Detector Control Systems

Paris Moschovakos

paris@moschovakos.com

ISOTDAQ school 2024

Start/Stop Detectors

Paris Moschovakos

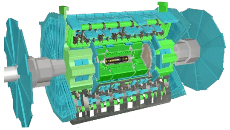
paris@moschovakos.com

ISOTDAQ school 2024

<prelude>

Paris Unpacked

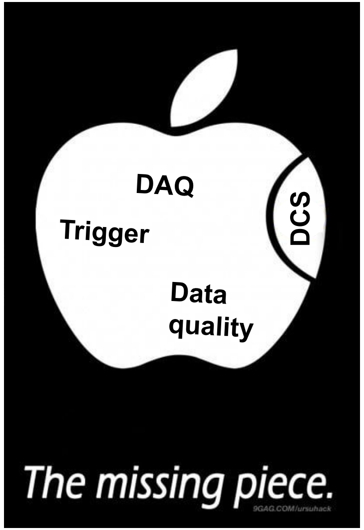
- ~10 years of DCS development for ATLAS detector: the system responsible for the vital task of monitoring and controlling it
- Experience from developing OPC UA servers for CERN
  - OPC UA servers are like librarians for detector data, gathering it and sharing it securely with systems that need it
- Member of the C++ standardization committee



- I would like to acknowledge **Stefan Schlenker**, Piotr Nikiel, Ben Farnham and many others… as many of the ideas presented in this material are stemming out of their work

In the ISOTDAQ context

- To put another piece of the global picture in your knowledge arsenal
- Give you insights on the integration of TDAQ systems with the rest of the pieces that comprise a real large experiment in HEP
- I would like to warmly thank Markus Joos and the colleagues in the ISOTDAQ organizing committee for giving us this opportunity



Control System

- **Definition:** A system that manages, commands, directs, or regulates the behavior of other devices or systems
- **Components:** Typically includes sensors (to measure values), processors (to make decisions), and actuators (to effect changes)
- **Purpose:** To keep the controlled variables within a desired range, ensuring stability, accuracy, and efficiency
- **Examples in Everyday Life:** Thermostats in homes, cruise control in cars, and automatic doors in buildings

Detector Control System

- **Relevance to Detectors:** In Detector Control Systems, it ensures that the detectors are working properly, gathers the data they collect, and makes decisions based on that data.
- **Applications in Science and Industry:** Used in large-scale scientific experiments (like particle physics), manufacturing processes, and safety systems

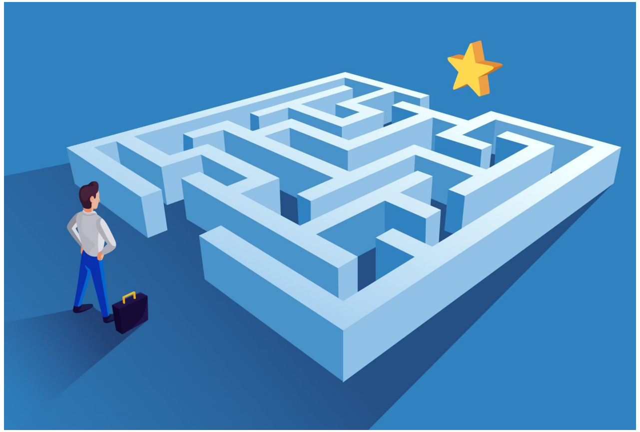
Motivation

- **Handle Complexity**: Easily manage millions of channels and sensors
- **Efficiency**: Quickly set up and operate the entire system
- **Monitoring and Troubleshooting**: Keep an eye on important variables with trend plots and instantly know if something goes wrong through an alarm screen
- **Data Logging and Analysis**: Records the history of all readouts, allowing for later analysis to understand patterns or detect anomalies
- **Customize**: Fine-tune settings for precision control of each element
- **Ensure Safety**: Keep operations within safe limits with automated monitoring

</prelude>

Navigating the DCS maze

- What are the principles of DCS, what it responsible for?
- Basic DCS concepts
- In an experiment - interface with DAQ
- OPC UA as the DCS middleware
- How to build an OPC UA server with quasar



Detector Control System in Large Experiments

“

Simplicity is the
ultimate sophistication.

Leonardo Da Vinci

Expectations & Principles

- Detector Expectations
  - Continuous control and monitoring of field elements and processes
  - Deals with the dynamic aspects of the system
  - Management of low and high level data
- Architectural principles to achieve them
  - Reliability
  - Availability
  - Operability
  - Flexibility / Extensibility / Maintainability
  - Security

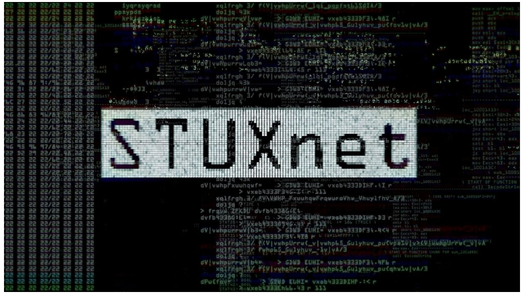
What is SCADA

- **Definition**: SCADA stands for Supervisory Control And Data Acquisition
- **Function**: It's like the brain of an industrial system, monitoring and controlling processes, gathering data from sensors, and allowing operators to manage everything from a computer.
- **Components**: SCADA systems typically include sensors and control devices (such as PLCs), communication networks, a cluster of computers, and terminals with software to control and monitor the processes.
- **Applications**: Used in various industries like manufacturing, power generation, airports, and nuclear plants, for managing large-scale processes efficiently.

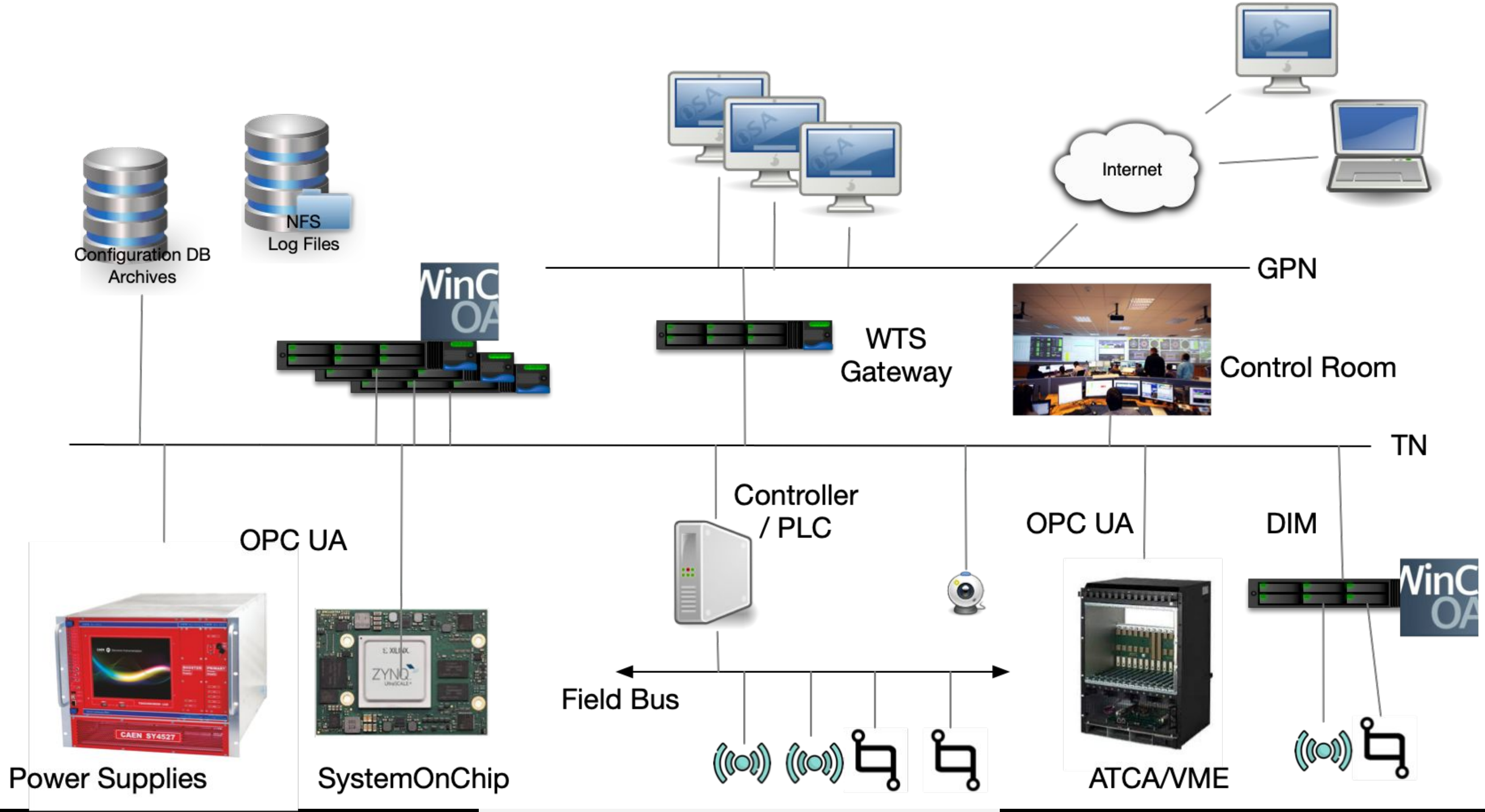
Train Operation Command Room of Deqing Hefei-Hangzhou Railway Depot

What is WinCC OA

- **Definition**: WinCC OA stands for WinCC Open Architecture. It's a part of Siemens WinCC (Windows Control Center) suite originally built by ETM
- **Flexibility**: Unlike traditional SCADA, WinCC OA is highly customizable and scalable, making it great for complex and large-scale systems
- **Features**: It offers real-time process visualization, data archiving, alarm management, and supports various communication protocols
- **LHC Application**: At the Large Hadron Collider experiments, WinCC OA is chosen for its flexibility and scalability to manage the enormous data and control needs of the particle detectors.
  a. It used in all 4 major experiments (ATLAS, CMS, ALICE, LHCb) and various smaller
  b. It is used by the LHC accelerator
  c. CERN's electrical Network
  d. CERN's cooling and ventilation

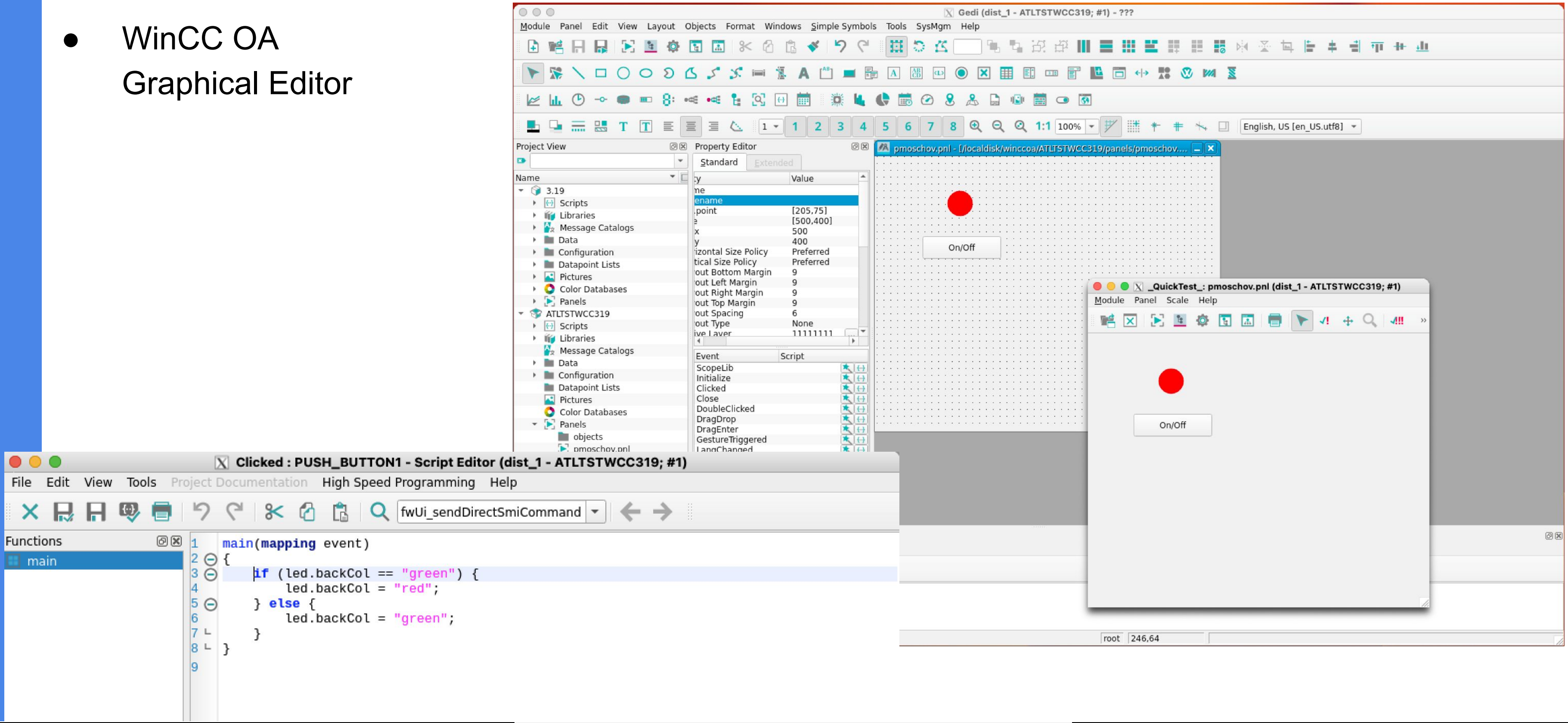


How WinCC OA is related to the Iran's nuclear plants?


Internet
NFS
Log Files
Configuration DB
Archives
GPN
WTS
Gateway
Control Room
TN
Controller
/ PLC
OPC UA
OPC UA
DIM
Field Bus
Power Supplies
SystemOnChip
ATCA/VME

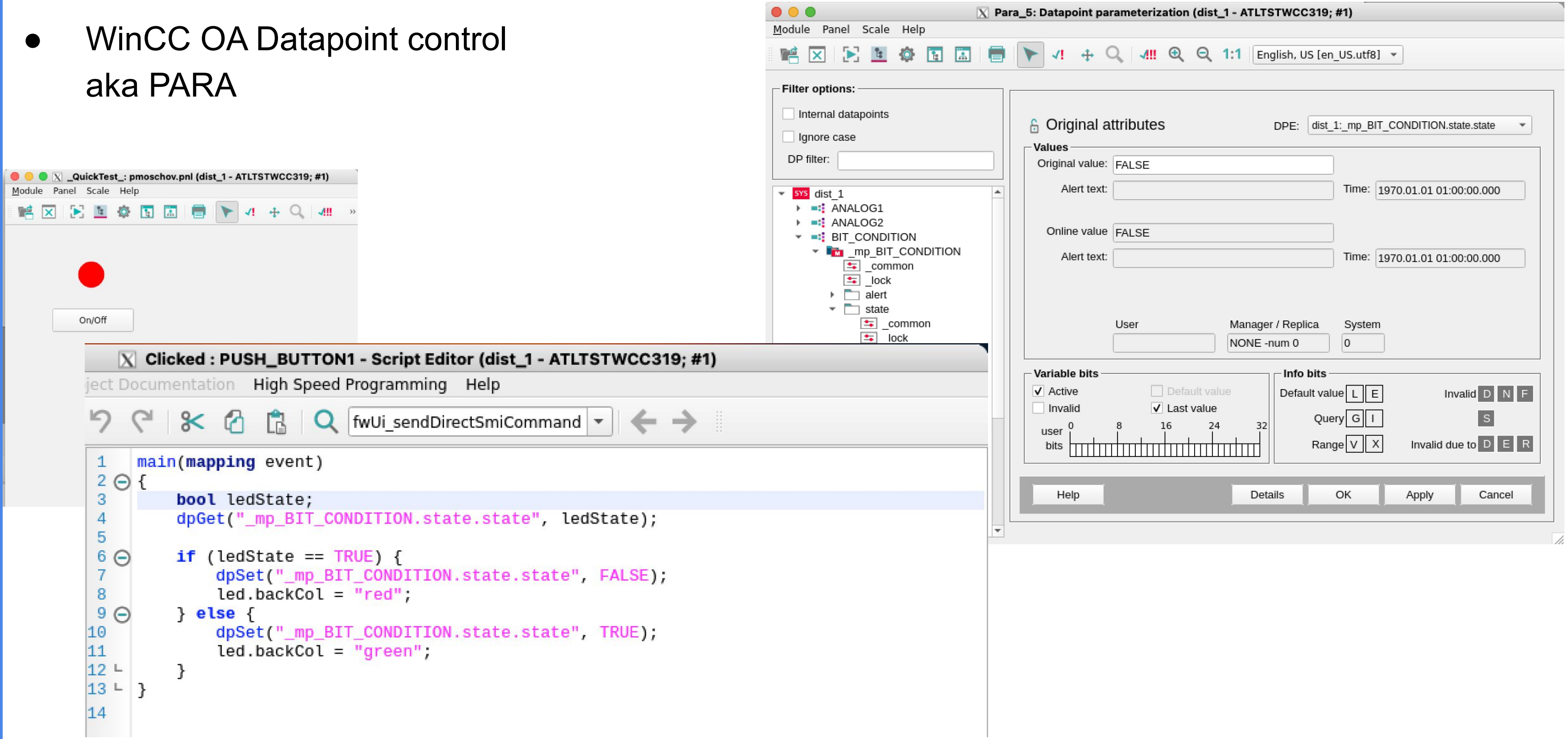
Hello DCS World Example - WinCC OA

- WinCC OA Graphical Editor



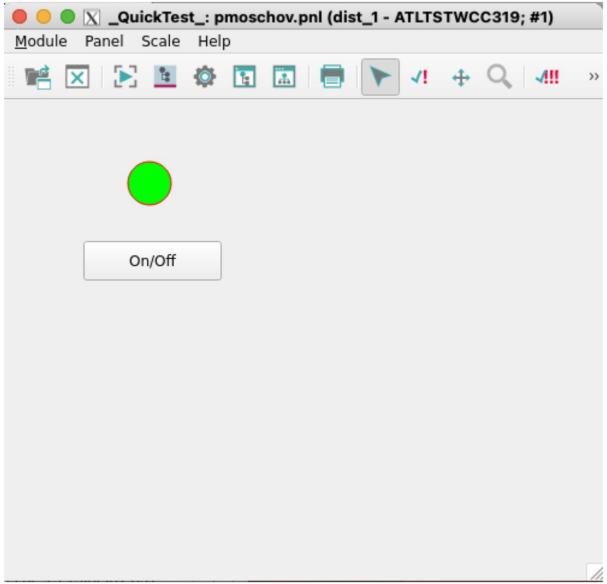
Lets connect the LED to a Datapoint

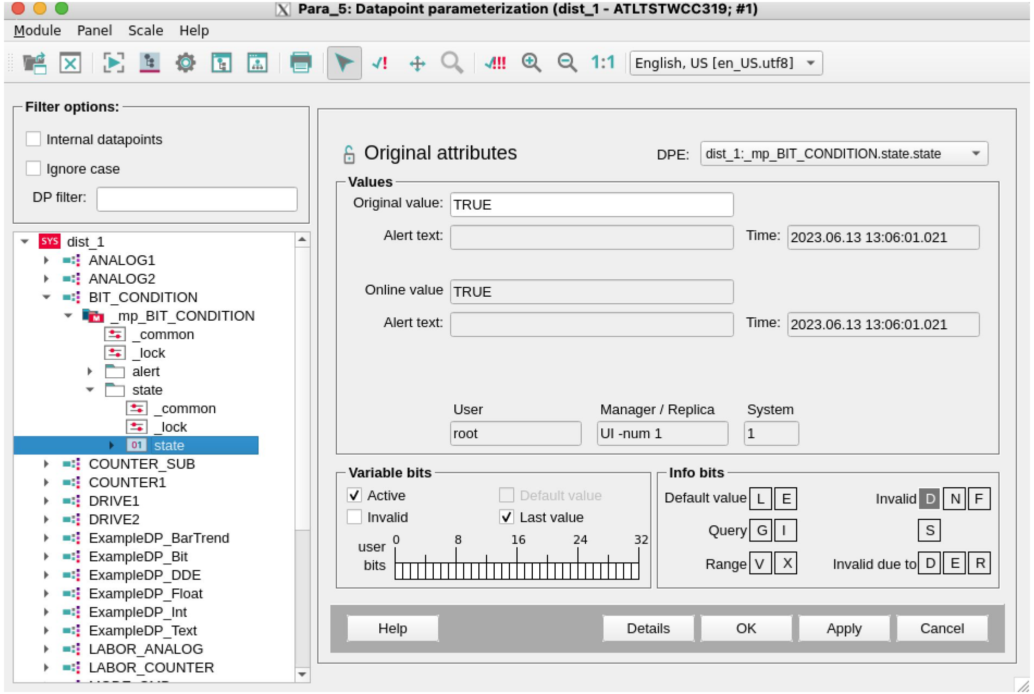
- WinCC OA Datapoint control aka PARA



Lets connect the LED to a Datapoint

- WinCC OA Datapoint control aka PARA



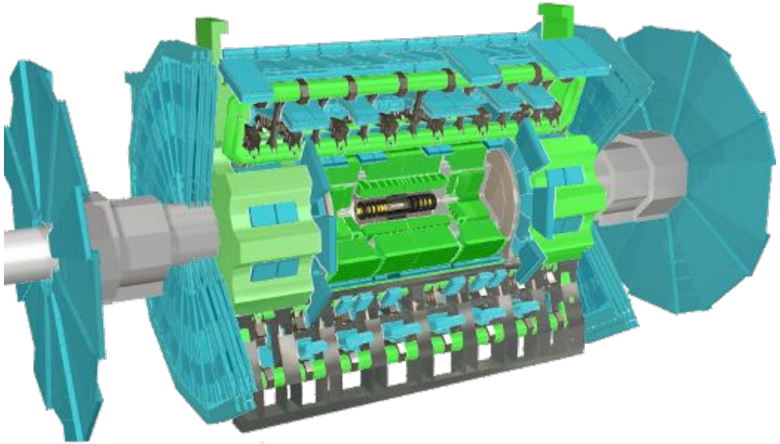


DCS Hierarchies - CERN's FSM

Nice but how that scales up to millions of parameters?

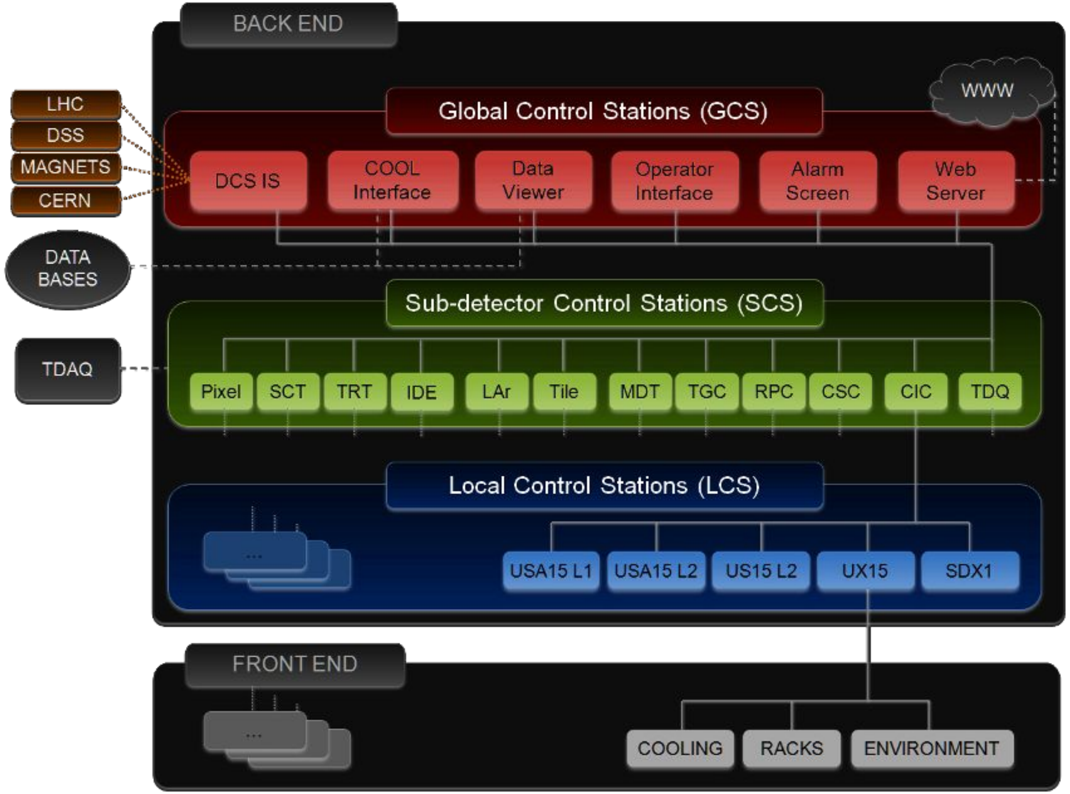
- **Definition**: FSM stands for Finite State Machine. CERN's FSM is a framework that organizes the control systems into a hierarchical structure.
- **Hierarchy**: Imagine a tree where the trunk is the main control system, branches are sub-systems, and leaves are devices like sensors and actuators.
- **States & Transitions**: Each element (branch or leaf) can be in a state (like 'ON', 'OFF', 'READY') and can change states based on conditions or commands.
- **Role in SCADA**: FSM helps manage complex systems at CERN by breaking them down into manageable parts, providing a structured way to monitor and control each part within the SCADA environment.
- **LHC Application**: In the LHC, FSM is crucial for handling the complexity of the detectors and accelerators, enabling efficient operation and monitoring.

DCS in ATLAS detector

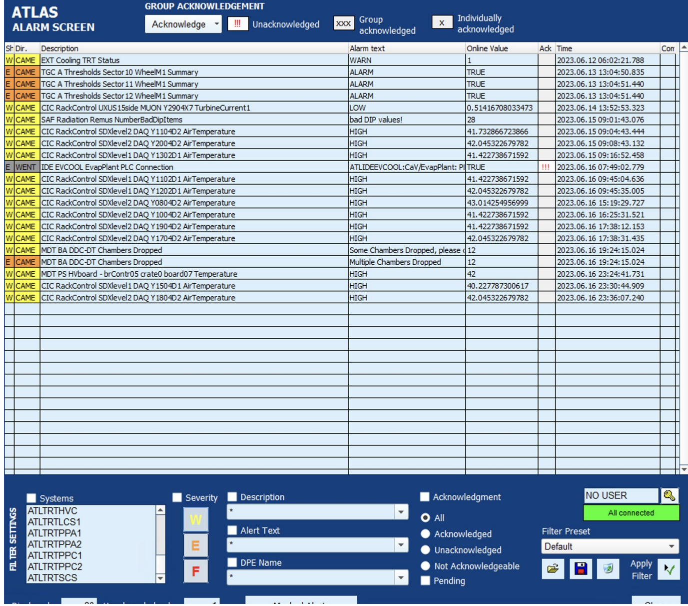


In Numbers

- In ATLAS alone the DCS system monitors and controls over 10^7 parameters in more that 140 local control stations
- Archiving of selected parameters is done to an Oracle database
- It is interfaced with DAQ and Run Control
- It also links to external control system (LHC, Infrastructure)
- It is remotely accessible: directly or via web applications



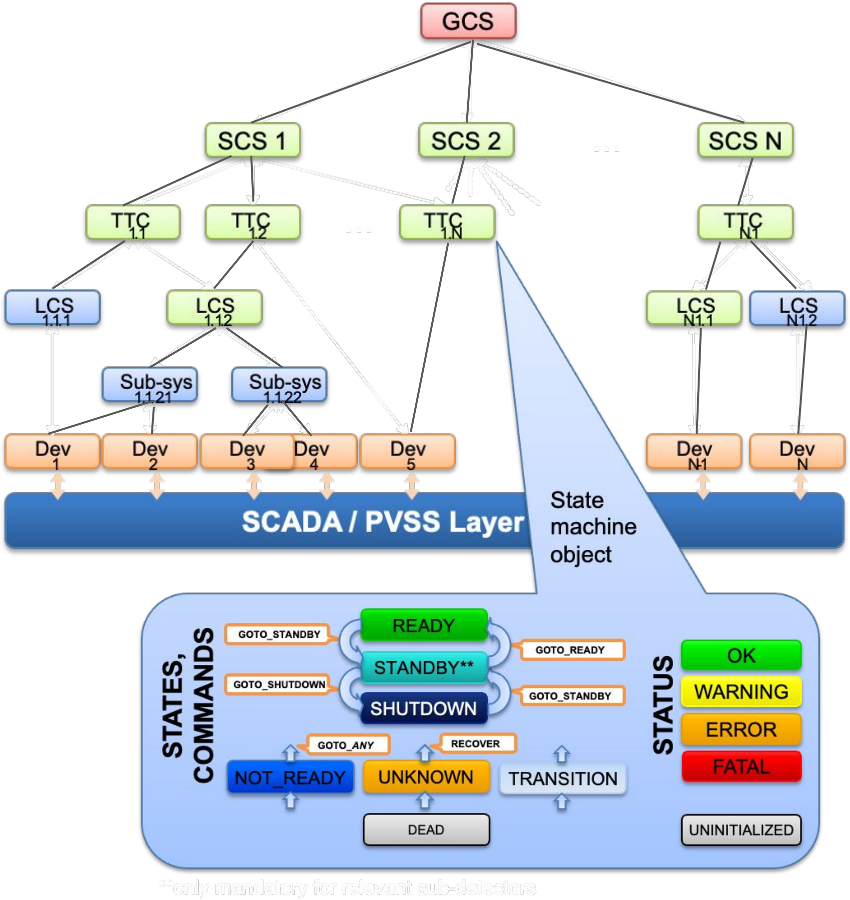
ATLAS Alarms



- Active and non-acknowledged alarms
  - Alarm = parameter out of good range
  - Options for each alarm
  - Mask alarm on UI level until the alarm condition goes
- Facilitates alarm reporting to experts via email
- Display trend plot of value
- Alarm help on web page
- Acknowledgement:
  - Only for some alarms
  - Operator needs to explicitly interact with them, otherwise they won’t disappear from screen when resolved
  - Unacknowledged, but resolved alarms are shown as „WENT“ with grey color
- Summary alert:
  - Single alarm entry hiding several alerts of same type (accessible via “Details”)
- Filters:
  - Different sets of filters exist, e.g. sub-detectors

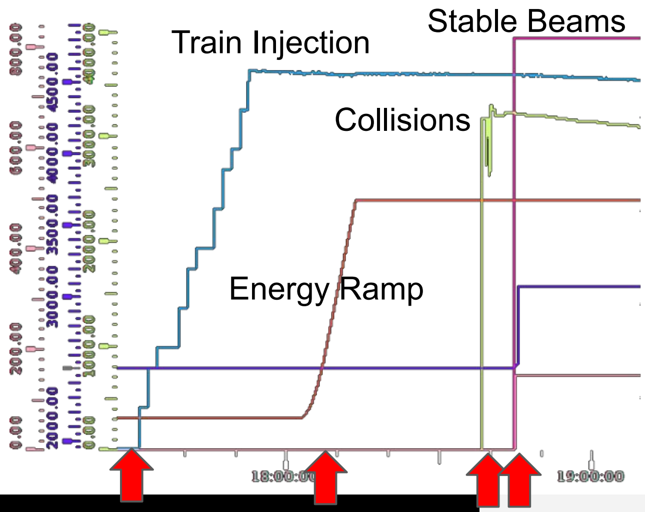
ATLAS FSM

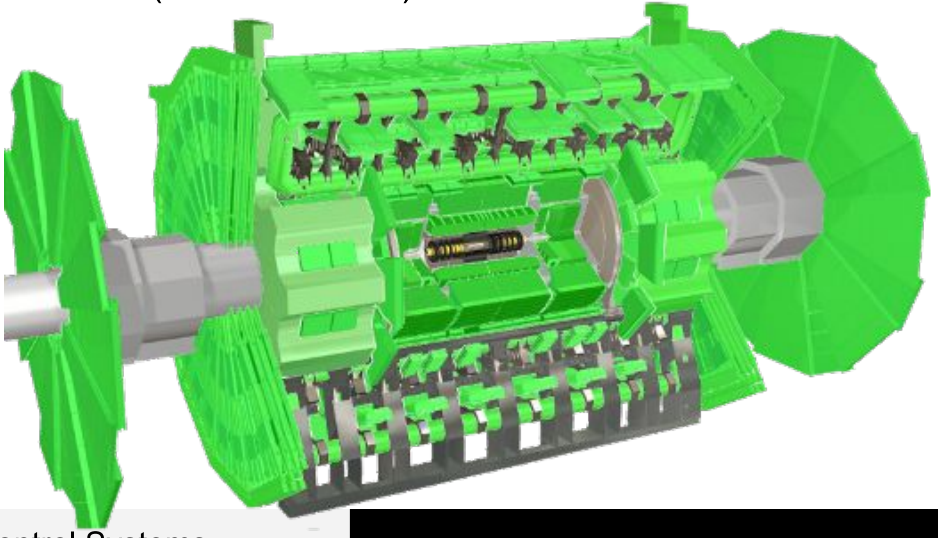
- State machine hierarchy
  - Detector hardware represented by FSM objects, hierarchically structured
- **State & Command propagation**
  - State model for devices (ON,OFF,…) and logical objects (READY, NOT_READY,…)]
  - Propagation upwards (using programmed logic for parent depending on child states)
  - Commands propagated downwards
- **Status**
  - Error handling upwards
- Object operations
  - Commands
  - Enable/disable (low level)
  - Take/release/include/exclude (high level)
  - Access control for all operations
- User Interface
  - Browsing and object operation…



LHC Interaction and interface with DAQ

- Synchronization of DCS with LHC operation and run control
  - Detector safety requires lower voltage levels during unstable beam conditions (beam injection, adjust, dump) =
  - State change automated, synchronization with DAQ run control, should take max. 5 minutes
  - Audible notifications from DCS for important beam related events
- Beam backgrounds and luminosity monitored via DCS (LHC FSM tree)





OPC UA

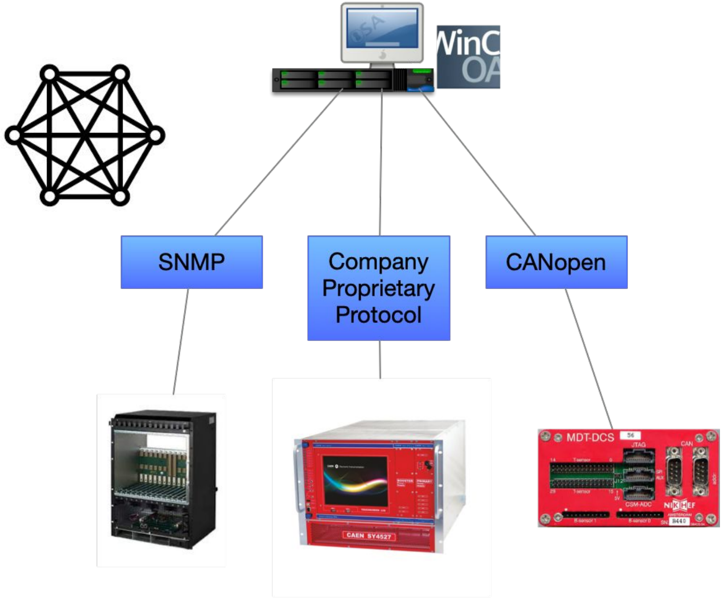
OPC UA®

Don't be satisfied with stories, how things have gone with others. Unfold your own myth.

- Rumi

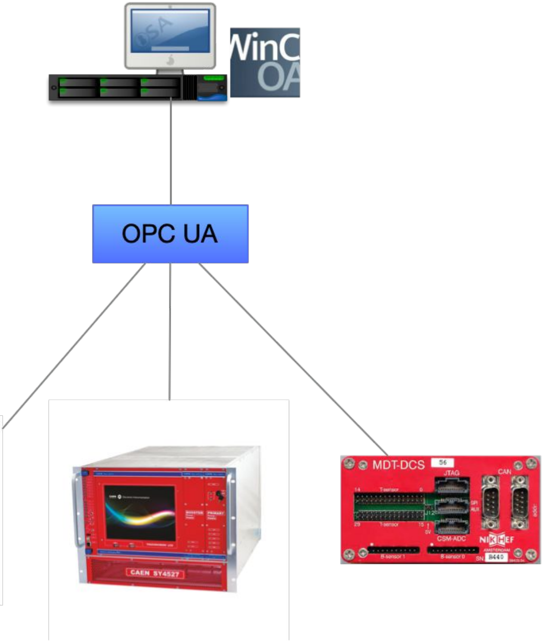
What is OPC UA?

- Open Platform Communication Unified Architecture (OPC UA) is a machine to machine communication protocol for industrial automation developed by the OPC Foundation.



What is OPC UA?

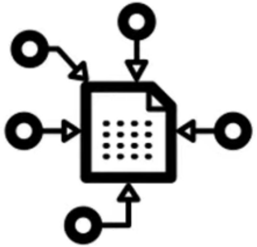
- Open Platform Communication Unified Architecture (OPC UA) is a machine to machine communication protocol for industrial automation developed by the OPC Foundation.



OPC UA - the standard middleware @CERN

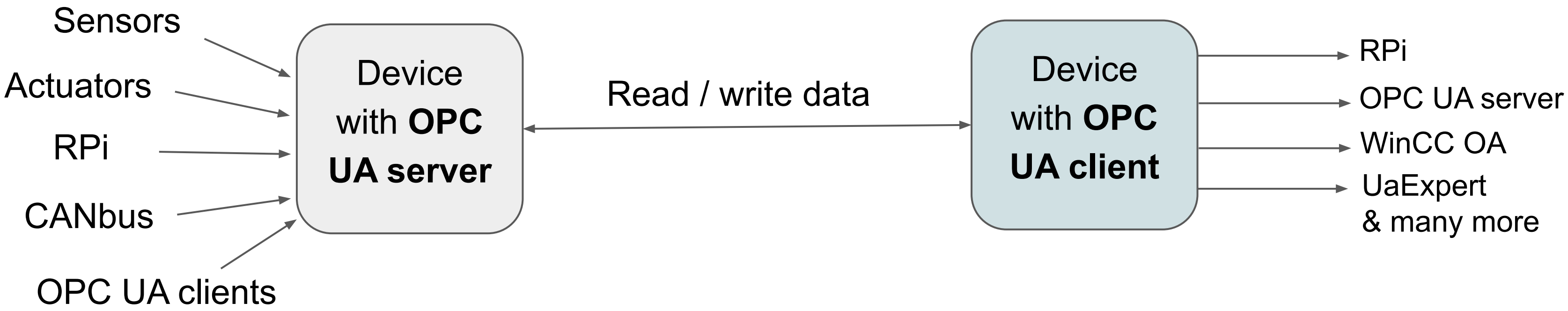
- Why OPC UA?
  - focuses on communicating with industrial equipment and systems for data collection and control
  - Open specification and various implementations available (free or commercial)
  - Cross-platform
  - Service-oriented architecture
  - Integral information model, which is the foundation of the infrastructure necessary for information integration where vendors and organizations can model their complex data into an OPC UA namespace
- Current experience with OPC UA @CERN
  - **Works natively** with the tools used in Detector Control Systems
  - CERN developed a framework for developing OPC UA servers
  - It is the standard **prefered** by the **"big" vendors** (e.g. CAEN, ISEG, Weiner, etc.) for their power supply devices
  - It is used by various custom devices in experiments used widely at CERN (ELMB, SoC)
  - CERN foresees to have **support** and provide **maintenance** on those solutions on the long term

OPC UA Applications & Examples

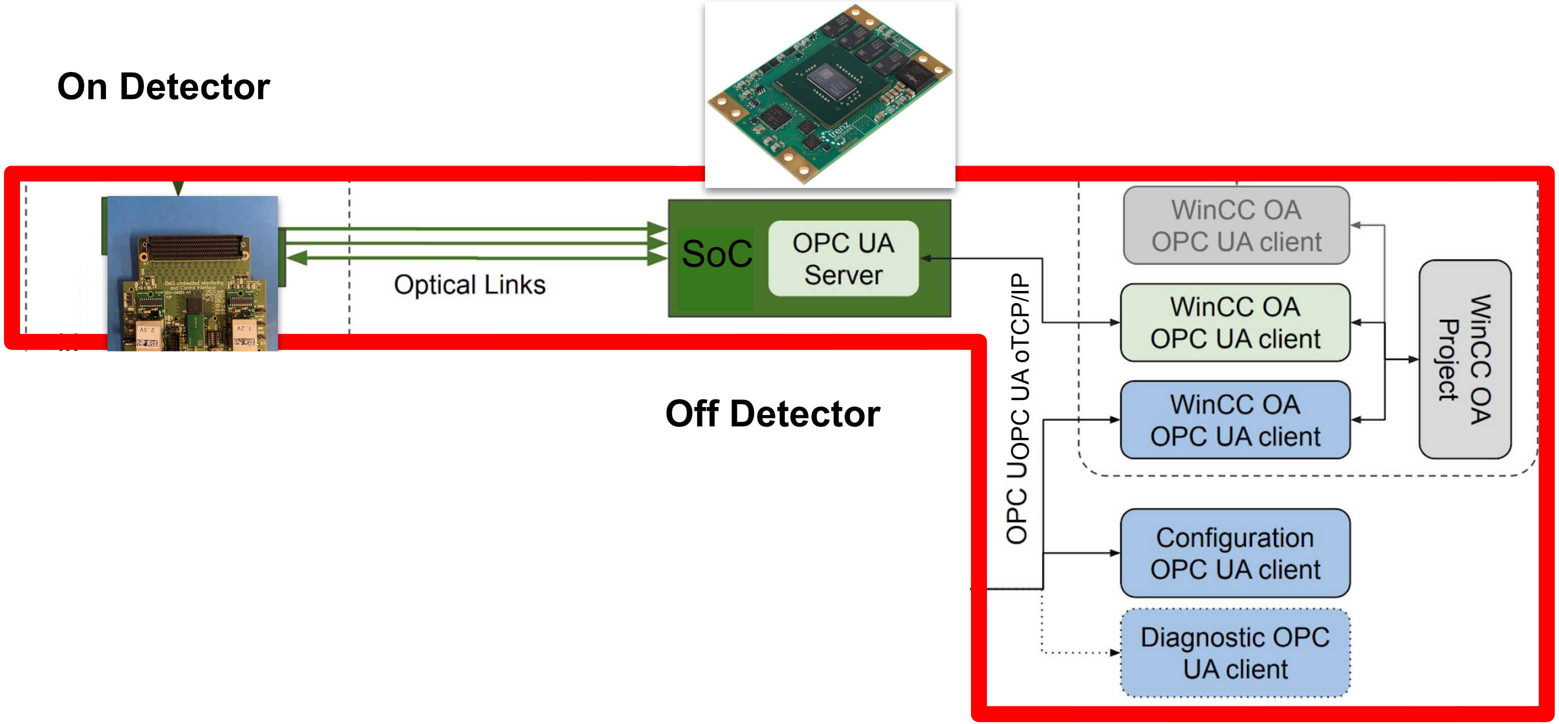


Sharing machine data easily

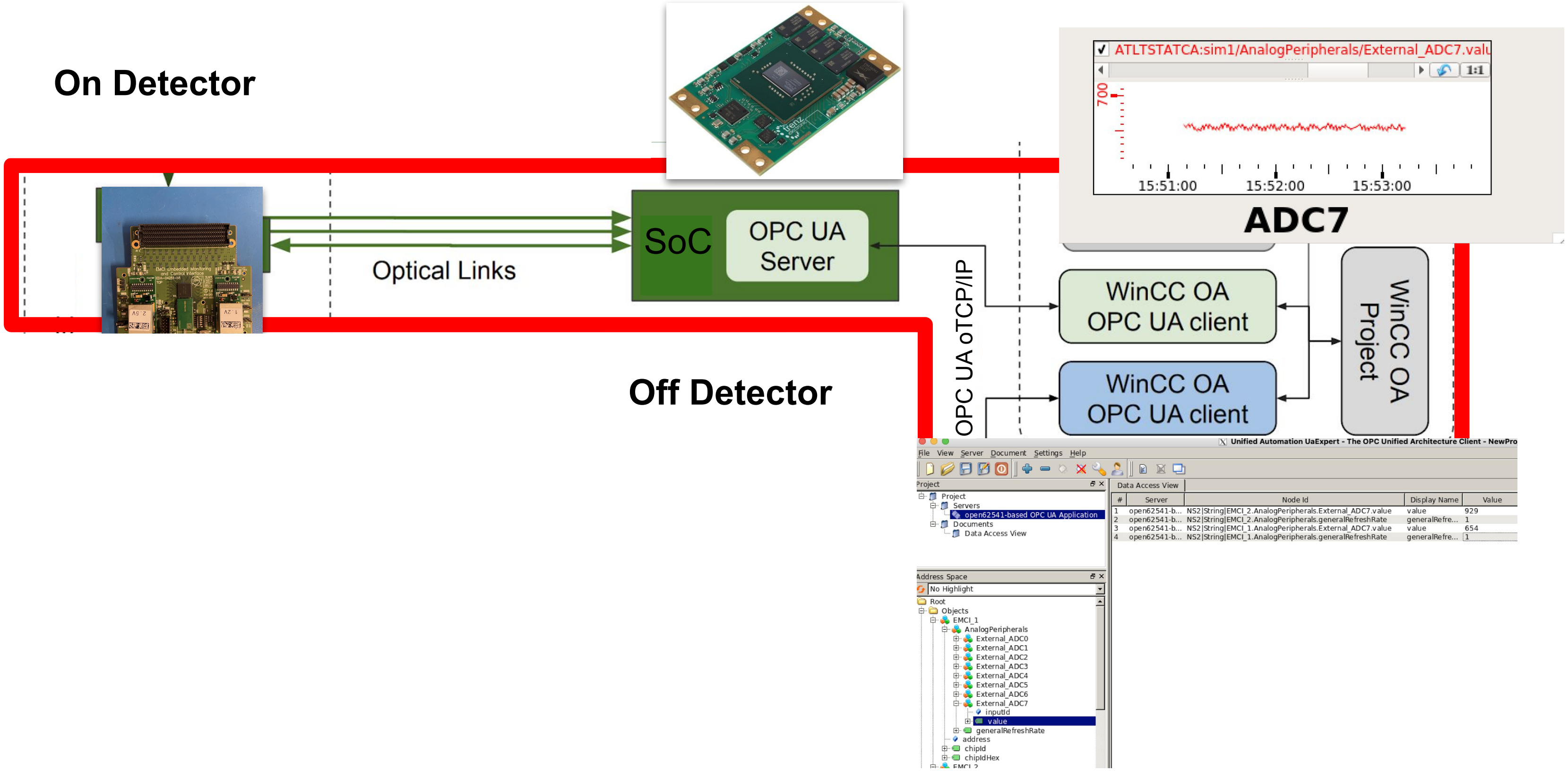
Gathering machine data
in standard structure



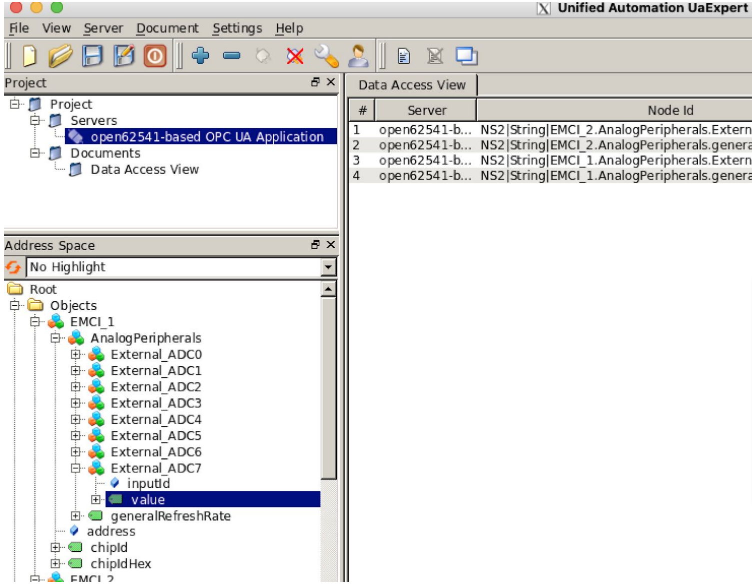
OPC UA server vertical slice



OPC UA server vertical slice

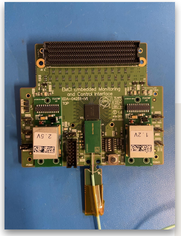


OPC UA clients

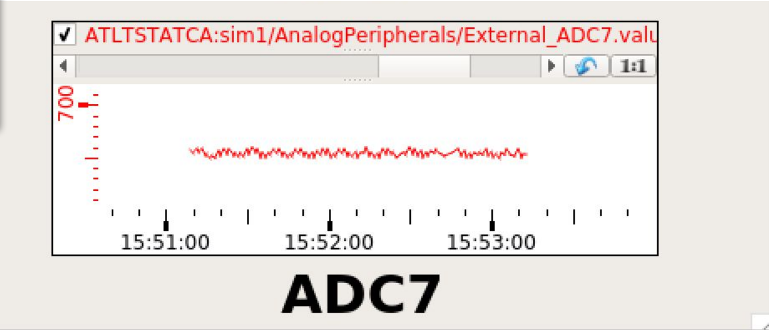


#	Server	Node Id	Display Name	Value
1	open62541-b...	NS2|String|EMCI_2.AnalogPeripherals.External_ADC7.value	value	929
2	open62541-b...	NS2|String|EMCI_2.AnalogPeripherals.generalRefreshRate	generalRefre...	1
3	open62541-b...	NS2|String|EMCI_1.AnalogPeripherals.External_ADC7.value	value	654
4	open62541-b...	NS2|String|EMCI_1.AnalogPeripherals.generalRefreshRate	generalRefre...	1

UaExpert OPC UA diagnostic client



WinCC OA Client



quasar

quasar - Quick opc UA Server Generation Framework

- A tool for accelerated C++ server development and more
- CERN-made framework for model-driven creation of OPC UA software components
  - Generate servers, clients, SCADA integration layer, etc…
- Made with effort efficiency in mind (design, development, testing, deployment)
- quasar-based software used in LHC experiments (JCOP) as well as beyond CERN
- quasar can build 100% free and open source OPC UA servers and clients
- Validated on different platforms, operating systems, software deployment strategies etc...
- Choice of OPC UA stack used: UA SDK (paid license), open62541 (free & open-source)
- Dependencies are all open source

Build your OPC UA server with quasar in 4 steps

- Make sure all dependencies are installed
  - https://quasar.docs.cern.ch/quasar.html
- Preparing the environment
  - `mkdir ~/tmp`
  - `cd tmp`
- Cloning quasar from github
  - `git clone https://github.com/quasar-team/quasar.git --recursive`
- Create a quasar project
  - From my temporary directory

1

    - `cd quasar`
    - `./quasar.py create_project ~/tmp/opcua-server`
  - Now make initial commit of this project
    - `cd ~/tmp/opcua-server`
    - `git init && git add -A && git commit -am "initial commit"`

Build your OPC UA server with quasar in 4 steps

Configure the build and build it

- Now let's use only 100% free and open-source components for OPC UA: the open62541 library and its 'adapter' called open62541-compat:

2
```
./quasar.py enable_module open62541-compat
```

- And we need to use a file called 'build config', for open62541:

3
```
./quasar.py set_build_config ~/tmp/quasar/open62541_config.cmake
```

- Finally, let's build the server

4
```
./quasar.py build
```

- That's it we have made our newly built OPC UA server!

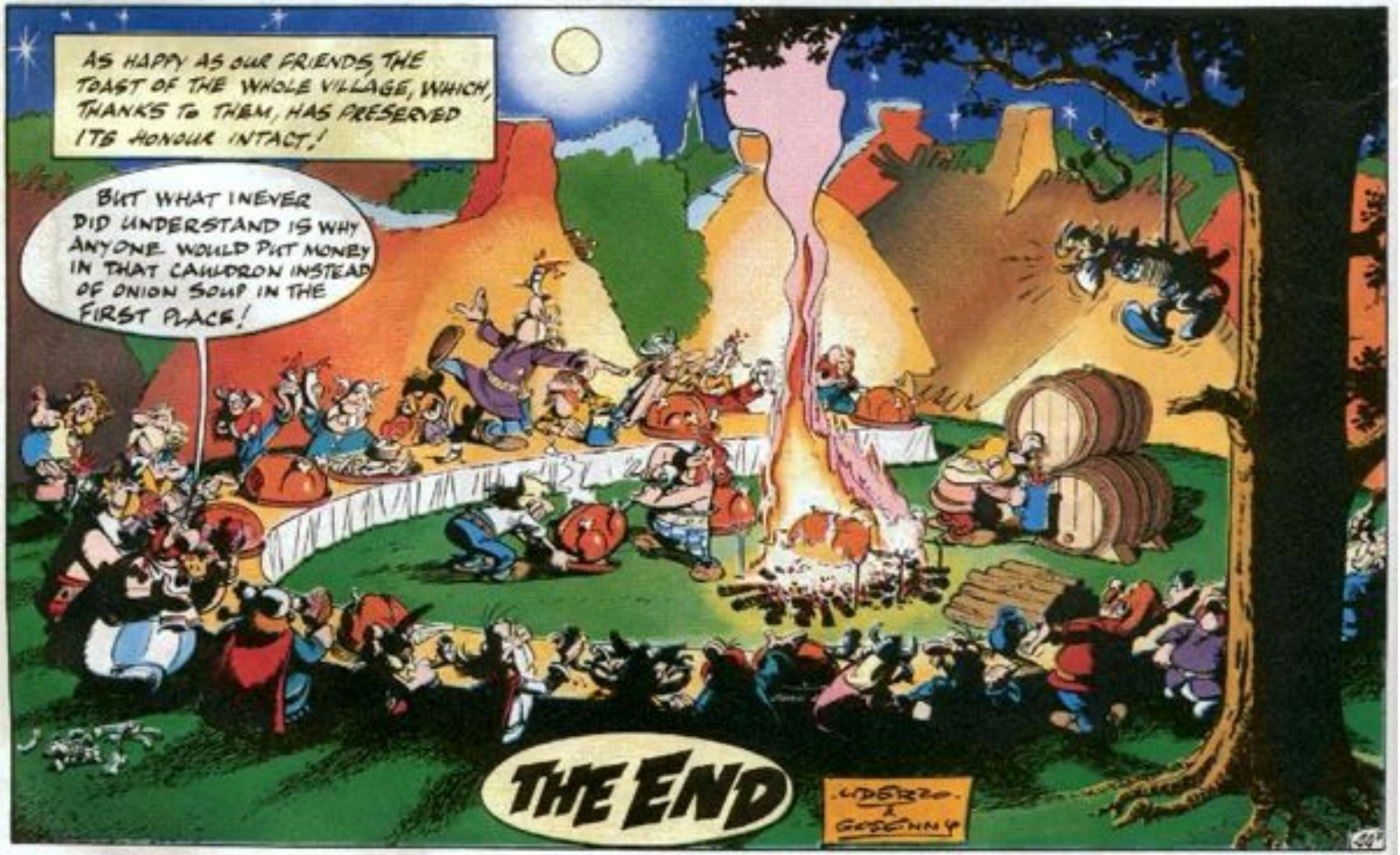
Further References

Check quasar’s youtube tutorial:
https://www.youtube.com/@quasaropcuatutorials4253/videos

Next steps: https://quasar.docs.cern.ch/

Summary

- Learned what is a detector control system and its importance in operating a large experiment
- And seen how large and complex control systems can be organized to be manageable
- Took a look at the detector control system of ATLAS
- And had a glimpse into OPC UA and its importance for homogenising the interface to the control systems
- Finally we shown a simple WinCC OA application, had a quick introduction into quasar framework and build our own OPC UA server


AS HAPPY AS OUR FRIENDS, THE TOAST OF THE WHOLE VILLAGE, WHICH, THANKS TO THEM, HAS PRESERVED ITS HONOUR INTACT!
BUT WHAT I NEVER DID UNDERSTAND IS WHY ANYONE WOULD PUT MONEY IN THAT CAULDRON INSTEAD OF ONION SOUP IN THE FIRST PLACE!
THE END