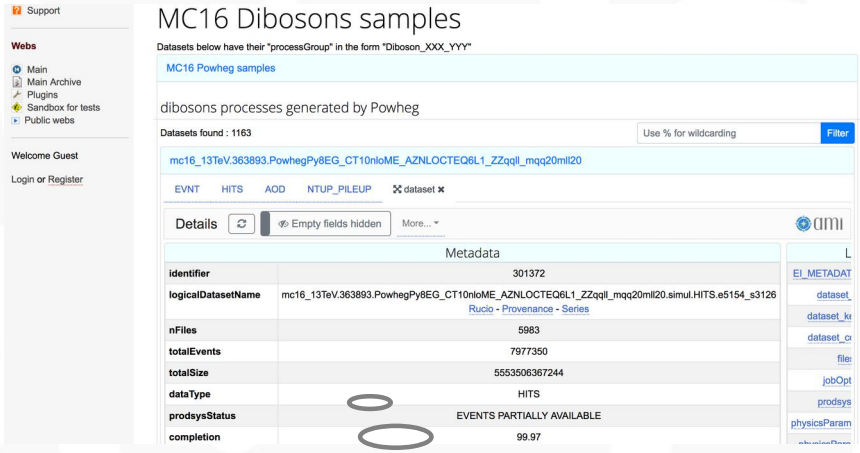
Screenshots

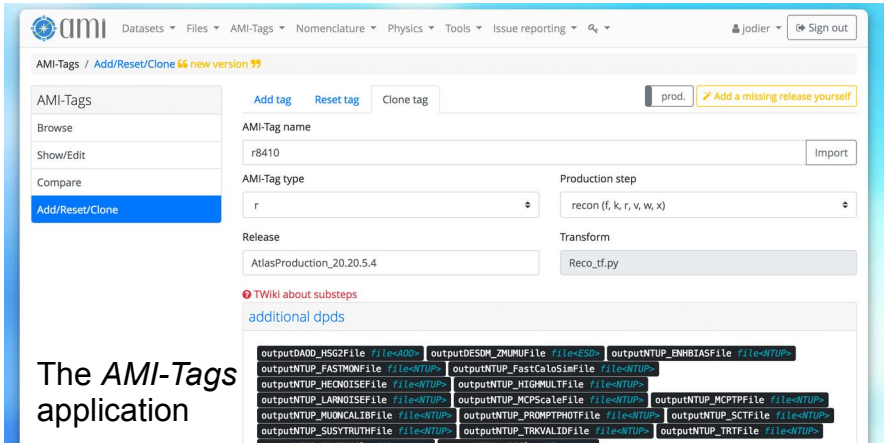
Searching ATLAS datasets by criteria



Displaying search results in AMI





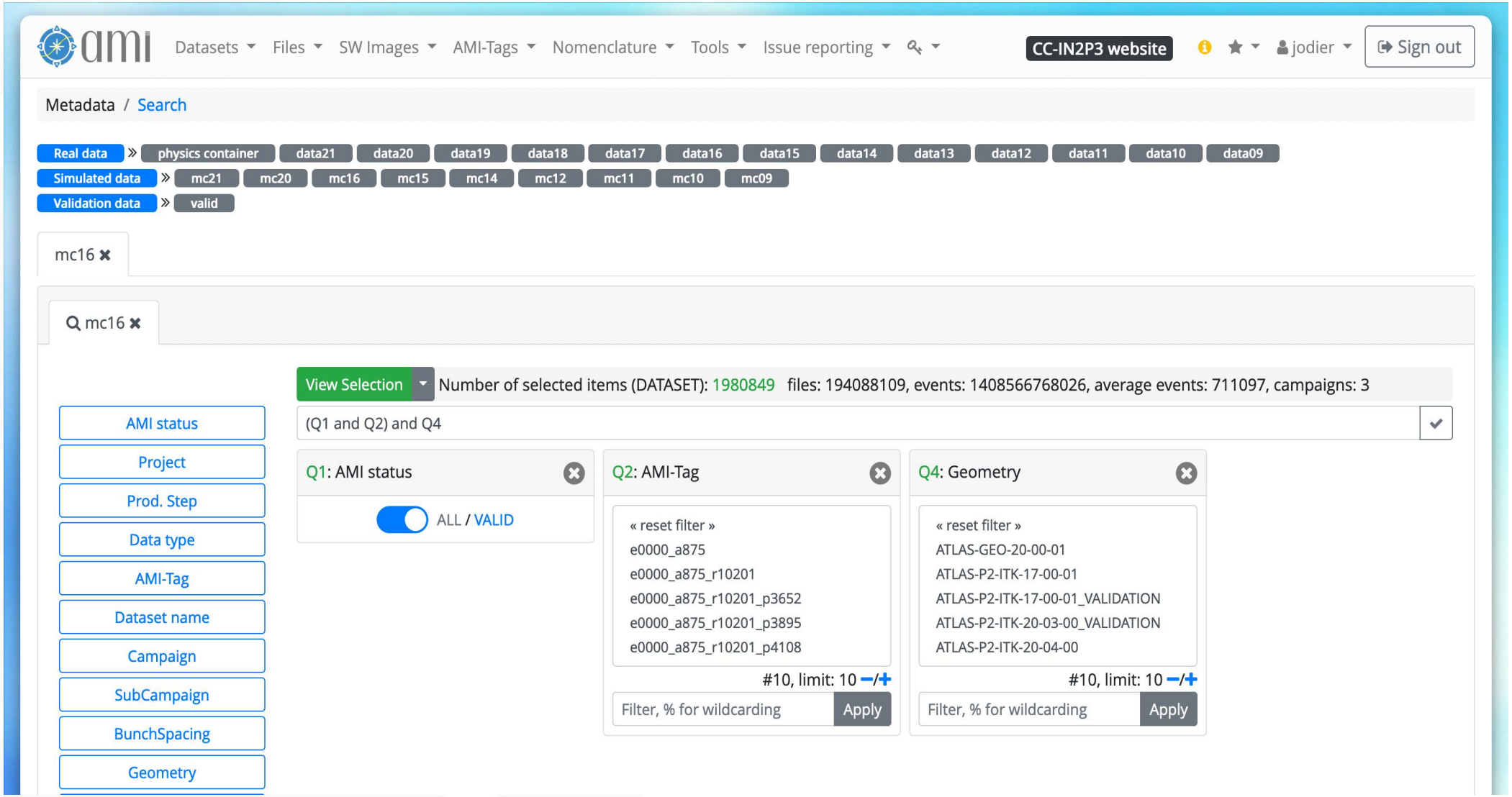


The *AMI-Tags* application

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

This control executes the **GetDatasetInfo** command

Screenshots

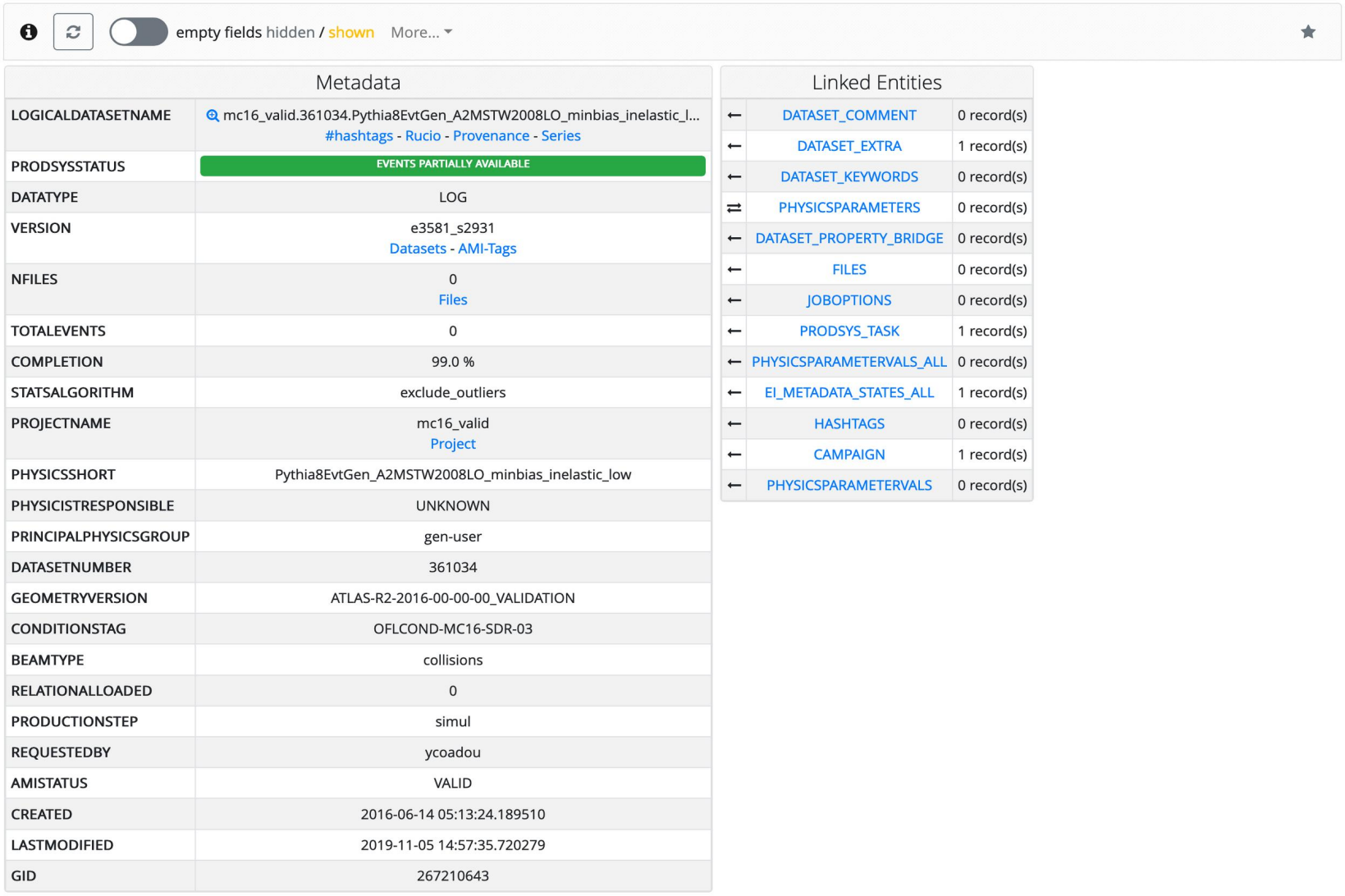


Screenshots

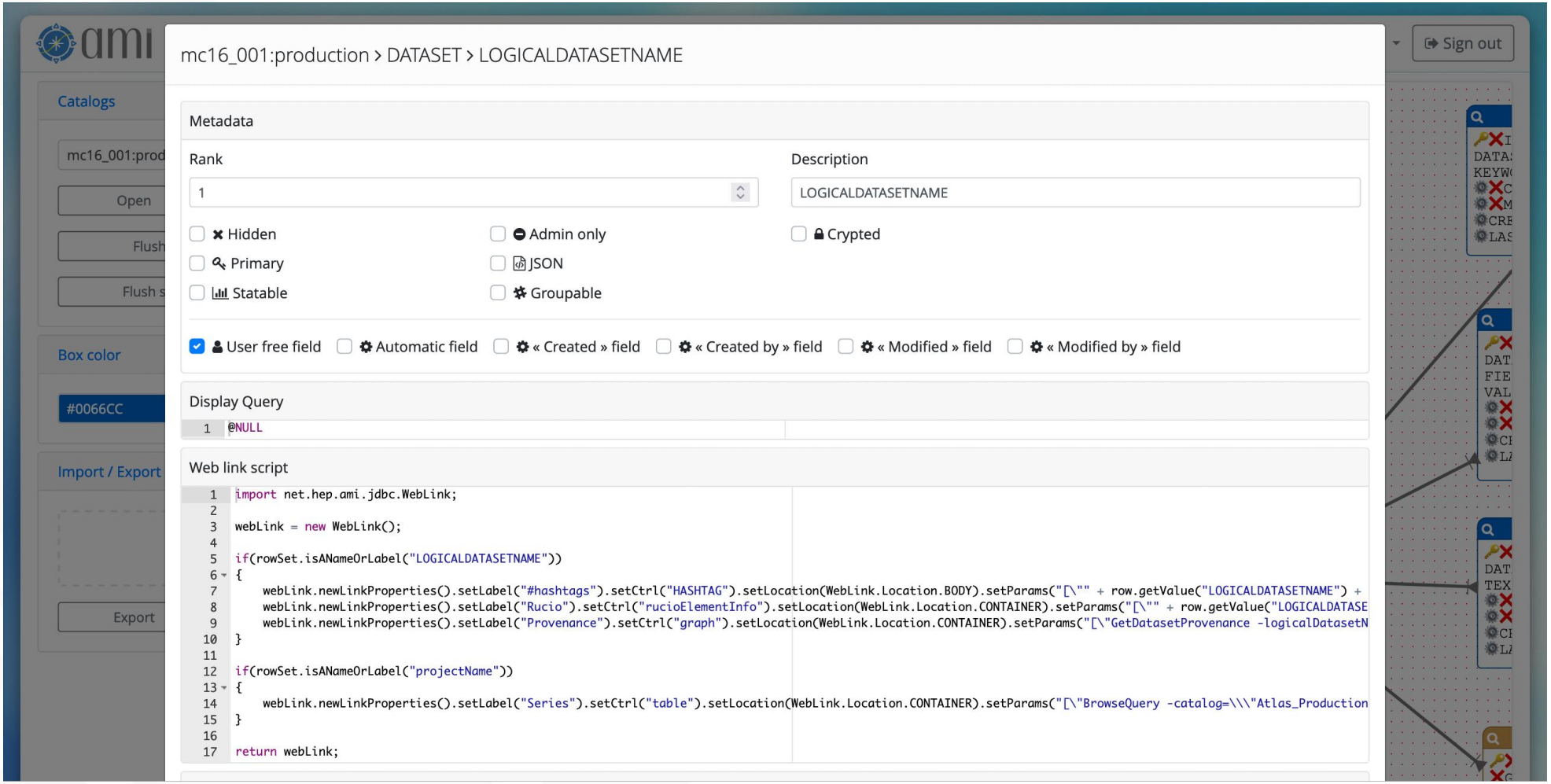
1 | 10 | Export | More | shown: 10, total: 1980839

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

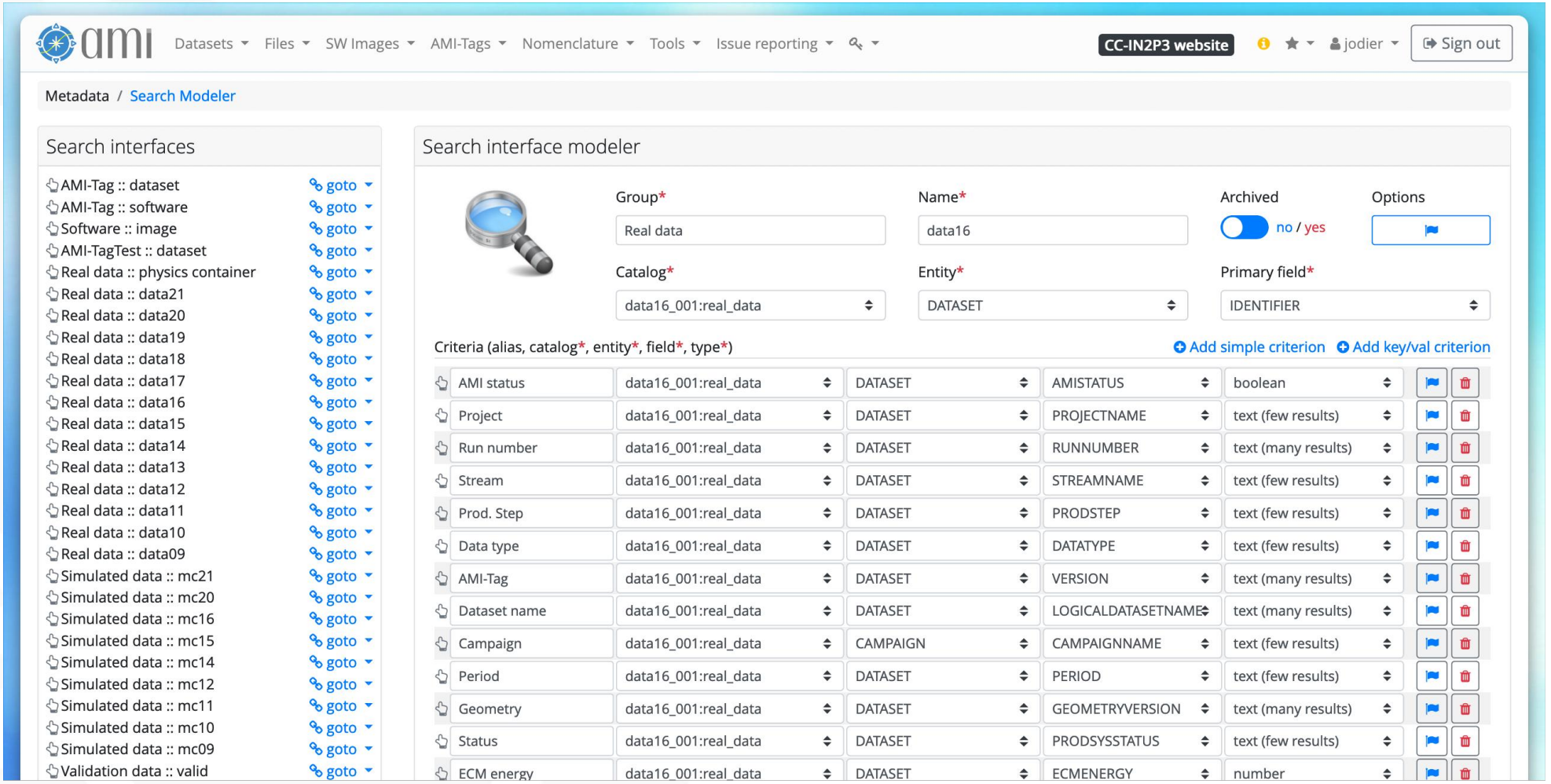
Screenshots



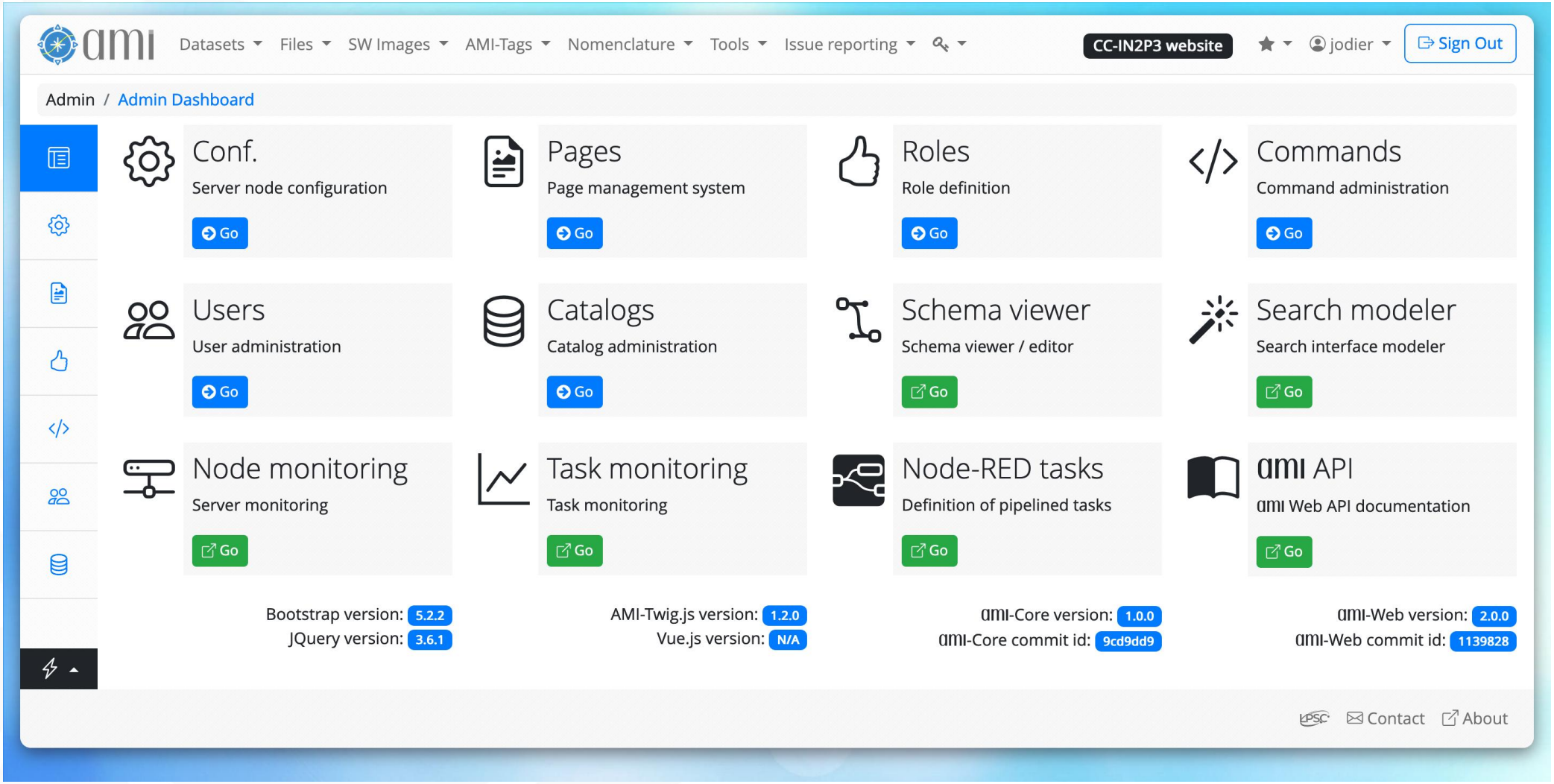
Screenshots



Screenshots



Screenshots



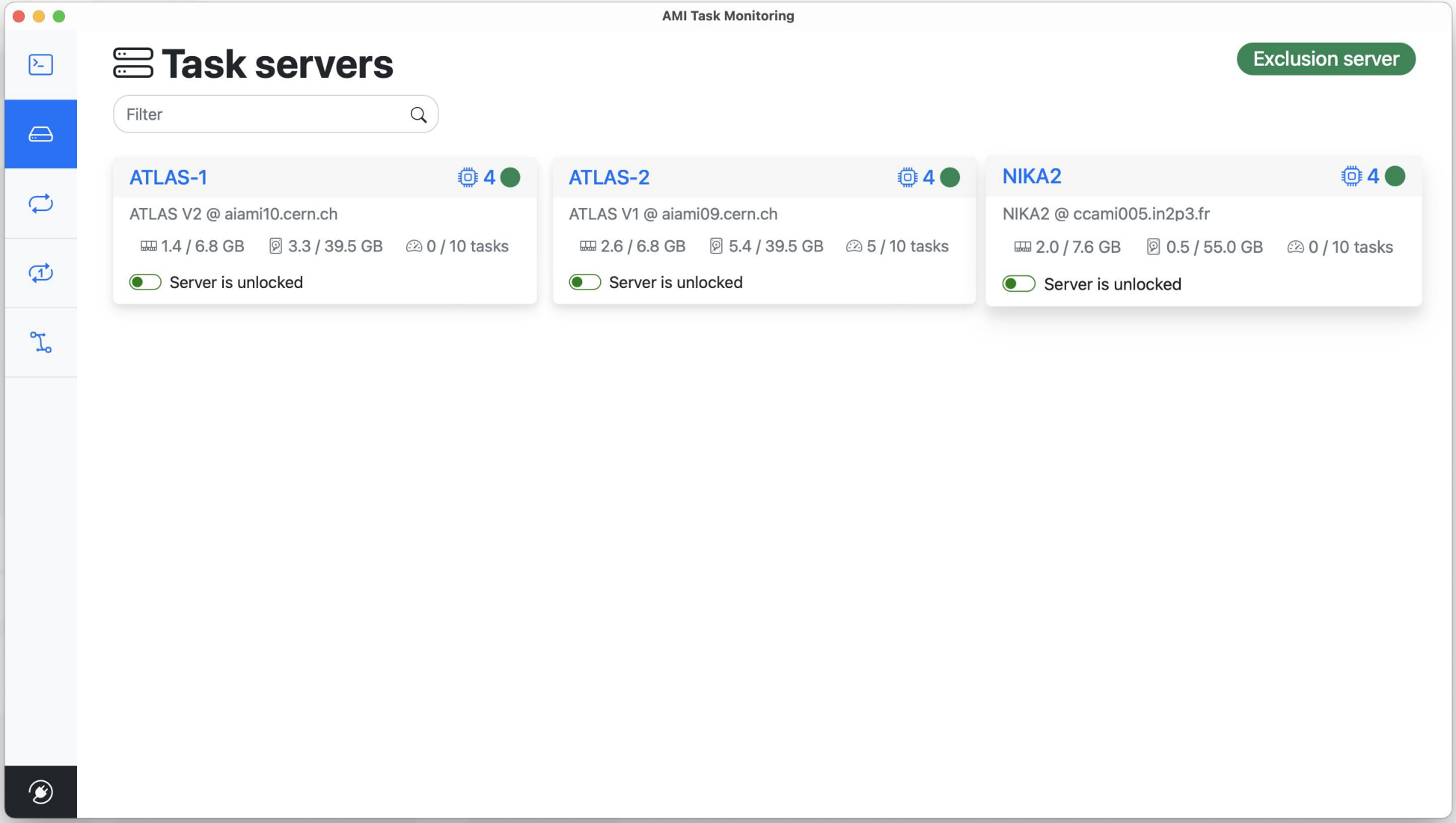
AMI Task Server and AMI Pipeline

AMI Task Server features

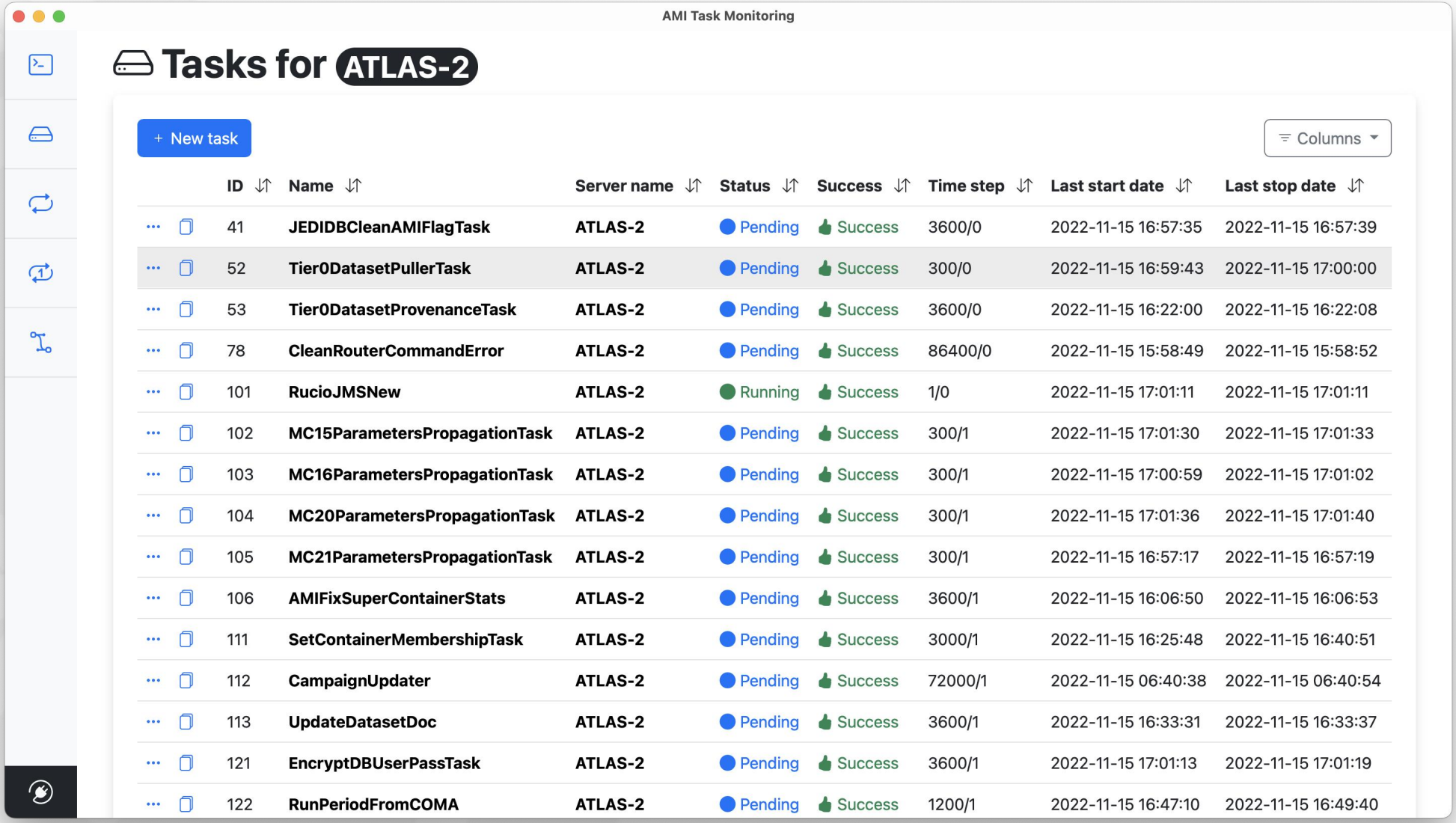
- The AMI Task Server is used for:
  - Extracting metadata from primary sources (pull mode)
  - (Re)Processing and storing metadata in AMI
- It can run any kind of tasks (shell, python, java, ...)
- Can optionally benefit from the AMI Java Core library
- Main features:
  - Kind of super CRON
  - The AMI Task Server is distributed
  - Control and monitoring 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
  - Image in Docker Hub

AMI Task Monitoring (app based on the AMI MQTT micro-framework)

AMI Task Monitoring (app based on the AMI MQTT micro-framework)

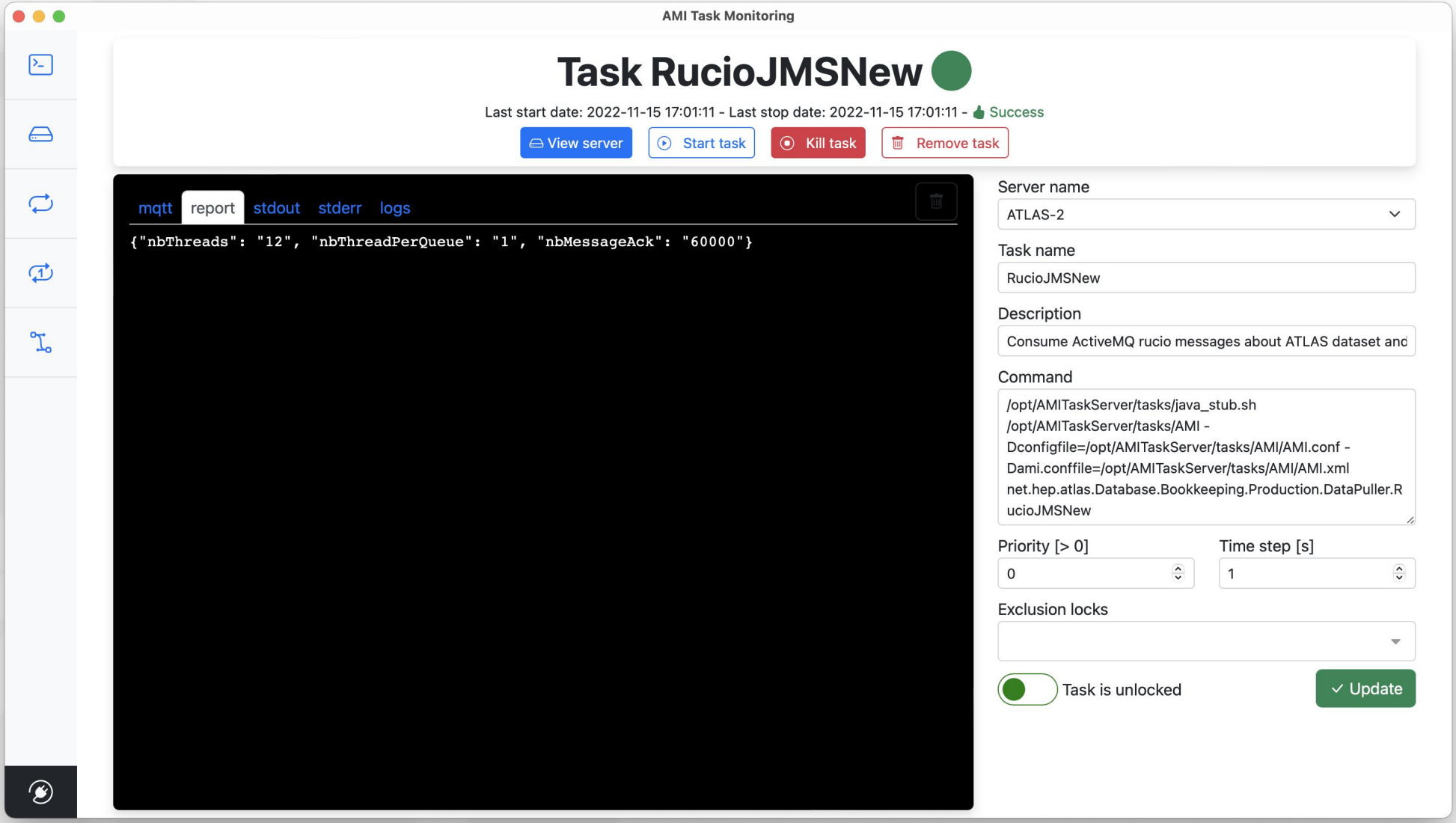


AMI Task Monitoring (app based on the AMI MQTT micro-framework)



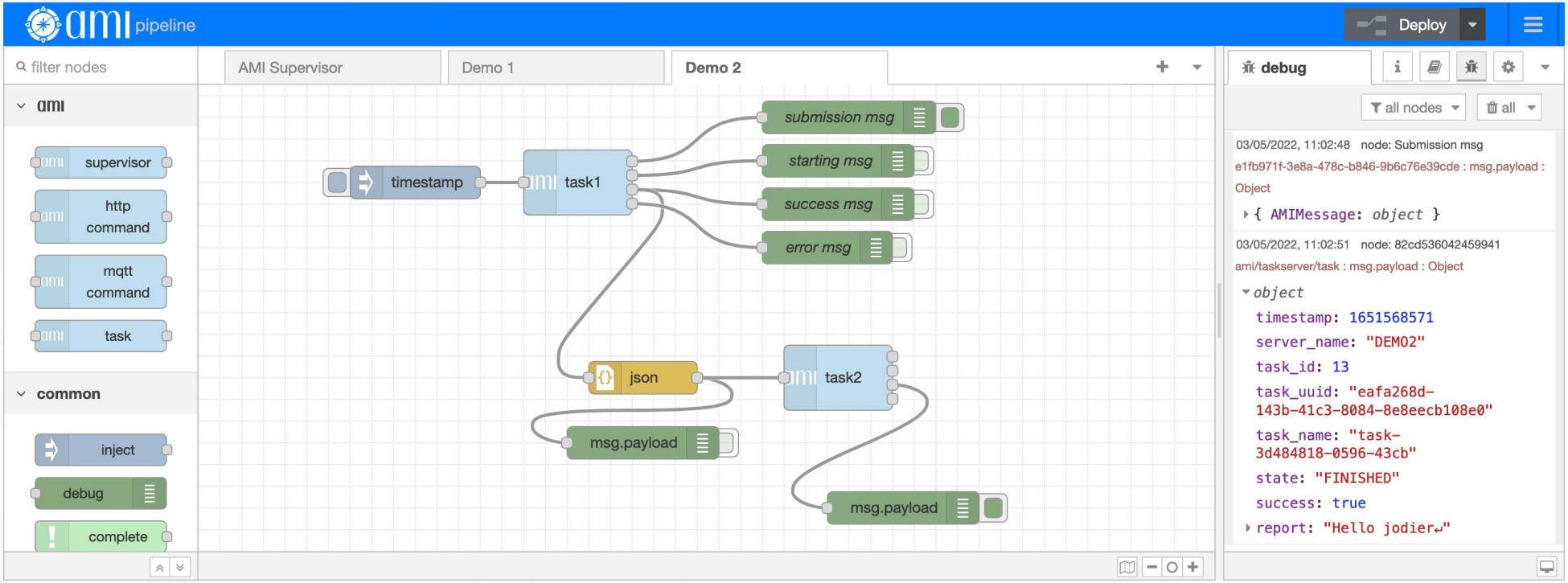
	ID	Name	Server name	Status	Success	Time step	Last start date	Last stop date
···	41	JEDIDBCleanAMIFlagTask	ATLAS-2	Pending	Success	3600/0	2022-11-15 16:57:35	2022-11-15 16:57:39
···	52	Tier0DatasetPullerTask	ATLAS-2	Pending	Success	300/0	2022-11-15 16:59:43	2022-11-15 17:00:00
···	53	Tier0DatasetProvenanceTask	ATLAS-2	Pending	Success	3600/0	2022-11-15 16:22:00	2022-11-15 16:22:08
···	78	CleanRouterCommandError	ATLAS-2	Pending	Success	86400/0	2022-11-15 15:58:49	2022-11-15 15:58:52
···	101	RucioJMSNew	ATLAS-2	Running	Success	1/0	2022-11-15 17:01:11	2022-11-15 17:01:11
···	102	MC15ParametersPropagationTask	ATLAS-2	Pending	Success	300/1	2022-11-15 17:01:30	2022-11-15 17:01:33
···	103	MC16ParametersPropagationTask	ATLAS-2	Pending	Success	300/1	2022-11-15 17:00:59	2022-11-15 17:01:02
···	104	MC20ParametersPropagationTask	ATLAS-2	Pending	Success	300/1	2022-11-15 17:01:36	2022-11-15 17:01:40
···	105	MC21ParametersPropagationTask	ATLAS-2	Pending	Success	300/1	2022-11-15 16:57:17	2022-11-15 16:57:19
···	106	AMIFixSuperContainerStats	ATLAS-2	Pending	Success	3600/1	2022-11-15 16:06:50	2022-11-15 16:06:53
···	111	SetContainerMembershipTask	ATLAS-2	Pending	Success	3000/1	2022-11-15 16:25:48	2022-11-15 16:40:51
···	112	CampaignUpdater	ATLAS-2	Pending	Success	72000/1	2022-11-15 06:40:38	2022-11-15 06:40:54
···	113	UpdateDatasetDoc	ATLAS-2	Pending	Success	3600/1	2022-11-15 16:33:31	2022-11-15 16:33:37
···	121	EncryptDBUserPassTask	ATLAS-2	Pending	Success	3600/1	2022-11-15 17:01:13	2022-11-15 17:01:19
···	122	RunPeriodFromCOMA	ATLAS-2	Pending	Success	1200/1	2022-11-15 16:47:10	2022-11-15 16:49:40

AMI Task Monitoring (app based on the AMI MQTT micro-framework)



AMI Pipeline features

- AMI provides a Node-RED-based low-code programming system for task pipeline definitions
- Image available on Docker Hub



3 commands to test the AMI ecosystem!

```
git clone https://github.com/ami-team/AMIDemo.git

cd AMIDemo

docker-compose up
```

- Runs: AMI server, AMI Task Server, AMI Task Exclusion, AMI Pipeline, Eclipse Mosquitto with our JWT authentication plugin, ...
- See more information there:
  - https://github.com/ami-team/AMIDemo

Conclusion

Conclusion

- AMI is mature metadata ecosystem of more than 20 years of existence
- 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 Web Framework
  - For developing metadata-oriented Web applications and graphic controls
- AMI Task Server
  - Distributed system for extracting, processing and storing metadata
- AMI Pipeline
  - Low-code programming for task pipeline definitions
- https://hub.docker.com/repository/docker/amiteam/

Questions?