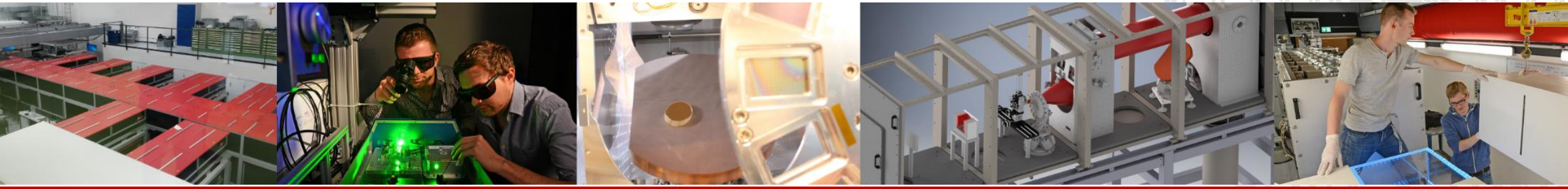
# Tango based data archiving for the CALA high power laser facility

Leonard Doyle(*), Max Gilljohann, Jens Hartmann, Johannes Gebhard, Felix Balling, Nils Weisse, Andreas Münzer, Andreas Döpp

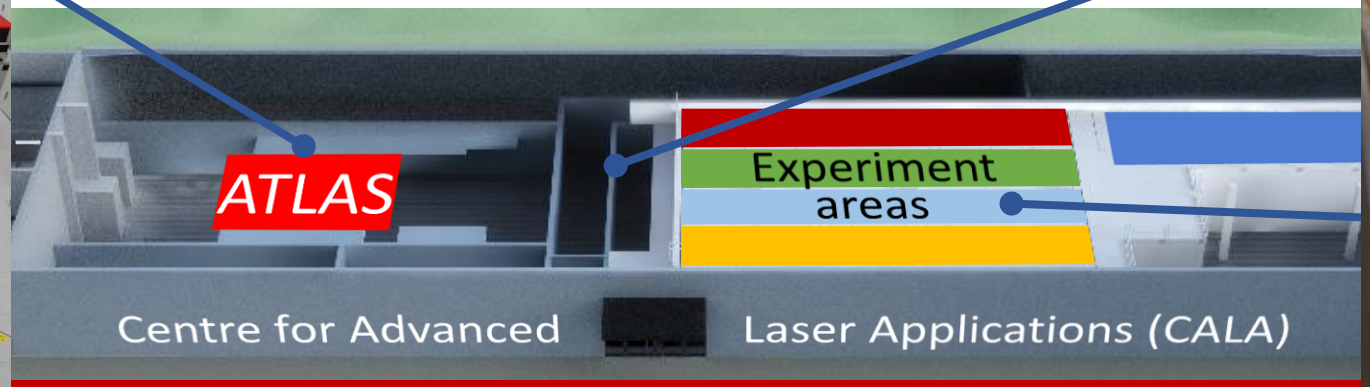
Prof. Dr. Jörg Schreiber, Prof. Dr. Stefan Karsch

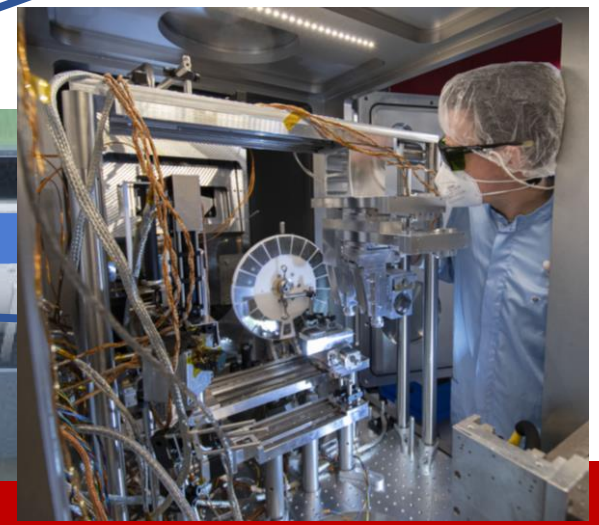
(*) leonard.doyle@physik.uni-muenchen.de



# CALA and the ATLAS-3000 laser

- Centre for Advanced Laser Applications in Garching near Munich, Germany
- ATLAS-3000
  - Petawatt class laser, 1Hz repetition rate
  - operational since 2017, PW in 2022?
- Experimental areas for
  - Laser ion acceleration (protons, carbon, gold)
  - Laser electron acceleration (LWFA, hybrid schemes)
  - Xray source (Thomson backscattering, …)
- Run by university, no user facility
- „student run“ = limited number of technicians, …



# Tango system in CALA



- Hardware control
  - Motion (Phytron Phymotion, PI Hexapods, Owis, Thorlabs, …)
  - Dazzler, Arduino, Picoscope …more to come (cameras,…)
  - Amplitude laser controllers (Masterpulse, Genpulse)
  - (Thales amplifier already based on Tango)
- Automatic Shotlog/ data archiving
  - Save important data for each shot
- Ongoing addition
  - Shotlog and monitoring of laser parameters
- Longterm addition
  - Automated optimization of parameters

# Praise to PyTango



- Python generally used by many students/researchers in other contexts
- No compilation
  - Less platform dependent
  - easier to hotfix
- Less error prone (memory leaks, pointers, …)
- Deployment == git clone, git pull
- PyTango high level API very nice
- Interop with evaluation scripts, existing GUIs, …


# Archiving System



## Requirements

- **Goal:** Automated shotlog, make writing of labbook easier, quicker, less error prone
- Trigger with laser shot (single shot up to 1Hz)
- Master:
  - save shotnumber, timestamp, user parameters (target type, manual comments, …)
- Motors, vacuum gauge, …:
  - save position/pressure/… (+state)
- Cameras, picoscope, …:
  - set filename, arm trigger, then save on hardware trigger
- After beam time:
  - easily readable by inexperienced user/student

## Implementation

- File based approach
  - Folder for each beam time, subfolder for each device
  - Text files (shot log, motors, energy meter, …)
  - Data files (camera, oscilloscope, …)
- Why not HDB++??
- KISS
- Data extraction easy/ not requiring expert knowledge/ custom tools
- Heterogeneous equipment
  - „hybrid devices“ not Tango-aware but using Tango shot number
- However: Long term performance limit??

# Archiving system: Implementation details



**ExperimentControlServer**

*Tango attributes:*
**ArchShotNumber** (event)
ArchShotTime
ArchPath
ArchSubscribers

*Tango commands:*
**SingleShot**():
  #increase ShotNo
  #grp_subscribers.
    cmd_asynch(`Archive`)
  #push_event(`ArchShotNumber`)
  #save own shotlog.txt
  #fire laser
Subscribe()
Unsubscribe()

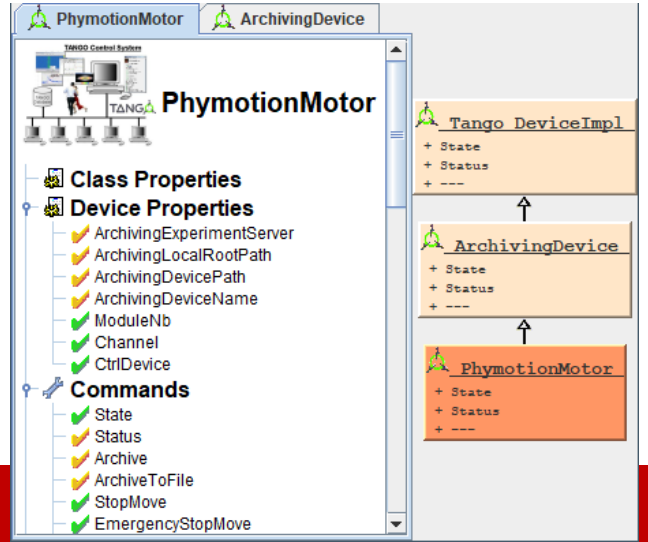
**PhymotionMotor**
**:ArchivingDevice**

*private* **ShotEvent**():
  #read position + state from HW
  #save .txt

*Tango attributes (inherited):*
ArchShotNumer
ArchPath, …

*Tango commands (inherited):*
Archive([ShotNo, ShotTime, …]):
  #inherited
  #update self attributes
  #self.ShotEvent()

device specific attributes, commands, …




- Master: ExperimentControlServer
  - keep track of ShotNumber
  - trigger archiving
  - trigger laser shot
- Archiving devices:
  - inherit from `ArchivingDevice`
  - simply override private (non-Tango) `ShotEvent()`
  - e.g. `self.archiver.savetxt(self.__position, self.state())`
- Hybrid devices:
  - DeviceProxy, not DeviceServer
  - subscribes to `data_ready_event` on ShotNumber
  - fetches others (timestamp, path, …) by attributes

# Archiving system: File structure

- Global folder (network share)
  - /project/cala_lion/Experiments
  - *similar for other caves, laser*
- Subfolder/filename for each device
  - `ExperimentPath/ArchDevicePath/ArchDeviceName`
  - `ExperimentPath` = 20210831
  - ArchDevicePath = property
    - `automatic` → `hexapod` for `lion/hexapod/target`
  - ArchDeviceName = property
    - `automatic` → `target.txt` for `lion/hexapod/target`

```
/project/cala_lion/Experiments/20210831
├── expcontrol
│   └── main.txt
├── hexapod
│   ├── oap.txt
│   └── target.txt
├── IncomingEnergymeter
│   └── PM100USB.txt
├── motor
│   ├── air-RCF-y.txt
│   ├── air-szinti-x.txt
│   ├── air-vertical.txt
│   ├── blocksh1.txt
│   ├── buncher-x.txt
│   ├── buncher-y.txt
│   ├── buncher-z.txt
│   ├── etalon-focus.txt
│   ├── lin-aperture.txt
│   ├── lin-dipole-slit.txt
│   ├── lin-dipole.txt
│   ├── lin-mic-X.txt
│   ├── lin-mic-Z.txt
│   └── lin-radeye-ch4.txt
```

- Future project: Python module to pull together all data for a specific shot

20210831/vacuum/tpg.txt

```
//lion/vacuum/tpg
ShotTime          Shot  Run  Flags   WriteTime          SensorState  Pressure
20210831_160900749  1    0    N    20210831_160900945   True         3.13e-06
20210831_212506655  3    0         20210831_212507777   True         2.76e-06
20210831_215736752  4    0         20210831_215737808   True         2.71e-06
20210831_215821948  5    0         20210831_215822965   True         2.71e-06
20210831_215837011  6    0         20210831_215838157   True         2.71e-06
20210831_220016177  7    0         20210831_220017319   True         2.71e-06
20210831_220025913  8    0         20210831_220027060   True         2.71e-06
20210831_220042187  9    0         20210831_220043218   True         2.71e-06
20210831_220051880  10   0         20210831_220052926   True         2.71e-06
```

# Conclusion



- Introduced ATLAS-3000 in the Centre for Advanced Laser Applications near Munich
- Details on archiving system
  - file based, on shot (max 1Hz)
  - still under development (no feedback if saving succeeded)
  - future performance problems?

- Interested in community experience
  - with on-shot/triggered archiving
  - with „inexperienced users“/ keeping the learning curve flat/ maintaining a system at limited resources

https://www.cala-laser.de/ https://www.pulse.physik.uni-muenchen.de/