PLC Based Automation at European XFEL

Tobias Freyermuth & Sylvia Huynh
PLC Developer and Support Engineer
Electrical and Electronic Engineering

European XFEL
February 6th 2025

Outline

- Engineering Systems surrounding the Programmable Logic Controllers (PLC)
- Soliciting Requirements and Processes
- Transition Towards Sustainable Tooling
- Extraction of Information for the Automation of System Integration
- Customisation of PLC Configurations and Hardware Behaviours
- Future Developments and Questions

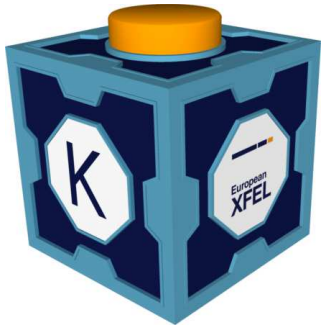
Engineering Systems interfacing into Programmable Logic Controllers (PLC): An Overview

https://www.vecteezy.com/free-vector/ip-network

PLC IP Network

Control IP Network

SCADA (Karabo)



Monitoring System

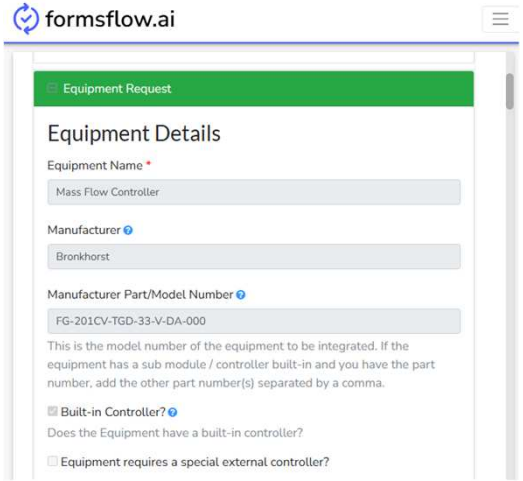
ZABBIX

PLC Terminals

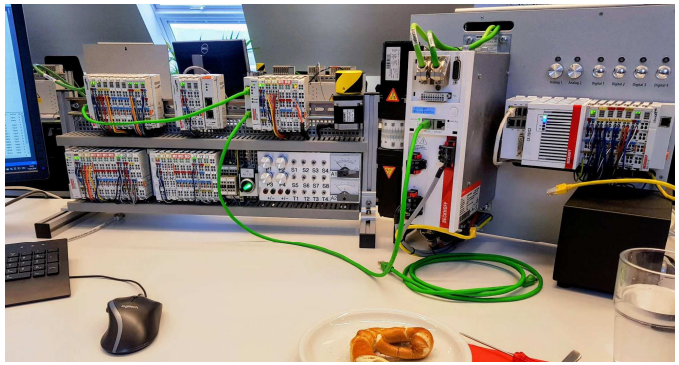
Array of equipment to be integrated directly into the PLC (eg. Motors, Valves, Pumps, Sensors)
~12,000 devices in operation

FPGA Timing Systems

Soliciting Requirements and Processes: A Bottom Up Approach



- An EIR (Equipment Integration Request) web form is completed
  - Details of the specific hardware
  - Feasibility of integration
  - Expected functionality
  - Mixture of both high and low level detail
- A ticket for tracking the integration process



- A cross-functionality team of device integrators is formed to commence integration.
  - This includes the review and development of the wiring, device functionality, SCADA integration and concludes with a final test with the requestor where possible.

Soliciting Requirements and Processes: Challenges

- This process is rather lengthy and involves a lot of discussion across teams during the review and implementation phase, especially where information is lacking, incorrect or unclear.
- Delays are often encountered
  - Obtaining updated manuals from the manufacturer
  - Encountering bugs in the hardware
  - Lack of prioritisation
- Testing
  - Test set-up are not always available
  - End-to-end testing

Toolings – a First Attempt

PLCMS DB  EPLAN Export  Project Data File  Interlock Definitions 

PLCMS
- Reads in data from all sources
- Data integrity checks and generates the PLC Project File

PLC Project File
- XML File, with a selected framework version
- Configuration and Initialization values

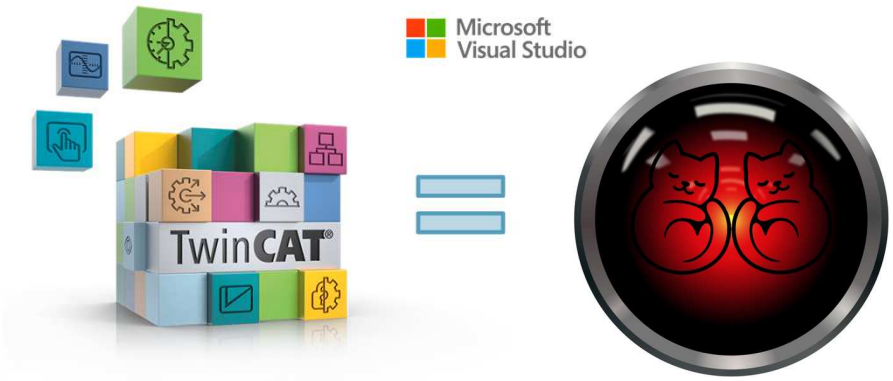
Project Builder
- Beckhoff XAE – C#
- Creates a TwinCAT PLC Project

PLC Project
- Ready for deployment

BECKHOFF TwinCAT

Transition Towards Sustainable Tooling: A Step Back, and a Review

- Work with your development environment
- Cohesion
- Ease of Maintenance
- Lower the entry hurdle



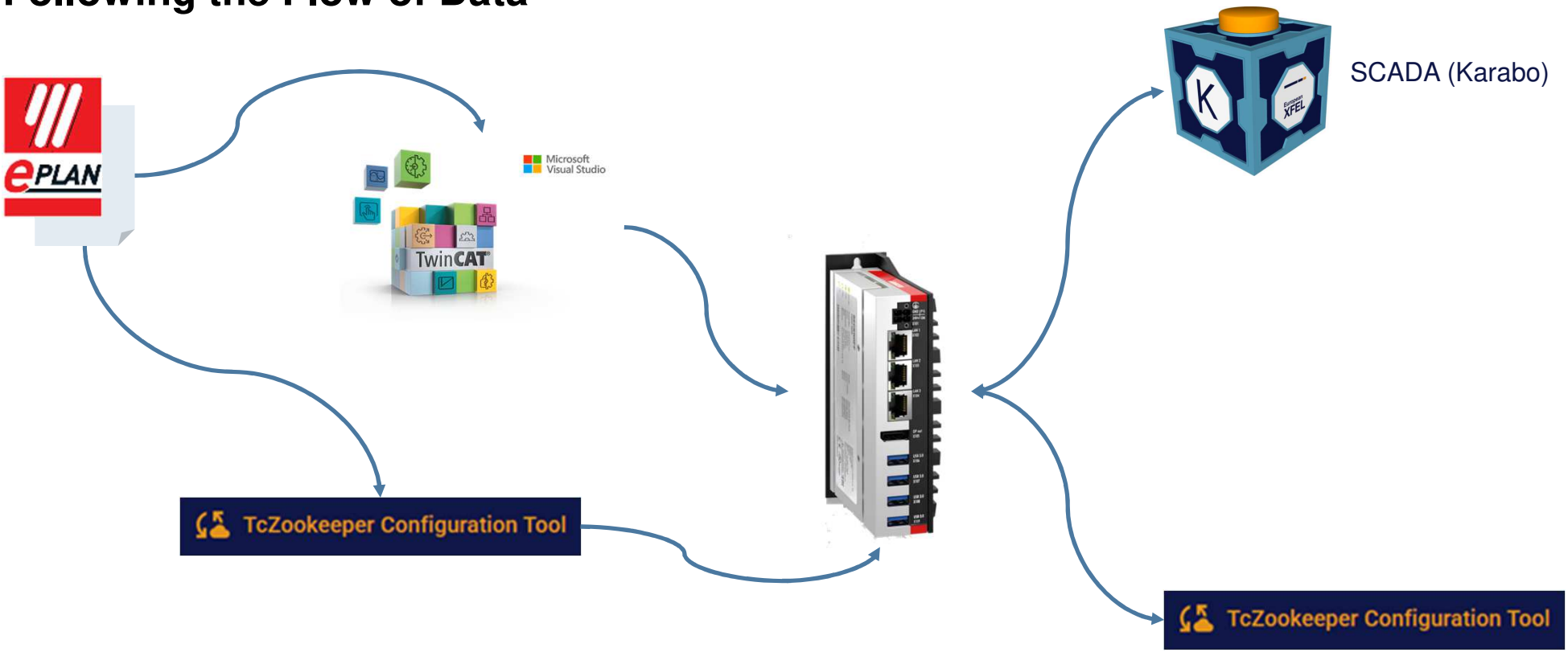
Extraction of Information for the Automatic Integration of Systems:

- What information is essential for the PLC?
- Which aspects would benefit the most from automation?
- How could that be achieved?

Narrowed down our focus to **three** main systems:

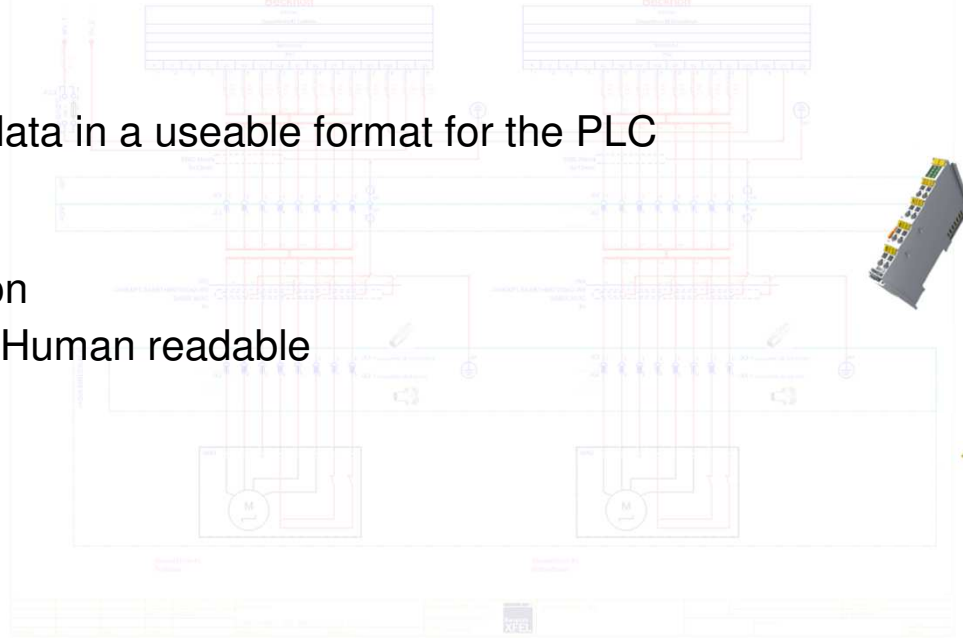
- Electrical wiring
- SCADA Interface
- PLC Interface itself
  - Focus on ease of development and configuration

Following the Flow of Data

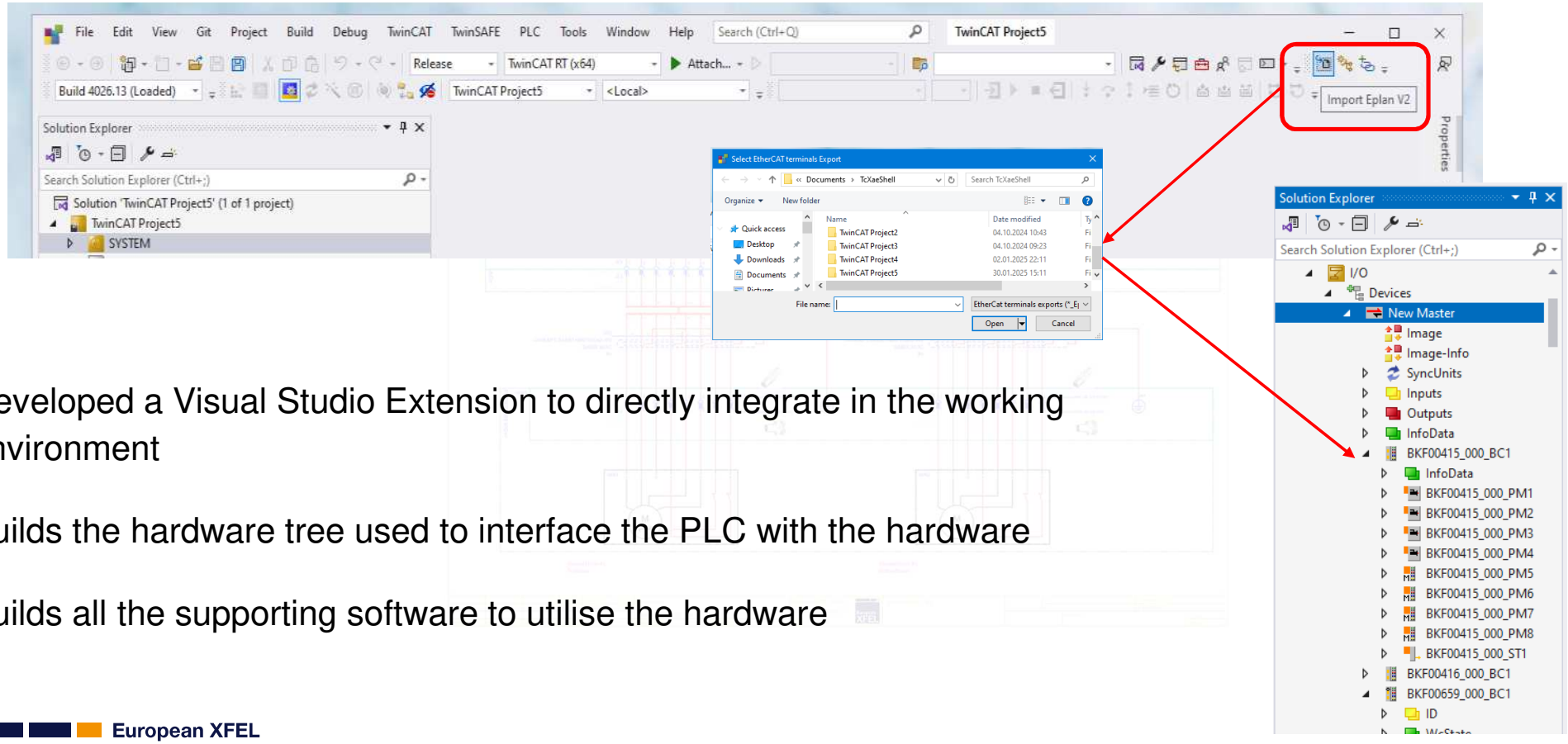


PLC – ECAT Integration:
EPLAN Exports – A Tale of Two Wires.

- The wiring diagram holds crucial information regarding how one piece of hardware connects with another.
- EPLAN Export to capture the data in a useable format for the PLC
- Develop a parser which:
  - Extracts relevant information
  - Machine readable but also Human readable
  - Acts as a Linter



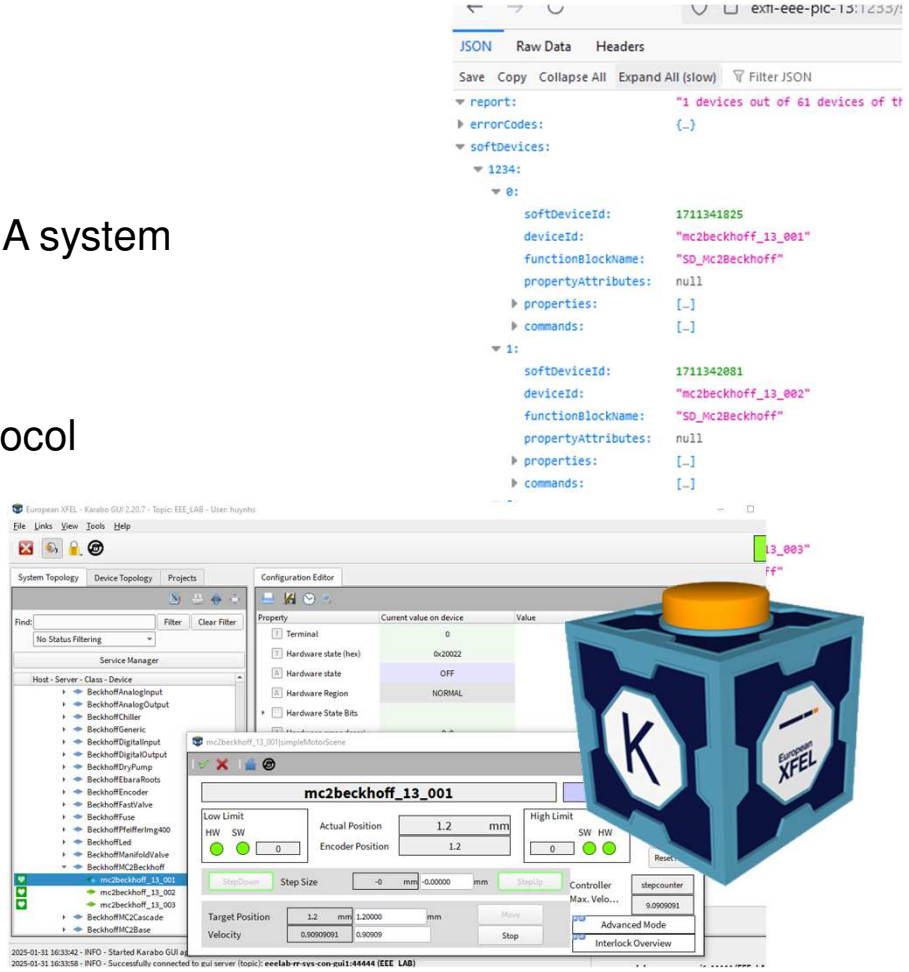
PLC – ECAT Integration: PLC Project Build Tool



- Developed a Visual Studio Extension to directly integrate in the working environment
- Builds the hardware tree used to interface the PLC with the hardware
- Builds all the supporting software to utilise the hardware

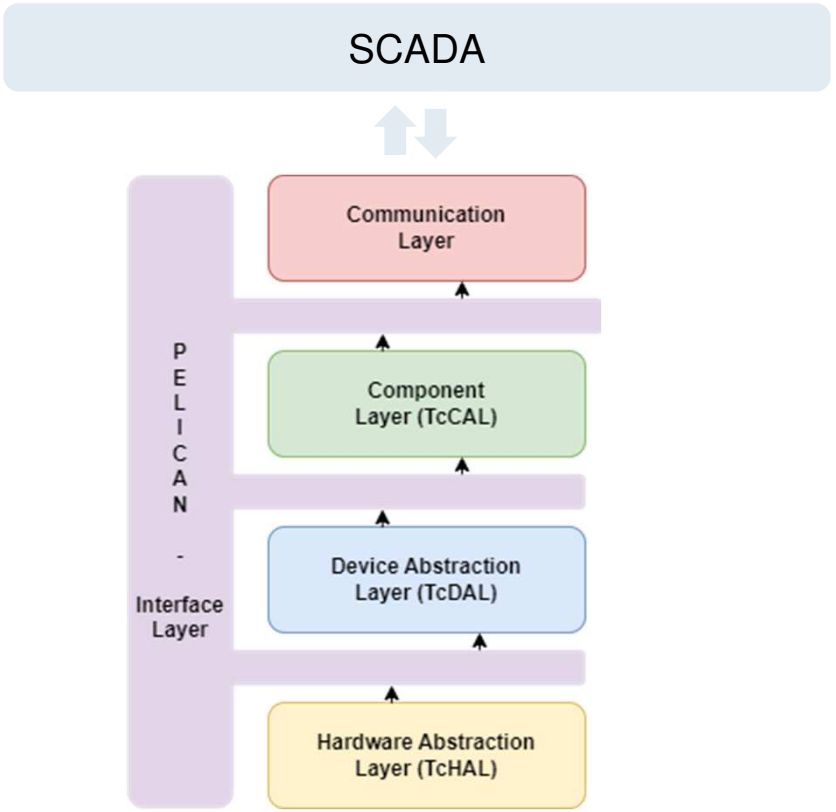
PLC – SCADA Integration:
SCADA – A User Interface

- PLCs are mostly useful when paired as part of a SCADA system
- PLC code built with modularity with abstraction
- Single communication layer to cater to the SCADA protocol
- Enables complex functionality to remain in the PLC
- Passes only information needed for the user



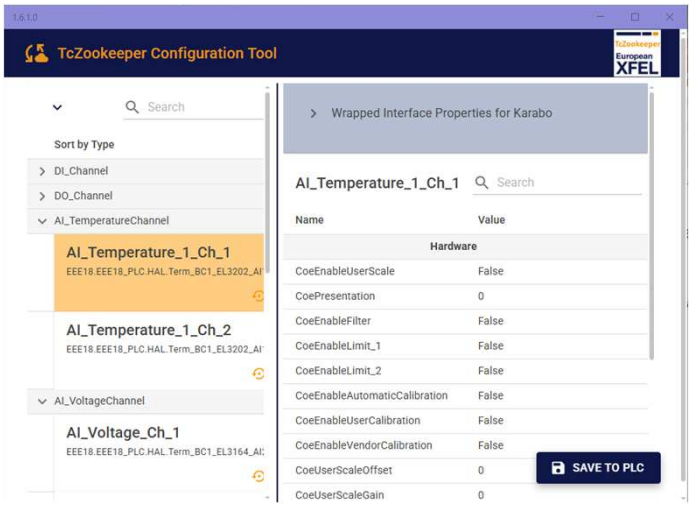
The PLC Interface

- PLC architecture relies on a layered abstraction approach
- Enables a clear division of scope
- Simplifies:
  - Testing
  - Debugging
  - Developing
  - Configuration



Customisation of PLC Configurations and Hardware Behaviours:

- Each PLC interfaces into several hundred devices
- To configure these devices a Configuration Tool was developed
- Simplifies the interface to the SCADA System
- Hardware settings and device functionality is obfuscated from the user



Future and Ongoing Developments

- Complete Hardware layer to support additional EtherCAT based fieldbus devices
- Improve testing methods
  - Test Driven Development (TDD)
  - Integration tests
- Maintenance plan to adapt to upgrades
  - EPLAN version updates
  - TwinCAT updates
  - VisualStudio (Extensions) updates

Image credit to : https://www.xfel.eu

Thank you for your attention,
- the PLC team.