Opening the floor to PLCs and IPCs : CODESYS in UNICOS.

E. Blanco Vinuela, M. Koutli, T. Petrou, J. Rochez (CERN, Engineering Department, Geneva, Switzerland)

ABSTRACT

The CERN UNICOS (UNified Industrial COntrol System) framework is widely used in many process control domains (e.g. Cryogenics, Cooling, Gas Systems, Ventilation, Vacuum…) to produce highly structured standardized control applications for the two CERN approved industrial PLC product families, Siemens and Schneider. The CODESYS software suite, developed by the 3S (Smart Software Solution), provides an independent **IEC 61131-3** programming environment for industrial controllers. The complete CODESYS based development includes:

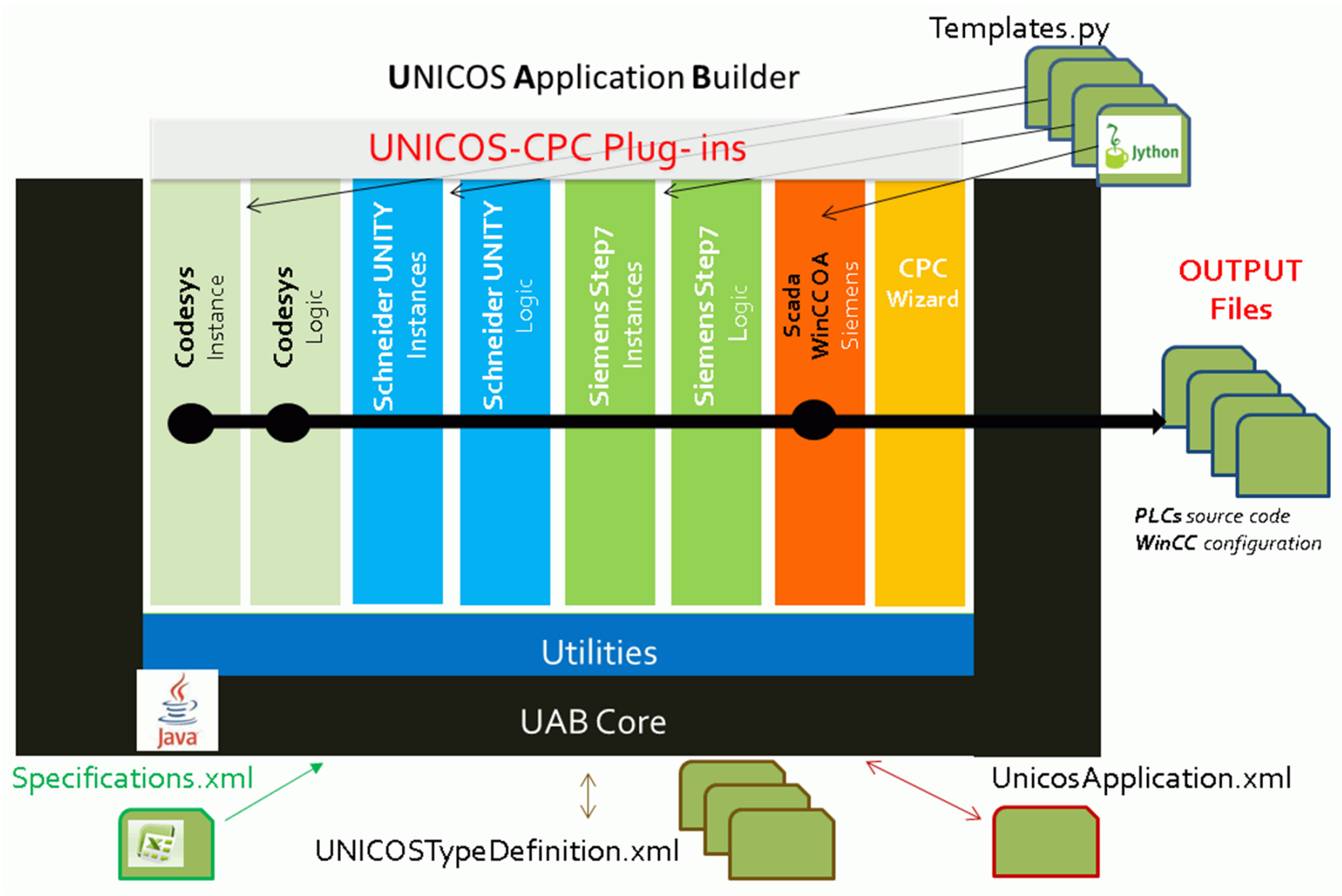
(1) a dedicated Java™ module plugged in an automatic code generation tool, the UAB (UNICOS Application Builder),

(2) the associated UNICOS baseline library for industrial PLCs and IPCs (Industrial PC) **CODESYS v3** compliant,

(3) the Jython-based templates to deploy device instances and control logic.

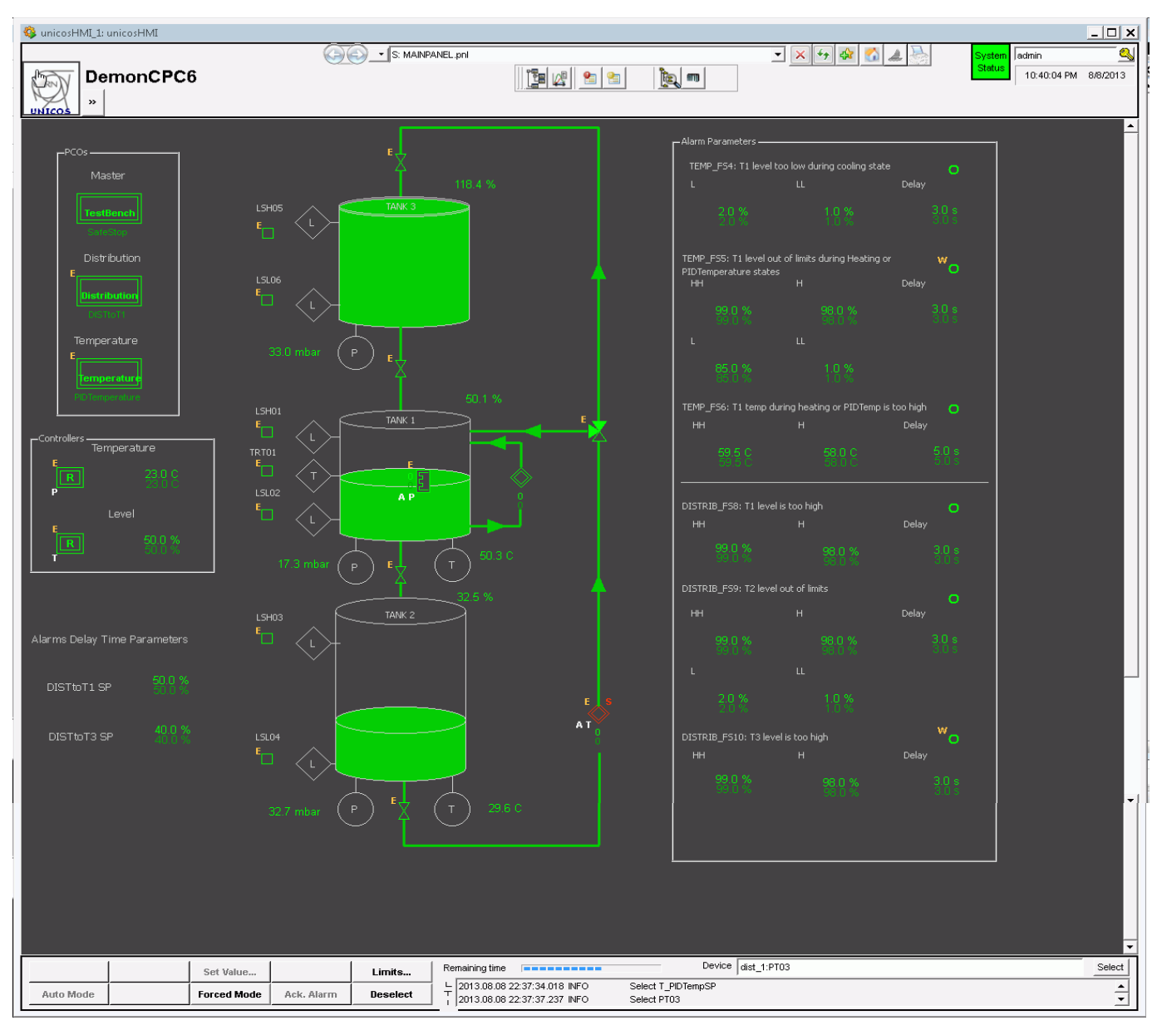
The availability of this development opens the UNICOS framework to a wider community of industrial PLC manufacturers (e.g. Beckhoff, ABB, WAGO…) and, as the CODESYS control Runtime works in standard Operating Systems (Linux, W7…), UNICOS could be deployed to any IPC.

GENERATION PROCESS USING UNICOS AUTOMATIC GENERATION TOOLS



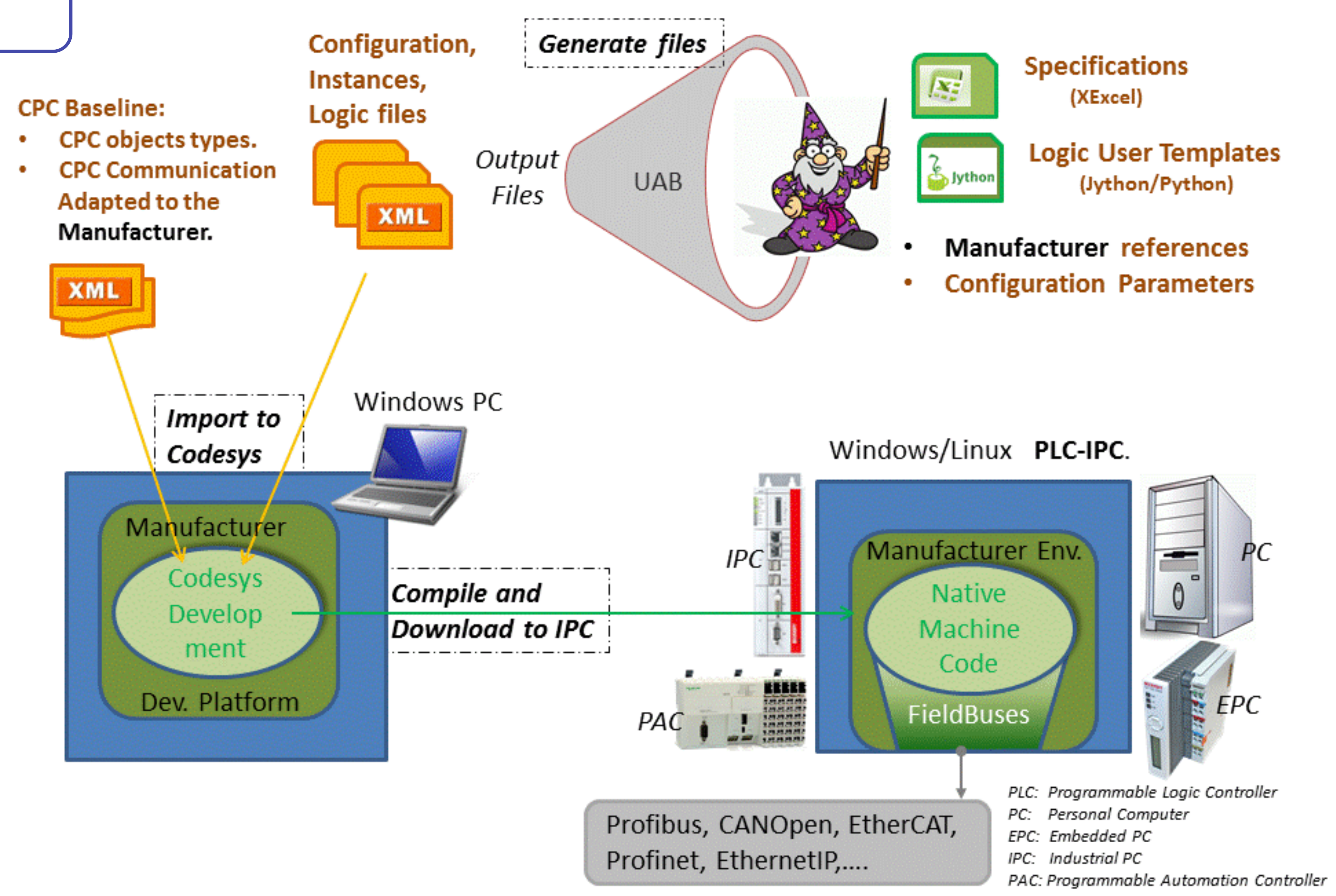
UAB architecture
Only represented the plug-ins related to CPC (Continuous Control Process)

The **plug-ins** are developed in **Java™**, while the **Templates** use a scripting language: ***Jython***. The format of the **specification** file is based on ***xls/xml***.
The format of the first four **output** files is **PLCopen XML**. These files are afterwards imported in the CODESYS integrated software. The latter file format is text-based and will be imported in the WinCC OA SCADA system.



SCADA : Screenshot of a WinCC OA synoptic

Codesys Control Process generation path under UNICOS CPC



CPC Baseline:

- CPC objects types.
- CPC Communication Adapted to the Manufacturer.

Configuration, Instances, Logic files

XML

Generate files

Output Files

UAB

Specifications (XExcel)

Logic User Templates (Jython/Python)

- Manufacturer references
- Configuration Parameters

Import to Codesys

Windows PC

Manufacturer

Codesys Development

Dev. Platform

Compile and Download to IPC

Windows/Linux **PLC-IPC**.

IPC

PAC

Manufacturer Env.

Native Machine Code

FieldBuses

PC

EPC

Profibus, CANOpen, EtherCAT, Profinet, EthernetIP,….

PLC: Programmable Logic Controller
PC: Personal Computer
EPC: Embedded PC
IPC: Industrial PC
PAC: Programmable Automation Controller

Inputs to the UAB CODESYS

- The **specification file** based on the **functional analysis** of the plant and containing all the instances of the CPC objects.
- The **Logic User Templates** that define the **specific process control logic**. A default template is provided to the user as a base, but the process control engineer can develop his own.
- The **configuration parameters**, which should be filled in the wizard to define the PLC-SCADA **communication** parameters and the **mapping** of the PLC memory according to the needs of the project.

Outputs of the UAB CODESYS and Inputs to the PLC and SCADA

- The **controller instances**: all the UNICOS-CPC objects described in the specification file are represented by a POU (Programmable Organization Unit). In each POU all the instances of this object are executed.
- The **controller communication**: it contains all the necessary parameters (e.g. IP address…) required from the PLC/IPC to **establish the communication with the SCADA.**
- The **process control logic**: it contains the specific logic required by the specifications of the plant control.
- The **topology**: this file defines the cycle scan and the **execution order** of all the application POUs.
- The **supervision database**: a file containing all CPC objects instances of the application. It has also the communication parameters for the IPC access and also the mapping of all signal addresses to be exchanged with the IPC.

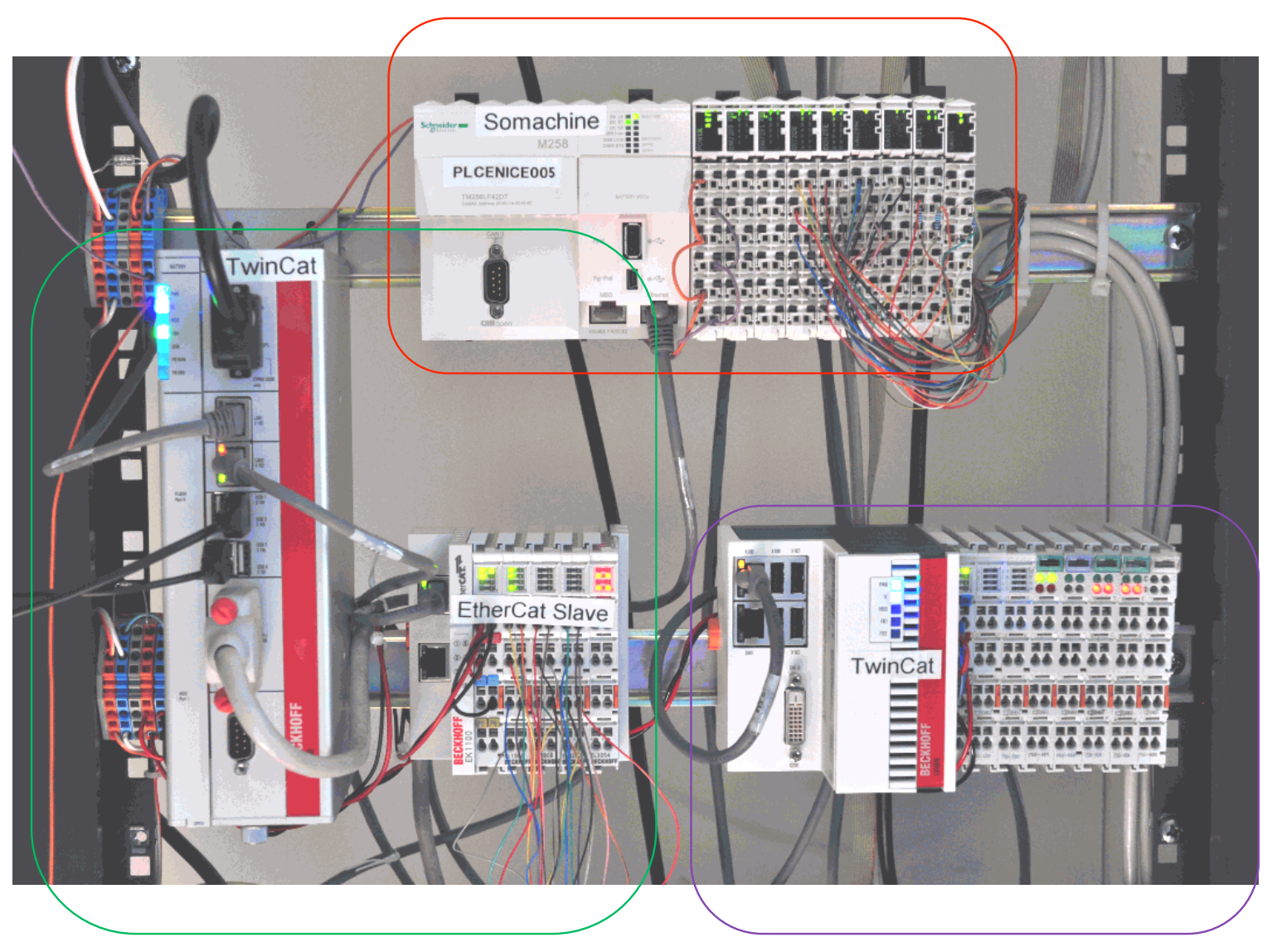
Why CODESYS?

- It follows the **IEC 61131-3 standard** and supports the ***PLCOpen* XML** standard exchange file format.
- It is executed as an independent **runtime system into Windows or Linux operating systems** opening the floor to any IPC.
- Select the most suited platform for the real needs regardless of supplier.
- Avoid supplier specific language-extensions that prevent the inter-changeability.
- Reduce cost and time for migrating from devices manufactured by different vendors and porting a program from one vendor's programming software to another
- Around **300 manufacturers** have integrated this product as the development environment for their hardware.
- The integrated compiler transforms the CODESYS control code into native machine code to be downloaded to the hardware. The most popular **CPU families are supported**.

Why IPCs?

- Nowadays many manufacturers offer a **large choice of hardware IPCs CODESYS compliant**.
- **Large archiving** capabilities: In opposition to a standard PLC, the internal **memory** of an IPC is far **less constrained**.
- Advanced calculations: IPCs running CODESYS can integrate **complex algorithms** developed in high level programming languages (e.g. C++).
- **Performance**: Hardware cycle time considerable faster than PLCs

Schneider SoMachine PLC



Beckhoff TwinCat IPC/PAC

TwinCat Embedded PLC

The hardware prototype.

Local operation:
A subset of the UNICOS CPC SCADA functionality
Is implemented into Siemens and Schneider touch panels.

Industrial Controls
Engineering Department

Engineering Department