MAX PLANCK GESELLSCHAFT

FRITZ-HABER-INSTITUT
MAX-PLANCK-GESELLSCHAFT

How can we get EPICS out there?

5 minutes of philosophy ...

EPICS

AGENDA

01 The Problem

02 Windows and GUI config

03 Provide a ready to use system

04 What's going on around it?

05 Our next (last?) attempt

01

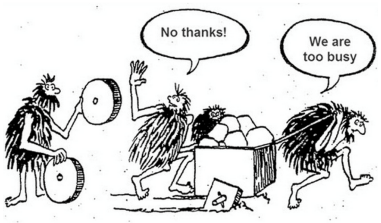
The Problem

Why use of EPICS is harder than it looks?
(at collaboration meetings like this one)

Why use of EPICS is harder than it looks?

User experiences

- Only familiar with Windows - OS
- Are only familiar with graphical user interfaces (GUI or browser)
- They have no experience with the command line, ‘Terminal’ ?
- They have never heard of ‘make’ ….
- Nor are they willing to actually learn it (“we don’t have time”)



Documentation

- A lot of outdated information (EPICS has been around for a long time, and a lot has been published about it; the internet never forgets)
- Documentation written by professionals
- Even the slightest inaccuracies lead to frustration for those who try their hand at it

E.g. at ahttps://docs.epics-controls.org/en/latest/getting-started/installation-linux.html

```
cd $HOME/EPICS/TEST/testIoc
makeBaseApp.pl -t example testIoc
makeBaseApp.pl -i -t example testIoc
make
cd iocBoot/ioctestIoc
chmod u+x st.cmd
ioctestIoc> ./st.cmd
#!../../bin/darwin-x86/testIoc
< envPaths
```

02

Windows and GUI config

So that users can work in their familiar environment

So that users can work in their familiar environment

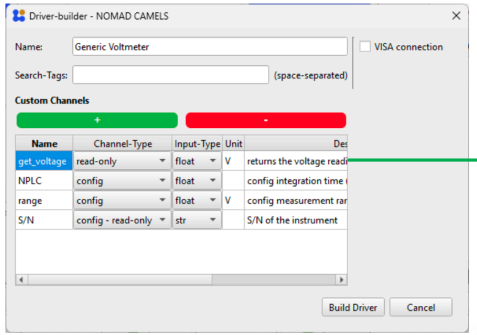
CAMELS – NOMAD (as part of the NFDI FAIRmat Consortium)

CAMELS has mainly developed into a LabView replacement. That's a good thing in itself, as it means we can move away from commercial software and licence costs.

https://fau-lap.github.io/NOMAD-CAMELS/

CAMELS provides a graphical user interface (GUI) that can be used to setup instrument control and measurement protocols. The GUI then generates Python code that interfaces with bluesky to communicate with the instruments and orchestrate the measurement. CAMELS can also be used to communicate with large-scale, distributed systems implemented with EPICS.

Instrument drivers: channels → physical quantities



Add URIs to definition of physical quantities (ontology/taxonomy of physical quantities)

Unfortunately, it doesn't put the EPICS concept into practice :-(

A single computer system to which the devices are connected via USB. Monitoring and control are limited to this device, poor real-time performance, etc. It follows the familiar approach used in, for example, LabVIEW.

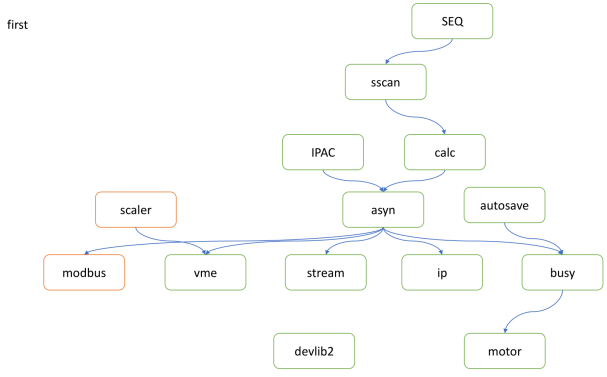
03

Provide a ready to use system

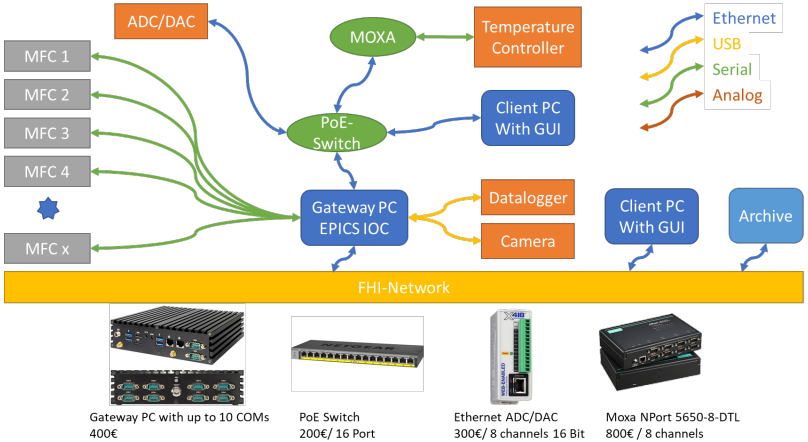
EPICS already installed with a sample (standard) IOC

Provide a ready to use system

- Create user 'epics'?
- Installing neccessary software (Packages)?
- Install basic dnsmasq config for DHCP and DNS?
- Install procServ to /opt ?
- Install EPICS base?
- Install FHI support modules compilation?
- Create FHI folder for user epics? (RELEASE.local)



https://gitlab.fhi.mpg.de/epics-tools/epics-edge-setup



04

What’s going on around it?

EPICS collaboration

What's going on around it?

There are countless examples within the EPICS community …

- conda install conda-forge::epics-base (https://e3.pages.ess.eu/)
- EPICS Workbench for VS Code and JetBrains (https://github.com/diverhao/epics-workbench, Hao Hao)
- vscode-epics (https://marketplace.visualstudio.com/items?itemName=nsd.vscode-epics, CEA Saclay)
- …

05

Our next (last?) attempt

Embedded Webserver running on the IOC

Embedded Webserver running on the IOC

Inspired by the CAENels Power Supply Series “Remote Control Manual”

- IOC Maintenance (update, health check, …)
- IOC configuration (load db files, change st.cmd, ….)