MXSlowControl

MeπX

Stefan Lunkenheimer

MAGIX Collaboration Meeting 2017-02-17

Topics

EPICS Overview

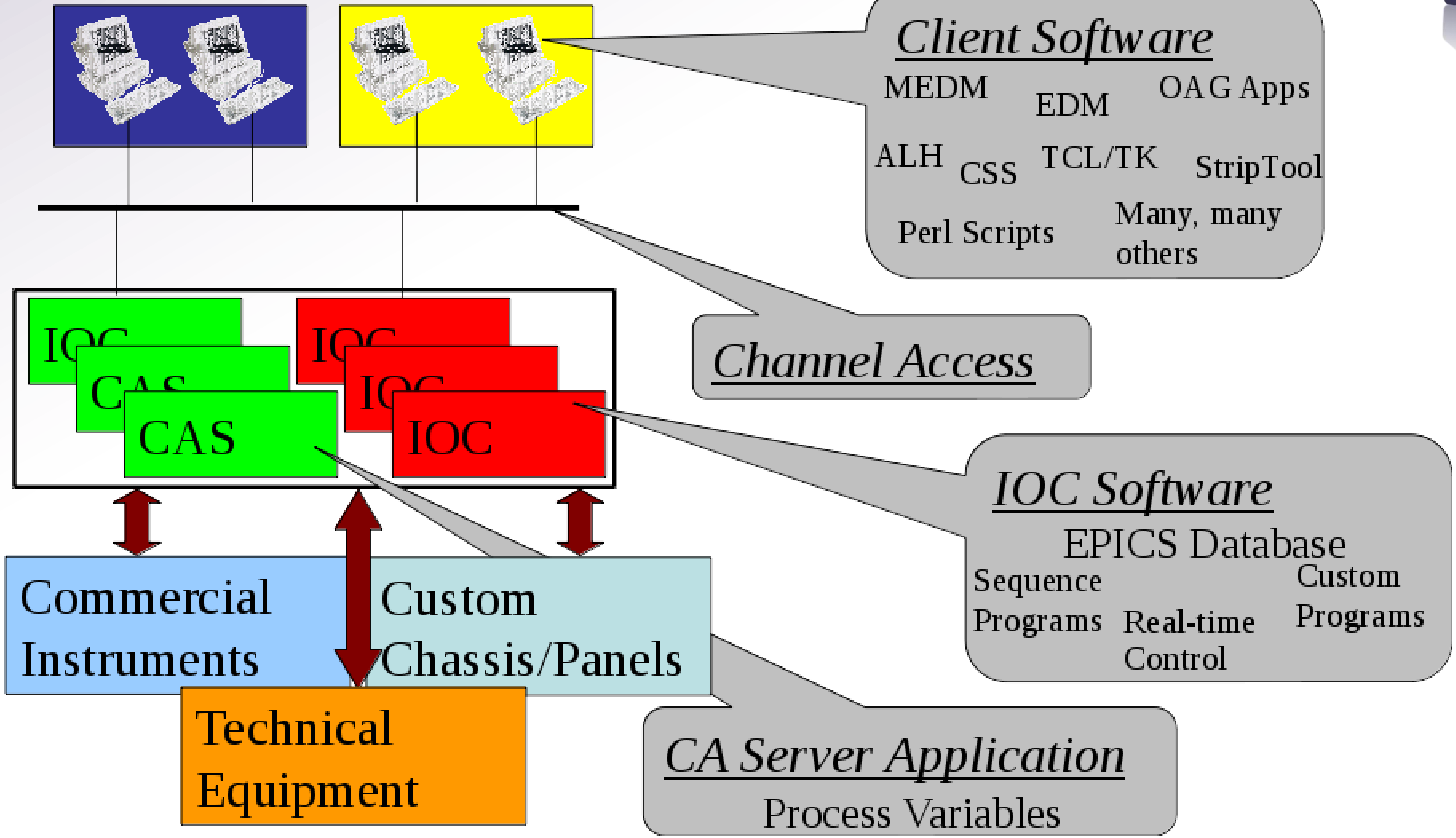
Status quo

Strategy / Outlook

EPICS

- Software to control physical experiments
- Distributed system
    - Multiple closed systems
- Maintained by community and external
    - A lot of solutions available

Structure



Client Software

MEDM EDM OAG Apps

ALH CSS TCL/TK StripTool

Perl Scripts Many, many others

Channel Access

IOC Software

EPICS Database

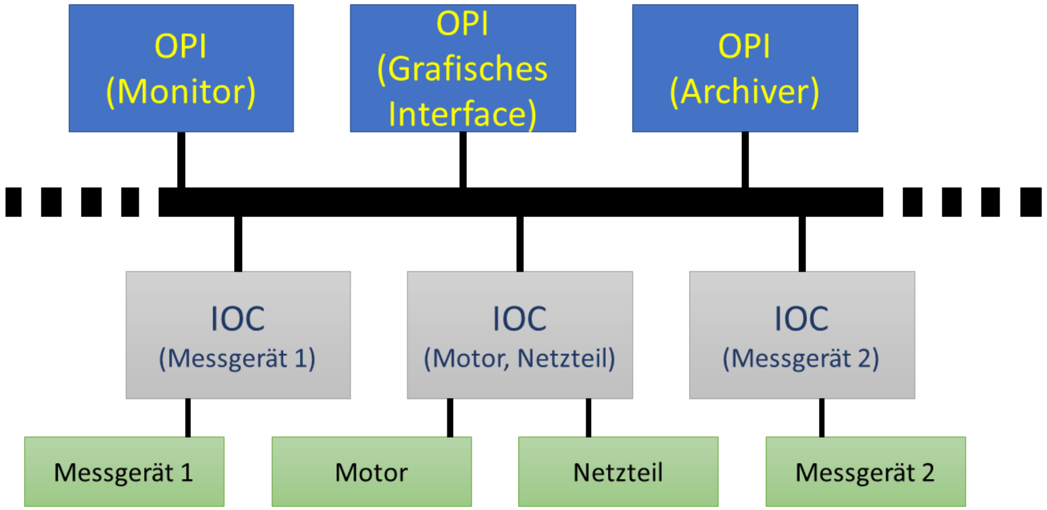
Sequence Programs Real-time Control Custom Programs

CA Server Application

Process Variables

Structure

- Channel Access
  - Communication Protocol
- Input-Output Controller (IOC)
- Operator Interface
  - EPICS Clients
  - Communicate via CA



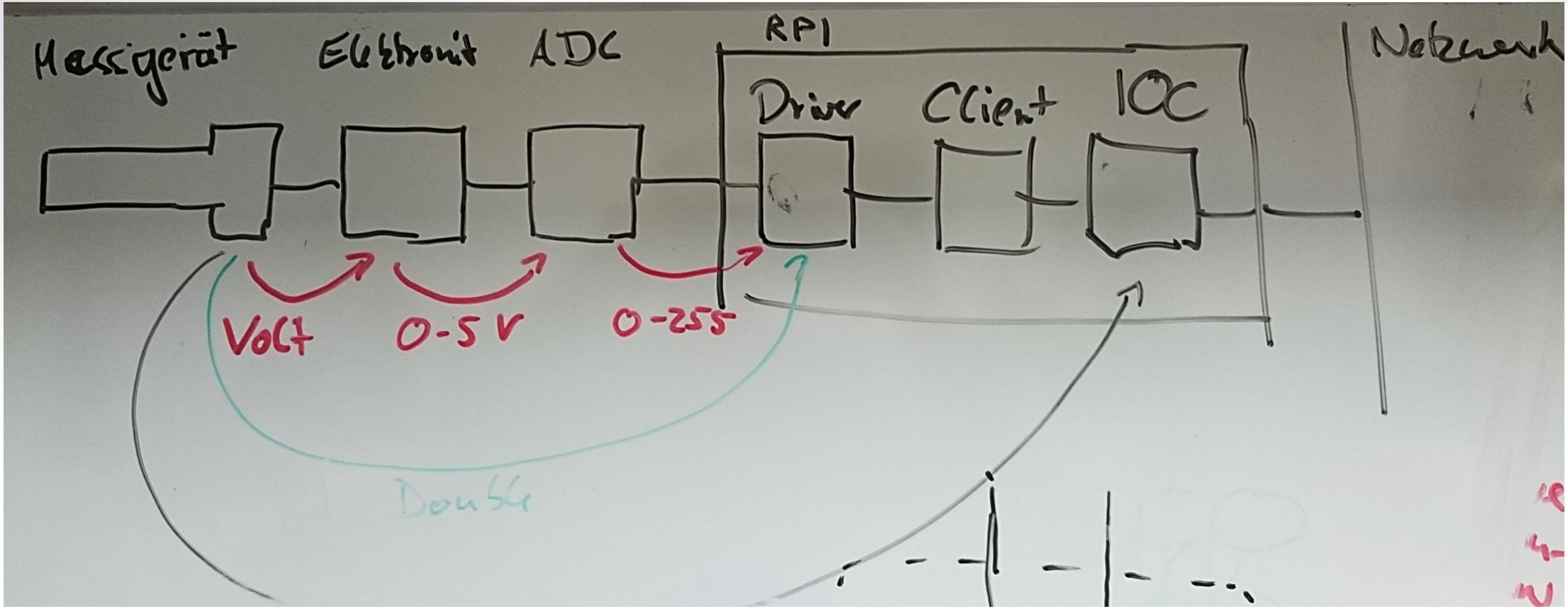
Server

- Multiple servers (IOC‘s)
- Defines Process-Variables with database
- Span virtual bus (Channel Access)
- Configuration for each IOC needed

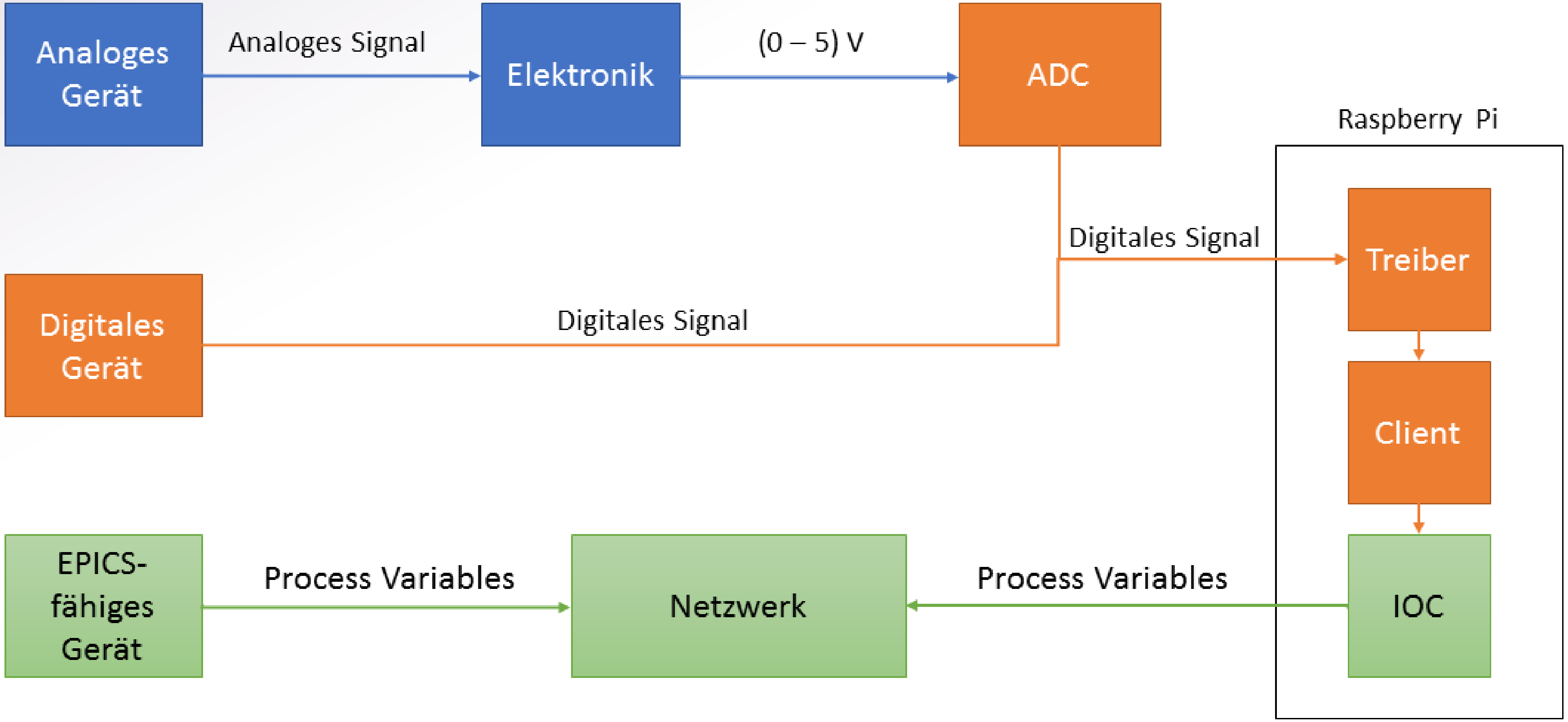
Client

- CSS, python, c++, many others
- Can get or put stuff in the PV's
- Monitoring
- Self configured

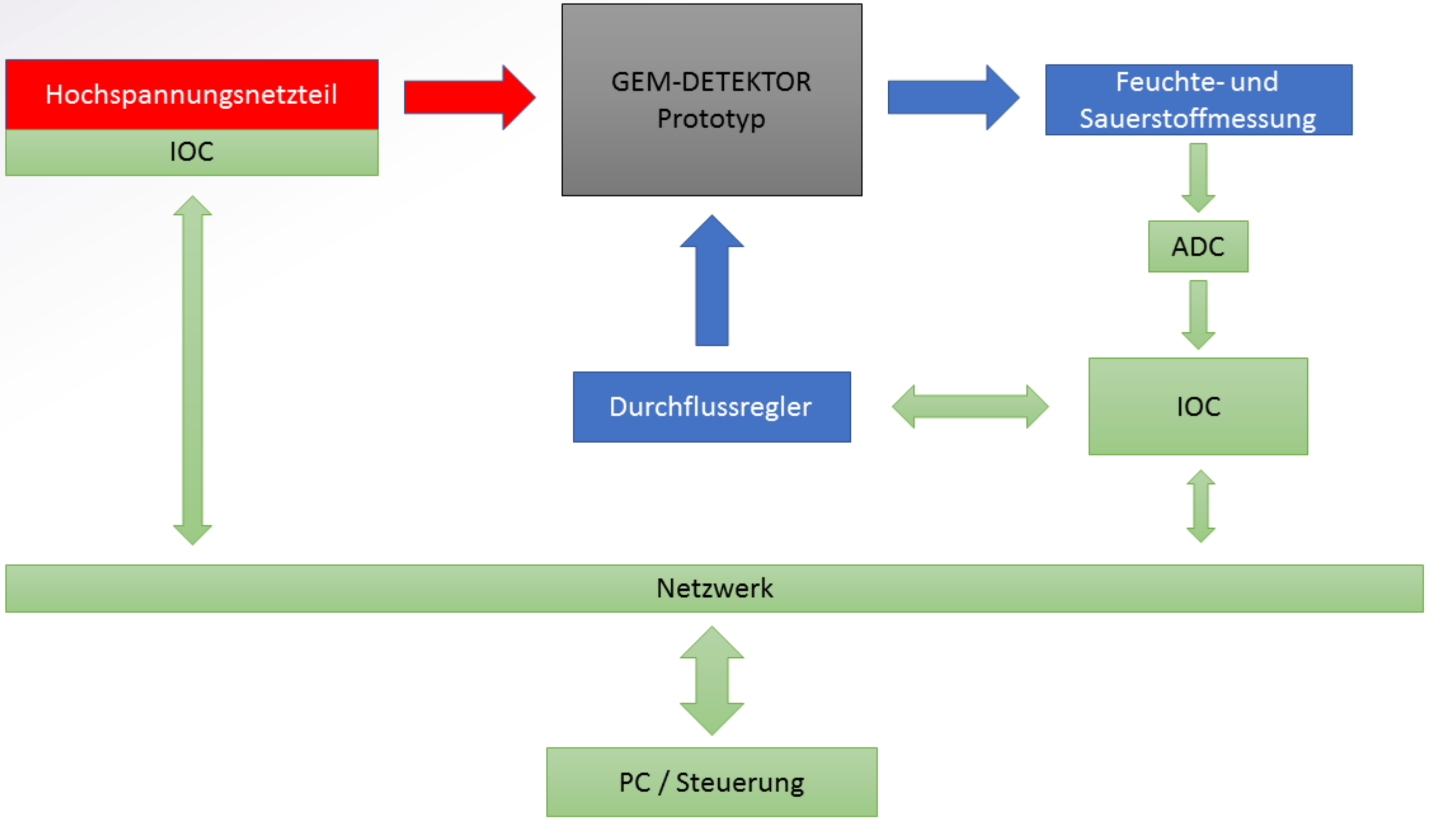
Basic device control



Basic device control



Test setup GEM-Detector

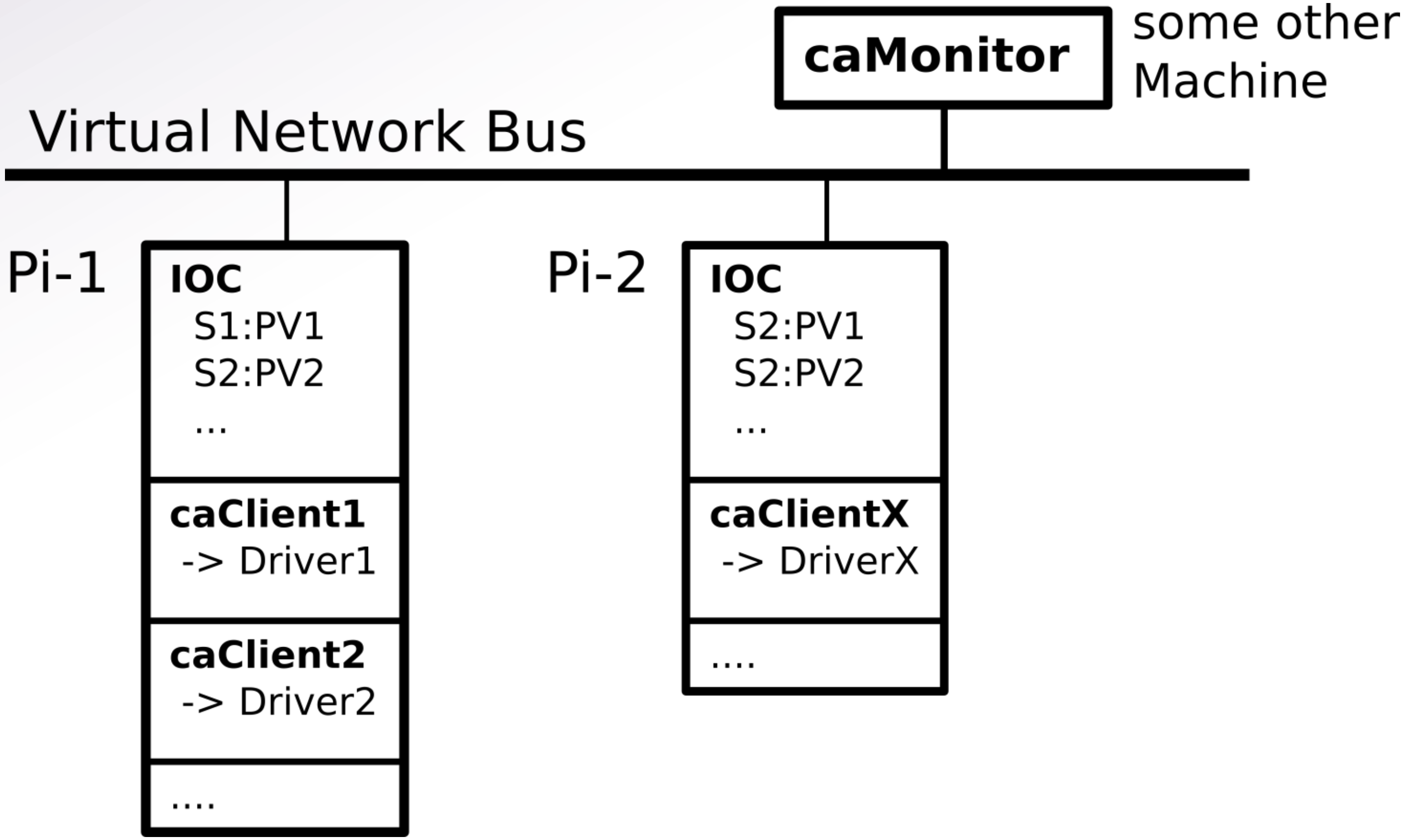


Status quo

IOC Configuration

- Basic supply of Process Variables understood
- Basic knowlegde of database structure
- Communication with IOC understood
- Control of devices via caClient (python, c++)

Structure


caMonitor
some other Machine
Virtual Network Bus
Pi-1
IOC
S1:PV1
S2:PV2
...
caClient1
-> Driver1
caClient2
-> Driver2
....
Pi-2
IOC
S2:PV1
S2:PV2
...
caClientX
-> DriverX
....

MXcaClient

- Self made caClient in c++
  - Interface between PV‘s – drivers
- Client loads the device driver
- Clients fill / read the PV‘s in fixed time intervalls
  - Polling
  - Controlling the device
- Configuration via XML (MXWare)

Strategy / Outlook

What next?

- Learn more about EPICS
- Control devices directly with IOC
    - ➢Streamdevice
- Improve configuration
- Build a GUI
- …

Software

- Write drivers
- Expand/Improve caClient
- Build GUI
- Build control monitor application
    - Safety aspect
    - Logicial operations

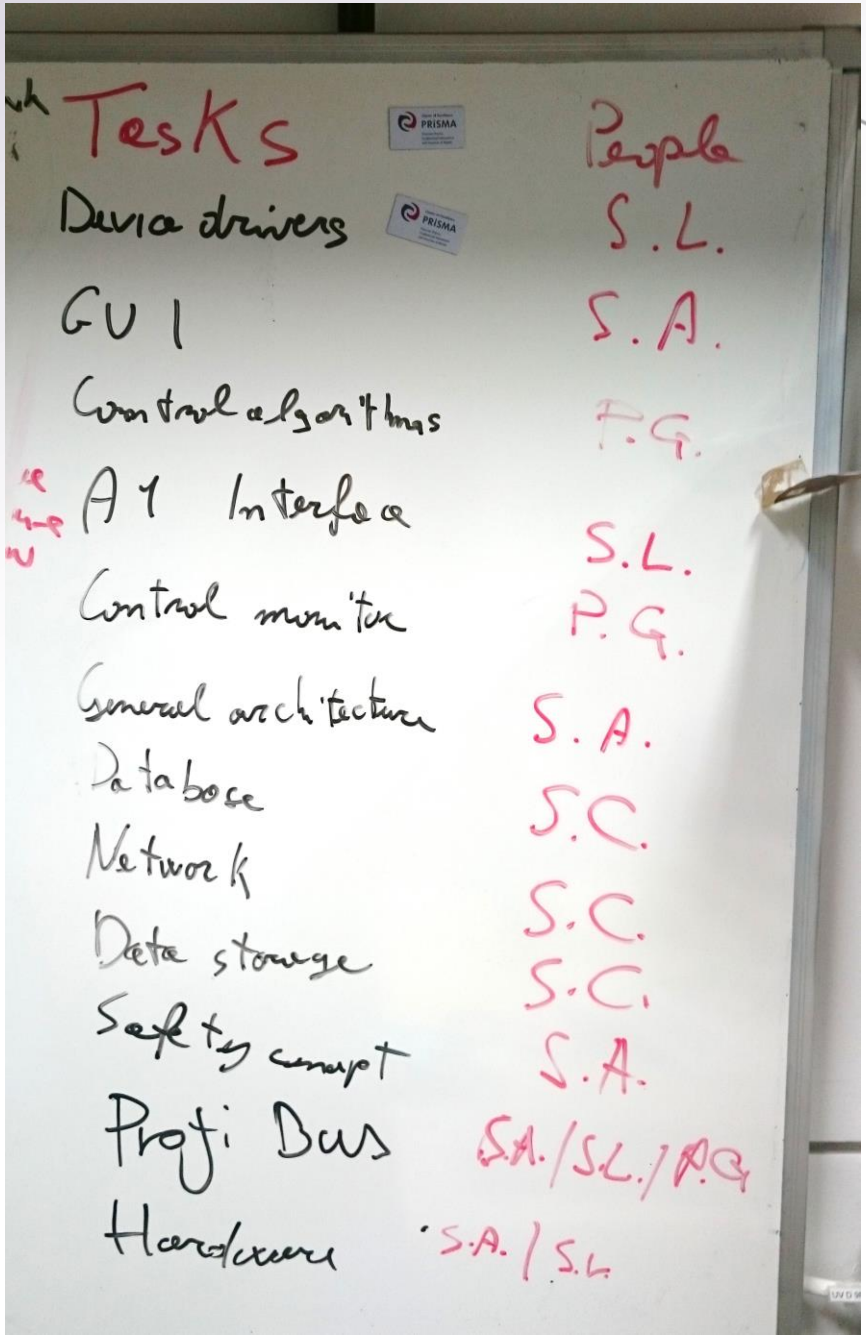
Specific work for beam time

- A1 Interface
- Data storage
- Profi Bus – pumping station
- Other Hardware

Collaboration work

- List important parameters
- Monitoring via Internet
- Discuss driver configuration individually
- Safety discussion
- Control monitor arrangement

Sharing Working



Tasks	People
Device drivers	S.L.
GUI	S.A.
Control algorithms	P.G.
A1 Interface	S.L.
Control monitor	P.G.
General architecture	S.A.
Database	S.C.
Network	S.C.
Data storage	S.C.
Safety concept	S.A.
Profi Bus	S.A./S.L./P.G.
Hardware	S.A./S.L.

SFB 1044

PRiSMA

THANK YOU FOR YOUR ATTENTION!

http://magix.kph.uni-mainz.de

Massachusetts Institute of Technology

University of Ljubljana

Westfälische Wilhelms-Universität Münster

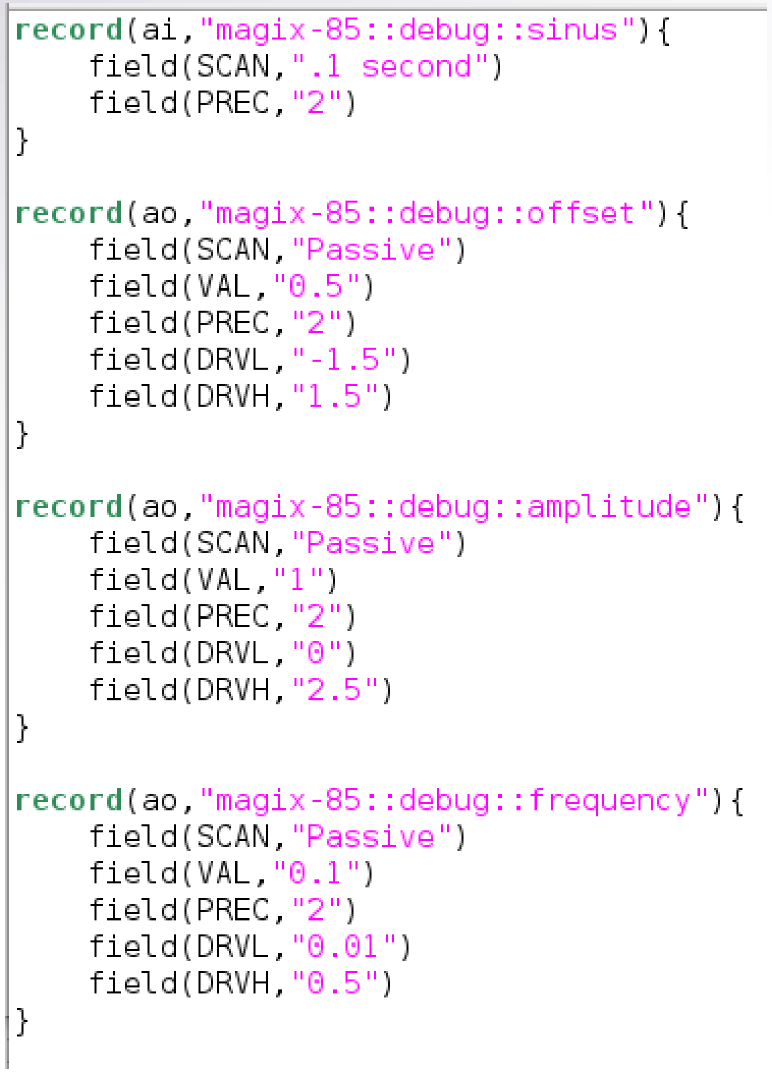
Johannes Gutenberg Universität Mainz

JG|U

BACKUP

Example configuration

IOC Database



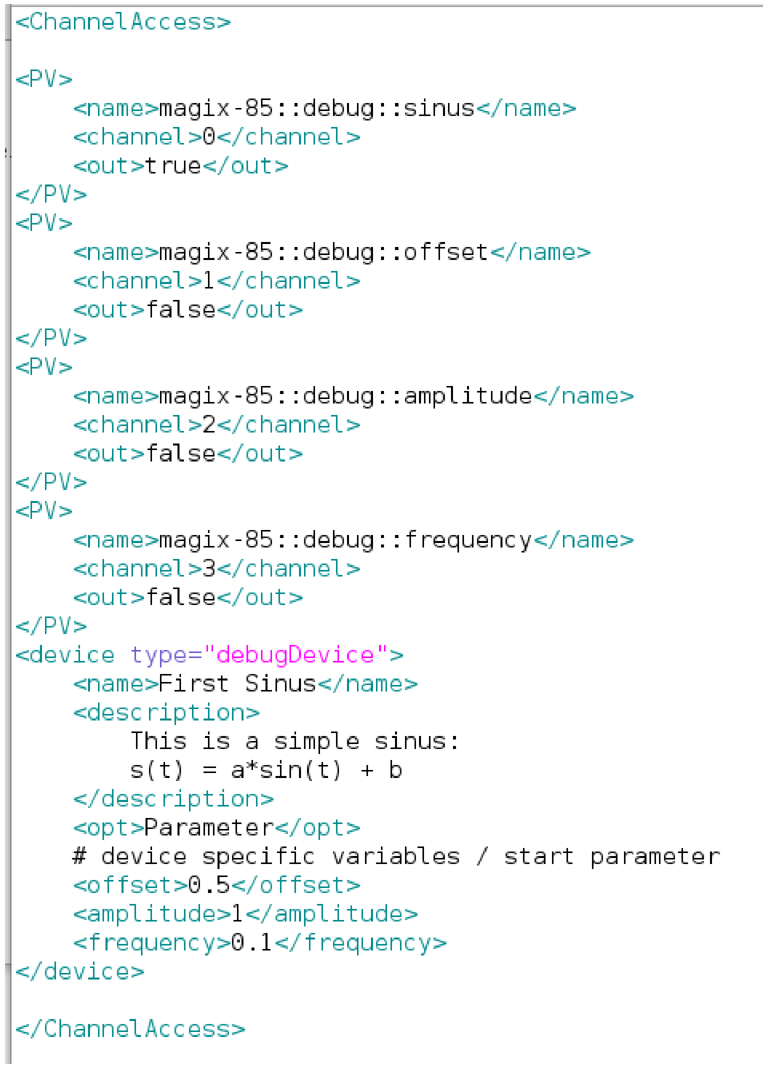
```
record(ai,"magix-85::debug::sinus"){
    field(SCAN,".1 second")
    field(PREC,"2")
}

record(ao,"magix-85::debug::offset"){
    field(SCAN,"Passive")
    field(VAL,"0.5")
    field(PREC,"2")
    field(DRVL,"-1.5")
    field(DRVH,"1.5")
}

record(ao,"magix-85::debug::amplitude"){
    field(SCAN,"Passive")
    field(VAL,"1")
    field(PREC,"2")
    field(DRVL,"0")
    field(DRVH,"2.5")
}

record(ao,"magix-85::debug::frequency"){
    field(SCAN,"Passive")
    field(VAL,"0.1")
    field(PREC,"2")
    field(DRVL,"0.01")
    field(DRVH,"0.5")
}
```

MXcaClient configuration file



```
<ChannelAccess>

<PV>
    <name>magix-85::debug::sinus</name>
    <channel>0</channel>
    <out>true</out>
</PV>
<PV>
    <name>magix-85::debug::offset</name>
    <channel>1</channel>
    <out>false</out>
</PV>
<PV>
    <name>magix-85::debug::amplitude</name>
    <channel>2</channel>
    <out>false</out>
</PV>
<PV>
    <name>magix-85::debug::frequency</name>
    <channel>3</channel>
    <out>false</out>
</PV>
<device type="debugDevice">
    <name>First Sinus</name>
    <description>
        This is a simple sinus:
        s(t) = a*sin(t) + b
    </description>
    <opt>Parameter</opt>
    # device specific variables / start parameter
    <offset>0.5</offset>
    <amplitude>1</amplitude>
    <frequency>0.1</frequency>
</device>

</ChannelAccess>
```