Data Analytics for Industrial Control Systems

Abhit Patil, on behalf of
BE-ICS and Siemens openlab collaboration

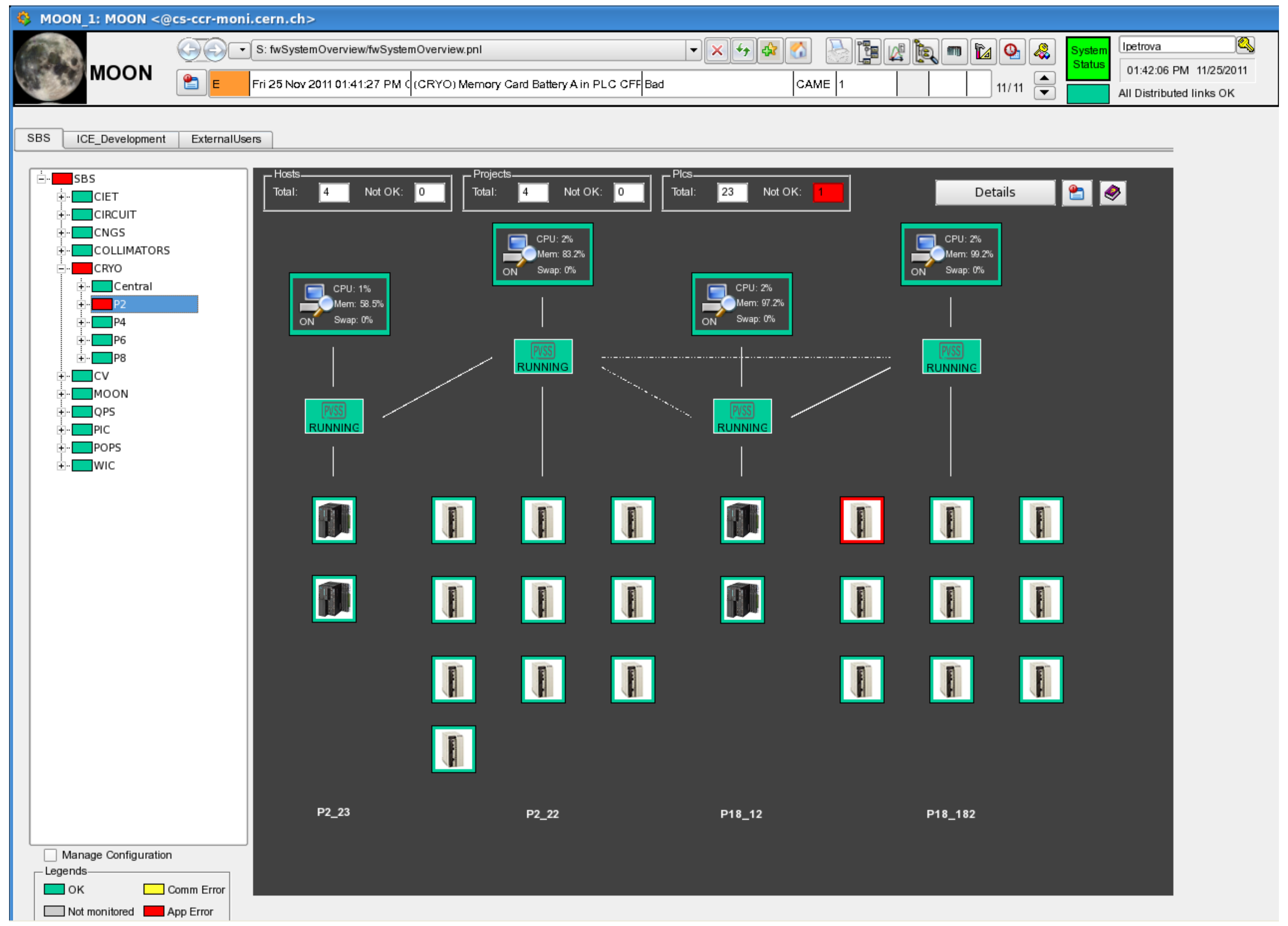
26th March 2024

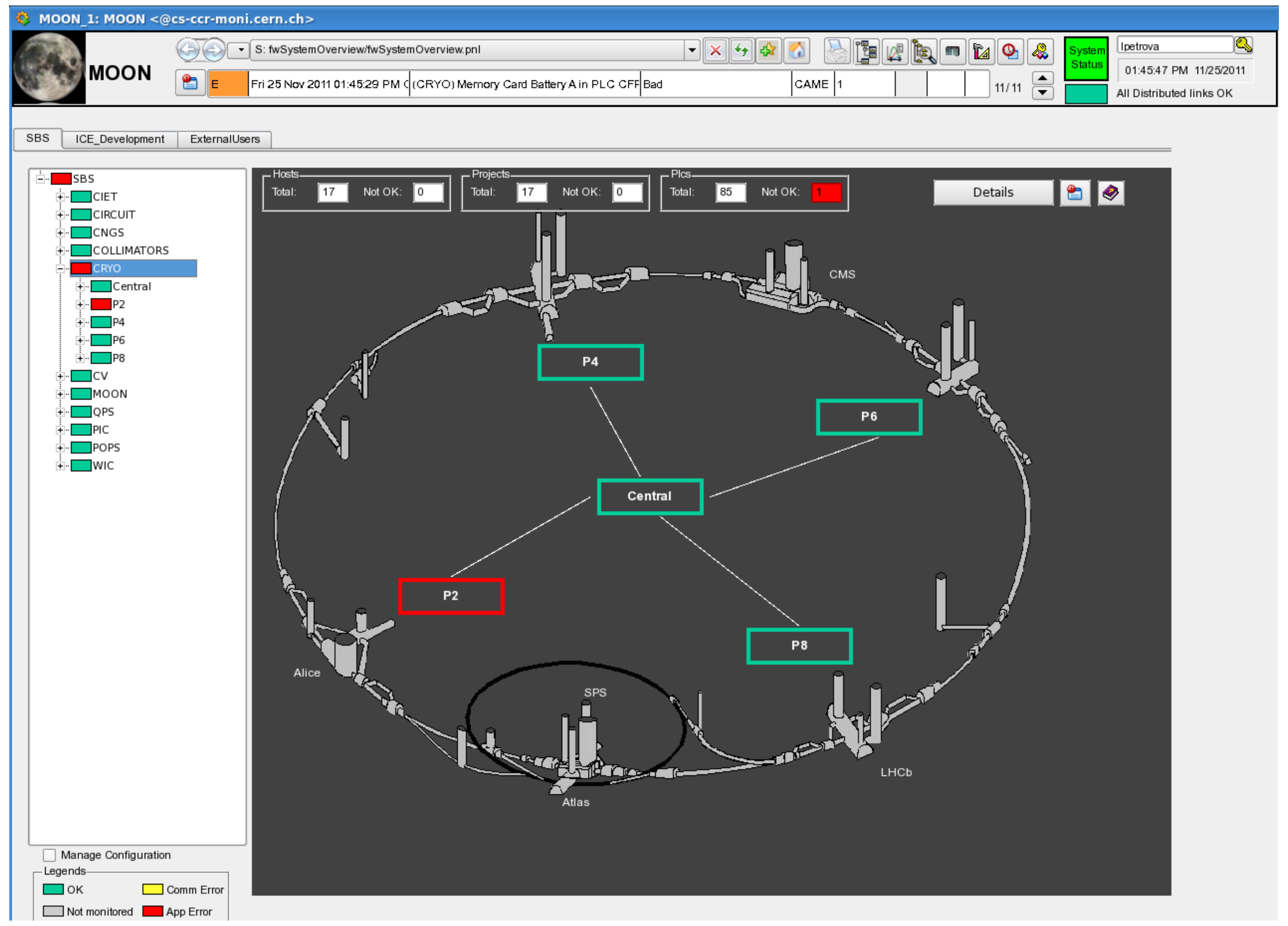
Subprojects

- Openlab seventh phase: 2021-2023
  - Device Monitoring
  - Edge Computing
- Openlab eighth phase: 2024-2026
  - Device Monitoring (Continued)
  - Industrial co-pilot

Collaborative model

- Siemens provides the latest solutions to tackle challenges at CERN.
- CERN’s complex control infrastructure is a testbed for evaluating and refining these solutions.
- Siemens product owners receive feedback, enabling them to modify or add new features.
- We co-develop prototype applications that integrate with various Siemens tools



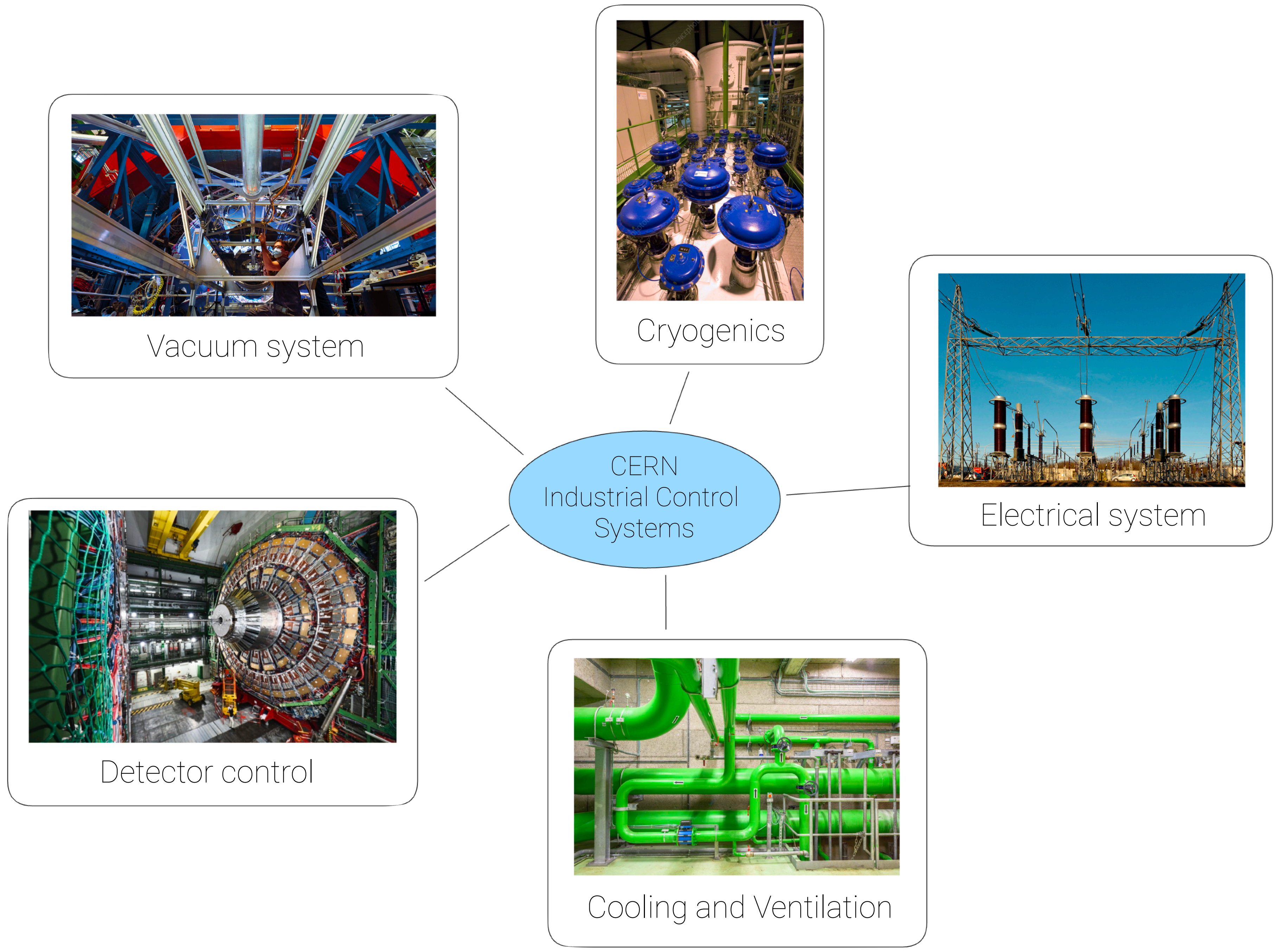


Device Monitoring

Objective

- Monitor & Diagnose large array of heterogenous industrial control devices
- Two main components
  - Data Aggregation
  - Data Visualization
- Aggregation
  - Collect and consolidate device diagnostics info in real-time
- Visualization
  - Visualize the entire system’s status in an intuitive manner

Challenges



- **Autonomy leading to heterogeneity:** The engineering teams responsible for their control system select the PLC models and software frameworks
- **Limited Influence:** Our group, BE-ICS, provides centralized service and product recommendations but cannot make procurement decisions
- **Lack of standardization:** The teams develop their code independently and hence BE-ICS cannot enforce uniform standards for diagnostics

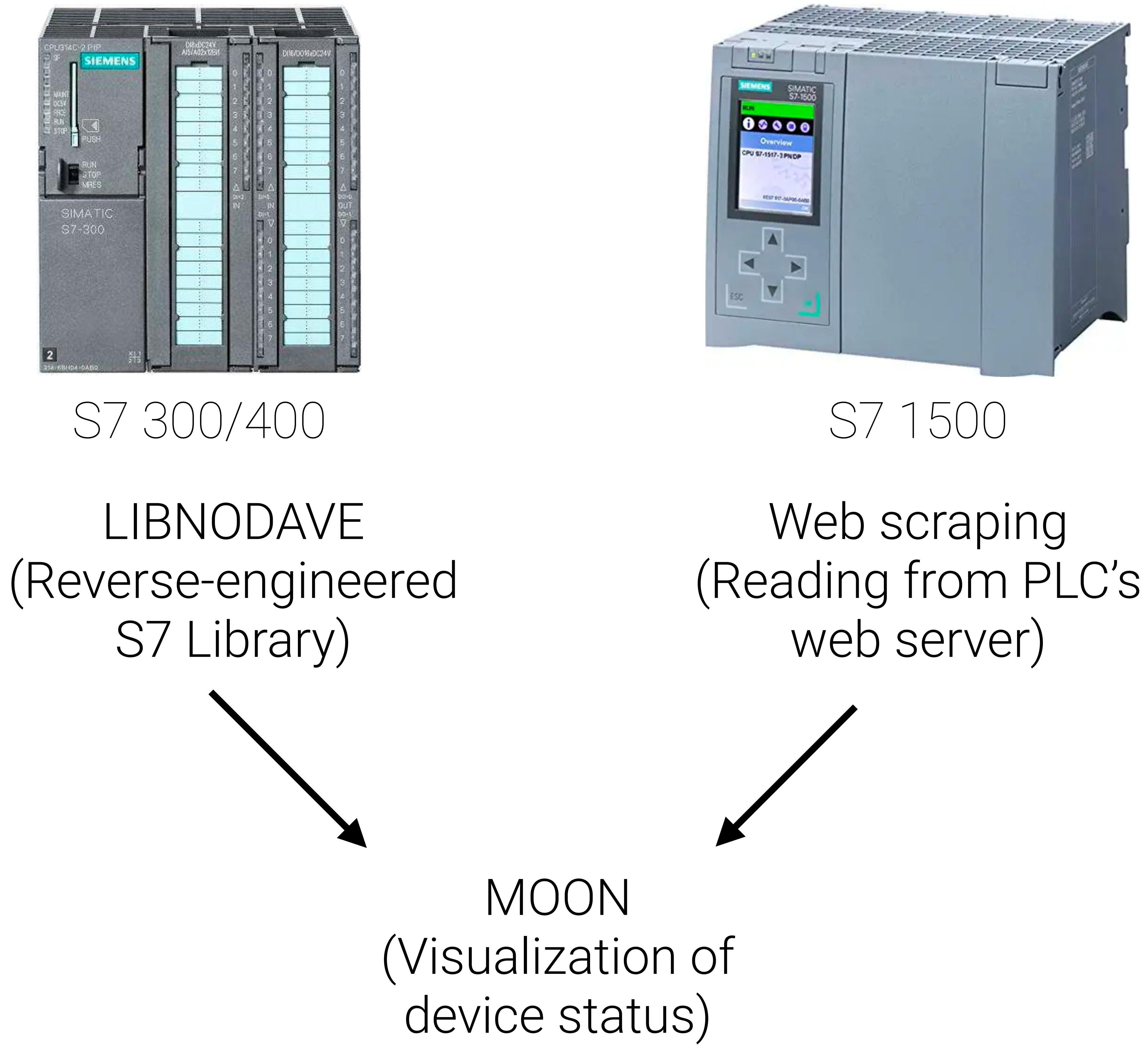
Example scenario

Cooling towers at CERN

- During upgrades, cooling & ventilation engineering team decides to procure several S7-1500 PLCs
- They develop the control code using frameworks developed by BE-ICS
- They do not transfer diagnostics information to PLC data block, preventing it from being transmitted northbound
- Consequently, a monitoring agent based on reverse-engineered library, has to be used to retrieve diagnostic information

Monitoring agents

- Different methods for extracting diagnostics based on the PLC model
- S7-300/400 - Agent based on <u>LIBNODAVE</u>
  - Old reverse-engineered library not supported or maintained by Siemens
- S7-1500 - Web scrapping
  - Extraction of HTML data from the PLC's web server
- SE PLCs



Siemens
S7 400


```
{
    "SI_CONN": 1,
    "SI_SF": 0,
    "SI_MODE_RUN": 1,
    "SI_BUS_1": 0,
    "SI_BUS_2": 0,
    "SI_BUSF": 0,
    "SI_TIME": "31/10/22 - 15:52:38;459",
    "SI_INFO": "Siemens PLC S7 300; CPU slot: 2.
            Module ID: 6ES7 317-2EK14-0AB0
            Hardware ID: 8
            Firmware ID: V 3.2.12
            PLCCycleTime : 18
            PLCLastStartDate : 26/10/22 - 14:17:55;736
            Request for manual warm restart",
    "SI_DIAG": "31/10/22 - 15:52:38;459
            PLCDiag 1 : 26/10/22 - 14:17:55;741
            Mode transition from STARTUP to RUN
            PLCDiag 2 : 26/10/22 - 14:17:55;736
            Request for manual warm restart
            PLCDiag 3 : 26/10/22 - 14:17:55;637
            Mode transition from STOP to STARTUP
            PLCDiag 4 : 26/10/22 - 14:17:03;262
            STOP caused by PG STOP operation or by SFB20 STOP
            PLCDiag 5 : 26/10/22 - 09:31:49;022
            Mode transition from STARTUP to RUN
            PLCDiag 6 : 26/10/22 - 09:31:49;017
            Request for manual warm restart
            PLCDiag 7 : 26/10/22 - 09:31:48;919
            Mode transition from STOP to STARTUP
            PLCDiag 8 : 26/10/22 - 09:30:51;093
            STOP caused by PG STOP operation or by SFB20 STOP
            PLCDiag 9 : 26/10/22 - 09:12:23;509
            Mode transition from STARTUP to RUN
            PLCDiag 10 : 26/10/22 - 09:12:23;504
            Request for manual warm restart",
    "SI_REPLY": 1
}
```


Schneider Electric
M340/580

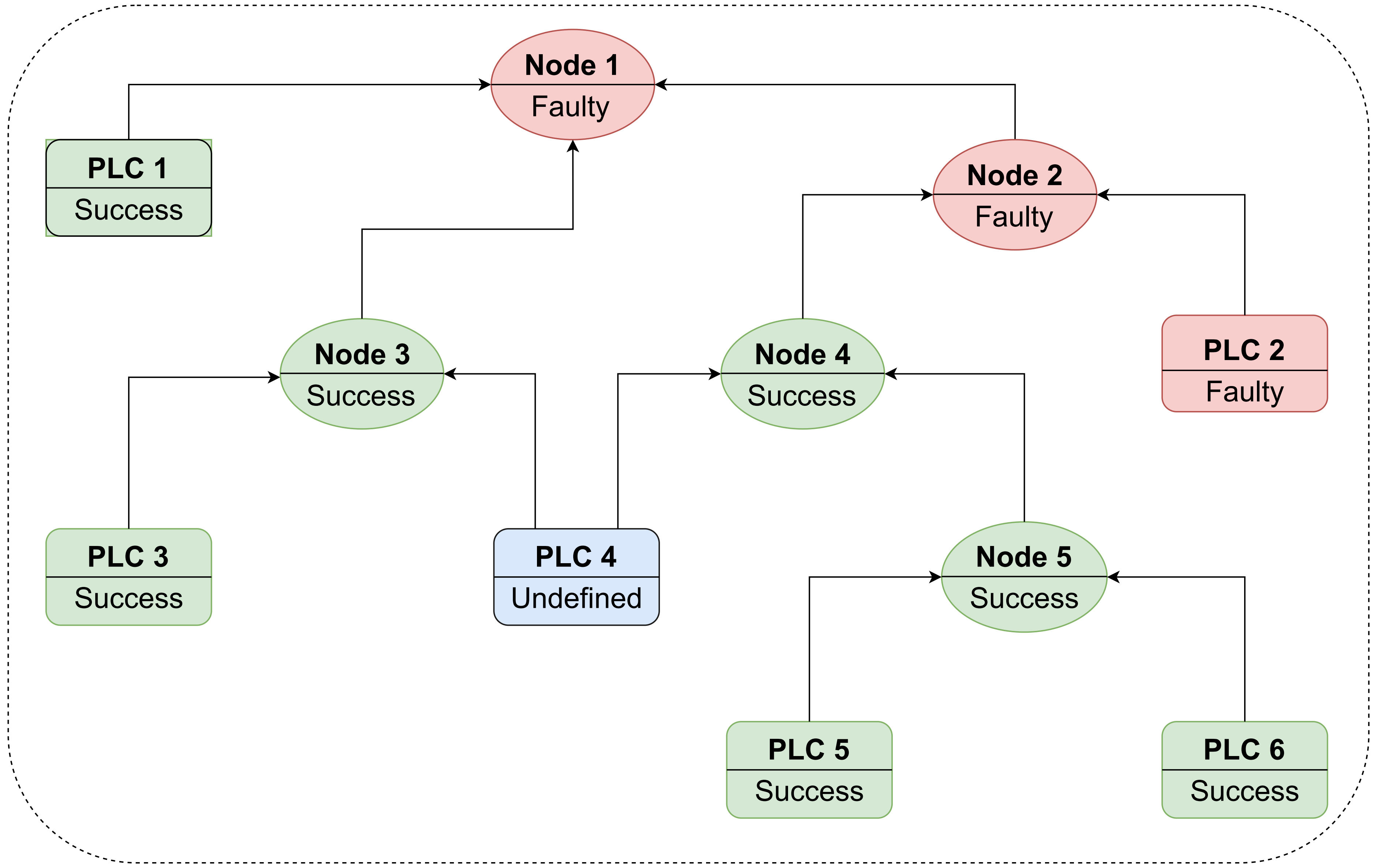

```
{
    "SC_CONN": 1,
    "SC_MODE_RUN": 1,
    "SC_MCB_APP": 0,
    "SC_BATT": 0,
    "SC_MCB_DAT": 0,
    "SC_IE_IO": 1,
    "SC_DTIME": 0,
    "SC_TIME": "PLC Current Time: 09/12/2022 16:00:40",
    "SC_INFO": "IP: 172.26.2.60
            Commercial version of Processor: 2.20
            Firmware version of Processor: 9
            Xway address/Station number (Premium only): 0
            Xway address/Network number (Premium only): 0",
    "SC_DIAG": "PLC Last Stop: 18/11/2022 14:33:00 (Code 0x0004: Power outage)
            CPU Error: 0
            Block Error Type: 0xdef1: Character string transfer error
            Programmed MAST Task Cycle: 0 ms
            Current MAST Task Cycle: 6 ms
            Programmed FAST Task Cycle: 0 ms
            Current FAST Task Cycle: 0 ms",
    "SC_REPLY": 1
}
```


Agents retrieving diagnostic information from variety of PLCs and converting it to standard formats (JSON)

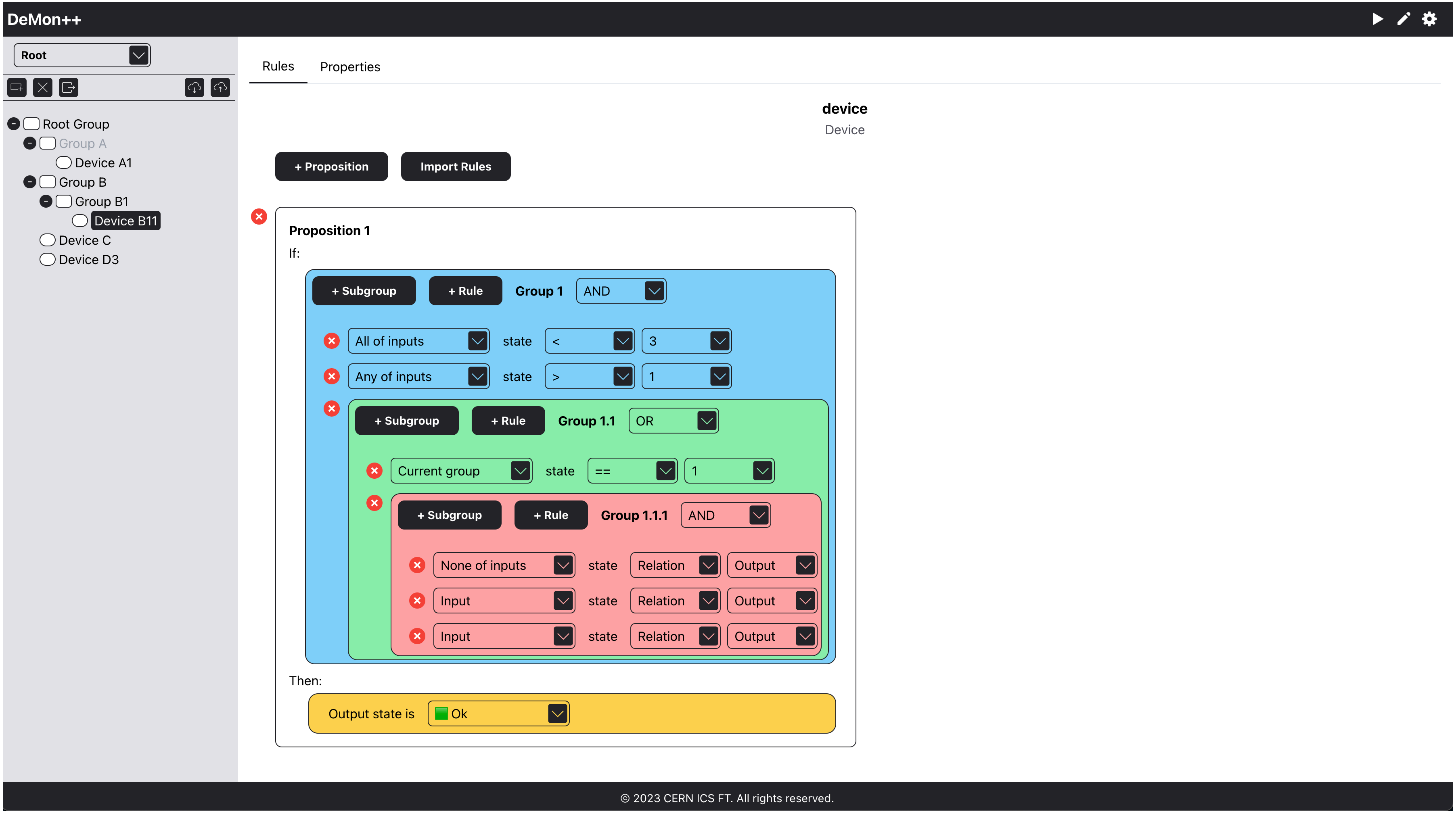
Data visualization

- We developed an application for real-time visualization of system health
- Based on finite state machine concepts
- Tree representation to capture the hierarchical relationship between devices
- Intuitive navigation, enabling operators to locate errors quickly

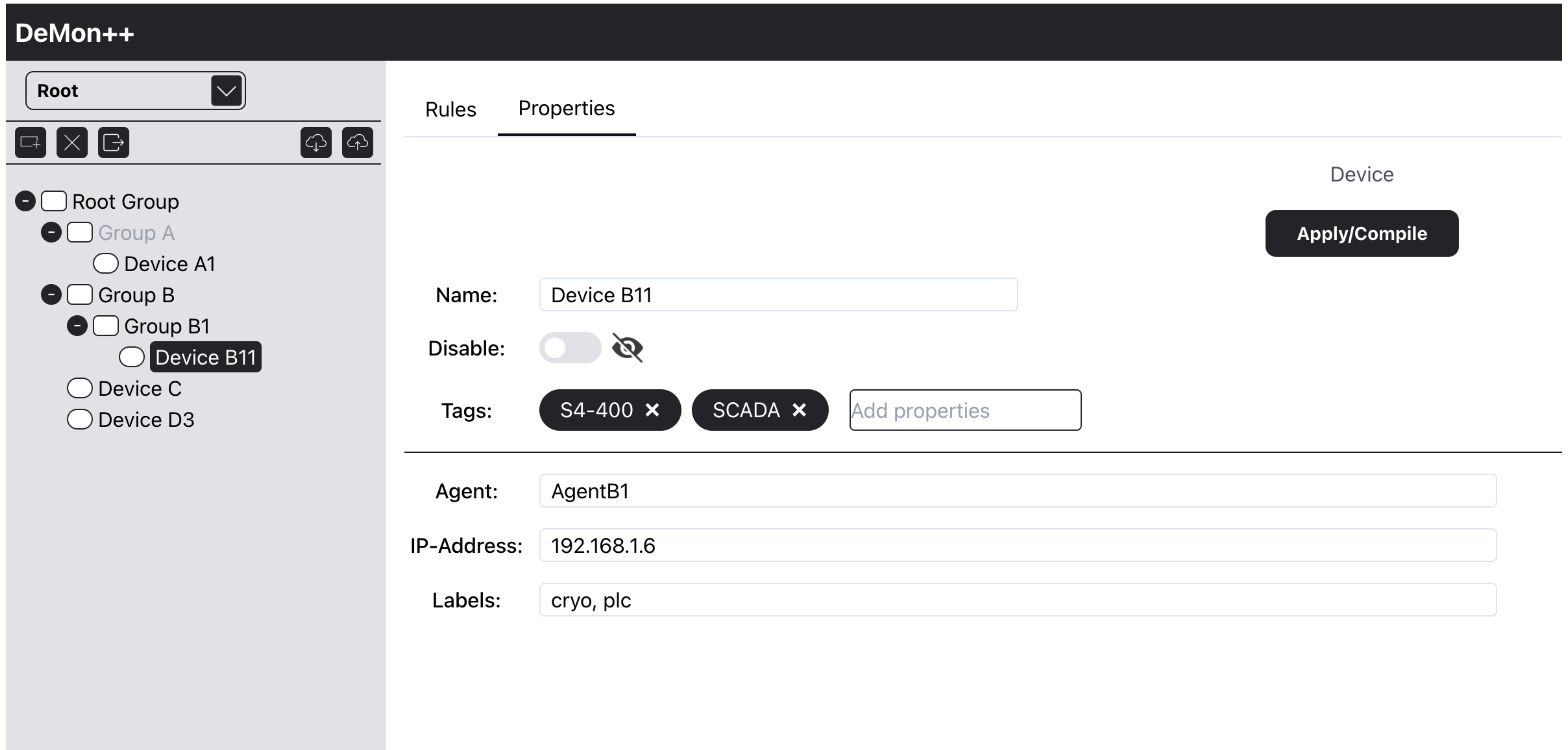
Hierarchical Finite State Machine



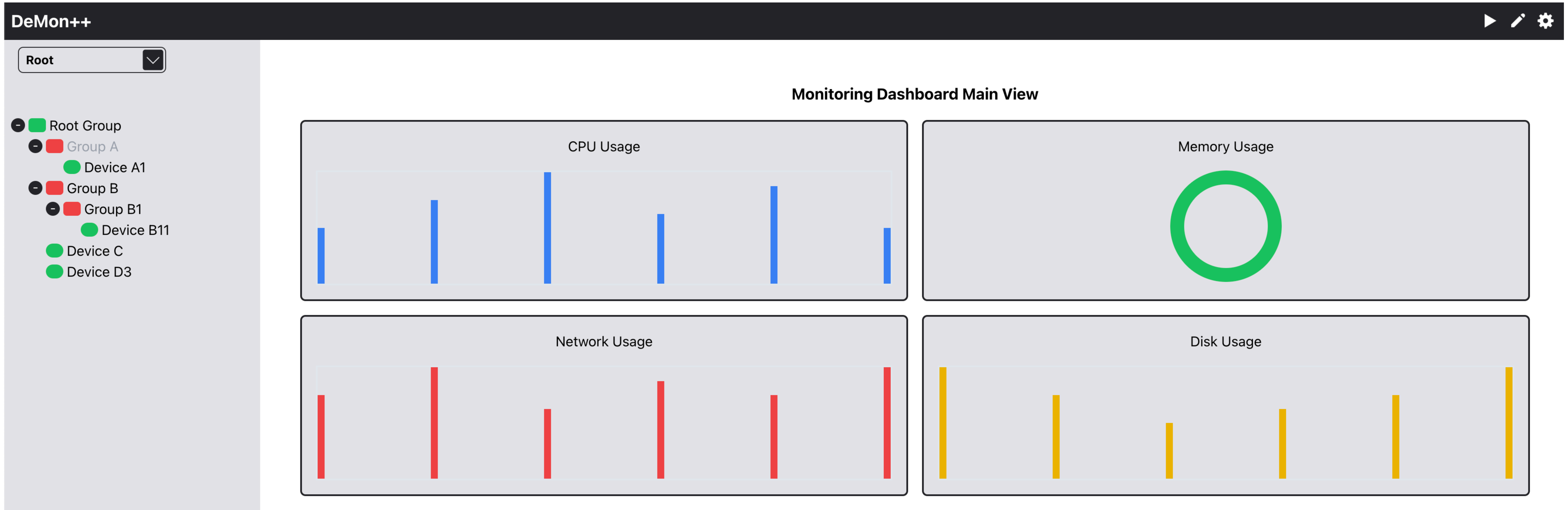
Hierarchical visualization to summarise system health



User-interface to configure the device tree



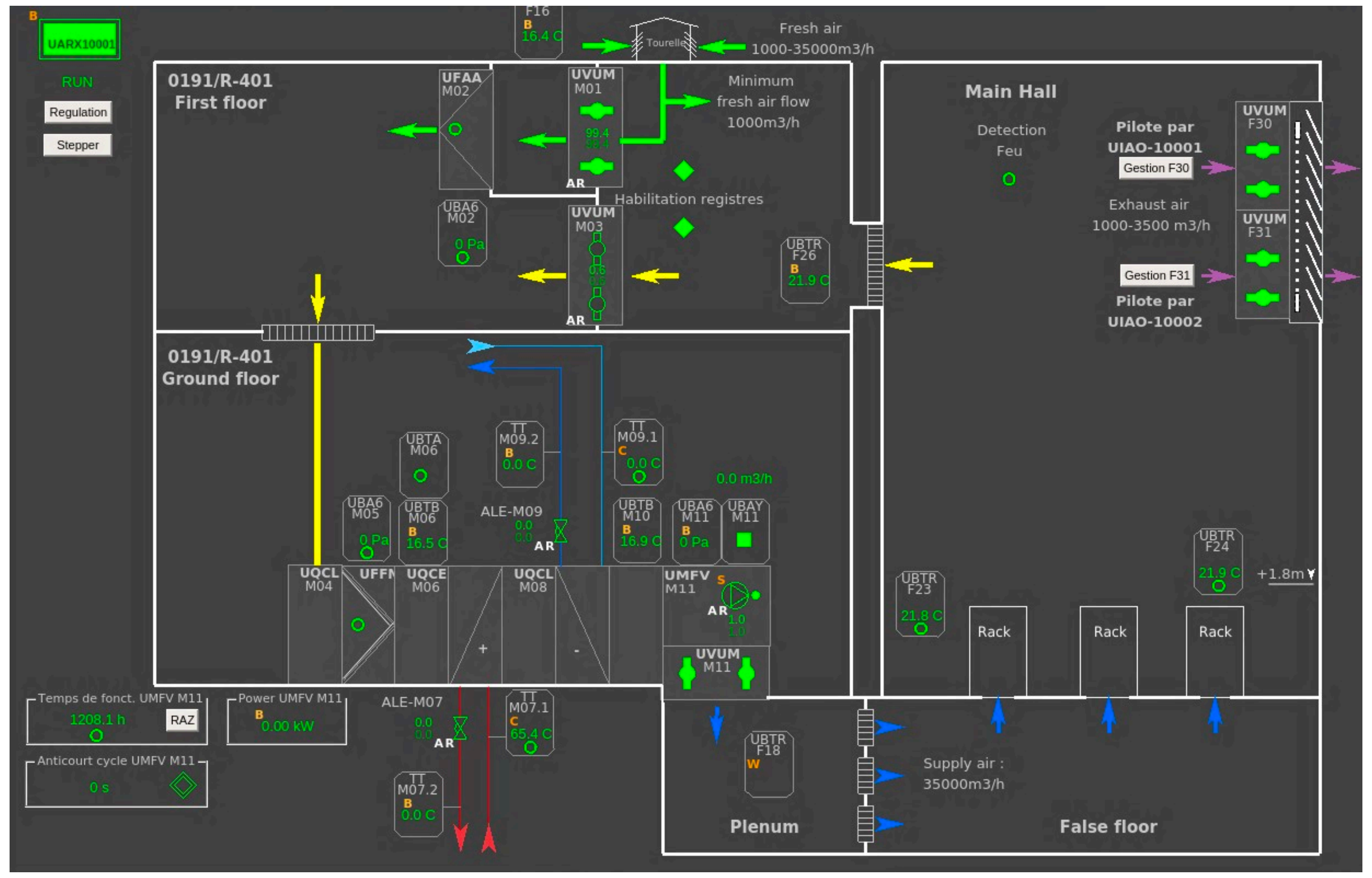
Parameters configured to interact with Siemens monitoring solutions

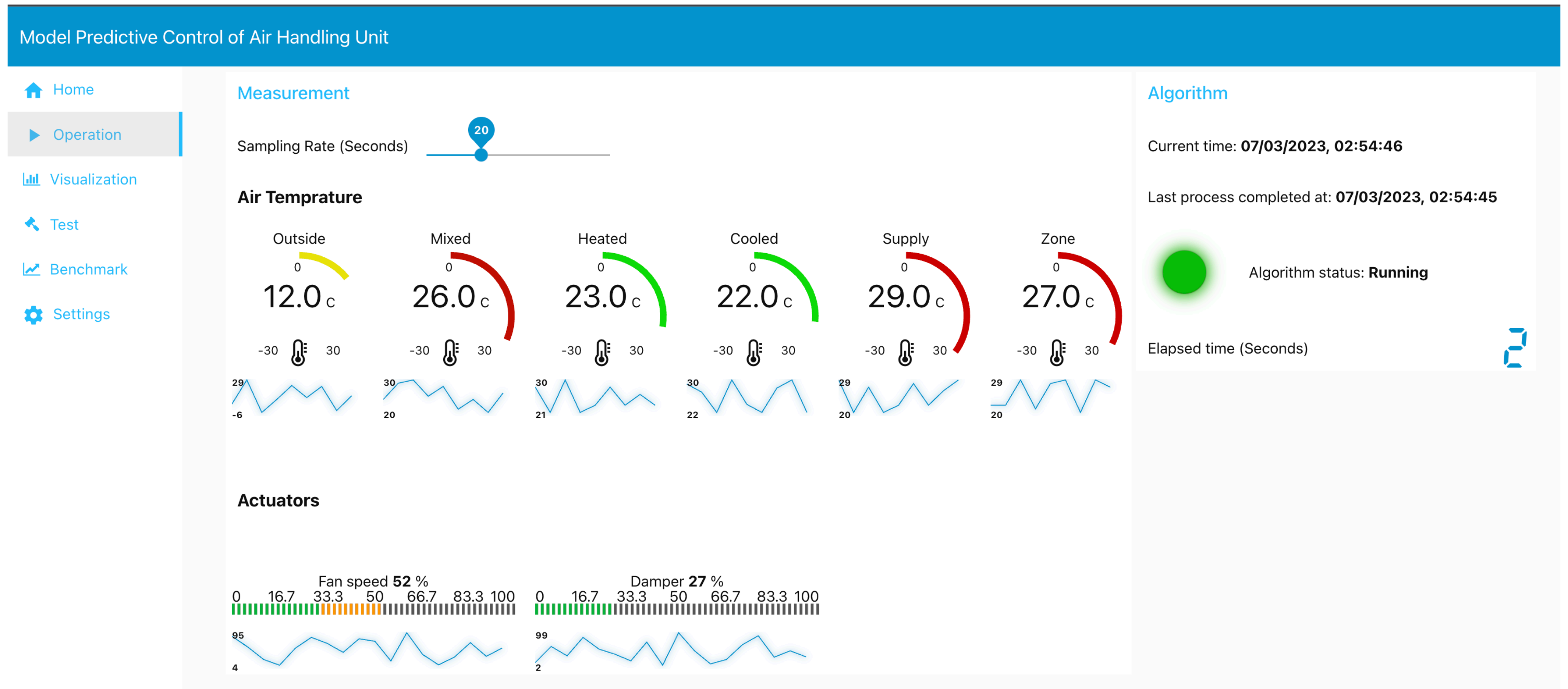


Main-view that runs the logic that updates visualization in real-time

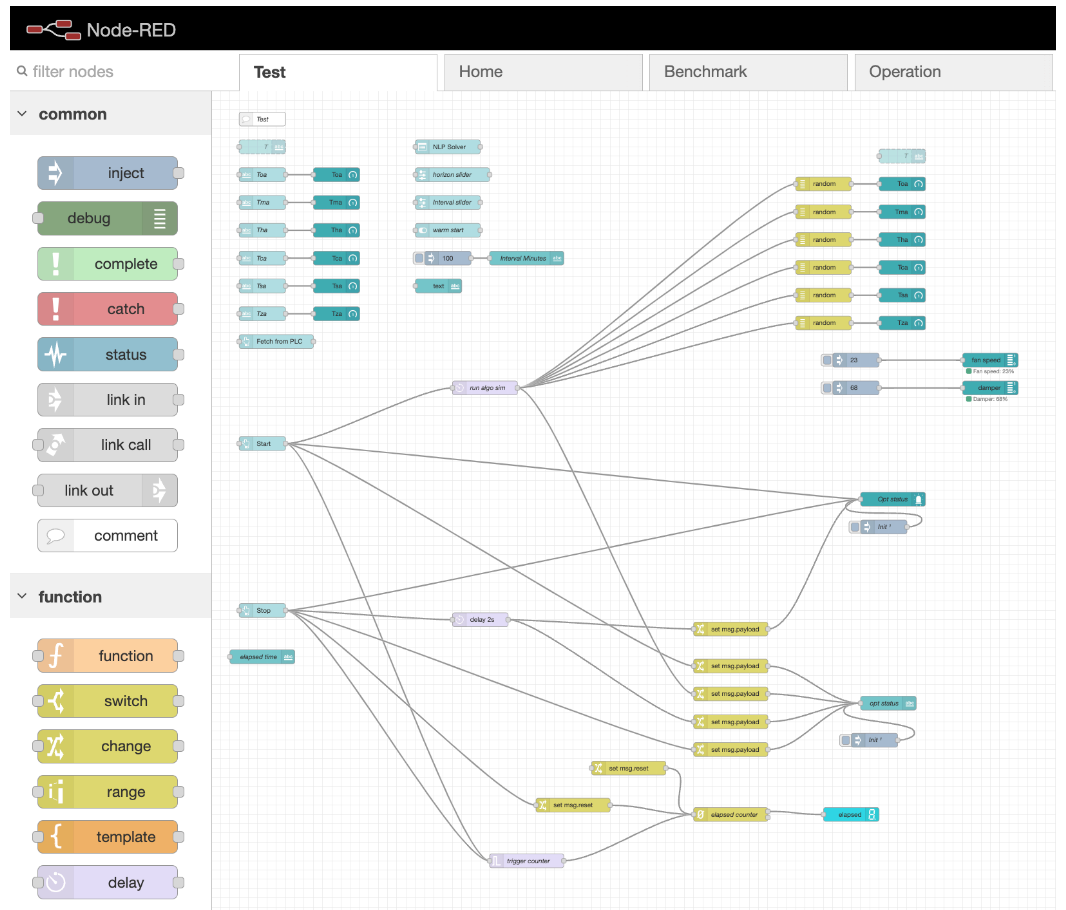
Next steps

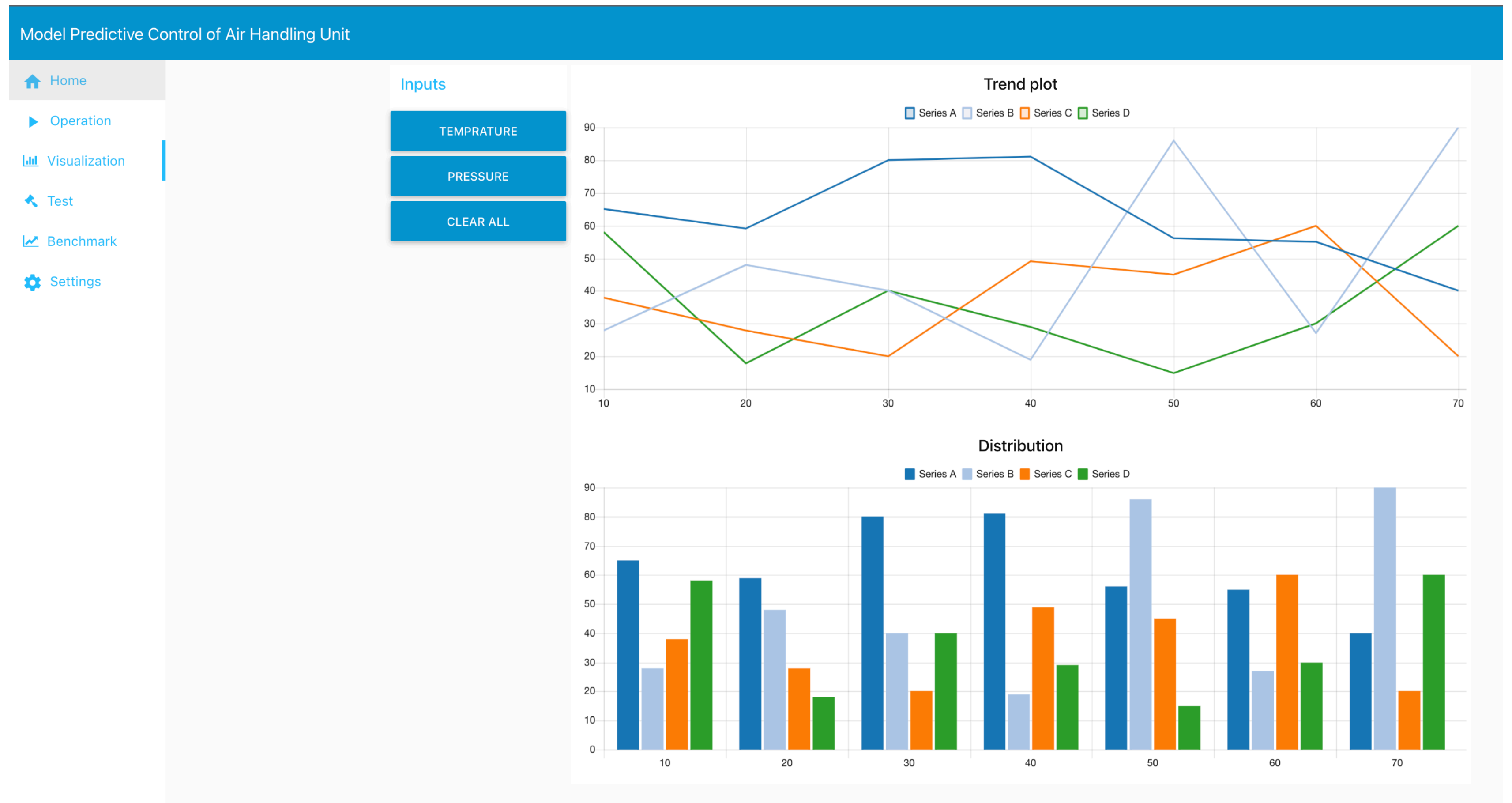
- Explore officially supported communication drivers and libraries that allow retrieval of diagnostic information
- Build tools around it and integrate it with the visualization tool
- Collaboratively work with product owners to add or modify features of the existing Siemens monitoring solutions





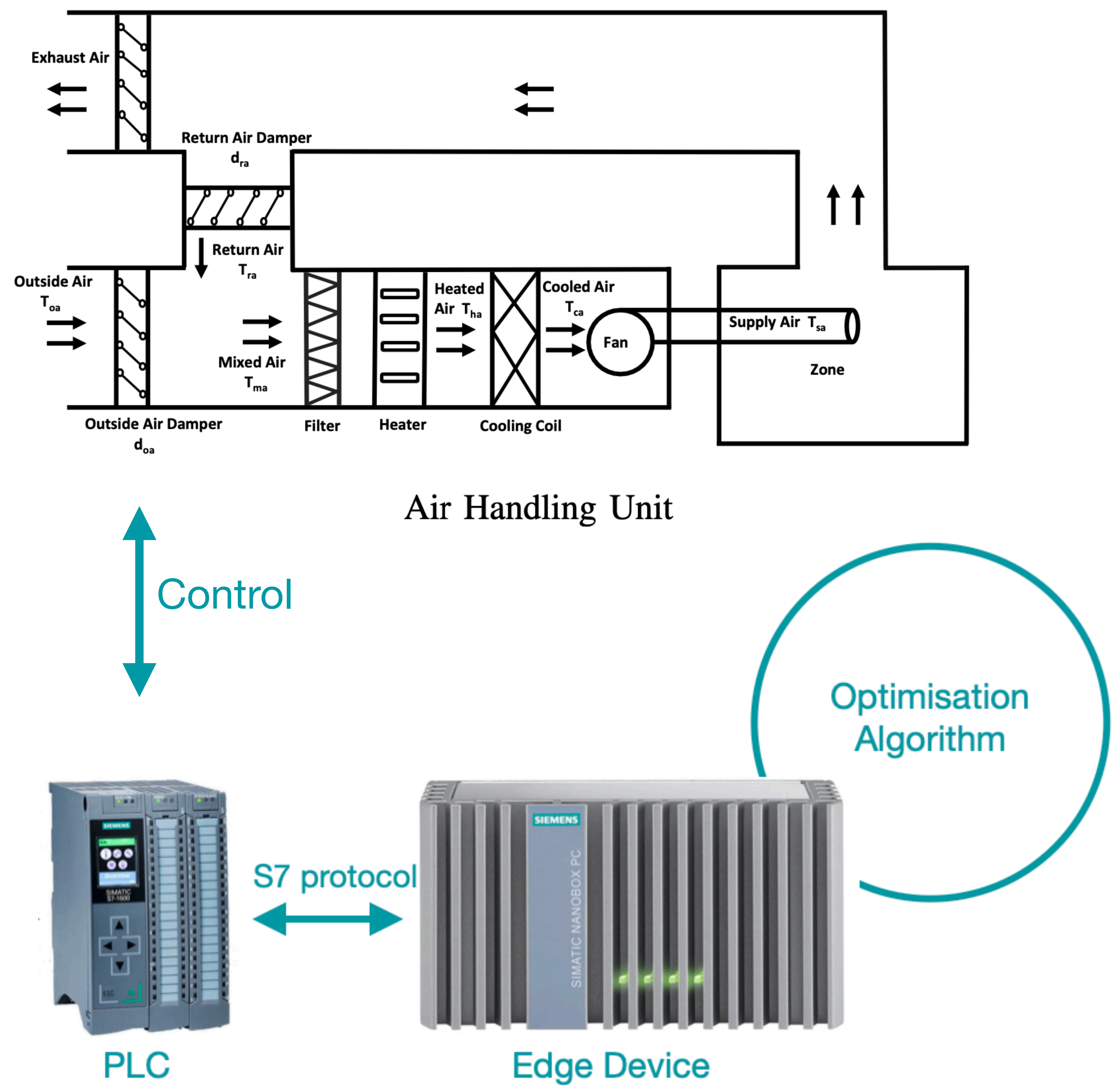
Edge Computing





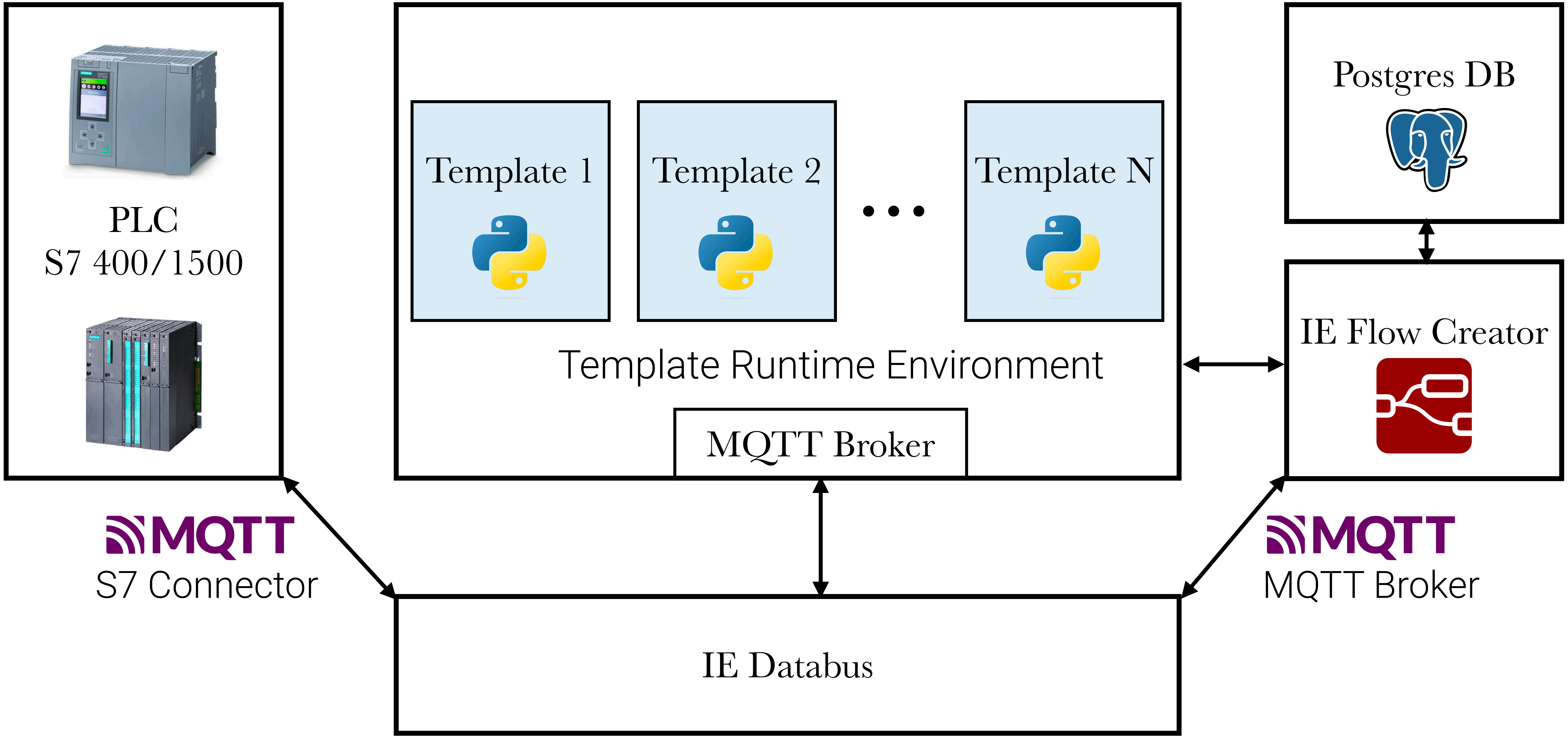
Use case: Optimising HVAC systems

- Advanced control algorithms can optimize HVAC processes to reduce energy consumption
- Model Predictive Control for Air Handling Units - [F Ghawash et al. 2022]
- Previous setup: Run the algorithm on SCADA servers and send control signals to PLC
- Proposed Upgrade: Develop, Run, and Test the algorithm on an Edge device



Edge device to deploy MPC algorithms for HVAC

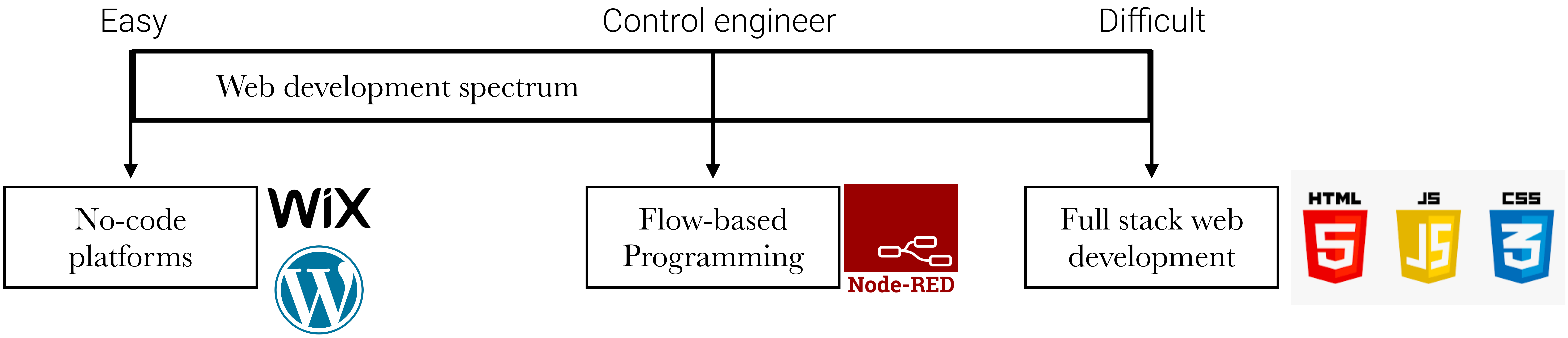
DevOps on Edge

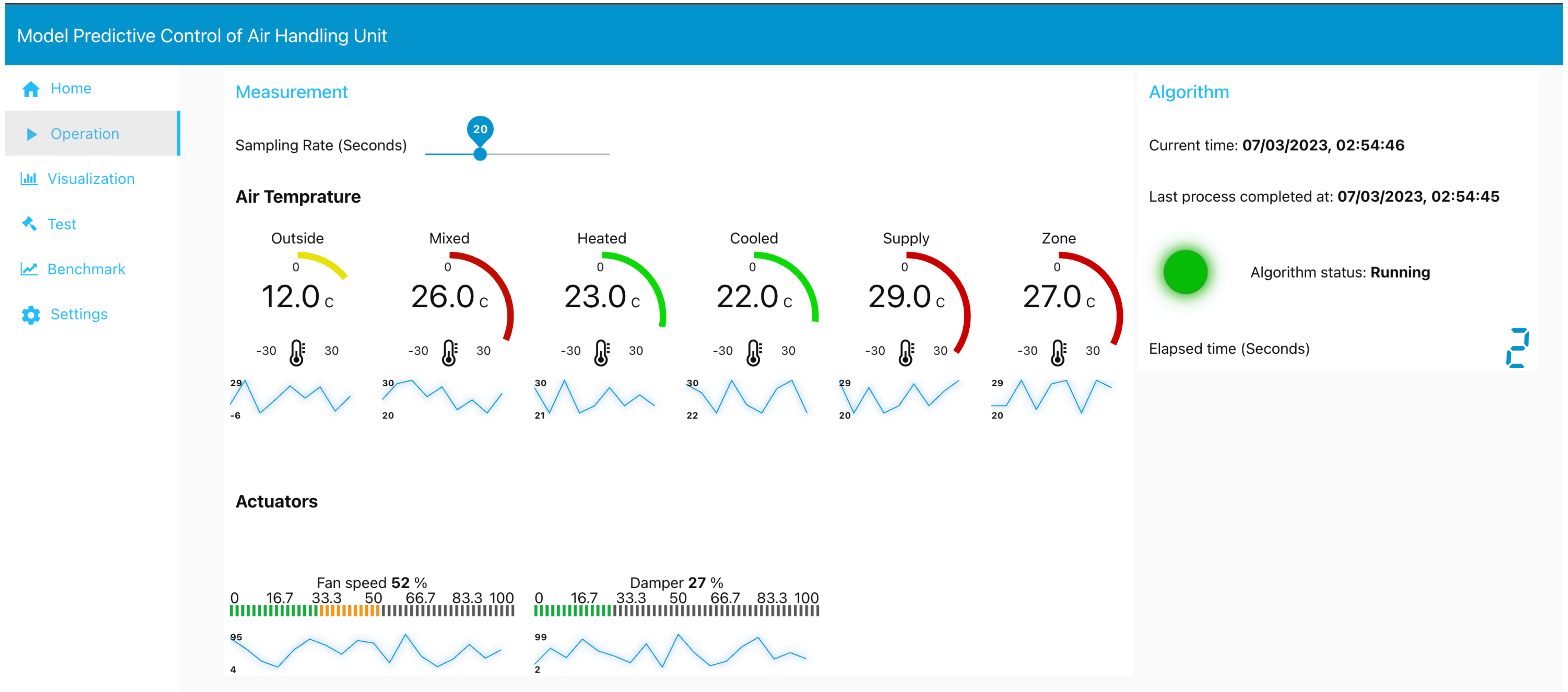


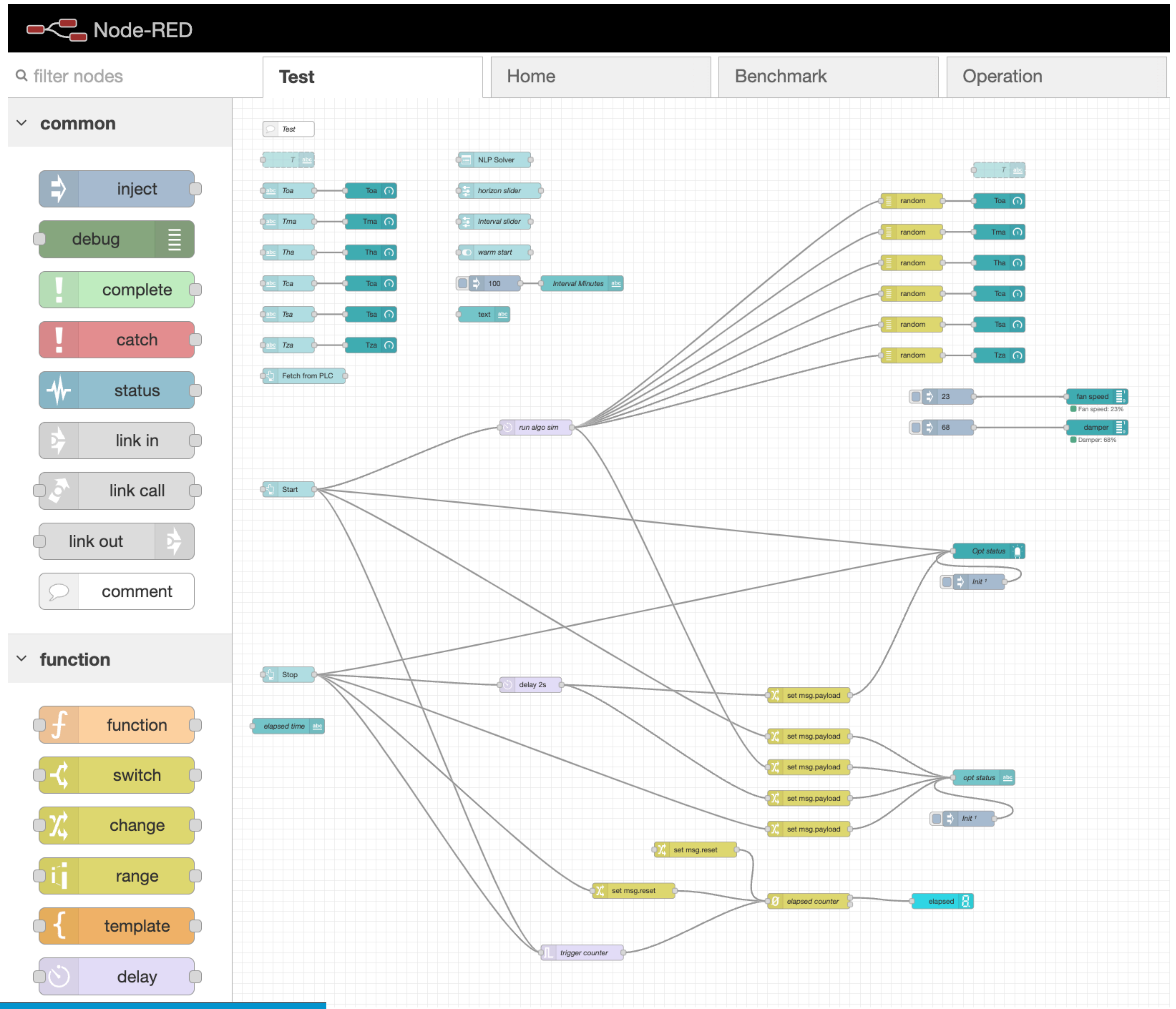
The architecture of 'Template Framework' - A DevOps platform for Edge

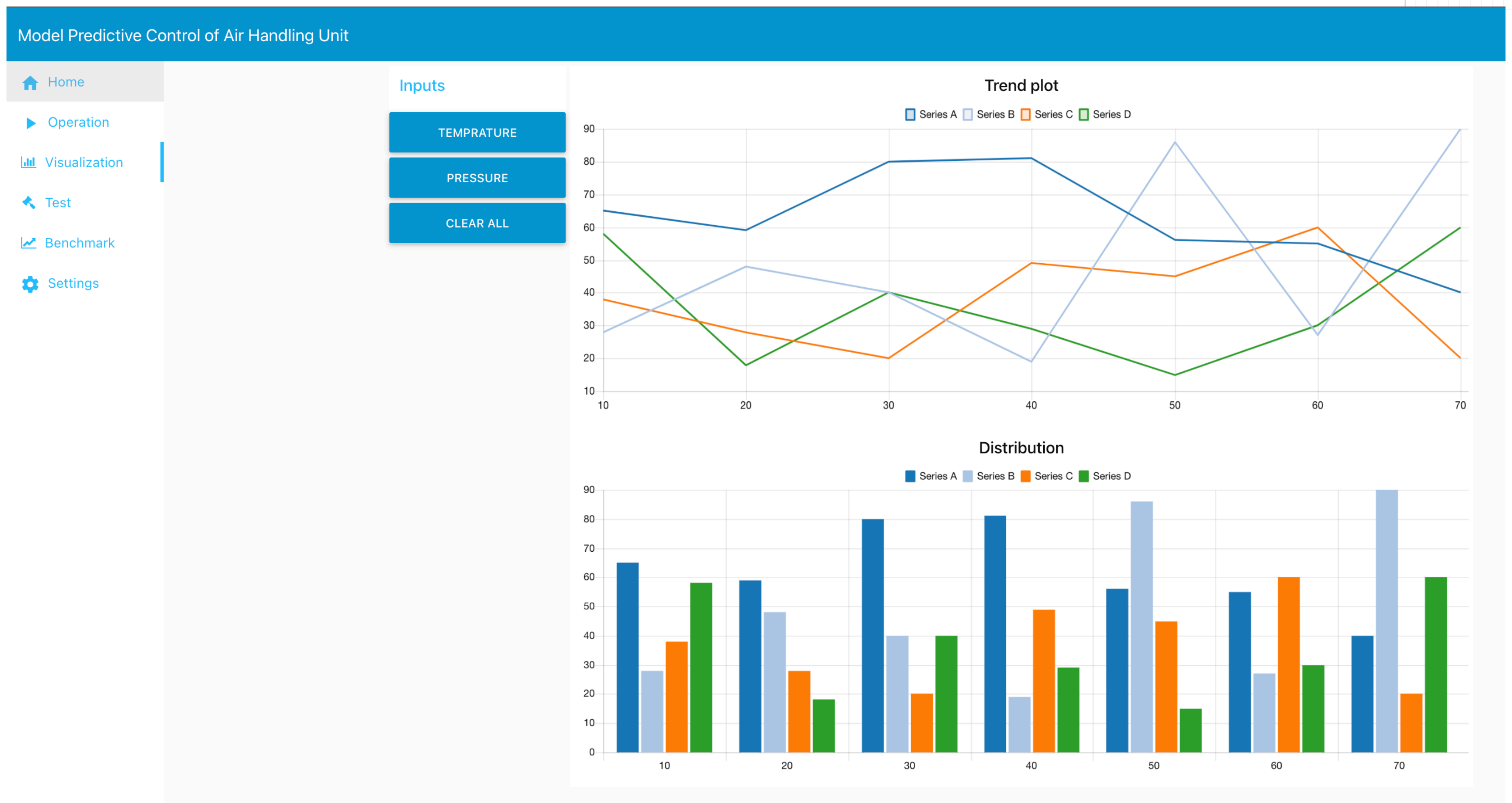
HMI on Edge

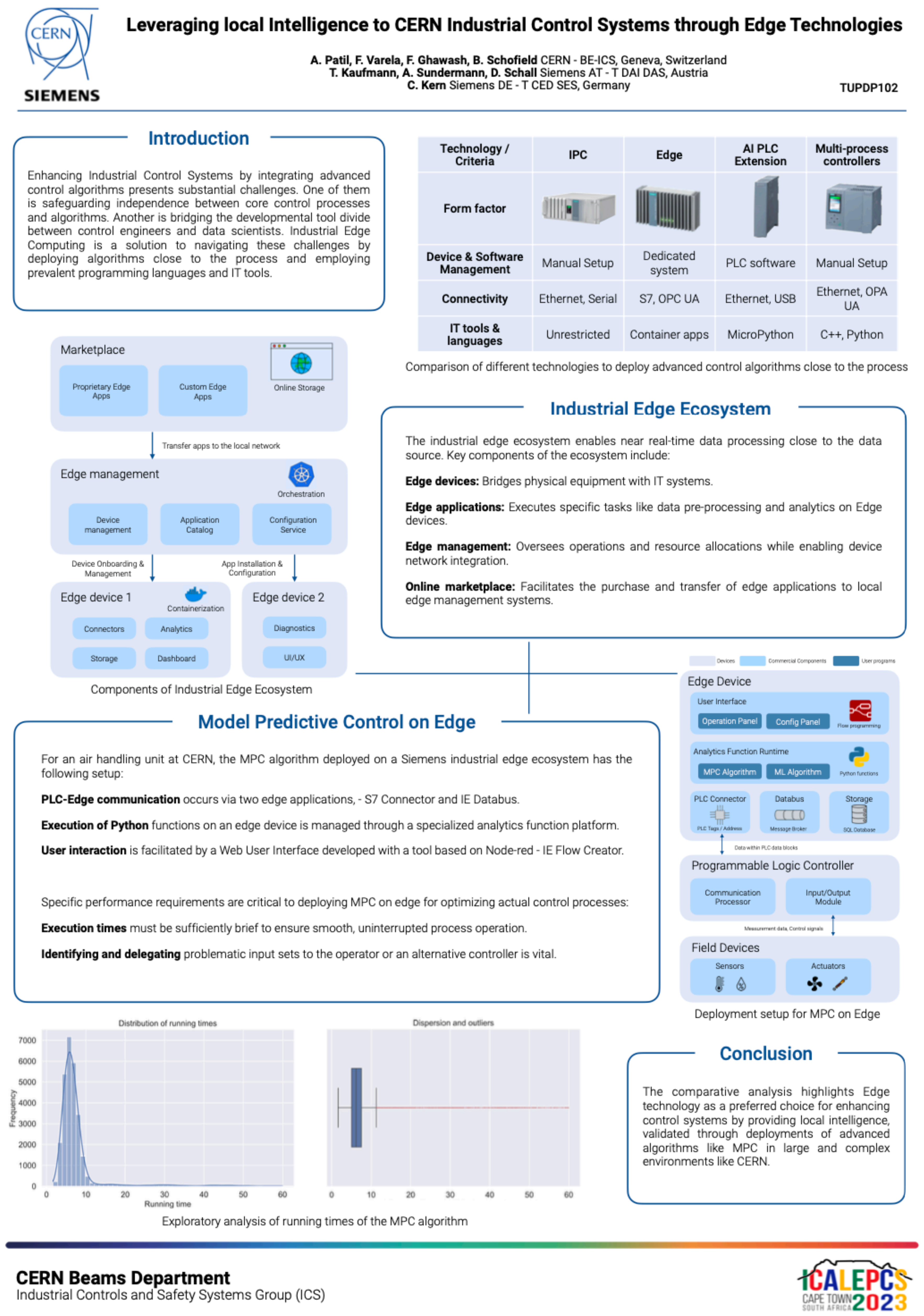
- Web application to interact with the MPC algorithm
- Developed with another Edge application: IE flow creator
  - Based on flow-based programming tool: Node-red
- The application provides node.js components and angular.js templates to build a full-stack web application with less effort




Model Predictive Control of Air Handling Unit
Home
Operation
Visualization
Test
Benchmark
Settings
Measurement
Sampling Rate (Seconds)
20
Air Temprature
Outside
Mixed
Heated
Cooled
Supply
Zone
12.0 c
26.0 c
23.0 c
22.0 c
29.0 c
27.0 c
Actuators
Fan speed 52 %
Damper 27 %
Algorithm
Current time: 07/03/2023, 02:54:46
Last process completed at: 07/03/2023, 02:54:45
Algorithm status: Running
Elapsed time (Seconds)


Node-RED
filter nodes
Test
Home
Benchmark
Operation
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
template
delay


Model Predictive Control of Air Handling Unit
Home
Operation
Visualization
Test
Benchmark
Settings
Inputs
TEMPRATURE
PRESSURE
CLEAR ALL
Trend plot
Series A
Series B
Series C
Series D
Distribution



Poster

LEVERAGING LOCAL INTELLIGENCE TO CERN INDUSTRIAL CONTROL SYSTEMS THROUGH EDGE TECHNOLOGIES

A. Patil, F. Varela, F. Ghawash, B. Schofield, CERN, Geneva, Switzerland,
T. Kaufmann, A. Sundermann, D. Schall, Siemens AT - T DAI DAS, Austria,
C. Kern, Siemens DE - T CED SES, Germany

Abstract

Industrial processes often use advanced control algorithms such as Model Predictive Control (MPC) and Machine Learning (ML) to improve performance and efficiency. However, deploying these algorithms can be challenging, particularly when they require significant computational resources and involve complex communication protocols between different control system components. To address these challenges, we showcase an approach leveraging industrial edge technologies to deploy such algorithms. An edge device is a compact and powerful computing device placed at the network's edge, close to the process control. It executes the algorithms without extensive communication with other control system components, thus reducing latency and load on the central control system. We also employ an analytics function platform to manage the life cycle of the algorithms, including modifications and replacements, without disrupting the industrial process.

Furthermore, we demonstrate a use case where an MPC algorithm is run on an edge device to control a Heating, Ventilation, and Air Conditioning (HVAC) system. An edge device running the algorithm can analyze data from temperature sensors, perform complex calculations, and adjust the operation of the HVAC system accordingly. In summary, our approach of utilizing edge technologies enables us to overcome the limitations of traditional approaches to deploying advanced control algorithms in industrial settings, providing more intelligent and efficient control of industrial processes.

INTRODUCTION

The latest advances in AI and ML, along with time-tested methods like MPC, offer new ways to enhance the functionality of industrial control systems [1]. For instance, these techniques can improve system reliability through anomaly detection, enable energy-efficient operation of complex industrial processes, and extend equipment life through predictive maintenance. Nevertheless, enhancing industrial control systems through such techniques poses several challenges.

One significant challenge is ensuring that the core processes of the system and the algorithms operate independently and do not interfere with one another. This process independence ensures that the demands of complex algorithms do not jeopardize the safe operation of the core process and overburden its resources. Also, deploying complex algorithms on the existing control infrastructure may only be possible if specialized hardware components like GPUs or AI processors are available. These components were relatively uncommon in industrial control setups until recently. However, new control hardware, such as multi-processor PLCs and AI expansion cards, have emerged, making this deployment possible.

Another challenge is the notable disparities between the focus areas of control engineers and data scientists when devising control systems. Control engineers primarily concentrate on industrial communication protocols, control devices, PLC programming, and SCADA development. In contrast, data scientists and software engineers focus on creating new control strategies using Python or C++ and utilize software development tools like package managers and containers. New computing paradigms tailored to industrial control systems have been developed that bridge this divide and integrate information technology (IT) tools into operational technology (OT). Examples include integrating control systems with Cloud computing, High-Performance Computing (HPC), and Edge computing.

This article mainly focuses on solutions that address these challenges and provide local intelligence to a control system, i.e., intelligence close to the process, allowing faster analysis of streamed data and lightening the load on the different layers of the control system by reducing network latency and traffic. We start by comparing various techniques for leveraging local intelligence. We emphasize Industrial Edge Computing as an emerging solution that provides benefits such as separation of concern, simplification of algorithm development, and easy application lifecycle management. Finally, we will share insights from implementing an advanced optimization algorithm on state-of-the-art edge technologies and validating its use in a real-world setting at CERN.

LEVERAGING INTELLIGENCE TO INDUSTRIAL CONTROL SYSTEMS

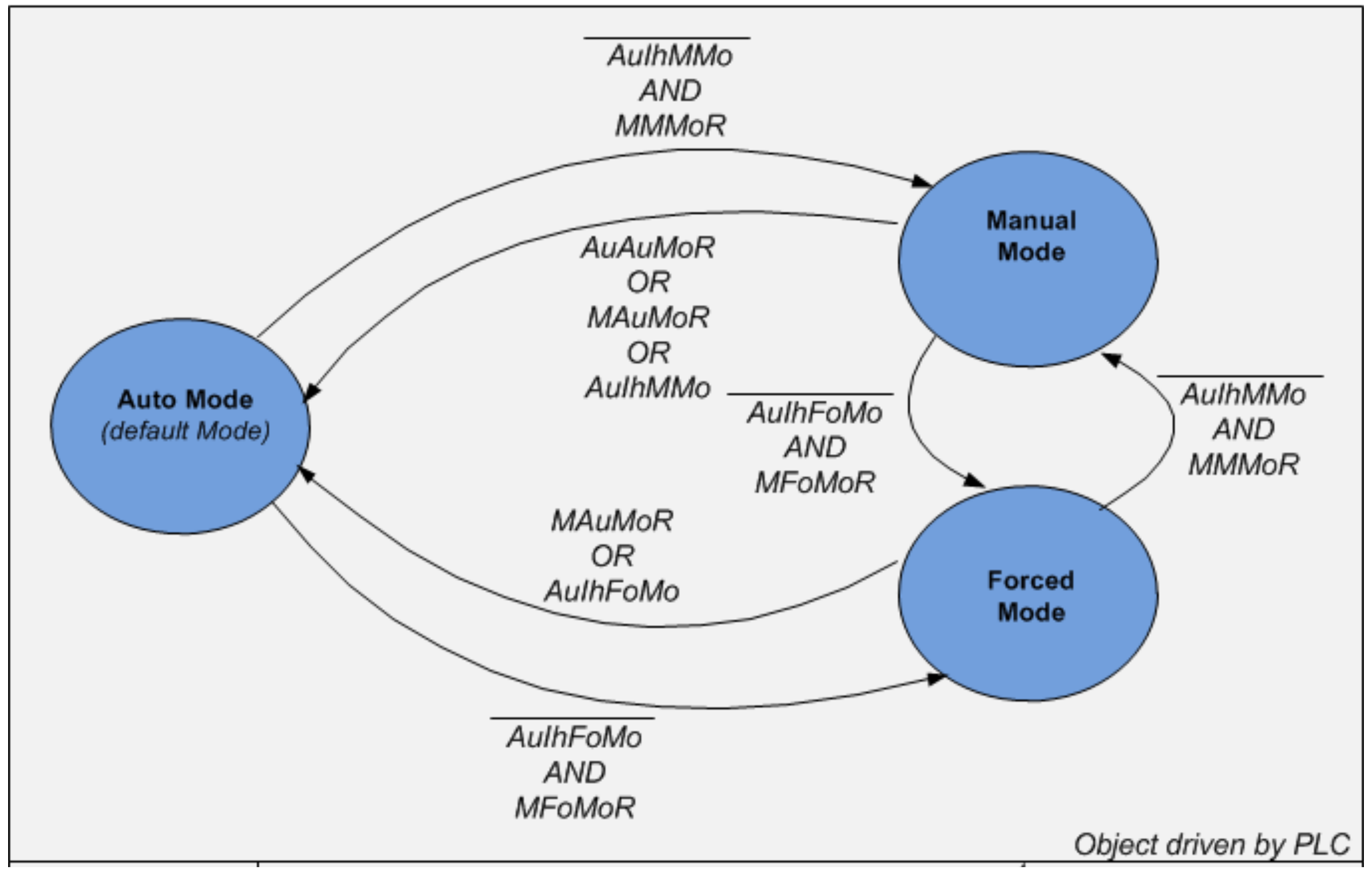
In recent years, the capability to deliver intelligence close to industrial processes has progressed from conventional setups such as bare-metal Industrial PCs to more advanced systems like multi-process controllers and edge technologies. Some of these setups are outlined below.

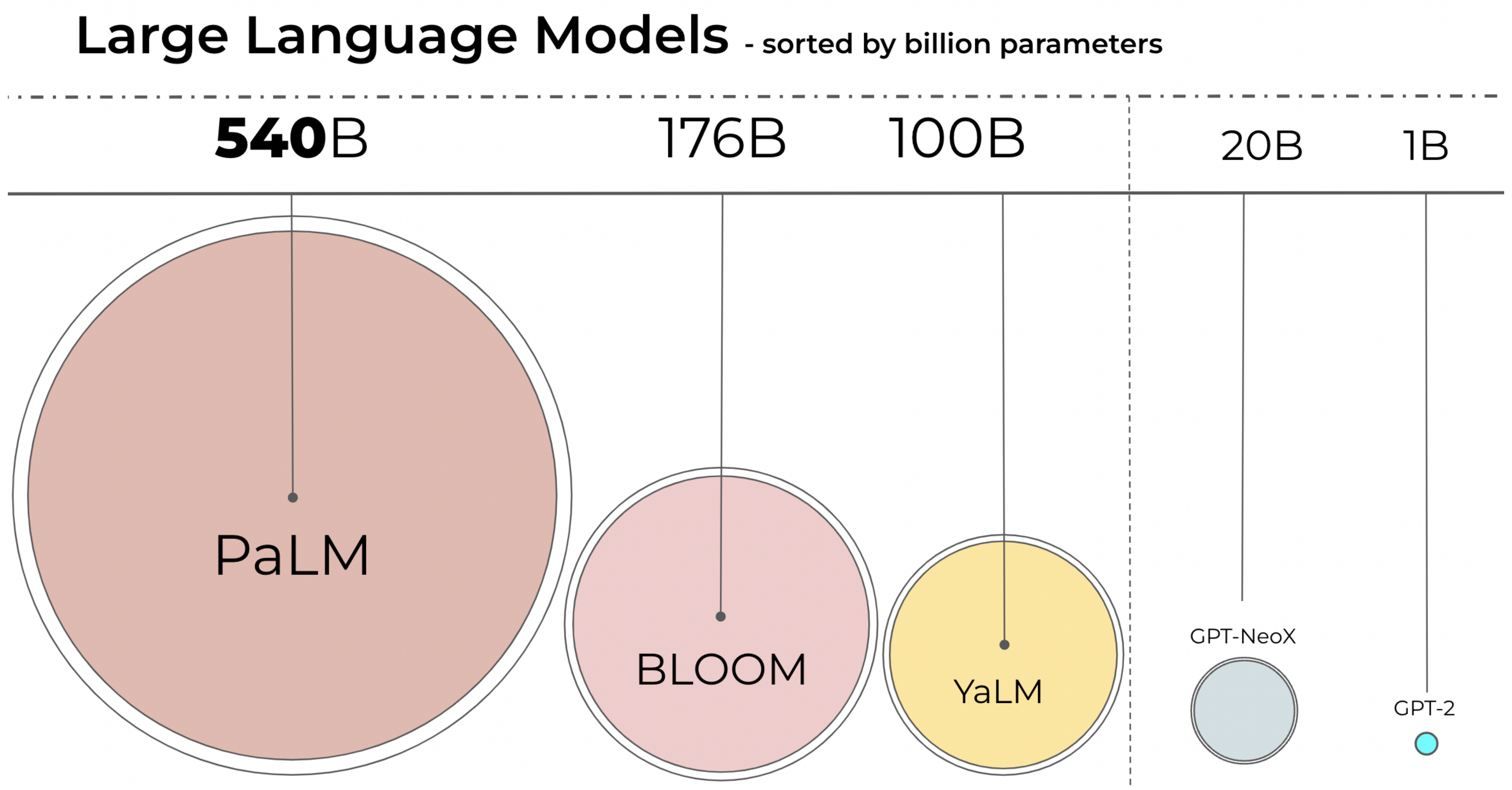
- **Multi-process controllers:** PLC vendors have acknowledged developers' needs to program control components in languages beyond the IEC 61131-3 standard [2], adopting higher-level languages like C++ and Python. For instance, the Siemens S7-1518 Multi-Functional Platform (MFP) has a Linux OS alongside its standard PLC OS that primarily supports C++. Communication between the OSs is via an Ethernet virtual switch, eliminating additional hardware and separating

Paper

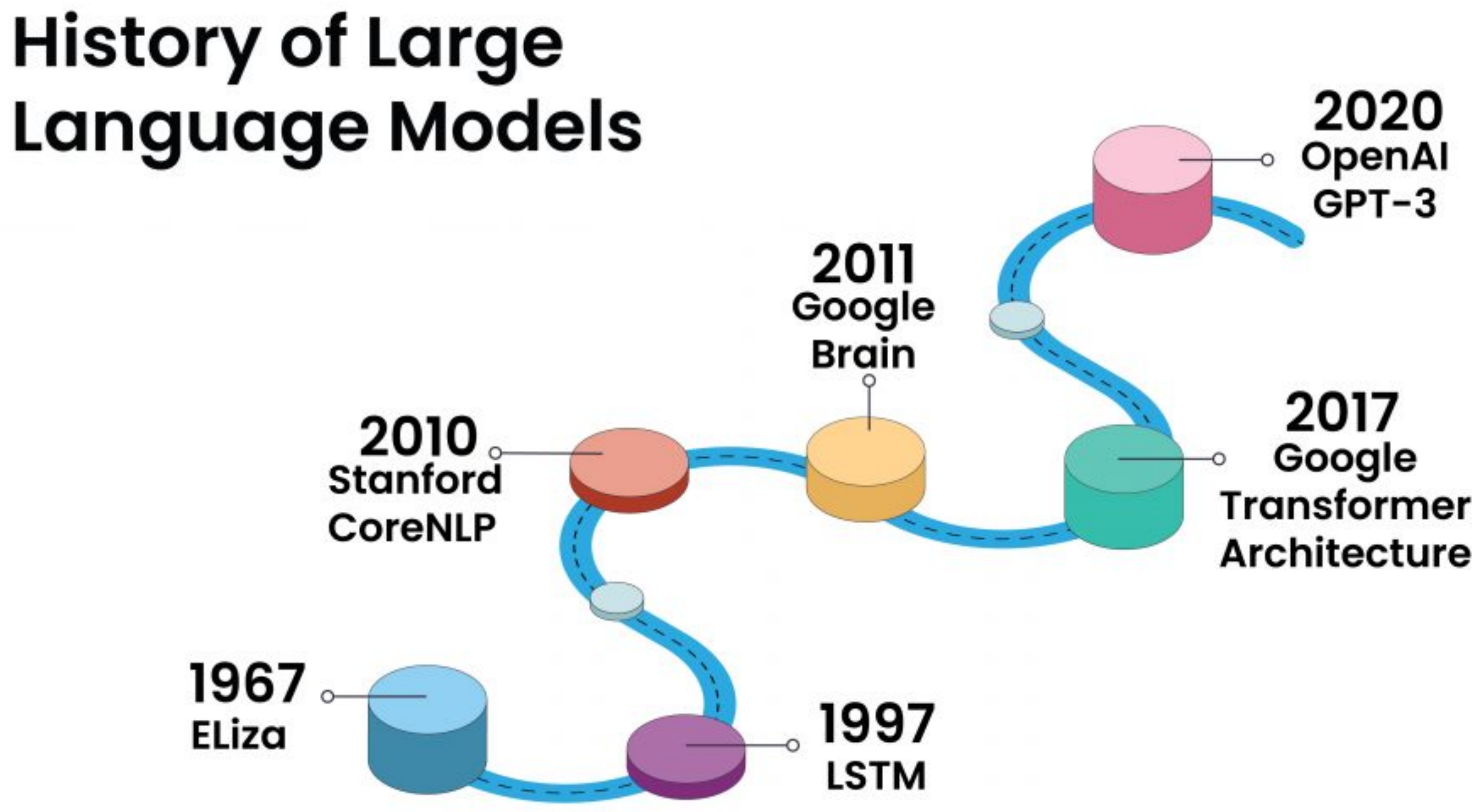
Conclusion

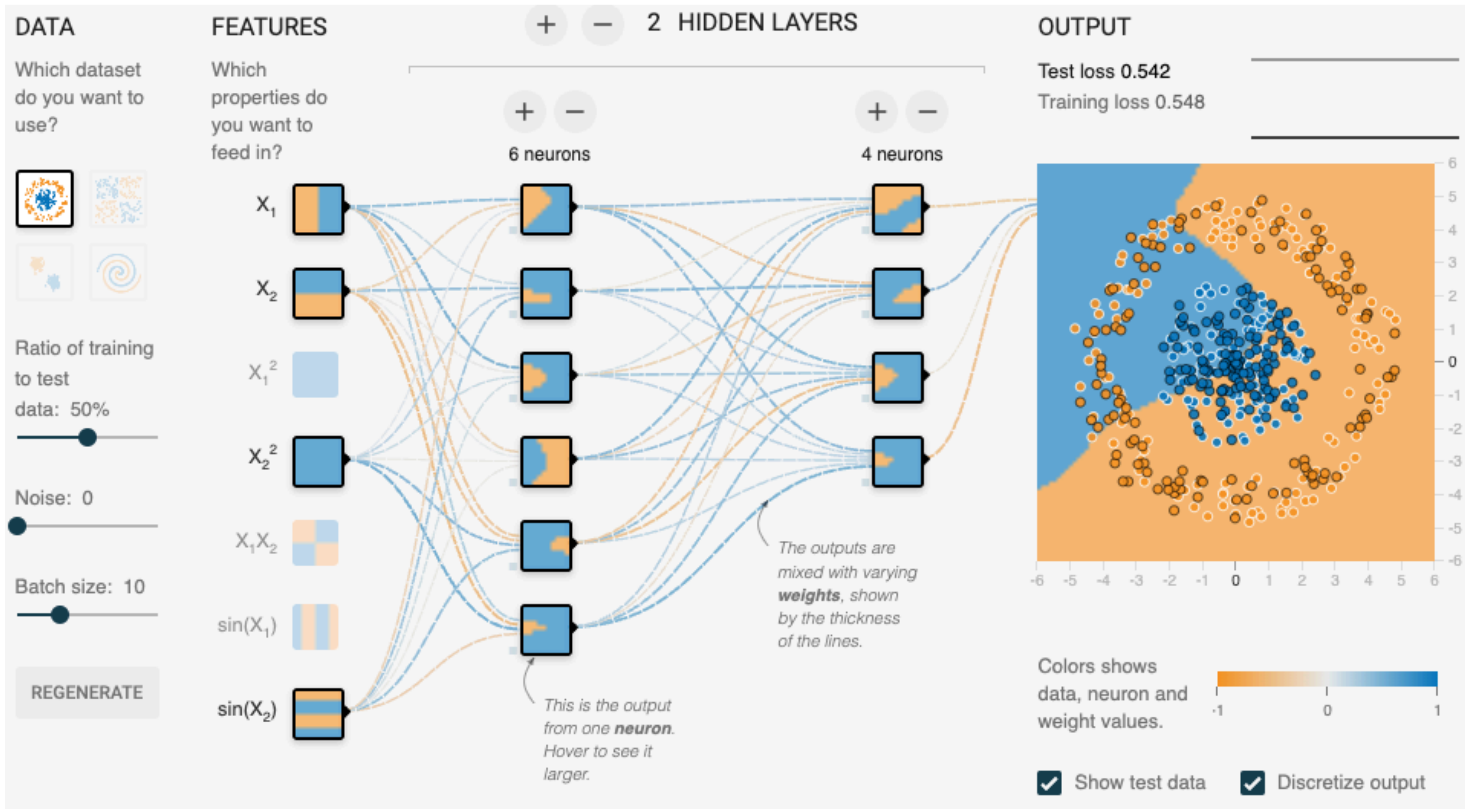
- Edge ecosystem consists of Drivers, Applications, and Runtime environments to augment a PLC with complex algorithms
- Supports various industrial communication protocols
- Advanced control algorithms can be deployed using a specific DevOps platform
- Web applications using flow-based programming can be developed to provide a user interface





Industrial Co-Pilot





Background

- The recent rise of Large Language Models with billions of parameters
- It can be leveraged as a tool for industrial control code development
- Many popular LLMs are proprietary and require a license
- Parallel movement in the LLM space: open source LLMs
    - Llama 2, BLOOM, etc.
- Possibility for more flexibility and customization and transparency

Objective

- Initiate the development of a tool to generate industrial control code
    - WinCC OA Control Code for the SCADA layer
    - PLC Code for the Control layer
- Explore the application of LLMs in generating code documentation
- Build a benchmarking suite to test the quality of various open-source LLMs

Thank you