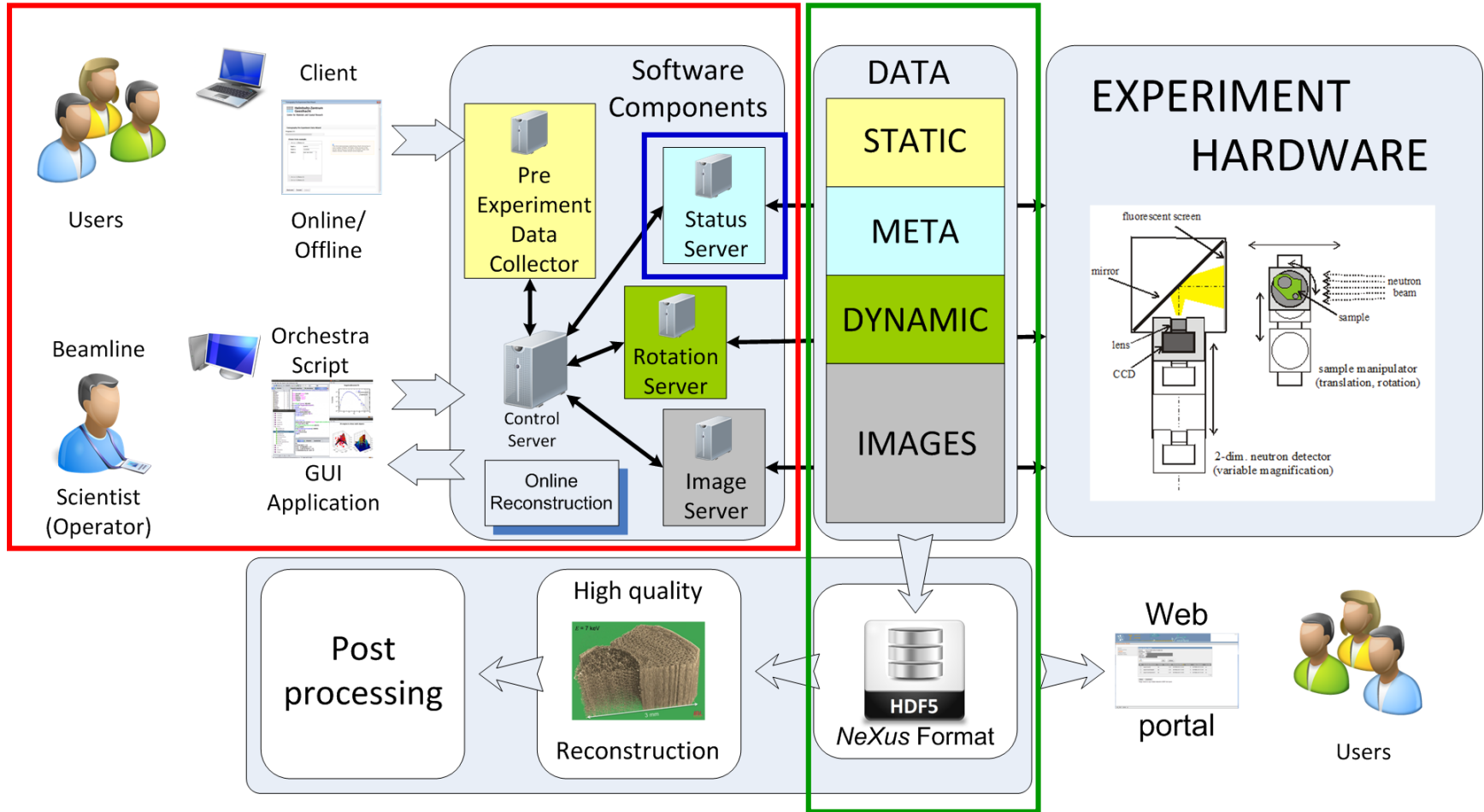


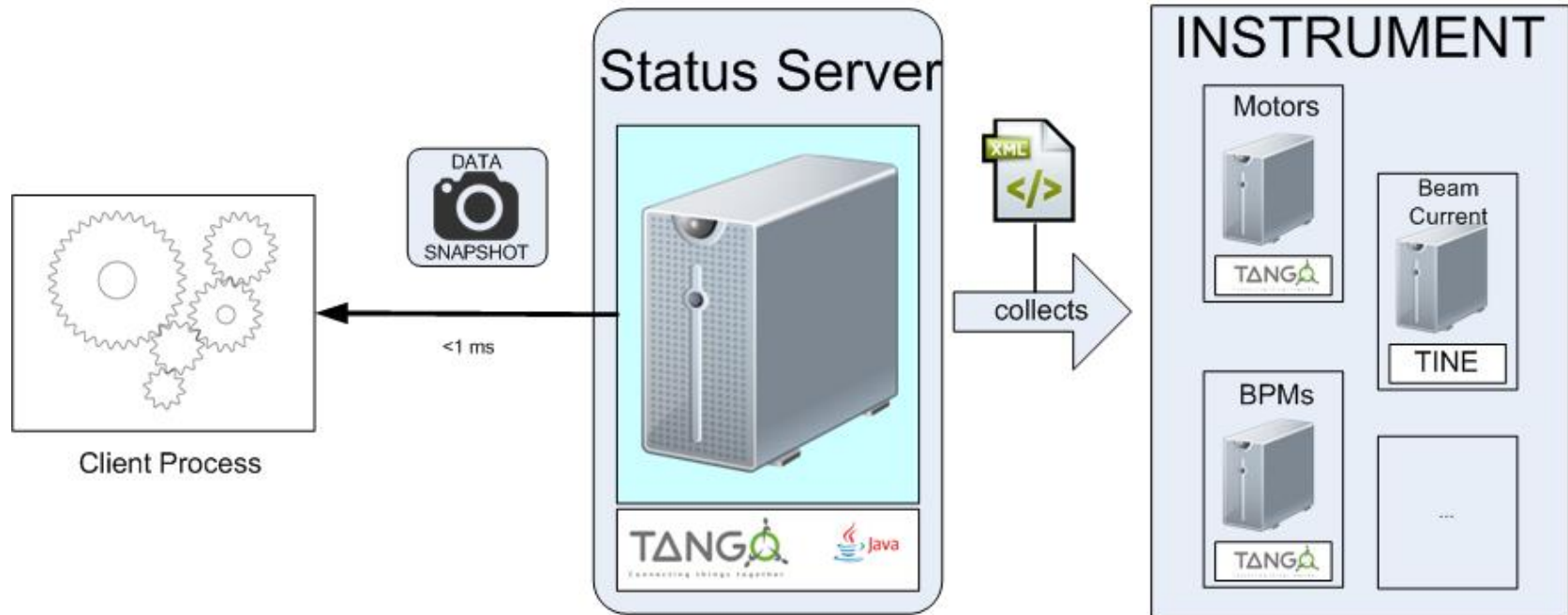
Igor Khokhriakov

**Integrated Control System Environment
for High-Throughput Tomography.
Status report.
i.e. Tango@HZG**

X-Environment



Status Server



FEATURES:

- Forms continuous timeline of the experiment
- Simple configuration, defined in XML
- Values can be obtained through event or polling mechanism
- Different interpolation strategies: last, nearest and linear

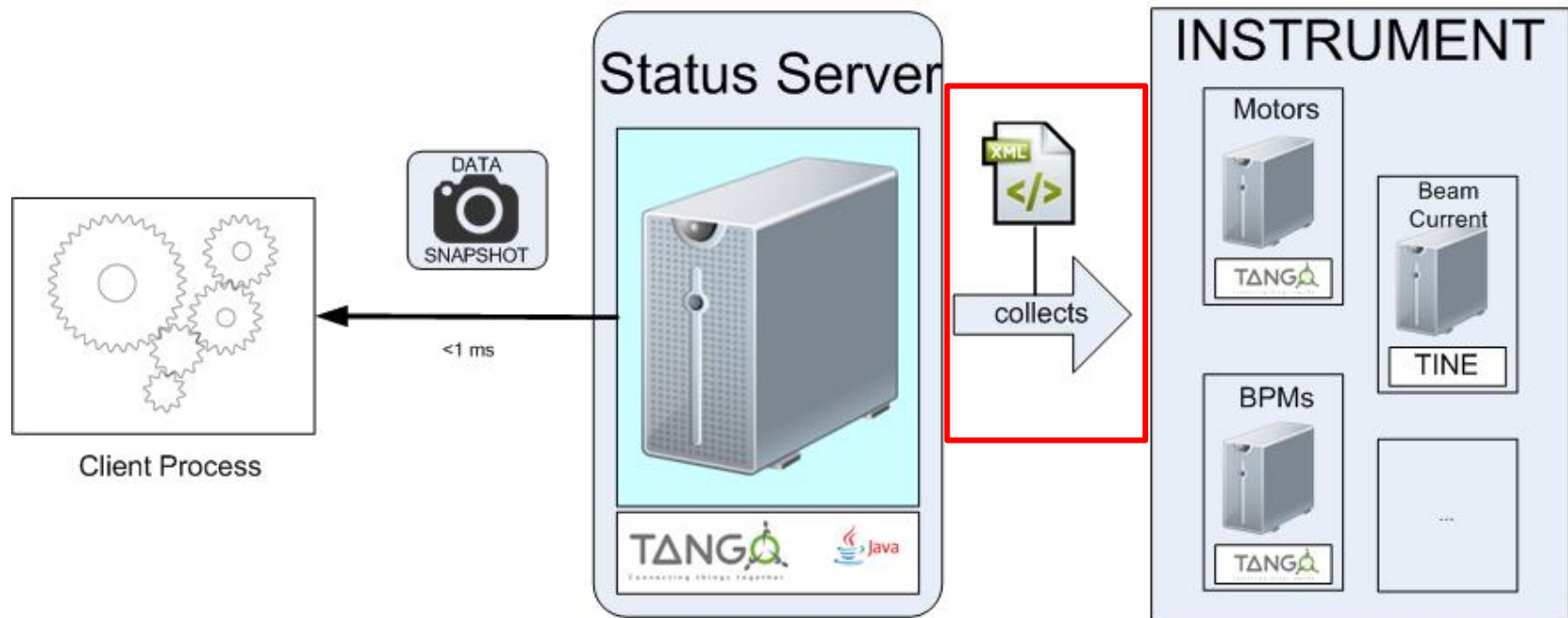
StatusServer problem #1

Random timeouts:

```
[main] INFO org.jacorb.orb - Initialising ORB with ID:
[main] INFO org.jacorb.orb.giop - ClientConnectionManager: created new
    ClientGIOPConnection to 131.169.65.104:10000 (6b927fb)
[main] INFO org.jacorb.orb.iiop.ClientIIOPConnection - Connected to
    131.169.65.104:10000 from local port 56378
[main] INFO org.jacorb.orb.giop - ClientConnectionManager: created new
    ClientGIOPConnection to 131.169.65.240:39478 (477b4cdf)
[main] INFO org.jacorb.orb.iiop.ClientIIOPConnection - Connected to
    131.169.65.240:39478 from local port 56379
[ClientMessageReceptor0] INFO org.jacorb.giop.conn - Received CloseConnection on
    ClientGIOPConnection to 131.169.65.104:10000 (6b927fb)
[ClientMessageReceptor0] INFO org.jacorb.orb.iiop.ClientIIOPConnection - Client-side
    TCP transport to 131.169.65.104:10000 closed.
[main] WARN org.jacorb.giop.conn - Abnormal connection termination. Lost 1
    outstanding replie(s)!
[main] INFO org.jacorb.orb.iiop.ClientIIOPConnection - Client-side TCP transport to
    131.169.65.240:39478 closed.
[main] ERROR org.tango.client.ez.proxy.DeviceProxyWrapper -
    DeviceProxyWrapper#executeCommand has failed.
```

StatusServer problem #2

Badly written 3rd party Tango servers:



Proposal: integrate into Tango kernel *clever* logic

https://bitbucket.org/Ingvord/tango_enhancement/wiki/Home

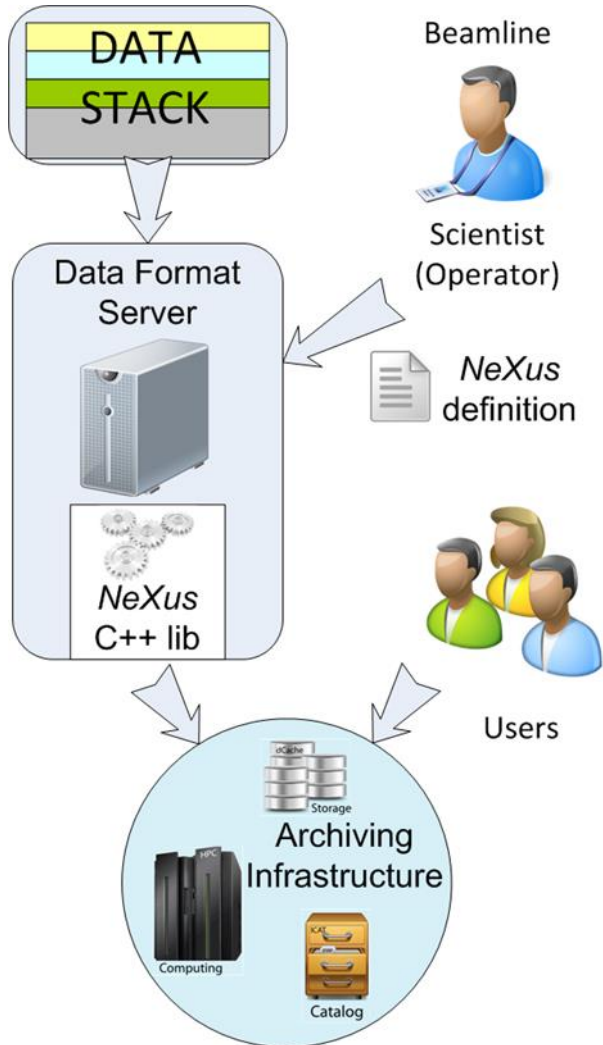
StatusServer problem #3

Manually set polling and event properties for

~50 Tango servers via Jive IS PAINFULL...

Is there a better way?

DataFormatServer



YAML NeXus definition:

```

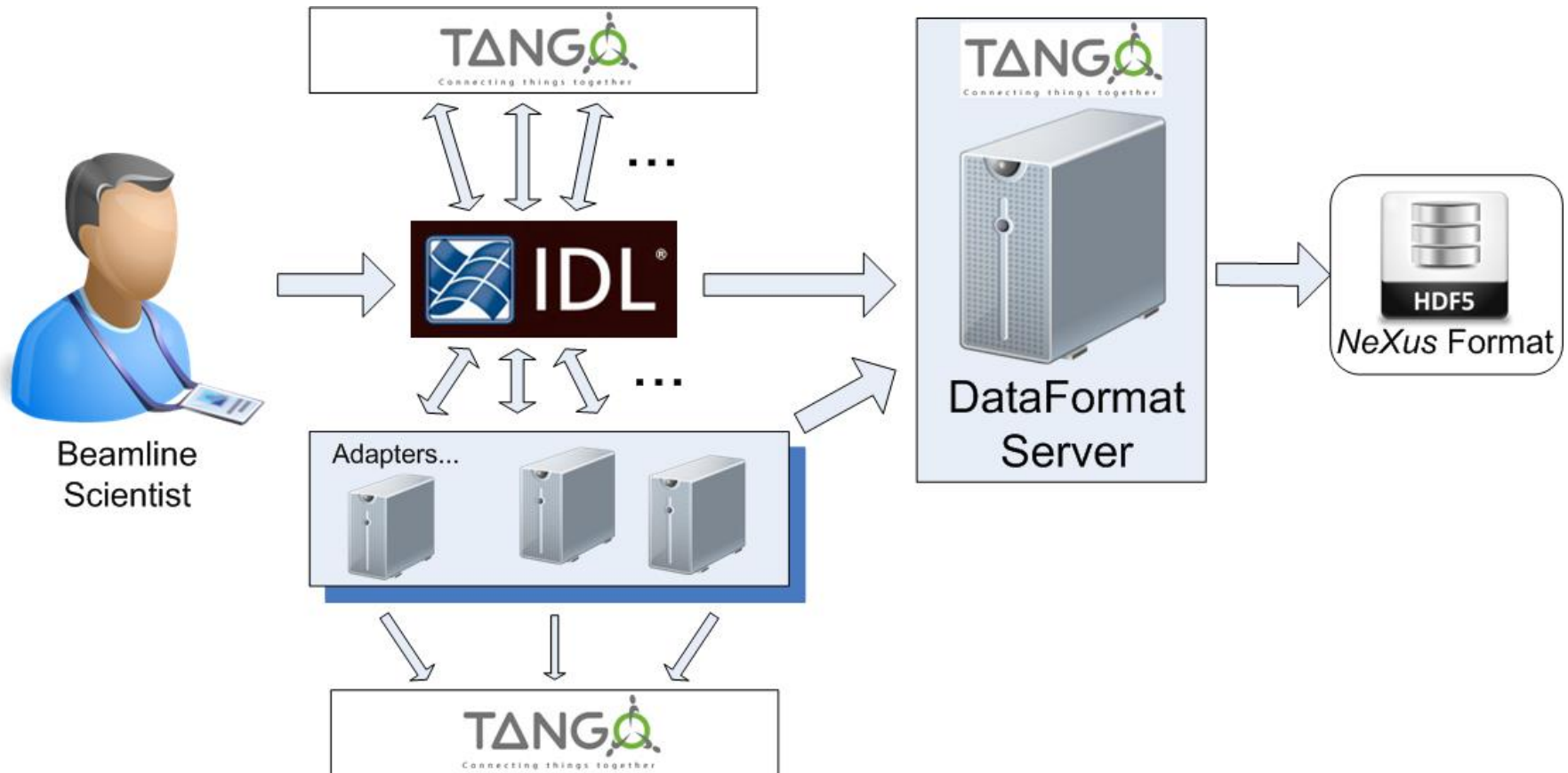
1 entry:
2   definition: NX_CHAR
3   end_time: NX_DATE_TIME
4   start_time: NX_DATE_TIME
5   title: NX_CHAR
6   data:
7     data: --> /NXentry/NXinstrument/detector:NXdetector/data
8     image_key: --> /NXentry/NXinstrument/detector:NXdetector/image_key
9     rotation_angle: --> /NXentry/NXsample/rotation_angle
10  instrument:
11    detector:
12      data: NX_INT[nFrames, xsize, ysize]
13      distance: NX_FLOAT
14      image_key: NX_INT[nFrames]
15      x_pixel_size: NX_FLOAT
16      x_rotation_axis_pixel_position: NX_FLOAT
17      y_pixel_size: NX_FLOAT
18      y_rotation_axis_pixel_position: NX_FLOAT
19    source:
20      name: NX_CHAR
21      probe: NX_CHAR
22      type: NX_CHAR
23    monitor:
24      data: NX_FLOAT[nFrames]
25    sample:
26      name: NX_CHAR
27      rotation_angle: NX_FLOAT[nFrames]
28      x_translation: NX_FLOAT[nFrames]
29      y_translation: NX_FLOAT[nFrames]
30      z_translation: NX_FLOAT[nFrames]
  
```

NXComposer:

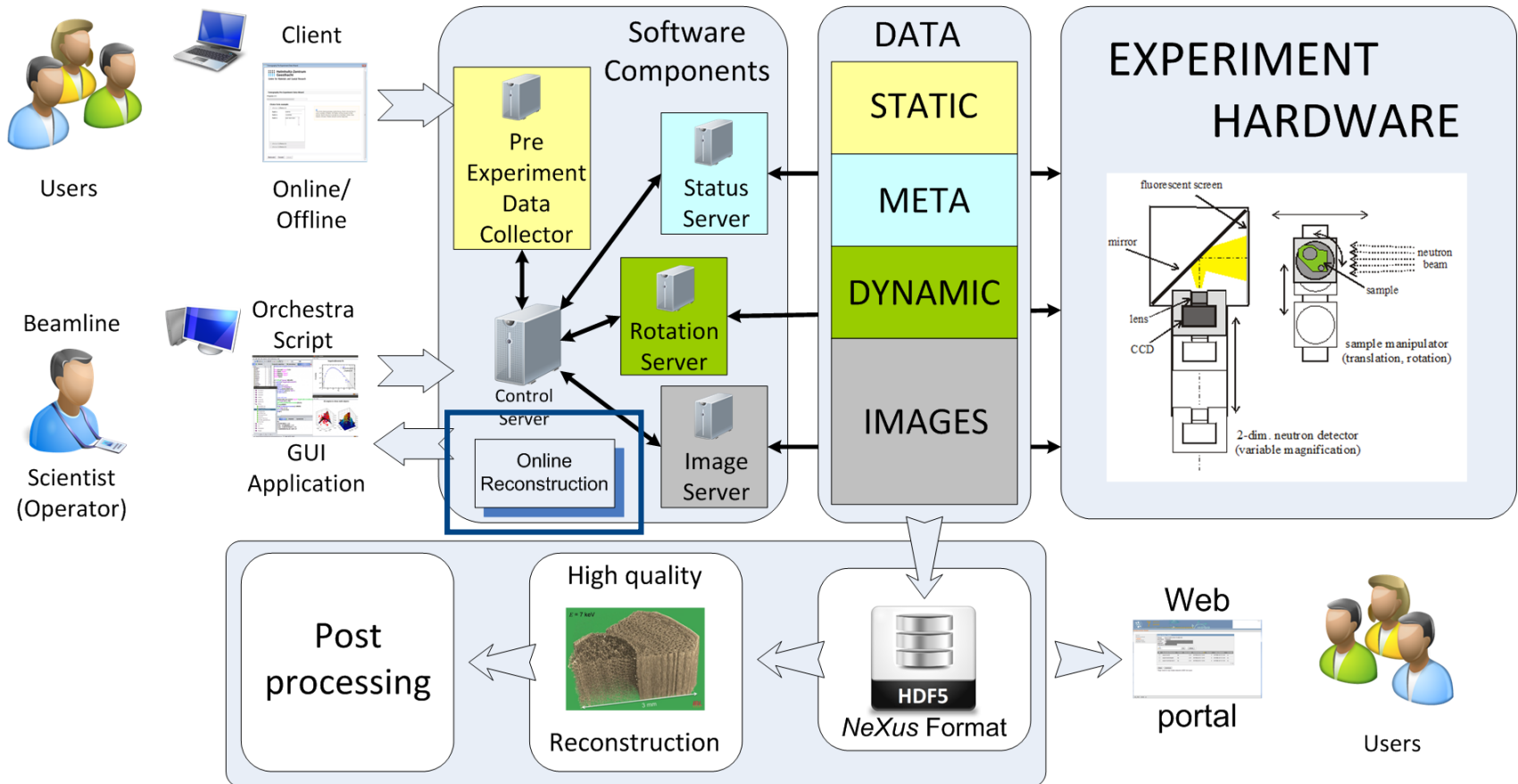
The screenshot shows the NXComposer GUI. The main window displays a tree view of the NeXus definition for 'NXtomo.nxdl'. The tree structure is as follows:

Name	Type	Value
definition: NXtomo	group	
symbols		
doc		
group: entry	NXentry	
field: title		
field: start_time	NX_DATE_TIME	
field: end_time	NX_DATE_TIME	
field: definition		
group: instrument	NXinstrument	
group:		
group: detector	NXdetector	
field: data	NX_INT	
field: image_key	NX_FLOAT	
field: x_pixel_size	NX_FLOAT	
field: y_pixel_size	NX_FLOAT	
field: distance	NX_FLOAT	
field: x_rotation_axis_pixel_position	NX_FLOAT	
field: y_rotation_axis_pixel_position	NX_FLOAT	
group: sample	NXsample	
group: control	NXmonitor	
group: data	NXdata	

DataFormatServerInfrastructure



X-Environment



CUDA real-time reconstruction server

A C++ Tango server that performs a reconstruction in parallel to the experiment.

FEAUTURES:

- Based on KIT's UFO project.
- Main goal: to give users close to real time reconstruction
- Deal with up to 1000 frames/s (1 frame = 2K x 2K)



ezTangORB

Consider the following example:

```
DeviceProxy proxy = new DeviceProxy("some-device");
DeviceAttribute attribute = proxy.read_attribute("some-attribute");
if(result.hasFailed()){
    throw new Exception("Can not read attribute.");
}
int dataFormat = result.getDataFormat();
int dataType = result.getType();
double result;
switch(dataType){
    case Tango_DEV_Double:
        switch(dataFormat){
            case _SCALAR:
                result = attribute.extractDouble();
                ...
        }
    ...
}
...
```

Using the API the same result can be achieved with the following code:

```
TangoProxyWrapper proxy = new TangoProxyWrapper("some-device");
double result = proxy.<Double>readAttribute("some-attribute");
...
```

This library will be a part of the Tango distribution since version 9.

Thank you!

Questions?

Thank you!

Questions?

Control Server [IDL Scripts]

Currently implemented using IDL2Tango Java bridge – a special java library abstracts interaction with Tango:



```
pro TangoClient
SETENV, 'IDLJAVAB_CONFIG=D:\MyProjects\IDLWorkspace8\StatusServerIDL\idljavabrc'

joDeviceProxy = OBJ_NEW("IDLJavaObject$hzg_wpn_idl_IDLDeviceProxy", "hzg.wpn.idl.IDLDeviceProxy", "sys/tg_test/1")

;wait until it is running
joDeviceProxy->waitUntil, "running"

;assume that device is in 'fault' state
;wait until it is no longer fault
joDeviceProxy->waitUntilNot, "fault"

;write/read string scalar
joDeviceProxy->writeAttribute, "string_scalar", "Hello World!!!"
string = joDeviceProxy->readAttributeString("string_scalar")
PRINT, string ;Hello World!!!

;write/read double image
joDeviceProxy->writeAttribute, "double_image", DBLARR(3, 2)
dblImage = joDeviceProxy->readAttribute("double_image")
PRINT, dblImage ;3x2 0.0

;execute command
dblArr = joDeviceProxy->executeCommand( "DevVarDoubleArray", DBLARR(3,1))
PRINT, dblArr ;three 0.0
```

In the future it is planned to have a dedicated GUI application that will also perform a quick reconstruction of the sample to provide better user experience. This will be done using UFO library from KIT

Miscellaneous

- WebCam Tango Server (published in Tango repository)
- DataMatrixDecoder Tango Server (published in Tango repository)
- PXICI 2.0 CL Frame Grabber Tango Server (not yet published)
- Android control application for Fischertechnics on DESY open days 2013

Tango meeting in München
Acquaintance with Andy Gotz → mobile

Tango meeting in Barcelona
Have a talk

Regular Lund meetings
Every time new poster

San Diego SPIE conference
Poster and publication presenting our system

Oslo JavaZone conference
Acquaintance with JetBrains company (makes professional software development) → workshop
for Tango meeting in Krakow
