PAUL SCHERRER INSTITUT

Nikhil Biyani :: Experiment Control Software Developer :: Paul Scherrer Institut

Experiment Control Project Updates and Discussion

6th ECP Workshop I 11th December 2017

Agenda

Developments

SINQ AMOR Simulation
EPICS and NICOS Integration
NICOS and Kafka Integration
NeXus File Writing in NICOS
AMOR end to end integration

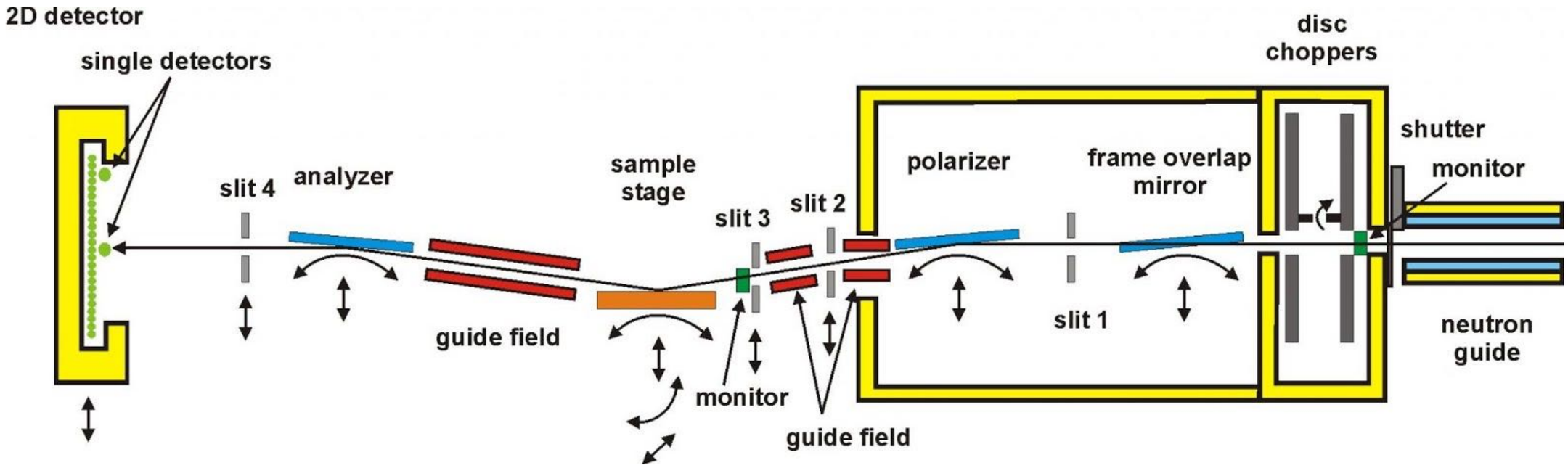
Discussions

Proposed Architectural Changes
File Writing
Ownership and Configuration management

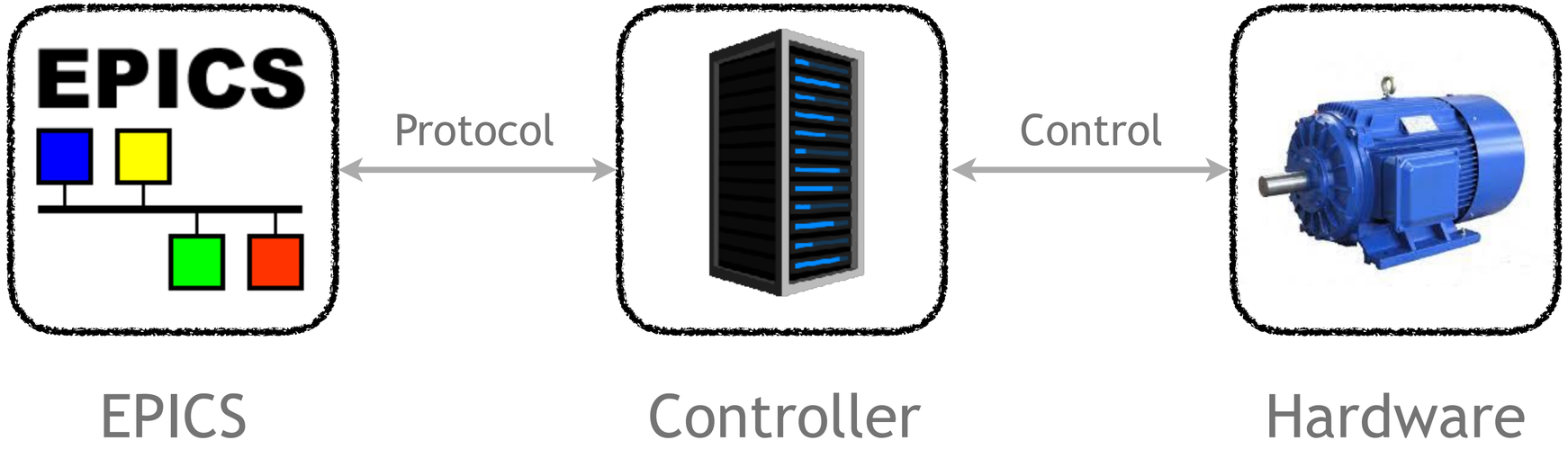
SINQ AMOR Simulation

AMOR Reflectometer at SINQ PSI

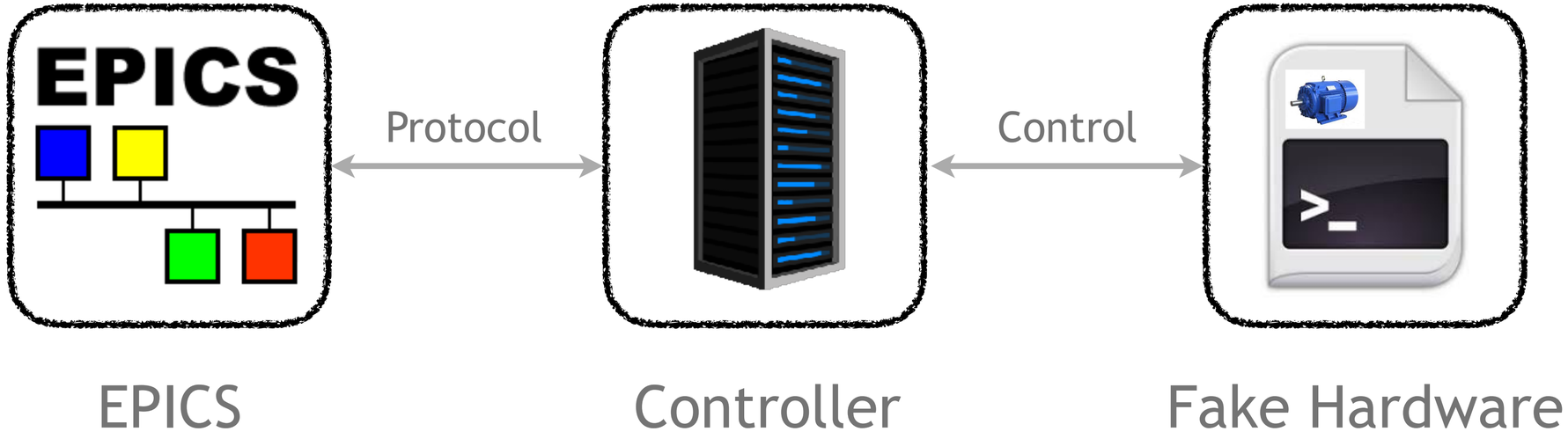
Motors | Magnets | Counter channels | Multi-disc choppers | Shutter …



SINQ AMOR Deep Simulation



SINQ AMOR Deep Simulation



Simulation Features

Includes base layer of servers which implement the **protocols of real hardware devices**

Includes simulation of **neutron event stream** from real data from an AMOR data file

Provides a way to **simulate error states** of real hardware

Provides **EPICS IOCs and PVs** which can be used to control the devices

PAUL SCHERRER INSTITUT

EPICS and NICOS Integration

EPICS base classes available in NICOS

EpicsReadable

EpicsMoveable

EpicsWindowTimeout

EpicsAsynController

...

Derived common EPICS classes

EpicsMotor

EpicsChopper

EpicsDetector

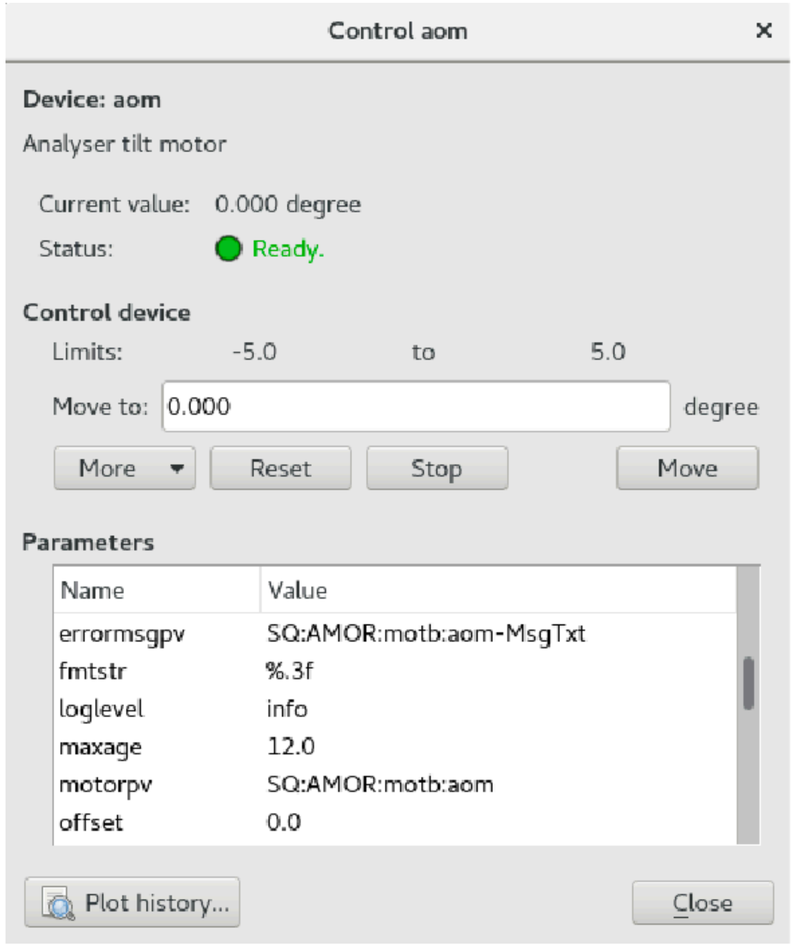
EpicsMotor

- Features incorporated and motor records from EPICS used:
  - Movement
    - RBV, VAL, STOP
  - Speed
    - VELO
  - Offset
    - OFF
  - Limits
    - HLM, LLM,
  - Status
    - DMOV, MOVN, MISS, HOMF, HOMR, LVIO, LLS, HLS
  - Reporting errors and a reset-mechanism incorporated
    - Error Message PV
    - Error Bit PV
    - Reset Error PV

EpicsMotor

Example Commands

```
aom.move(3.0)
aom.stop()
aom.offset = -1.0
aom.speed = 20
aom.read()
```



EpicsChopper

- Can control chopper with multiple discs
- Change speed of the master disc
- Change phase and speed ratio of the slave discs

- Specific for the Astrium Choppers (at SINQ):
  - Displays disc properties such as:
    - Loss Current, Vibration, Temperature, Water Flow, Vacuum, etc.

EpicsChopper

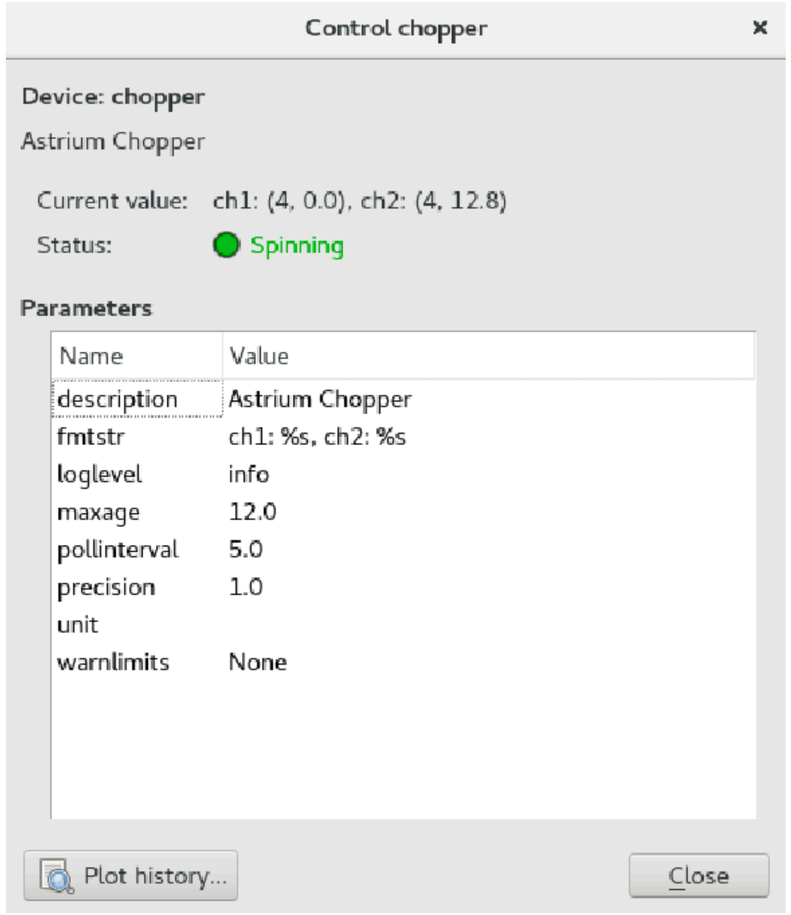
Example Commands

```
# Change speed of the master disc
chopper.chspeed(100)

# Change speed ratio for disc 2
chopper.chratio(ch2, 2)

# Change phase for disc 2
chopper.chphase(ch2, 90)

# Stop changing the speed
chopper.stop()
```



EpicsDetector

- EpicsPassiveChannel
  - Read channel count using EPICS PVs
- EpicsActiveChannel
  - Read channel count and set preset using EPICS PVs
- EpicsDetector (using the scaler record)
  - Manage multiple channels simultaneously
  - Start and stop the counting (using time or monitor preset)
  - (In context of simulation) Starts and stops the event generation

EpicsDetector

Example Commands

```
# Set a time preset of 10 sec
preset(t=10)

# Set a monitor preset of 10000
preset(n=10000)

# Start for 10 seconds
count(t=10)

# Get the value of channels
detector.read()
```

Other EPICS classes..

EpicsMagnet

EpicsJulaboController

EpicsKepcoPowerSupply

…

PAUL SCHERRER INSTITUT

NICOS and Kafka Integration

Kafka Helper classes in NICOS

- KafkaSubscriber:
  Subscribe to a topic, sends callbacks for new message arrival

- KafkaStatusTopicHandler:
  Decodes status messages from status topics, useful to check if a service is running

- ProducesKafkaMessages:
  Sends messages to Kafka topic

- EpicsKafkaForwarder:
  Wrapper for Forwarder

- NeXusFileWriterSink:
  Handles File writing using Kafka-to-nexus

Status

NICOS Cache

The NICOS Cache Daemon now uses Kafka on the backend

All historic meta data for devices: e.g. motor positions, device status, experiment information now live in Kafka

EPICS Forwarder

NICOS can configure and issue commands to the Forwarder

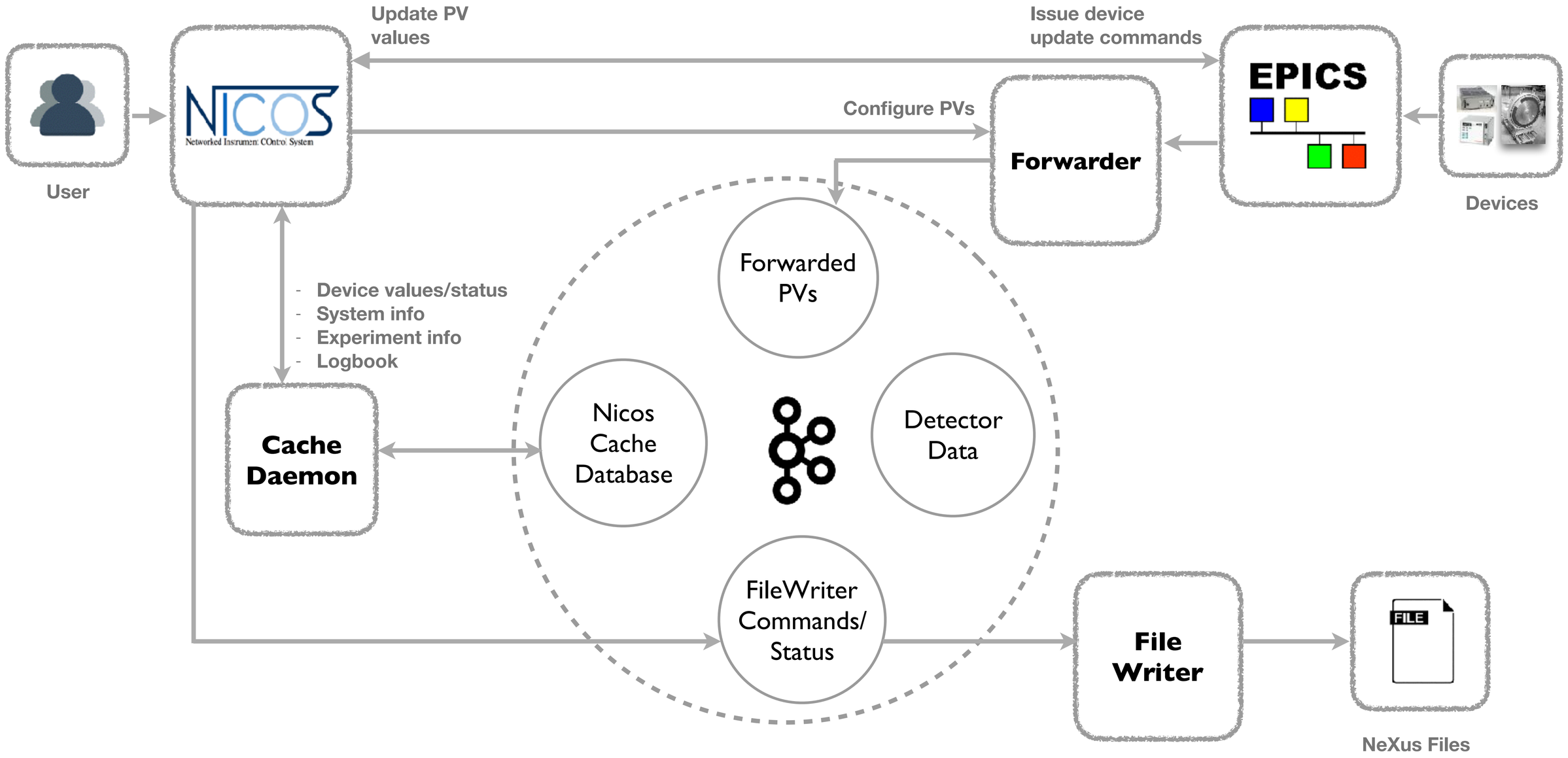
All PVs and their updates live in Kafka

NeXus File Writer

NICOS can configure and provide NeXus template to the file writer

Data is taken from Kafka and written to NeXus files

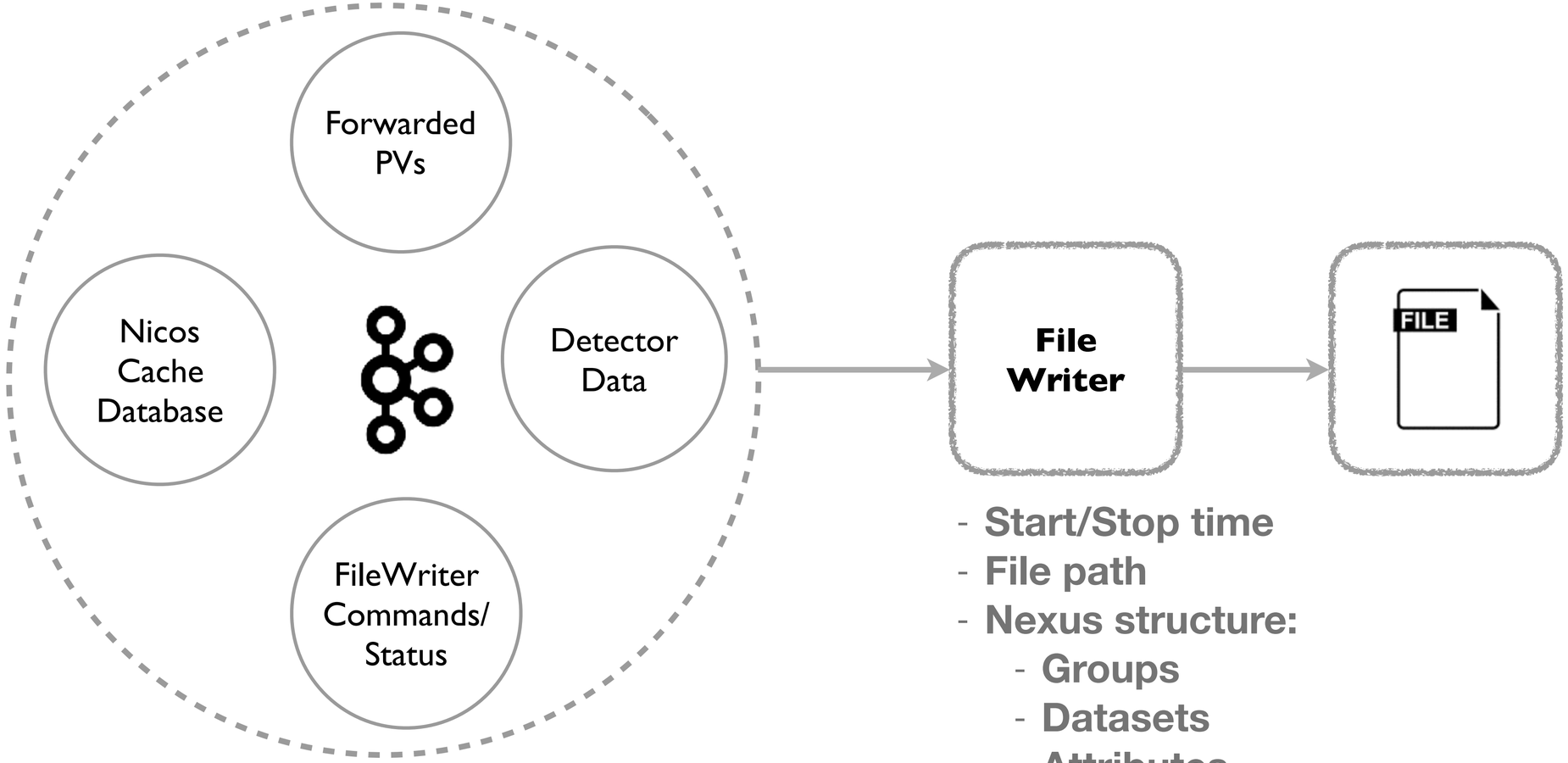
EPICS - Kafka - NICOS



PAUL SCHERRER INSTITUT

NeXus File Writing in NICOS

Setup for File Writer



- Start/Stop time
- File path
- Nexus structure:
  - Groups
  - Datasets
  - Attributes
  - Kafka Streams

Providing NeXus Structure - The challenges

- Allow multiple NeXus hierarchies
- Provide properties of static devices in NeXus structure
- Some device parameters should always be streamed from Kafka
  - E.g. Counts from detectors
- Some devices that change during the file writing should be streamed from Kafka
- Event data streaming

Solution: Nexus Templates

Python dictionary to represent hierarchy (keys mapped to one of the implemented nexus element)

NXAttribute
value
type

NXDataset
value
type
attributes

NXGroup
class
children
attributes

KafkaStream
class
topic
source
….

EventStream
(NXevent_data)
Kafka event data
configuration

DeviceStream
(NXlog)
device
Parameter

Solution: Nexus Templates with Placeholders

Allow use of non-constant values in NeXus templates using placeholders which fetch the value just before writing the file

Placeholder
Fetches the correct value when required

DevicePlaceholder
device parameter

NXAttribute
Placeholder
type

NXDataset
Placeholder
type
attributes

DeviceAttribute
DevicePlaceholder
type

DeviceDataset
DevicePlaceholder
type
attributes

Example template

```
template = {
        "entry1:NXentry": {
            "sample: NXsample": {
                "distance": NXDataset(325.0, dtype=double, some_attr=32.0),
                "height": NXDataset(DeviceValuePlaceholder(‘dev’)),
                "property": DeviceDataset('dev', 'param', unit='K'),
            },
            "INST:NXinstrument": {
                "name": NXDataset("Instrument"),
                "detector: NXdetector": {
                    "data": EventStream(topic="EventTopic", source="SrcName")
                },
                "control: NXmonitor": {
                    "mode": DeviceDataset('detector', 'mode', 'string'),
                    "preset": DeviceDataset('detector', 'preset'),
                    "monitor1": DeviceStream(‘c1'),
                },
            },
        }
    }
```

PAUL SCHERRER INSTITUT

AMOR end-to-end integration

AMOR in NICOS

NICOS – guest at localhost:1301

Application Script control Windows Tools Help

Connect Exit View Editor Scans Logbook

Experiment Setup

Proposal	p1
Title	TOF experiment
Users	Nikhil Biyani
Local Contact	nikhil.biyani@p...
Setups	analyser, chopp...
Samples	AMOR Simulation
Environments	
Detectors	psd_tof
Scans	aoz
Remark	

Instrument | Script Builder | Device History | Log files

>> count(t=10)

All output | Errors/Warnings

```
[17:10:20] creating device u'fom' (Frame overlap tilt motor)...
[17:10:20] creating device u'ftz' (Frame overlap z position of rotation axis motor)...
[17:10:20] creating device u'mom' (Tilt monochromator motor)...
[17:10:21] creating device u'moz' (Monochromator z position of rotation axis motor)...
[17:10:21] creating device u'mty' (Monochromator y position motor)...
[17:10:22] creating device u'mtz' (Monochromator z position relative to rotation axis motor)...
[17:10:22] creating device u'pby' (Polarizer magnet)...
[17:10:22] creating device u'psd_tof' (EL737 counter box that counts neutrons and starts streaming eve
[17:10:22] creating device u'timepreset' (Used to set and view time preset)...
[17:10:22] creating device u'sch' (Sample chi motor)...
[17:10:23] creating device u'serial1' (Controller of the devices connected to serial 1)...
[17:10:23] creating device u'serial2' (Controller of the devices connected to serial 2)...
[17:10:23] creating device u'serial3' (Controller of the devices connected to serial 3)...
[17:10:23] creating device u'som' (Sample omega motor)...
[17:10:24] creating device u'soz' (Sample z lift of base motor)...
[17:10:24] creating device u'stz' (Sample z translation on sample table motor)...
[17:10:24] creating device u'xlz' (Counter z position distance laser motor)...
[17:10:25] standard detectors are now: psd_tof
[17:10:25] setups loaded: startup, chopper, counter, serial_controllers, diaphragm, monochromator, sa
[17:10:54] >>> [guest 2017-11-28 17:10:54] SetNexusTemplate('template2')
[17:10:54] NexusDataSink: Finished setting template
[17:11:10] >>> [guest 2017-11-28 17:11:10] count(t=10)
```

Control devices

Filter:

Name	Value	Status
moz	0.000 mm	Ready.
mty	0.000 mm	Ready.
mtz	0.000 mm	Ready.
polarizer		
pby	0.000 A	Off
sample		
fma	0.000 A	Off
sch	0.000 degree	Ready.
som	0.000 degree	Ready.
soz	0.000 mm	Ready.
stz	0.000 mm	Ready.
serial_controllers		
cter1		
serial1		
serial2		
serial3		
system		
Amor		
Exp		
KafkaForwarder		Forwarding..
NexusFileWriter		Writing 1 files
Sample	AMOR Simulation	
Space	61.814 GiB	61.81 GiB free

Status

- Most of the SINQ AMOR devices integrated
- Unit tests for various devices integrated using pytest
- EPICS forwarder automatically configured on NICOS startup
- File Writing automatically started when the counting is turned on:
  - All devices meta-data written in files
  - Detector data written
  - Devices that move when beam is on can be tracked in files
- If required, historic data can be easily rewritten in NeXus files

Control AMOR using your machine today!

Download: dm-sinq-amor
https://github.com/ess-dmsc/dm-sinq-amor

Create a VM using Vagrant (based on ICS image)

Use Ansible playbooks to setup and run everything

Start NICOS and play with devices

Live Demo

Next steps

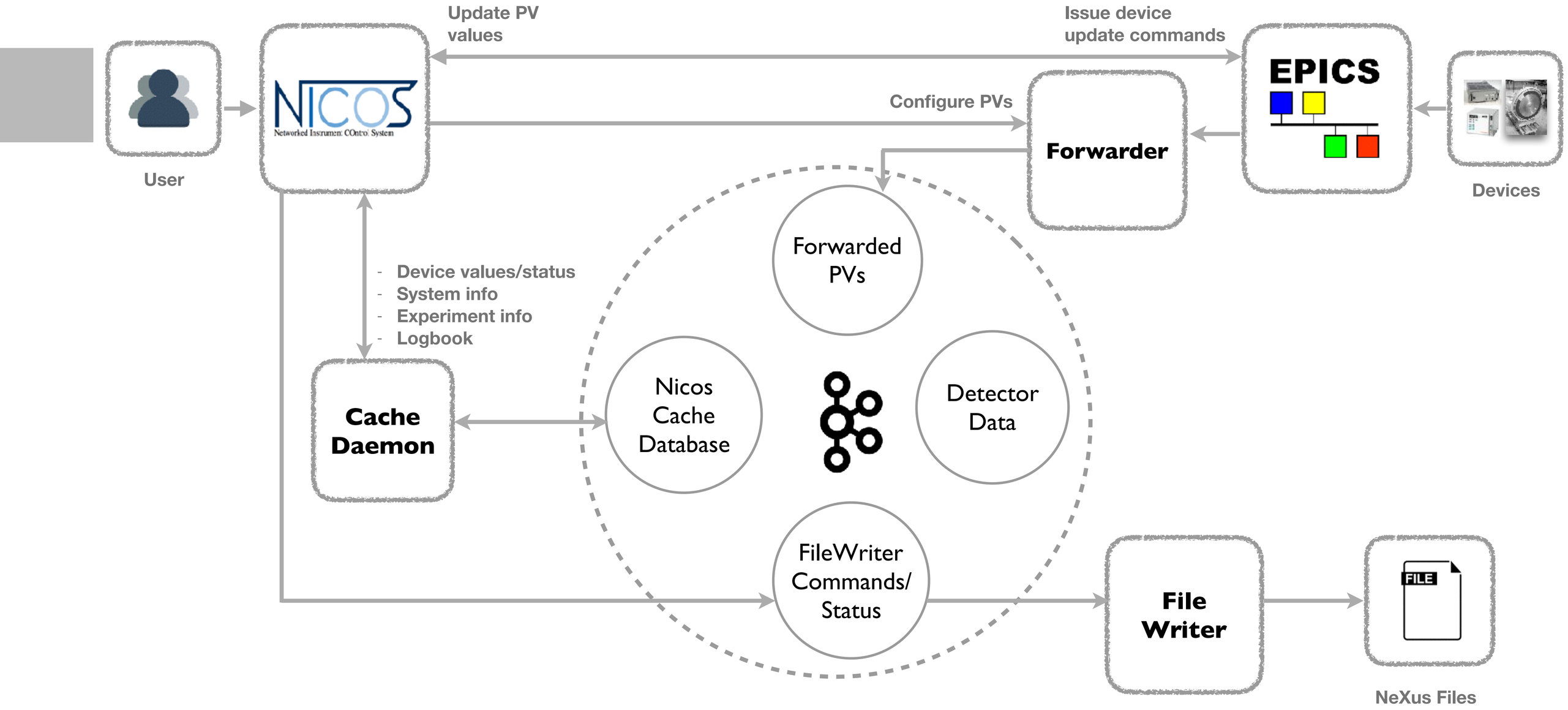
- **Virtual motors in NICOS**
  Moves various slave motors

- **Synchronise NICOS to EPICS polling**
  NICOS should be synchronised with EPICS polling and should get updates as fast as EPICS polling

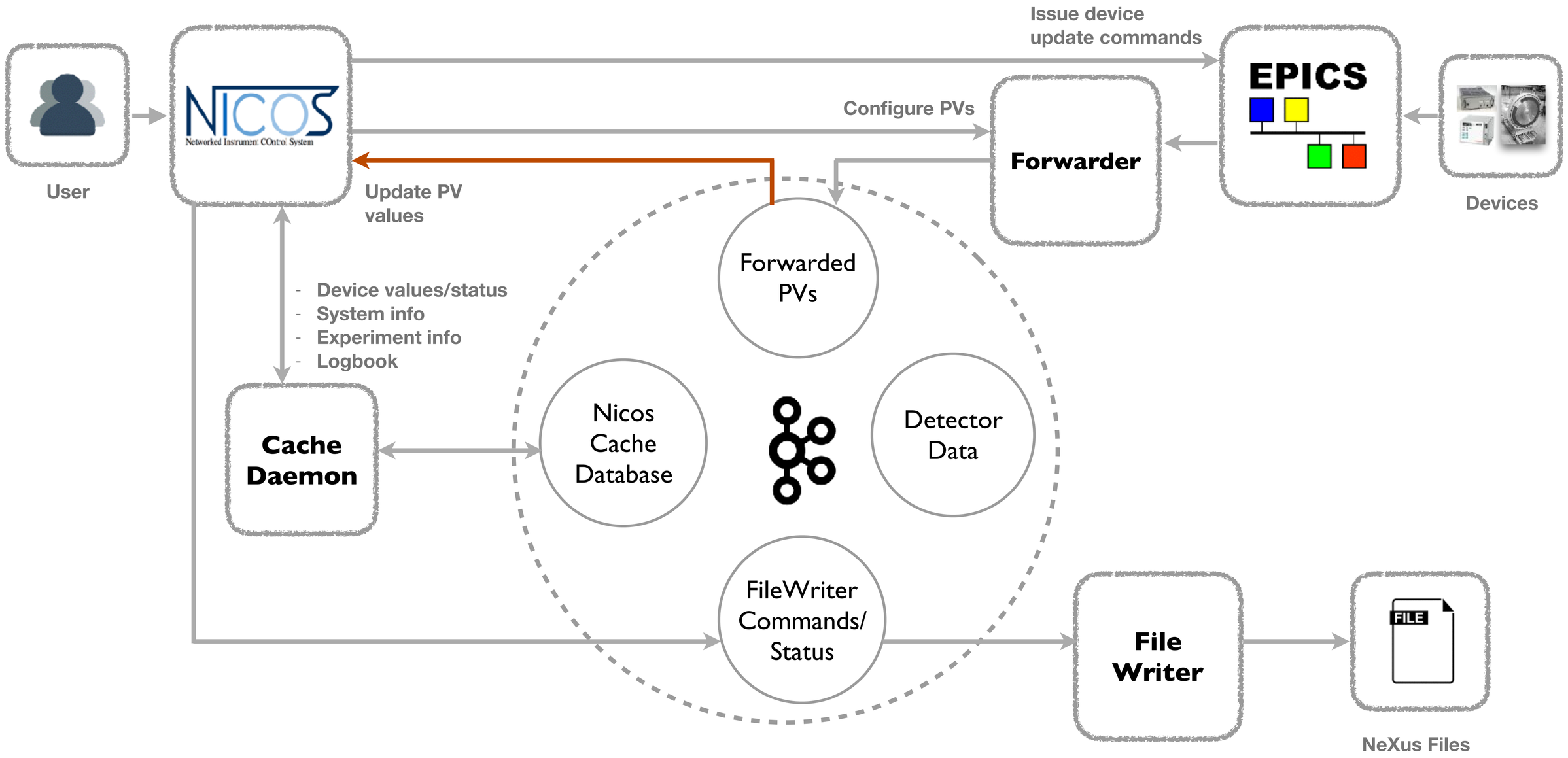
Discussions

Proposed Architectural changes

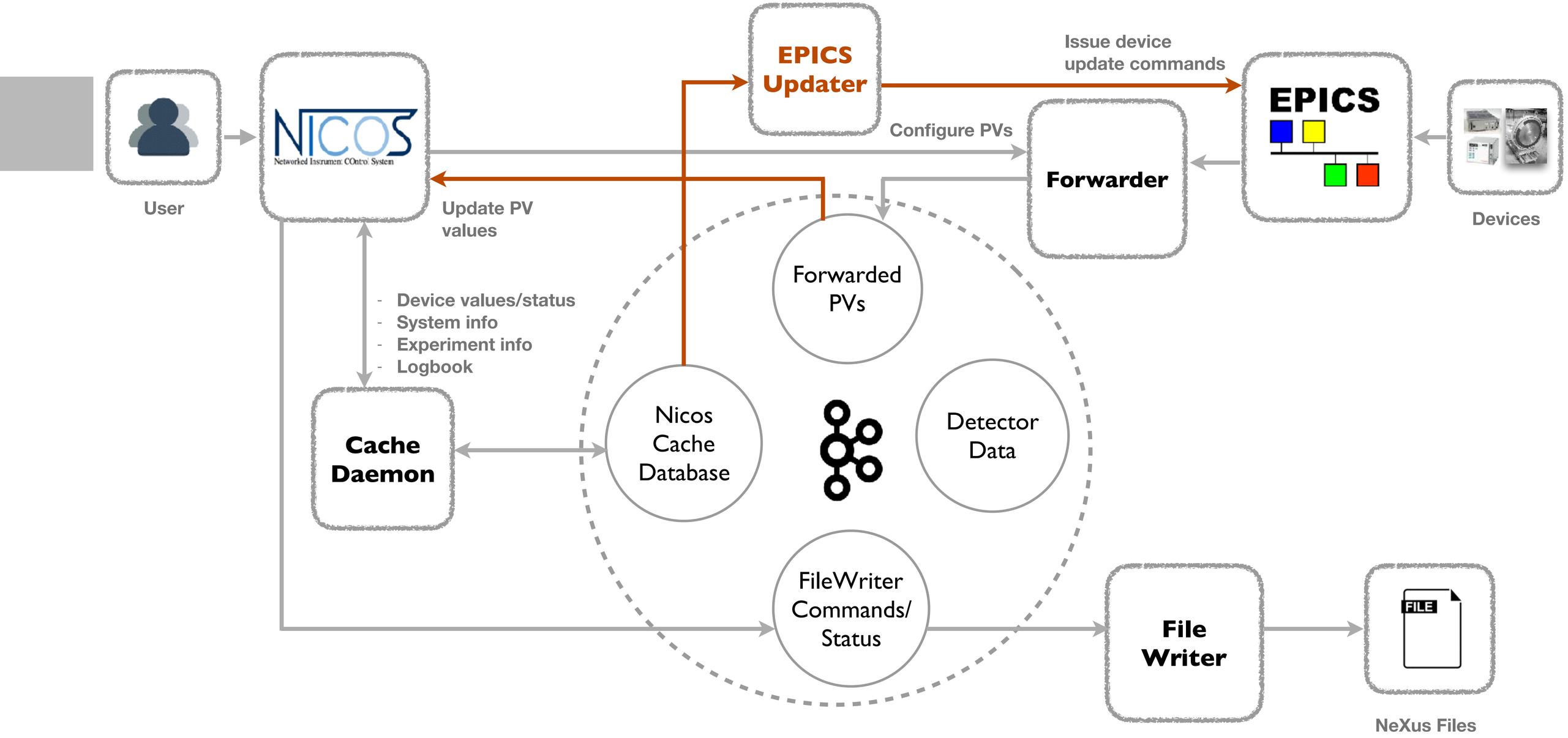
Current Architecture



Synchronise with EPICS Polling/Forwarded PVs



Update devices values from NICOS cache



File Writing

How to get the last known value?

- The solution with NICOS cache:
  - Use two topics
    Log compacted topic for current values
    Normal topic for saving history

Poll every n seconds

New update command dumps the values of the PVs to the topics

NICOS provides the value in nexus structure

Write some values by default?

- Timestamp units from streams?
- Start and stop time (can’t be provided with nexus structure)

Ownership and Configuration management

Configuration management

- Who manages Kafka brokers and topics?
- How does NICOS know about the detector topics?

Who owns/starts up the services?

- Probable order:
  - Kafka
  - IOCs/Detector data
  - Forwarder/File Writer
  - NICOS services
  - UI

- What happens if some service crashes?

Thanks to..

- Colleagues at PSI
- Instrument Data Team
- Data Management Team
- NICOS FRM II Team

PAUL SCHERRER INSTITUT

Questions?