

The decade long transition from a custom control system to EPICS

Edmund Blomley on behalf of the KIT team



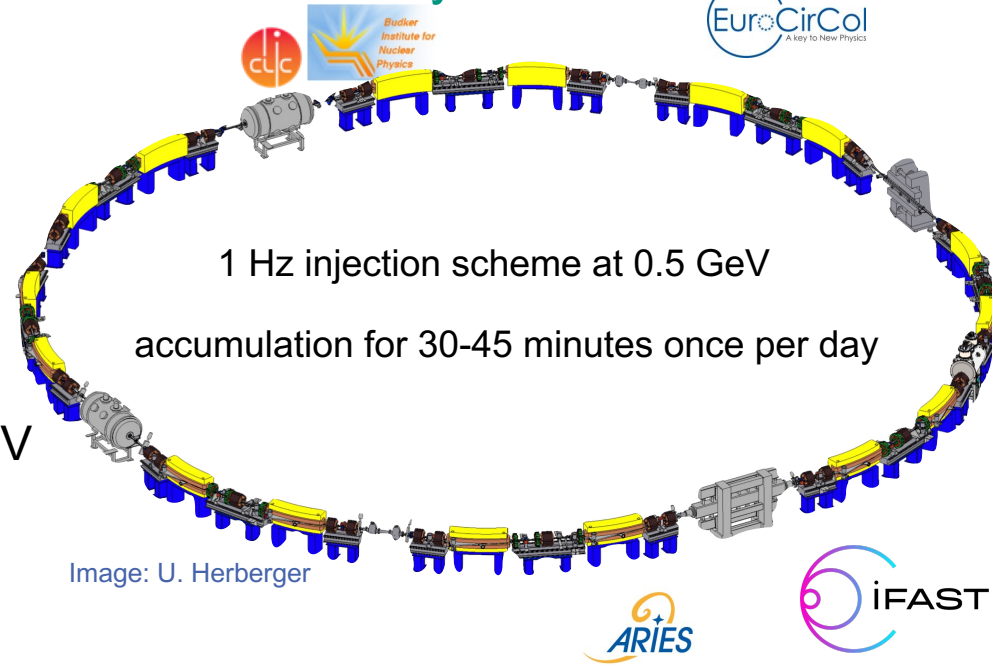
Karlsruhe Research Accelerator (KARA)

■ KIT synchrotron lightsource & accelerator test facility

- until 2015 known as „ANKA“

■ Key parameters

- Build 2001
- Circumference: 110.4 m
- Ramped storage ring: 0.5 - 2.5 GeV



The Original Control System: ACS

■ ACS

- ALMA common software (<https://ascl.net/1302.003>)
- Advanced control system

■ Motivation

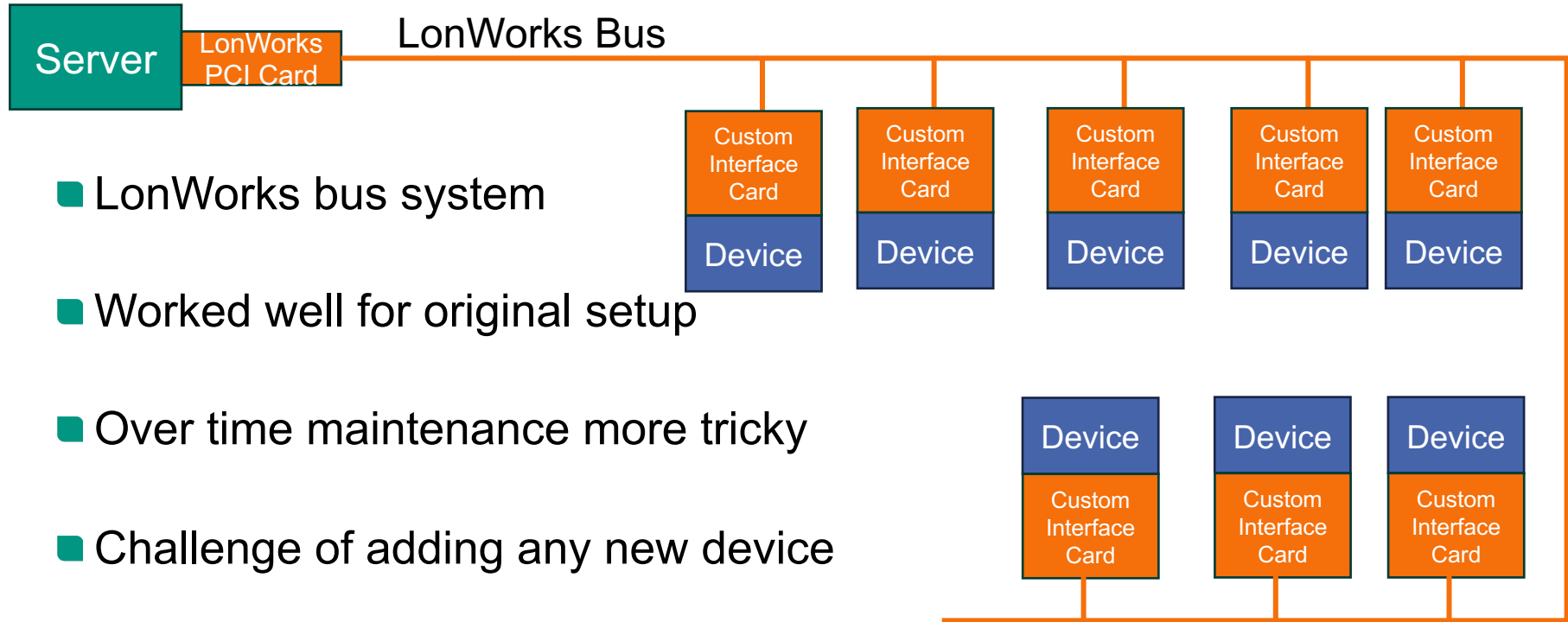
- Object-oriented interface abstraction layer
- Provide same interface for all power supplies independent of manufacturer

■ Custom extensions for ANKA

- LonWorks bus (at that time, cheap and efficient for the amount of data)
- Special hardware control cards

First Control System

ACS & LonWorks Layout



- LonWorks bus system
- Worked well for original setup
- Over time maintenance more tricky
- Challenge of adding any new device

Migration to a new Control System?

- Issues with ACS in regard to scale– and extendability
- Around 2007 looking for an alternative control system
- No dedicated controls group
 - “What is CERN doing?”
- Decision was made to introduce to PVSS/WinCC
 - GUI(!)
- support for PLCs
- Integration of MRF timing system via UDP
 - By choice to avoid VME

Second Control System

EPICS: First Contact

- 3-4 years of investment into PVSS
 - Spoiler alert: PVSS didn't get much traction on the accelerator side
- Around 2010: Tango or EPICS?
- New BPM electronics
 - Comparing Tango and EPICS implementation
- Decision was made to switch to EPICS
- Control System Studio (CSS) as the new GUI

Third Control System

Design Decisions

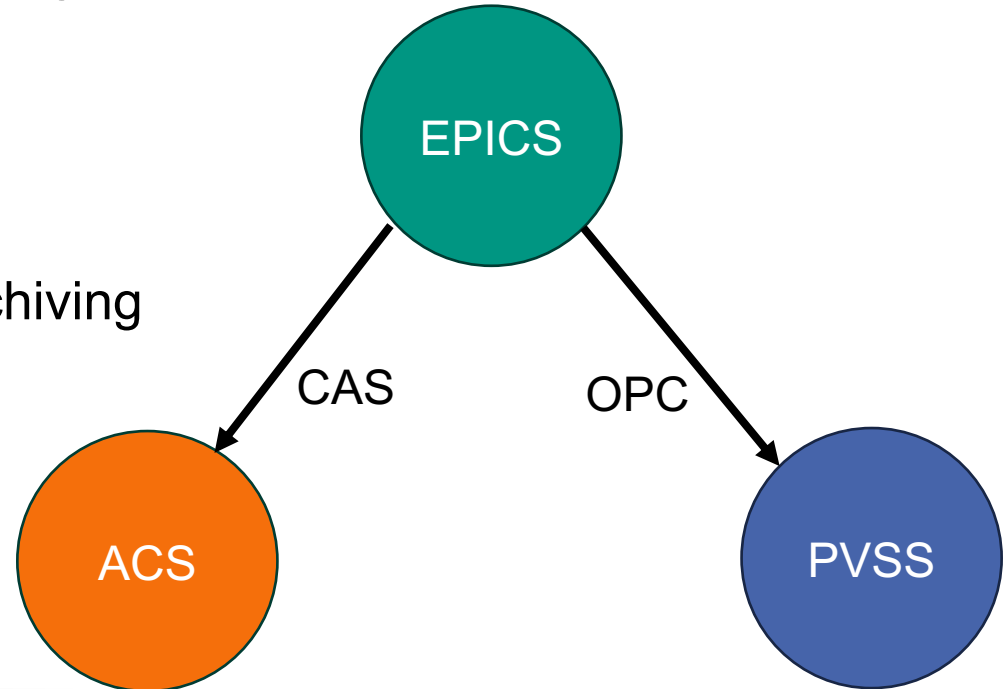
- Ubuntu desktop and server as main OS
 - Use virtual machines as much as possible
- Custom EPICS build toolchain creating Debian packages
 - Only build modules we need
 - Add local patches as needed
- Cassandra database for archiving raw data
- Naming conventions...
 - `<accelerator>:<location>:<device group>:<device name/number>:<properties>`
 - Avoid too many abbreviations
 - Local patch of EPICS base to allow PV length up to 255 characters

Migration Plan

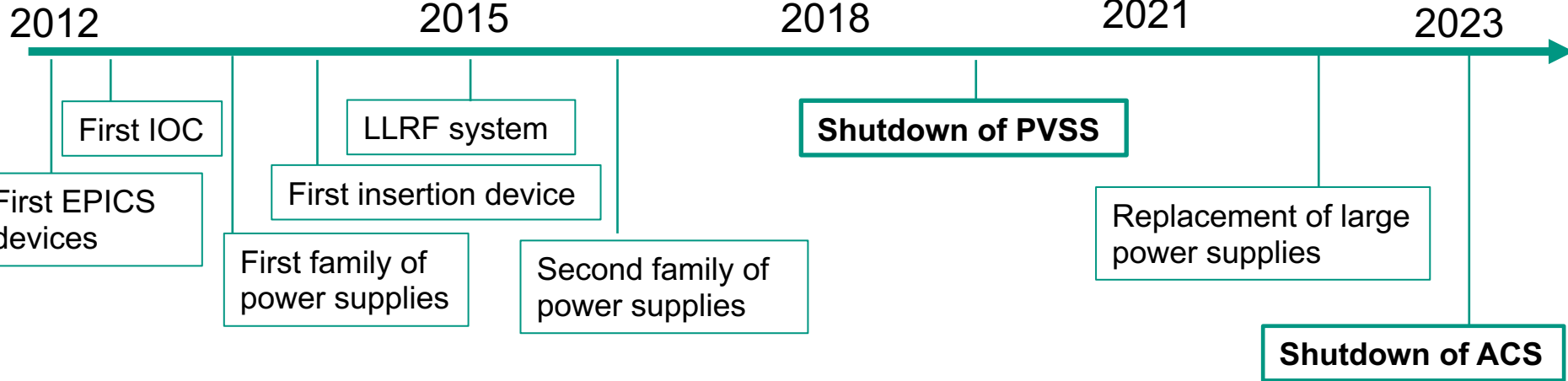
- Ethernet or serial communication directly possible?
 - Beam current DCCT: create IOC to read out serial interface
- Possible to exchange hardware interface?
 - Storage ring quadrupole magnets: replaced controls interface
- Exchange individual devices
 - Ideally commercially available (no custom power supplies)
 - TCP/UDP (or serial) interface mandatory
- No migration of complex systems
 - Analog LLRF: replace whole system

Gateways

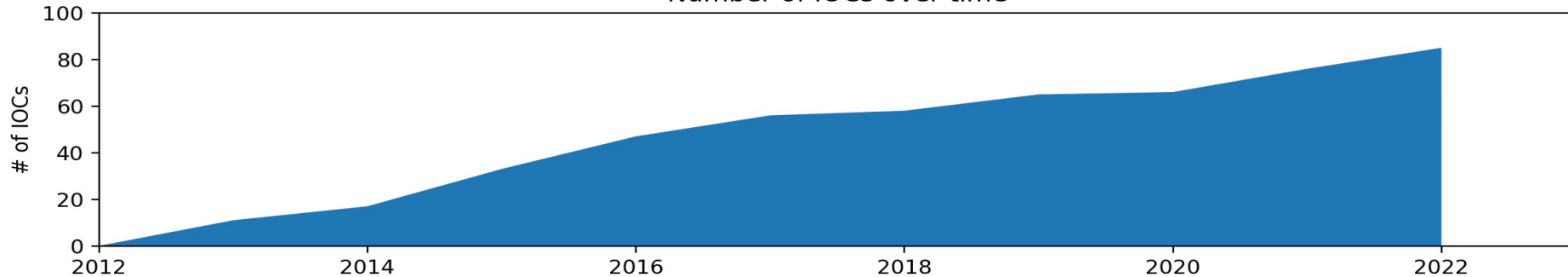
- Implement gateways while “waiting“ for native implementation
- At least basic access
 - Status, on, off, set value
- Allow GUI control, alarming, archiving via EPICS/CSS



Migration Milestones



Number of IOCs over time



Status

~94 IOCs

~700 IP devices

~80,000 PVs

~30 PLCs

- Recent focus on Python-based IOCs and AI integrations
- Centralised IOC management via Salt

TUPDP030

THPDP020

■ Next in line:

- Migration to Phoebus
- Fully transition to OPCUA (for PLCs)
- Bluesky?

More accelerators!

FLUTE

cStart

Laser Wakefield Accelerator...

Community Contribution?

- Working on new controls website
- Open access to our internal EPICS repositories and/or build products
 - Lots of automated workflows (CI/CD using GitLab)
 - EPICS Debian tool chain
 - Python-to-Debian tool chain (pyepics, softIOC, caproto, etc...)
 - IOCs for commercial devices
 - Establish documentation workflow (GitLab -> Readthedocs)
- Make our EPICS developments more visible
 - OPCUA, MRF, ...
 - Execute Device Support

Not just changing Control Systems...

