

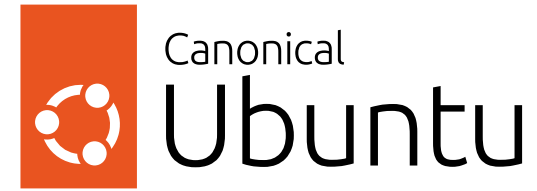
Packaging & Deployment at KIT

24.04.2026

E. Blomley – Spring EPICS collaboration meeting 2026

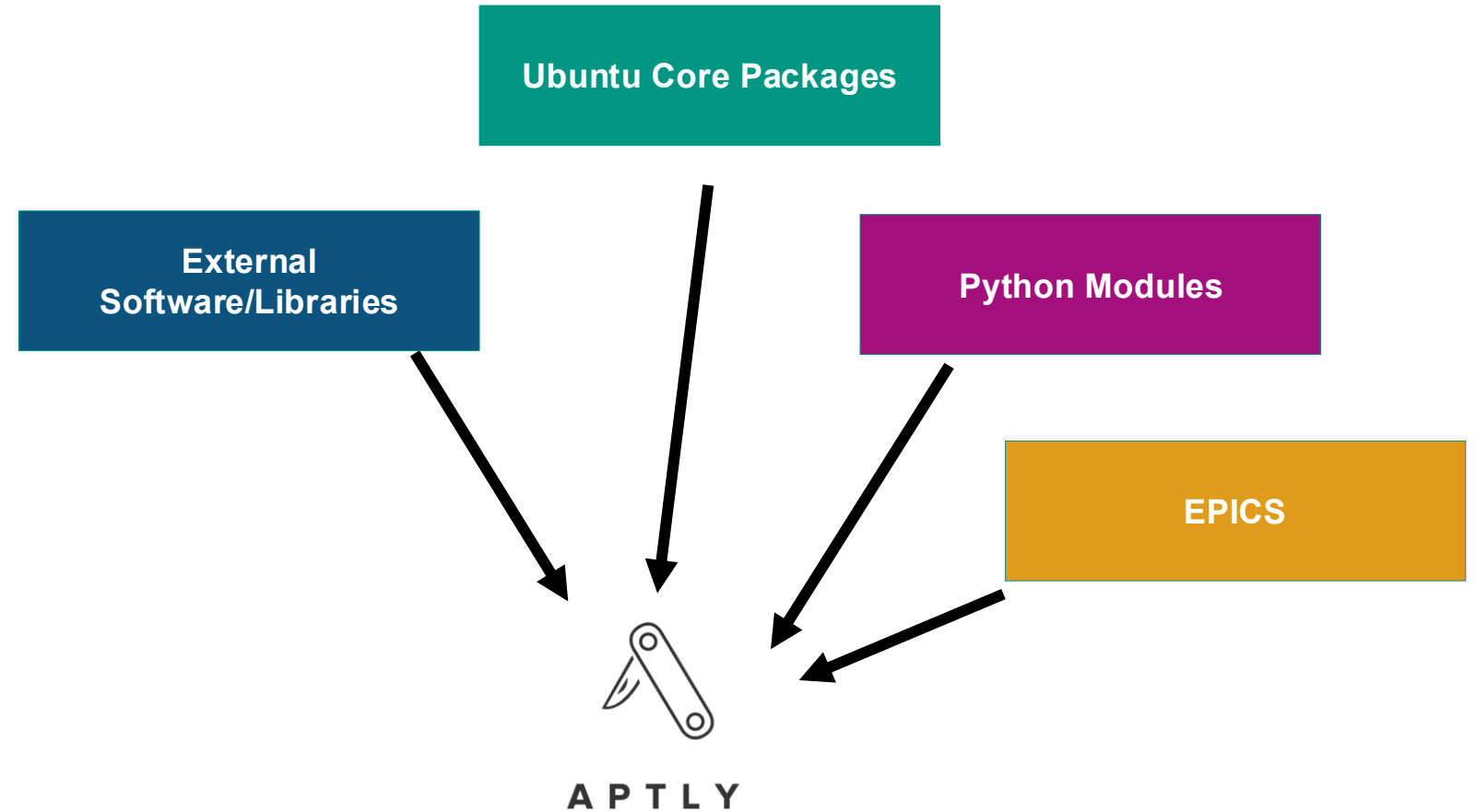
Software Stack

- EPICS: 2012
- Ubuntu: ~ 2014
- Salt: ~ 2016
- GitLab CI: ~ 2020



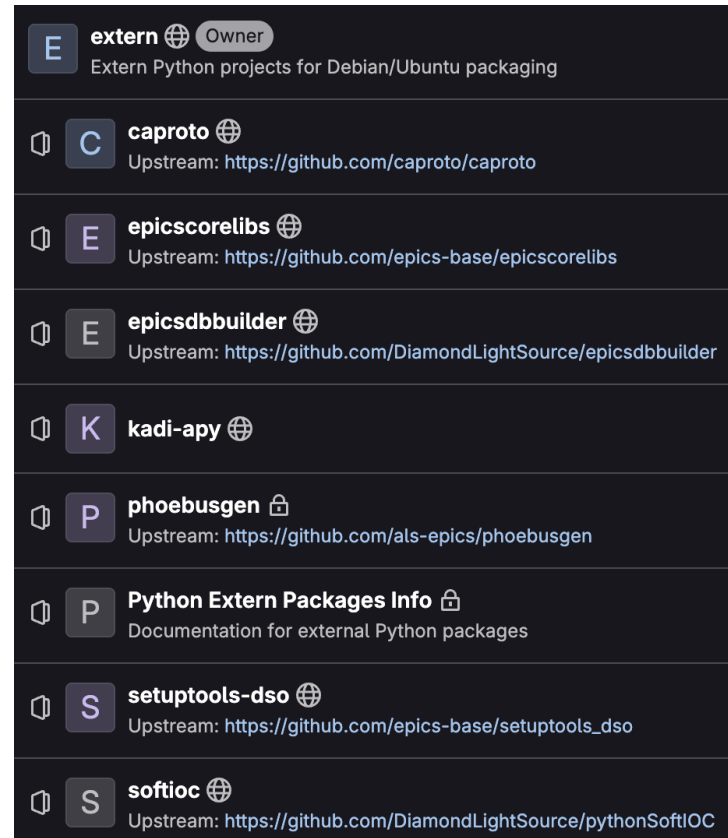
Packaging – Ubuntu LTS & Debian Packages

- Basic rule (since 2016?):
.deb .deb .deb .deb
- Use it as “whitelist” filter



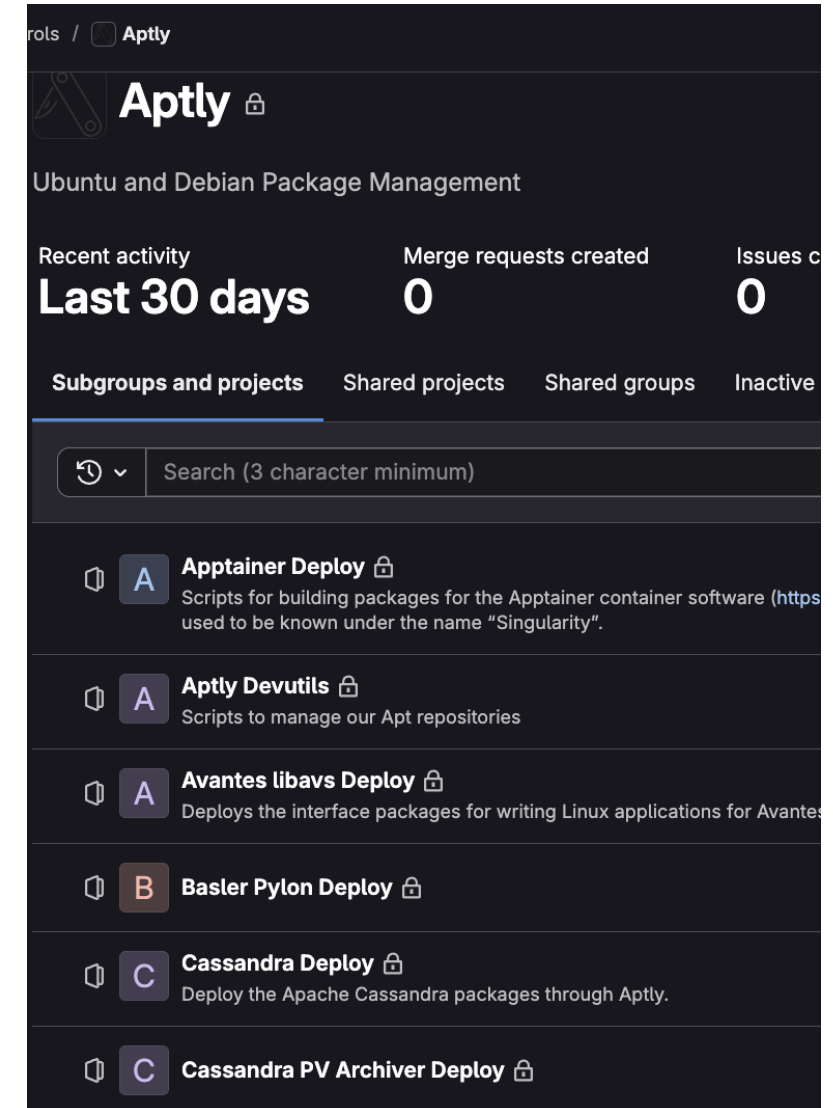
External Libraries & Python

- Allow all Ubuntu Python packages
- Bundle fixed set of core EPICS related libraries
- Mirror or host external .deb
- Build .deb via CI



The screenshot shows the 'extern' group page in Aptly. The group is owned by 'extern' and is used for Debian/Ubuntu packaging. It contains the following projects:

- caproto** (C): Upstream: <https://github.com/caproto/caproto>
- epicscorelibs** (E): Upstream: <https://github.com/epics-base/epicscorelibs>
- epicsdbbuilder** (E): Upstream: <https://github.com/DiamondLightSource/epicsdbbuilder>
- kadi-apy** (K)
- phoebusgen** (P): Upstream: <https://github.com/als-epics/phoebusgen>
- Python Extern Packages Info** (P): Documentation for external Python packages
- setuptools-dso** (S): Upstream: https://github.com/epics-base/setuptools_dso
- softioc** (S): Upstream: <https://github.com/DiamondLightSource/pythonSoftIOC>



The screenshot shows the Aptly web interface for the 'Aptly' group. The group is used for Ubuntu and Debian Package Management. The recent activity for the last 30 days shows 0 merge requests created and 0 issues closed. The page lists several subgroups and projects:

- Apptainer Deploy** (A): Scripts for building packages for the Apptainer container software (<https://github.com/apptainer/apptainer>) used to be known under the name "Singularity".
- Aptly Devutils** (A): Scripts to manage our Apt repositories
- Avantes libavs Deploy** (A): Deploys the interface packages for writing Linux applications for Avantes
- Basler Pylon Deploy** (B)
- Cassandra Deploy** (C): Deploy the Apache Cassandra packages through Aptly.
- Cassandra PV Archiver Deploy** (C)

EPICS Packaging

- Package EPICS to Debian
- BUT: do not follow Debian rules
 - /opt/epics/base-<version>
 - /opt/epics/modules/<xyz>
 - /opt/epics/extensions/<abc>

```
EPICS_LINSTAT_VERSION="1.1.0"
EPICS_LINSTAT_ARCHIVE_URL="https://github.com/mdavidsaver/linStat/archive/refs/tags/$EPICS_LINSTAT_VERSION"

EPICS_MODBUS_VERSION="3.4"
EPICS_MODBUS_ARCHIVE_URL="https://github.com/epics-modules/modbus/archive/R3-4.tar.gz"
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.tar.gz"
EPICS_MOTOR_DEPEND_MODULES=( asyn busy ipac seq )
EPICS_MOTOR_SUB_MODULES=(
  motorMotorSim
  motorNewport
  motorOms
  motorOmsAsyn
  motorPIGCS2
  motorPhytron
  motorSmarAct
  motorThorLabs
)
```

Files

Search files (*.vue, *.rb...)

> docs

> patches

✓ shlib

apply_patches.sh

build_lib_open62541.sh

build_module.sh

build_software.sh

canonicalize_path.sh

config.sh

download_file.sh

download_software.sh

download_software_component.sh

extract_archive.sh

extract_major_minor_version.sh

extract_software.sh

find_dependencies_or_executables.sh

get_archive_ext_files_var_name.sh

get_archive_strip_components.sh

get_component_dependency_modules.sh

```
34
35 # List of all EPICS modules that shall be built. There must be a
36 # version and URL for each of the modules listed here.
37 EPICS_MODULES=(
38   areaDetector
39   asyn
40   autosave
41   bacnet
42   busy
43   ca-gateway
44   calc
45   execute
46   iocLock
47   iocStats
48   ip
49   ipac
50   linStat
51   modbus
52   motor
53   motorSymetrie
54   mrf
55   opcua
56   open62541
57   pcas
58   pmac
59   pvxs
60   pyDevSup
61   recCaster
62   s7nodave
63   s7plc
64   seq
65   snmp
66   sscan
67   std
68   stream
69 )
```

<https://gitlab.kit.edu/kit/ibpt/controls/epics/distribution/epics-build-tool>

Build CI

Sources (mostly GitHub)

Build (& Bug) Patches

“Vanilla” EPICS

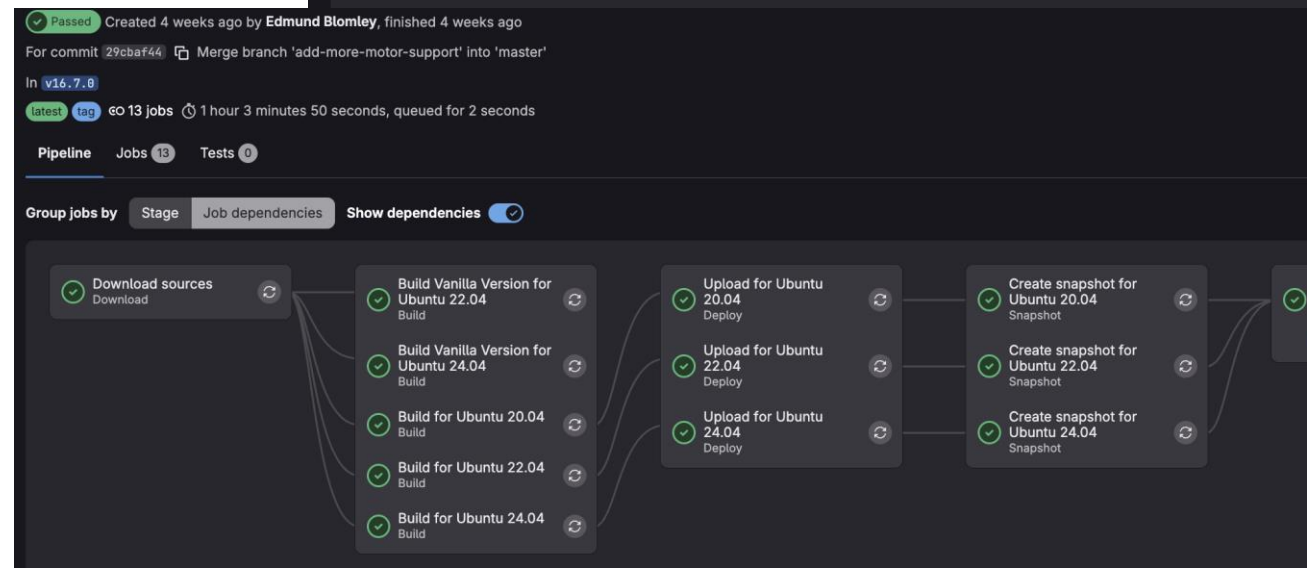
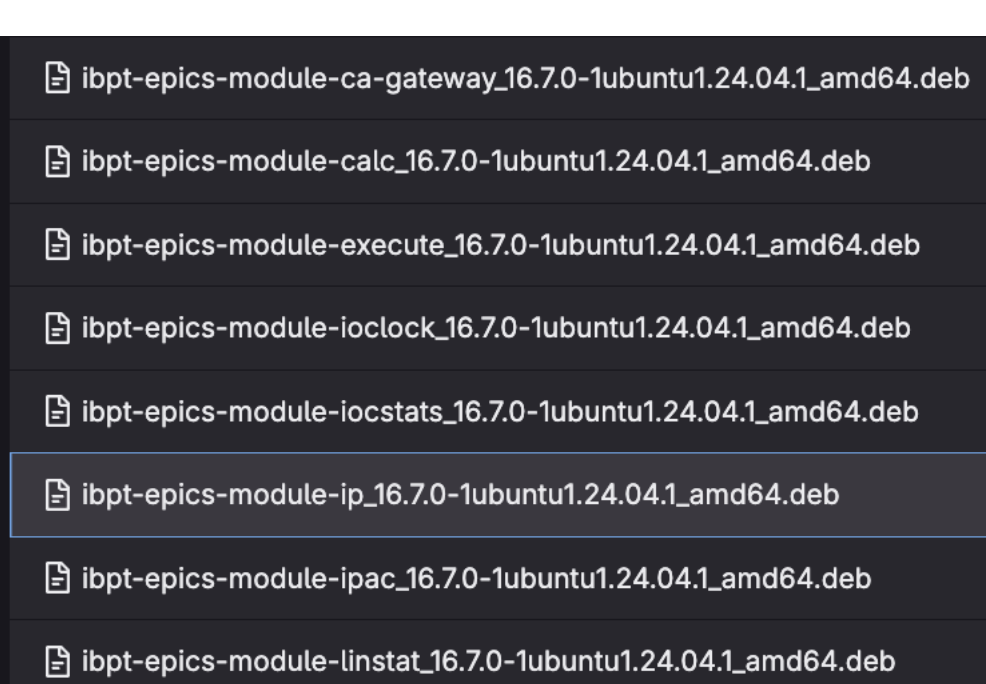
Internal Modules

IBPT (*) Patches

IBPT EPICS

Patch & Vanilla split since 2025

```
6      prompt("Record Name")
7      special(SPC, NOMOD)
8      -      size(61)
9      +      size(256)
10
11     field(DESC, DBF_STRING) {
12         prompt("Descriptor")
13     }
14     diff -Naur base-7.0.5.orig/modules/libcom/src/misc
15     +++ base-7.0.5/modules/libcom/src/misc
16     @@ -70,7 +70,7 @@
17     #endif
18
19     /** \brief Size of a record name incl
20     #define PVNAME_STRINGSZ 61
21     #define PVNAME_STRINGSZ 256
```

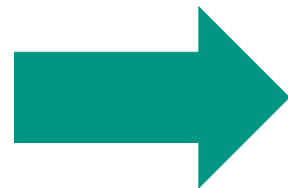
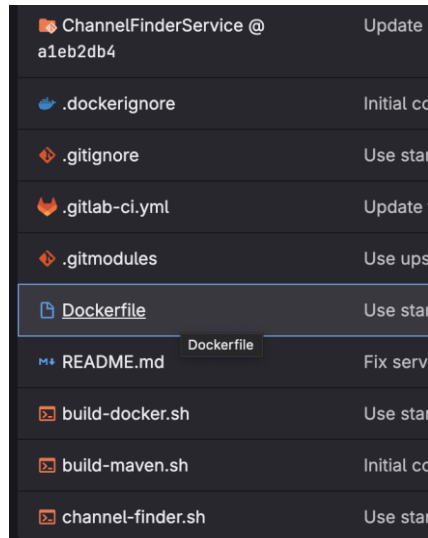


Publish to public webserver missing

*Institute for Beam Physics & Technology (IBPT) as part of Karlsruhe Institute of Technology (KIT)

From Zero to EPICS....

- IF you run Ubuntu LTS (no other requirements):
 - < 15 minutes from 0 EPICS to working OPC-UA PLC communication
 - Tested in a photovoltaics lab
- BUT: only solves the EPICS problem
- Services are NOT .deb packages



Deployed as Docker services via Salt

IOC Management

- GitLab Project Template
 - .gitignore
 - .gitattributes
 - .gitlab-ci.yml
 - README.md template
- makeBaseApp.pl
 - + run executable
- Manually set up modules
- CI only for README.md publication
 - After meeting:
 - Add db->markdown list
 - Add db->markdown visual
 - Add tree-sitter based lintings
 - Add build + run stage with shell output parsing

```
run 103 B
1 #!/bin/sh
2 cd "`dirname "$0"`"
3 ioc=`head -n 1 st.cmd | grep -E "^#\!" | cut -b 3-`"
4 exec "$ioc" st.cmd
5
```

📁 .gitlab	update templates
📁 OsciRsMeasurementApp	Added option to faste
📁 configure	initial commit
📁 iocBoot	Added possibility to r
🔗 .gitattributes	update default LFS ob
🔗 .gitignore	re-add autosave
🔗 .gitlab-ci.yml	Add default readme
📄 Makefile	initial commit
📄 README.md	Slightly updated read
📄 README.md	
R&S Oscilloscope Measurements	
Overview	

?!?

~200 – 300 IOC GitLab projects

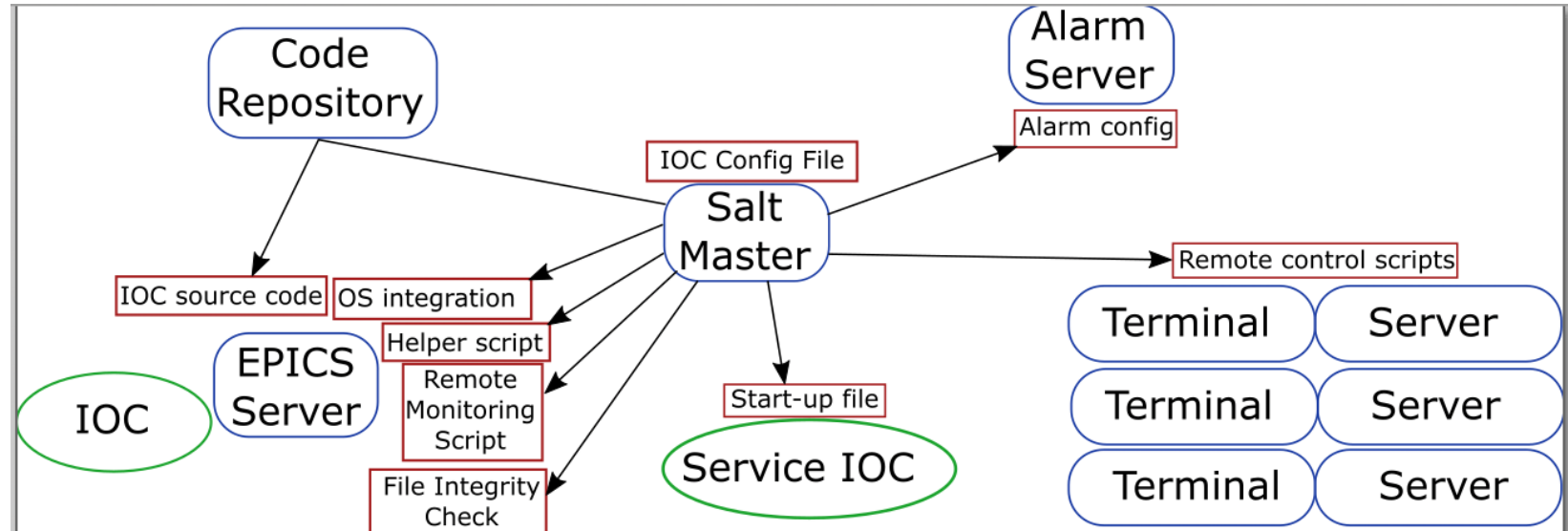
- Base & module version upgrades painful....
 - **Require module?**
 - **IOC structure which is just .db .st ?**

IOC Deployment

- Salt: Single Source of Truth
 - All configured IOC instances
 - Which Gitlab project to which host
 - Additional configuration options
 - Host has `ibpt-epics-base` & `ibpt-epics-modules-all` installed

```
[...]  
beam-info:  
  service-name: Beam Info  
  description: Current,energy,lifetime,dump status  
  dir-name: beam-info  
  host: acc-pc-epics01.anka-acc.kit.edu  
[...]  
camera-url:  
  autostart: False  
  service-name: URL Cameras  
  description: Controls URL cameras via AreaDetector  
  dir-name: camera-areadetector  
  host: acc-pc-cam01.anka-acc.kit.edu  
[...]
```

- Automated via Salt
 - Target host:
 - a. Clone IOC
 - b. Build
 - c. Systemd config
 - d. Local launch script
 - e. Monitoring scripts
 - Global:
 - a. Alarm server
 - b. Remote control scripts

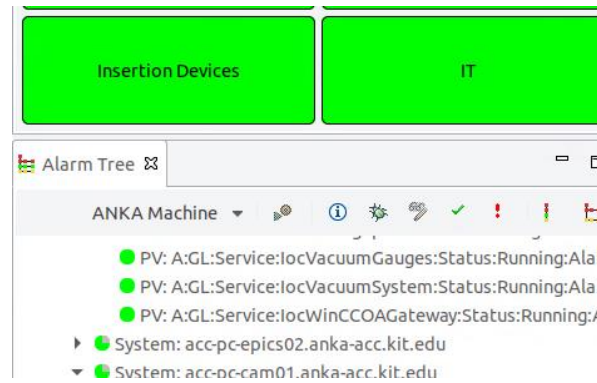


IOC Management

- IOCs run in screen sessions wrapped in systemd processes

```
ankaop@acc-pc-oc04:~$ ioc-manage
ioc-acs-ps-adapter
ioc-agilent-func-gen
ioc-agilent-ion-pump
ioc-agilent-ion-pump2
ioc-amplifier-barthel
ioc-amplifier-rs
ioc-auto-pilot
ioc-aux-diag
ioc-basler-camera
ioc-bba
ioc-bbb-amplifier
ioc-bbb-global
ioc-beam-info
```

Terminal – independent of host



Salt injects xml config for alarm server

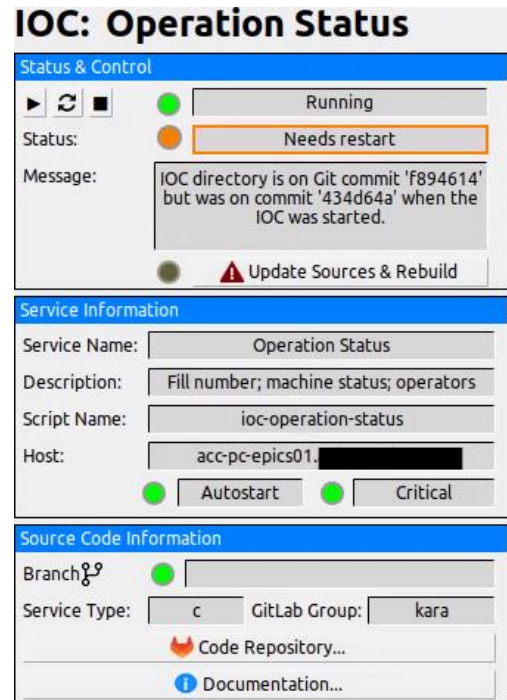
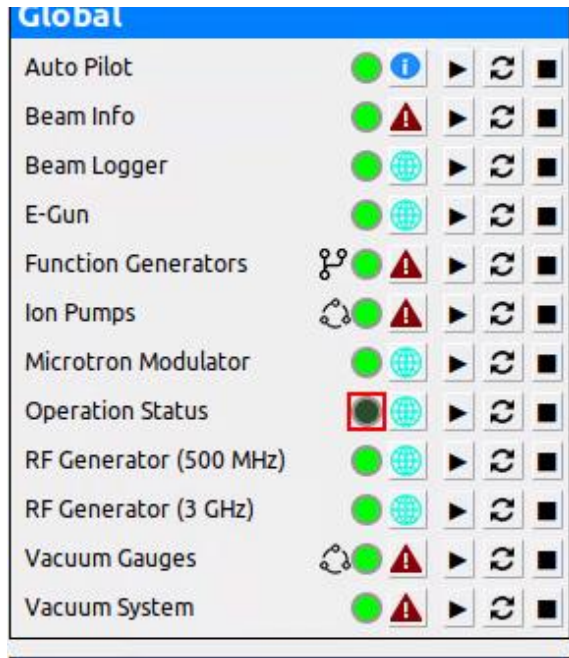
State	Service
OK	IOC: URL Cameras
Local site lbpt, acc-pc-epics01.anka-acc.kit.edu	
OK	IOC: ACS to EPICS PS Adapter
OK	IOC: Auto Pilot
OK	IOC: BBB Amplifier Barthel
OK	IOC: BBB Amplifier Milmega
OK	IOC: BBB Amplifier R+S
OK	IOC: BBB Global
OK	IOC: Beam Based Alignment
OK	IOC: Beam Info

Salt creates local script for IT monitoring integration (CheckMK)

- System logging: Elastic -> Kibana stack
- KARA runs 150-160 IOC instances

IOC to Manage IOCs

- IOC status and basic IOC control via EPICS with panel integration
- Inject st.cmd file via Salt to Service IOC



Linstat for every IOC service

Execute device support for EPICS

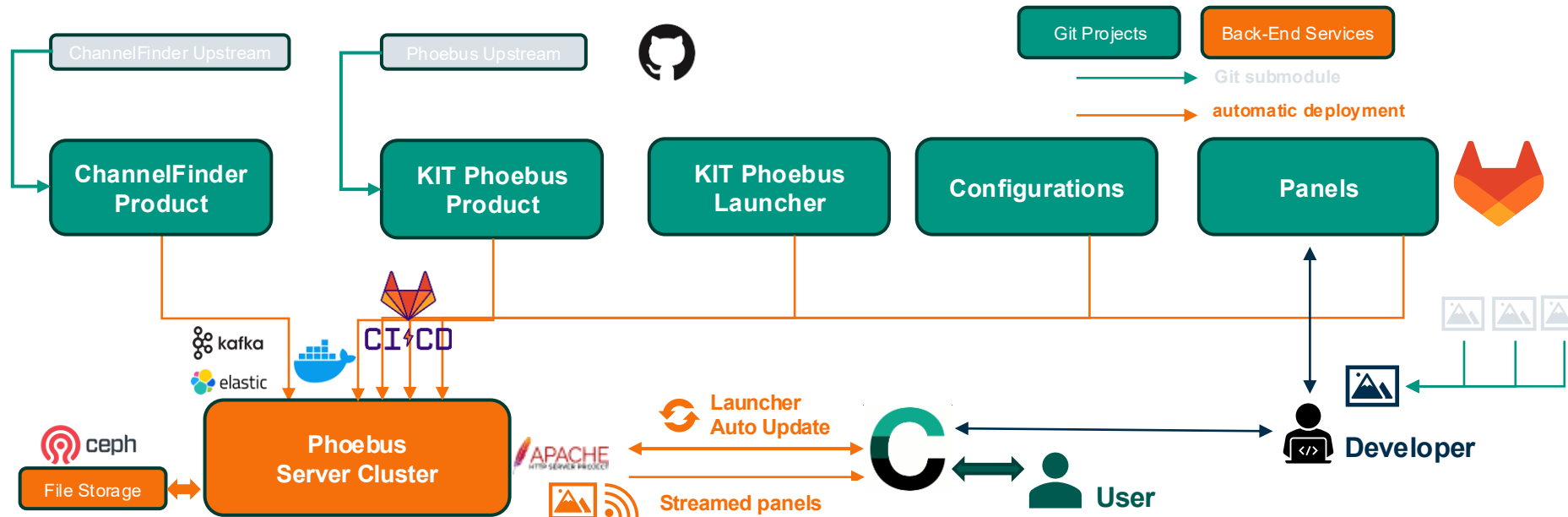
The execute device support is a general-purpose software device support that can be used to run any program from an EPICS IOC. Typically, this will be scripts (e.g. Bash, Python), but it is perfectly possible to run binary programs as well.

The possibility to use arbitrary scripts facilitates a wide range of possible applications, like sending notifications, writing log files, calling web services, querying databases, and many more.

The execute device support always runs program asynchronously, so a program will never block operation of the IOC. Data can be passed from the IOC using arguments (command-line parameters), environment variables, and a stream of characters (supplied to the program on the standard input). Both the program's exit code and its output can be passed back to the EPICS IOC.

<https://github.com/KIT-IBPT/epics-execute>

Honorable Mention: Phoebus



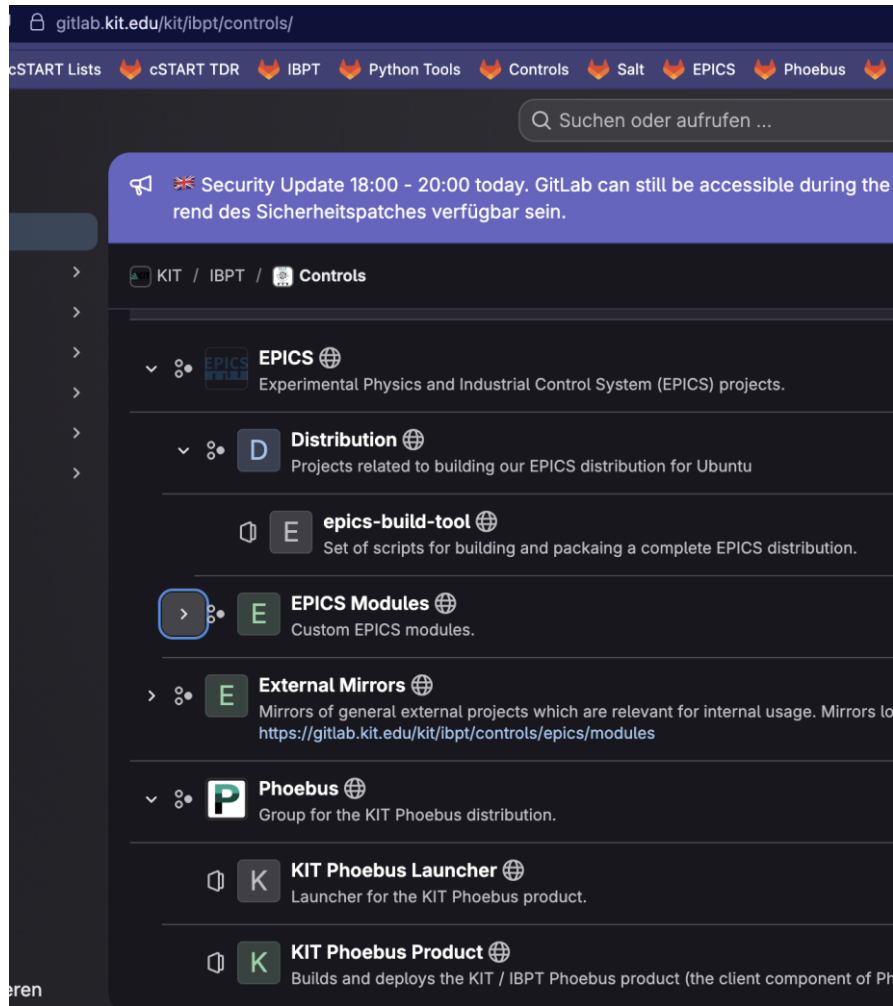
Honorable Mention: Embedded Systems, Backups and monitoring

- Central IT provides VEEAM for backups and CHECK_MK for IT monitoring
- We only backup data drives
- No EPICS server, terminals, salt master, ...
 - Use distributed CephFS file cluster for autosave

- Embedded systems hardly managed/automated
 - ~80-90 embedded systems vs 150-200 IOCs
 - Basic idea: VM to manage embedded systems via SSH (backup and configuration)

Resources

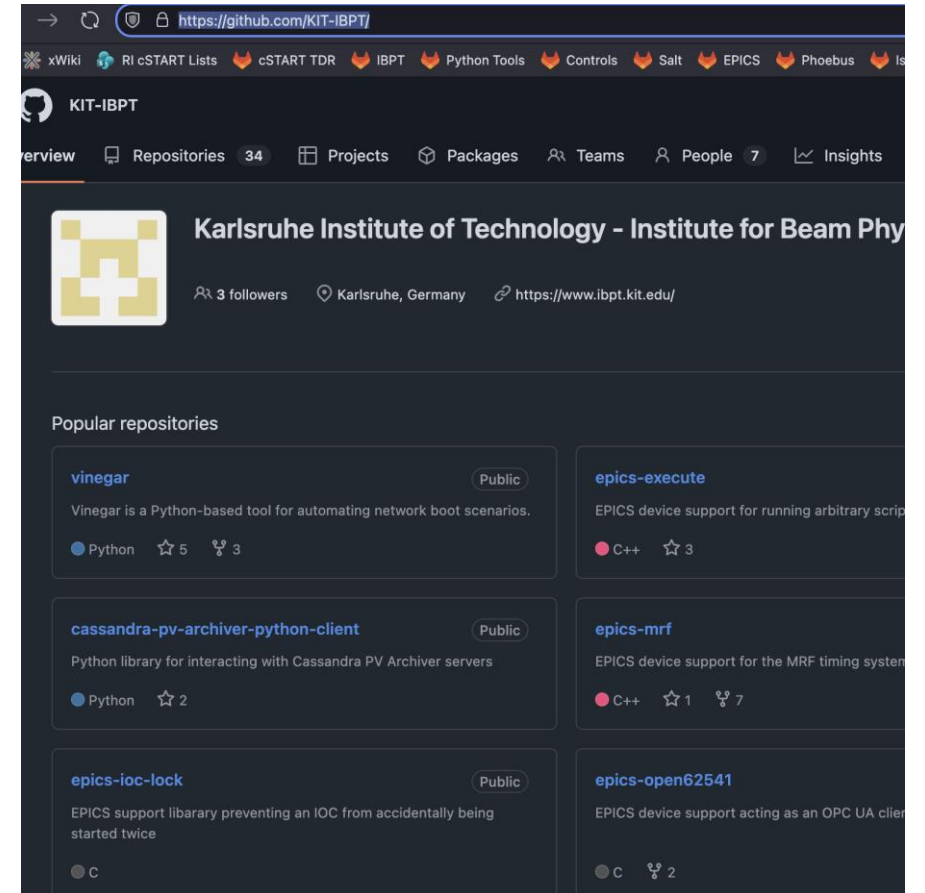
<https://gitlab.kit.edu/kit/ibpt/controls/>



The screenshot shows the GitLab interface for the repository `kit/ibpt/controls`. At the top, there is a navigation bar with icons for various projects: cSTART Lists, cSTART TDR, IBPT, Python Tools, Controls, Salt, EPICS, and Phoebus. A search bar is present with the text "Suchen oder aufrufen ...". A blue notification banner at the top states: "Security Update 18:00 - 20:00 today. GitLab can still be accessible during the rend des Sicherheitspatches verfügbar sein." Below the notification, the breadcrumb path is "KIT / IBPT / Controls". The main content area lists several categories of projects:

- EPICS** (Experimental Physics and Industrial Control System (EPICS) projects.)
- Distribution** (Projects related to building our EPICS distribution for Ubuntu)
- epics-build-tool** (Set of scripts for building and packaing a complete EPICS distribution.)
- EPICS Modules** (Custom EPICS modules.) - This category is highlighted with a blue circle.
- External Mirrors** (Mirrors of general external projects which are relevant for internal usage. Mirrors lo <https://gitlab.kit.edu/kit/ibpt/controls/epics/modules>)
- Phoebus** (Group for the KIT Phoebus distribution.)
- KIT Phoebus Launcher** (Launcher for the KIT Phoebus product.)
- KIT Phoebus Product** (Builds and deploys the KIT / IBPT Phoebus product (the client component of PH...

<https://github.com/KIT-IBPT/>



The screenshot shows the GitHub profile page for `KIT-IBPT`. The profile information includes the name "Karlsruhe Institute of Technology - Institute for Beam Physics", 3 followers, and the location "Karlsruhe, Germany". The website link is <https://www.ibpt.kit.edu/>. The page displays a list of popular repositories:

- vinegar** (Public): Vinegar is a Python-based tool for automating network boot scenarios. Python, 5 stars, 3 forks.
- epics-execute** (Public): EPICS device support for running arbitrary scrip. C++, 3 stars.
- cassandra-pv-archiver-python-client** (Public): Python library for interacting with Cassandra PV Archiver servers. Python, 2 stars.
- epics-mrf** (Public): EPICS device support for the MRF timing system. C++, 1 star, 7 forks.
- epics-ioc-lock** (Public): EPICS support library preventing an IOC from accidentally being started twice. C, 2 forks.
- epics-open62541** (Public): EPICS device support acting as an OPC UA client. C, 2 forks.