PAUL SCHERRER INSTITUT

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

What is SciCat?

- 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

Data Catalogue Purpose

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

Topic

1 Introduction

2 Data Catalog Purpose

3 Data Model

4 Architecture Goals and Rationale

5 Architecture Overview

6 Roadmap

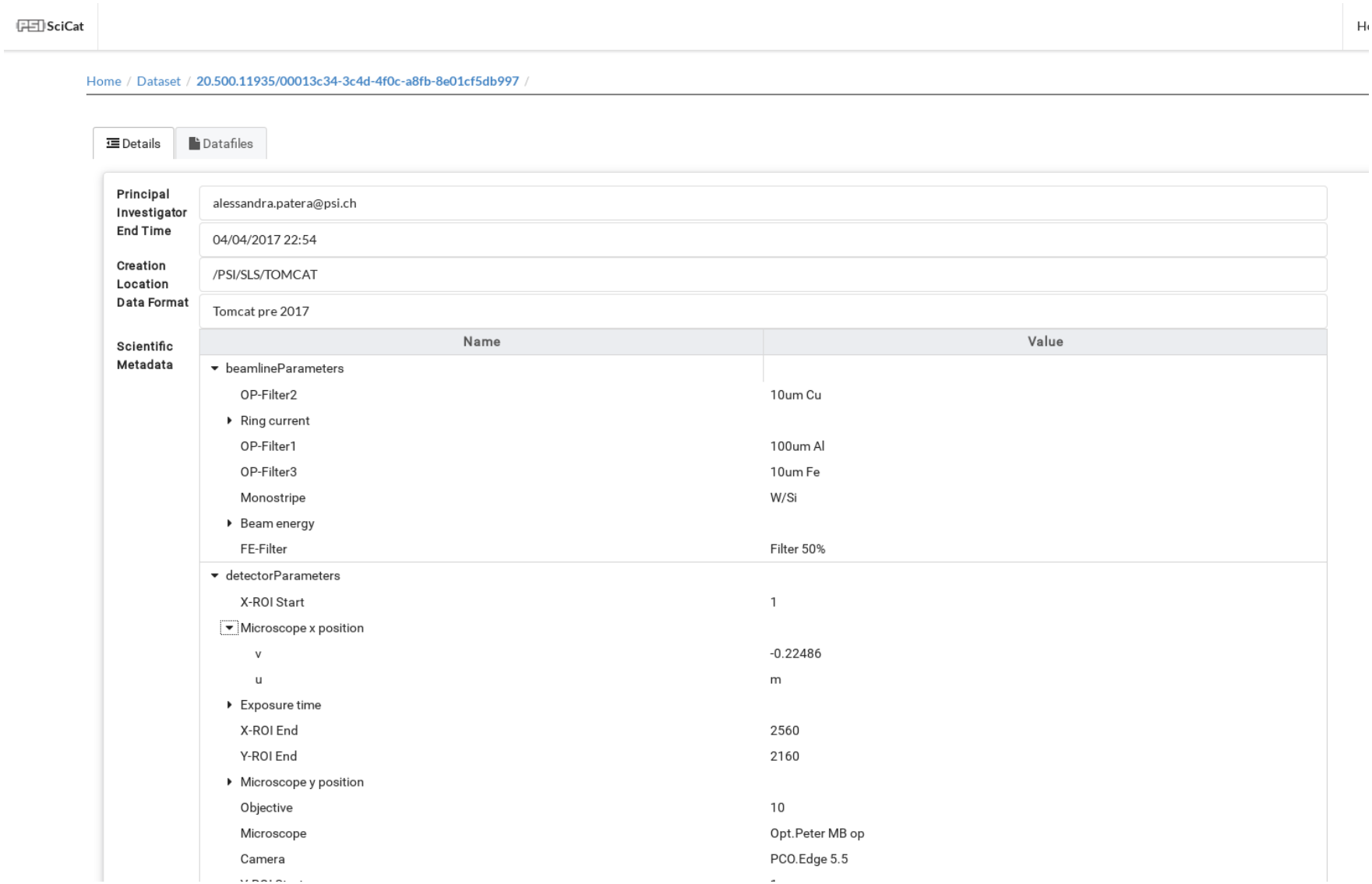
Dataset Concept and Access Policy

- 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

- 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. . .

Example of Scientific Meta Data



1 Introduction

2 Data Catalog Purpose

3 Data Model

4 Architecture Goals and Rationale

5 Architecture Overview

6 Roadmap

(Non-functional) Goals and Challenges

- 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

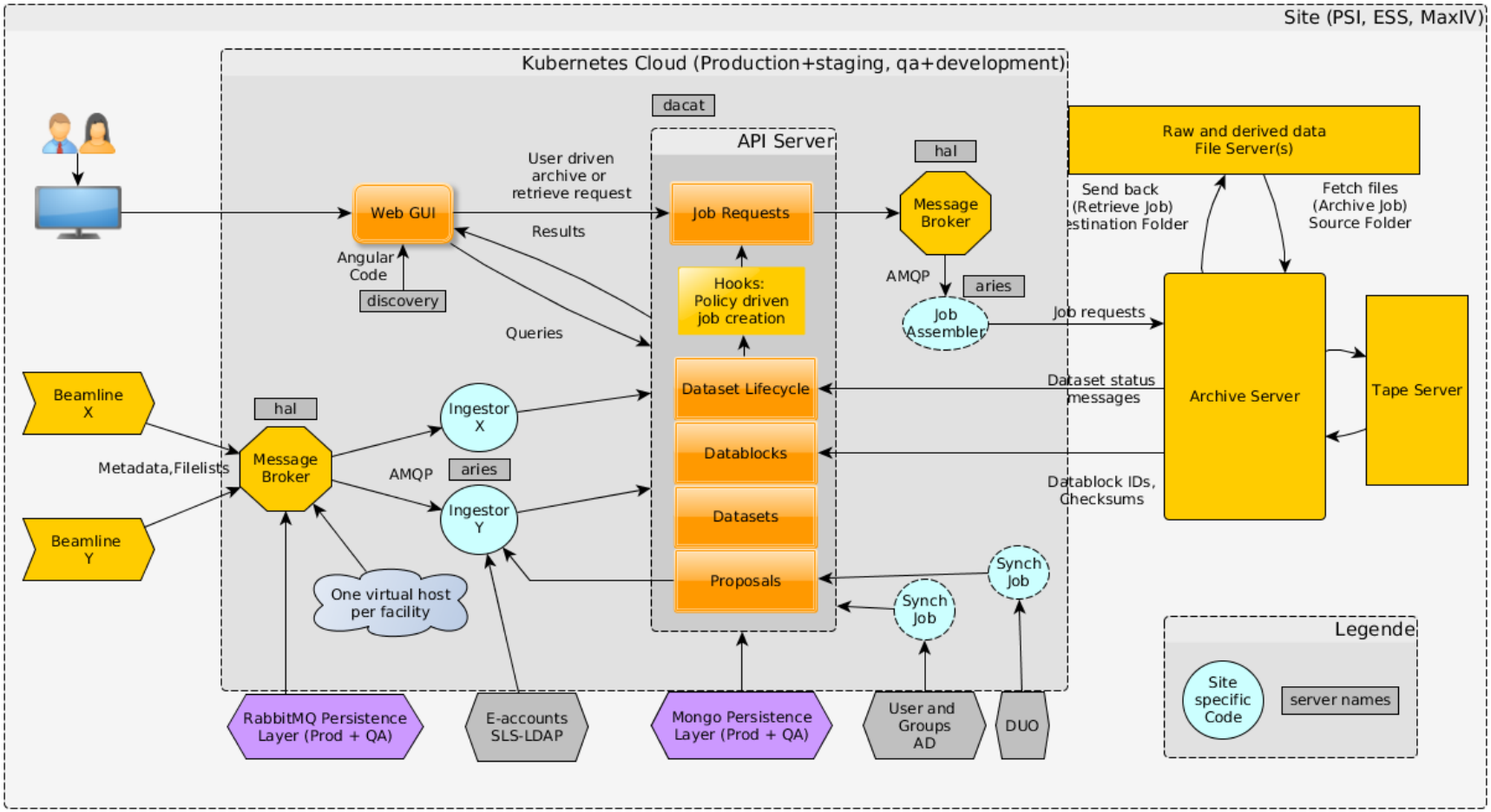
3 Data Model

4 Architecture Goals and Rationale

5 Architecture Overview

6 Roadmap

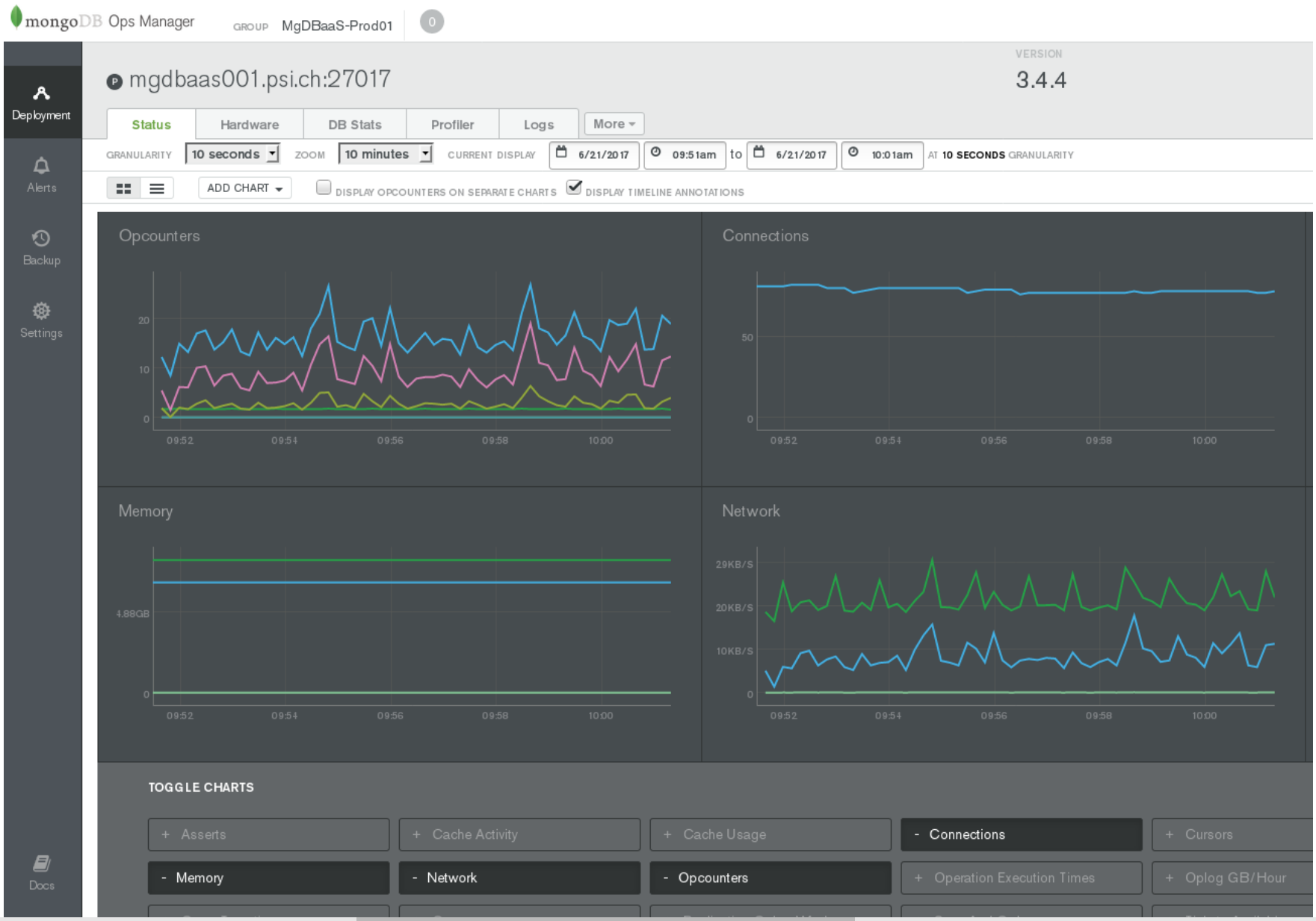
Architecture Graph and Data Flows



Datastore

- 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

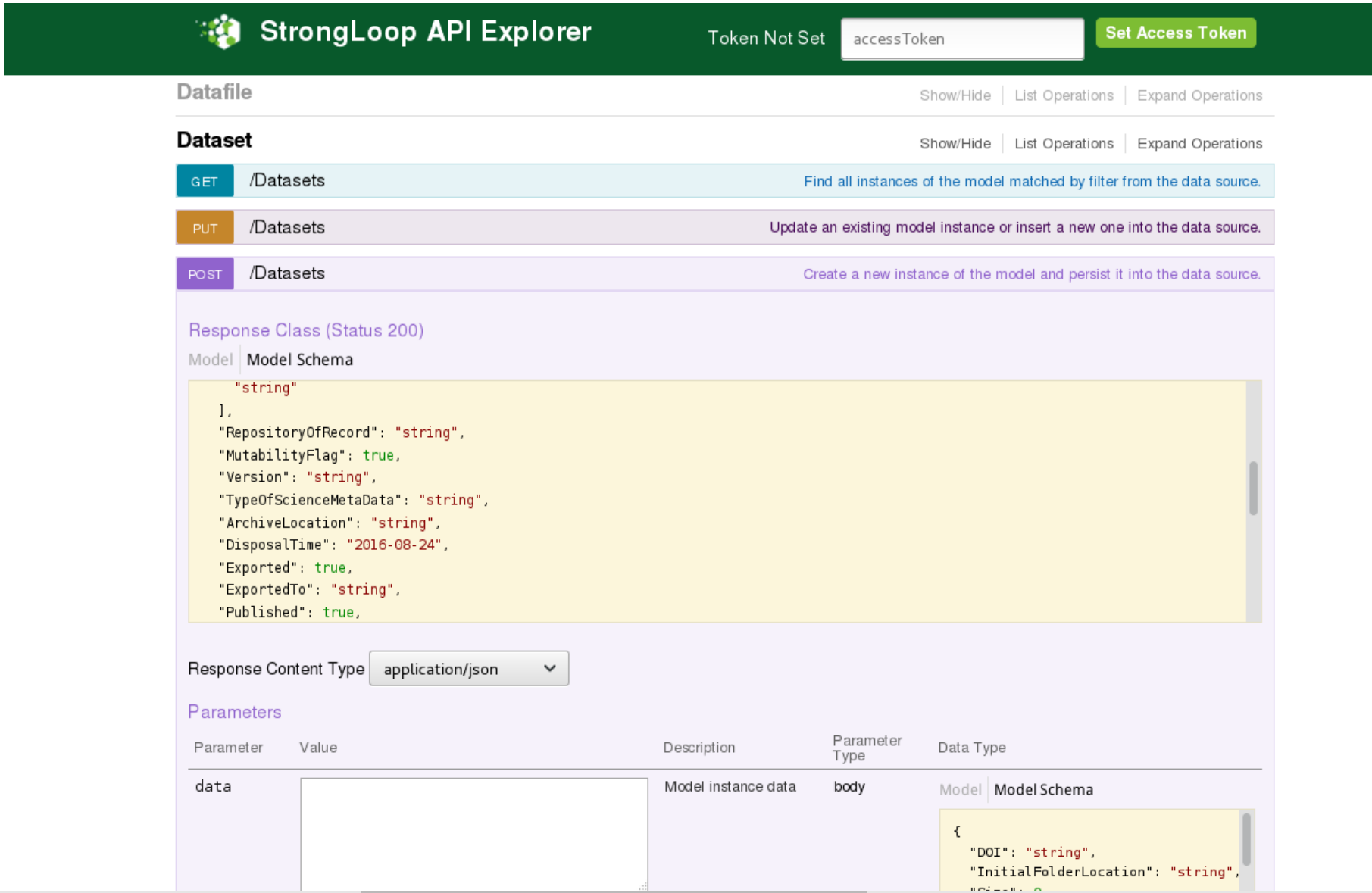
MongoDB Ops Manager



API Server

- 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

Screenshot of API Explorer

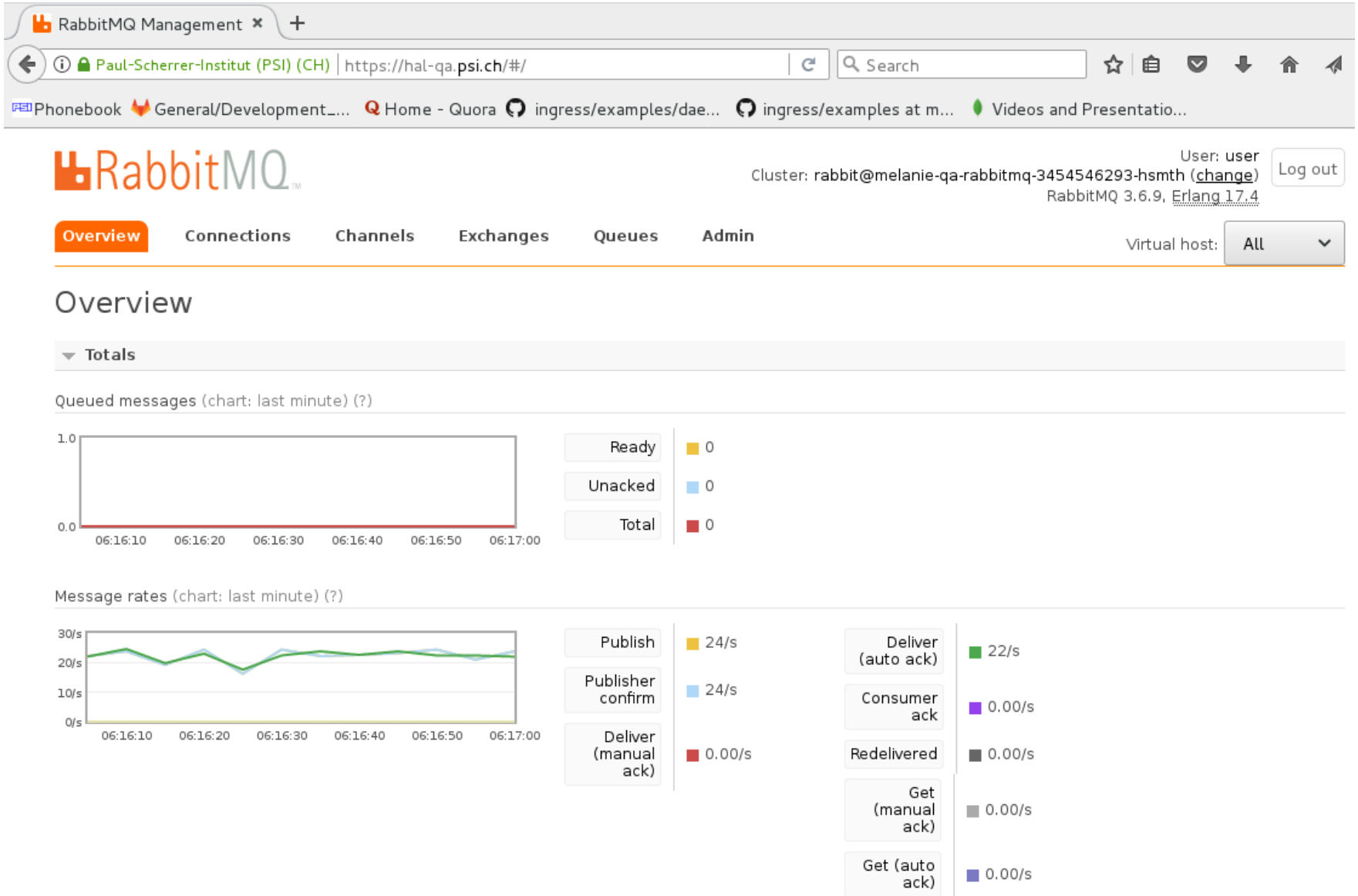


Data Ingestor

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

Message broker RabbitMQ Managament

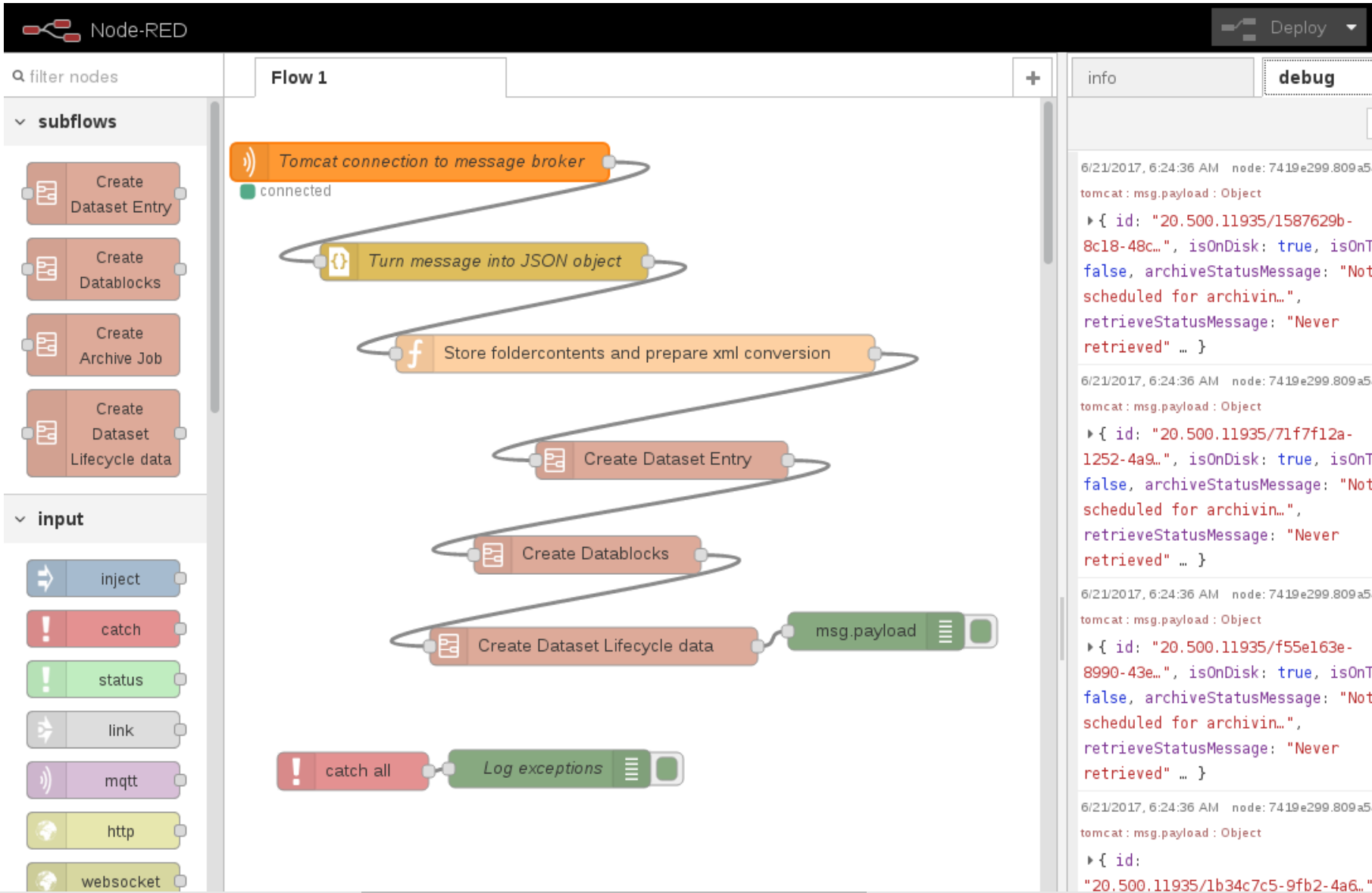
Ingesting Data from Beamlines



Data handling

- 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

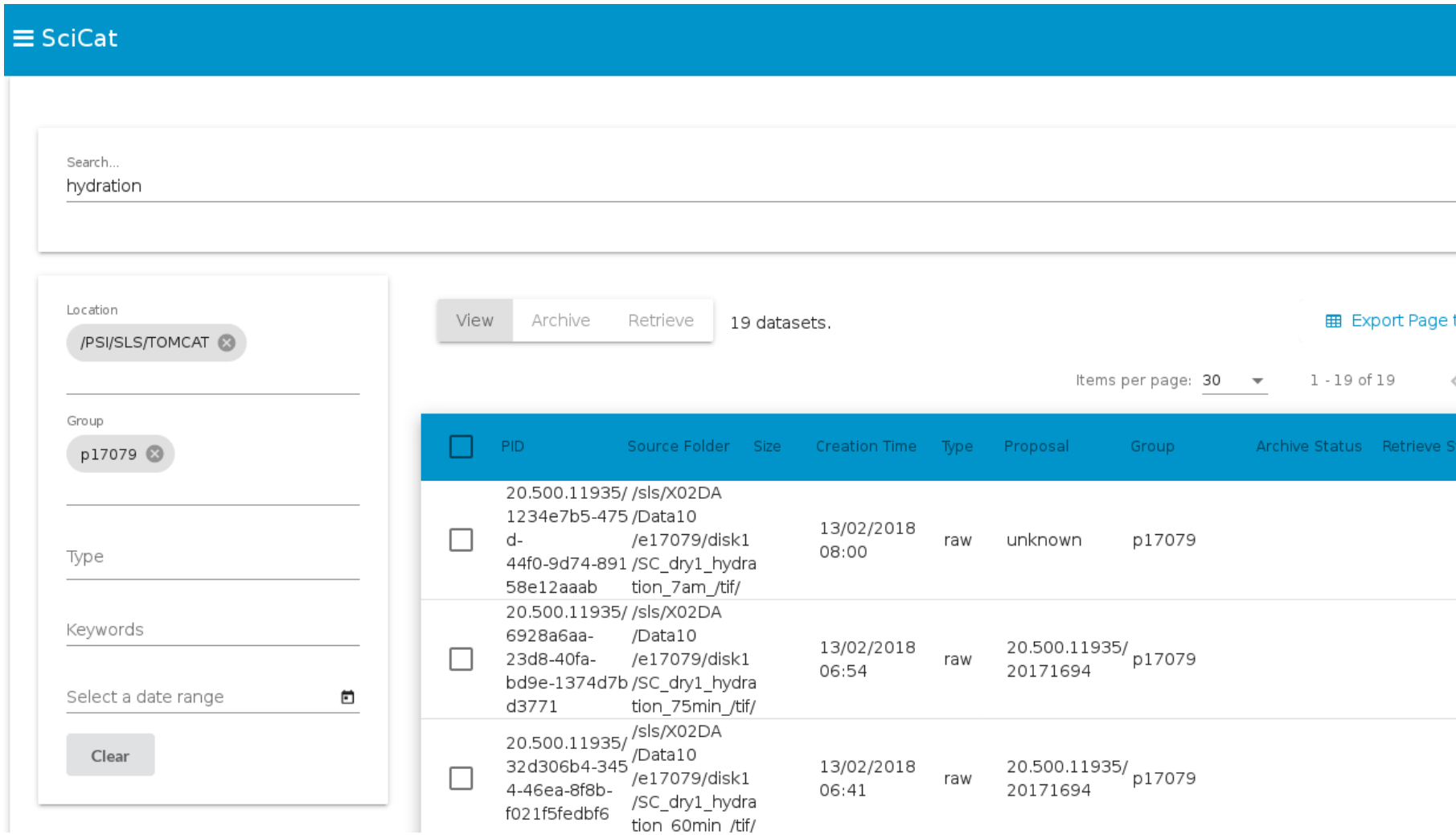
Node-Red Dataflow based coding



Frontend

- 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

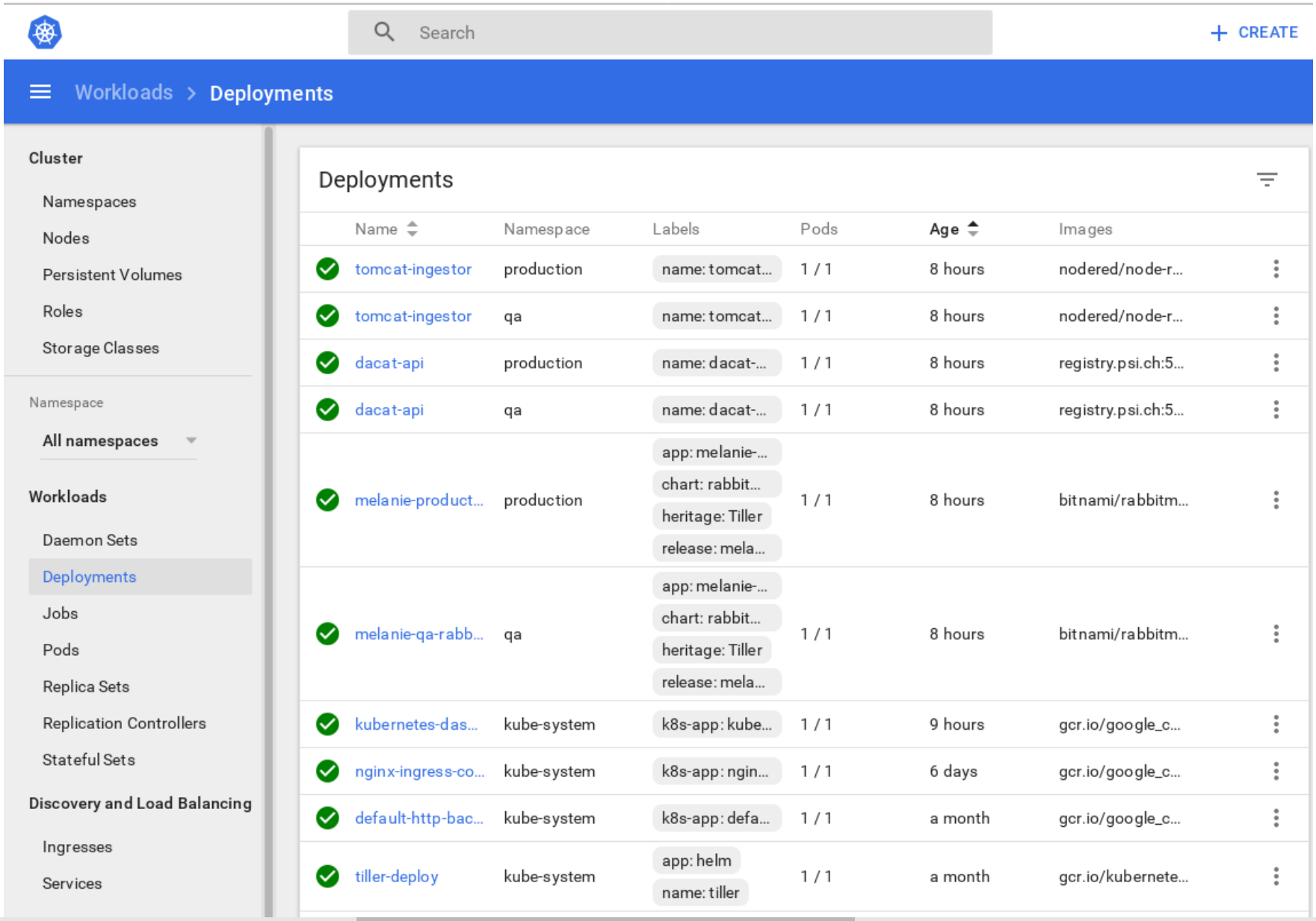
Frontend


SciCat
Search...
hydration
Location
/PSI/SLS/TOMCAT
Group
p17079
Type
Keywords
Select a date range
Clear
View
Archive
Retrieve
19 datasets.
Export Page t
Items per page: 30
1 - 19 of 19
PID
Source Folder
Size
Creation Time
Type
Proposal
Group
Archive Status
Retrieve S
20.500.11935/1234e7b5-475d-44f0-9d74-89158e12aaab
/sls/X02DA/Data10/e17079/disk1/SC_dry1_hydration_7am_/tif/
13/02/2018 08:00
raw
unknown
p17079
20.500.11935/6928a6aa-23d8-40fa-bd9e-1374d7bd3771
/sls/X02DA/Data10/e17079/disk1/SC_dry1_hydration_75min_/tif/
13/02/2018 06:54
raw
20.500.11935/20171694
p17079
20.500.11935/32d306b4-3454-46ea-8f8b-f021f5fedbf6
/sls/X02DA/Data10/e17079/disk1/SC_dry1_hydration_60min_/tif/
13/02/2018 06:41
raw
20.500.11935/20171694
p17079

Management and deployment

- 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

Kubernetes Management Dashboard



Topic

1 Introduction

2 Data Catalog Purpose

3 Data Model

4 Architecture Goals and Rationale

5 Architecture Overview

6 Roadmap

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.