PAUL SCHERRER INSTITUT

Christopher Gwilliams and Stephan Egli :: Paul Scherrer Institute

SciCat Project: Data Catalog System

PAUL SCHERRER INSTITUT

What is SciCat?

- A collaboration between PSI and ESS (and MaxIV) to create a data catalog management system
- Supporting the management of the whole data lifecycle
- Open source (https://github.com/scicatproject)
- Built with the requirements of scientists in the early stages
- Microservice architecture with the latest technologies

Topic

Data Catalogue Purpose

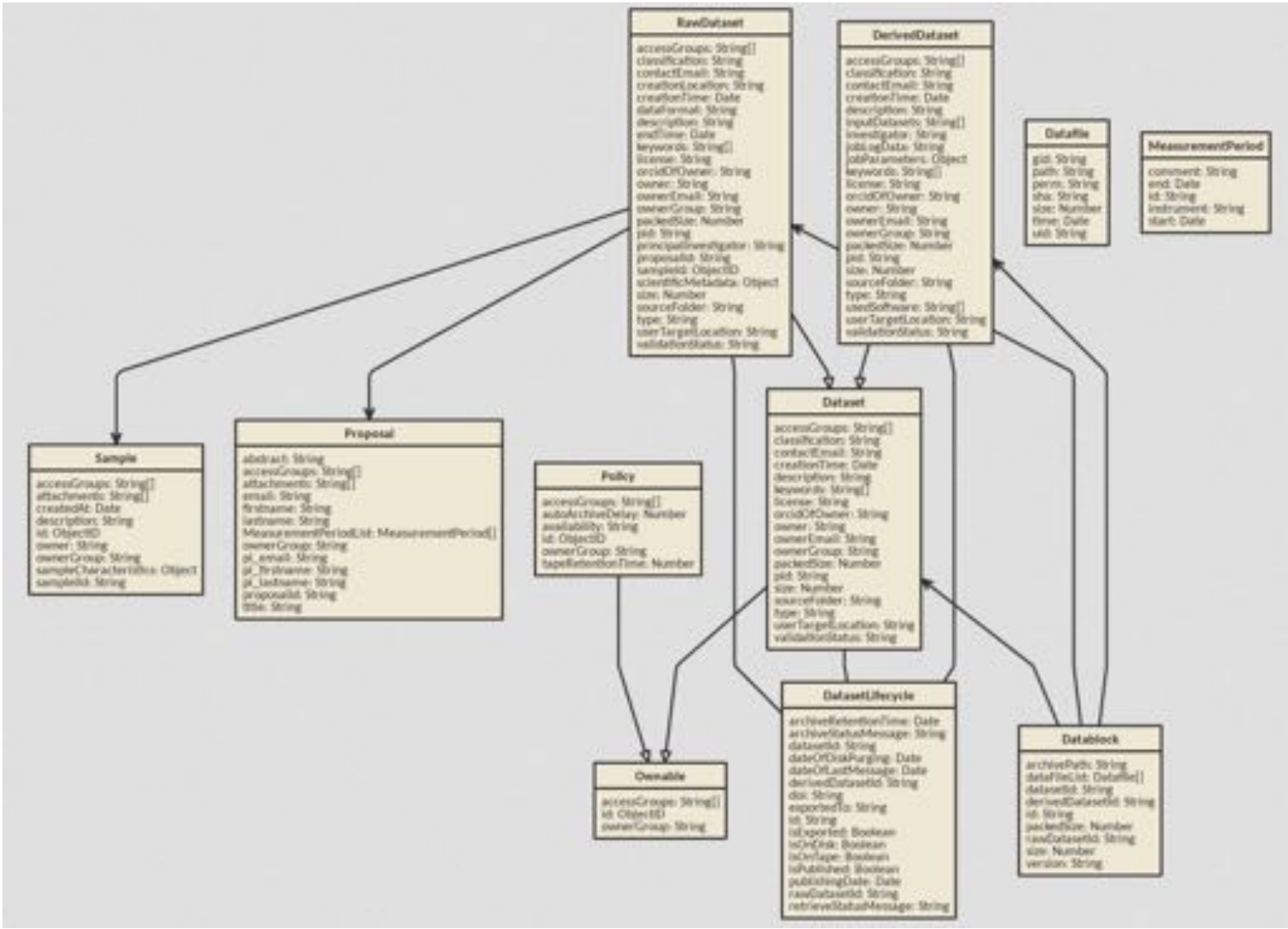
- Manage the meta data of raw and derived data which is taken at experiment facilities
- Meta data
  - administrative : data management lifecycle, ownership, filecatalog
  - scientific: describing the sample, beamline and experiment parameters relevant for the users data analysis
- Enables management of the lifecycle of the data from creation , data analysis and eventual deletion
  - 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 (i.e. the steps leading to the final results)
- Allows to check scientific integrity (checksum of data)
- Allows to find data based on the meta data (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

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 the AD Identity Management system
- Data model: Define common generic (fixed) meta data and allow at the same time completely flexible and rich structured (beamline, instrument specific) scientific meta data

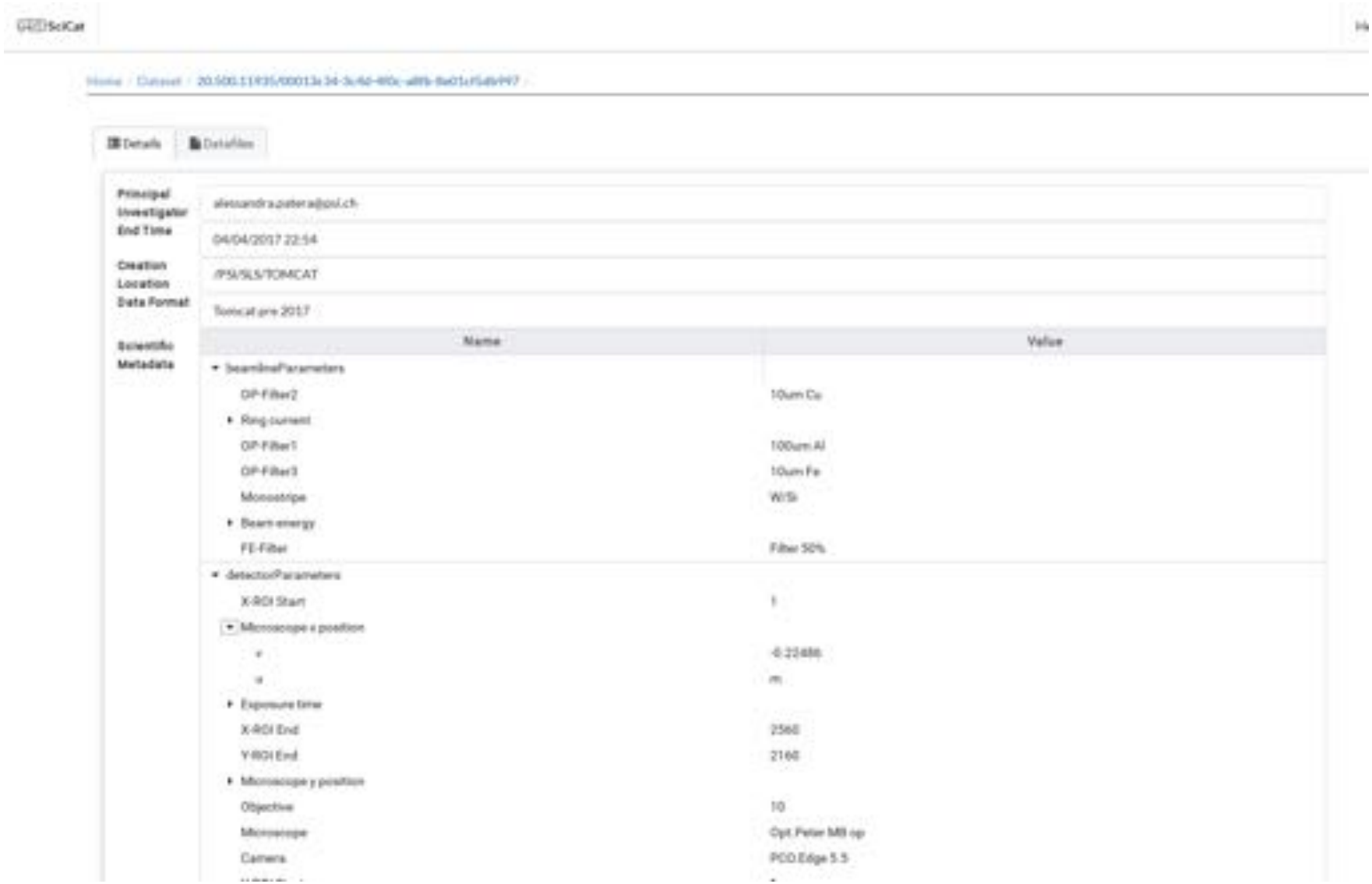
Administrative data model



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

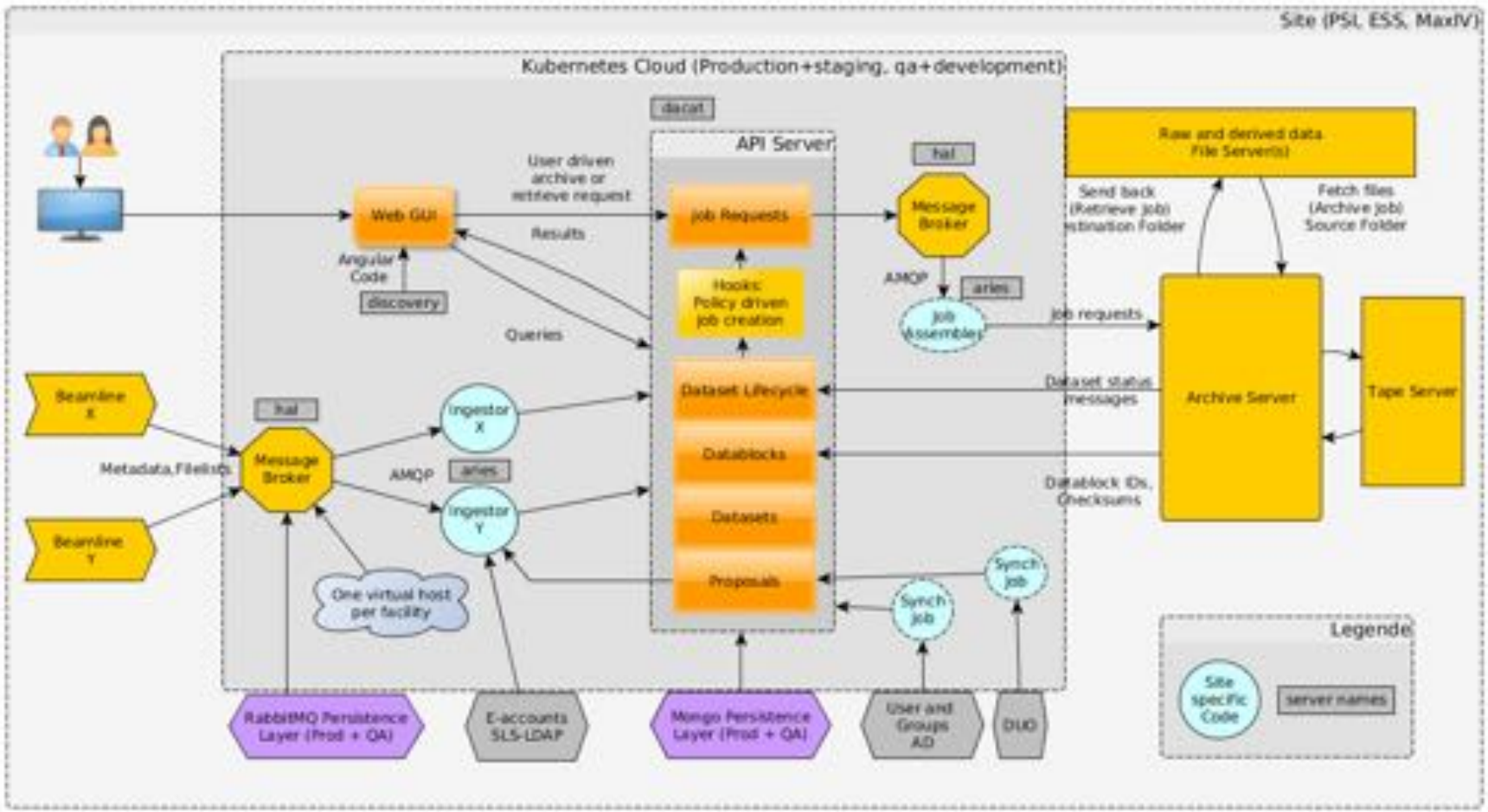


Topic

(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

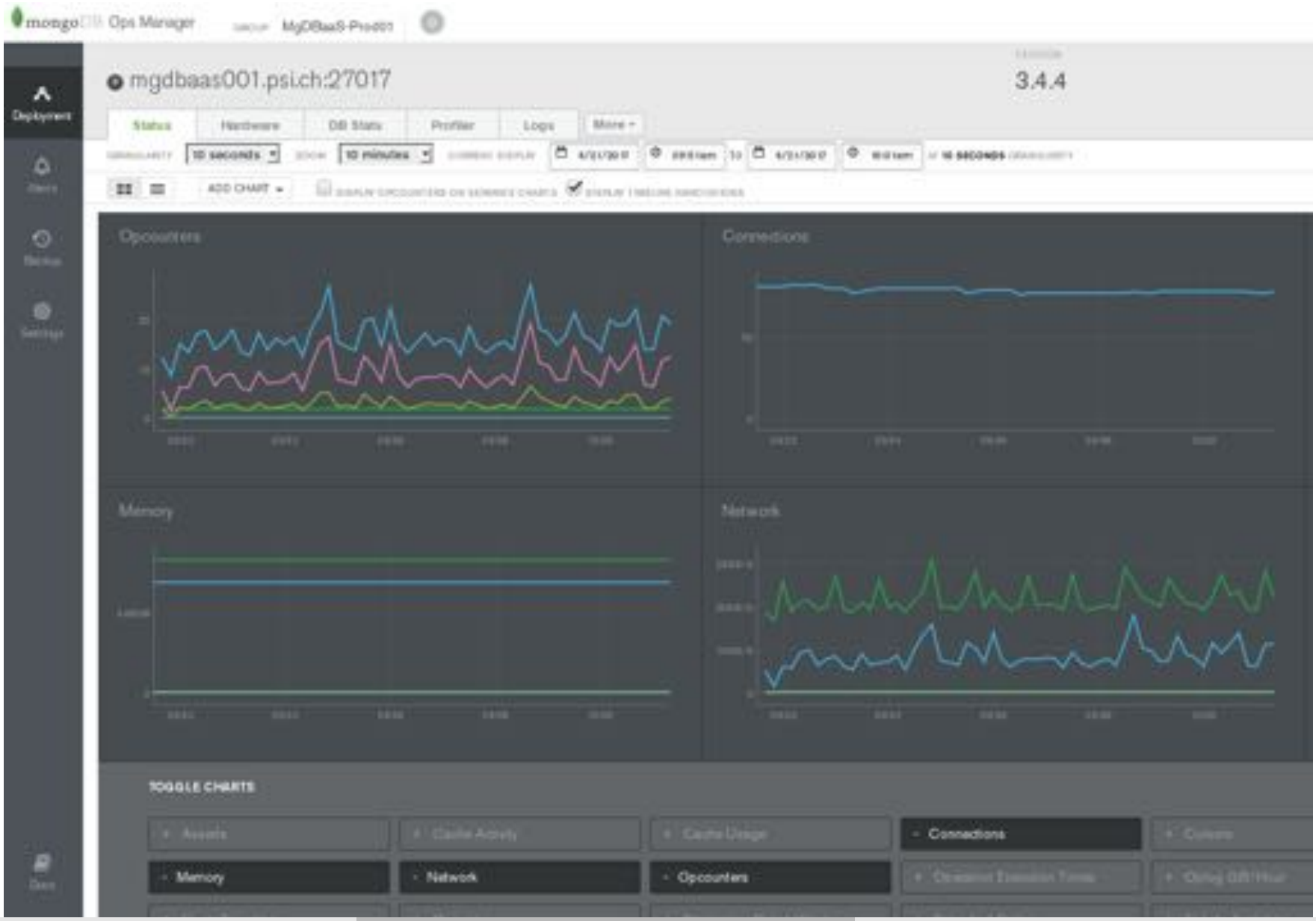
Architecture Graph and Data Flows



Datastore

- MongoDB backend
- NoSQL - no fixed structure and common data format (kind of)
- Map/Reduce queries and the option to use Javascript as the query language
- Powerful indexing and support for file storage
- Fault tolerant and drivers for most languages

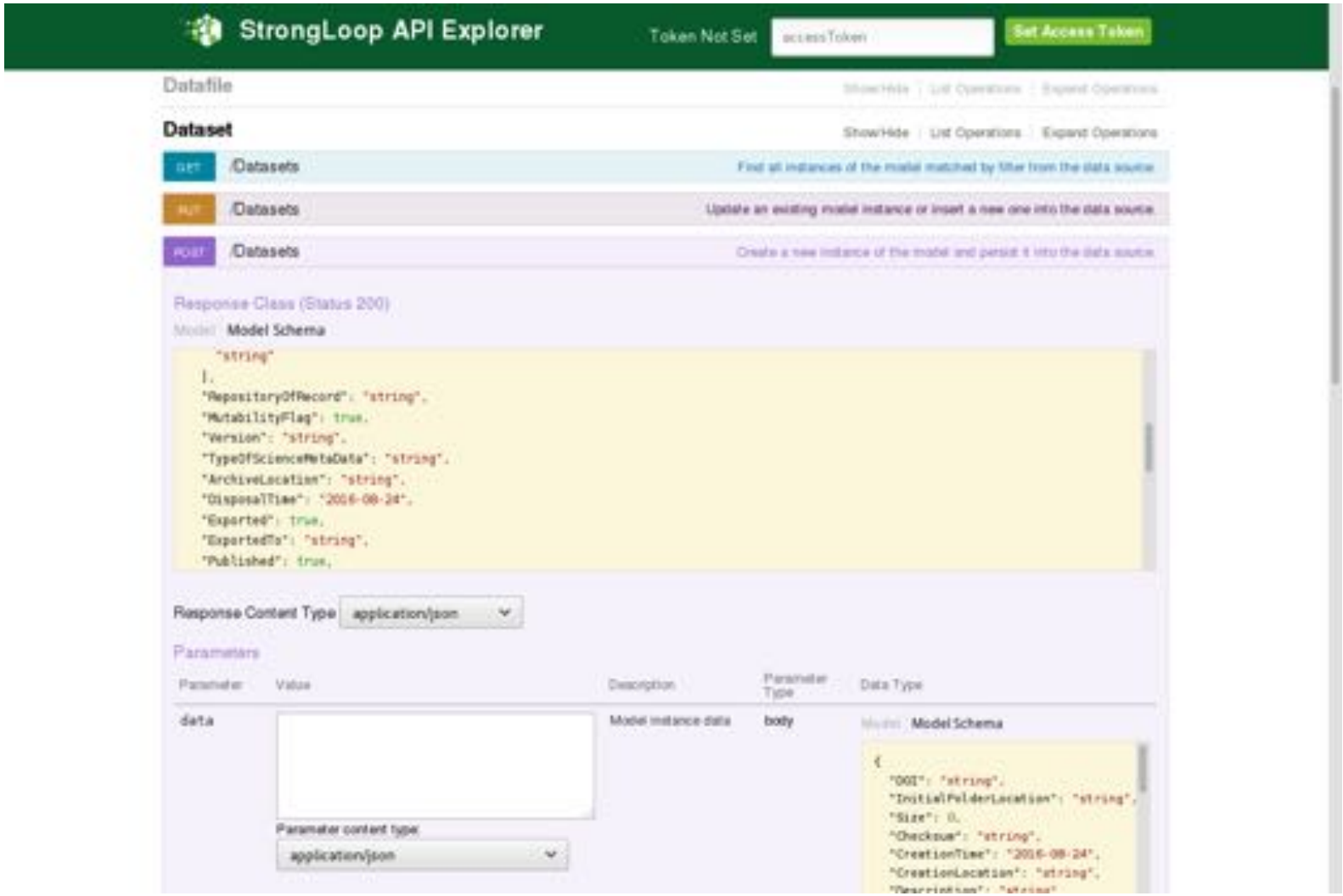
MongoDB


mongoDB Ops Manager
mgdbaas001.psi.ch:27017
3.4.4
Status
Hardware
DB Stats
Profiler
Logs
More
ADD CHART
Opcounters
Connections
Memory
Network
TOGGLE CHARTS
Connections
Memory
Network
Opcounters

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 and backend agnostic
- Logic can be added in Javascript files
- Auto documenting
- Swagger based API creation (open format)
- User accounts (through Active Directory) and System accounts (with roles for different administrative activities within the system i.e. Archiving)
- Many authorisation methods available through NPM packages

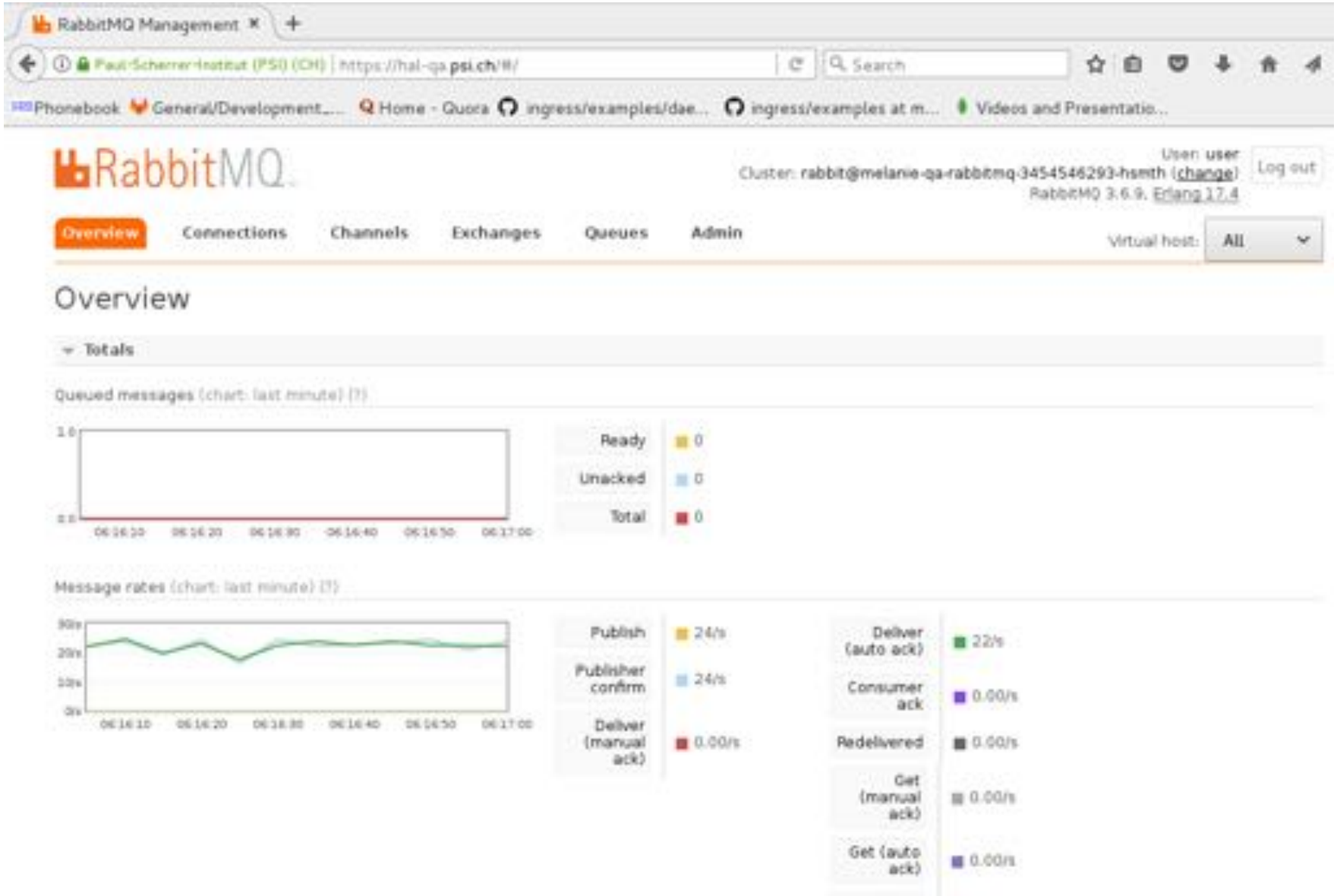
Screenshot of API Explorer



Data Ingestor

- RabbitMQ queuing system
- Data published to queues using many protocols
- Data source and format does not matter

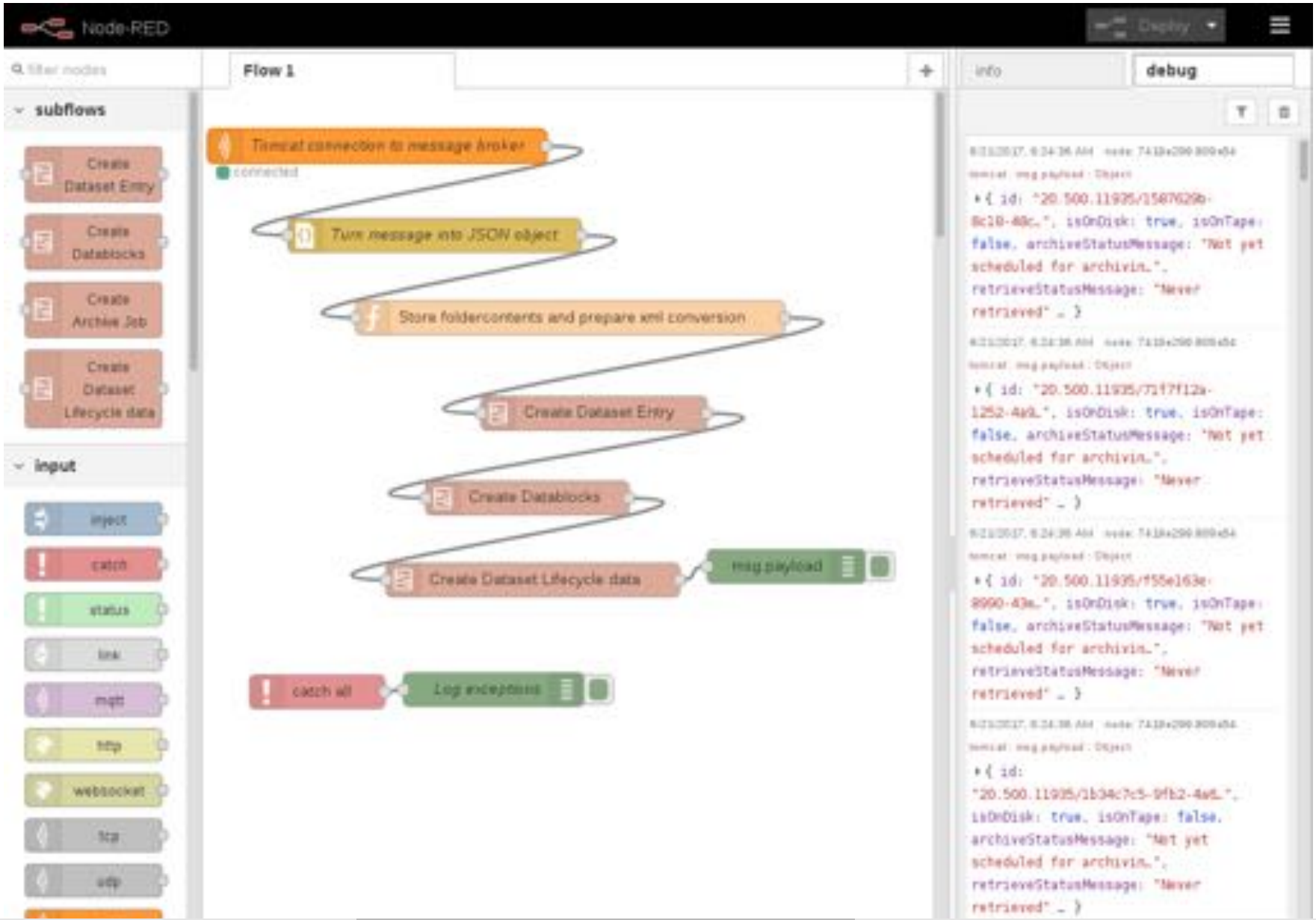
Message broker RabbitMQ Managament - Ingesting from Beamlines



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
- Flows publish formatted data to API server

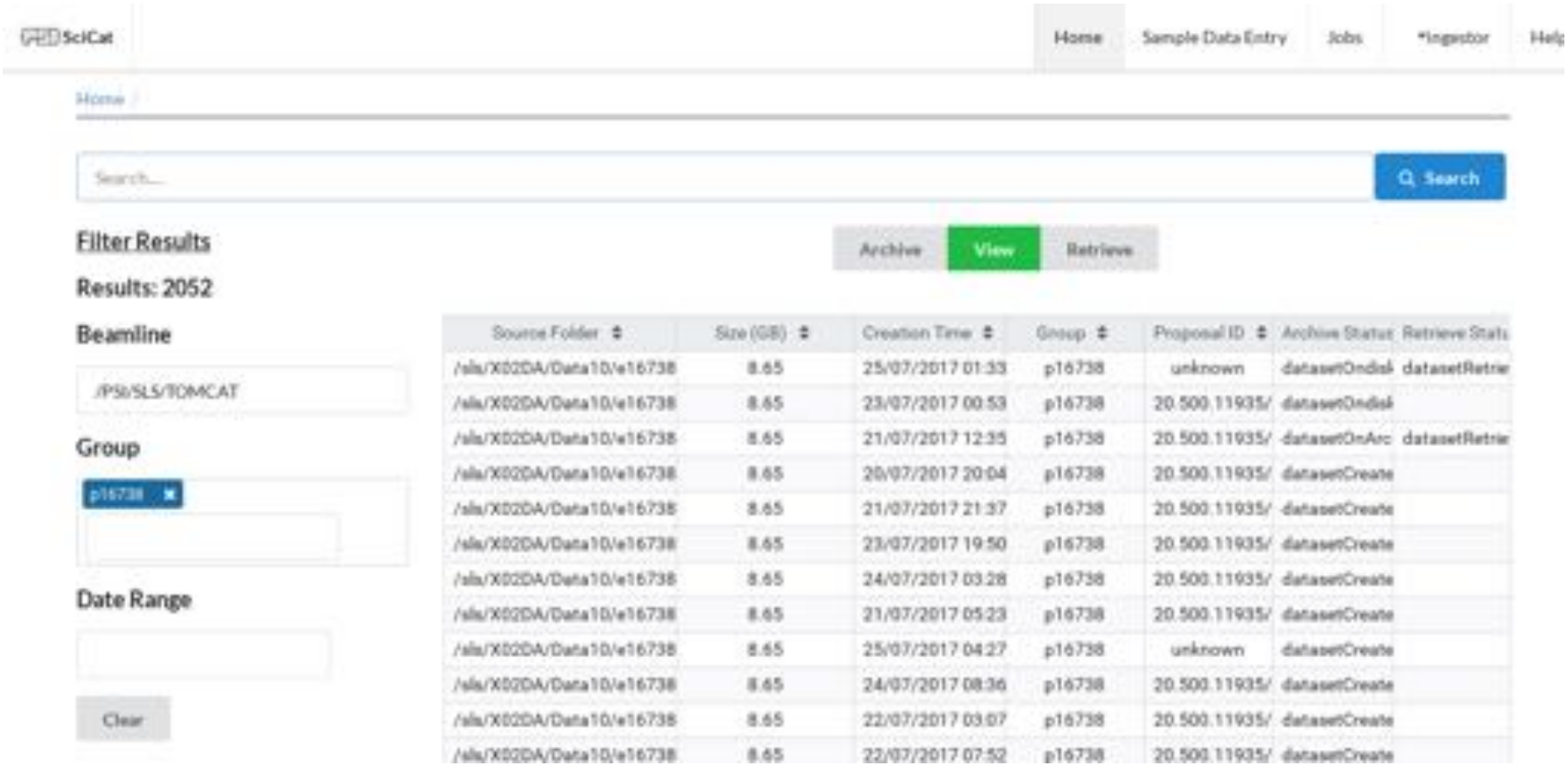
Node-Red Dataflow based coding



Frontend

- Built using Angular and Ngrx
- Created with Typescript
- Component based architecture to reuse throughout the application
- Responsive, Standards compliant and all other hip, web development words

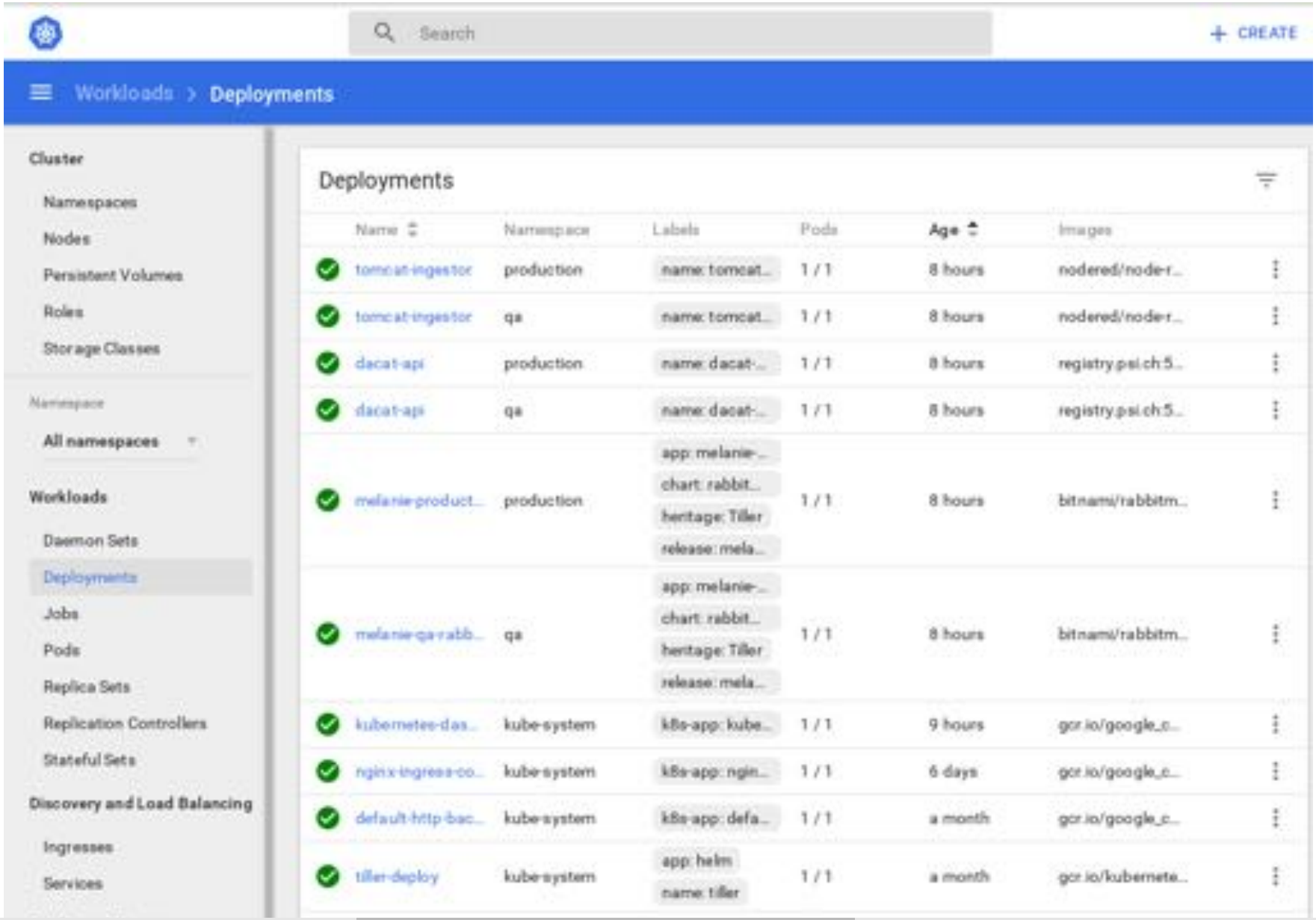
Frontend


SciCat
Home
Sample Data Entry
Jobs
*Ingestor
Help
Search...
Search
Filter Results
Results: 2052
Archive
View
Retrieve
Beamline
/PSI/SLS/TOMCAT
Group
p16738
Date Range
Clear
Source Folder
Size (GB)
Creation Time
Group
Proposal ID
Archive Status
Retrieve Stats
/sls/X02DA/Data10/e16738 8.65 25/07/2017 01:33 p16738 unknown datasetOndisk datasetRetrie
/sls/X02DA/Data10/e16738 8.65 23/07/2017 00:53 p16738 20.500.11935/ datasetOndisk
/sls/X02DA/Data10/e16738 8.65 21/07/2017 12:35 p16738 20.500.11935/ datasetOnArc datasetRetrie
/sls/X02DA/Data10/e16738 8.65 20/07/2017 20:04 p16738 20.500.11935/ datasetCreate
/sls/X02DA/Data10/e16738 8.65 21/07/2017 21:37 p16738 20.500.11935/ datasetCreate
/sls/X02DA/Data10/e16738 8.65 23/07/2017 19:50 p16738 20.500.11935/ datasetCreate
/sls/X02DA/Data10/e16738 8.65 24/07/2017 03:28 p16738 20.500.11935/ datasetCreate
/sls/X02DA/Data10/e16738 8.65 21/07/2017 05:23 p16738 20.500.11935/ datasetCreate
/sls/X02DA/Data10/e16738 8.65 25/07/2017 04:27 p16738 unknown datasetCreate
/sls/X02DA/Data10/e16738 8.65 24/07/2017 08:36 p16738 20.500.11935/ datasetCreate
/sls/X02DA/Data10/e16738 8.65 22/07/2017 03:07 p16738 20.500.11935/ datasetCreate
/sls/X02DA/Data10/e16738 8.65 22/07/2017 07:52 p16738 20.500.11935/ datasetCreate

Managment and deployment

- Kubernetes cluster (5 nodes)
- 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



Beamline integration

- Each beamline makes a cURL request to a predefined endpoint (language agnostic)
- There is one separate (node-red) container for each beamline, to maximize independence
- There is one "Virtual host" for the message broker for each facility

Sample Data Flow - Creation and Archiving

1. Data captured at beamline and cURL command publishes data (in any format) to RabbitMQ
2. Node-RED flow consumes data from the queue and formats according to RawDataset model
3. API server receives call and saves to MongoDB
4. Frontend website queries for datasets and presents to the logged in user with access
5. User chooses to archive dataset and command is sent to the API server
6. API server submits job to the archive system. Archive system updates the DatasetLifecycle

Deployment Process

- We use a Kubernetes Cluster as the basis for the deployment
- Do you have a Kubernetes cluster?
  - YES! Clone it and run it!
  - NO! Then grab a spare computer with a load of computing power
    - localdeploy is a prebuilt tool to create a Kubernetes cluster on your machine and deploy the basic services

Development principles and concepts

- We apply the "lazyness principle": avoid own coding wherever possible (e.g. use configuration driven development)
- Development and QA version for testing, Production version rolled out to beamline scientists
- Agile development with ESS and Max IV
- Regular meetings with scientists and requirements gathering. Issues created in repo.

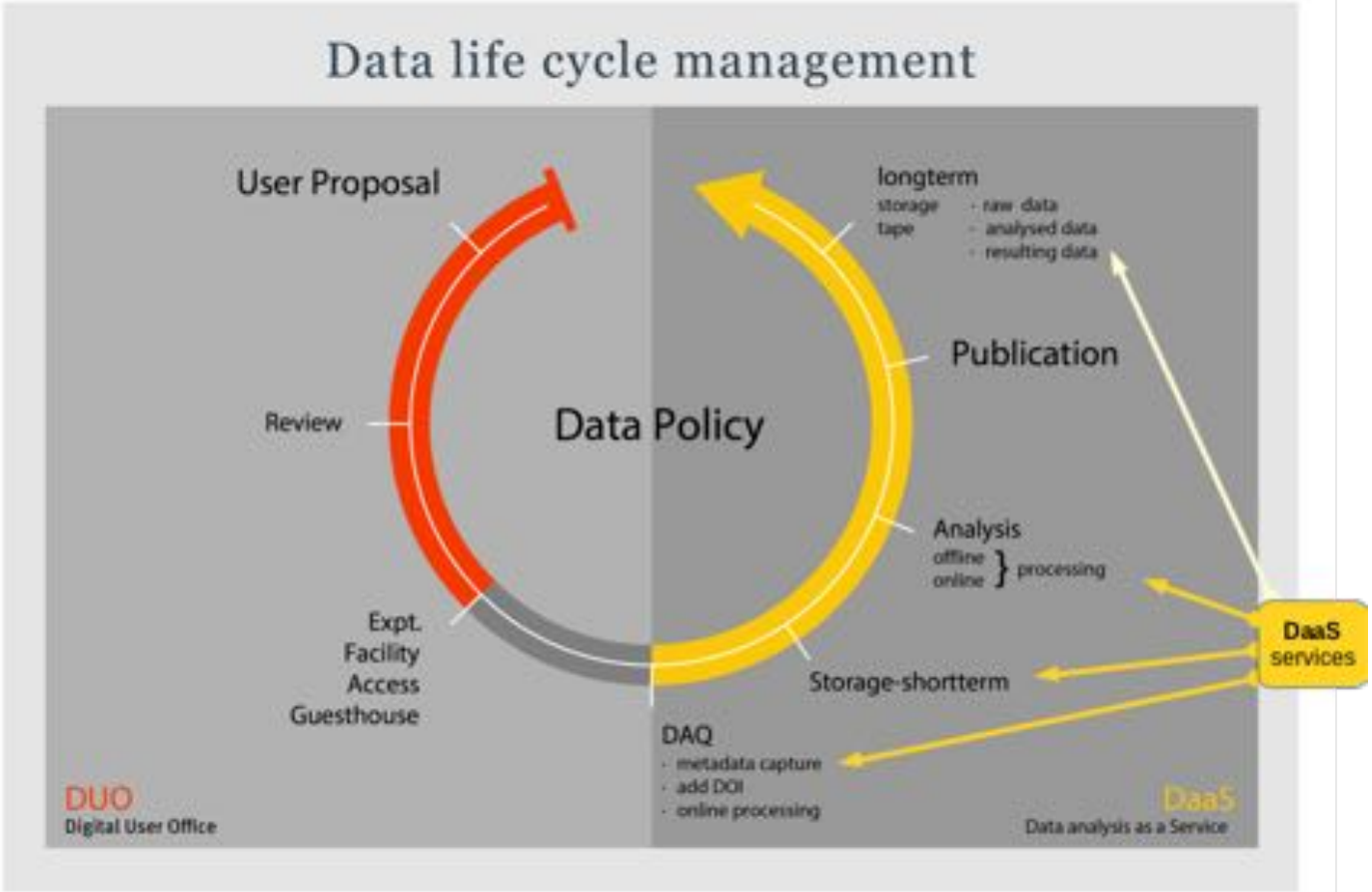
PAUL SCHERRER INSTITUT

Topic

Collaboration options

- SciCat Project - https://github.com/scicatproject
- Dedicated development started only about 6 months ago and we would like to see if our development can provide mutual benefits
- We want your feedback and experiences, as well as to identify areas with similar goals
- Standard Github procedure - collaborate with merge requests
- PSI is very open to suggestions for improvements and welcomes concrete contributions, ideally in form of reusable components
- We prefer to actually have a common code base, not a "one-time fork-and-forget" situation

Data Lifecycle Management



Data policy defines (long term) goals concerning data storage, life cycle management, data access and ownership

- Implementation of data policy needs a central data catalog

Prototype Tests/Planned Features