



Stephan Egli :: Paul Scherrer Institute

SciCat: PSI-ESS-MaxIV Data Catalogue

- 1 Introduction
- 2 Data Catalog Purpose
- 3 Data Model
- 4 Architecture Goals and Rationale
- 5 Architecture Overview
- 6 Roadmap

- A collaboration between PSI, ESS and MaxIV to create a data catalog management system for scientific data
- Aims to support the management of the whole data lifecycle
- Main Developers so far: Christopher Gwilliams (now Luke Gorman will take over), Hannes Petri, Gareth Murphy, Stephan Egli
- Common Codebase with Gitflow based development workflows
- Open source (<https://github.com/scicatproject>)
- Modular design to cope with evolving requirements from scientists
- Technical architecture based on Microservices with the latest web technologies

- 1 Introduction
- 2 Data Catalog Purpose**
- 3 Data Model
- 4 Architecture Goals and Rationale
- 5 Architecture Overview
- 6 Roadmap

Enable management of the lifecycle of the data from creation , data analysis and eventual deletion

- Manage the meta data of raw and derived data which is taken at experiment facilities
- Meta data
 - administrative : data management lifecycle, ownership, filecatalog
 - scientific: describing the sample, beamline and experiment parameters relevant for the users data analysis
- Data can be linked to proposals and samples
- Data can be linked to publications (DOI, PID)
- Data can be migrated to and from longterm storage on tape
- Helps keeping track of data provenance
- Helps to check scientific integrity (checksum of data)
- Makes data findable (your own data and other peoples public data)
- In the long term:
 - help to automate standardized analysis workflows
 - support the standardization of data formats

- 1 Introduction
- 2 Data Catalog Purpose
- 3 Data Model**
- 4 Architecture Goals and Rationale
- 5 Architecture Overview
- 6 Roadmap

- Meta data is linked to **Datasets**, which are **collection of files**, e.g. all files produced during a data taking run
- Each dataset gets a globally unique persistent identifier (PID)
- Each dataset is uniquely assigned to one pgroup
- Only members of the pgroup have access to the raw data and meta data belonging to the pgroup
- Only after the embargo period (typically 3 years) the data becomes public
- The pgroup membership can be defined via processes supported by the digital user office DUO (Roles: BM, PI, MP)
- The pgroups are stored centrally in an Identity Management system
- Define common generic administrative meta data plus flexible, instrument specific, scientific meta data

- Scientific meta data is up to the beamline managers to define (in collaboration with the users)
- Aim for standardization, e.g. via use of HDF5 and Nexus formats
- The catalog per se does not pose any limits here
- See example on next page. . .

Details

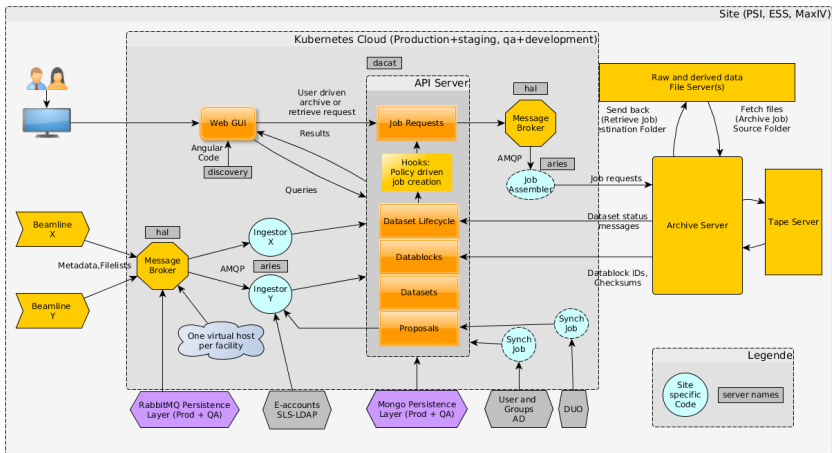
Datafiles

Principal Investigator	alessandra.patera@psi.ch	
End Time	04/04/2017 22:54	
Creation Location	/PSI/SLS/TOMCAT	
Data Format	Tomcat pre 2017	
Scientific Metadata	Name	Value
	▼ beamlineParameters	
	OP-Filter2	10um Cu
	▶ Ring current	
	OP-Filter1	100um Al
	OP-Filter3	10um Fe
	Monostripe	W/Si
	▶ Beam energy	
	FE-Filter	Filter 50%
	▼ detectorParameters	
	X-ROI Start	1
	▼ Microscope x position	
	v	-0.22486
	u	m
	▶ Exposure time	
	X-ROI End	2560
	Y-ROI End	2160
	▶ Microscope y position	
	Objective	10
	Microscope	Opt.Peter MB op
	Camera	PCO.Edge 5.5

- 1 Introduction
- 2 Data Catalog Purpose
- 3 Data Model
- 4 Architecture Goals and Rationale**
- 5 Architecture Overview
- 6 Roadmap

- Flexibility in terms of
 - covering the needs of the researchers , especially in terms of scientific meta data handling
 - integration into existing environments
 - ease of interfacing
 - from the beginning have customization to and deployments in other sites in mind
- Speed of changes
 - allow to add new instruments easily
 - allow to add new features in short time
- Longterm stability without being constrained in meeting new requirements
 - allow for constant evolution both in terms of features and volume. This concerns the whole DevOps processes
- Optionally: enable users to make customizations themselves

- 1 Introduction
- 2 Data Catalog Purpose
- 3 Data Model
- 4 Architecture Goals and Rationale
- 5 Architecture Overview**
- 6 Roadmap



- MongoDB backend
- NoSQL - Document based storage, collections, "schemaless"
- Map/Reduce/Facet Aggregation queries
- Powerful indexing and support for file storage
- Fault tolerant and drivers for most languages

mgdbaas001.psi.ch:27017

VERSION

3.4.4

Deployment

[Status](#)
[Hardware](#)
[DB Stats](#)
[Profiler](#)
[Logs](#)
[More ▾](#)

 GRANULARITY [10 seconds](#) ZOOM [10 minutes](#) CURRENT DISPLAY [6/21/2017](#) [09:51am](#) to [6/21/2017](#) [10:01am](#) AT [10 SECONDS](#) GRANULARITY

 ADD CHART DISPLAY OPCODES ON SERVERS CHARTS DISPLAY TIMELINE ANNOTATIONS

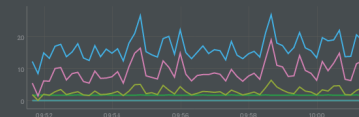
Alerts

Backup

Settings

Docs

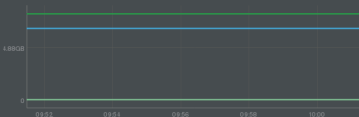
Opcounters



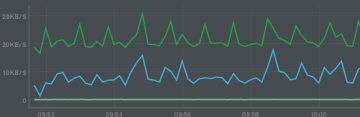
Connections



Memory



Network



TOGGLE CHARTS

+ Asserts

+ Cache Activity

+ Cache Usage

- Connections

+ Cursors

- Memory

- Network

- Opcounters

+ Operation Execution Times

+ Opllog GB/Hour

- Loopback (API creation framework based on NodeJS and Express)
- APIs auto generated from JSON configuration to configure data model
 - Can create SDKs for many target languages
- Plugins for authentication
- Business logic can be added in Javascript files
- Auto documenting
- Swagger based API creation (OpenAPI standard)
- User accounts (through central IDM system) and functional accounts with roles for different administrative activities within the system i.e. Archiving



Datafile

[Show/Hide](#)[List Operations](#)[Expand Operations](#)

Dataset

[Show/Hide](#)[List Operations](#)[Expand Operations](#)

GET

/Datasets

[Find all instances of the model matched by filter from the data source.](#)

PUT

/Datasets

[Update an existing model instance or insert a new one into the data source.](#)

POST

/Datasets

[Create a new instance of the model and persist it into the data source.](#)

Response Class (Status 200)

Model | Model Schema

```

"string"
],
"RepositoryOfRecord": "string",
"MutabilityFlag": true,
"Version": "string",
"TypeOfScienceMetaData": "string",
"ArchiveLocation": "string",
"DisposalTime": "2016-08-24",
"Exported": true,
"ExportedTo": "string",
"Published": true,

```

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<input type="text"/>	Model instance data	body	Model Model Schema <pre> { "DOI": "string", "InitialFolderLocation": "string", </pre>

- RabbitMQ queuing system
- Data published to queues using many protocols
- Data source and format does not matter
- Now alternatively also via Apache Kafka

Ingesting Data from Beamlines

RabbitMQ Management x +

Paul-Scherrer-Institut (PSI) (CH) | https://hal-qa.psi.ch/#/ Search

Phonebook General/Development... Home - Quora ingress/examples/dae... ingress/examples at m... Videos and Presentatio...

RabbitMQ Cluster: rabbit@melanie-qa-rabbitmq-3454546293-hsmth (change) User: user Log out
RabbitMQ 3.6.9, Erlang 17.4

Overview Connections Channels Exchanges Queues Admin Virtual host: All

Overview

Totals

Queued messages (chart: last minute) (?)

Ready	0
Unacked	0
Total	0

Message rates (chart: last minute) (?)

Publish	24/s
Publisher confirm	24/s
Deliver (manual ack)	0.00/s
Deliver (auto ack)	22/s
Consumer ack	0.00/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s

- Node-RED
- Javascript visual programming flow framework for the Internet of Things
- Data received from RabbitMQ and formatted according to the data model
- Dataflows publish formatted data to API server

The screenshot displays the Node-RED web interface. The main workspace shows a flow named "Flow 1" with the following nodes connected in sequence:

- Tomcat connection to message broker** (orange): Status "connected".
- Turn message into JSON object** (yellow).
- Store foldercontents and prepare xml conversion** (orange, function node 'f').
- Create Dataset Entry** (brown).
- Create Datablocks** (brown).
- Create Dataset Lifecycle data** (brown).
- msg payload** (green, output node).
- catch all** (red) and **Log exceptions** (green) are connected to the end of the flow.

The left sidebar contains a "filter nodes" search bar and two categories of nodes:

- subflows:** Create Dataset Entry, Create Datablocks, Create Archive Job, Create Dataset Lifecycle data.
- input:** inject, catch, status, link, mqtt, http, websocket.

The right sidebar shows the "debug" console with the following log output:

```

6/21/2017, 6:24:36 AM node: 7419e299.809a5
tomcat: msg.payload: Object
  > { id: "20.500.11935/1587629b-8c18-48c.", isOnDisk: true, isOnT
false, archiveStatusMessage: "Not
scheduled for archivin.",
retrieveStatusMessage: "Never
retrieved" .. }

6/21/2017, 6:24:36 AM node: 7419e299.809a5
tomcat: msg.payload: Object
  > { id: "20.500.11935/71f7f12a-1252-4a9.", isOnDisk: true, isOnT
false, archiveStatusMessage: "Not
scheduled for archivin.",
retrieveStatusMessage: "Never
retrieved" .. }

6/21/2017, 6:24:36 AM node: 7419e299.809a5
tomcat: msg.payload: Object
  > { id: "20.500.11935/f55e163e-8990-43e.", isOnDisk: true, isOnT
false, archiveStatusMessage: "Not
scheduled for archivin.",
retrieveStatusMessage: "Never
retrieved" .. }

6/21/2017, 6:24:36 AM node: 7419e299.809a5
tomcat: msg.payload: Object
  > { id:
"20.500.11935/1b34c7c5-9fb2-4af6..

```

- Built using Angular 5 (6 in preparation) and NgRx
- Created with Typescript
- Component based architecture to reuse throughout the application
- Now based on Angular Material Widget Set
- Responsive, Standards compliant and all other hip, web development words

SciCat

Search...

hydration

Location

/PSI/SLS/TOMCAT

Group

p17079

Type

Keywords

Select a date range

Clear

View

Archive

Retrieve

19 datasets.

Export Page

Items per page: 30

1 - 19 of 19

<input type="checkbox"/>	PID	Source Folder	Size	Creation Time	Type	Proposal	Group	Archive Status	Retrieve S
<input type="checkbox"/>	20.500.11935/ 1234e7b5-475	/sls/X02DA /Data10		13/02/2018 08:00	raw	unknown	p17079		
<input type="checkbox"/>	44f0-9d74-891 58e12aaab	/e17079/disk1 /SC_dry1_hydra tion_7am_/tif/							
<input type="checkbox"/>	20.500.11935/ 6928a6aa-	/sls/X02DA /Data10		13/02/2018 06:54	raw	20.500.11935/ 20171694	p17079		
<input type="checkbox"/>	23d8-40fa- bd9e-1374d7b	/e17079/disk1 /SC_dry1_hydra d3771							
<input type="checkbox"/>	20.500.11935/ 32d306b4-345	/sls/X02DA /Data10		13/02/2018 06:41	raw	20.500.11935/ 20171694	p17079		
<input type="checkbox"/>	4-46ea-8f8b- f021f5fedbf6	/e17079/disk1 /SC_dry1_hydra tion 60min /tif/							

- Different deployment options, with and without container technology are available
- Preferred method is via a Kubernetes cluster
- Each microservice is built into a docker file and saved in a registry
- Kubernetes and Helm package manager deploys the containers and handles:
 - Routing
 - Scaling
 - Server failures
 - Updates



Q Search

+ CREATE

Workloads > Deployments

Cluster

Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes

Namespace

All namespaces ▾

Workloads

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers











Stateful Sets

Discovery and Load Balancing

Ingresses

Services

Deployments 

Name ↕	Namespace	Labels	Pods	Age ↕	Images	
 tomcat-ingestor	production	name: tomcat...	1 / 1	8 hours	nodered/node-r...	⋮
 tomcat-ingestor	qa	name: tomcat...	1 / 1	8 hours	nodered/node-r...	⋮
 dacat-api	production	name: dacat...	1 / 1	8 hours	registry.psi.ch:5...	⋮
 dacat-api	qa	name: dacat...	1 / 1	8 hours	registry.psi.ch:5...	⋮
 melanie-product...	production	app: melanie-... chart: rabbit... heritage: Tiller release: mela...	1 / 1	8 hours	bitnami/rabbitm...	⋮
 melanie-qa-rabb...	qa	app: melanie-... chart: rabbit... heritage: Tiller release: mela...	1 / 1	8 hours	bitnami/rabbitm...	⋮
 kubernetes-das...	kube-system	k8s-app: kube...	1 / 1	9 hours	gcr.io/google_c...	⋮
 nginx-ingress-co...	kube-system	k8s-app: ngin...	1 / 1	6 days	gcr.io/google_c...	⋮
 default-http-bac...	kube-system	k8s-app: defa...	1 / 1	a month	gcr.io/google_c...	⋮
 tiller-deploy	kube-system	app: helm name: tiller	1 / 1	a month	gcr.io/kubernet...	⋮

- 1 Introduction
- 2 Data Catalog Purpose
- 3 Data Model
- 4 Architecture Goals and Rationale
- 5 Architecture Overview
- 6 Roadmap**

After a consolidation/refactoring phase and making experiences in connecting first instruments during the last months we now focus on

- Finalizing support for attachments
- Adding support for automated policy decisions (e.fg. which data to archive): Autumn 2018
- Adding support for the publication workflow (target date end of 2018)
- Adding extended HDF5 support
- Adding aggregation/statistical overview plots for an overview of current data volumes and expected growth

This is non exhaustive list. We want to react on feedback from the users, this may change the order of features to be added as well as add new so far unforeseen feature requests.