

EP,CICS

Experimental Physics, Chemistry and Industrial Control System — EPICS for the small chemistry lab

AGENDA

TALK OVERVIEW

Four chapters, from a provocative title to open questions

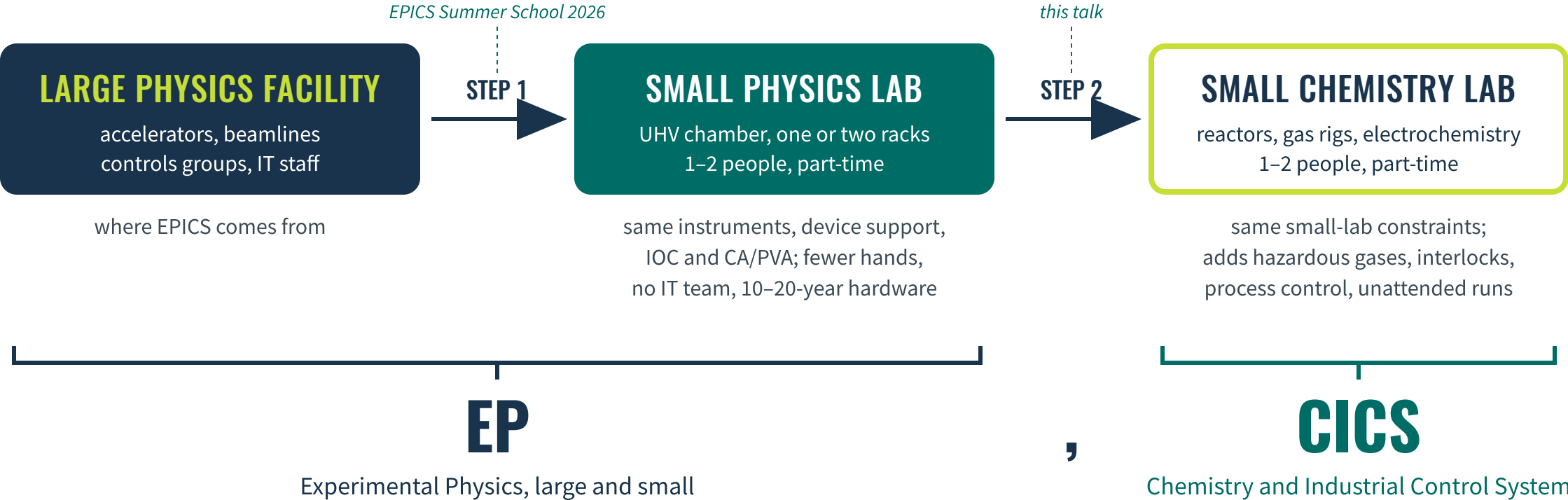
- 1. Same Bench, Different Discipline** — large facility → small physics lab → chemistry bench
- 2. What Chemistry Adds** — interlocks, hazards, closed-loop process control
- 3. Projects & Examples** — what already runs at FHI
- 4. Conclusion & Discussion** — questions for the collaboration

WHY "EP,CICS"

- EPICS = **Experimental Physics** and Industrial Control System. The "physics" in the name carries a lot of unstated assumptions
- Small chemistry labs (gas-handling rigs, catalytic reactors, surface-science setups) face the **same control problem** as small physics labs: few hands, long-lived hardware, no dedicated IT
- The question for this talk: what changes, and what doesn't, when the experiment is a **chemical process** instead of a beamline?

Short answer up front: most of it carries over directly. The part that changes is how much weight the IOC carries as a **safety layer**.

TWO STEPS, ONE FRAMEWORK



Each step keeps the same EPICS core. What changes is who runs it, and what it has to protect.

CHAPTER 1

SAME BENCH, DIFFERENT DISCIPLINE

From large facility to small lab, from physics to chemistry

STEP 2: THE SAME CONSTRAINTS IN A CHEMISTRY LAB

STILL TRUE IN A CHEMISTRY LAB

- 1 – 2 people covering controls, part-time, usually a PhD student or a postdoc
- Reactors and gas racks kept running 10–20 years
- No dedicated software/IT team
- Budget favours reuse over custom builds
- Setups get **rebuilt** for every new catalyst or reaction

WHAT THAT DEMANDS

- A **stable, boring** core with few independent components
- Vendor- and OS-agnostic drivers
- Something the **next student** can still read
- Configuration, not code, when the setup changes
- Long-term community support, not one maintainer

THE CHEMISTRY BENCH, INSTRUMENT BY INSTRUMENT

CATEGORY	TYPICAL INSTRUMENTS	INTERFACE	EPICS TODAY
Gas flow	Mass-flow controllers	RS232/485, Modbus	<code>.proto</code> or <code>modbus</code>
Pressure	Gauges, back-pressure regulators	analogue, RS232	<code>.proto</code> / ADC
Temperature	Furnace & heater controllers	Modbus RTU/TCP	<code>modbus</code>
Valves	Solenoid & pneumatic valves	digital I/O, RS232	DIO / <code>.proto</code>
Gas safety	H ₂ , CO, toxic-gas detectors	relay + 4–20 mA, Modbus	DIO / ADC
Analytics	Gas chromatograph (Agilent)	vendor .NET SDK only	C# IOC
Electrochemistry	Potentiostat	vendor Python SDK	P4P soft IOC

Top five rows: routine integration. Last two: closed vendor software, where the new work sits. FHI has done it for both.

On the instrument level, a chemistry bench is mostly a vacuum lab with more gas bottles.
Most of it is relabeling a problem EPICS has already solved.

CHAPTER 2

WHAT CHEMISTRY ADDS

Hazards, interlocks, and process control that physics rarely needs

STEP 2: WHERE PHYSICS AND CHEMISTRY LABS DIFFER

	SMALL PHYSICS LAB	SMALL CHEMISTRY LAB
Main hazard	HV, cryogenics, vacuum	flammable, toxic, reactive gases ; overpressure; heat
Typical failure	lost beamtime, broken filament	release, fire, runaway reaction
IOC's main job	data acquisition, sequencing	process control + interlocking
Control loops	mostly in the instrument	across instruments: temperature ↔ flow ↔ pressure
Time constants	μs – s	s – hours (ramps, long runs)
Unattended	sometimes	routinely : overnight and weekend runs

The IOC-as-safety-layer argument from the small-lab talk gets **stronger** in chemistry, not weaker.

THE IOC AS A SAFETY LAYER, AND WHAT IT IS NOT

HARDWIRED LAYER: STAYS HARDWARE

- Gas detector relay → solenoid valve closes, **no software in the path**
- Burst discs, pressure relief, emergency stop
- Certified where regulation requires it
- EPICS is **not** a SIL-rated safety system, and shouldn't pretend to be

IOC LAYER: CATCHES IT EARLIER AND EXPLAINS IT

- Cross-instrument permits: no H₂ unless T, p, and gas detection are all OK
- **Fail-safe on INVALID:** a dead serial link closes the valve, it doesn't freeze it
- Controlled shutdown sequence instead of a hard trip
- Every trip time-stamped and archived, so you can see *why*

CLOSING THE LOOP ON THE REACTION

- Physics small labs mostly **log**; a reactor has to be **driven**: temperature ramps, flow ratios, pressure set points, held for hours
- Already in the toolbox: `epid` / `pid` records for software loops, the **sequencer** (SNL) for ramp-and-soak programs, `calcout` for derived set points
- Many loops are better left **in the instrument** (furnace PID, MFC flow loop); the IOC supervises set points and limits
- Example from hp-spm: the VAT variable leak valve runs its own **pressure-control mode**, and the IOC only sends the target pressure and watches the result

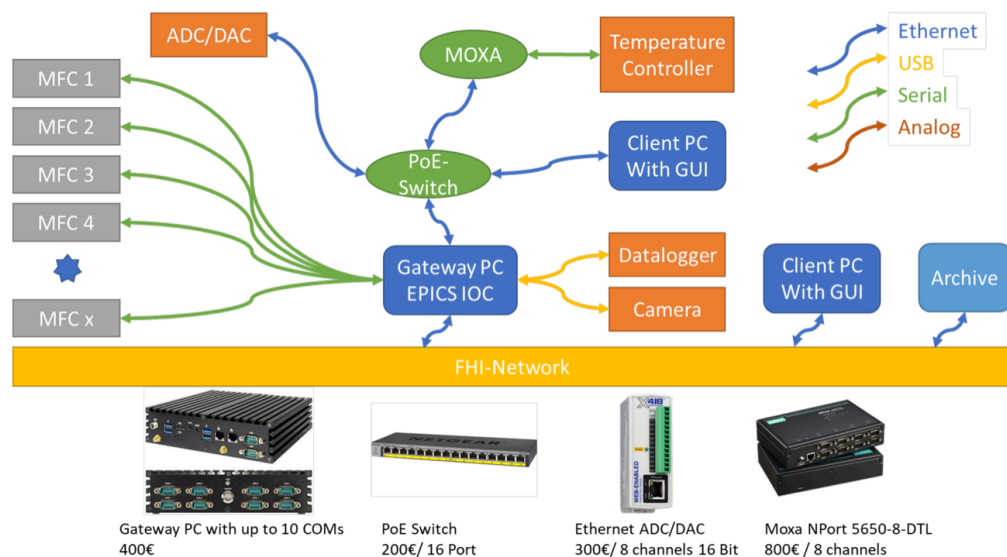
Rule of thumb: fast loops stay in the device, cross-instrument loops go in the IOC, experiment recipes go in a client above it.

CHAPTER 3

PROJECTS & EXAMPLES

What already runs at FHI

ONE RECIPE FOR EVERY NEW LAB: THE EDGE SERVER

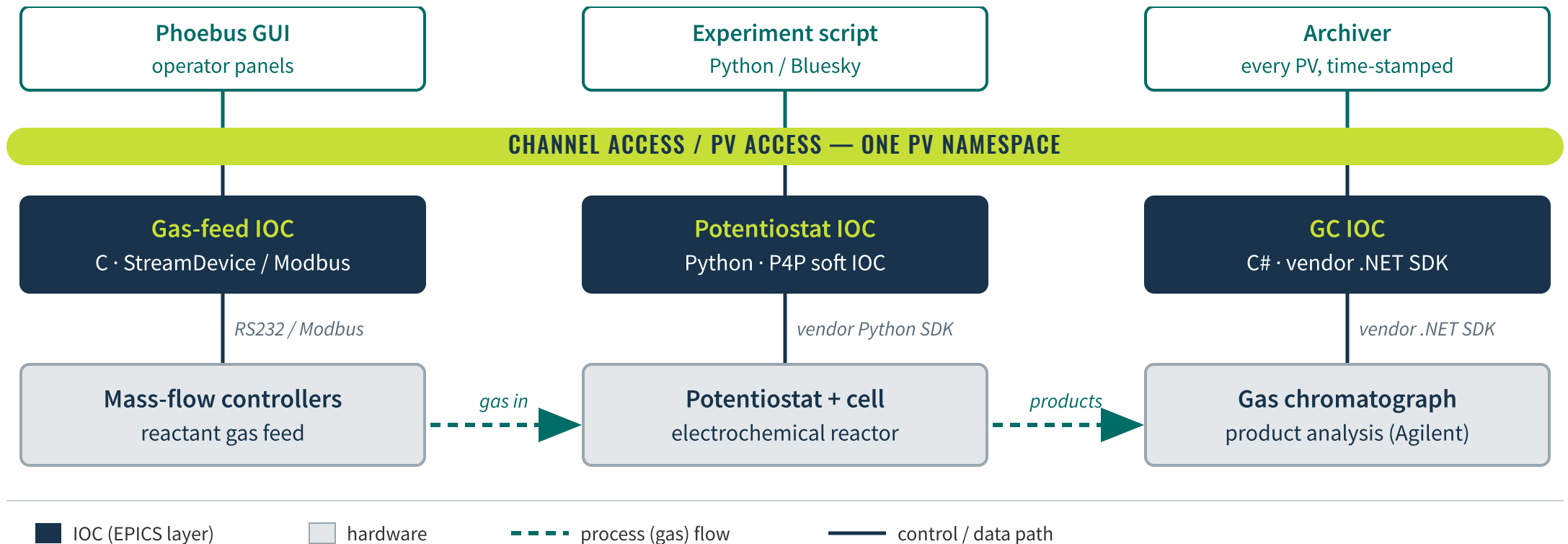


- Each new small-lab setup at FHI, including the MFC-based gas rigs shown here, gets the same ~400 € gateway PC with up to 10 COM ports, running one EPICS IOC, reached over the FHI network by GUIs and an archiver
- `epics-edge-setup` provisions it end to end. A chemistry group plugs in its instruments; it doesn't have to install Linux

FROM THE LABS: THE IOC NOTICES, IT DOESN'T JUST LOG

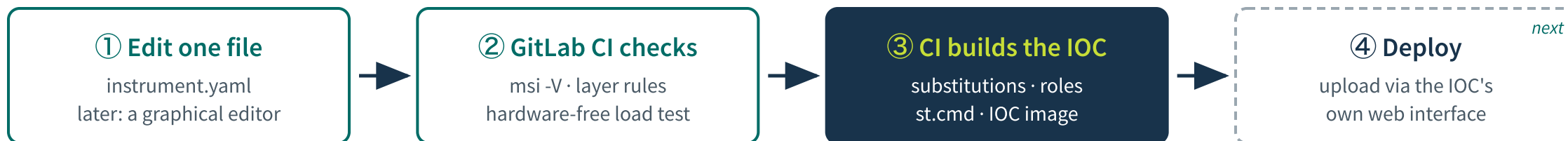
- **Chemical reactor:** the IOC monitors the gas-detection system and shuts off the gas supply
- **UHV surface science (hp-spm):** a leak is caught through the vacuum pressure sensors, and the affected section is isolated using the valve pressure sensors
- **RGA filament:** can't be switched on by accident, because the interlock checks system pressure first
- **Gas dosing:** the VAT leak valve doses reactant gas into UHV in pressure-control mode, with the target set from the IOC
- **Gas analysis:** Agilent GCs are now PVs on the network, next to everything else (next slide)
- 24 commercial instruments in hp-spm, almost all of them one `.proto` + one `.db` file

AN ELECTROCHEMISTRY BENCH ON EPICS



- Three kinds of IOC, **one PV namespace**: clients don't care what language an IOC is written in
- The two "closed box" instruments only ship a vendor SDK, so their IOCs are **Python** and **C#**, not C

BUILD AN IOC BY EDITING ONE FILE

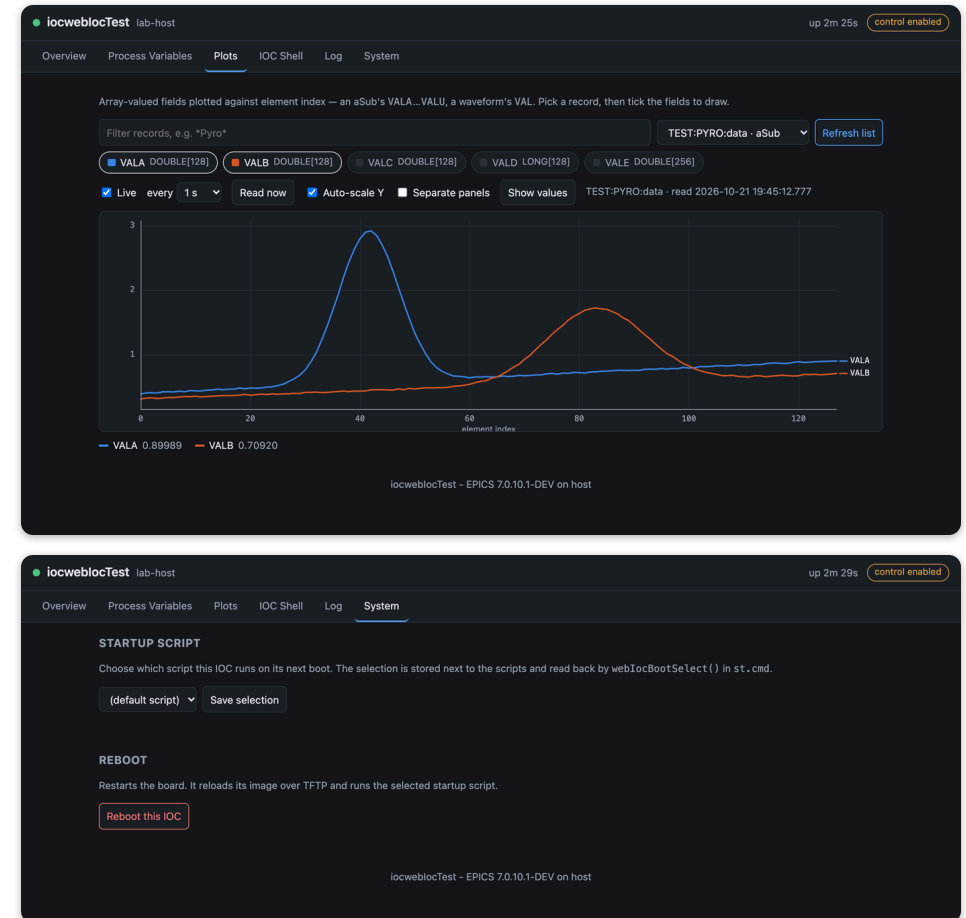


```
# instrument.yaml: the only file a user edits
identity: {name: RX2, prefix: "RX2:"}
devices:
  - id: MFC1
    support: mfc_serial
    bus: {type: serial, dev: /dev/COM3, baud: 38400}
  - id: TC1
    support: furnace_modbus
    bus: {type: tcp, host: "10.0.0.21:502"}
roles:
  FEED:FLOW: {source: "RX2:MFC1:Flow", unit: mln/min}
  REACTOR:T: {source: "RX2:TC1:PV", unit: K}
```

- **Goal:** users with no programming or Unix experience combine existing devices for a new experiment by editing **one file**
- Device units and procedures are reused 1:1; only the definition is per setup
- GitLab CI/CD **checks** the file, then **builds the whole IOC** from it
- A graphical editor on top of the same file comes later

EVERY IOC GETS A WEB INTERFACE

- HTTP server **linked into the IOC**: on RTEMS one image, about +77 kB
- **Today**: PVs, array plots, IOC shell, live log, next startup script, reboot. Read-only without credentials
- Starts **before `iocInit`**: a broken `st.cmd` still leaves the board reachable
- **Next**: upload a new IOC image and `st.cmd` from CI through the browser (step ④)



Screenshots: host test build of the same module (Plots and System tabs).

RELABELING VS. GENUINELY NEW

ALREADY SOLVED, JUST RENAMED

- MFCs, gauges, heaters, valves → StreamDevice / Modbus
- Interlocks → `calcout` , alarm severities, IVOA
- Ramps and loops → `epid` , sequencer
- Archiving and alarms → Archiver Appliance, Phoebus alarm server
- Remote and unattended operation → CA/PVA

WHERE NEW WORK IS NEEDED

- **Closed analytics:** vendor SDKs only. Solved at FHI with C# and Python IOCs, but each instrument is new work
- **Recipes:** versioned reaction programs that a chemist can edit without touching a `.db` file
- **Safety documentation:** how to describe an IOC interlock to a safety officer
- **Research data:** linking PVs to sample, catalyst, and run metadata (FAIR / ELN)

CHAPTER 4

CONCLUSION & DISCUSSION

Open questions for the collaboration

TAKEAWAYS

- **Two steps, one framework:** a small physics lab is a large facility with fewer hands; a small chemistry lab is a small physics lab with more hazards. The same EPICS core serves all three: that's **EP,CICS**
- Most of the chemistry bench is **relabeling**: StreamDevice, Modbus, `calcout` and `epid` already cover it
- What chemistry adds is **weight on the IOC as a safety layer**: cross-instrument permits, fail-safe on INVALID, every trip archived. This sits *beside* hardwired protection, not in place of it
- The genuinely new work is at the edges: closed analytics, recipes, and research-data links. A closed-loop electrochemistry bench at FHI shows it can be done, with EPICS underneath and automation on top

QUESTIONS FOR THE ROOM

The bitter pill: many instrument vendors (GCs, potentiostats, ...) offer no open protocol, only a .NET or Python SDK. Then the IOC *has* to be C# or Python. That weakens the "compiled C IOC" argument, and we are not happy about it. But it is where small chemistry labs are today.

- How do you deal with closed vendors: wrap the SDK, push the vendor for an open protocol, or avoid the hardware?
- Who already runs EPICS on a **chemistry experiment**? What did you have to write yourself?
- **Shared device support** for common lab hardware (MFCs, furnace controllers, gas detectors) in one place?
- **Worked examples** (interlocks, ramp-and-soak, fail-safe defaults), or a small **working group**?

Bring your counter-examples. This talk is meant to start a discussion, not close one.



THANK YOU

Questions & discussion welcome

Heinz Junkes · junkes@fhi-berlin.mpg.de · Fritz-Haber-Institut der Max-Planck-Gesellschaft

SOURCES & FURTHER READING

- **hp-spm**: UHV-SPM EPICS IOC, 24 commercial instruments, one `.proto` / `.db` pair per device:
`gitlab.fhi.mpg.de/fhi-automation/hp-spm`
- **epics-edge-setup**: provisioning script for the FHI EPICS edge server: `gitlab.fhi.mpg.de/epics-tools/epics-edge-setup`
- **StreamDevice**, **asyn**, **modbus**, **std** (`epid`), **sequencer**: the EPICS support modules this talk relies on, all via epics-controls.org
- "EPICS in Small Laboratories", EPICS Summer School 2026: the longer companion talk (IOC vs. Python, real-time priorities on RTEMS/Linux)
- EPICS Collaboration: documentation, training, and the full module ecosystem: epics-controls.org

BACKUP

BACKUP SLIDES

For questions and discussion

STEP 1, RECAP: LARGE FACILITY VS. SMALL PHYSICS LAB

THE SAME

- The same commercial instruments: pumps, gauges, temperature controllers, power supplies, motion
- Decades of device support, already written: a `git clone`, not a project
- The same IOC, the same CA/PVA, the same tools
- The same need for reliable, unattended operation

DIFFERENT

- 1 – 2 people, part-time, instead of a controls group
- No IT team: the platform has to come as one recipe (the edge server)
- Hardware kept 10–20 years
- The prejudice "EPICS is too heavy for three racks" (it isn't)

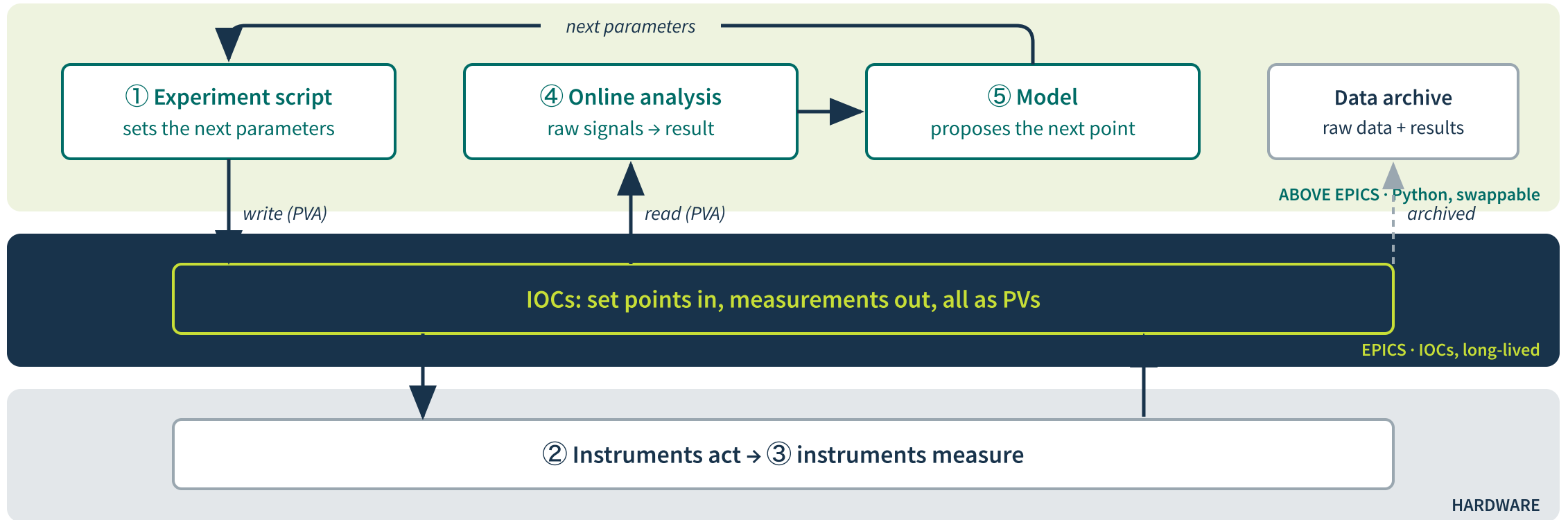
Summer-school conclusion: small labs have fewer hands, not lower reliability requirements.

AN INTERLOCK IS JUST A FEW RECORDS

```
# gas_interlock.db – illustrative
record(calcout, "$(P)H2:Permit") {
    field(INPA, "$(P)GasDet:Alarm CP MS")    # H2 detector, 0 = OK
    field(INPB, "$(P)Reactor:P CP MS")       # reactor pressure
    field(INPC, "$(P)Reactor:T CP MS")       # reactor temperature
    field(CALC, "A=0 && B<$(P)MAX && C<$(T)MAX")
    field(OOPT, "When Zero")                 # only ever writes "close"
    field(DOPT, "Use OCAL")
    field(OCAL, "0")
    field(IVOA, "Set output to IVOV")        # INVALID input -> close
    field(IVOV, "0")
    field(OUT, "$(P)H2:Valve PP")
}
```

No C, no Python service. Re-opening stays a deliberate operator action, and the hardwired detector relay still closes the valve independently.

AUTOMATION ON TOP, EPICS UNDERNEATH



- Set → run → measure → analyse → model proposes the next point → repeat, with **no operator in the loop**
- Everything above the PVA line is Python and can change from one experiment to the next; the IOCs below it stay as they are