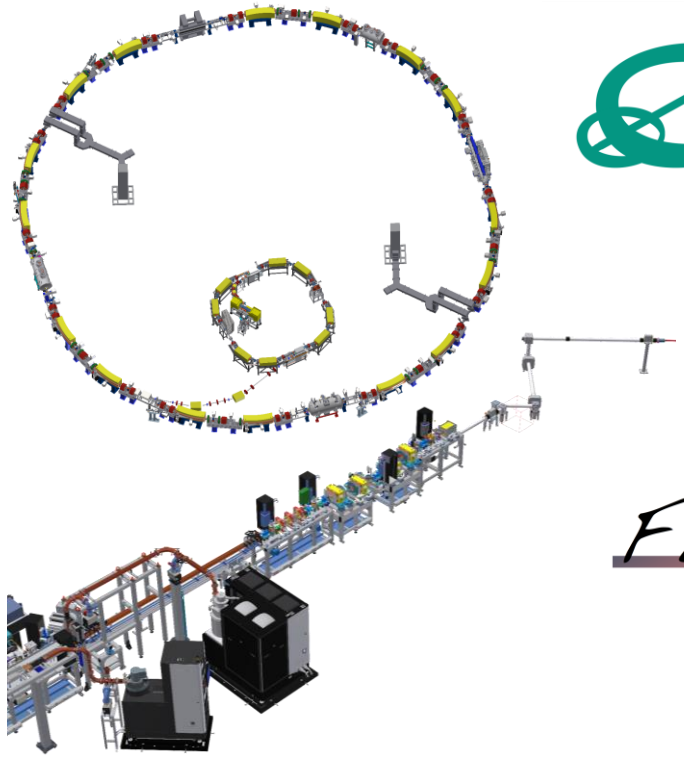
EPICS Module for Gating EPICS Alarms (and more)

21.04.2026

E. Blomley – Spring EPICS collaboration meeting 2026

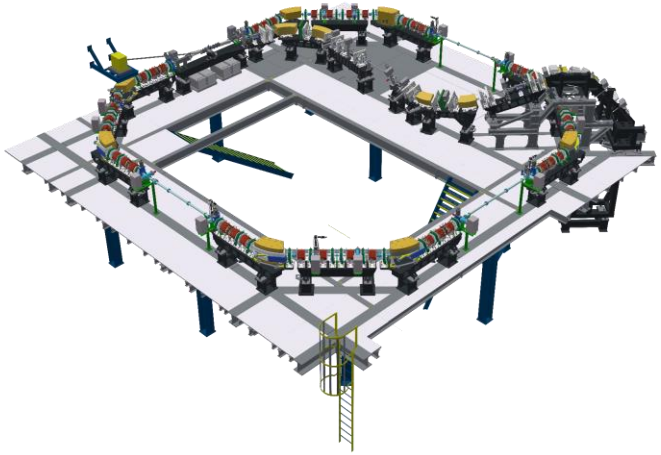
KIT Accelerators

- KARA (Karlsruhe Research Accelerator)
  - Mix of user operation and accelerator research
  - Injection: 1h, energy ramp 3 min, operation: 23h
- FLUTE (Ferninfrarot Linac- und Test-Experiment)
  - Experiments with very different operational setups
- cSTART
  - Storage ring for non-equilibrium physics
  - Injector: FLUTE **OR** laser plasma accelerator
  - Installation starts ~ Mid 2026, commissioning ~ 2027



KARA
Karlsruhe Research Accelerator

FLUTE



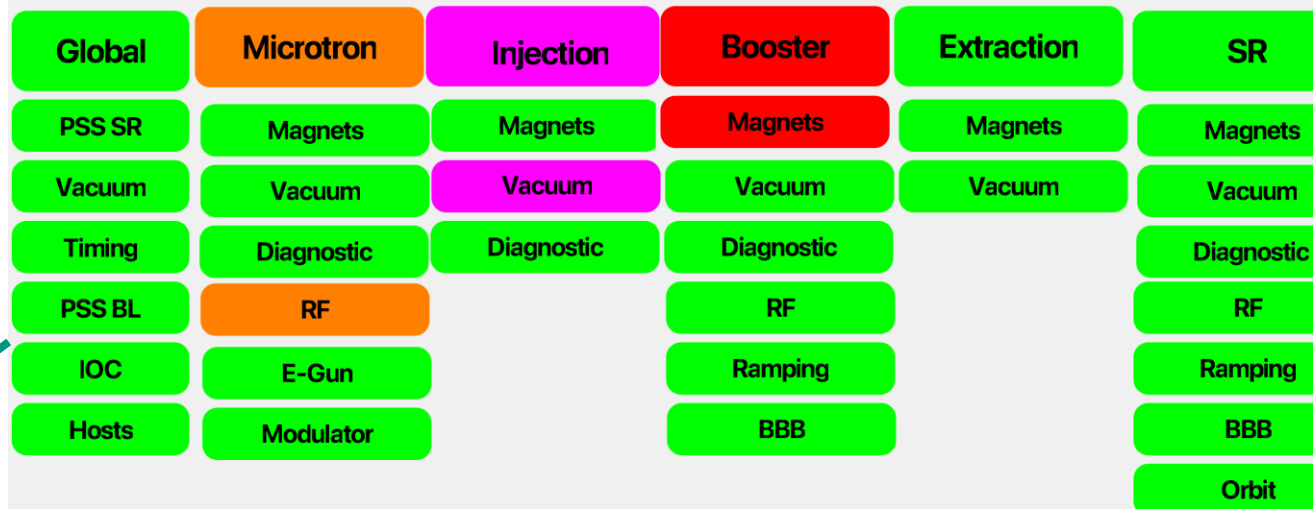
cSTART

Many different operational modes

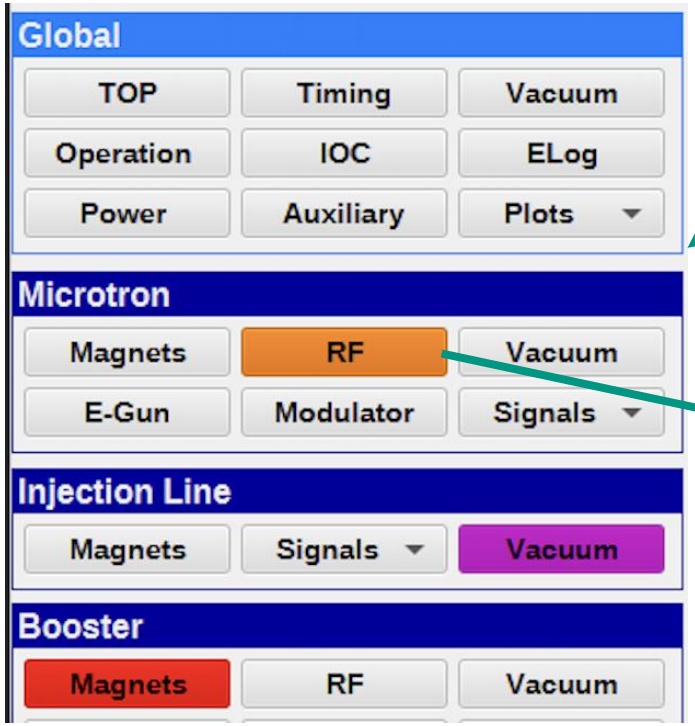
GitLab
PROXMOX
Canonical Ubuntu
EPICS
SALT PROJECT
python™

Motivation: Alarms Guide the Operator

- Alarm summaries mapped to navigation menu
- Support operator to quickly identify potential issues
- Lead to panel where the issue can be fixed



1. and 2. level alarm hierarchy reflects our navigation



Panel shows at least one minor alarm

Gating Alarms?

- “Alarms” in EPICS and this talk:
  - The alarm **severity** and alarm **threshold fields** of records

```
record(bi, Kicker:Status:On) {
    field(ZSV,  “MAJOR“)
    field(OSV, “NO_ALARM“)
}
```

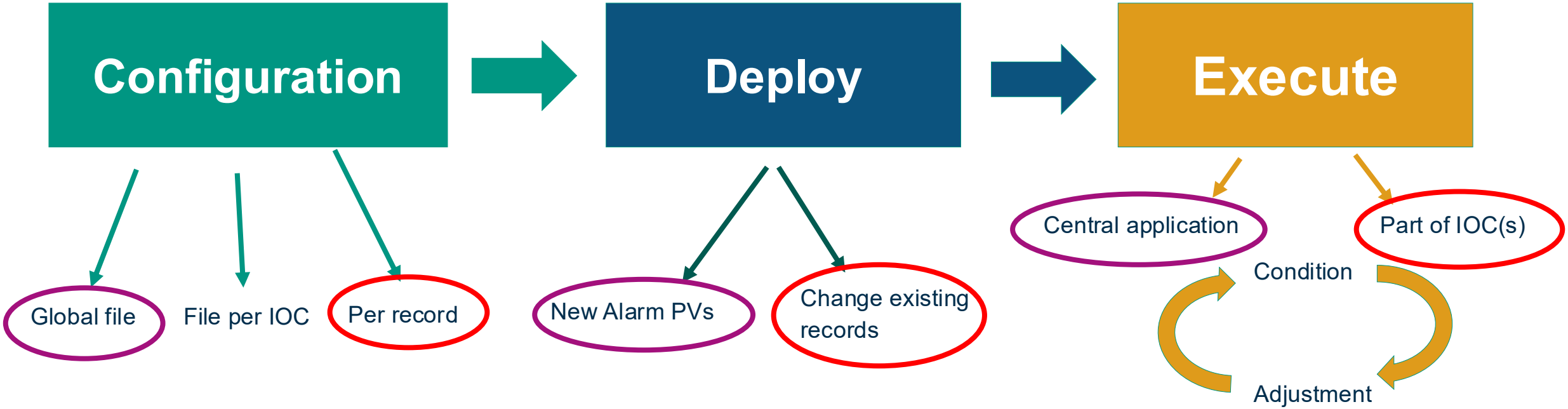
- Make “alarms” (dis)appear based on certain conditions
  - Kicker magnet **OFF** → **Alarm,** **ON** → **No alarm**
  - Kicker magnet **OFF** → **No alarm, ON** → **Alarm,**

```
record(ai, Magnet:Current) {
    field(LOLO,  “5“)
    field(LLSV, “MAJOR“)
    field(LOW,  “8“)
    field(LSV, “MINOR“)
    field(HIGH,  “14“)
    field(HSV, “NO_ALARM“)
}
```

- Not talking about actual (hard-wired) alarms & interlocks: machine protection, personal safety systems, …

~~Gating Alarms~~ → Dynamic EPICS notifications to support operators

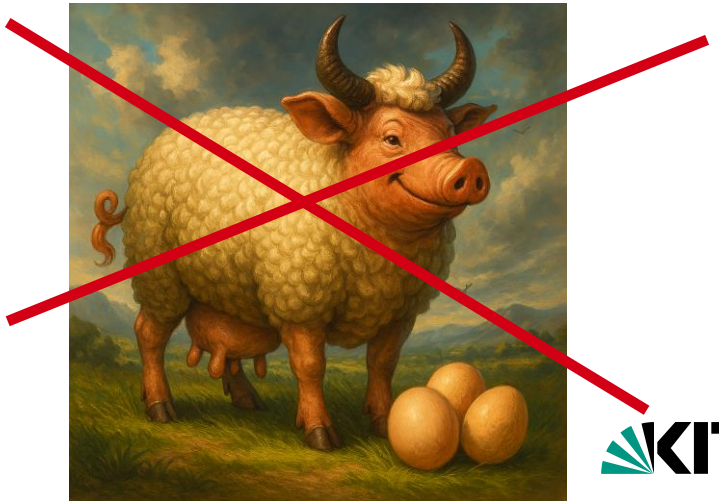
Possible System Designs



Best approach? Depends…

System we used for the last 10 years

Setup we are moving to



Introducing New EPICS Module: Conditional Field Adjuster

- Features
  - Use **info** fields for configuration
  - Only requires **EPICS base**
  - **Passive** while IOC is running

```
record(bi, Kicker:Status:On) {
    field(ZSV,  "MAJOR")
    field(OSV, "NO_ALARM")

    info(cfa:select, „STATUS:PV CP")

    info(cfa:1:ZSV, „NO_ALARM")
    info(cfa:1:OSV, „MAJOR")
}
```

STATUS:PV == 0: Injection

STATUS:PV == 1: Operation

In-place, compact & easy to understand and use.

Scales with IOC.

Configuration & Usage

Configuration

- configure/RELEASE
- myIocApp/src/Makefile
  myloc_DBD += conditionalFieldAdjuster.dbd
  myloc_LIBS += conditionalFieldAdjuster

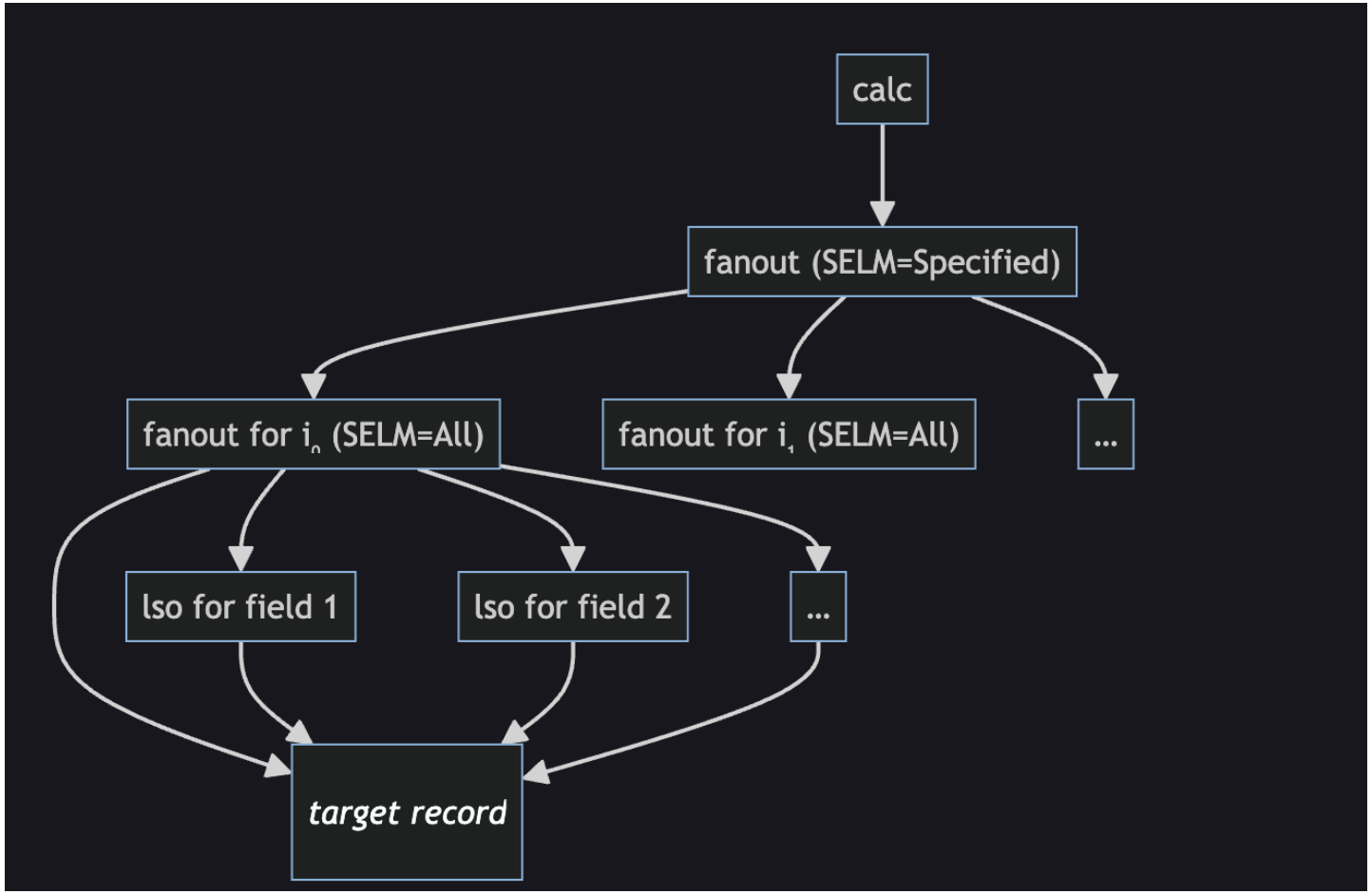
No st.cmd modifications

- `info(cfa:select, "<PV> CP")` - gate PV, expects positive integer
- `info(cfa:<int>:<field>, "<value>")`
  - <int>: integer to match the select PV
  - <field>: any field which exists for the record and is writable while IOC is running
- Default values are the actual record fields
- `info(cfa:scan, "Passive")` – optional

```
record(ai, "$(P)$(R)Value") {
    field(DESC, "Value simulator")
    field(VAL, "3.14159")
    field(PINI, "YES")
    field(PREC, "3")
    field(LOW, "3.0")
    field(LSV, "MAJOR")
    field(HIGH, "3.2")
    field(HSV, "MAJOR")

    # CFA fields
    info("cfa:select", "$(P)$(R)Mode CP")

    info("cfa:1:LOW", "3.15")
    info("cfa:1:LSV", "MINOR")

    info("cfa:2:HIGH", "3.18")
}
```

Backend

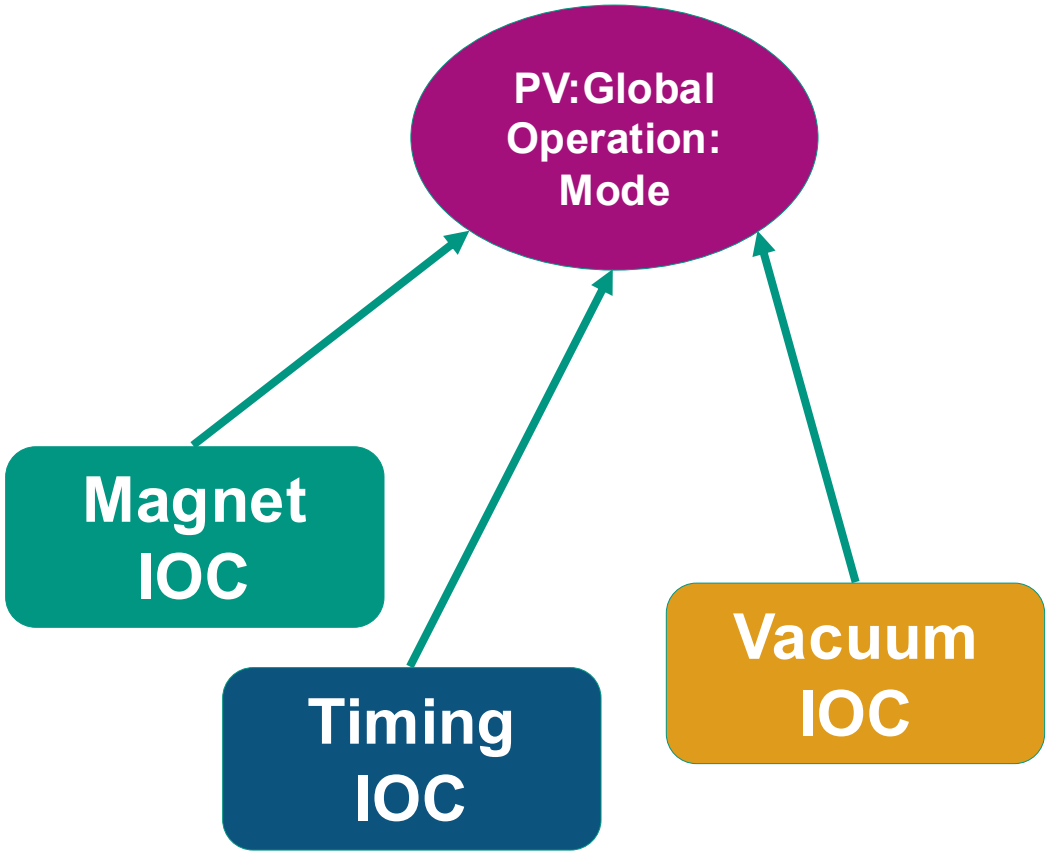
- Auto-generated records injected into IOC during startup
- **initHookAfterInitDevSup**
- Record naming:
  - **cfa:<30 random characters>**
  - Avoid potential PV length conflicts



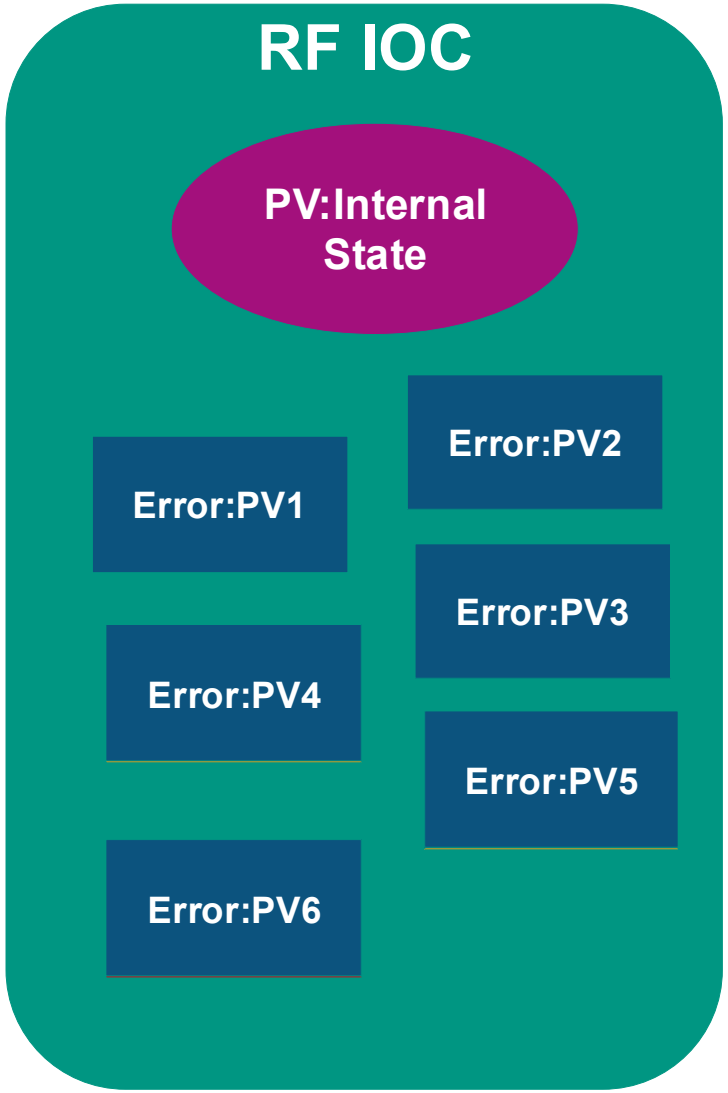

```
iocRun: All initialization complete
epics> dbl
Example:Value
Example:Status
Example:Mode
cfa:f8oCFa2otjp1QJV7gECnKsOWicGLDf
cfa:QGWLG2CfLgsdPbEgDDvnSv0k0IBbiB
cfa:xMl4rHqHH3GQLQdeWAuW3dGUzhD2c2
cfa:GhaAqRaxZD9vhnPrjq31Xk5oTwOOKK
cfa:rfAl7prQG7HFEkkLlcRKKiKH3AUzUp
cfa:ZVaJnIDZywMhqN5Sew0tsMKc1oxpvd
cfa:yoODPuFKU8rr1fTO556KAD8ruZOzBs
cfa:3ekKJ3szy60z2pnJmwjQBObm2wyX9H
cfa:jgTGhIVrmGKke0UMjuw2dAyGaSmClW
cfa:SPHr6lKaoGCIq1p65kq8jNgUJm8ayB
cfa:RPvaY7dKQbTDrvrEsvdNyCZTjuhzaf
cfa:wol7XAqmjtZzCiFHplXocZP5AZErQ6
cfa:MtoF8w8O63cDcsunRIKG5zUfKRME7W
cfa:Agk0BRP6vAbknW50y9ijgpNqX4Wh4H
cfa:AKFNo5PPRRY7RRN5xUPoMbqtyUZO9x
cfa:eF3pny483hWnjsjyhoAB2UowBVnMne
epics>
```

Considering to add configuration options for record naming

More on the Gate PV



Global Approach



Local Approach

Only for Alarms?

- Adjust SCAN rate?
- Precision?
- LOPR/HOPR?
- CALC parameters?
- …

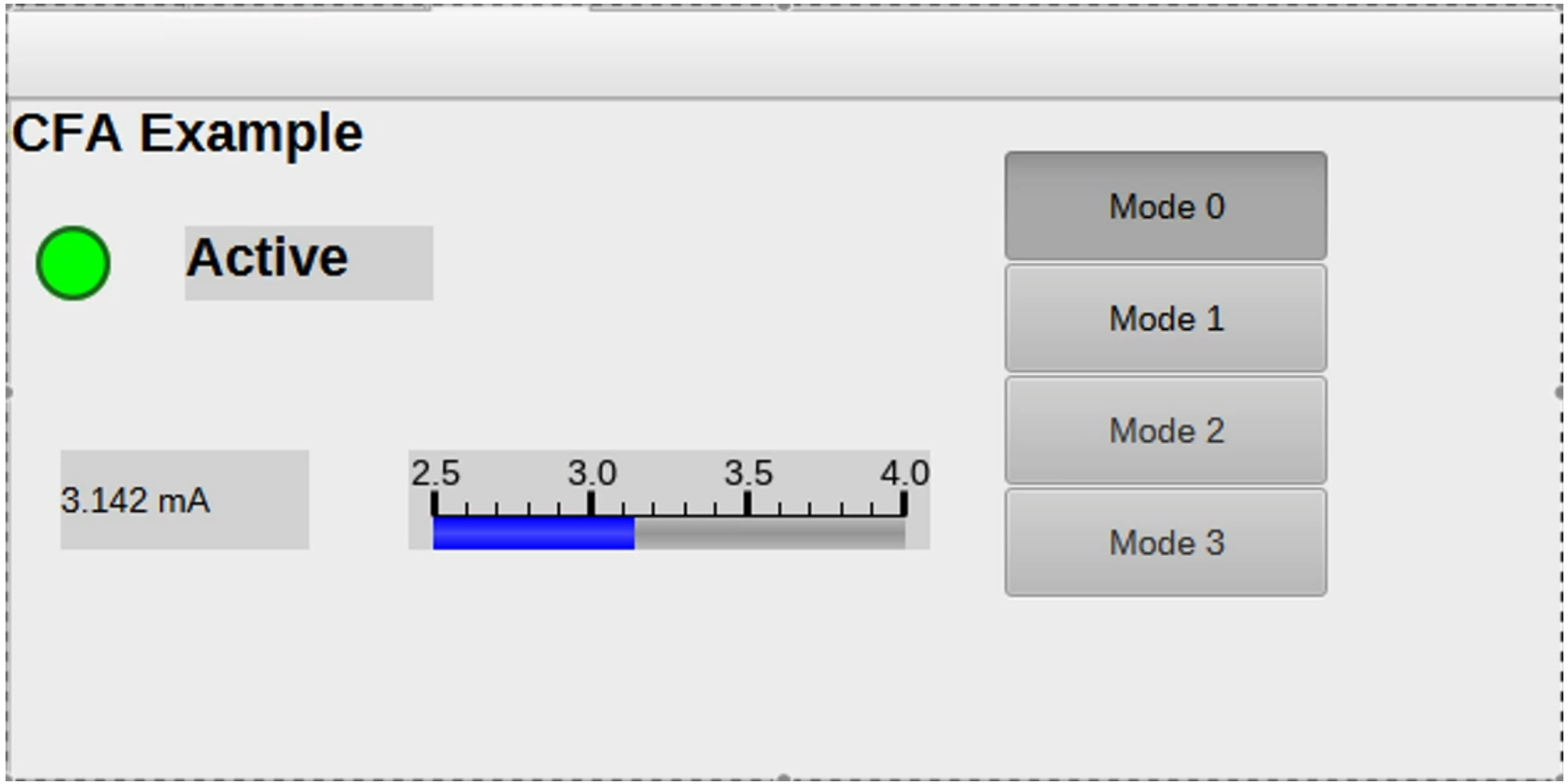
Can be used for any writable EPICS field

Limitations:

- Using as a central application
- Embedded systems
- Python-based IOCs
- Changes require IOC redeployment

```
record(ai, "$(P)$(R)Value") {
    field(DESC, "Value simulator")
    field(VAL, "3.14159")
    field(PINI, "YES")
    field(PREC, "3")
    field(LOW, "3.0")
    field(LSV, "MAJOR")
    field(HIGH, "3.2")
    field(HSV, "MAJOR")
    field(SCAN, "1 Second")
    field(EGU, "mA")

    # CFA fields
    info("cfa:select", "$(P)$(R)Mode CP")

    info("cfa:1:LOW", "3.15")
    info("cfa:1:LSV", "NO_ALARM")
    info("cfa:1:PREC", "1")
    info("cfa:1:EGU", "MW")
    info("cfa:1:SCAN", "Passive")

    info("cfa:2:HIGH", "3.18")
}
```

Demonstration




```
record(bi, "$(P)$(R)Status") {
    field(DESC, "Status simulator")
    field(VAL, "1")
    field(PINI, "YES")
    field(ZNAM, "Inactive")
    field(ZSV, "MAJOR")
    field(ONAM, "Active")
    field(OSV, "NO_ALARM")

    # CFA fields
    info("cfa:select", "$(P)$(R)Mode CP")
    info("cfa:1:ZSV", "NO_ALARM")
    info("cfa:1:OSV", "MAJOR")
}
```

```
record(ai, "$(P)$(R)Value") {
    field(DESC, "Value simulator")
    field(VAL, "3.14159")
    field(PINI, "YES")
    field(PREC, "3")
    field(LOW, "3.0")
    field(LSV, "MAJOR")
    field(HIGH, "3.2")
    field(HSV, "MAJOR")
    field(SCAN, "1 second")
    field(EGU, "mA")
    field(LOPR, "2.5")
    field(HOPR, "4.0")

    # CFA fields
    info("cfa:select", "$(P)$(R)Mode CP")

    info("cfa:2:LOW", "3.15")
    info("cfa:2:LSV", "MINOR")
    info("cfa:2:PREC", "5")
    info("cfa:2:EGU", "MW")
    info("cfa:2:SCAN", "Passive")

    info("cfa:3:HIGH", "3.18")
    info("cfa:3:LOW", "3.15")
    info("cfa:3:LSV", "MAJOR")
    info("cfa:3:LOPR", "-60")
    info("cfa:3:HOPR", "300")

}
```

Summary

- Introducing new EPICS module: Conditional Field Adjuster (CFA)
- Allows to define (alarm) fields in a more flexible way
- Use cases beyond alarming
- Phoebus properly displays the field adjustments

Next steps

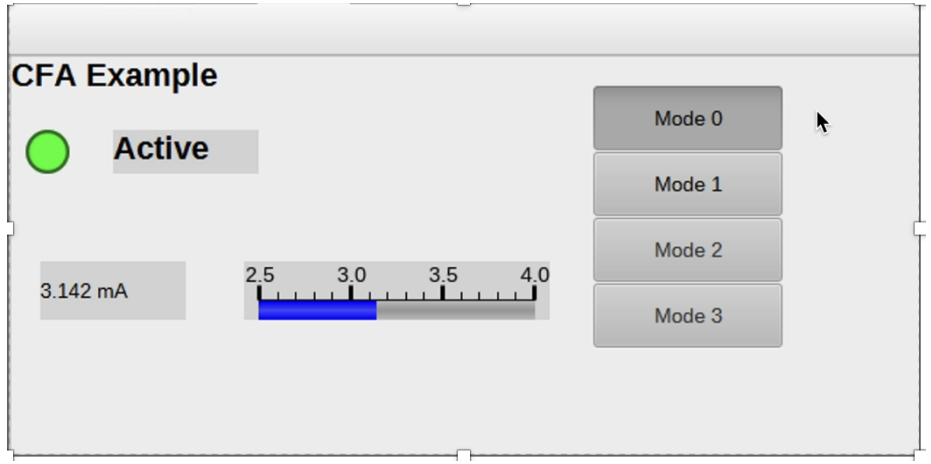
- More real-word testing
- Add tests
- Add more examples
- Start using in production
- Release 1.0

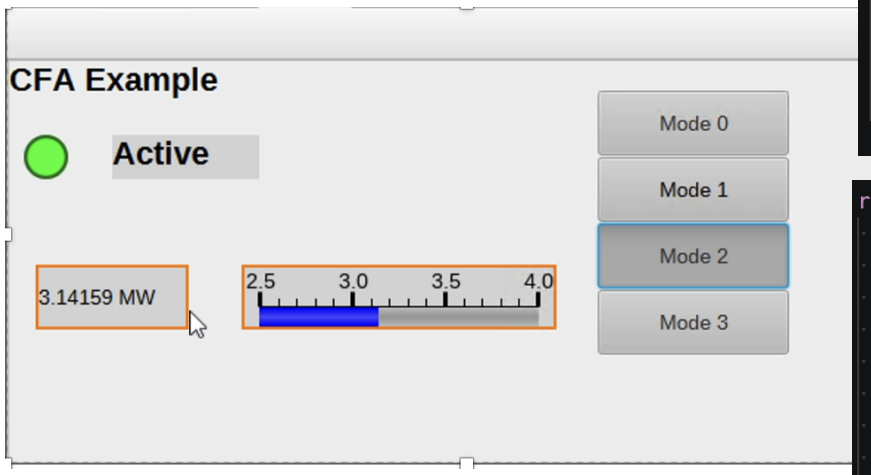
- Try current preview version!
- More technical description in README

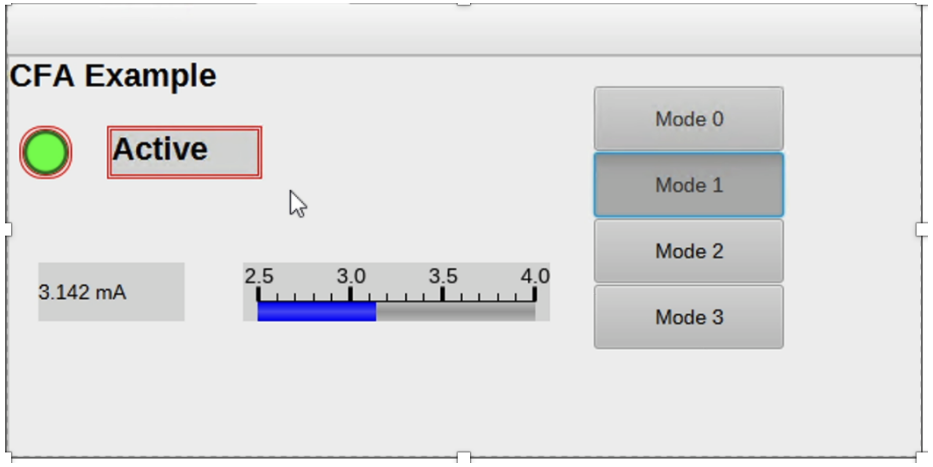
https://github.com/KIT-IBPT/conditional-field-adjuster

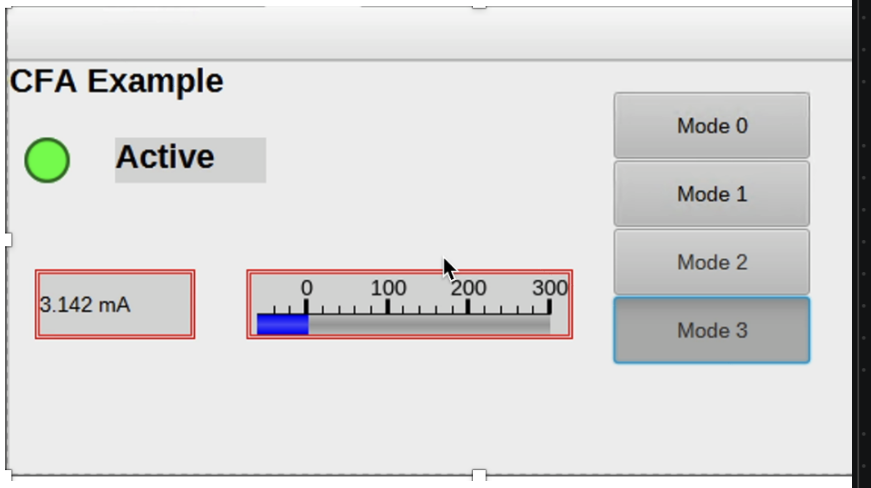
Backup - Demonstration

Demonstration Backup










```
record(bi, "$(P)$(R)Status") {
    field(DESC, "Status simulator")
    field(VAL, "1")
    field(PINI, "YES")
    field(ZNAM, "Inactive")
    field(ZSV, "MAJOR")
    field(ONAM, "Active")
    field(OSV, "NO_ALARM")

    # CFA fields
    info("cfa:select", "$(P)$(R)Mode CP")
    info("cfa:1:ZSV", "NO_ALARM")
    info("cfa:1:OSV", "MAJOR")
}
```

```
record(ai, "$(P)$(R)Value") {
    field(DESC, "Value simulator")
    field(VAL, "3.14159")
    field(PINI, "YES")
    field(PREC, "3")
    field(LOW, "3.0")
    field(LSV, "MAJOR")
    field(HIGH, "3.2")
    field(HSV, "MAJOR")
    field(SCAN, "1 second")
    field(EGU, "mA")
    field(LOPR, "2.5")
    field(HOPR, "4.0")

    # CFA fields
    info("cfa:select", "$(P)$(R)Mode CP")

    info("cfa:2:LOW", "3.15")
    info("cfa:2:LSV", "MINOR")
    info("cfa:2:PREC", "5")
    info("cfa:2:EGU", "MW")
    info("cfa:2:SCAN", "Passive")

    info("cfa:3:HIGH", "3.18")
    info("cfa:3:LOW", "3.15")
    info("cfa:3:LSV", "MAJOR")
    info("cfa:3:LOPR", "-60")
    info("cfa:3:HOPR", "300")

}
```