



Dr. Mikko Heikkilä  
Technical Manager

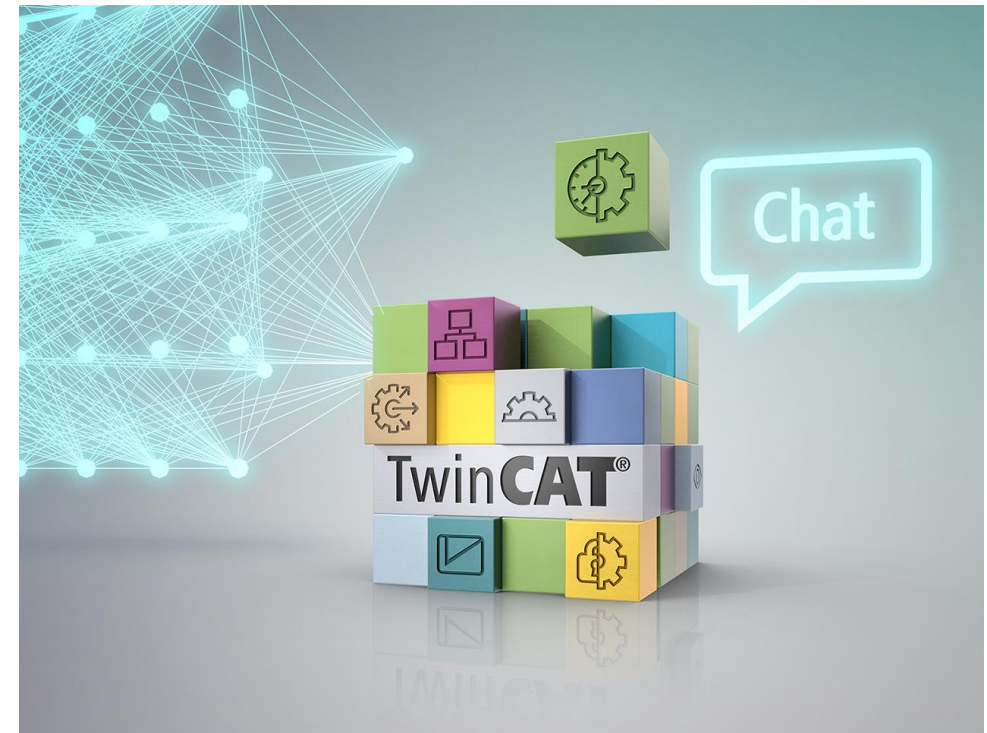
## TwinCAT Machine Learning

Artificial intelligence seamlessly  
integrated at **control level**



## TwinCAT Chat

Automation projects with  
AI-assisted **engineering**

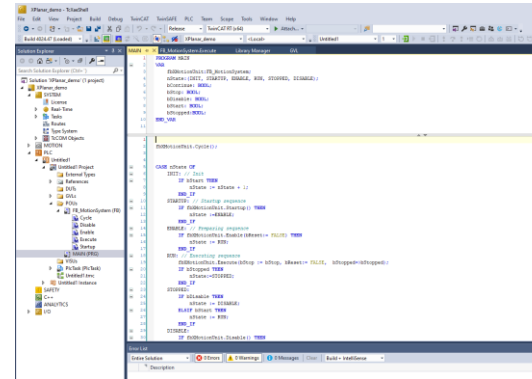


# TwinCAT 3: The flexible software solution for PC-based control

BECKHOFF

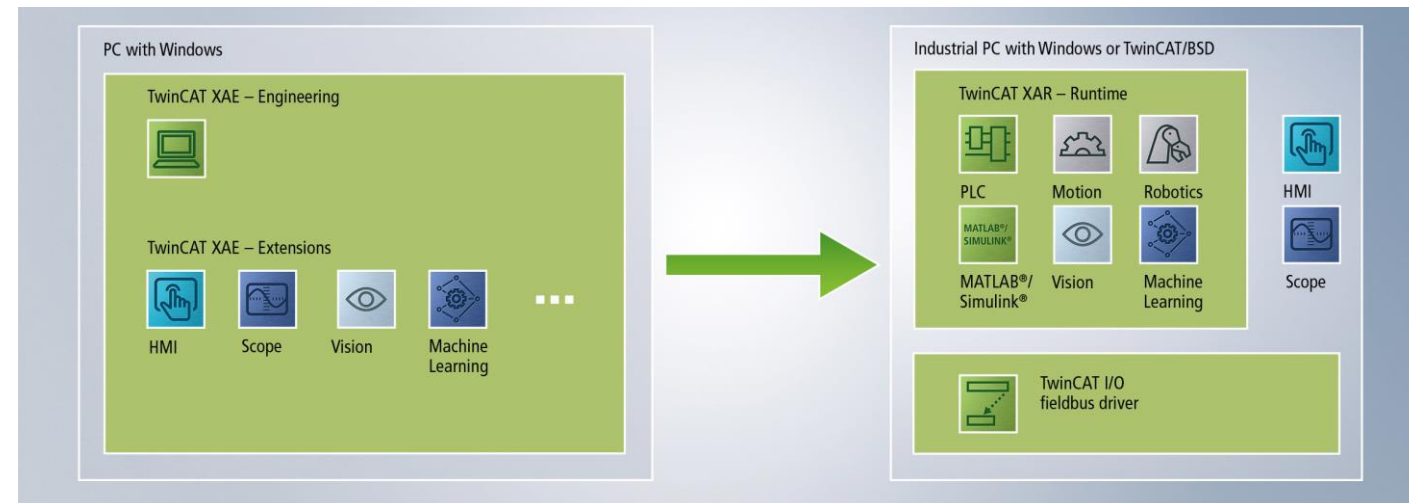
## Engineering

- TwinCAT XAE allows programming and configuration of hardware in one tool.
- Supports IEC 61131-3, C/C++, and MATLAB®/Simulink®.
- Integrated debugging and diagnostic features.



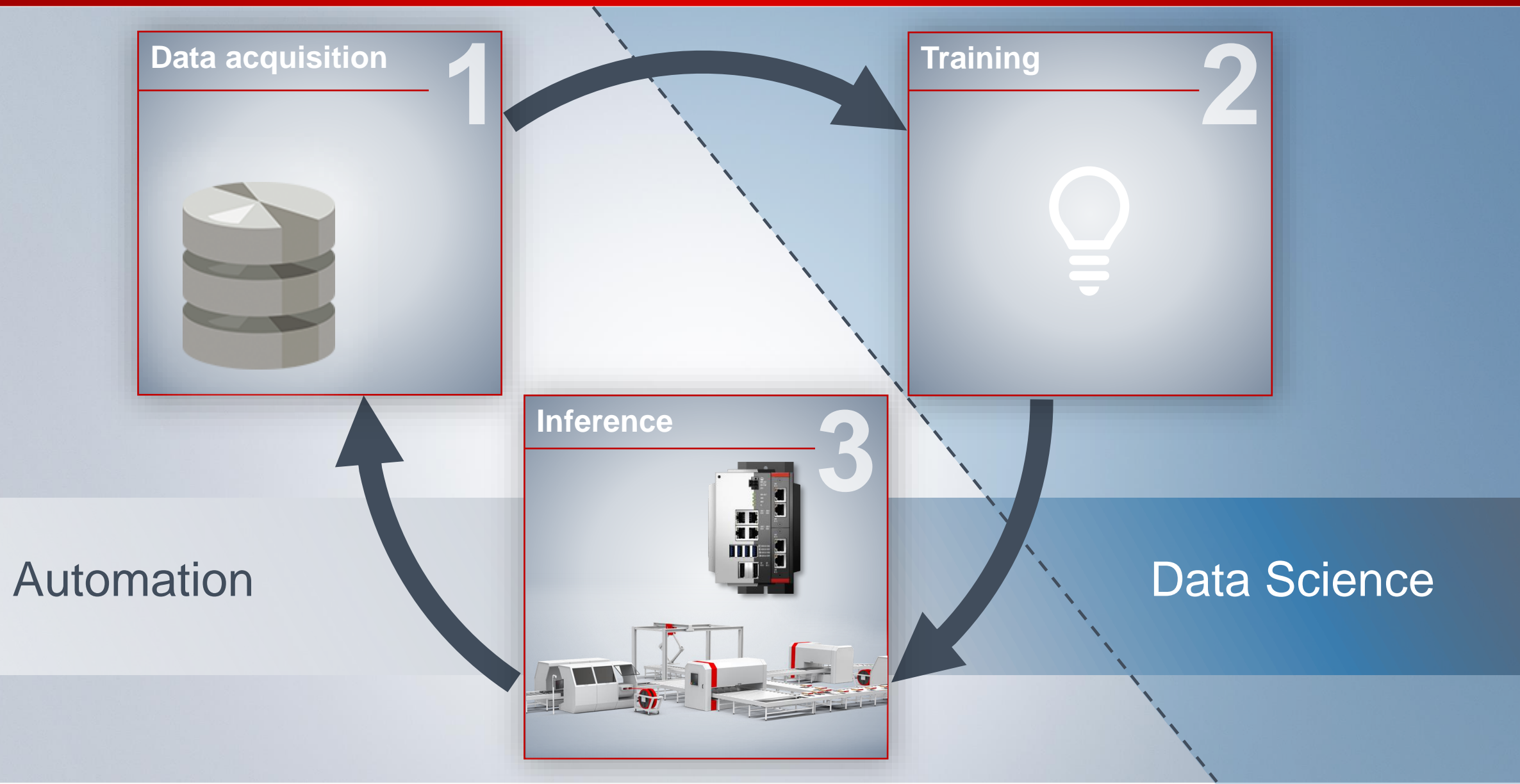
## Runtime – Control level

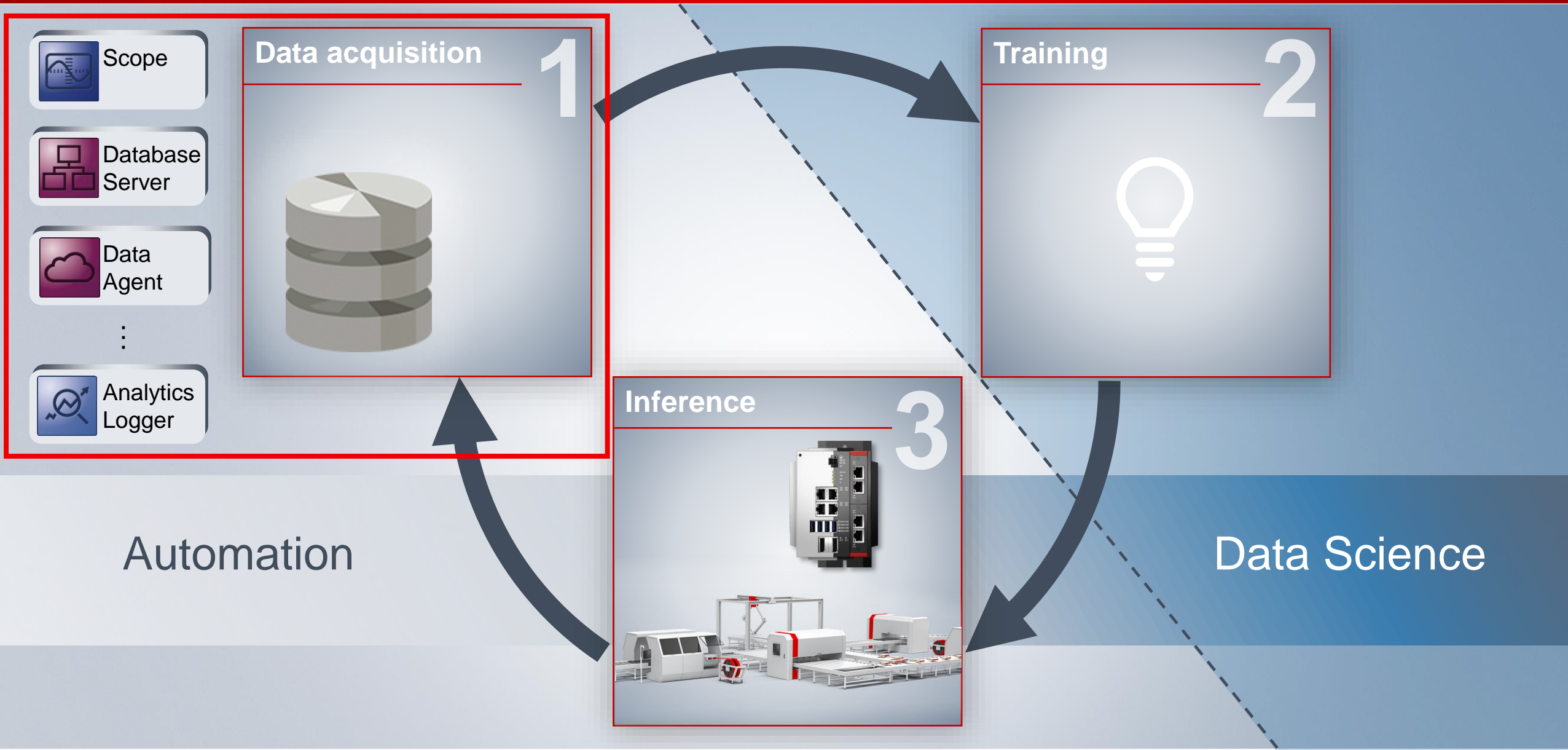
- TwinCAT XAR: Real-time runtime for machine control.
- Modular architecture supports application-specific extensions.
- Simultaneous real-time execution and OS operation.



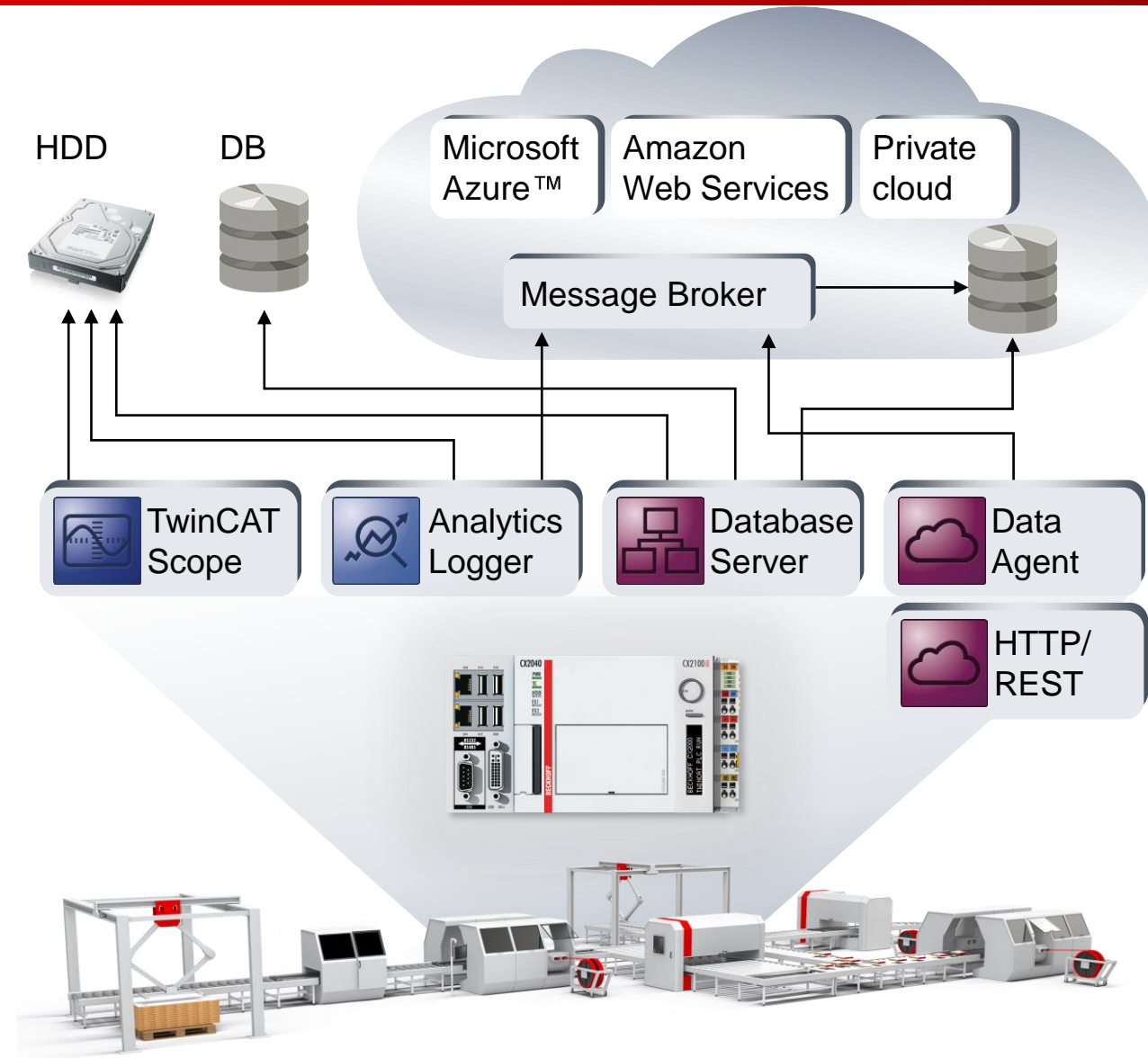




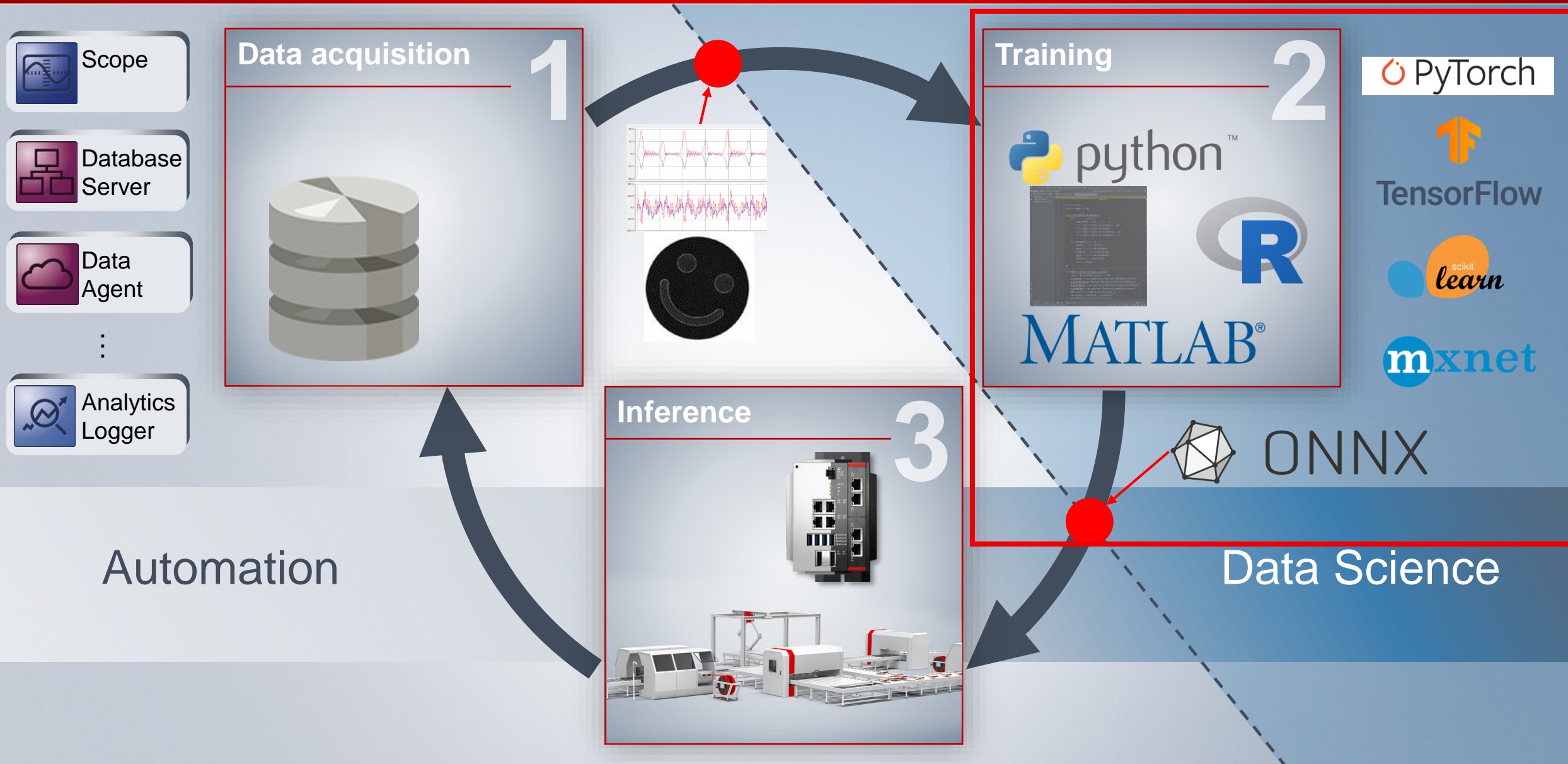




- Different demands for collecting machine data:
  - SQL or noSQL
  - File-based
  - Local or remote
  - Limited port releases
  - Cloud-based data lake, and more
  - Product family for image acquisition
  - Image (pre)processing
  - Image storage





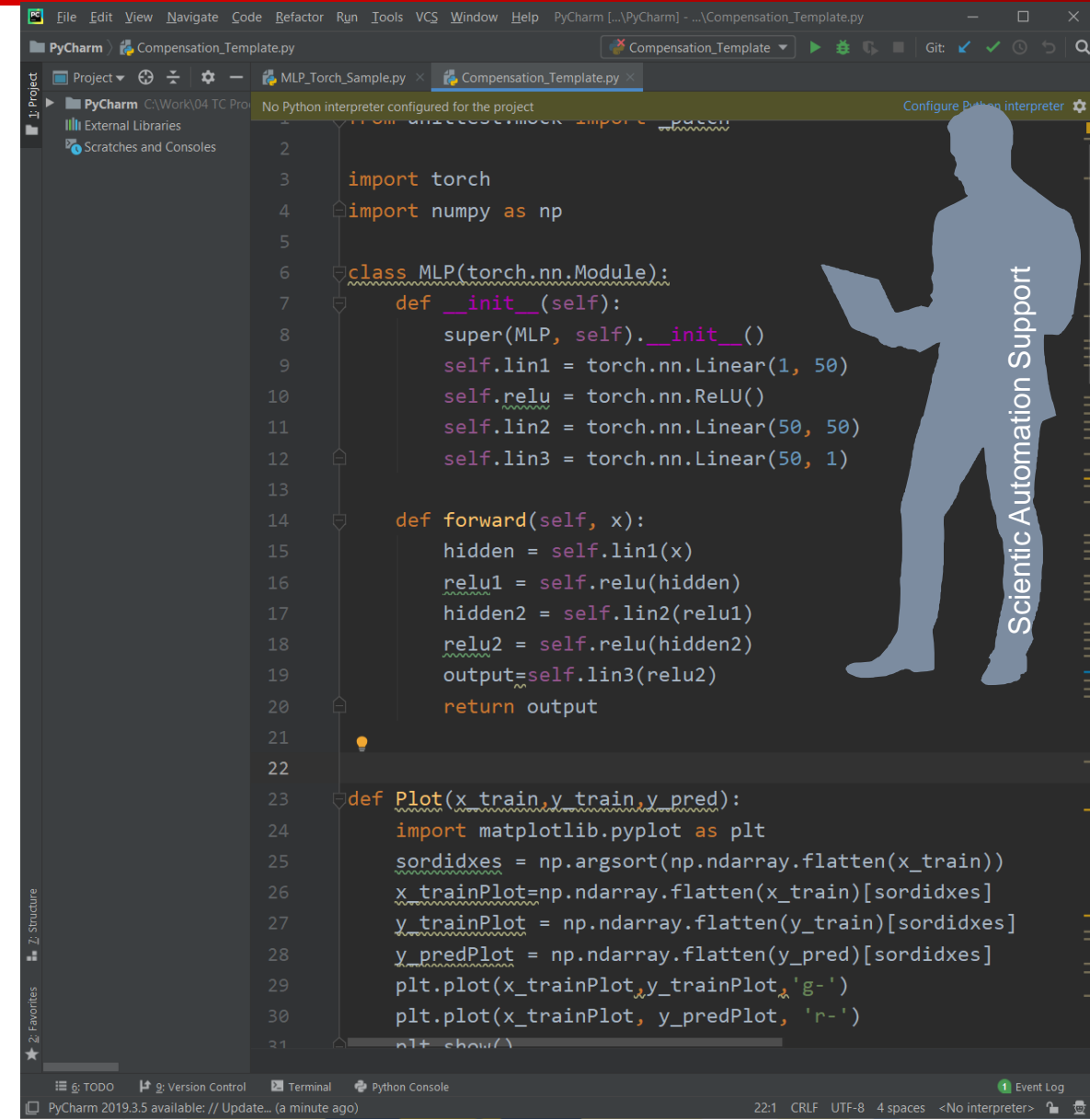


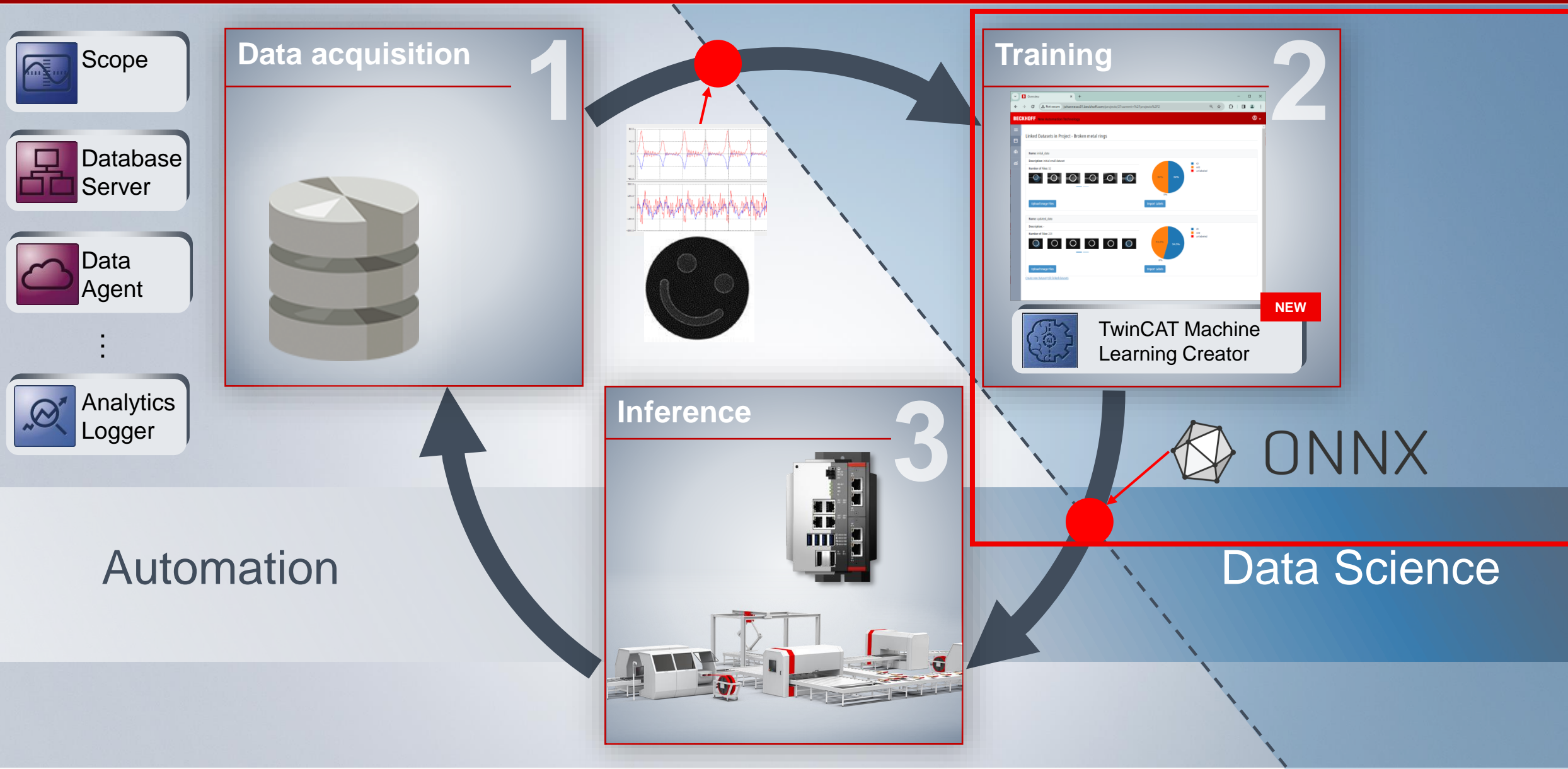


# Data modelling, training of AI models

BECKHOFF

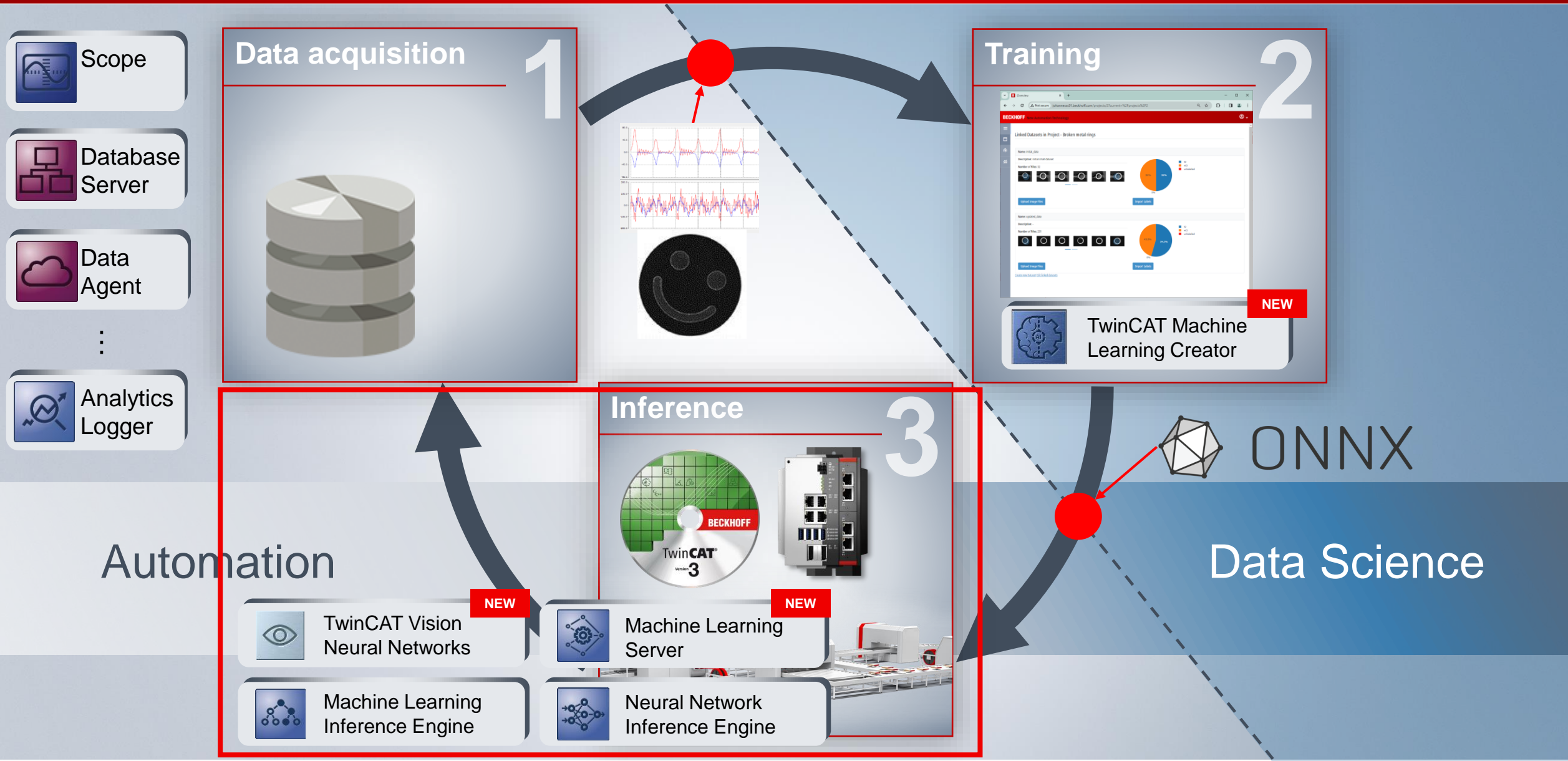
- More than 80% of all data scientists work with Python or R code
- Most of the libraries used for data labeling, cleaning and modelling are free of charge
- an interoperability standard is essential:





# Automation and Data Science | Deployment of AI models in the shop floor

BECKHOFF

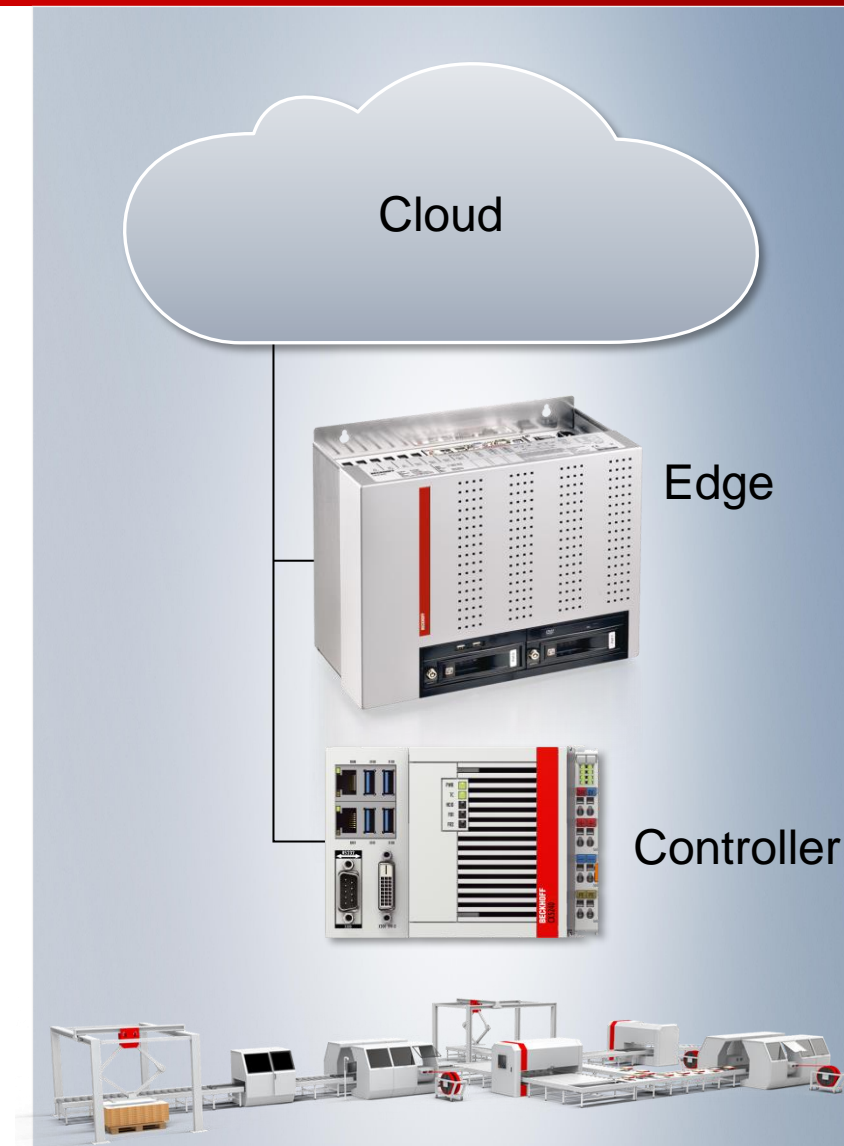




# Deployment of AI models

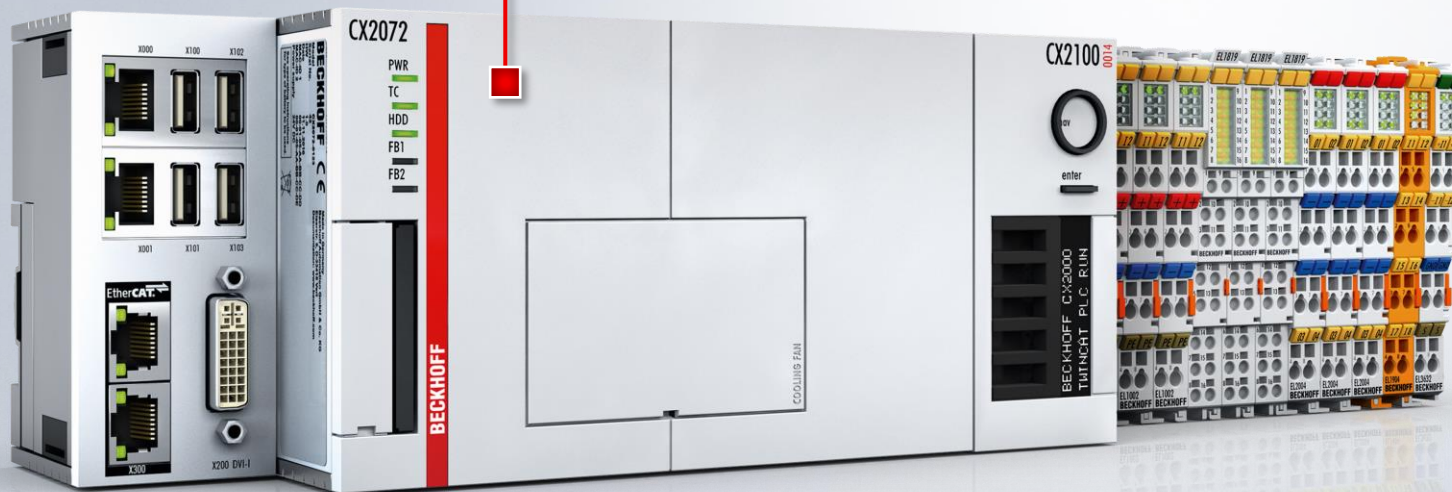
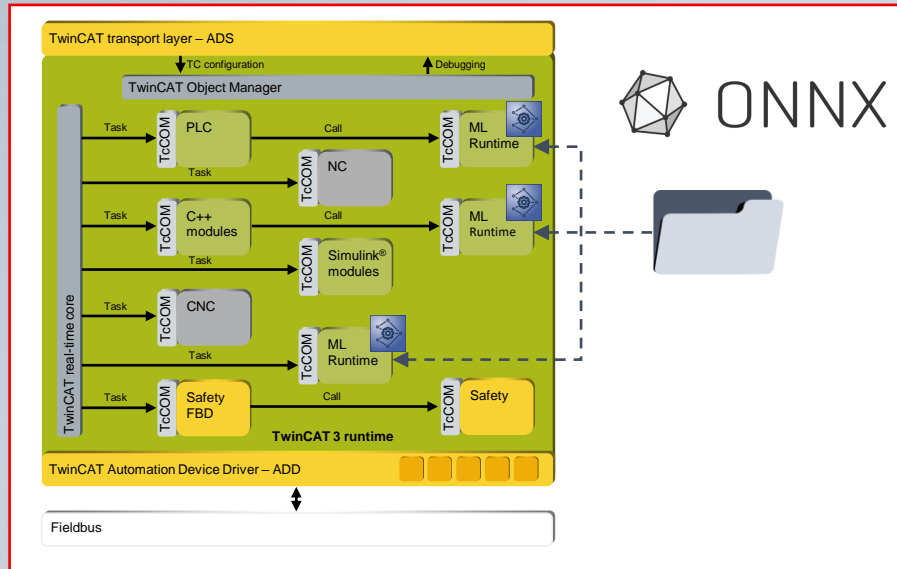
BECKHOFF

- Options for the integration
  - deterministic execution of AI model in the PLC code on standard controllers (local processing)
  - accelerated execution of AI model on GPU and manycore CPU, called from the PLC (local or edge)
  - deployment of docker containers with communication interface to TwinCAT runtime (local or edge)
  - deployment on a cloud system
  - deploying AI models with MATLAB and Simulink



# Deterministic execution of AI model in the PLC code on standard controllers (local processing)

BECKHOFF



# Real-time Inference | PLC API

BECKHOFF

The screenshot displays the TwinCAT Project11 - TcXaeShell IDE. The left sidebar shows the Solution Explorer with the project structure. The main editor window shows the PLC program code for the MAIN program. The code is as follows:

```
1 PROGRAM MAIN
2 VAR
3   fbprediction : FB_MlPrediction; // ML Interface
4   sModelFile   : T_MaxString := 'C:/models/TrigonometryMLP.bml'; // converted onnx file
5   nInputDim, nOutputDim : UDINT;
6   dtype : ETcMlDataType := ETcMlDataType.E_MLLDT_FP32_REAL;
7   nnInput : ARRAY[1..10] OF REAL;
8   nnOutput : ARRAY[1..10] OF REAL;
9
10 CASE nState OF
11   0: // idle state
12     IF bLoadConfig THEN
13       bLoadConfig:=FALSE;
14       nState:=10;
15     END_IF
16   10: // config state
17     fbprediction.stPredictionParameter.MlModelFilepath:=sModelFile; // provide model path and name
18     fbprediction.stPredictionParameter.MaxConcurrency := 1; // no concurrent calls to this fb
19     IF fbprediction.Configure() THEN // load model
20       // check input/output dimensions and datatype
21       bFailed:=fbprediction.GetInputDim(nInputDim);
22       _FAILED(bFailed);
23       bFailed:=fbprediction.GetOutputDim(nOutputDim);
24       _FAILED(bFailed);
25       bFailed:=fbprediction.CheckPreferredIODataTypes(ETcMlDataType.E_MLLDT_FP32_REAL,ETcMlDataType.E_MLLDT_FP32_REAL,bPreferred);
26       _FAILED(bFailed);
27       bFailed:=fbprediction.CheckSupportedIODataTypes(ETcMlDataType.E_MLLDT_FP32_REAL,ETcMlDataType.E_MLLDT_FP32_REAL,bSupported);
28       _FAILED(bFailed);
29     END_IF
30   20: // Predict state
31     fbprediction.Predict(
32       pDataInp:=ADR(nnInput) ,
33       nDataInpDim:= nInputDim,
34       fmtDataInpType:= dtype,
35       pDataOut:=ADR(nnOutput) ,
36       nDataOutDim:= nOutputDim,
37       fmtDataOutType:= dtype,
38       nEngineId:= nCurrentEngineID,
39       nConcurrencyId:= 0); // concurrency not in use in this sample. see concurrency sample in infosys for non-blocking parallelization
40
41     IF fbprediction.bError THEN // error handling
42       nState := 999;
43       hrErrorCode := fbprediction.hrErrorCode;
44     ELSEIF bLoadConfig THEN // load (updated) model
45       bLoadConfig := FALSE;
46       nState := 10;
47     END_IF;
48
49   999: // Error state
50     // add error handling here
51 END CASE
```

Three red arrows point to specific parts of the code:

- load model**: Points to the configuration and loading logic (lines 17-28).
- execute model**: Points to the Predict function call (lines 31-39).
- update model**: Points to the error handling and reloading logic (lines 41-47).



**1****Compact: C6030**

- 11<sup>th</sup> Gen. Intel® Core™-i
- CPU + iGPU

**2****Embedded: CX2072**

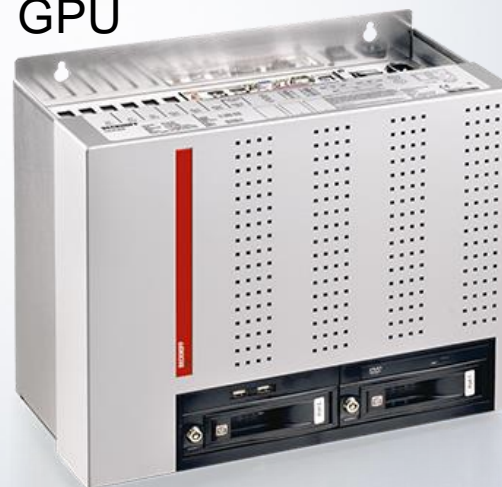
- Intel® Xeon® D
- AMD Embedded Radeon™ GPU

**4****Compact + GPU: C6043**

- 13<sup>th</sup> Gen. Intel® Core™-i
- NVIDIA® RTX™ A4500 (ex factory)

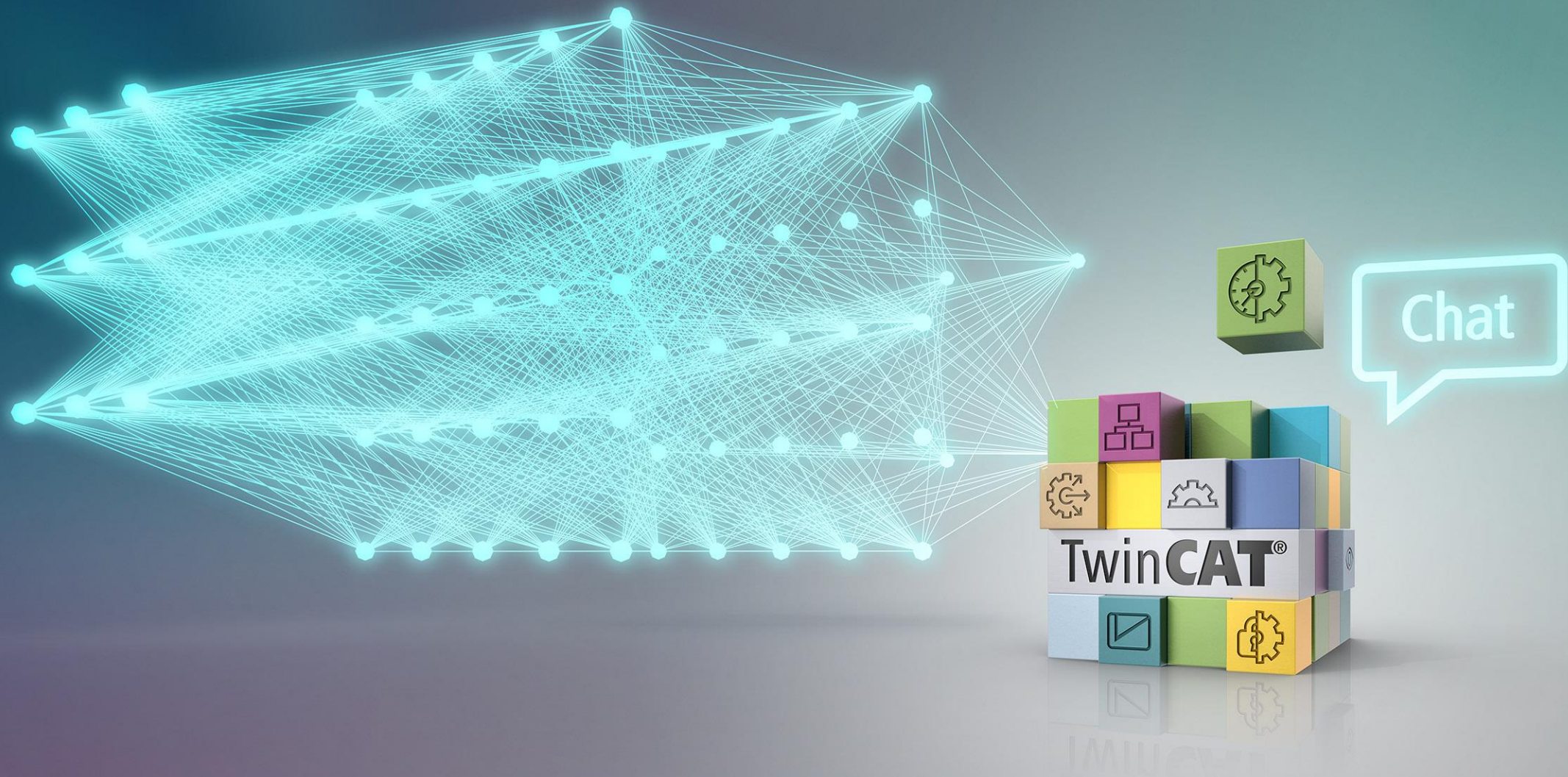
**3****Industrial PC: C6640, C6650**

- 13<sup>th</sup> Gen. Intel® Core™-i
- free slot for dGPU: max. 210 mm PCIe, 70 W TDP

**5****Industrial PC: C6675**

- 13<sup>th</sup> Gen. Intel® Core™-i
- free slot for dGPU: full size PCIe, 300 W TDP



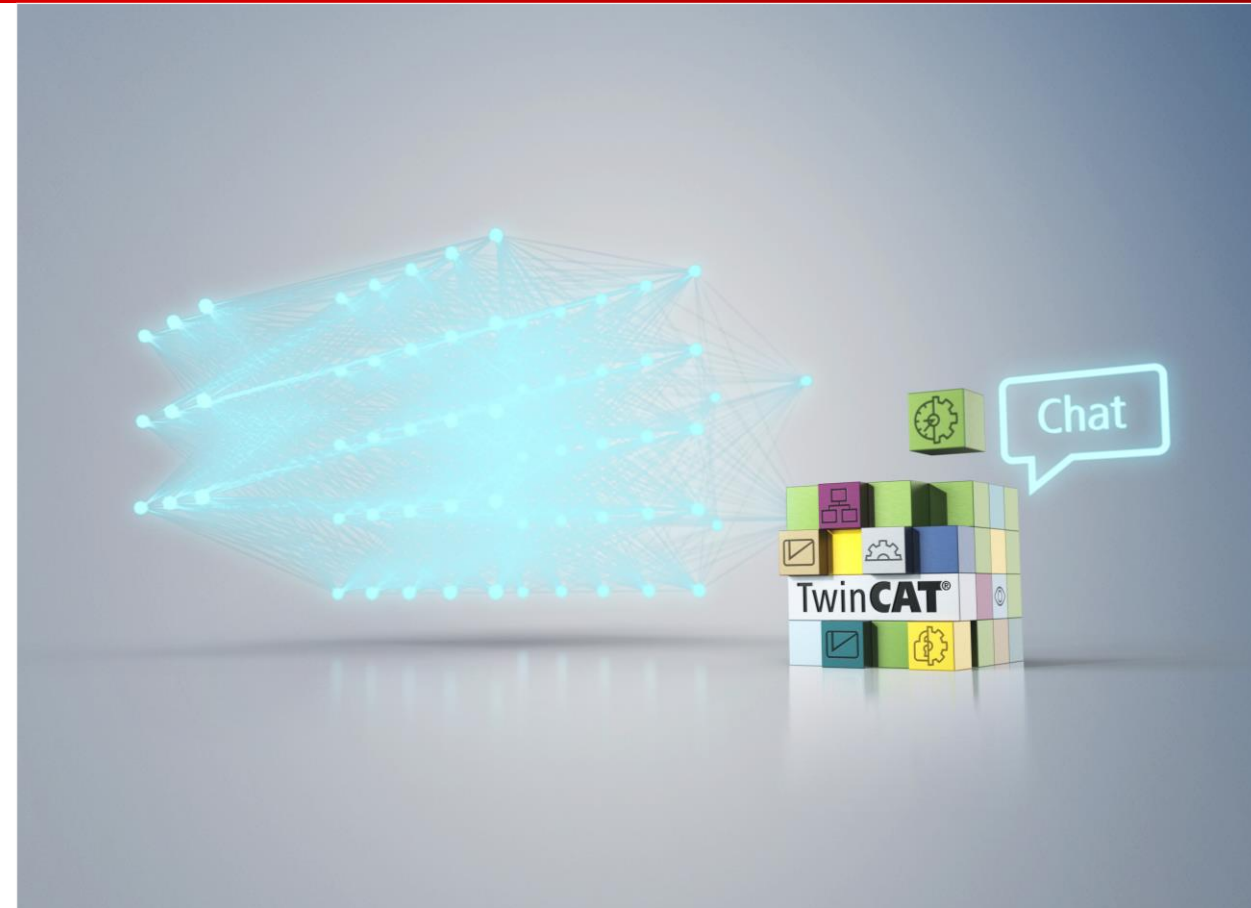




## The rise of LLM chatbots

### Opportunities for programmers

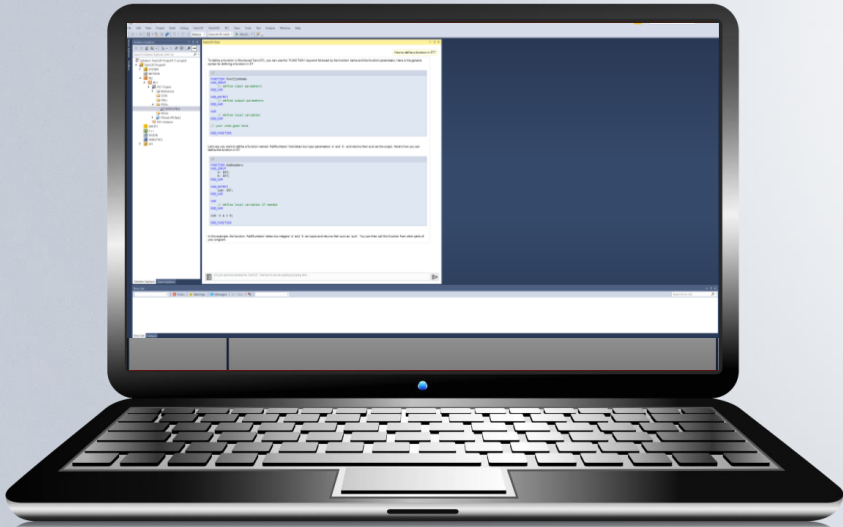
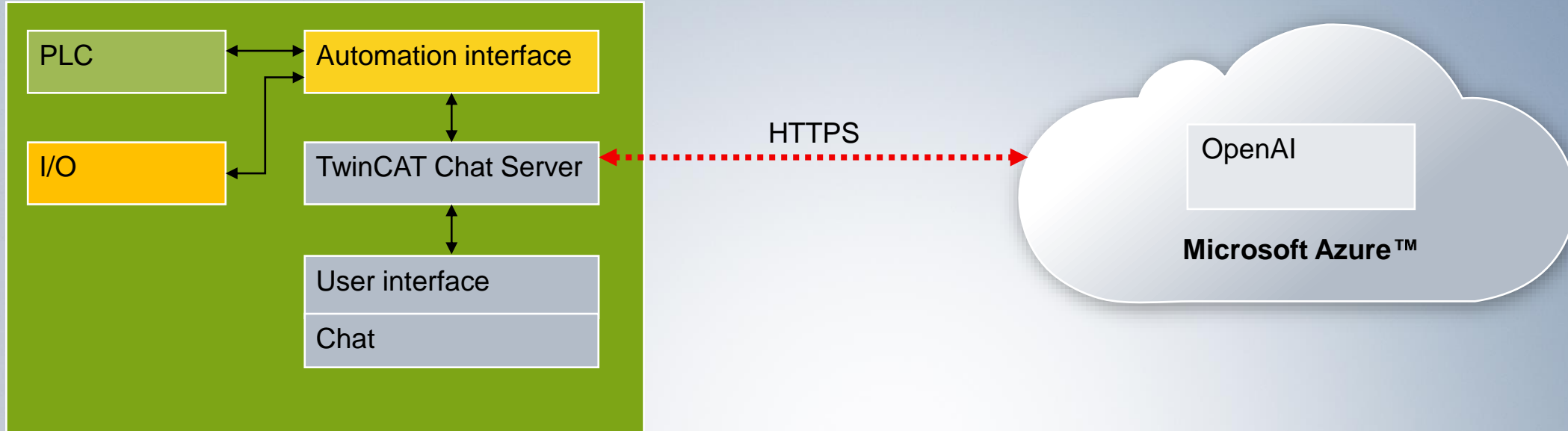
- offers automated simple generation, documentation, addition, and optimization
- facilitates especially repetitive programming tasks
- reduces routine tasks for the programmer
- cuts down on the time required and minimizes sources of error





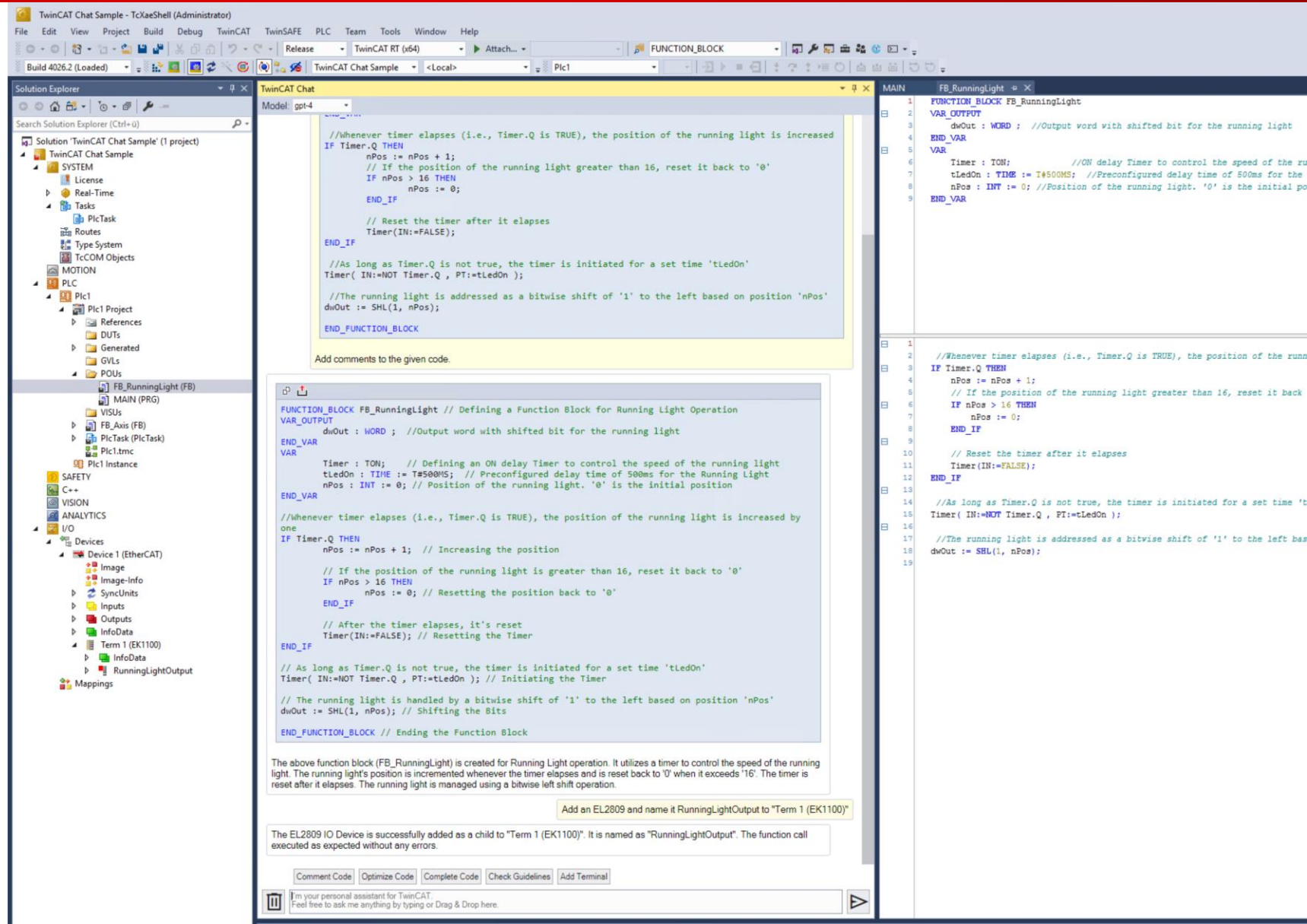
# TwinCAT Chat | ChatGPT integration in TwinCAT I/O

BECKHOFF



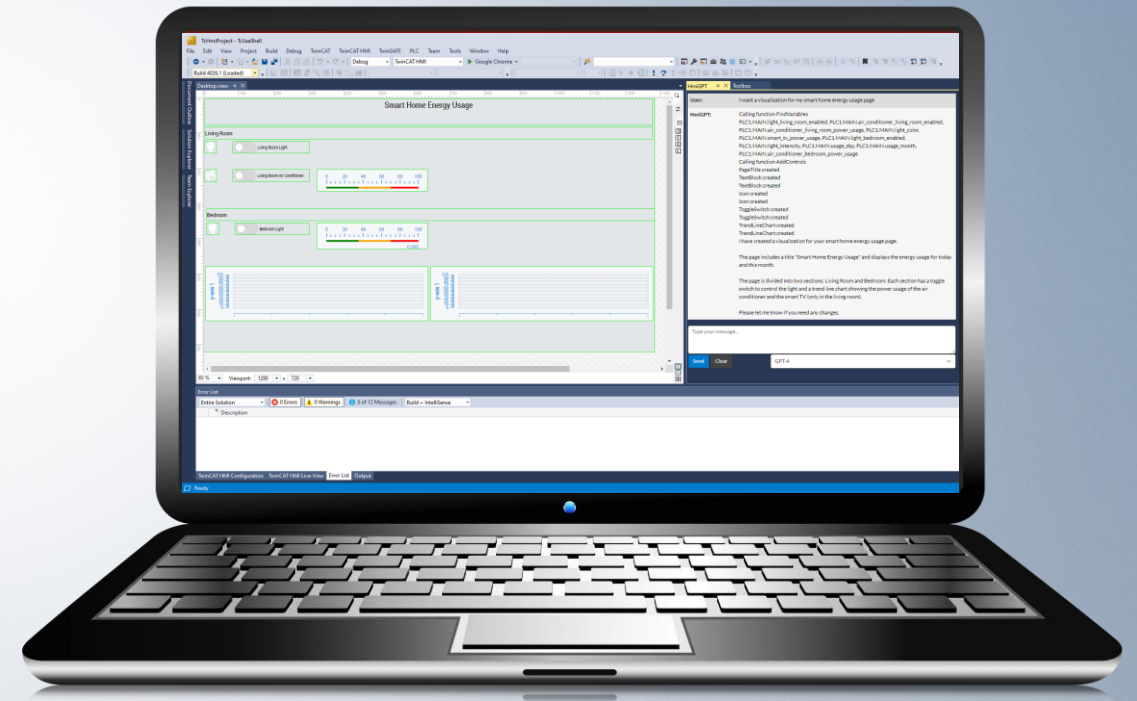
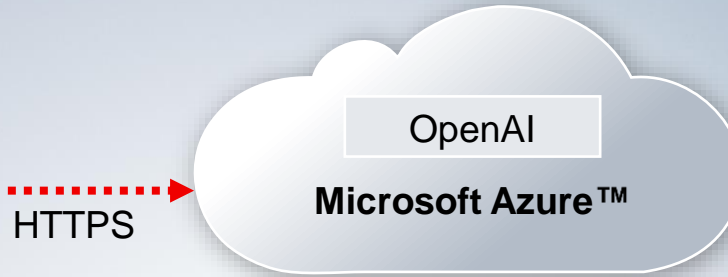
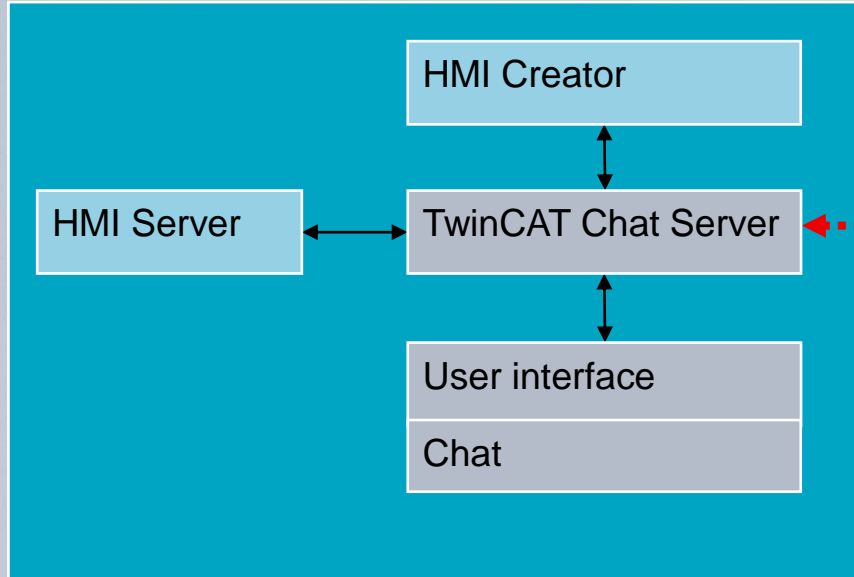
## Common **chat** window in Visual Studio

- includes Beckhoff Infosys content
- IEC 61131-3 syntax highlighting
- use cases
  - generating code
  - commenting code
  - completing code
  - add terminals
  - create mappings
  - add axis

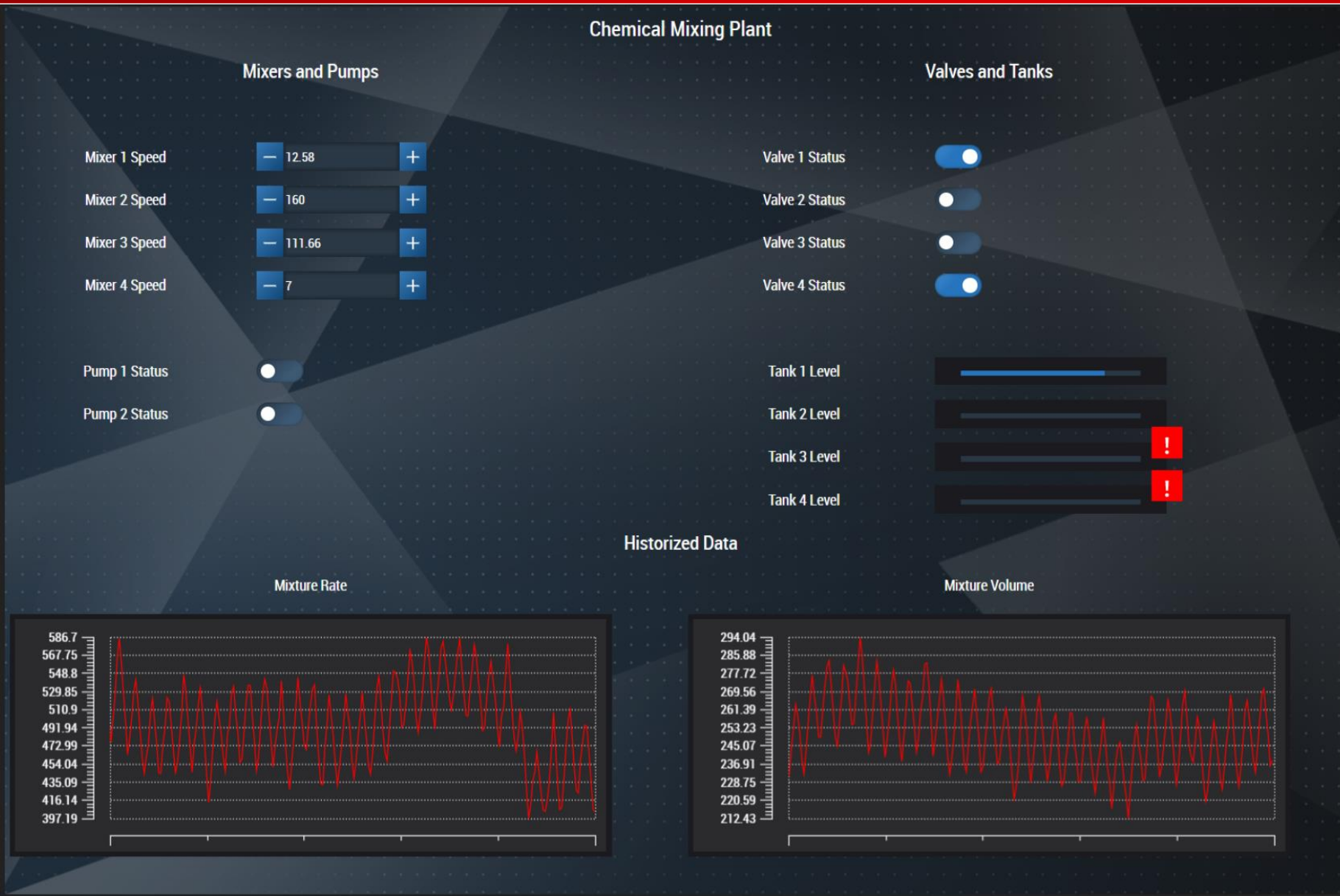


# TwinCAT Chat HMI | ChatGPT integration in TwinCAT HMI

BECKHOFF







I need an HMI for controlling a chemical mixing plant. I have the following PLC variables in PLC1.MAIN: mixture\_rate, mixture\_volume, mixture\_temperature, mixture\_pressure, mixer1\_speed, mixer2\_speed, etc. Group the mixer speeds and pump statuses together on the left side of the page, valve statuses and tank levels on the right side, and historized data at the bottom of the page.

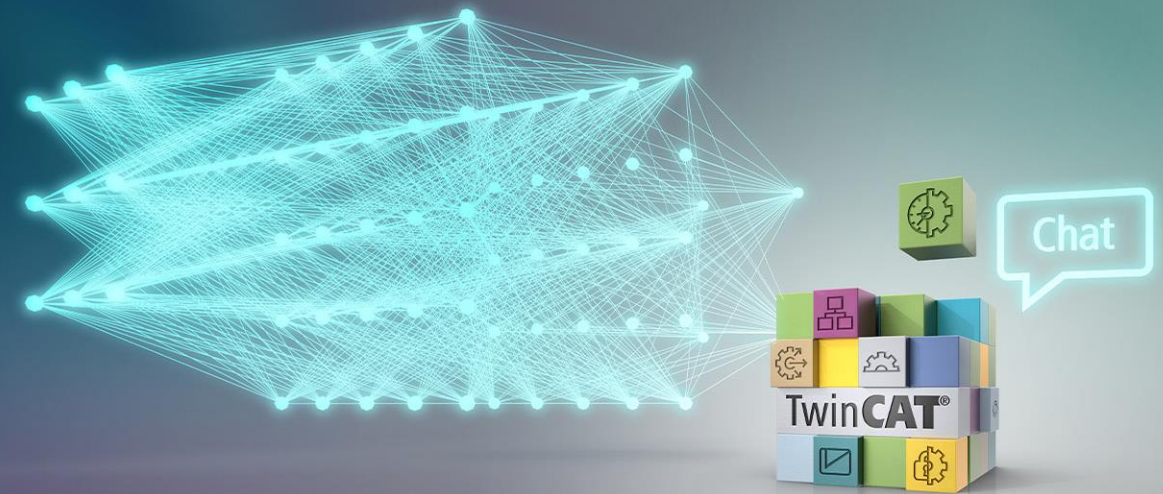
Artificial intelligence seamlessly integrated at **control level**

- Modular hardware/software for AI integration into PLCs.
- Aim to make AI adoption broader beyond experts.
- Examples
  - machine-integrated quality checking
  - reducing waste quantities
  - collaborative and context-aware robotics
  - machine optimization
  - predictive maintenance
- Stay tuned to the technology



### Automation projects with AI-assisted engineering

- Generative AI is here to stay
- Integration of chatbot technologies into TwinCAT Engineering
- Aims to increase productivity of programmers and improves instant support
- Stay tuned to the technology





Questions ?

BECKHOFF

