EPICS as DCS

TRG/DAQ Workshop

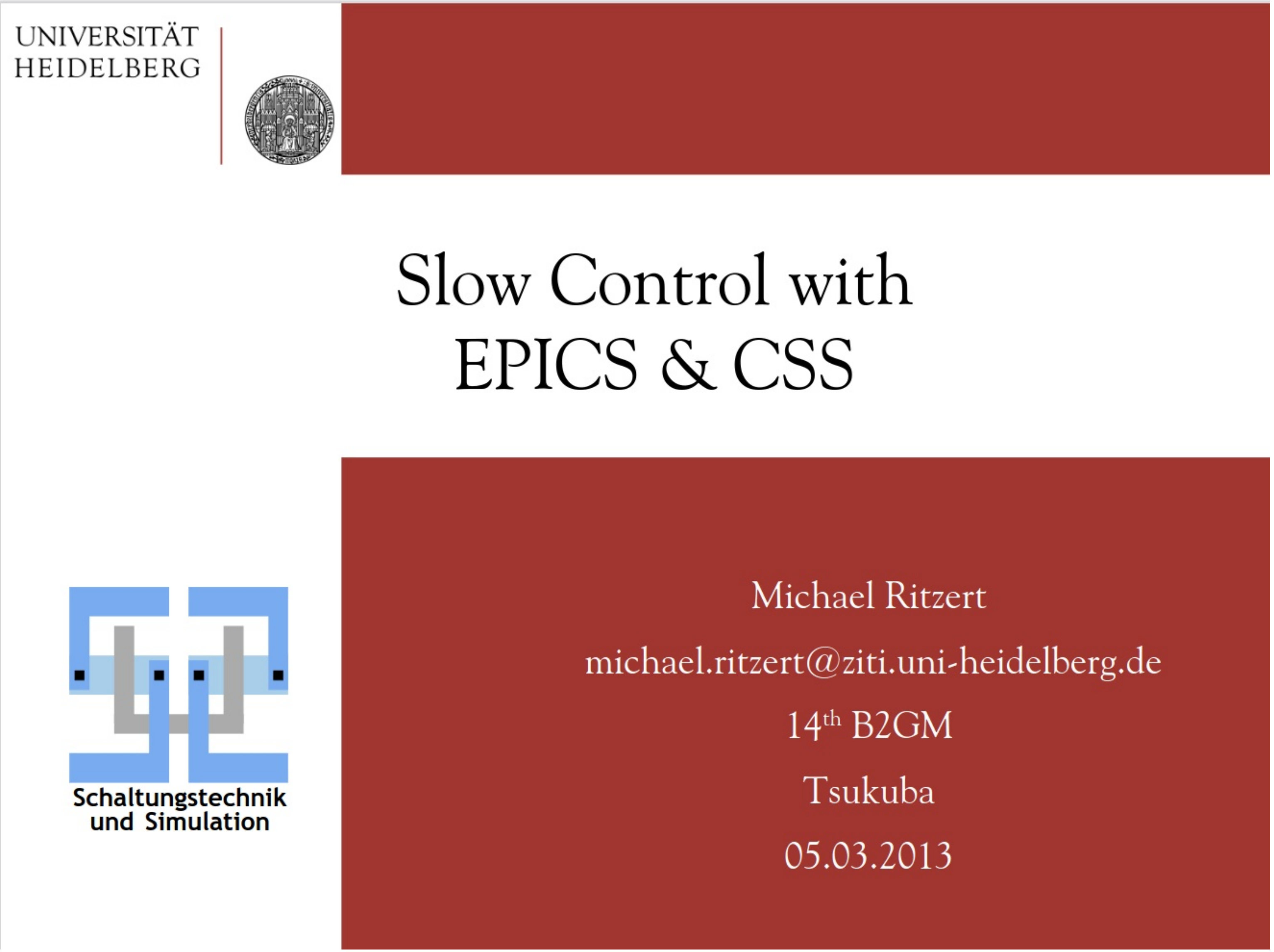
29.11-2.12.2022, Nara

Björn Spruck

EPICS

- Many slides stolen and inspired from

https://en.wikipedia.org/wiki/EPICS



UNIVERSITÄT
HEIDELBERG

Slow Control with EPICS & CSS

Schaltungstechnik und Simulation

Michael Ritzert

michael.ritzert@ziti.uni-heidelberg.de

14th B2GM

Tsukuba

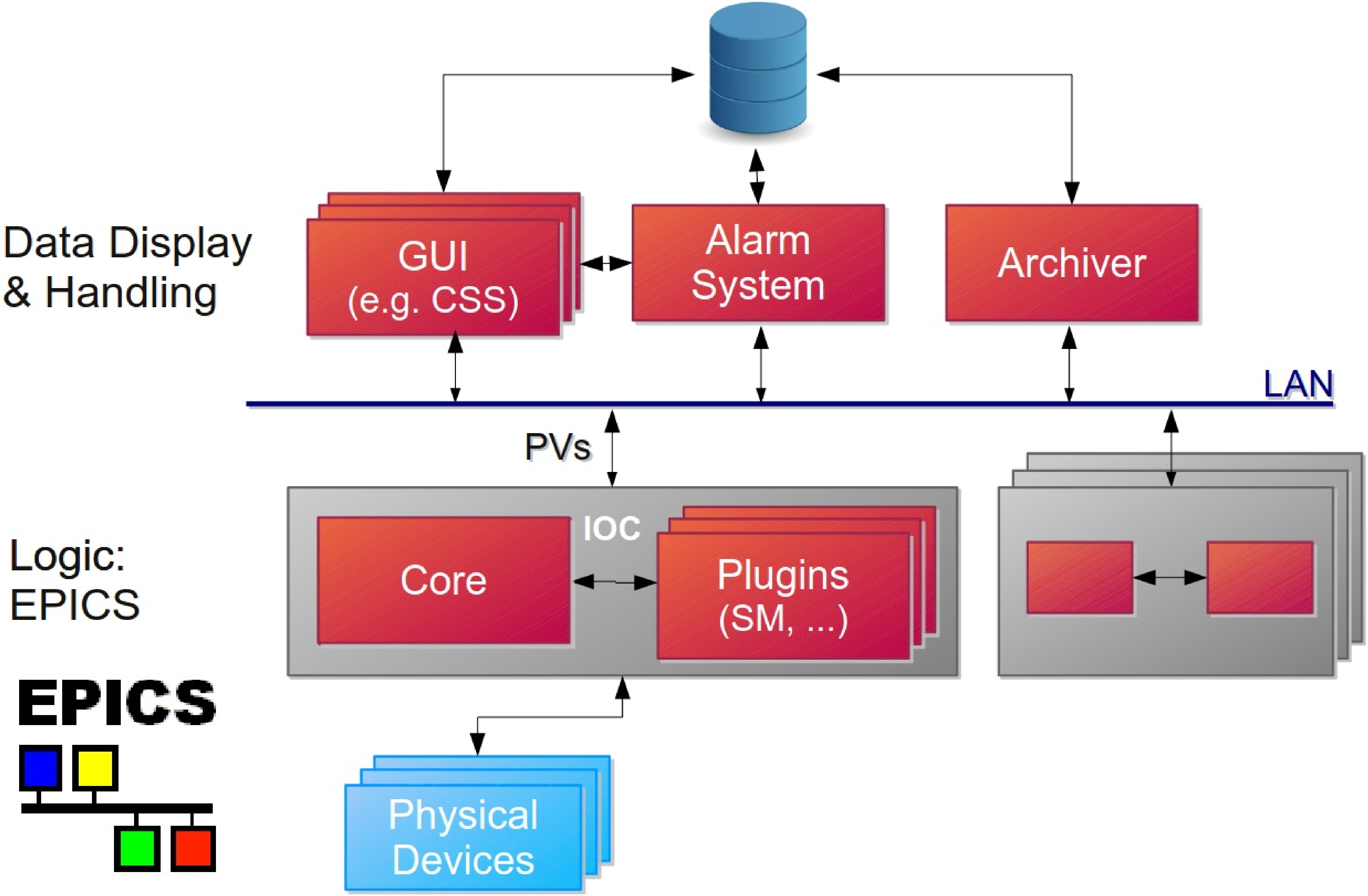
05.03.2013

EPICS

https://epics-controls.org/

- Experimental Physics and Industrial Control System
  - Used in many physics experiments worldwide.
    - Fermilab
    - SLAC
    - DESY, GSI
    - KEK(!), J-PARC
    - Lawrence Berkeley National Laboratory
    - Los Alamos National Laboratory
    - ITER
    - Spallation Neutron Source
  - Much knowledge available in the physics community.
  - Many of our requirements well known to the community and handled.
- Mature system
  - It is used since many years (first release: 1994) with active development and community
  - Available on a lot of architectures (e.g. crate controllers)
  - Many (typical) devices directly supported
- Experience and knowledge within the physics community (not only HEP)
  - Within Belle II: SuperKEKB, SVD, PXD, …

EPICS Architecture



Nomenclature: “PV”

- **P**rocess **V**ariable
  - Logical representation of a variable in the system.
    - Requested value of a setting, e.g.
      - Control bit in an FPGA/ASIC.
      - Set voltage of a PS
    - Current state of a parameter, e.g.
      - Temperature
      - Actual output voltage of a PS
  - Contains (amongst other things):
    - Current value (with unit)
    - Optional: Lower and upper bound for normal operation, alarm limits and severity
      - Alarm handler looks at this.
      - Also shown in most GUI displays.
    - Information on accuracy.
  - Different data type: binary / digital / analog / …
    - PV can be anything from a binary value, enum up to a megabyte size 2d-hitmap
  - All PVs are accessible over the network by their names in exactly the same way, independent of their implementation details.

Nomenclature: “IOC”

- **I**nput **O**utput **C**ontroller
  - Owner of specific PVs
  - Typically interfaces with hardware
  - Can also be plain software (“softIOC”)
  - Typically C/C++ but also Python is supported
  - C-like Sequencer language, simplifies developing State Machines
- IOCs independent of each other, no central instance (no “master”)
- Within an IOC, different threads/sequencers can run independently
- IOCs find each other by UDP unicast and then connect by TCP-IP
  - Disconnections automatically handled → possibility for alarms
  - Jave implementation

More Features

- Modular system
  - EPICS systems are inherently modular: Data provided by many IOCs running on different systems.
  - For single-device tests, EPICS can easily be run with only the minimal number of IOCs and the corresponding GUIs.
  - After system integration, the IOCs are just run in parallel, and the GUIs connected by hyperlinks.
- Scalable: 10s of 1000s of PVs, 100s of IOCs
- Full ecosystem around EPICS:
  - GUIs: CSS (older) and Phoebus (new) are obvious choices.
  - Alarm system
  - Archiver: several implementations
- Gateway: bridge between networks, shielding with access rules (e.g. write protect, user based)

- Lbnl: there are also companies which provide commercial support for EPICS integration

Control System Studio (CSS)

- GUI for EPICS
- Implemented in Java with and using Eclipse
- Combination of core features and user-specific addons.
- Features:
  - Simple creation of control panels („OPIs“) with graphical editor.
  - Large number of GUI items to display and modify PVs available.
- Easily extensible.
  - PV browser, PV trend display (live and archived)
  - Alarm GUI
  - Electronic Logbook integration
  - Plugins: e.g. NSM2 and DQM (not used anymore)
  - Run as standalone Java application, or started from the web.
  - Automatically check for and download new versions of panels keeping all clients up to date.
- Export displays to web server for easy remote access.
- CSS is nearly EOL → upgrade to Phoebus

EPICS Versions

- Version 3.xx is our default
- Version 4: PV-access as additional new protocol: structs (user specific). Not aware that V4 is used by anyone in Belle II
- Version 7 includes both 3+4
  - (version 7 is included in our basf2 external packages used by DQM)
  - → safe to upgrade (PXD will use 7.0.7 soon)

More on PVs

- PVs can be defined to execute update every few s or faster (“SCAN”)
- Calculation PVs can do computations on PVs, put them into series etc. This often avoids the need to write code to e.g. calculate a mean of 10 readings.
- Processing connection automatically trigger other PVs and their update.

Security

- Within the local network, anyone can read and write PVs. Communication is not encrypted. Gateways can be used to limit access to potentially critical PVs.
    - (such as PXD internal network is secured by PXD gateway to shield the critical module and power supply from bdaq net)

- NB: in Belle II, now every shifter has remote access to the bdaq network and (potentially) could do harm to the system. But this is not limited to EPICS but also to any nsm2 variable etc.

Summary

- EPICS is a mature and widely used basis for detector control systems
- The provided tools fit our needs and fulfill their purpose
- From experience we (PXD, SVD, …) had in operation until now, there is no reason to change to another basis for Slow Control.