



Karlsruher Institut für Technologie



KARA
Karlsruhe Research Accelerator



State of Controls at the KIT Accelerator Facilities

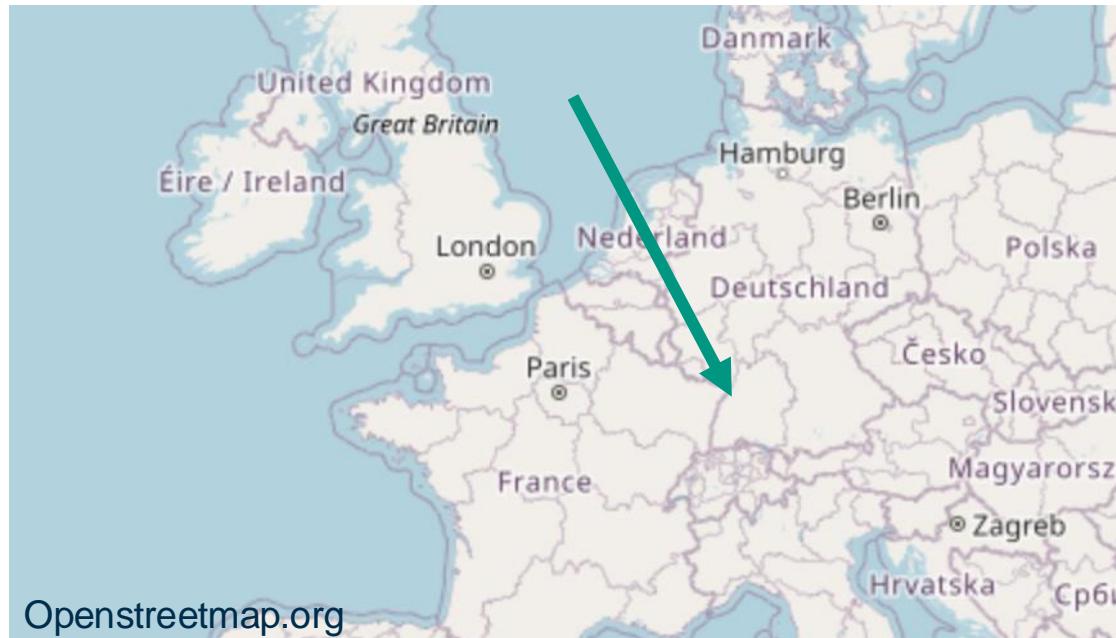


- 1. KIT & Accelerators**
- 2. Controls Environment**
- 3. Hardware**
- 4. Miscellaneous**
- 5. Outlook**

Karlsruhe Institute of Technology

Merge of University and Research Center

- **Founded 2009**
- **~ 10,000 employees**
- **~ 22,000 students**
- **2025: 200 years of KIT**



Institute for Beam Physics and Technology

- **Founded 2016**
- **Split of institutes to operate “KIT Synchrotron”**
 - **(Partner institute operates beamlines)**
- **~ 70 employees**
- **~ 2 people working on controls**
 - **(+ external contractors)**

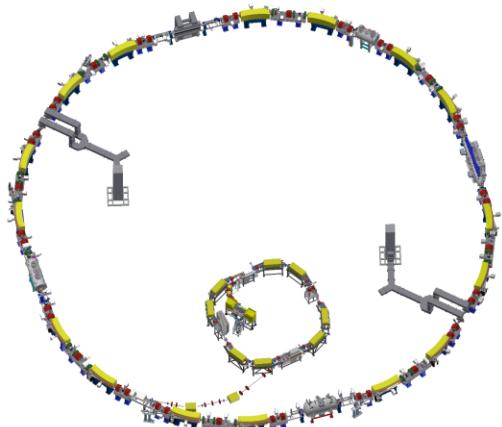


Campus North (former Research Center Karlsruhe)

KIT / IBPT Accelerators

KARA

- KArlsruhe Research Accelerator
- First beam 2001
- 0.5 – 2.5 GeV **storage ring**
- 110.4 m circumference
- Migrated 2012-2022 to EPICS*
- (By now) accelerator test facility
- 50 days for accelerator studies
- Research on short bunches (~1 ps)



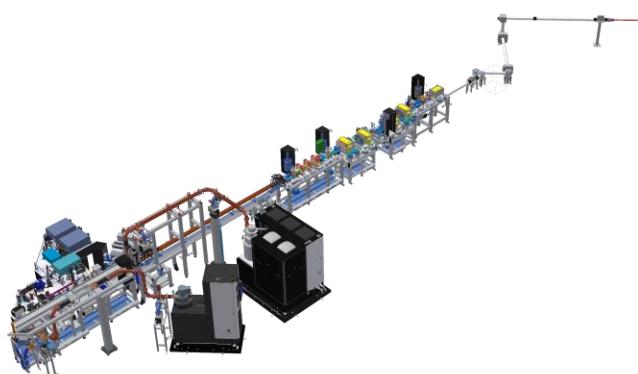
4

08.04.2025

Edmund Blomley - State of Controls at KIT Accelerators

FLUTE

- 50 MeV **linac**
- > 20 m
- Up to 50 Hz repetition rate
- Up to 1 nC charge
- Ultra-short bunches (~ 1 fs)
- THz radiation



cSTART

- Storage ring for non-equilibrium physics
- Lifetime < 1s
- Inject from FLUTE
- Inject from Laser Plasma Accelerators
- Will be installed 3 m above FLUTE
- **Commissioning 2027**



*The Decade Long Transition to EPICS, Collab. Meeting, Fall 2023



Controls Environment

- **Ubuntu** LTS (rolling out 24.04)
- Most servers **virtual machines**
- IT orchestration via **Salt**
- **EPICS 7**
 - Custom build pipeline
 - Debian packaging
 - Allows for local patching
 - Pick and choose any module (version or commit)
- Custom archiver appliance using (NoSQL) **Apache Cassandra**
- Control System Studio (in transition to **Phoebus**)
 - Details see talks last two collab. meetings
 - Panels served via webserver
- Recent additions:
 - **Kibana** + **Filebeat** stack for (IOC) logging
 - **ChannelFinder** (without recCaster)
- **Important:** Access to EPICS and panels from Office network (read-only)

EPICS build configuration

```
EPICS_MODBUS_VERSION="3.4"
EPICS_MODBUS_ARCHIVE_URL="https://github.com/epics-modules/modbus/archive/R3-4.1"
EPICS_MODBUS_DEPEND_MODULES="asyn"

EPICS_MOTOR_VERSION="7.3.1"
EPICS_MOTOR_ARCHIVE_URL="https://github.com/epics-modules/motor/archive/R7-3-1.1"
EPICS_MOTOR_DEPEND_MODULES="asyn busy ipac seq"
EPICS_MOTOR_SUB_MODULES="motorGDCBell motorMotorSim motorNewport motorOms motorOsc"

EPICS_MOTOR_MOTORMOTORSIM_VERSION="1.2"
EPICS_MOTOR_MOTORMOTORSIM_ARCHIVE_URL="https://github.com/epics-motor/motorMotor"

EPICS_MOTOR_MOTORNEWPORT_VERSION="1.2.1"
EPICS_MOTOR_MOTORNEWPORT_ARCHIVE_URL="https://github.com/epics-motor/motorNewport"

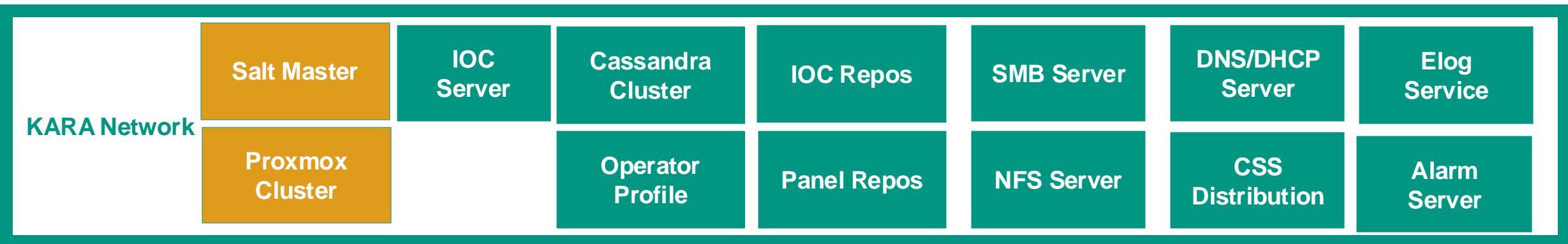
EPICS_MOTORSYMETRIE_VERSION="20230526"
EPICS_MOTORSYMETRIE_GIT_COMMIT="f5769fe11c70c7fb0a963470e326aaade3d621f9"
EPICS_MOTORSYMETRIE_ARCHIVE_URL="https://github.com/epics-motor/motorSymetrie/archive/20230526"
EPICS_MOTORSYMETRIE_DEPEND_MODULES="asyn motor pmac"
```

EPICS modules in use (**base-7.0.9**)

areaDetector-3.14	iocLock-1.0.0	opcua-0.11.0	seq-2.2.9
asyn-4.45	iocStats-3.2.0	open62541-1.2.0	snmp-1.1.0.4
autosave-5.11	ip-2.22	pcas-4.13.3	sscan-2.11.6
bacnet-2.2.2	ipac-2.16	picoharp-20221202	std-3.6.4
busy-1.7.4	modbus-3.4	pmac-2.6.5	stream-2.8.26
ca-gateway-2.1.3	motor-7.3.1	pyDevSup-1.2	
calc-3.7.5	motorSymetrie-20230526	s7nodave-3.0.2	
execute-1.1.6	mrf-1.4.1	s7plc-1.5.2	

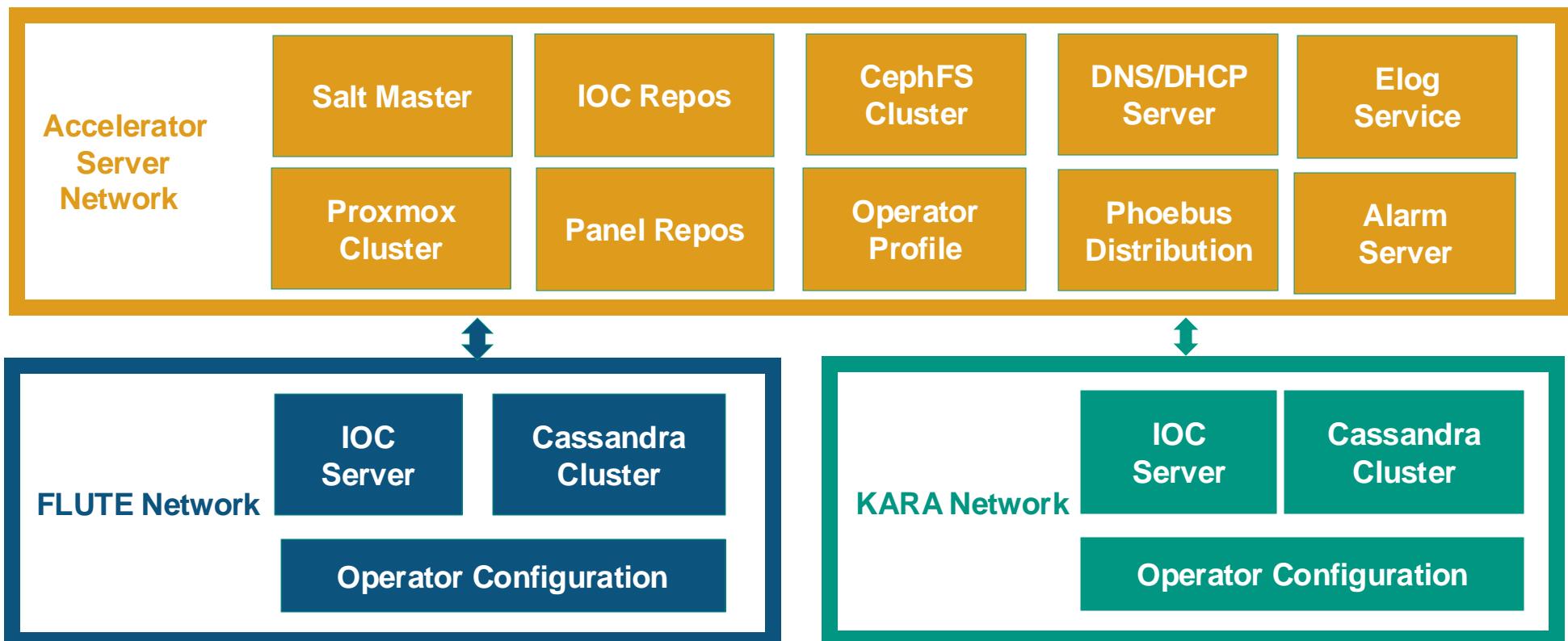
Historic Setup of Controls Services

- Separate networks running same/similar services, with different people
- Setting up services simple due to Salt, but maintenance, upgrading, configuring firewall rules, adding new features, ..., is/was time-consuming



Future Setup of Controls Services

- Central accelerator network services for both accelerators
- Everything is shared by default
 - For example IOCs used for both accelerators same source with two startup files
- Less hardware
- Less maintenance
- Less firewall management
- Less costs
- Introduce CephFS
- Less independence(!)

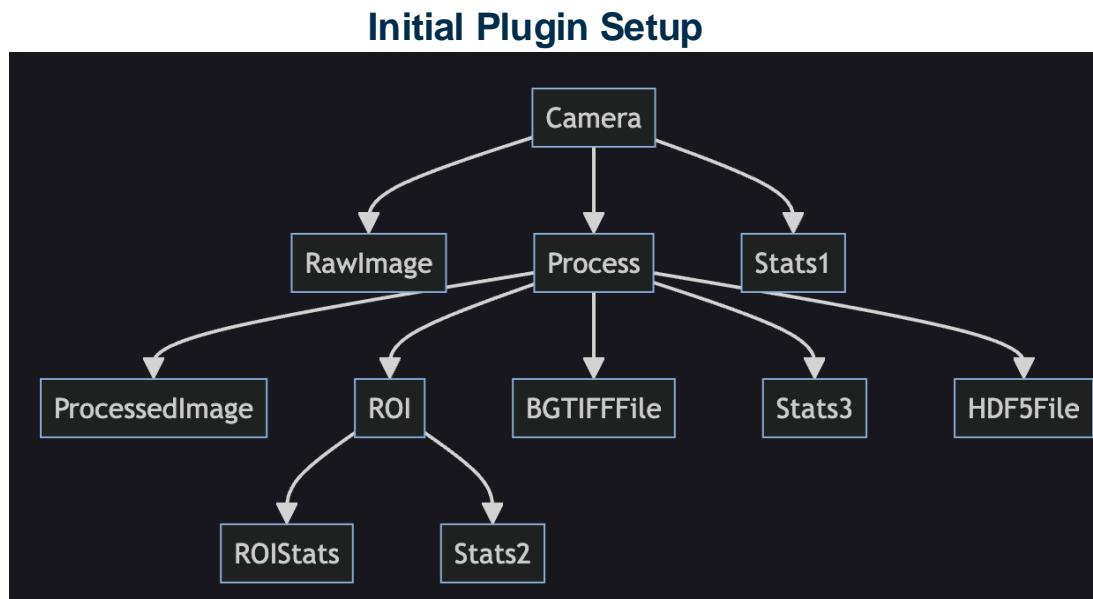


Transition to Digital Cameras

- KARA was mostly using analog cameras
 - One digital Basler camera for students projects
- FLUTE was built with digital cameras only
 - Originally only handful
- Raw image from EPICS
- All post-processing & analysis using custom Python scripts
 - Developed by bachelor/master students and PhDs
 - KARA and FLUTE scripts developed independently

Replaced all cameras with GigE (Basler)

- KARA now has ~20 & FLUTE ~ 10 digital cameras
- Switch to areaDetector + ADGeniCam + Plugins
 - Replaces ~70-80% of custom Python processing
- → need(ed) to learn how most plugins work and convince people to make use of it



```
#####
## EPICS R7.0.9
## Rev. 2025-02-28T10:55+0000
## Rev. Date build date/time:
#####
iocRun: All initialization complete
# Create autosave files for IOCStats
makeAutosaveFileFromDbInfo("iocStats_settings_kara_fs", "autosaveFields_pass0")
create_monitor_set("auto_settings_kara_fs.req", 60)
create_monitor_set("iocStats_settings_kara_fs.req", 60)
epics> iocStats_settings_kara_fs.sav: 63 of 63 PV's connected
auto_settings_kara_fs.sav: 14896 of 14896 PV's connected
```

IOC with 14 cameras (only one is active at a time)

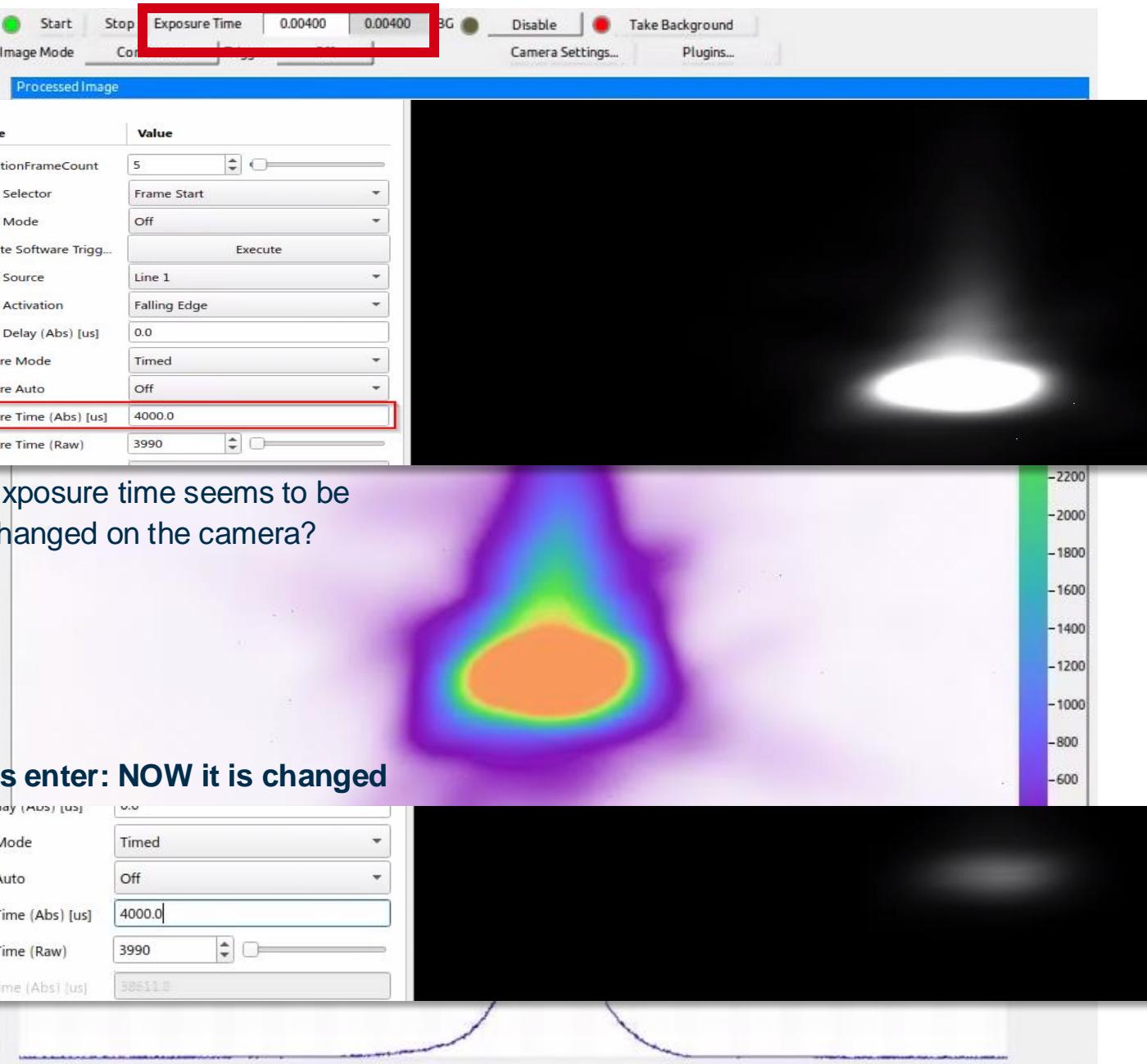
Digital Cameras

Next steps

- Finish user panels
- Default plugin configuration for diff. use cases
- New 50 Hz GigE5 cameras
- Data output & storage

Annoying Issues

- Occasional camera freezes
 - IOC or camera restart necessary
- Exposure time cannot be properly controlled via EPICS
 - Set and readback changes, but no effect
 - Workaround is to set exposure time via manufacturer software
 - Try with ADPylon(?)



PLCs & OPC-UA

- Mostly **Siemens S7-1500** (and some older versions)
 - Both for critical and non-critical general purpose applications
- Used S7 EPICS driver (s7nodave)
- Created module with open62541 support around 2018
 - <https://github.com/KIT-IBPT/epics-open62541>
 - 1:1 mapping of s7 protocol, no PLC structures
- epics-module/opca-ua 0.10.0 added open62451
 - Ralph created a compatibility layer for testing
 - Automatically **embedding** open62541 lib during .deb building
 - Migrated first couple of IOCs to use **0.11.0** without issues
 - One IOC running 10 subscriptions and 2500 data points

Comment in an IOC using our old driver

- Now could come up with some proper tests for benchmarking / comparing CPU loads, subscription rates, etc...
- Next steps: learn the linking syntax...

Showing 2 changed files ▾ with 3 additions and 3 deletions

EnergyDiagnosticsPlcApp/src/Makefile

```
... ... @@ -17,11 +17,11 @@ EnergyDiagnosticsPlc_DBDB += bas  
17 17  
18 18 # Include dbd files from all support applications:  
19 19 EnergyDiagnosticsPlc_DBDB += asyn.dbd  
20 - EnergyDiagnosticsPlc_DBDB += open62541.dbd  
20 + EnergyDiagnosticsPlc_DBDB += opcua.dbd  
21 21  
22 22 # Add all the support libraries needed by this IOC  
23 23 EnergyDiagnosticsPlc_LIBS += asyn  
24 - EnergyDiagnosticsPlc_LIBS += open62541  
24 + EnergyDiagnosticsPlc_LIBS += opcua  
25 25  
26 26 # EnergyDiagnosticsPlc_registerRecordDeviceDriver.c  
27 27 EnergyDiagnosticsPlc_SRCS += EnergyDiagnosticsPlc_
```

✓ configure/RELEASE

```
...     ...     @@ -24,7 +24,7 @@
24     24     # Variables and paths to dependent modules:
25     25     MODULES = /opt/epics/modules
26     26     ASYN = $(MODULES)/asyn-4.45
27     - OPEN62541 = $(MODULES)/open62541-1.2.0
27     + OPCUA = $(MODULES)/opcua-0.11.0
28     28
```

Hardware – Honorable Mentions

CaenELS Power Supplies with Embedded EPICS 7

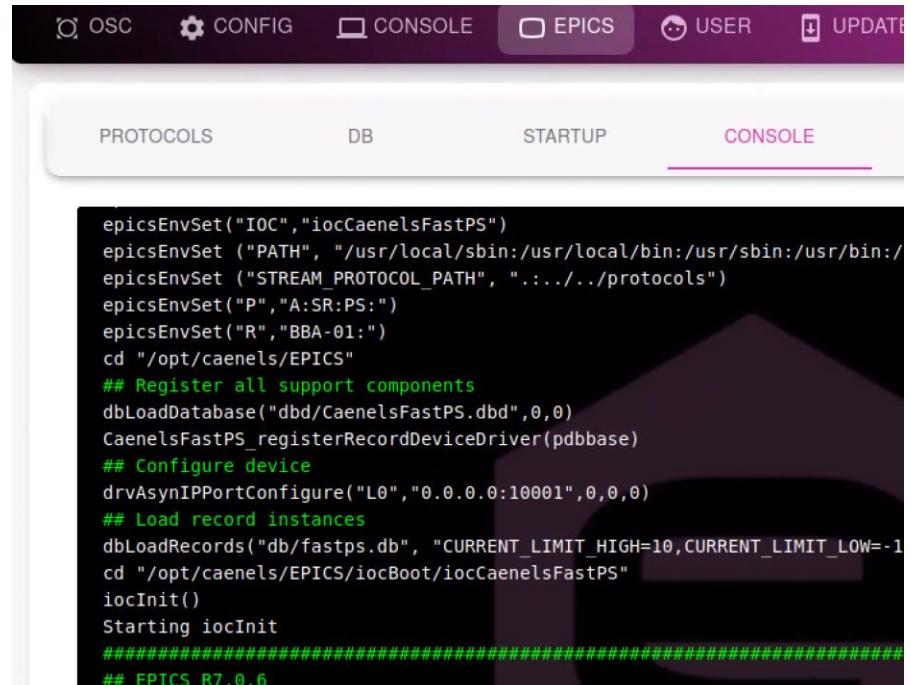
- New main PS for KARA (and cSTART)
- Well done web interface with EPICS integration
- Allows to overwrite protocol and db files via scripts

MRF Timing EPICS Module via UDP

- To avoid CPU boards we developed UDP EPICS driver for MRF-230
- Also support for uTCA MRF-300
- Have now started to add MRF-300 VME-EVR/EVM, should be ready this summer
- <https://github.com/KIT-IBPT/epics-mrf>

EPICS Driver for ELMO Motion Controller GDCBell

- New motor module for single axis stepper motor
- Challenge: learning about details of EPICS motor and general motion control at the same time



```
epicsEnvSet("IOC","iocCaenelsFastPS")
epicsEnvSet ("PATH", "/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/")
epicsEnvSet ("STREAM_PROTOCOL_PATH", ".:/..:/protocols")
epicsEnvSet("P","A:SR:PS:")
epicsEnvSet("R","BBA-01:")
cd "/opt/caenels/EPICS"
## Register all support components
dbLoadDatabase("dbd/CaenelsFastPS.dbd",0,0)
CaenelsFastPS_registerRecordDeviceDriver(pdbase)
## Configure device
drvAsynIPPortConfigure("L0", "0.0.0.0:10001",0,0,0)
## Load record instances
dbLoadRecords("db/fastps.db", "CURRENT_LIMIT_HIGH=10,CURRENT_LIMIT_LOW=-1")
cd "/opt/caenels/EPICS/iocBoot/iocCaenelsFastPS"
iocInit()
Starting iocInit
#####
## EPICS R7.0.6
```

CaenELS EPICS console webinterface



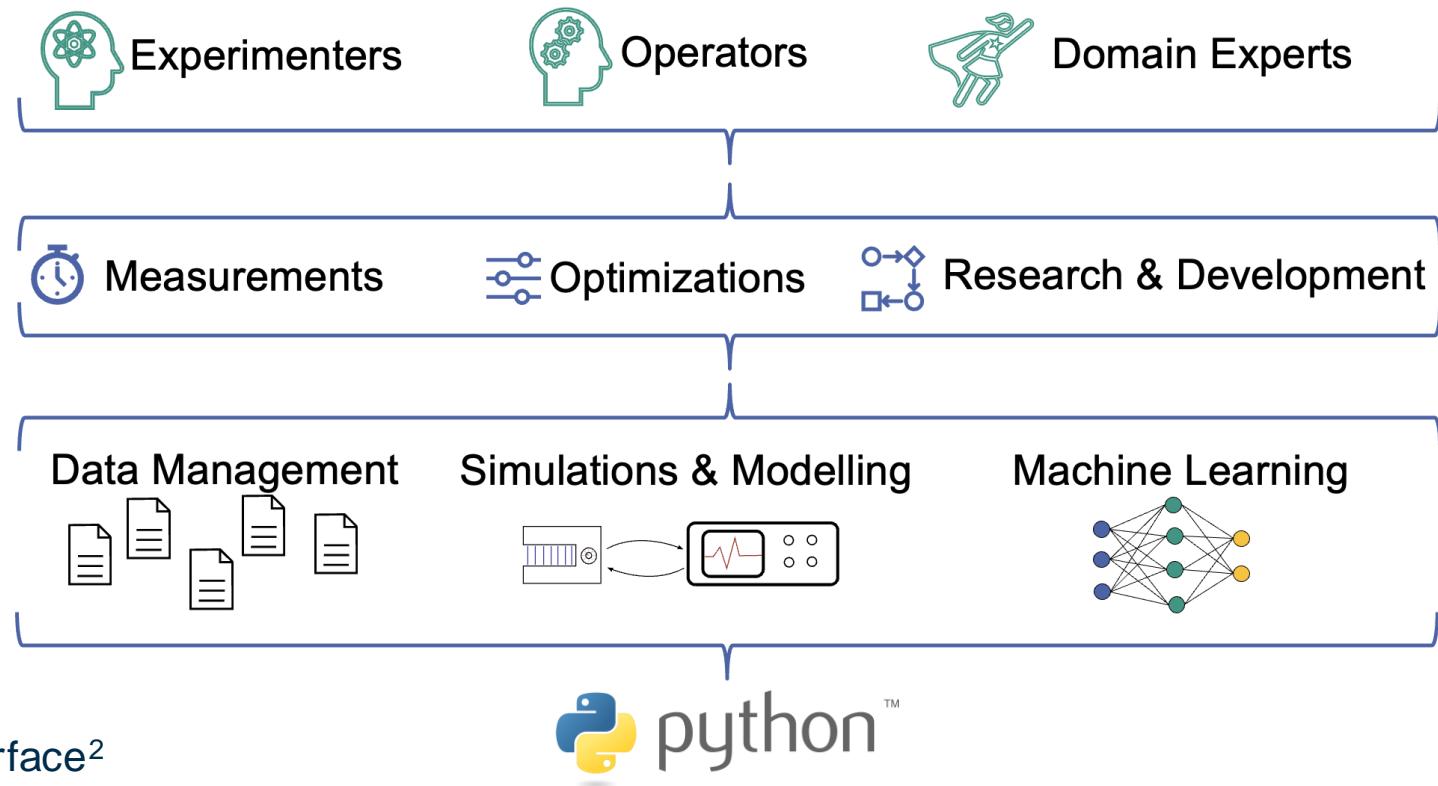
Elmo MC Gold Drive

Python Tools

- **pythonSoftIOC** for EPICS interface
- **Machine learning** via (Apptainer) **Container**
 - Live optimising of KARA injector¹
- **Debianizing** of:
 - Caproto, pyepics, softioc, setuptools_dso
epicsdbbuilder, epicscorelibs
 - <https://gitlab.kit.edu/kit/ibpt/python-tools/extern>
- Site-specific Python modules for easier to use interface²

Implementation of:

- Measurement routines?
- Hardware abstraction?
- Simulations & models?



```
from ibpt import accelerator
from ibpt.accelerator import kara
from ibpt.epics import get_pv
from ibpt.pvs import get_pv_string
from ibpt.utils.network import is_internal_network

if is_internal_network():
    energy = epics.get_pv("beam_energy")
    print(f"accelerator.get} runs with {energy} GeV")
    # -> KARA runs with 2.5 GeV
    energy_pv = get_pv_string("beam_energy")
    print(f"You can cross check it with 'caget {energy_pv}'")
    # -> You can cross check it with 'caget A:SR:BeamInfo:01:Energy'
```

```
from ibpt import accelerator
from ibpt.accelerator import flute
from ibpt.epics import get_pv
from ibpt.pvs import get_pv_string
from ibpt.utils.network import is_internal_network

if is_internal_network():
    energy = epics.get_pv("beam_energy")
    print(f"accelerator.get} runs with {energy} GeV")
    # -> FLUTE runs with 0.05 GeV
    energy_pv = get_pv_string("beam_energy")
    print(f"You can cross check it with 'caget {energy_pv}'")
    # -> You can cross check it with 'caget F:LIN-1:BeamInfo:01:Energy'
```

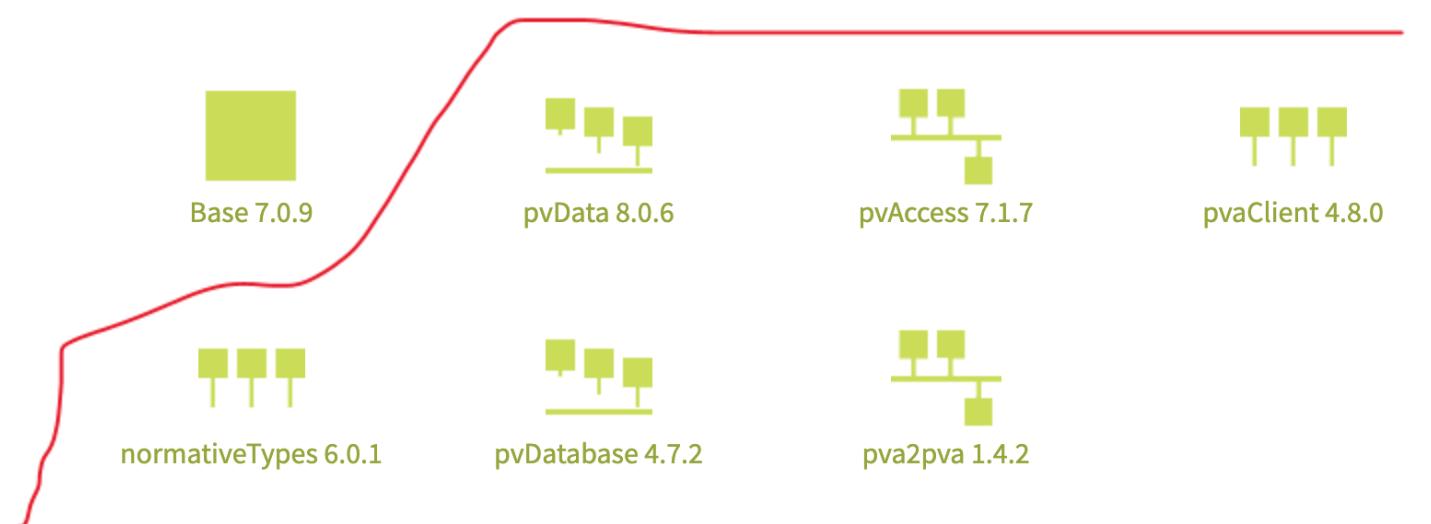
→ pyAML Collaboration

¹<https://doi.org/10.1103/PhysRevAccelBeams.26.034601>

²<https://doi.org/10.18429/JACoW-PCaPAC2022-THPP9>

pvAccess?

- EPICS 7 in use since 2018 (from **3.14.12.5** to **7.0.1.1**)
- But so far Channel Access only(!)
- Plans to start with pvAccess “soon”™ for:
 - Camera data (using areaDetector’s pvA module)
 - i-Tech Spark platform (BPMs & BLMs)
- Open tasks:
 - Just learn all the “new” stuff (concepts, normative types, pvxs, ...)
 - Set up pva gateways
 - How to handle archiving? (probably still CA)
 - Python integration



Outlook

edmund.blomley@kit.edu

- Finish migration to Ubuntu 24.04 and latest EPICS version
 - Shutdown at KARA next week...
 - Including several general IOC improvements
- Polish areaDetector integration and present/convince "users"
- Continue migration of OPC-UA
- Migrate CSS panels
 - Student helping with that is starting next week
 - Actually switch to Phoebus at some point
- Re-structure accelerator networks
 - Including re-assignment of ALL IP addresses
- (couple of other projects I did not have the time to mention...)
- All the "normal" controls stuff and support: "Please add THIS NEW device, maybe until tomorrow?"
- LPA?

Invitation

- Make use of our test facility
- EPICS driver? IOCs? Plugins?
- Software? Feedback loops?
- Instrumentation & Diagnostics?
- Measurement workflows?
- Even vacuum components...

<https://gitlab.kit.edu/kit/ibpt/>

Not getting boring...

cSTART coming online (first components) towards end of 2026