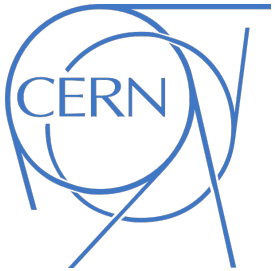



# ATCA Shelf Manager Controls & Monitoring

## Status of AtcaOpcUa server and Integration in DCS Applications

Paris Moschovakos, Piotr Nikiel, Stefan Schlenker

paris.moschovakos@cern.ch

15th xTCA Interest Group Meeting
2020/05/13

# Outline

- Intro: ATLAS ATCA DCS Scheme
- Plans set in 14 xTCA Interest Group Meeting
- Progress on AtcaOpcUa Server
- Integration with WinCC OA
- Integration with ATLAS FSM
- Use cases in ATLAS
- Conclusions

# Introduction

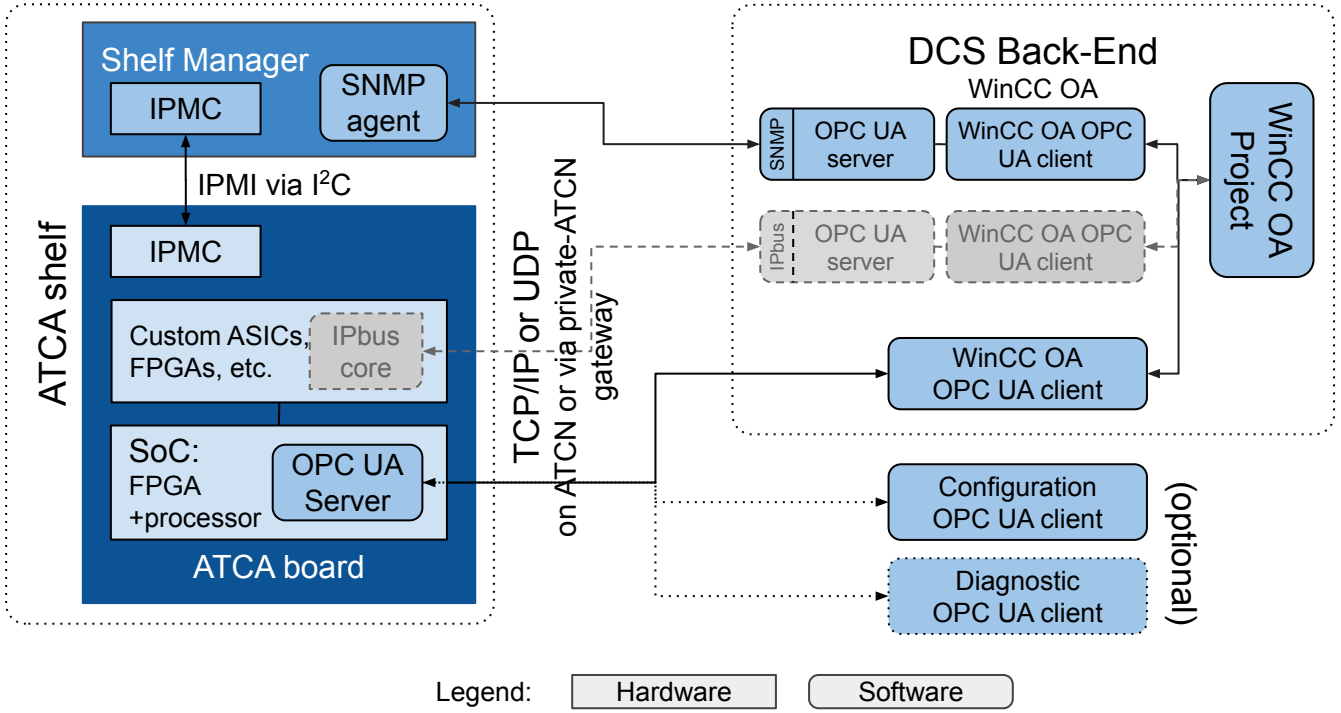



Diagram from S. Schlenker
DCS: Requirements Document for HL-LHC
EDMS Id: 2276493 v.1

# ATLAS Off-Detector Electronics Based on ATCA

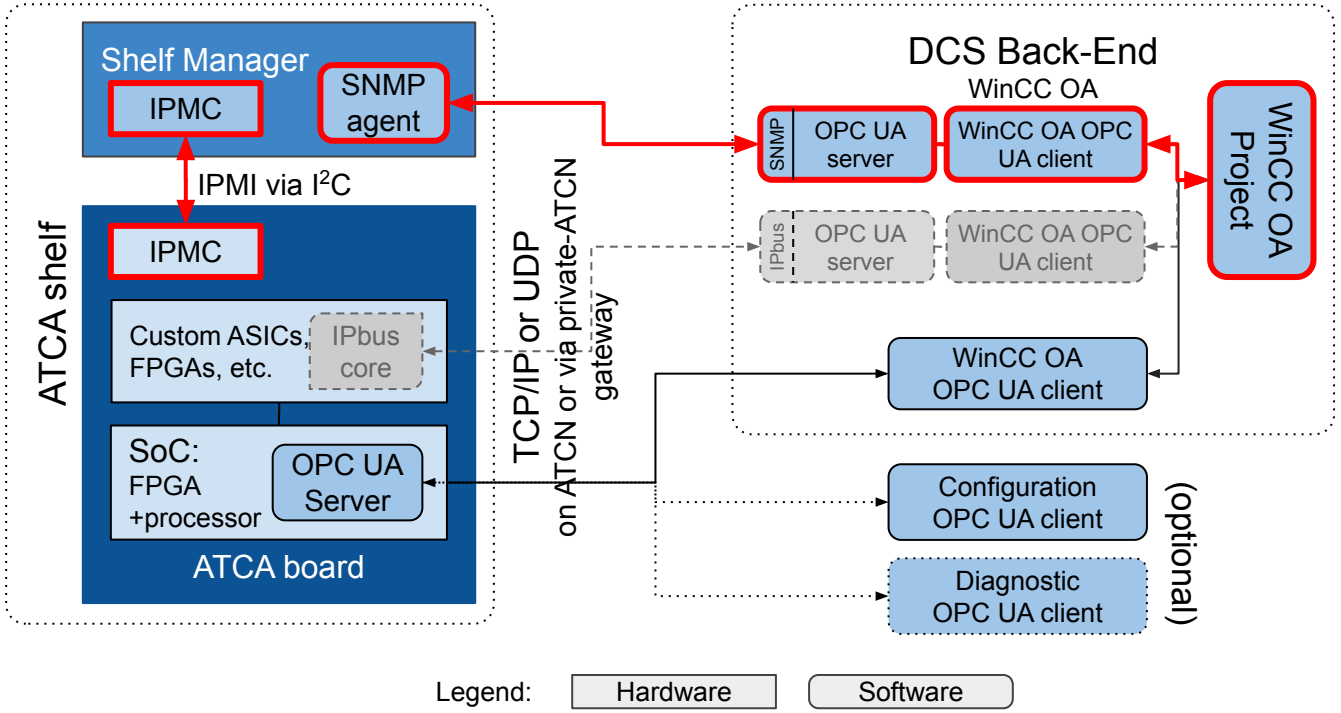
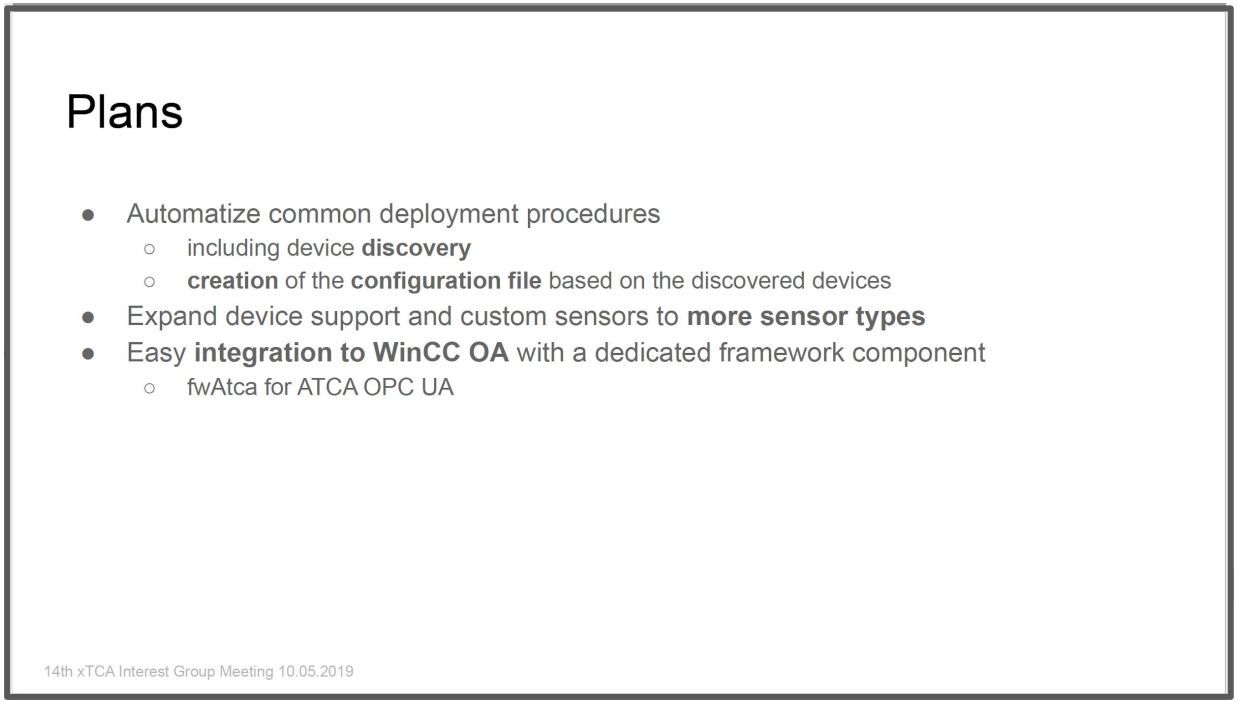



Diagram from S. Schlenker
DCS: Requirements Document for HL-LHC
EDMS Id: 2276493 v.1

# 14th xTCA Interest Group Meeting - Plans

- Plans from last year's xTCA meeting:



## Plans

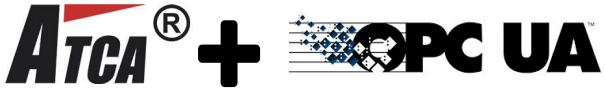
- Automatize common deployment procedures
  - including device **discovery**
  - **creation** of the **configuration file** based on the discovered devices
- Expand device support and custom sensors to **more sensor types**
- Easy **integration to WinCC OA** with a dedicated framework component
  - fwAtca for ATCA OPC UA

14th xTCA Interest Group Meeting 10.05.2019

# AtcaOpcUa Server

# Why combining ATCA with OPC UA?

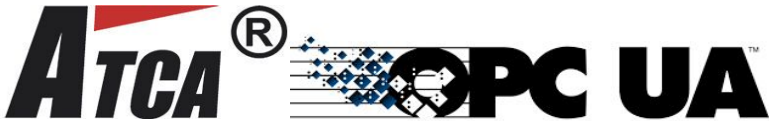
- OPC UA
  - focuses on communicating with industrial equipment and systems for data collection and control
  - Open specification and various implementations available (free or commercial)
  - Cross-platform
  - Service-oriented architecture
  - Integral information model, which is the foundation of 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
  - ATLAS DCS and BE-ICS developed a framework, quasar, 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 through the quasar framework

# AtcaOpcUa Server for ShM management
## Basic Points

- A quasar OPC UA server for managing ATCA shelves via the **shelf manager path** based on the **SNMP** external interface to monitor/control activities using IPMI
- For “CERN-standard” **Pigeon Point** Shelf Managers (ShMM 500, ShMM 700R)
  - Compatible with **xTCA**
- The ATCA Software is **template**-based on the MIB and auto **generated** using Jinja2
- Provides automatic **hardware discovery** walking over the SNMP tree
  - Only existing entries are populated
  - Throttling traffic towards for specific sensor types



# AtcaOpcUa
## Traffic Handling Improvements

ATCAs that are highly populated by sensors/IPMC/boards can congest IPMB towards the shelf manager.

- ~O(100) of variables can be handled smoothly for individual shelves
- That is one of the reasons for the second path (SoC)

To ensure smooth traffic

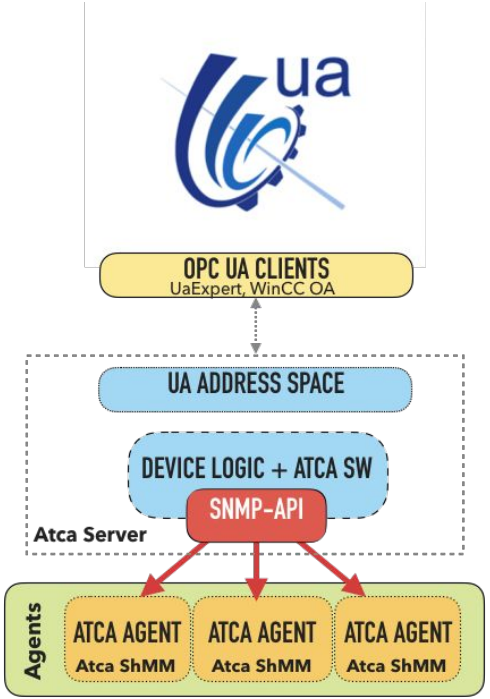
- A **survey** has been conducted within ATLAS ATCA shelves users to prioritize data relevant to DCS monitoring over excessive information and **based on the feedback from the community** in-server polling groups were used to categorize sensors (and their variables) with different priorities
- Support for a selection of advanced variables was introduced
- The server, while supporting all basic variables, by default populates the DCS-important ones during automatic discovery
- Introduced support for TELCO alarms which can capture and inform of emergency situations
    - Automatic actions based on alarming events should be done within-shelf manager using Platform Event Filtering mechanism

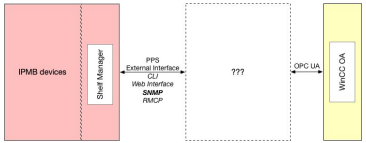
# SNMP module for OPC UA

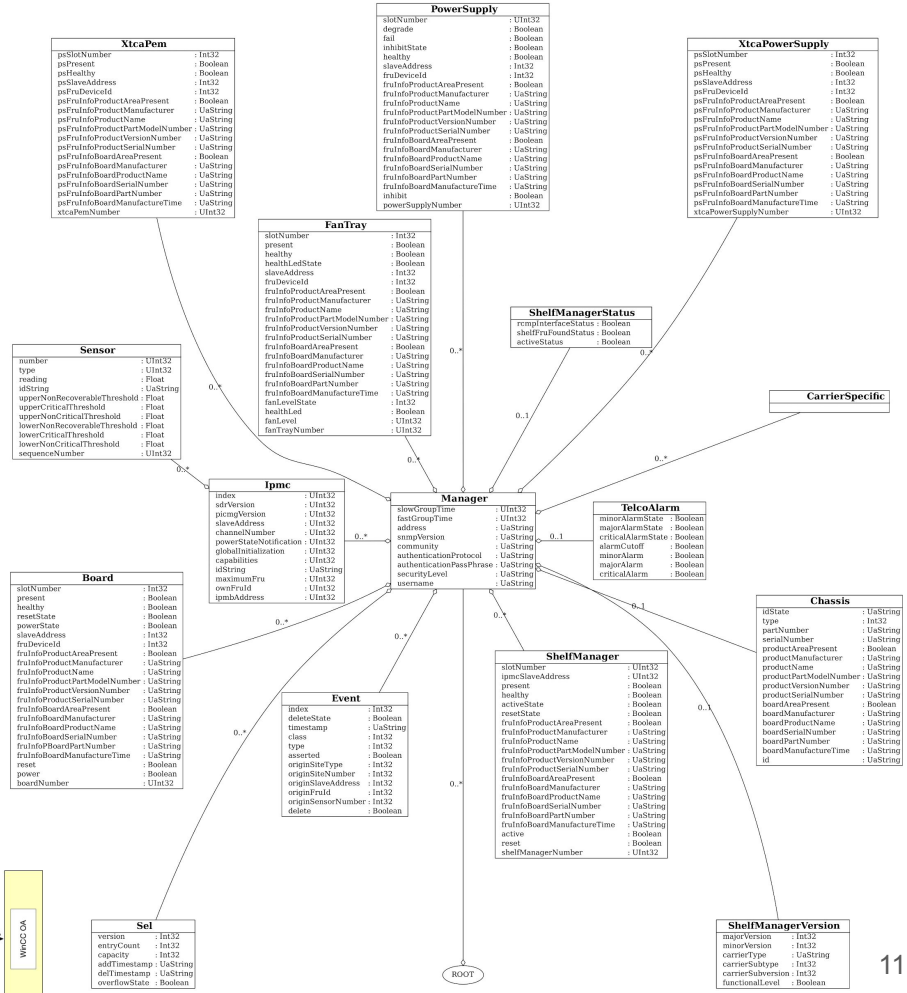
- An **ongoing effort** to provide a generalized C++ module has been initiated by Central DCS team that can be reused for various **SNMP-based devices** requiring control and monitoring using **OPC UA** solutions
- This module is **based on the AtcaOpcUa** server software backend experience
- The main motivation is to provide a commonalized way of interfacing to
  - Phase-II upcoming **power supplies** that use standard SNMP
  - the **AC/DC power rectifier** systems for ATCA shelves that are foreseen to be procured by centrally by CERN. It is actually part of CERN requirements
- **Work-In-Progress** in collaboration with BE-ICS to ensure long term maintenance

# AtcaOpcUa

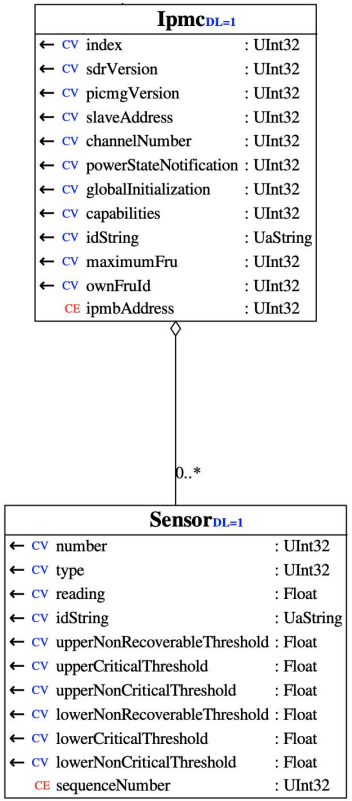
## Design Model Visualization



# IPMC and sub-detector specific hardware

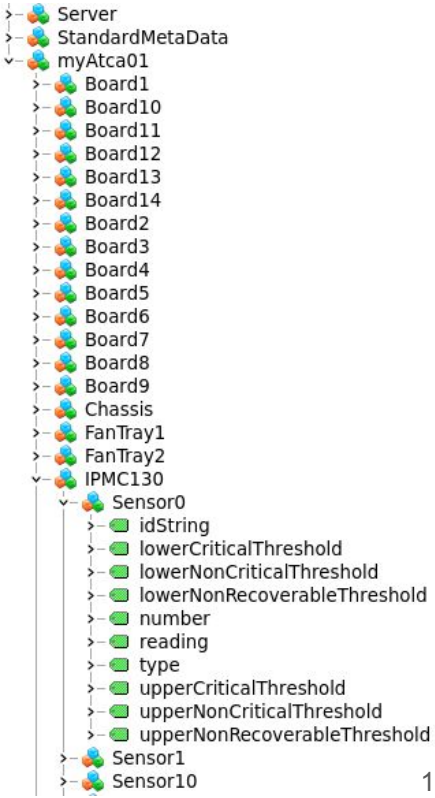



- Generic representation of IPMC and sensors following the hardware representation
  - Sensors “belong” to IPMCs
- Any **IPMC** that conforms with the standard can be monitored
  - The connected custom **sensors** are also monitorable using their IPMC address and sequence number
- A sophisticated mechanism that distinguishes in-between types of sensors was developed
  - temperatures, voltages, fans speed etc
  - sensor types enriched
  - mechanism to facilitate **addition** of ad-hoc types of sensors was introduced
- IPMC and sensors are **automatically discovered** and populated into the server

# The AtcaOpcUa server in action

```
2020-03-02 09:34.40.416591 [MetaAmalgamate.cpp:3200, INF] StandardMetaData.Log configuration found in the configuration file, configur
2020-03-02 09:34.40.416661 [MetaAmalgamate.cpp:3133, INF] general non-component log level will be [INF]
2020-03-02 09:34.40.416743 [MetaAmalgamate.cpp:2578, INF] setting log level to [INF]
2020-03-02 09:34.40.416813 [MetaAmalgamate.cpp:3118, INF] no StandardMetaData.Log.ComponentLogLevels configuration found in the config
2020-03-02 09:34.40.416900 [MetaAmalgamate.cpp:3069, INF] configuration for logging component  handle [0] name [CalcVars] using value
2020-03-02 09:34.40.417015 [MetaAmalgamate.cpp:2516, INF] setting component [name:CalcVars id:0] to level [INF]
2020-03-02 09:34.40.417101 [MetaAmalgamate.cpp:3219, INF] no StandardMetaData.SourceVariableThreadPool configuration found in the conf
2020-03-02 09:34.40.417763 [MetaAmalgamate.cpp:3236, INF] no StandardMetaData.Quasar configuration found in the configuration file, co
2020-03-02 09:34.40.417852 [MetaAmalgamate.cpp:3250, INF] no StandardMetaData.Server configuration found in the configuration file, co
2020-03-02 09:34.40.418049 [SnmpBackend.cpp:64, INF] [asml1c-stf0.cern.ch] Using SNMP version 2c
2020-03-02 09:34.40.438717 [CalculatedVariablesEngine.cpp:262, INF, CalcVars]  #ParserVariables: 1181 #CalculatedVariables: 0 #Synchro
2020-03-02 09:34.40.439041 [CalculatedVariablesEngine.cpp:297, INF, CalcVars] Optimized(suppresed) 1181 ParserVariables not used in an
2020-03-02 09:34.40.439104 [CalculatedVariablesEngine.cpp:262, INF, CalcVars]  #ParserVariables: 0 #CalculatedVariables: 0 #Synchroniz
2020-03-02 09:34.40.439149 [QuasarServer.cpp:78, INF] Initializing Quasar server.
2020-03-02 09:34.40.441714 [opcserver.cpp:157, INF] Opened endpoint: opc.tcp://pcaticstest08.dyndns.cern.ch:48050
2020-03-02 09:34.40.441777 [QuasarServer.cpp:48, INF] Server main loop started!
```

| Node Id | Display Name | Value | Datatype | Source Timestamp | Server Timestamp | |
|---|---|---|---|---|---|---|
| NS2\|String\|myAtca01.IPMC92.idString | idString | Upper Fan Tray | String | 9:38:28.522 AM | 9:38:53.592 AM | Good |
| NS2\|String\|myAtca01.IPMC92.Sensor10.idString | idString | Fan Tach 2 | String | 9:38:38.244 AM | 9:39:25.453 AM | Good |
| NS2\|String\|myAtca01.IPMC92.Sensor10.reading | reading | 5400 | Float | 9:41:16.578 AM | 9:41:16.578 AM | Good |
| NS2\|String\|myAtca01.IPMC92.Sensor11.idString | idString | Fan Tach 3 | String | 9:38:38.738 AM | 9:39:46.154 AM | Good |
| NS2\|String\|myAtca01.IPMC92.Sensor11.reading | reading | 5460 | Float | 9:40:17.874 AM | 9:40:17.874 AM | Good |
| NS2\|String\|myAtca01.IPMC92.Sensor9.idString | idString | Fan Tach 1 | String | 9:38:37.442 AM | 9:40:10.623 AM | Good |
| NS2\|String\|myAtca01.IPMC92.Sensor9.reading | reading | 5460 | Float | 9:41:16.504 AM | 9:41:16.504 AM | Good |
| NS2\|String\|myAtca01.IPMC92.Sensor8.idString | idString | Temp Out Right | String | 9:38:36.907 AM | 9:40:18.490 AM | Good |
| NS2\|String\|myAtca01.IPMC92.Sensor8.reading | reading | 19 | Float | 9:40:17.303 AM | 9:40:22.683 AM | Good |
| NS2\|String\|myAtca01.IPMC92.Sensor7.idString | idString | Temp Out Center | String | 9:38:35.835 AM | 9:40:26.703 AM | Good |
| NS2\|String\|myAtca01.IPMC92.Sensor7.reading | reading | 22 | Float | 9:41:16.301 AM | 9:41:16.301 AM | Good |



# Integration with WinCC OA

# WinCC OA and Integration Objectives

- **Standard** at CERN for Detector Control Systems
- Commercial and custom detector hardware is monitored and controlled through it
- It is the **interface** to the **shifter** in the control room
- Provides **interface to OPC UA servers** and integrates well with it

Objectives:

- ATCA shelves that use AtcaOpcUa should be **integrated** into the DCS
- Provide **monitoring** via WinCC OA UI
- Include **archiving** of **historical data** and **alarm** handling

Further ATLAS objectives:

- Integrate in **ATLAS FSM tree** structure

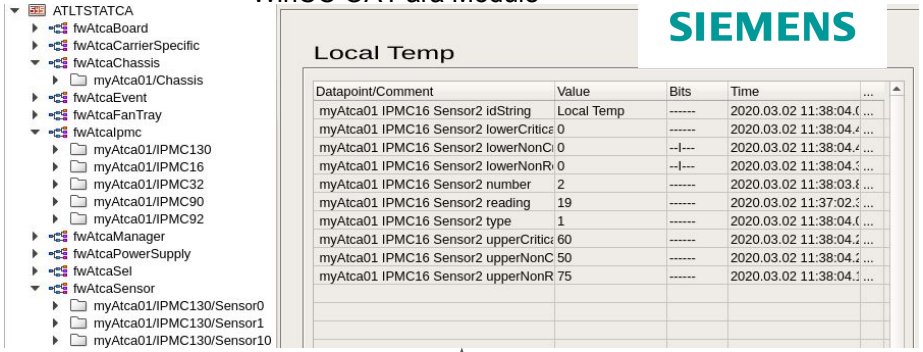
# WinCC OA data availability
## fwAtca

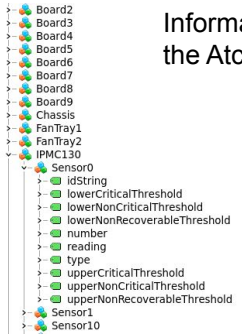
- WinCC OA integration is done by the **fwAtca** tool (uses fwQuasar)
- It **creates datapoints** for the **discovered hardware** based on the automatically created xml of the AtcaOpcUa server
- By default and **optionally creates additional configuration** for the datapoints including
  - Archiving
  - Alarms
  - Descriptions

# WinCC OA data availability

- fwAtca provides a **library** with **helper** functionality intended **for** use in sub-detector DCS experts **custom scripts**
- Functionality to set **alarm** limits on sensors **based on hardware** defined **thresholds** (non-critical, critical, non recoverable)
  - Communication to the hardware is required

WinCC OA Para Module



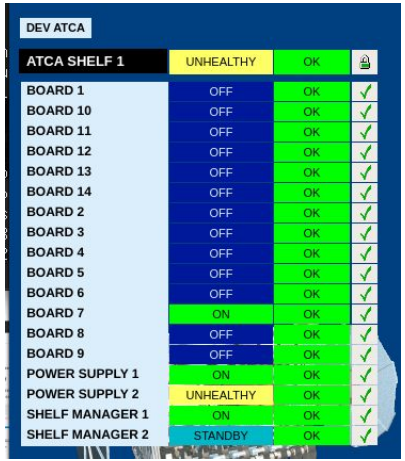

Information as exposed by the AtcaOpcUa server

fwAtca tool UI

# Integration with ATLAS FSM

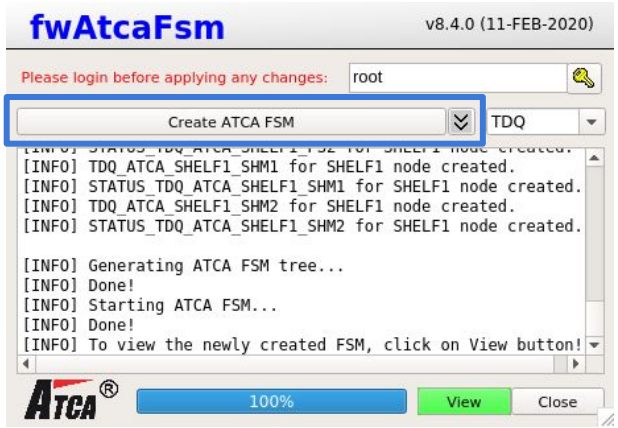
# ATLAS FSM integration
## fwAtcaFsm

- Many sub-detectors with suchlike (CERN-standard) ATCA setups that need to integrate with ATLAS FSM
- A **common centralized way** of generating the ATLAS FSM tree for all ATCA was chosen, allowing for custom user extensions
- The motivation is to provide easy **uniform monitoring** (bottom to top state/status propagation)
- To allow shifters have an overview with a **nicely informative visual interface**
- The **fwAtcaFsm** tool **automates** the tedious **development** procedure by additionally
    - **Identifying** the sub-detector it is deployed into
    - **Handling** the **FSM** tree **states** during creation
    - **Creating the** sub-detector specific **ATCA FSM** tree based on its discovered hardware
    - Generate, Start, examine the FSM tree etc.



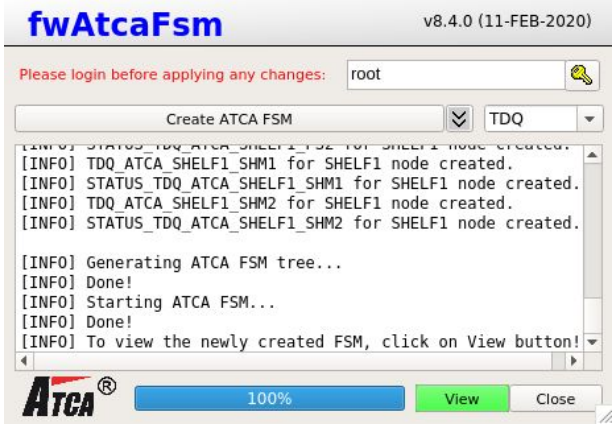
# ATLAS FSM integration
## fwAtcaFsm Usage

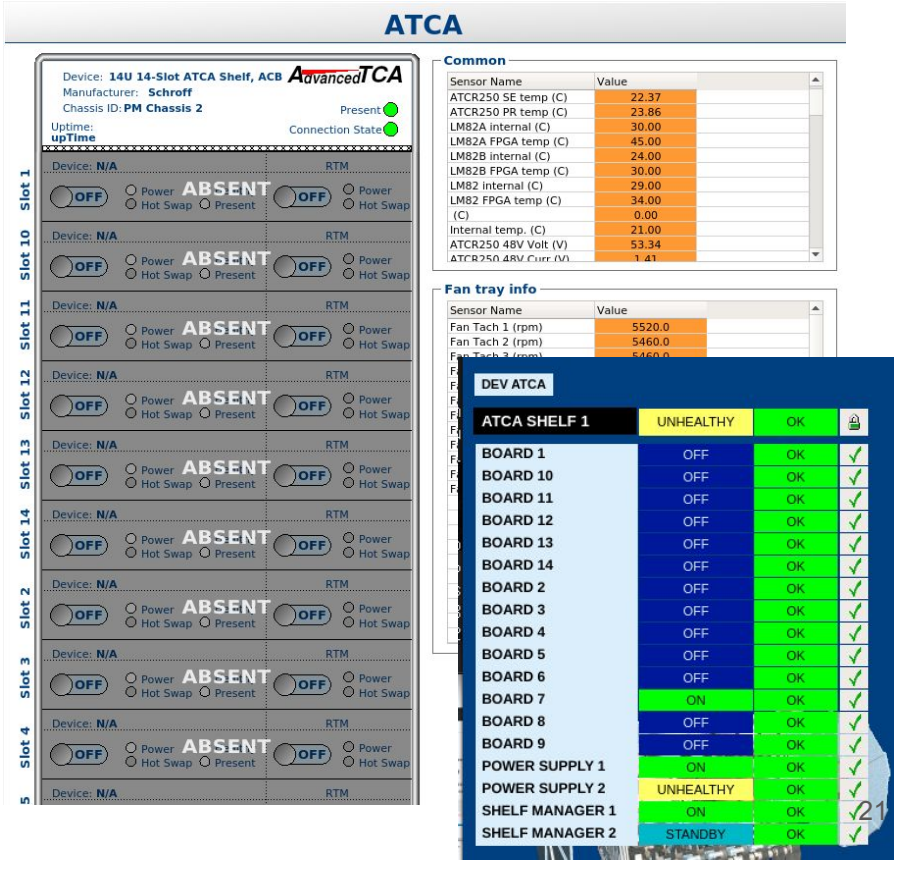
- Pleasant one-click operation
  - Literally <1 minute task



# ATLAS FSM integration
## fwAtcaFsm

- Pleasant one-click operation
  - Literally <1 minute task
  - Results to a **fully functional** ATLAS **FSM** ATCA **project (tree+UIs)**

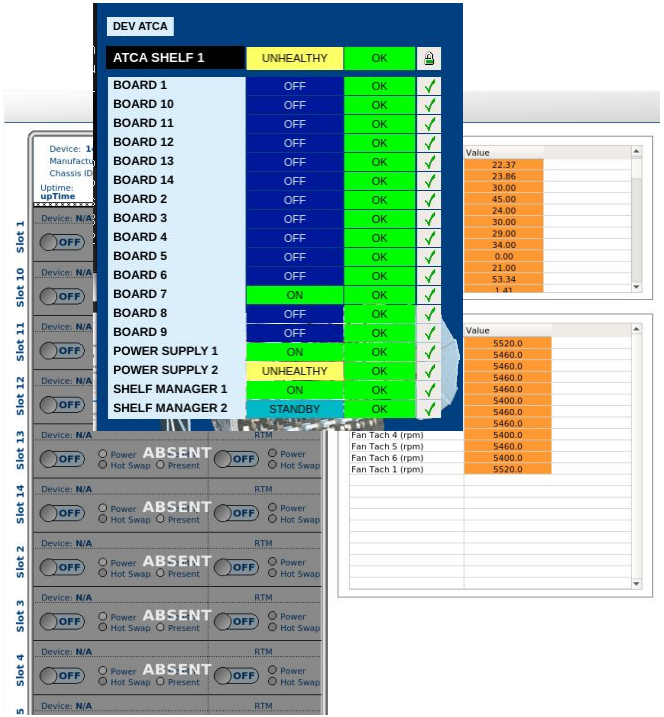


# ATLAS FSM integration

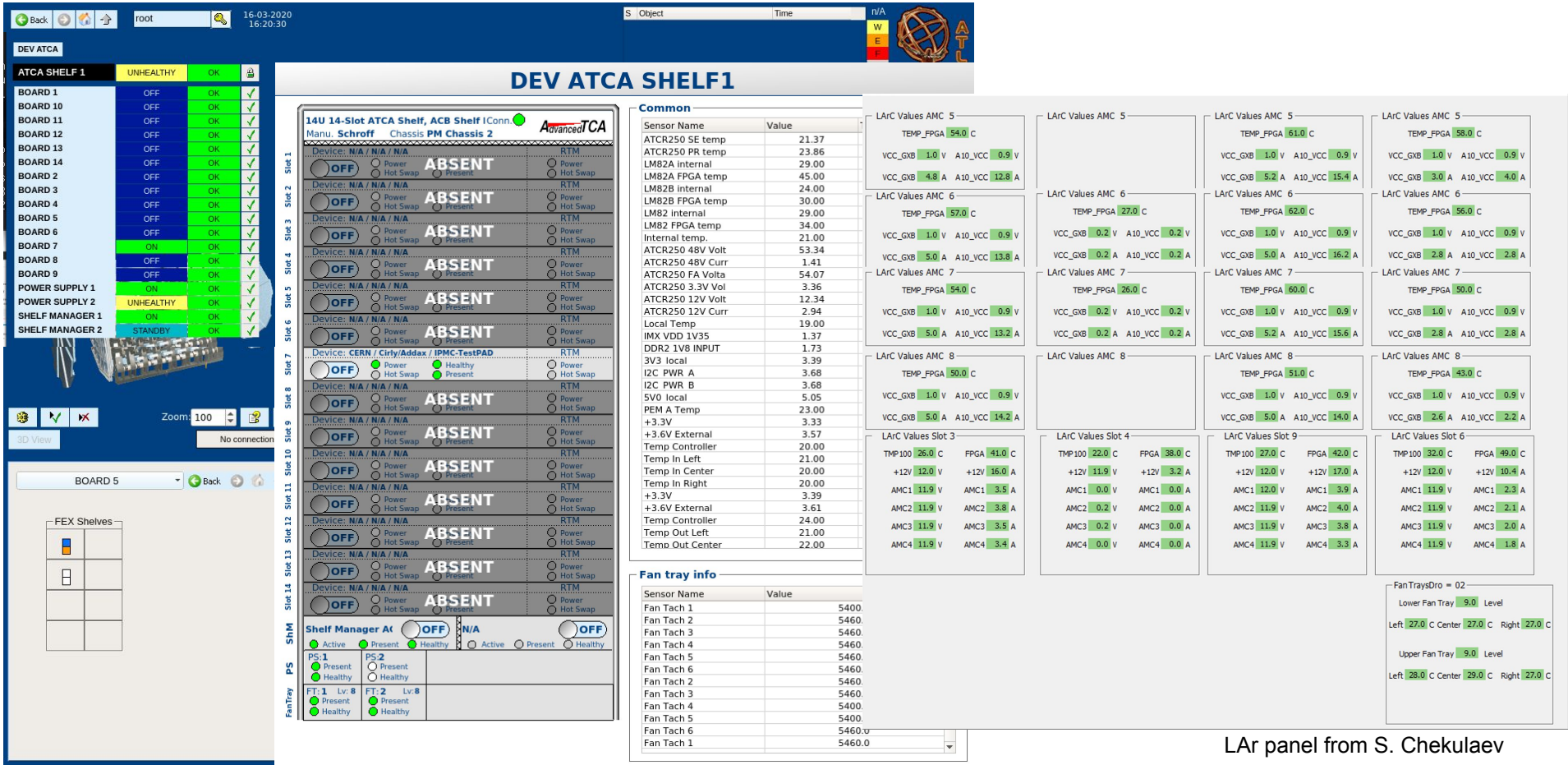
## ATCA FSM API & accompanying functionality

- The fwAtcaFsm framework package exposes additionally an **ATCA FSM API** (WinCC OA library) with **functionality to handle the flow** of the procedure to create the tree
  - Sub-detectors developers can use this method to create custom scripts in order to **integrate and extend** with more complex solutions (integrate with information from other ATCA-related OPC UA servers)
- Comprises functionality for custom board information
  - Functionality that **binds external-source information** to **board node status**. The ATCA FSM API user provides relevant to blade datapoints in a sensitivity list which is taken into account for the calculation of the status of the node



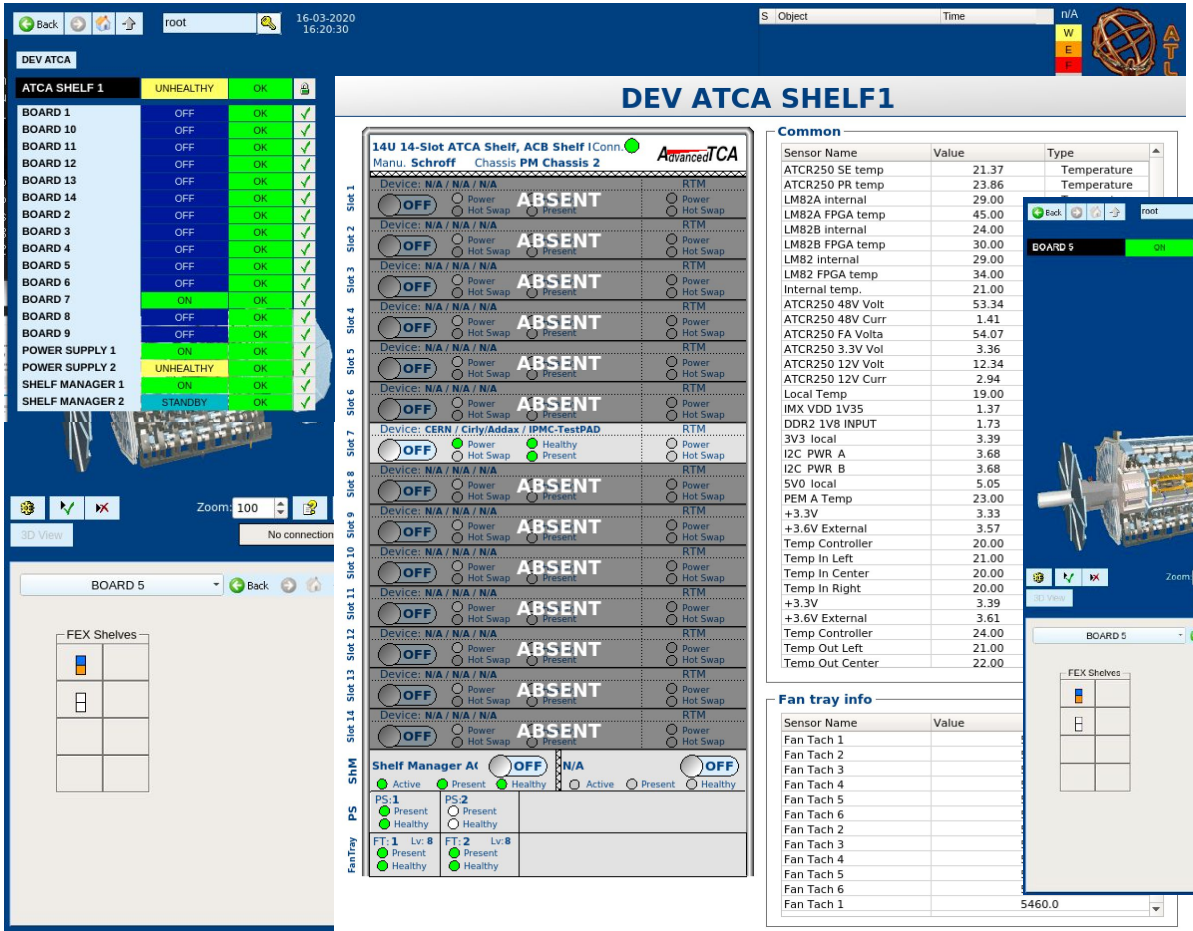

```
fwAtcaFsm_addDpesToBoardStatus(string boardDp,
                               dyn_string appendToSensitivitydpeList)
```

# LAR ATCA integrated in ATLAS FSM




LAr panel from S. Chekulaev

# STF L1Calo ATCA integrated in ATLAS FSM




Example in L1Calo, left fwAtcaFsm, right custom path for additional board information



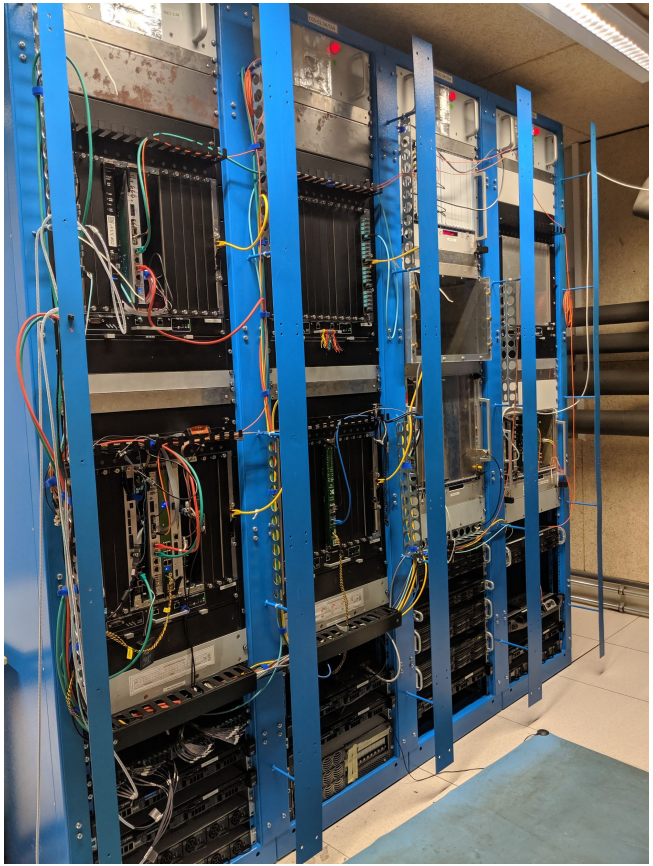
L1Calo panel from P. Thompson

# Use cases and applications

# Current AtcaOpcUa Use Cases in ATLAS

| | ATCA Shelves | OPC UA ATCA Servers | fwAtca | fwAtcaFsm | Status |
|---|---|---|---|---|---|
| **LAr** | 3 | 2 | ✓ | ✓ | In P1 and in EMF test setup |
| **TDAQ** | 6 | 1 | ✓ | ✓ | In STF development machine |
| **CSC** | 1 | 1 | | | In development machine |
| **NSW** | 4 | 3 | ✓ | ✓ | In integration sites and in development setup |

Big effort with sub-detectors to facilitate deployment in their test setups. Thanks to P. Thompson and TDAQ team for providing their hardware



Picture from Surface Test Facility pre-Covid

# Final Points

- The ATCA software ecosystem has made progress and fulfilled the required functionality
- An automatic discovery mechanism was introduced, covering standard and custom variables and sensors
- The performance has been improved taking into account users DCS-needs
- A set of tools for easy integration in WinCC OA complementing the controls ecosystem
  - Including archiving and alarm handling
- An extra set of tools for easy integration in ATLAS FSM has been introduced
  - One-click from datapoints to tree
- The software gained mileage in various sub-detectors ATCA setups including 2 LAr ATCA in P1

**Thank you!**

# AtcaOpcUa server and WinCC OA integration references

- General interest group for OPC UA (releases, news, feedback etc)
  - opc-ua-atca
- Main project page
  - https://gitlab.cern.ch/atlas-dcs-opcua-servers/AtcaOpcUa/-/releases
  - Suggested v. 0.9.1
- fwAtca - WinCC OA integration for AtcaOpcUa
  - https://gitlab.cern.ch/atlas-dcs-fwcomponents/fwAtca
  - Suggested v. 8.4.2
- fwAtcaFsm - ATLAS FSM integration
  - https://gitlab.cern.ch/atlas-dcs-fwcomponents/fwAtcaFsm
  - Suggested v. 8.4.2

**Thank you!**

# Backup