ESRF
The European Synchrotron

Metadata management at ESRF

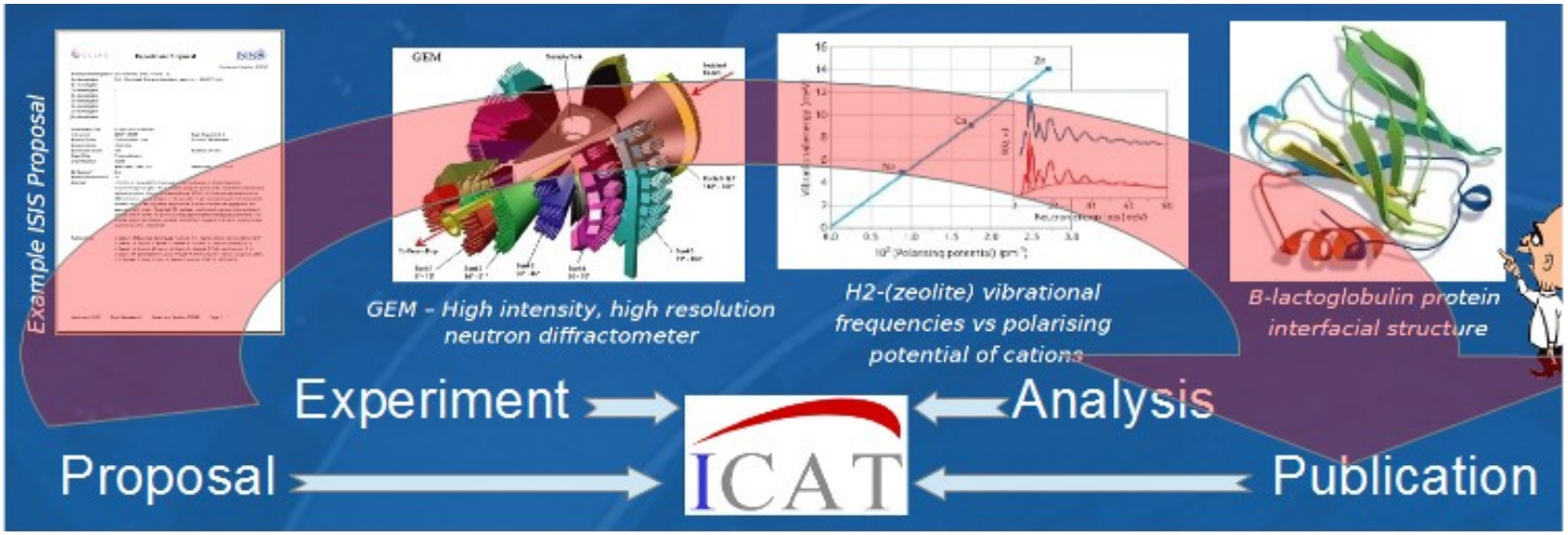
- Objective
- System design
- Beamline integration
- Project status
- HDF5 & NeXus for metadata

Background

Ojective:

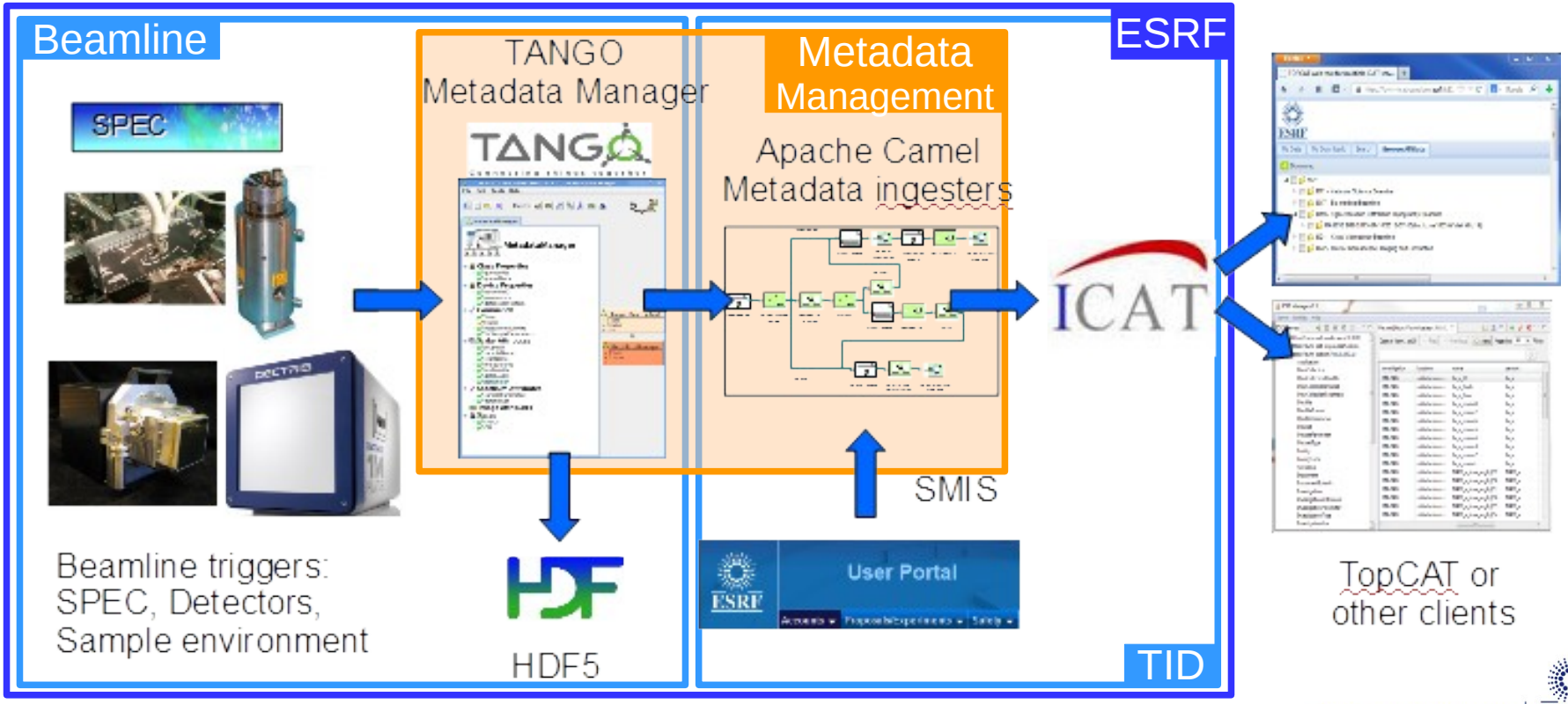
Design and implement a solution for gathering generic and specific metadata on all beamlines at the ESRF, store it in a database and make it available to users for browsing, searching and data analysis locally and remotely via a web browser.

Metadata catalogue: ICAT



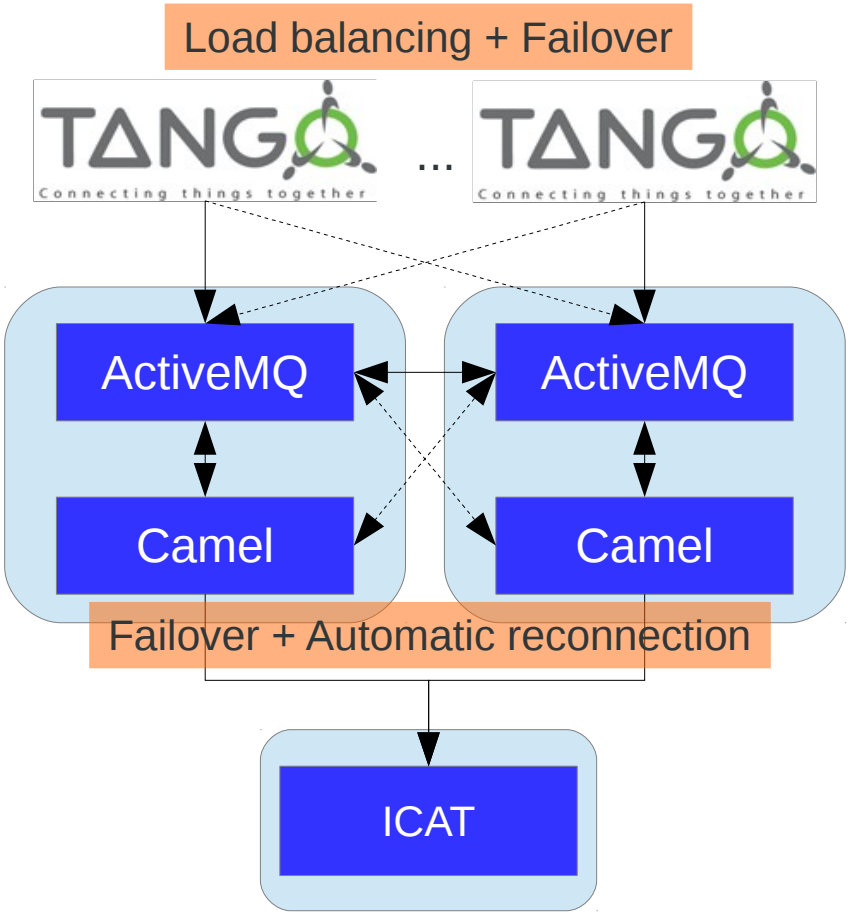
General system design

- Stateful Tango device on each beamline to collect metadata
  - Metadata stored in local HDF5 file and sent to registration queue
- Stateless queuing and processing system
  - Retrieves additional metadata from SMIS (users, title)
  - Format and stores all metadata in ICAT (incl. access rights)



Queuing system: JMS – Apache ActiveMQ

- Queues act as buffers in case of peak activity or slow process
- Integrity: message acknowledgement, persistence, transactions, failover
- Scalability: concurrent access, load balancing, network of brokers



ActiveMQ™

Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send

Connections

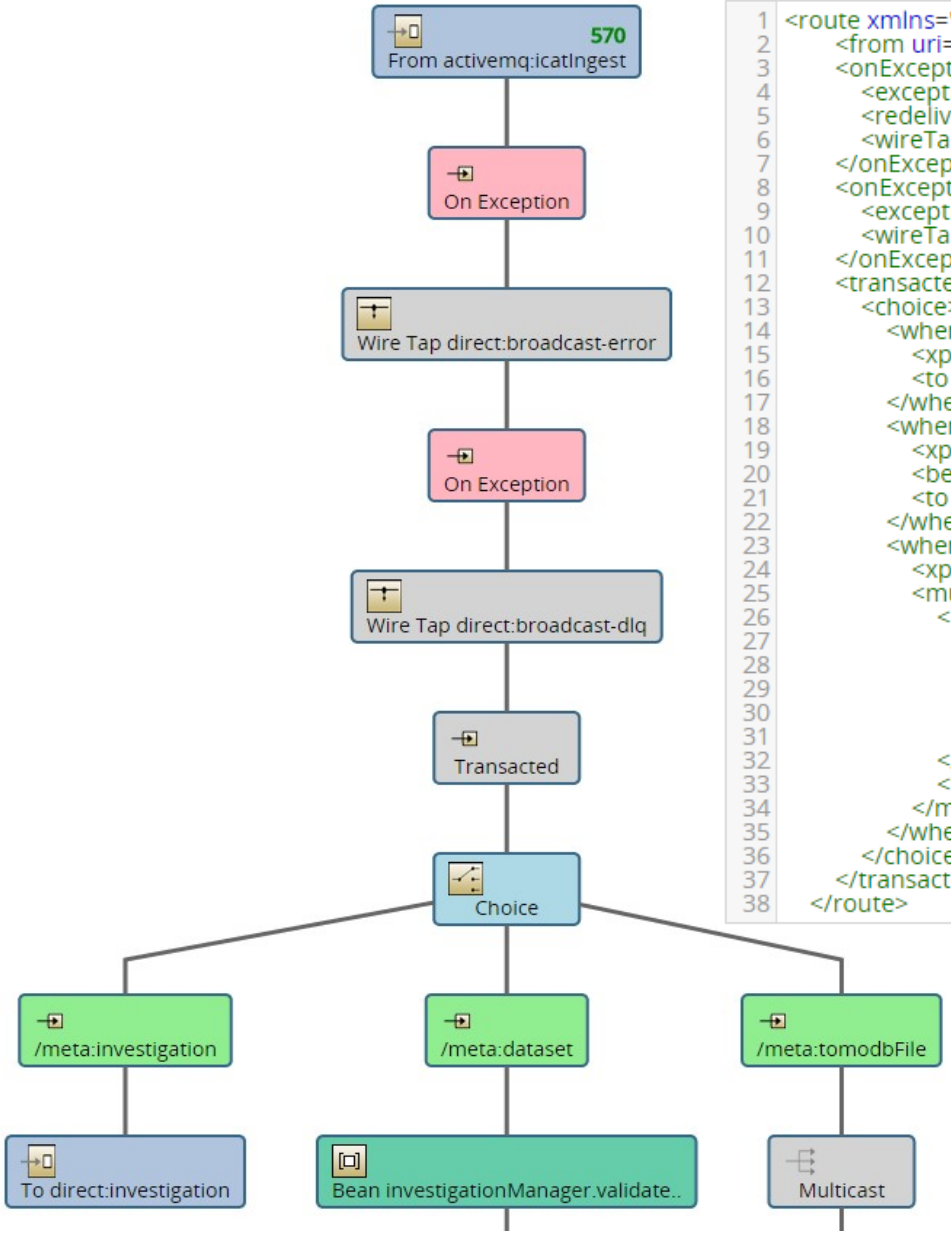
Connector tcp

Name ↑	Remote Address	Active	Slow
ID:bcu-mq-01.esrf.fr-40679-1421245783903-0:1	tcp://127.0.0.1:60900	true	false
ID:bcu-mq-01.esrf.fr-40679-1421245783903-0:10	tcp://127.0.0.1:60899	true	false
ID:bcu-mq-01.esrf.fr-40679-1421245783903-0:2	tcp://127.0.0.1:60898	true	false
ID:bcu-mq-01.esrf.fr-40679-1421245783903-0:3	tcp://127.0.0.1:60897	true	false
ID:bcu-mq-01.esrf.fr-40679-1421245783903-0:4	tcp://127.0.0.1:60904	true	false
ID:bcu-mq-01.esrf.fr-40679-1421245783903-0:5	tcp://127.0.0.1:60896	true	false
ID:bcu-mq-01.esrf.fr-40679-1421245783903-0:6	tcp://127.0.0.1:60901	true	false
ID:bcu-mq-01.esrf.fr-40679-1421245783903-0:7	tcp://127.0.0.1:60902	true	false
ID:bcu-mq-01.esrf.fr-40679-1421245783903-0:8	tcp://127.0.0.1:60905	true	false
ID:bcu-mq-01.esrf.fr-40679-1421245783903-0:9	tcp://127.0.0.1:60903	true	false
specialNC:localhost:outbound	tcp://160.103.236.113:53246	true	false
standardNC:localhost:outbound	tcp://160.103.236.113:53247	true	false

Connector stomp

Name	Remote Address	Active	Slow
ID:bcu-mq-01.esrf.fr-57127-1422373075877-4:2	tcp://160.103.37.2:49016	true	false
ID:bcu-mq-01.esrf.fr-57127-1422373075877-4:3	tcp://160.103.37.2:49173	true	false
ID:bcu-mq-01.esrf.fr-57127-1422373075877-4:13	tcp://160.103.21.38:51627	true	false
ID:bcu-mq-01.esrf.fr-57127-1422373075877-4:14	tcp://160.103.21.38:51635	true	false

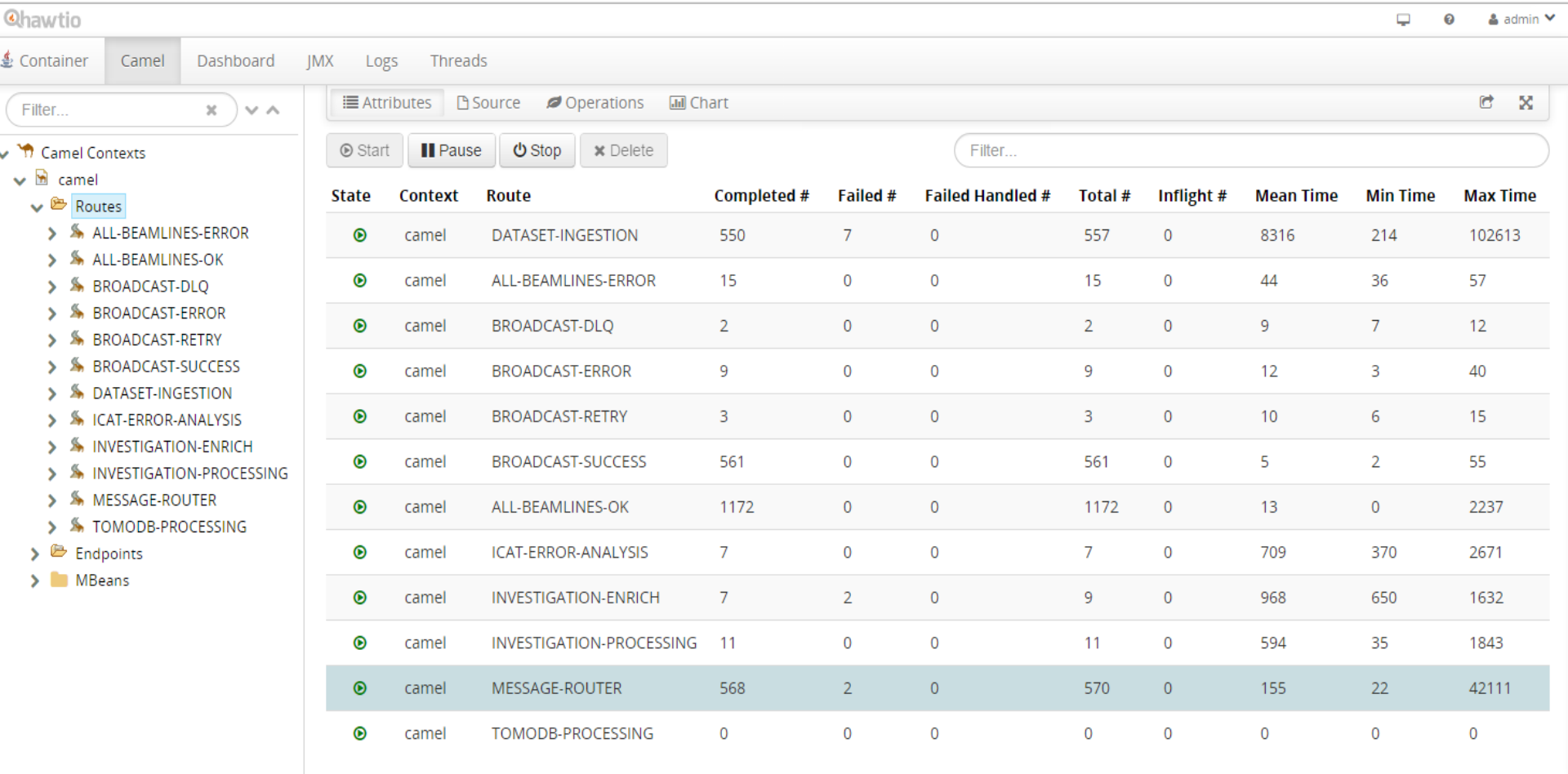
Processing system – Apache Camel




```
<route xmlns="http://camel.apache.org/schema/spring" id="MESSAGE-ROUTER">
    <from uri="activemq:icatIngest"/>
    <onException redeliveryPolicyRef="defaultRedelivery">
      <exception>fr.esrf.icat.client.exception.ICATDataException</exception>
      <redeliveryPolicy maximumRedeliveries="0"/>
      <wireTap uri="direct:broadcast-error"/>
    </onException>
    <onException redeliveryPolicyRef="defaultRedelivery" onRedeliveryRef="redeliveryWireTap"
      <exception>java.lang.Exception</exception>
      <wireTap uri="direct:broadcast-dlq"/>
    </onException>
    <transacted>
      <choice>
        <when>
          <xpath>/meta:investigation</xpath>
          <to uri="direct:investigation"/>
        </when>
        <when>
          <xpath>/meta:dataset</xpath>
          <bean ref="investigationManager" method="validateDataset"/>
          <to uri="activemq:datasetIngest"/>
        </when>
        <when>
          <xpath>/meta:tomodbFile</xpath>
          <multicast>
            <pipeline>
              <bean ref="tomodbFileManager" method="getInvestigation"/>
              <filter>
                <method bean="investigationManager" method="needToCreateInvestigation(
                <to uri="direct:investigation"/>
              </filter>
            </pipeline>
            <to uri="direct:tomodb"/>
          </multicast>
        </when>
      </choice>
    </transacted>
  </route>
```

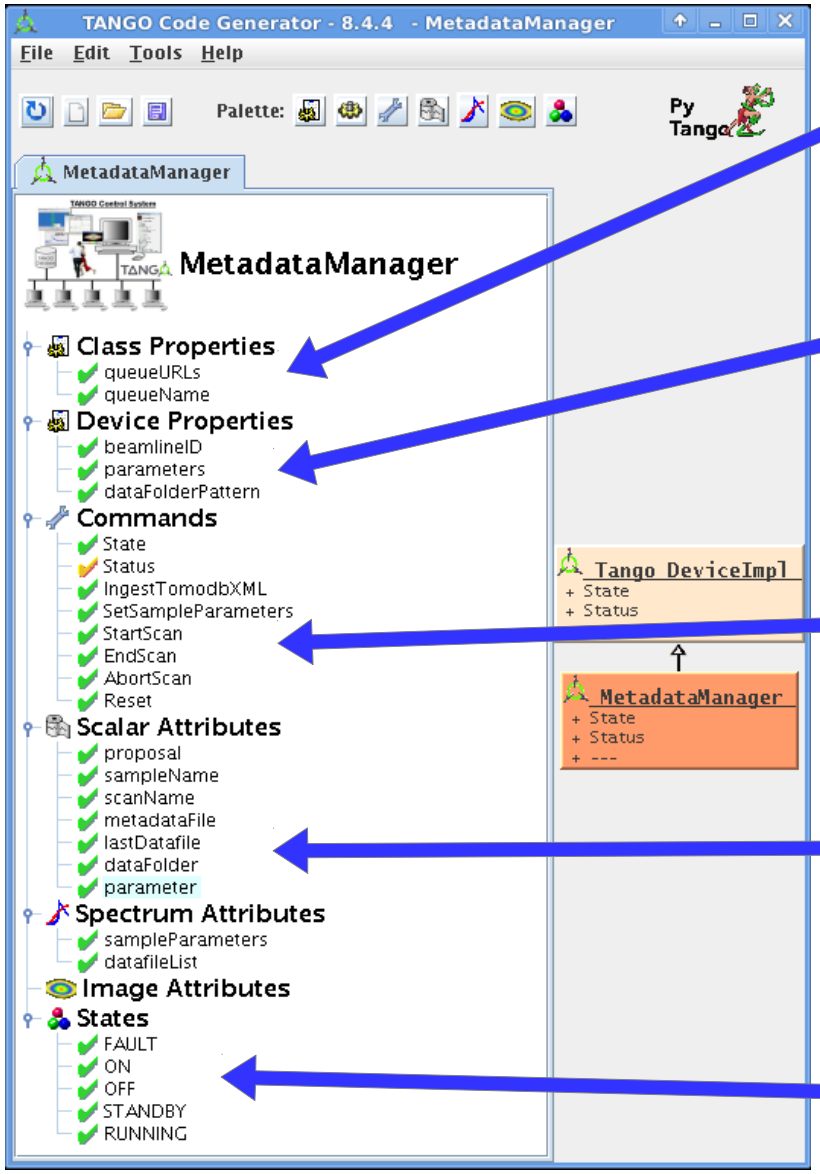
Admin console – Hawtio

- Web access to JMX through Jolokia JVM agent
- System monitoring (memory, disk, cpu, threads, logs)
- Route management (start / stop / pause / create new / modify existing)



State	Context	Route	Completed #	Failed #	Failed Handled #	Total #	Inflight #	Mean Time	Min Time	Max Time
	camel	DATASET-INGESTION	550	7	0	557	0	8316	214	102613
	camel	ALL-BEAMLINES-ERROR	15	0	0	15	0	44	36	57
	camel	BROADCAST-DLQ	2	0	0	2	0	9	7	12
	camel	BROADCAST-ERROR	9	0	0	9	0	12	3	40
	camel	BROADCAST-RETRY	3	0	0	3	0	10	6	15
	camel	BROADCAST-SUCCESS	561	0	0	561	0	5	2	55
	camel	ALL-BEAMLINES-OK	1172	0	0	1172	0	13	0	2237
	camel	ICAT-ERROR-ANALYSIS	7	0	0	7	0	709	370	2671
	camel	INVESTIGATION-ENRICH	7	2	0	9	0	968	650	1632
	camel	INVESTIGATION-PROCESSING	11	0	0	11	0	594	35	1843
	camel	MESSAGE-ROUTER	568	2	0	570	0	155	22	42111
	camel	TOMODB-PROCESSING	0	0	0	0	0	0	0	0

Tango device



- Where to send the metadata
  - PyTango ↔ JMS using STOMP
- What to send
  - Beamline configuration
  - Configurable metadata
- When to send
  - StartScan
  - EndScan / AbortScan
- Mandatory metadata
  - Proposal
  - Sample
  - Scan name
- States

Tango device configuration

- Parameters used as metadata are defined in the “parameters” property.
- Each parameter is used to define a dynamic attribute on the device, either:
  - A remote attribute, to read from another device:
    SR_Current = orion:10000/fe/id/16/SR_Current
  - A single array element of a remote attribute:
    param = id16na/device/attribute[0]
  - A list of remote attributes to concatenate:
    optics_positions = id16na/spec/dmir_zz/position,
    id16na/spec/mtz/position, id16na/spec/mty/position
  - *(TangoSpec motors and variables)*
  - A single element of a Spec associative array (through TangoSpec)
    param = id16na/spec/spec/FTOMO_PAR[param]
  - A local attribute that needs to be set on that device:
    scanType
- Dates are collected automatically:
  startDate when you StartScan, endDate when you EndScan

Beamline integration

- **Device installation**
  - Since version 1.2 standard blissinstaller (src install, no dependencies)
  - (TangoSpec installation and configuration if needed)
  - MetaExperiment: single device for proposal and sample definition
  - MetadataManager: configure several classes if the beamline runs very distinct experiments (ID16A: fluo vs. tomo)
- **Device configuration**
  - Each MetadataManager class has a single device with its own configuration, defining which parameters to register and from where
  - Manual configuration of parameters is tedious → Configuration GUI
- **Control system integration**
  - Set proposal, sample and scan name
  - Data folder (from or to device)
  - Calls to StartScan and EndScan

Example integration – ID16A

- **Our first client !**
  - Device installation done by me (1 device for tomo, 1 for fluo)
  - Device configuration and integration with spec done by P. Cloetens
  - Integration with GUI done by V. Valls
- **Configuration details:**
  - Storage ring parameters

```
machineMode = orion:10000/fe/id/16/SR_Filling_Mode
SR_Current = orion:10000/fe/id/16/SR_Current
insertionDeviceName
insertionDeviceGap = orion:10000/id/id/16ni/U18-3C_GAP_Position,
  orion:10000/id/id/16ni/U18-3D_GAP_Position
```

  - Beamline parameters

```
energy = id16ni/energy/master/position
monochromatorName = id16ni/energy/multilayer/positionId
filter = id16ni/attenuator/1/positionId, id16ni/attenuator/2/positionId,
  id16ni/attenuator/3/positionId
optics_sensors_labels
optics_sensors_values = id16ni/experiment-wago/kb/kb1_hmth,
  id16ni/experiment-wago/kb/kb1_vmth, id16ni/experiment-wago/kb/kb1_rx,
  id16ni/experiment-wago/kb/kb1_tx
```

Example integration – ID16A cont.

- Sample parameters

```
sample_motors
sample_positions = id16ni/motor/srot/position, id16ni/motor/sx/position,
  id16ni/motor/sy/position, id16ni/tripod/sample/zc,
  id16ni/motor/spy/position, id16ni/motor/spz/position,
  id16ni/motor/su/position, id16ni/motor/sv/position,
  id16ni/tripod/sample/pitch, id16ni/tripod/sample/roll,
  id16ni/tripod/frame/zc, id16ni/tripod/frame/pitch,
  id16ni/tripod/frame/roll
vacuum_labels
vacuum_values = id16ni/v-pen/111/pressure
```

- Tomography parameters

```
dark_N = id16ni/spec/zaptomo/DARK_N
ref_On = id16ni/spec/zaptomo/REF_ON
ref_N = id16ni/spec/zaptomo/REF_N
tomo_N = id16ni/spec/zaptomo/TOMO_N
y_Step = id16ni/spec/zaptomo/Y_STEP
```

- Other parameters

```
scanType
ccdtime = id16ni/spec/zaptomo/TOMO_EXPTIME
pixelSize = id16ni/ImagePixelSize/frelon1/pixelWidth
```

Example run - Experiment

- When experiment is started, spec/GUI calls StartScan
  - Start time recorded
  - Remote attributes read asynchronously
  - hdf5 file created with snapshot of parameter values
- When experiment ends, spec/GUI calls EndScan
  - End time recorded
  - Remote attributes read again
  - hdf5 file appended with snapshot of parameter values
  - XML message sent to queue
- If user abort the scan → AbortScan

  Nothing is registered
- If user aborts the scan but want to keep the data → InterruptScan

  Same as EndScan but dataset flagged as incomplete

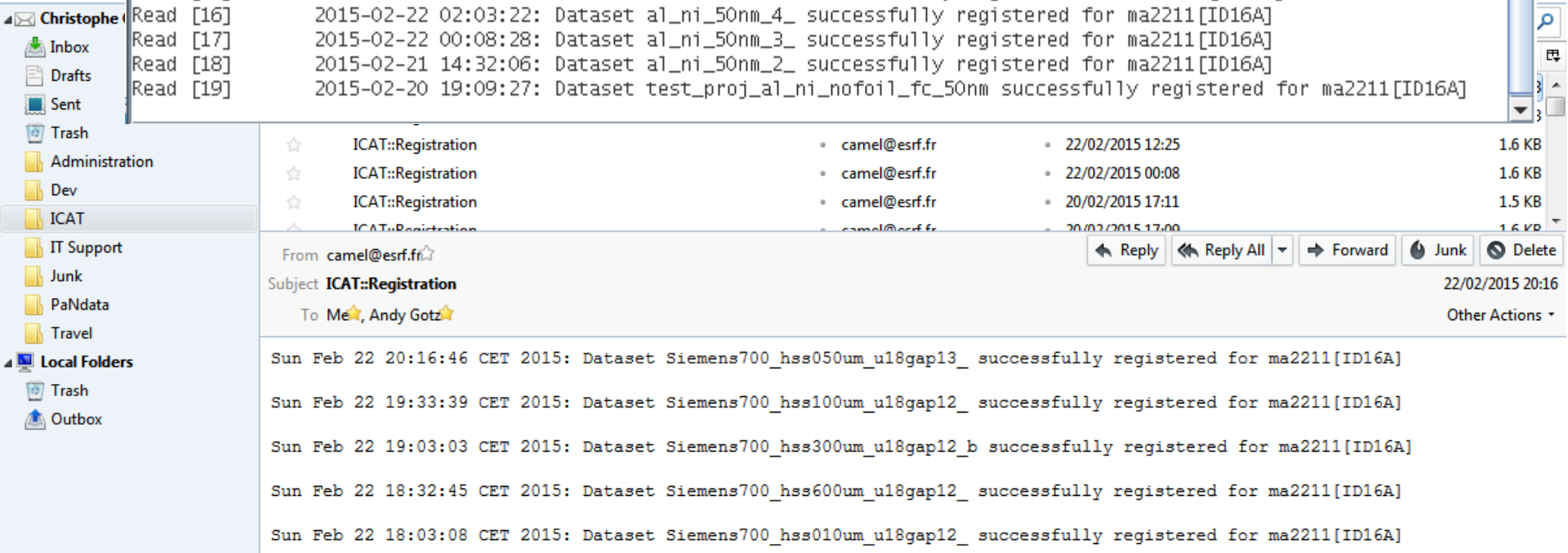
Example run – Behind the scene

- ActiveMQ receives the XML message
  - It is first stored locally and acknowledged
  - System verifies that the proposal exists
  - Data files are listed from data folder
  - Registration in ICAT
    - Dataset
    - Sample
    - Parameters
    - Datafiles
  - Registration feedback to beamline and admins

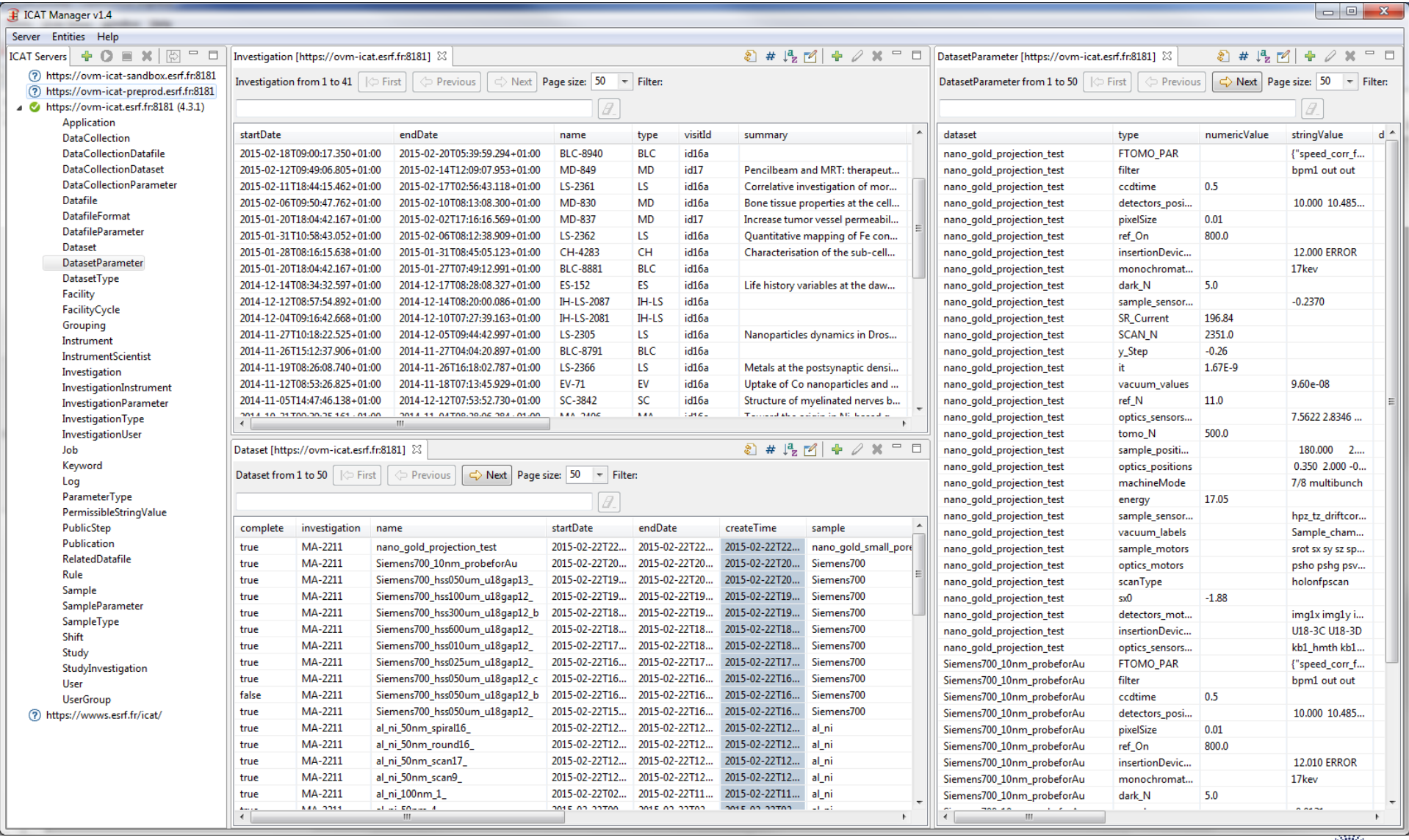
A few seconds after the end of the scan, users and beamline scientists can log in ICAT with their standard credentials, browse their datasets and the associated metadata, download the data and the metadata, all from a web browser.

Feedback from the ingestion process

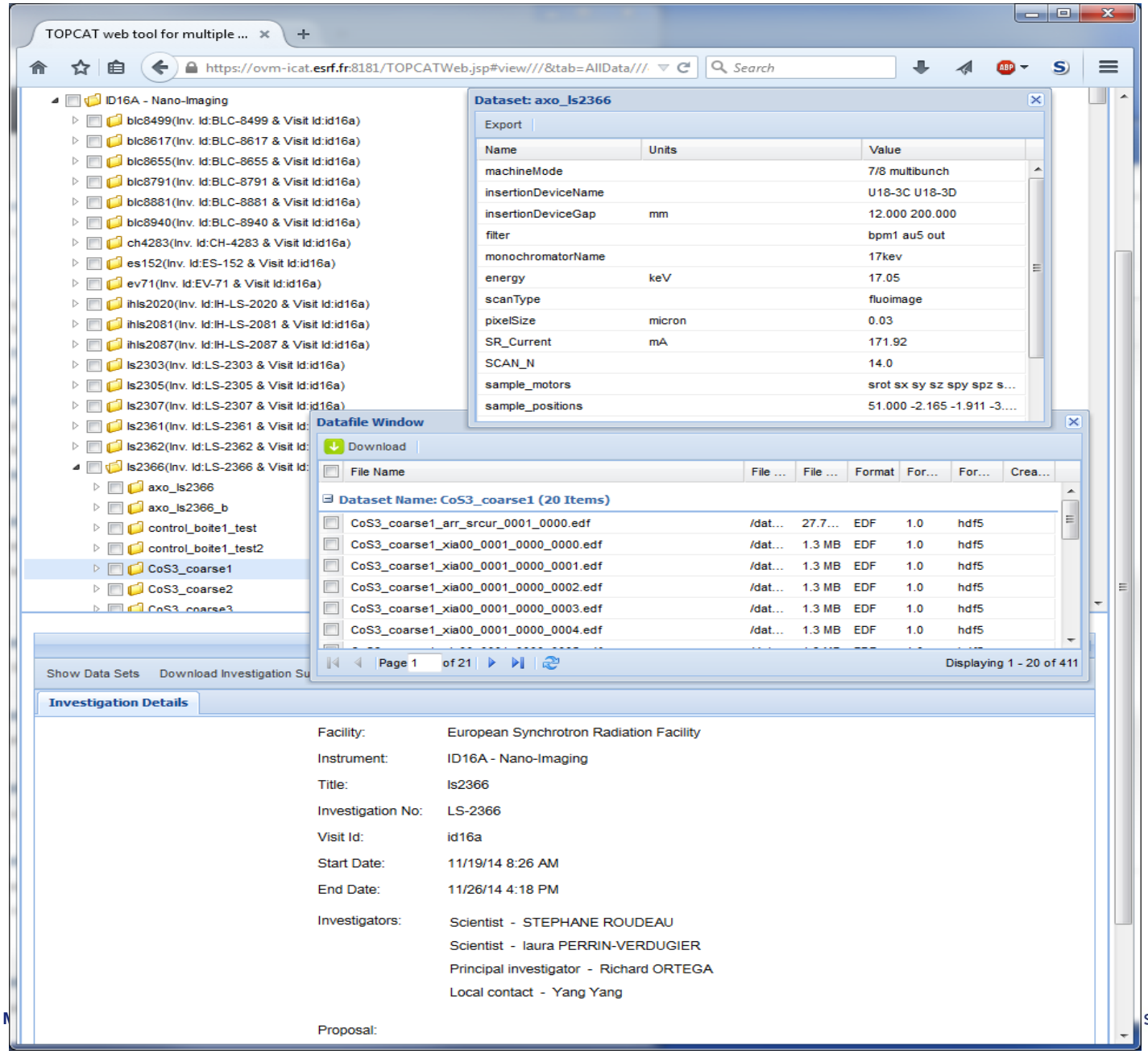
```
measure date: 23/02/2015 14:55:07 + 660ms
quality: VALID
dim x: 20
Read length: 20
Read [0]        2015-02-22 22:21:48: Dataset nano_gold_projection_test successfully registered for ma2211[ID16A]
Read [1]        2015-02-22 20:41:02: Dataset Siemens700_10nm_probeforAu successfully registered for ma2211[ID16A]
Read [2]        2015-02-22 20:16:46: Dataset Siemens700_hss050um_u18gap13_ successfully registered for ma2211[ID16A]
Read [3]        2015-02-22 19:33:39: Dataset Siemens700_hss100um_u18gap12_ successfully registered for ma2211[ID16A]
Read [4]        2015-02-22 19:03:03: Dataset Siemens700_hss300um_u18gap12_b successfully registered for ma2211[ID16A]
Read [5]        2015-02-22 18:32:45: Dataset Siemens700_hss600um_u18gap12_ successfully registered for ma2211[ID16A]
Read [6]        2015-02-22 18:03:08: Dataset Siemens700_hss010um_u18gap12_ successfully registered for ma2211[ID16A]
Read [7]        2015-02-22 17:21:43: Dataset Siemens700_hss025um_u18gap12_ successfully registered for ma2211[ID16A]
Read [8]        2015-02-22 16:48:15: Dataset Siemens700_hss050um_u18gap12_c successfully registered for ma2211[ID16A]
Read [9]        2015-02-22 16:19:56: Dataset Siemens700_hss050um_u18gap12_b successfully registered for ma2211[ID16A]
Read [10]       2015-02-22 16:13:53: Dataset Siemens700_hss050um_u18gap12_ successfully registered for ma2211[ID16A]
Read [11]       2015-02-22 12:28:20: Dataset al_ni_50nm_spiral16_ successfully registered for ma2211[ID16A]
Read [12]       2015-02-22 12:25:58: Dataset al_ni_50nm_round16_ successfully registered for ma2211[ID16A]
Read [13]       2015-02-22 12:23:30: Dataset al_ni_50nm_scan17_ successfully registered for ma2211[ID16A]
Read [14]       2015-02-22 12:21:28: Dataset al_ni_50nm_scan9_ successfully registered for ma2211[ID16A]
Read [15]       2015-02-22 11:25:14: Dataset al_ni_100nm_1_ successfully registered for ma2211[ID16A]
Read [16]       2015-02-22 02:03:22: Dataset al_ni_50nm_4_ successfully registered for ma2211[ID16A]
Read [17]       2015-02-22 00:08:28: Dataset al_ni_50nm_3_ successfully registered for ma2211[ID16A]
Read [18]       2015-02-21 14:32:06: Dataset al_ni_50nm_2_ successfully registered for ma2211[ID16A]
Read [19]       2015-02-20 19:09:27: Dataset test_proj_al_ni_nofoil_fc_50nm successfully registered for ma2211[ID16A]
```



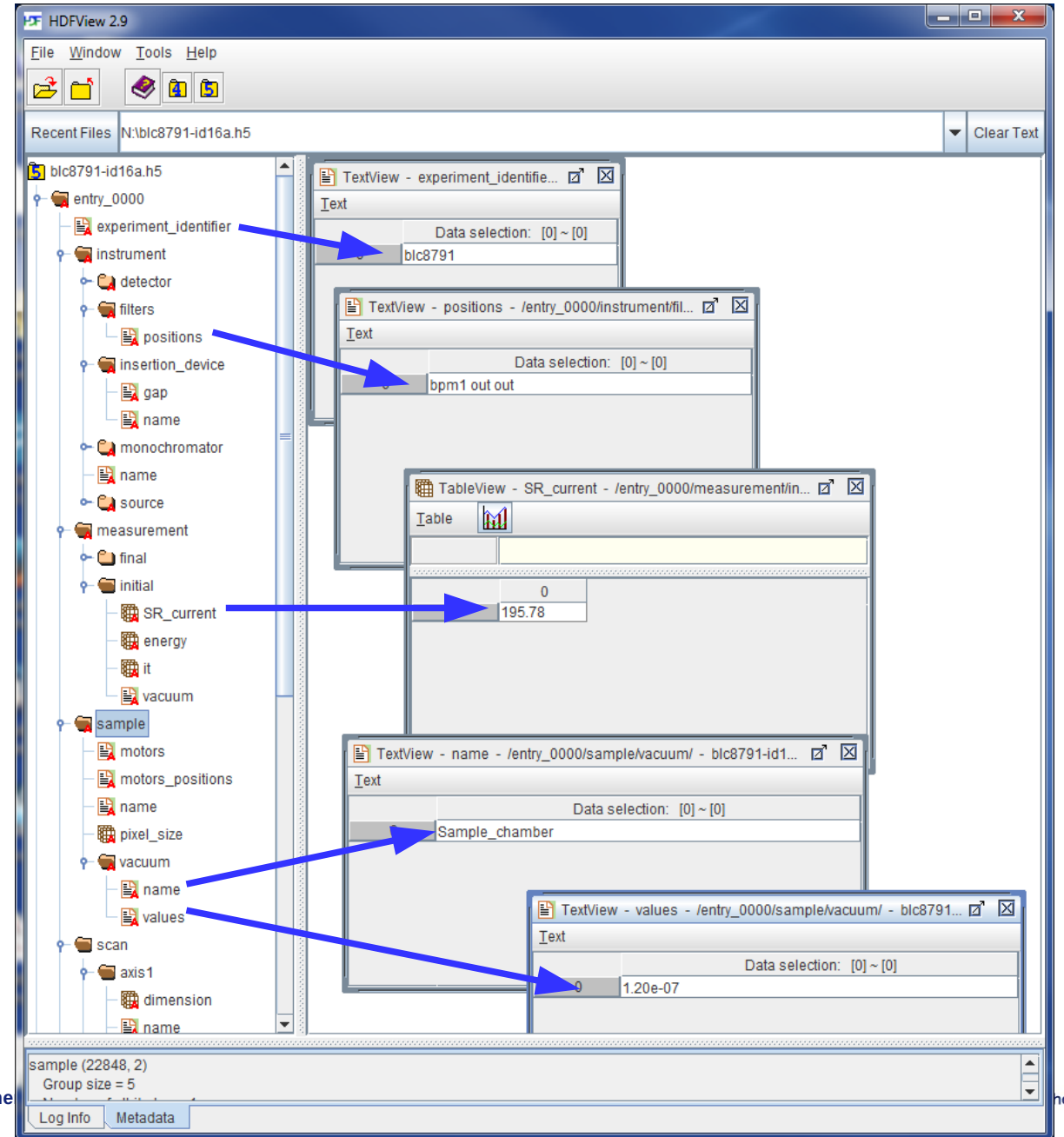
How it looks in the database



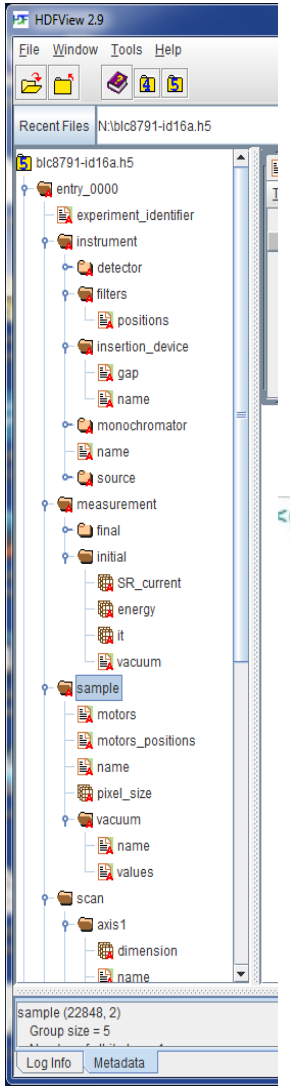
How it looks on the web



How it looks in the hdf5 file

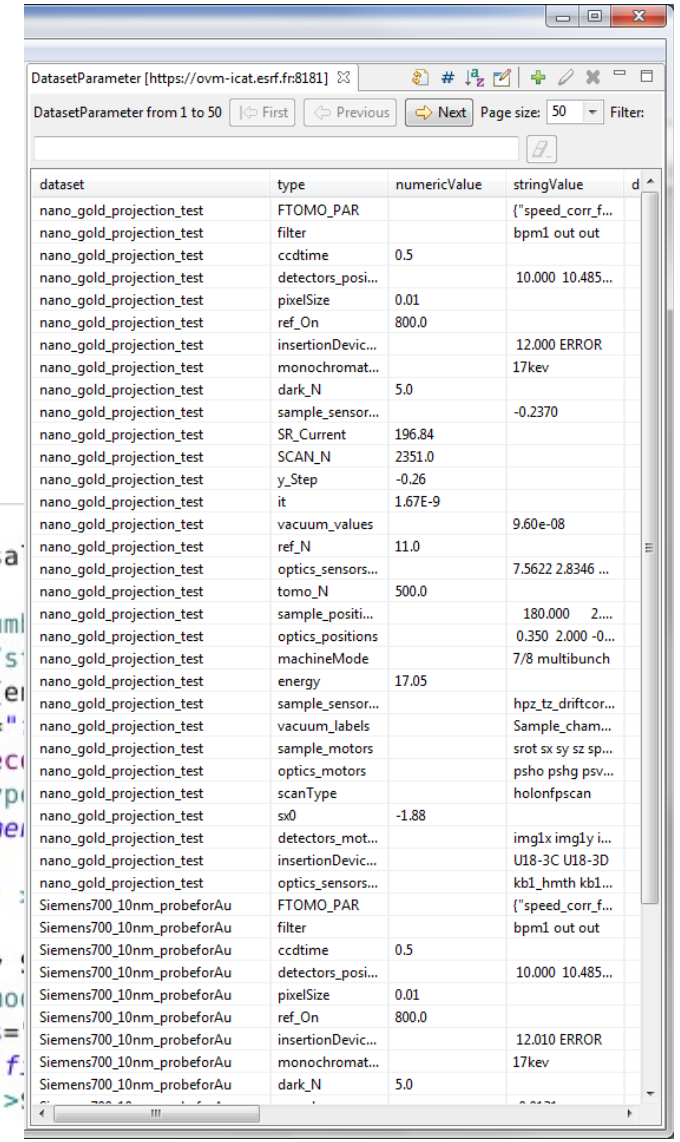


ICAT ↔ HDF5 mapping



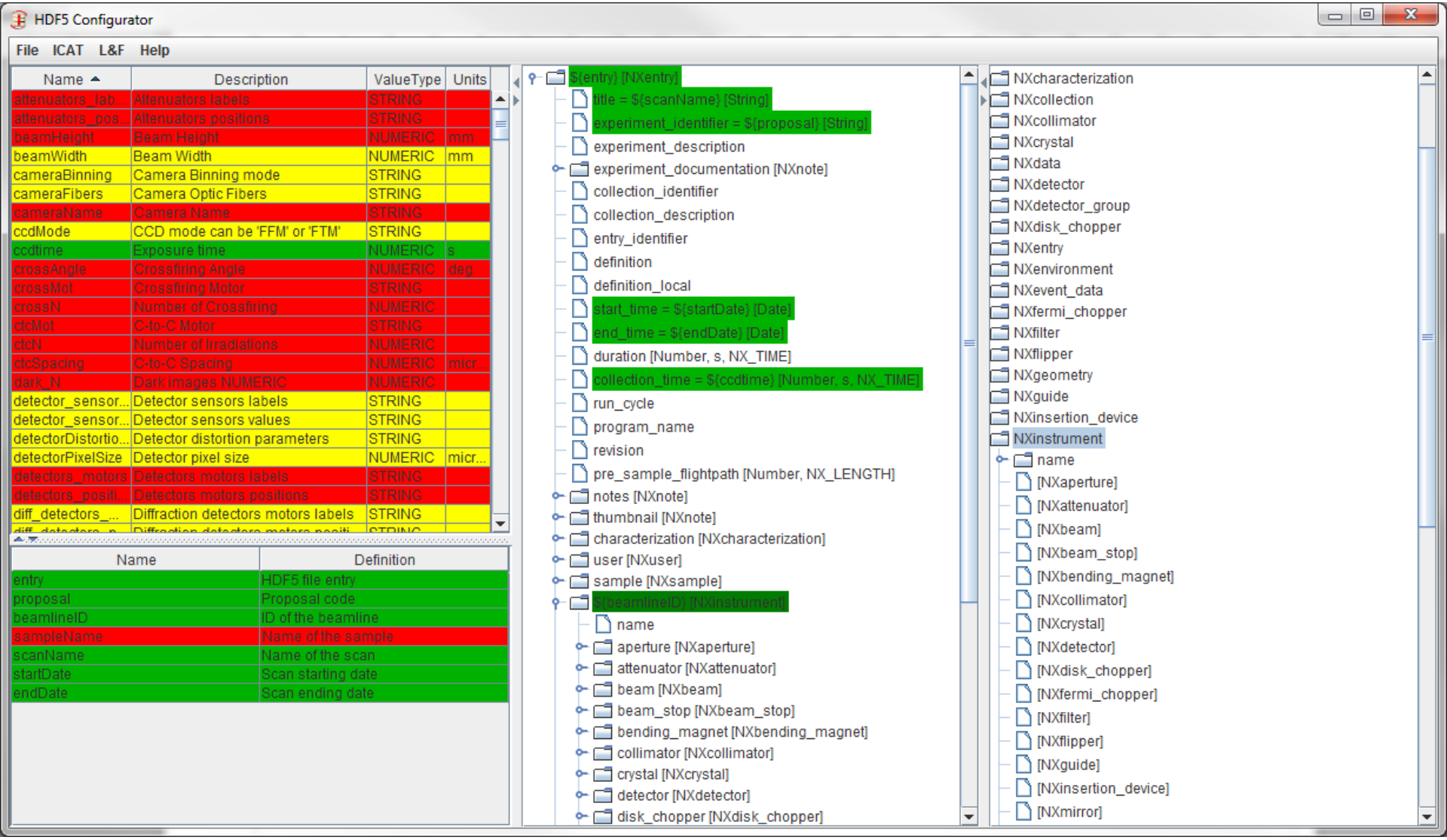
XML description of the HDF5 format with mapping of ICAT parameters

```
<group groupName="${entry}" class="@NXentry" >
    <experiment_identifier type="NX_CHAR" >${proposa
    <title type="NX_CHAR" >${scanName}</title>
    <scan_number type="NX_CHAR" >${SCAN_N}</scan_numl
    <start_time type="NX_DATE_TIME" >${startDate}</s
    <end_time type="NX_DATE_TIME" record="final">${e
    <duration type="NX_FLOAT32" units="min" record="
    <collection_time type="NX_FLOAT32" units="s" rec
    <scan_type type="NX_CHAR" >${scanType}</scan_typ
    <group groupName="instrument" class="@NXinstrume
        <name type="NX_CHAR" >${beamlineID}</name>
        <group groupName="source" class="@NXsource"
            <name type="NX_CHAR" >ESRF</name>
            <type type="NX_CHAR" >Synchrotron X-ray
            <mode type="NX_CHAR" >${machineMode}</mo
            <current_start type="NX_FLOAT32"  units=
            <current_end type="NX_FLOAT32" record="f
            <distance type="NX_FLOAT32" units="mm" >
        </group>
```



dataset	type	numericValue	stringValue	d
nano_gold_projection_test	FTOMO_PAR		{"speed_corr_f...	
nano_gold_projection_test	filter		bpm1 out out	
nano_gold_projection_test	ccdtime	0.5		
nano_gold_projection_test	detectors_posi...		10.000 10.485...	
nano_gold_projection_test	pixelSize	0.01		
nano_gold_projection_test	ref_On	800.0		
nano_gold_projection_test	insertionDevic...		12.000 ERROR	
nano_gold_projection_test	monochromat...		17kev	
nano_gold_projection_test	dark_N	5.0		
nano_gold_projection_test	sample_sensor...		-0.2370	
nano_gold_projection_test	SR_Current	196.84		
nano_gold_projection_test	SCAN_N	2351.0		
nano_gold_projection_test	y_Step	-0.26		
nano_gold_projection_test	it	1.67E-9		
nano_gold_projection_test	vacuum_values		9.60e-08	
nano_gold_projection_test	ref_N	11.0		
nano_gold_projection_test	optics_sensors...		7.5622 2.8346 ...	
nano_gold_projection_test	tomo_N	500.0		
nano_gold_projection_test	sample_positi...		180.000 2....	
nano_gold_projection_test	optics_positions		0.350 2.000 -0...	
nano_gold_projection_test	machineMode		7/8 multibunch	
nano_gold_projection_test	energy	17.05		
nano_gold_projection_test	sample_sensor...		hpz_tz_driftcor...	
nano_gold_projection_test	vacuum_labels		Sample_cham...	
nano_gold_projection_test	sample_motors		srot sx sy sz sp...	
nano_gold_projection_test	optics_motors		psho pshg psv...	
nano_gold_projection_test	scanType		holonfpscan	
nano_gold_projection_test	sx0	-1.88		
nano_gold_projection_test	detectors_mot...		img1x img1y i...	
nano_gold_projection_test	insertionDevic...		U18-3C U18-3D	
nano_gold_projection_test	optics_sensors...		kb1_hmth kb1...	
Siemens700_10nm_probeforAu	FTOMO_PAR		{"speed_corr_f...	
Siemens700_10nm_probeforAu	filter		bpm1 out out	
Siemens700_10nm_probeforAu	ccdtime	0.5		
Siemens700_10nm_probeforAu	detectors_posi...		10.000 10.485...	
Siemens700_10nm_probeforAu	pixelSize	0.01		
Siemens700_10nm_probeforAu	ref_On	800.0		
Siemens700_10nm_probeforAu	insertionDevic...		12.010 ERROR	
Siemens700_10nm_probeforAu	monochromat...		17kev	
Siemens700_10nm_probeforAu	dark_N	5.0		

HDF5 configuration from ICAT + NeXus classes



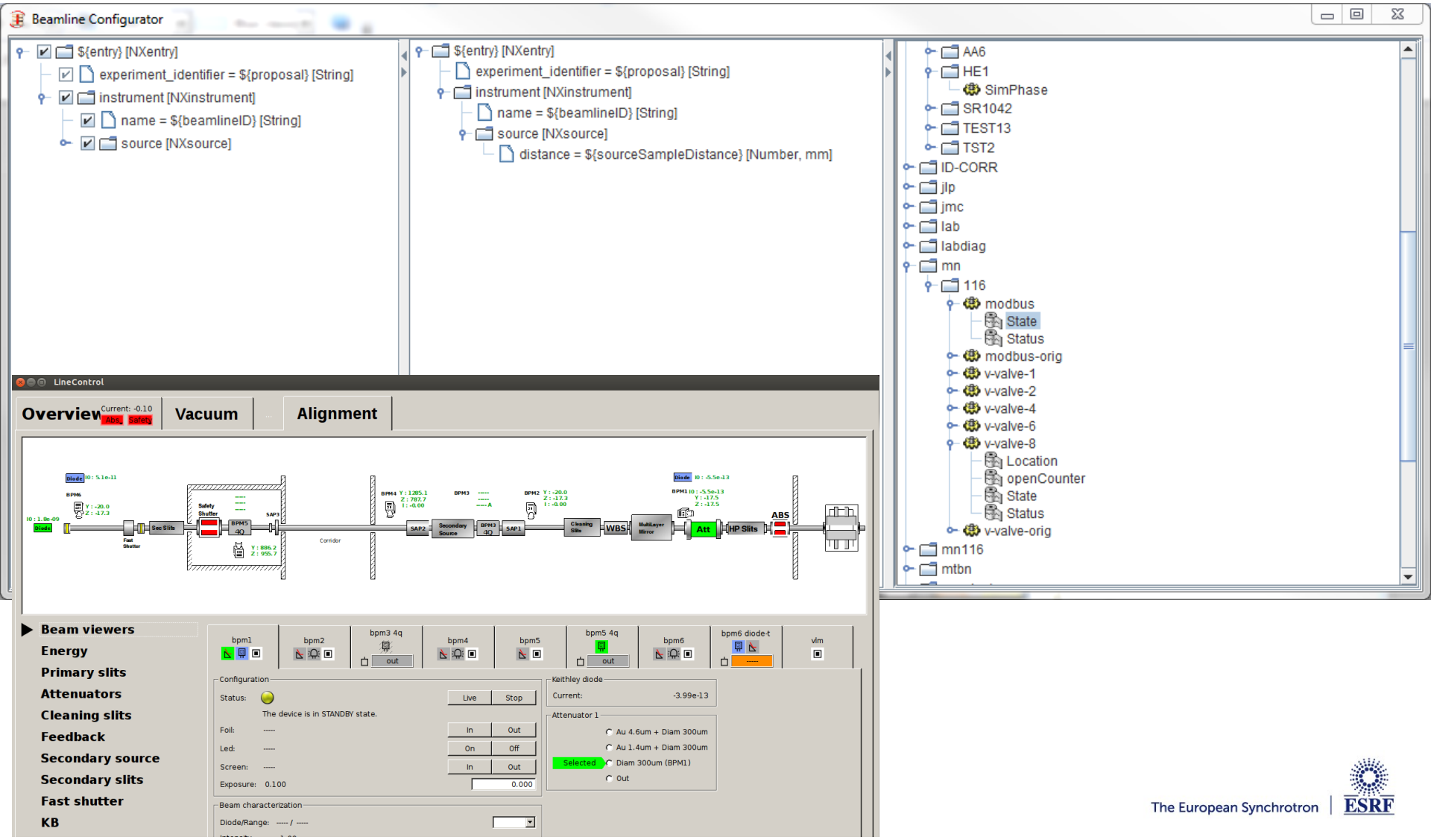
HDF5Export script

- python-icat script + bash script
- 'HDF5Export cleva MA-2240 newfile.hdf5'
  - Asks for password
  - Connects to ICAT
  - Retrieves all Datasets for Investigation MA-2240
  - Uses same HDF5Writer than Tango device
  - Export 1 hdf5 entry for each dataset in the file
  - Each entry has the format defined previously

- Allows beamline scientists to recreate on the fly an experiment summary
  - If original is lost
  - If hdf5 format changed

Beamline configuration from HDF5 format

- Users select (groups of) parameters they want to register
- They assign device attributes to the selected parameters



Project roadmap - past

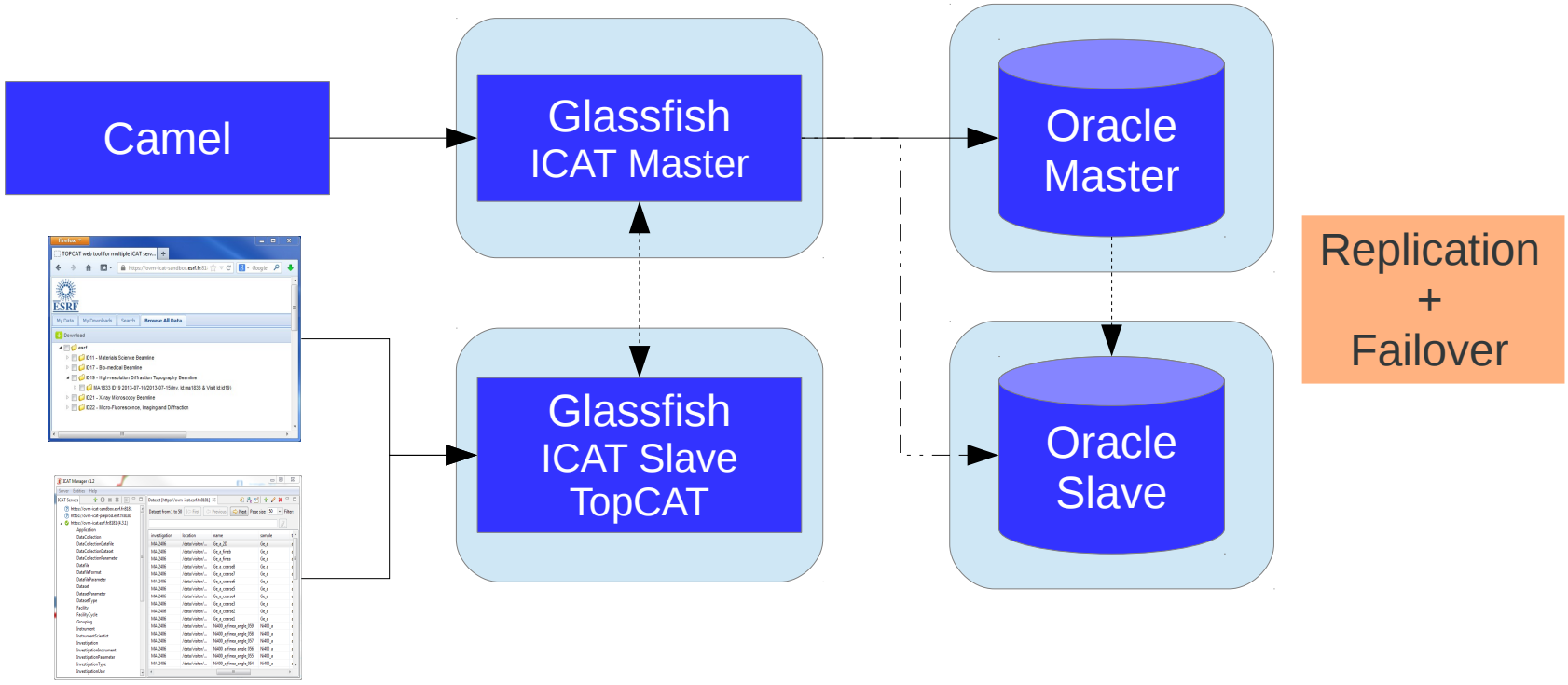
- **Started on the project October 2013**
  - Installation of ICAT, database, TopCAT + other modules
  - Finalization of pilot project
- **Initial tests with Camel started February 2014**
  - Development started March 2014
  - Discussion with ID16A scientists started Mai 2014
  - Initial test version June 2014
  - Modifications based on scientists feedback and initial testing
- **First production release end August 2014**
  - Integration and configuration for ID16A
- **First automatic metadata registration 30 September 2014**
  - BLC-8617 then LS-2307
- **Current version 1.5**

Status of integration

- **ID16A:**
  - Installed, configured and integrated to spec and GUI
  - Running for all experiments (incl. BLC and IH) since 2014/09/30
- **ID17:**
  - Installed, configured and integrated to spec and GUI
  - Running for MRT experiments since 2015/02/02
- **ID16B:**
  - Installed and configured
  - Integration on-hold
- **ID01:**
  - Installed
  - Configuration, integration and testing ongoing
- **ID06:**
  - Installed
  - Configuration, integration and testing ongoing

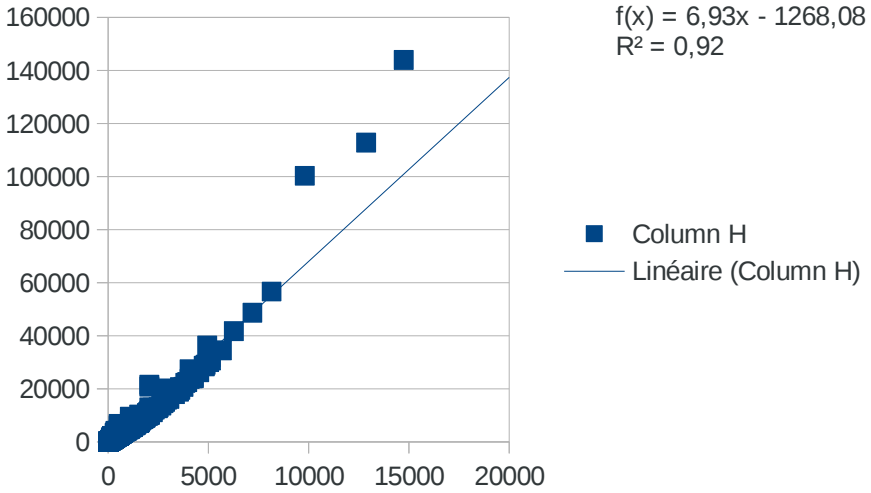
Project roadmap - future

- **Things remaining to do:**
  - Record processing step in addition to acquisition
  - Update backend (ICAT, TopCAT, IDS)
  - Improve installation of ICAT (WS + DB)



Registration statistics

- **In 13 weeks of operation on ID16A (as of 2015/02/11):**
  - 19 proposals,
  - 69 users,
  - 139 samples,
  - 1917 datasets,
  - 55 172 parameters,
  - 1 807 096 datafiles registered
- Average registration time: 8.3 s / dataset



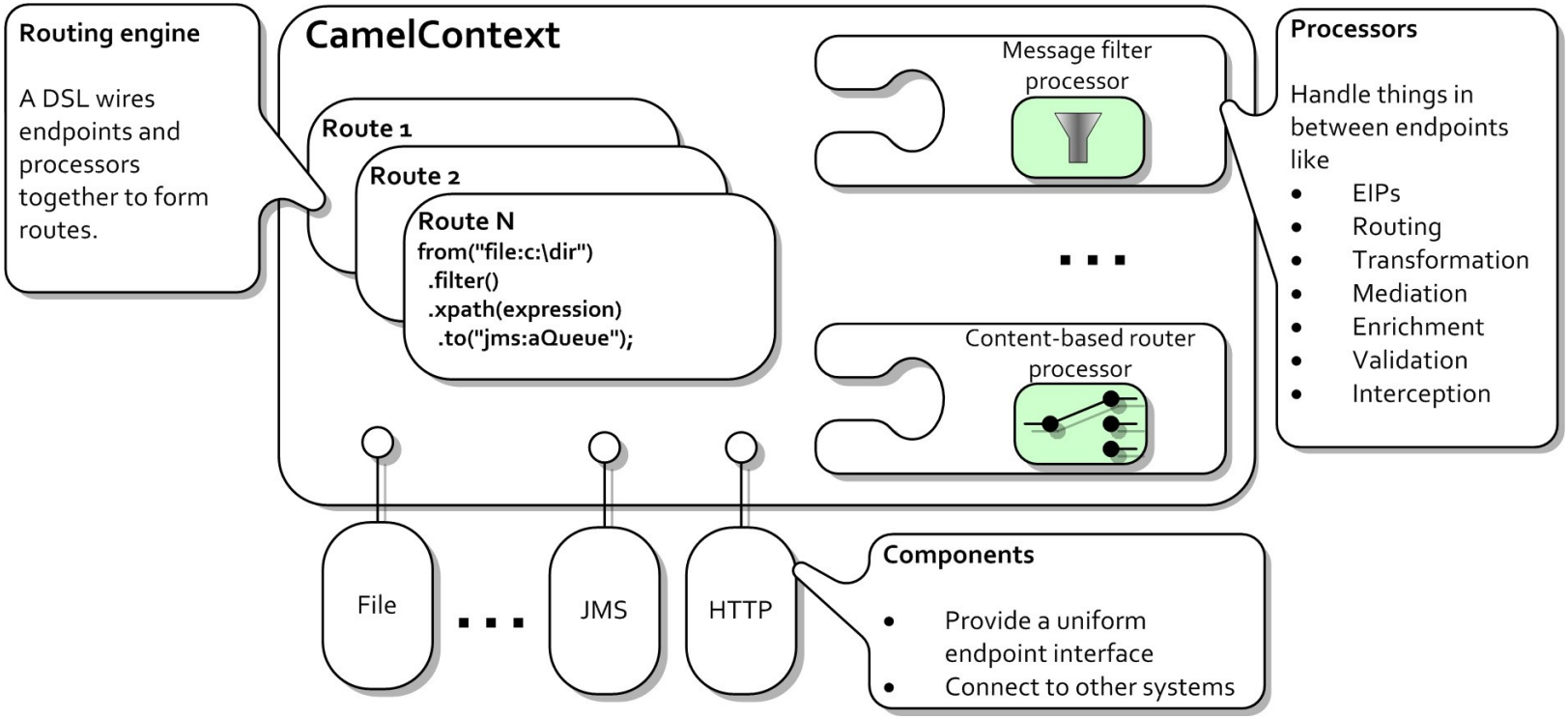
The European Synchrotron

Thank you for your attention

What is Camel ? (from http://java.dzone.com/articles/open-source-integration-apache)

Apache Camel is an open source Java framework that focuses on making integration easier and more accessible to developers. It does this by providing:

- Concrete implementations of the widely used Enterprise Integration Patterns
- Connectivity to a great variety of transports and APIs
- Domain Specific Languages (DSLs) to wire EIPs and transports together



Advantages of using Apache Camel

Most features already implemented

- Concurrent processing, thread pools, pooled connections
- Support transactions out of the box
- Automatic message translation (marshall/unmarshall, conversion)
  > 30 data format recognized & > 170 converters out of the box
- Configurable error handler, redelivery policy, logging and tracing

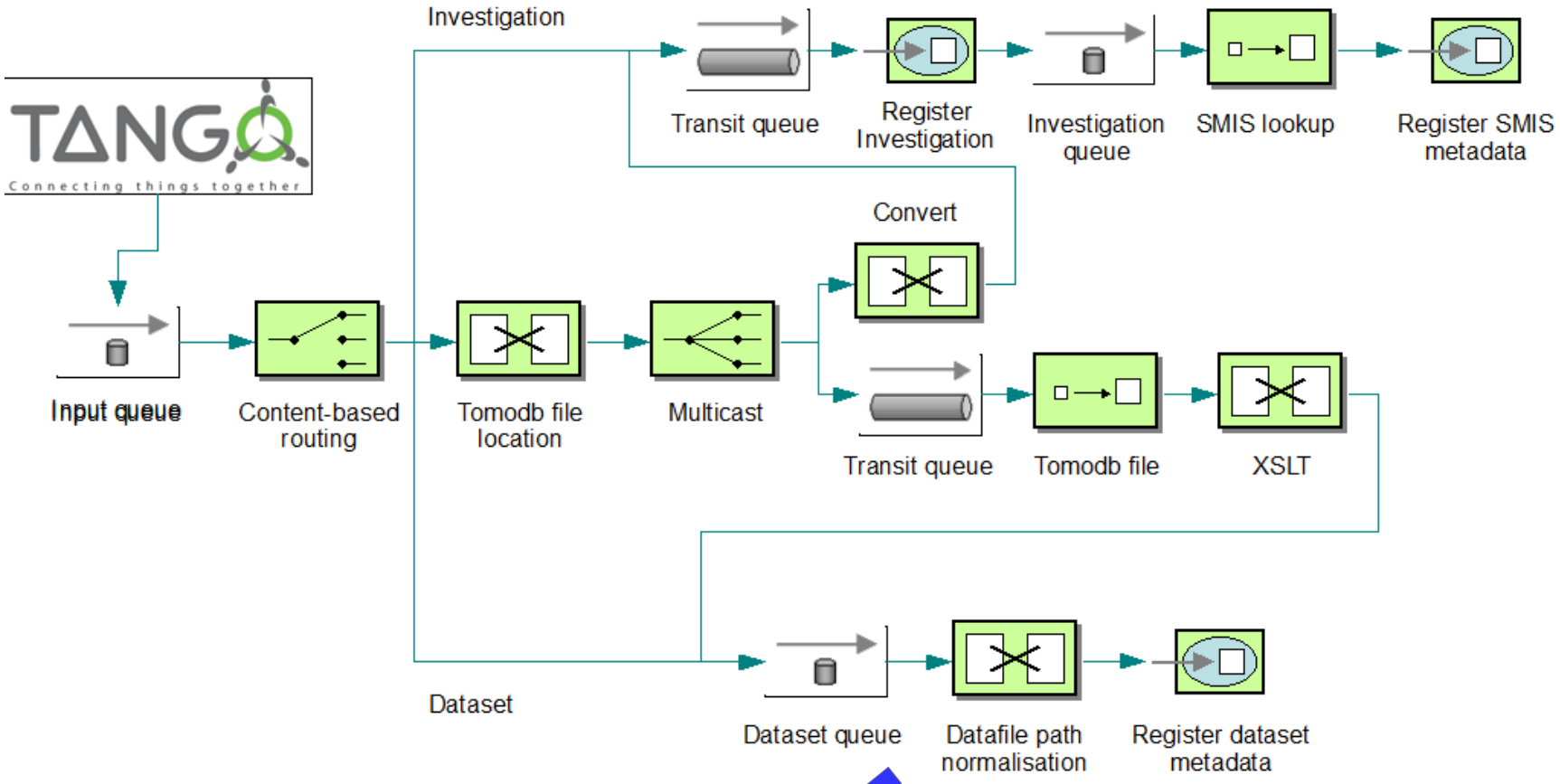
Technology agnostic

- Implements well-defined Enterprise Integration Patterns
- Independent of queuing system, transport technology, …

Very well integrated

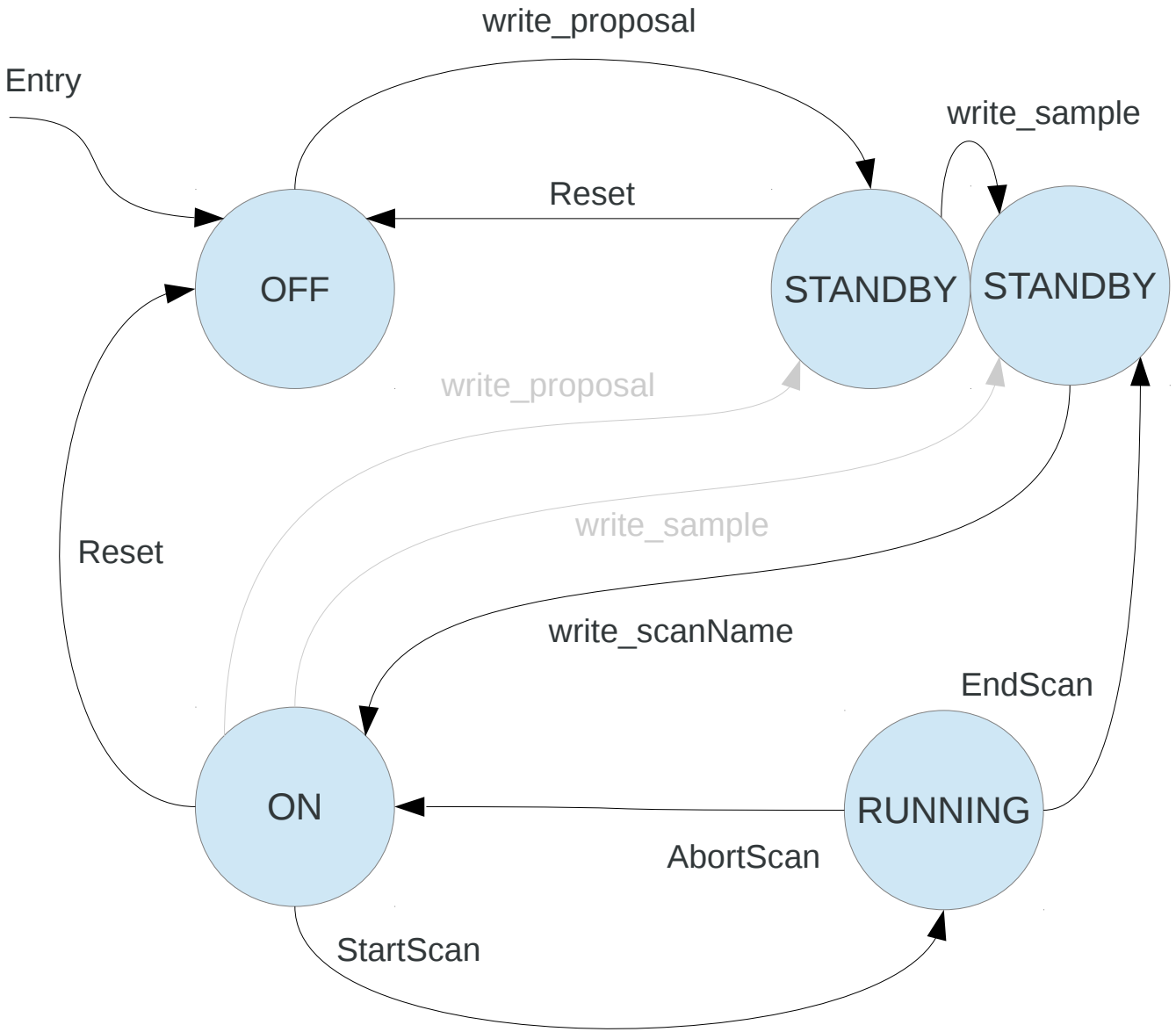
- Over 170 existing components, integration of Beans and POJO
- DSL for Java, Spring, Scala, Groovy, Annotations, Blueprint, …
- Options for deployment (standalone, EJB container, OSGi container)
- Pure Maven (archetypes, plugins), JUnit support, Spring integration

Processing details

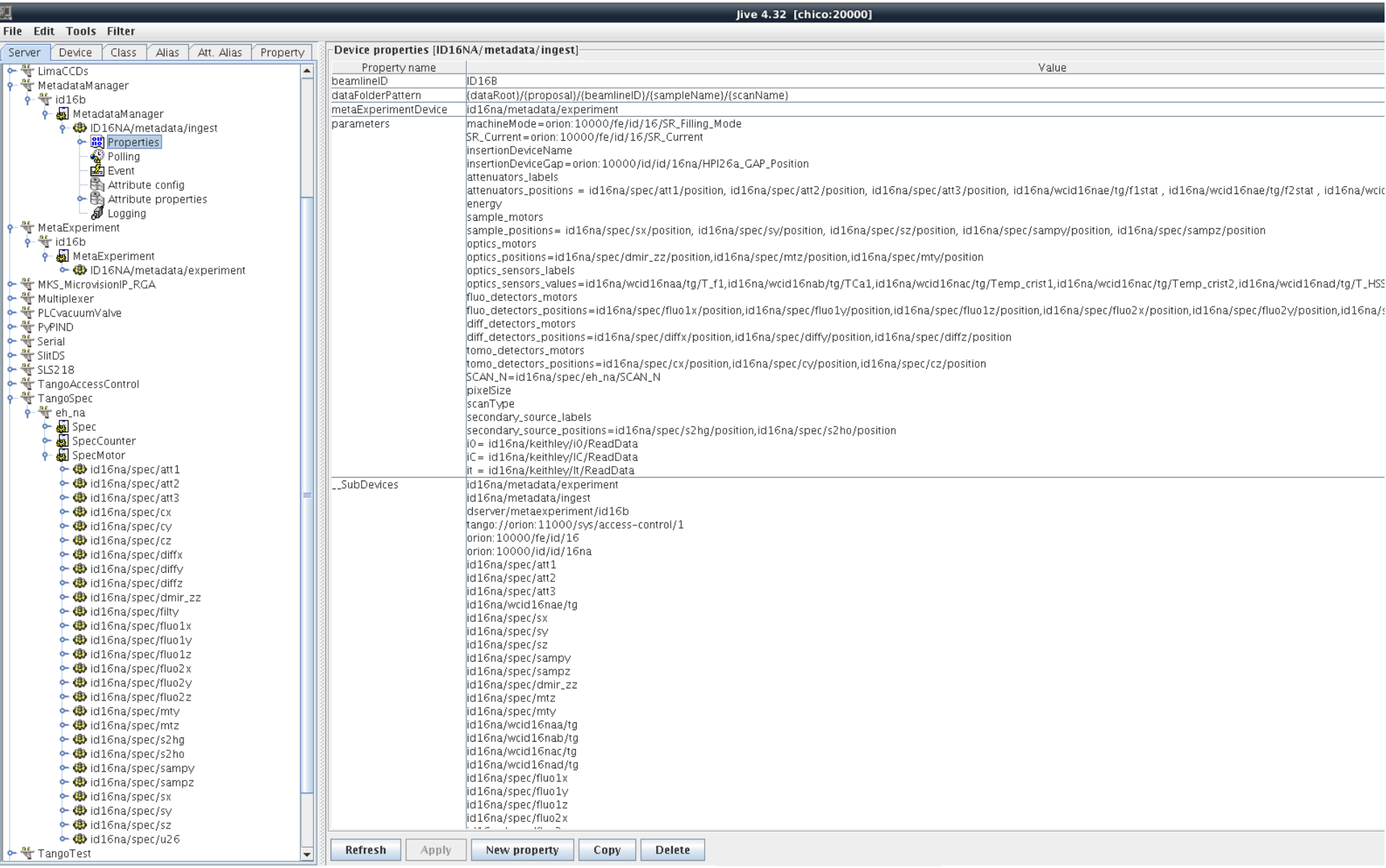



```
<camel:route id="DATASET-INGESTION">
      <camel:from uri="activemq:datasetIngest" />
      <camel:transacted />
      <camel:bean ref="fileNameConverter" method="convertSourceToDestination(Dataset)" />
      <camel:bean ref="datasetManager" method="ingest(Dataset)" />
</camel:route>
```

Tango device state machine



Beamline integration example



Example integration – ID16A cont.

- **Spec integration:**

“ Normally for a fluorescence scan/map we use the macro zapimage, that calls the macro function _zapscan.

I replaced it by a macro fluoimage that still calls _zapscan

+ takes care of opening/closing the X-ray shutter

+ takes care of saving in a way consistent with the database

+ interacts with the metadata system

It made life much, much easier with the two first user groups.

The way we work is as follows:

- Local contact sets the proposal through the DS id16ni/metadata/experiment

- User sets the sampleName through the Acquisition GUI (talking to the MetadataManager) when the sample is changed

- User sets the scanName either with the Acquisition GUI or the spec macro set_scanname (for sequencing maps, both talking to the MetadataManager)

- User selects the map to grab/resolution etc. with the Acquisition GUI, this results in a spec command "fluoimage" … ”

P. Cloetens 2014/10/16

Example run - Preparation

- Local contact sets the proposal
  - Investigation created in ICAT
  - Proposal details and access rights fetched from SMIS
  - Investigation updated in ICAT
- (Local contact sets the dataRoot for non standard proposal: BLC, IH, test)
- User sets the sample name
- User sets the scan name (must be unique for the proposal), then either
  - Device builds data folder path according to the configured pattern:

    {dataRoot}/{proposal}/{beamlineID}/{sampleName}/{scanName}

    /data/visitor/ls2307/id16a/sample_s1/sample_s1_scan_1

    (/data/id16a/inhouse1/commissioning/blc8499/id16a/sample_s1/scan_1)
  - User/system sets the data folder path to his liking

    No verification at all

Current (last) development

- **How to help configure the device on beamlines, ensure all ICAT parameters are mapped into the hdf5 files and the files follow NeXus format ?**
  - Build the hdf5 configuration from NeXus definitions
  - Assign ICAT parameters to NeXus fields/groups
  - Verify all ICAT parameters have a defined position in the file and all groups and fields are properly set
  - Select (groups of) parameters for use on the beamline from the hdf5 configuration
  - Assign Tango attributes to all selected parameter (automatically if possible: slits, insertion devices, others ?)
- **This way we have:**
  - Consistent configurations across ICAT, hdf5, beamlines
  - Consistent conversion ICAT ↔ hdf5
  - Easy way to configure hdf5 format AND devices on beamlines