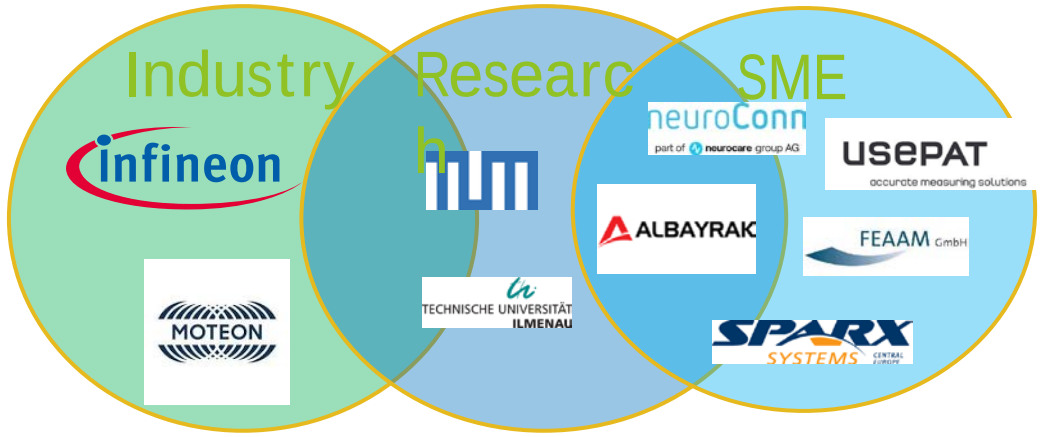
EUREKA PENTA ECOMAI:

ECOMAI: Ecological Motor Control and Predictive Maintenance with AI

ECOMAI Team



Germany

- Infineon Technologies AG
- Moteon GmbH
- neuroConn GmbH
- FEAAM GmbH
- Technische Universität München
- Technische Universität Ilmenau

Austria

- Sparx Systems Software GmbH + SCCH
- usePAT Gmbh

Turkey

- ALBAYRAK Makine Elektronik Sanayi ve Ticaret A.Ş.

Aeneas

Penta

SPONSORED BY THE
Federal Ministry of Education and Research

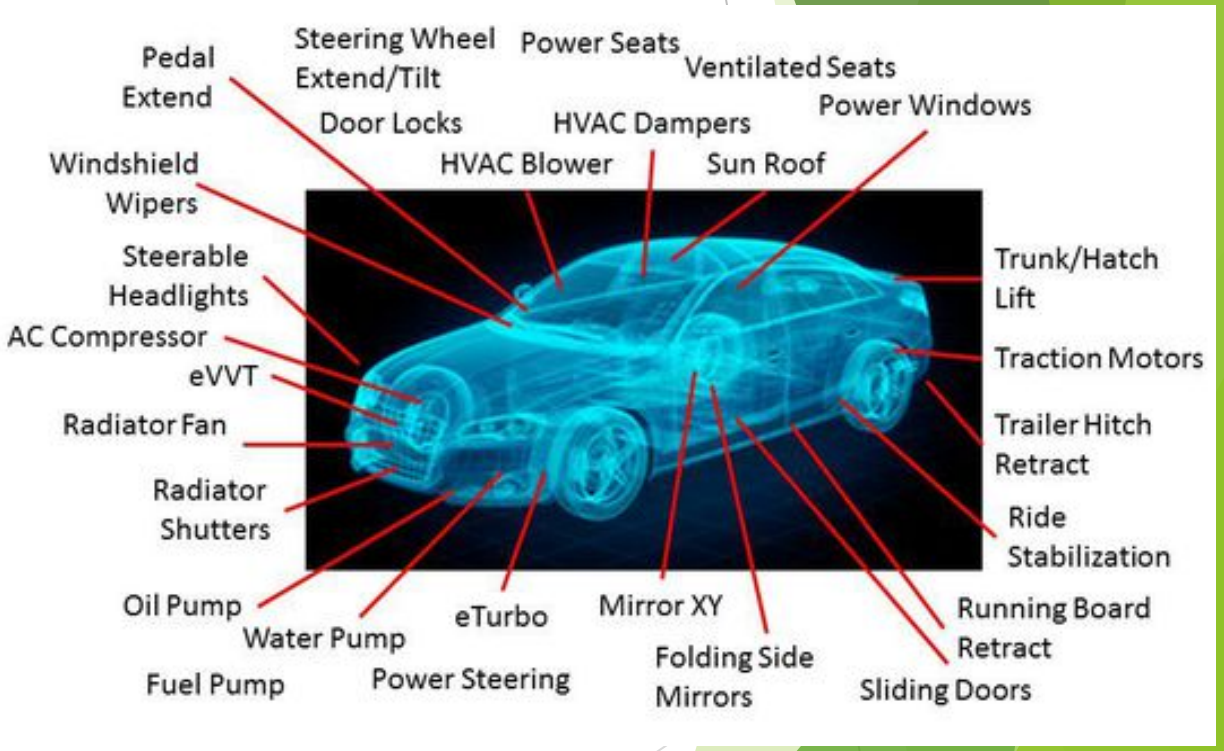
FFG
Forschung wirkt.

TÜBİTAK
EUREKA Ulusal Koordinasyon Ofisi

ECOMAI

Problem Statement

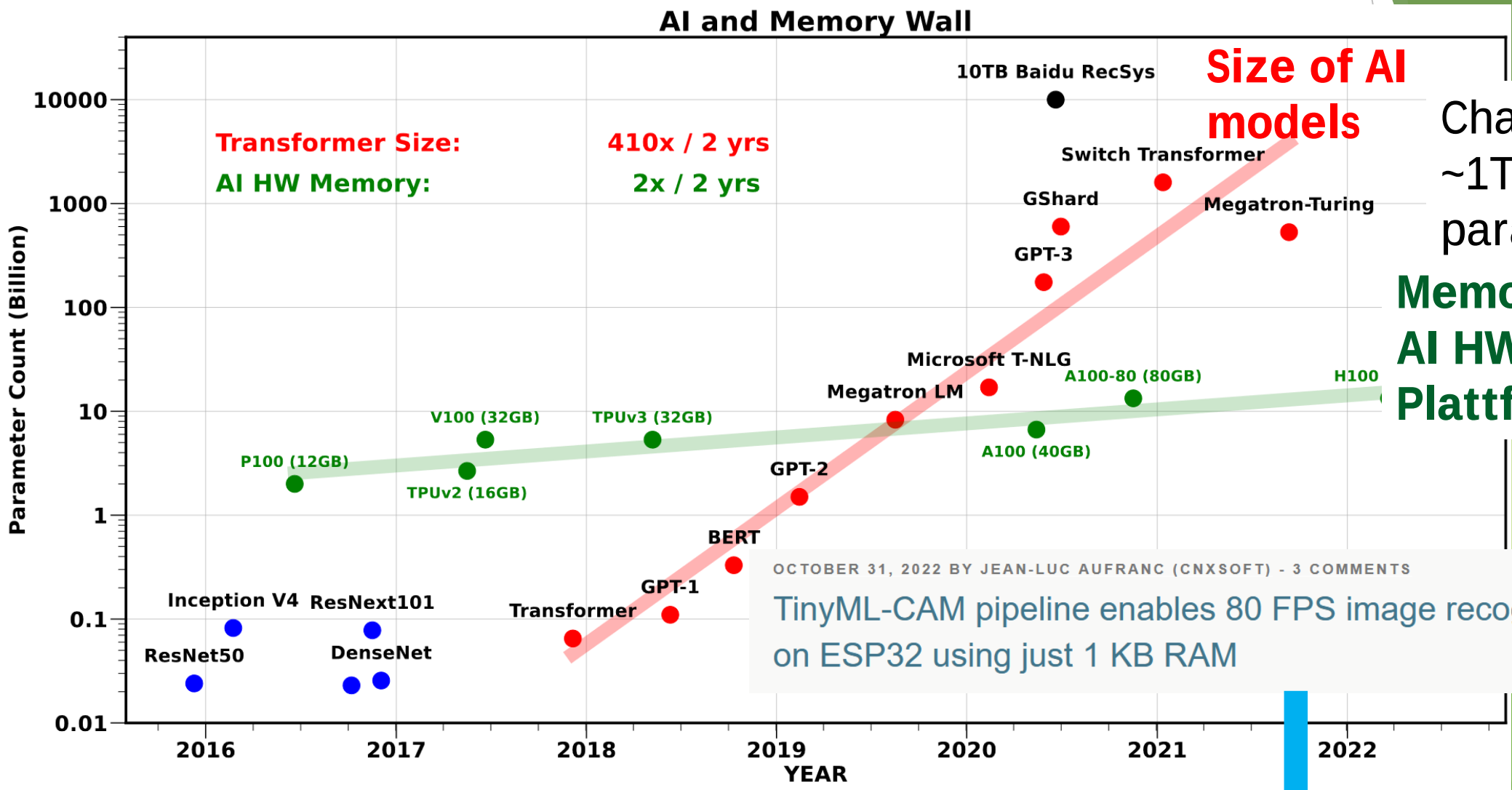
- Electrical Motors are an essential part of our daily live
- According to an FhG ISI study [2] electrical motors are responsible for 40% of the worldwide power consumption and in consequence for 20% of the worldwide CO2 emission. An **efficiency gain of about 30% is said to be feasible**.
- Condition monitoring of electrical machines, drives and applications is certainly not in widespread use (about 99.99% of all motors do not have this). **Unexpected downtime causes an economical damage of 56 billion € in industry and intralogistics worldwide and lowers overall equipment effectiveness to only 60%**
- Great potential to enhance motor systems with embedded AI systems
- But: Strict requirements in terms of energy and cost require specialized HW solutions



Goals of ECOMAI

- Develop a Edge AI Solution to be integrated into Motor drive systems consisting of:
  - Specialized AI hardware platform
  - Development kit with AI compiler, Model-based design and simulation environment
  - AI Applications for Predictive Maintenance and Energy-efficient Control to execute on the hardware
- Targets:
  - AI-enhanced Ecological Electrical Drive Systems: Reduced energy demand and longer lifetime
  - Reduced downtime of systems due to AI-enhanced predictive maintenance

AI becomes bigger and bigger – Anti Thesis: TinyML



Size of AI models

ChatGPT ~1Trillion params*

Memory of AI HW Plattforms

OCTOBER 31, 2022 BY JEAN-LUC AUFRANC (CNXSOFT) - 3 COMMENTS

TinyML-CAM pipeline enables 80 FPS image recognition on ESP32 using just 1 KB RAM

Figure: https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

https://www.cnx-software.com/2022/10/31/tinyml-cam-pipeline-esp32-fast-image-recognition/

*https://www.wired.com/story/how-chatgpt-works-large-language-model/

The hidden AI Revolution: tinyML

OCTOBER 31, 2022 BY JEAN-LUC AUFRANC (CNXSOFT) - 3 COMMENTS

TinyML-CAM pipeline enables 80 FPS image recognition on ESP32 using just 1 KB RAM

https://www.cnx-software.com/2022/10/31/tinyml-cam-pipeline-esp32-fast-image-recognition/

The Future of AI Is Tiny

Tiny AI reduces carbon footprints, brings deep learning at an affordable cost, creates context-aware consumer devices, cuts down data infrastructure, bolsters security, and more.

https://www.informationweek.com/data-management/the-future-of-ai-is-tiny/

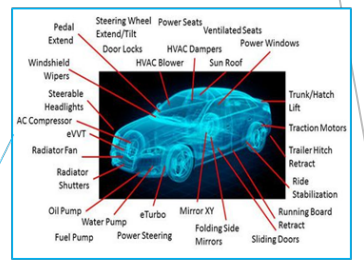
https://www.abiresearch.com/press/tinyml-device-shipments-grow-25-billion-2030-15-million-2020/

Technology Value Chain

Reduced Downtime of Transportation Systems and Machinery

Longer Lifetime of Drive Systems

Ecological Electrical Drives for Green Systems



Use Cases for Predictive Maintenance
Train Door System (TUR)
Electric Wheelbarrow (NL)
Wireless Charging Unit (NL)

Use Cases for Ecological Electrical Drives
Automotive Compressor System (GER)
Brushless Motor and Battery System (NL)
Electrical bike (GER)

Requirements Datasets for AI training

AI-enhanced Condition Monitoring
Condition Monitoring at the Edge (NL)
Federated monitoring for Predictive Maintenance (FI)

AI-enhanced Control for Energy-efficient Drives
Initial Motor Control Tuning (GER, NL)
Field Motor Control Tuning (GER, NL)

Model-based Design Kit for AI-enhanced Drive Systems
Model-based Design Framework (GER, AUT)
AI Compiler (GER, AUT)
Full Motor Drive System Virtual Prototypes (GER, AUT, NL)
Test Platform for Predictive Maintenance (NL)

Monitoring and Control Tasks HW/SW Architecture

Specialized Edge AI Hardware for Drive Systems
Low Power AI Compute Platform (GER)
Approximate Computing Engine (NL)
Scalable parallel DSP for 1D/2D DNNs (FI)
Low-power ultra-low latency wireless communication (FI)

Project Outcomes (Demonstrator, KPIs)

Project Outcome	Demonstrator	KPIs
Specialized Edge AI Hardware with AI Compiler	Hardware Chip Prototype (IFX) with AI Compiler (IFX,TUM,SCCH)	• Performance per Watt/Chip Area Unit • AI Memory Compression Ratio
Development Kit: Model-based Design Environment for AI-enhanced Motor System	Enterprise Architect - IoT-PML extension	• Design time reduction (Time-to—market) • Design Quality Improvement • Safety Qualification
Full-system Simulation Support	VP of motor system and AI HW	Motor System Energy Consumption Motor System Lifetime
AI Enhancement for Electrical Drive Systems – Ecological Control	Test environment dynamic load change motor control application	Motor System Energy Consumption Motor System Lifetime
AI Enhancement for Electrical Drive Systems– Predictive Maintenance	train door/PSD test platform with fault induced scenarios for AI model development to test a predictive maintenance use-case.	Prediction Accuracy Reduction of System Downtime
Lower Entry for SMEs into AI-Enhanced Motor Drive Technology	Evaluation on Use Cases from SME partners	Easy adoption of project results by SMEs

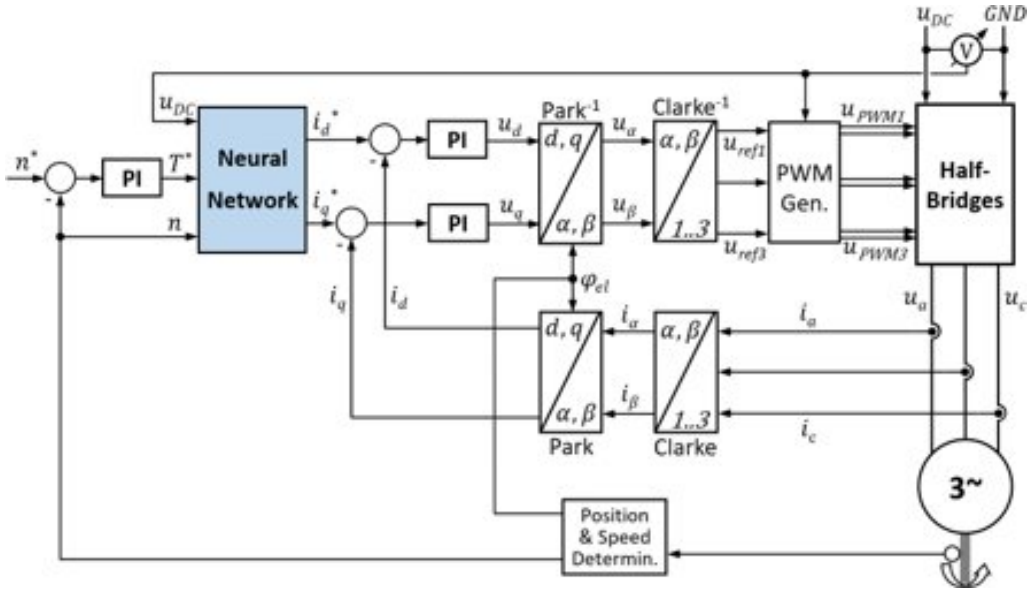
Highlights and demonstration (1)

- In-project trainings of use case partners in modeling and AI methodologies
- Testbed systems for dataset collection have been or are being set up

- First machine learning model types for the Platform Screen Doors and Process Monitoring are trained and validated

Highlights and demonstration (2)

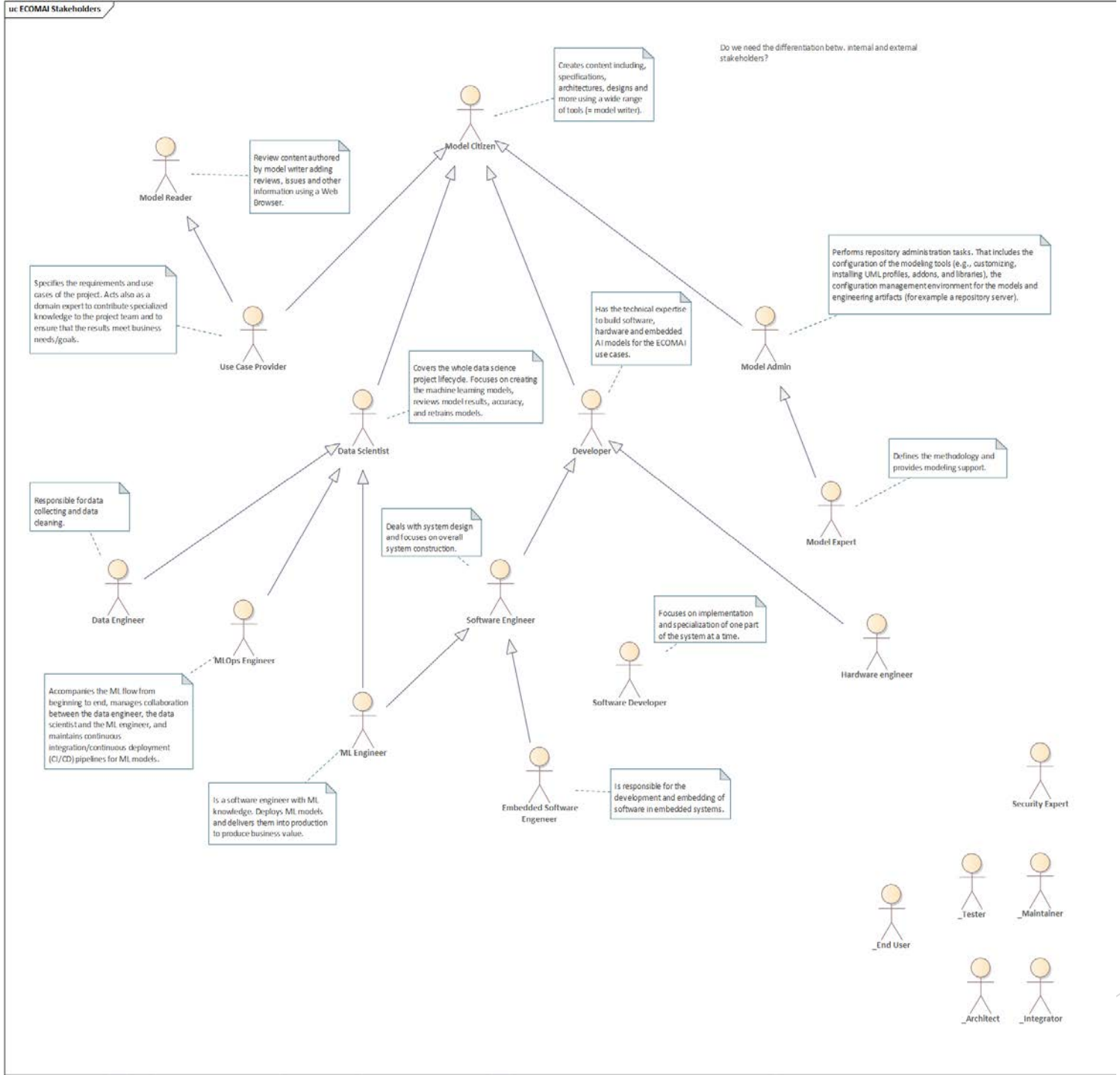
- **Control Patterns** for AI-enhanced Motor Control were defined

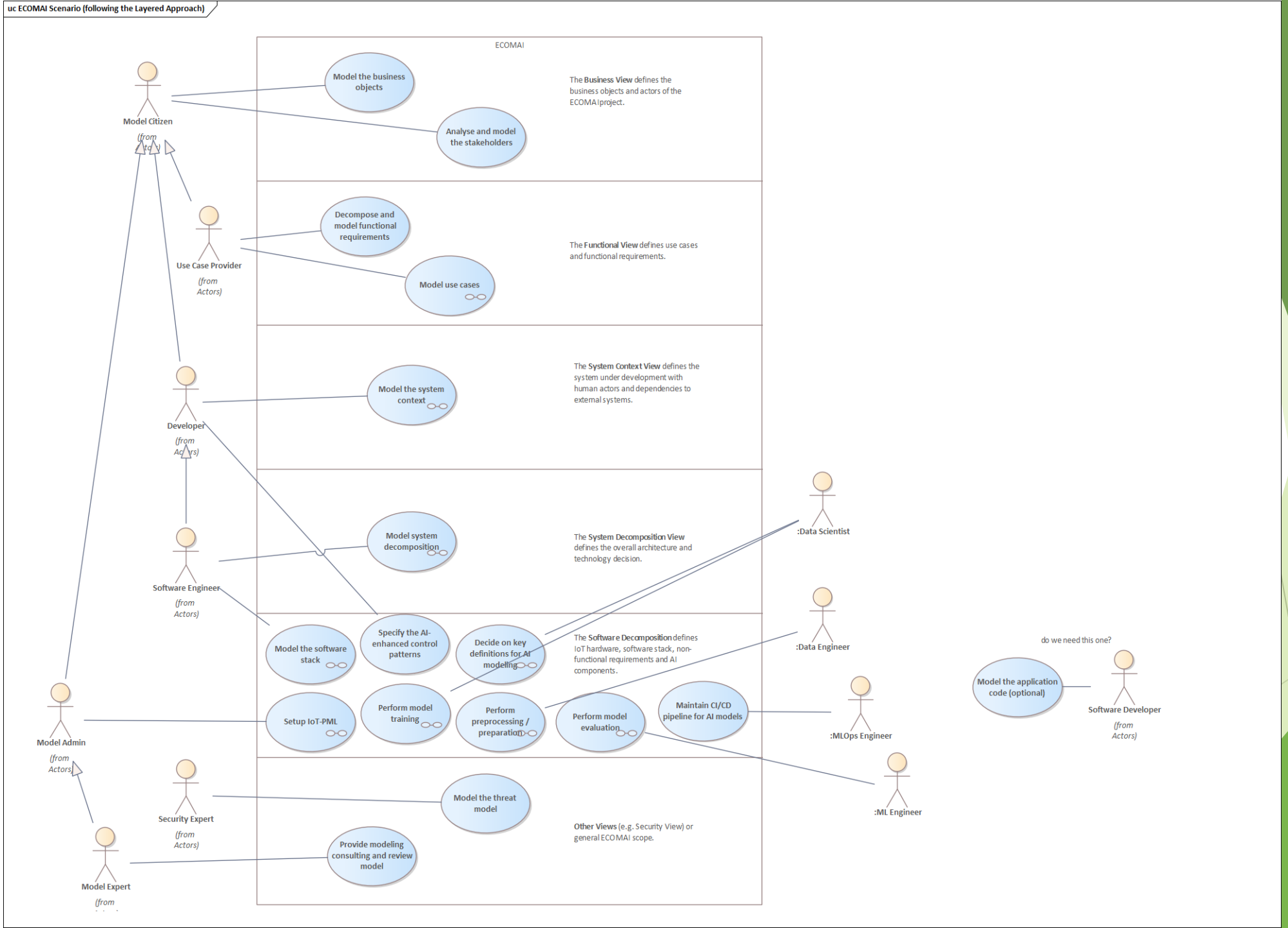


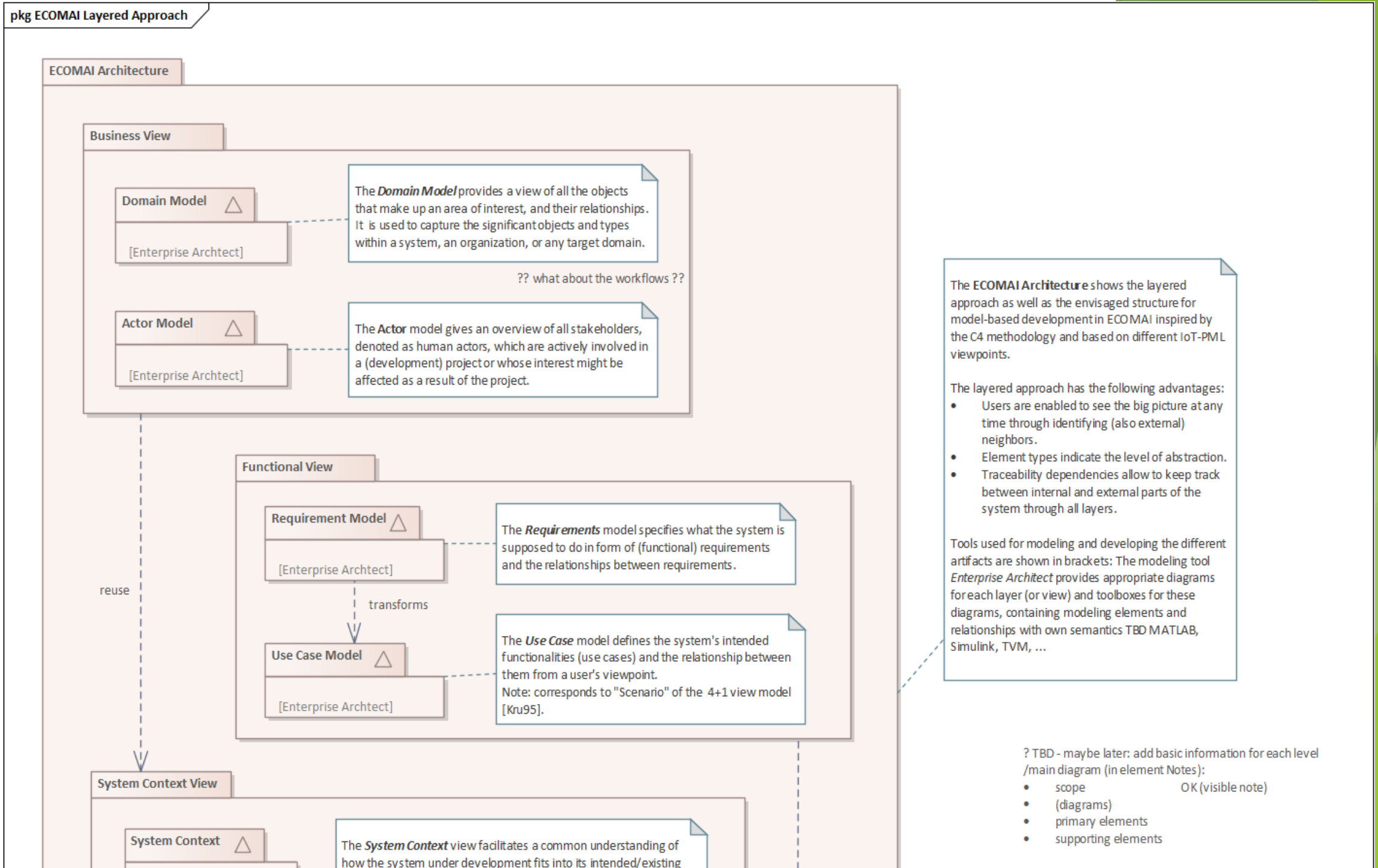
- Describe where in the loop deep learning models such as neural networks (NN) can be applied to replace/enhance classical control blocks

SCCH/SPX

- ECOMAI Development Kit
  - High Level: Archimate (currently only for Application Components, UML::Actors must be transformed to Archimate::Business Actors)
  - BPMN describing the high level process
  - VVML describing the Input/Output parameters in Conext to the domain model
  - UML describing the classes


uc ECOMAI Stakeholders
Do we need the differentiation betw. internal and external stakeholders?
Creates content including, specifications, architectures, designs and more using a wide range of tools (= model writer).
Model Citizen
Review content authored by model writer adding reviews, issues and other information using a Web Browser.
Model Reader
Specifies the requirements and use cases of the project. Acts also as a domain expert to contribute specialized knowledge to the project team and to ensure that the results meet business needs/goals.
Use Case Provider
Performs repository administration tasks. That includes the configuration of the modeling tools (e.g., customizing, installing UML profiles, addons, and libraries), the configuration management environment for the models and engineering artifacts (for example a repository server).
Model Admin
Has the technical expertise to build software, hardware and embedded AI models for the ECOMAI use cases.
Covers the whole data science project lifecycle. Focuses on creating the machine learning models, reviews model results, accuracy, and retrains models.
Data Scientist
Developer
Defines the methodology and provides modeling support.
Model Expert
Responsible for data collecting and data cleaning.
Deals with system design and focuses on overall system construction.
Data Engineer
Software Engineer
Focuses on implementation and specialization of one part of the system at a time.
MLOps Engineer
Hardware engineer
Software Developer
Accompanies the ML flow from beginning to end, manages collaboration between the data engineer, the data scientist and the ML engineer, and maintains continuous integration/continuous deployment (CI/CD) pipelines for ML models.
ML Engineer
Is a software engineer with ML knowledge. Deploys ML models and delivers them into production to produce business value.
Embedded Software Engeneer
Is responsible for the development and embedding of software in embedded systems.
Security Expert
_End User
_Tester
_Maintainer
_Architect
_Integrator


uc ECOMAI Scenario (following the Layered Approach)
ECOMAI
Model the business objects
Analyse and model the stakeholders
The Business View defines the business objects and actors of the ECOMA project.
Decompose and model functional requirements
Model use cases
The Functional View defines use cases and functional requirements.
Model the system context
The System Context View defines the system under development with human actors and dependencies to external systems.
Model system decomposition
The System Decomposition View defines the overall architecture and technology decision.
Model the software stack
Specify the AI-enhanced control patterns
Decide on key definitions for AI modeling
Setup IoT-PML
Perform model training
Perform preprocessing / preparation
Perform model evaluation
Maintain CI/CD pipeline for AI models
The Software Decomposition defines IoT hardware, software stack, non-functional requirements and AI components.
Model the threat model
Provide modeling consulting and review model
Other Views (e.g. Security View) or general ECOMAI scope.
Model Citizen
(from Actors)
Use Case Provider
(from Actors)
Developer
(from Actors)
Software Engineer
(from Actors)
Model Admin
(from Actors)
Security Expert
(from Actors)
Model Expert
(from
:Data Scientist
:Data Engineer
:MLOps Engineer
:ML Engineer
do we need this one?
Model the application code (optional)
Software Developer
(from Actors)


pkg ECOMAI Layered Approach
ECOMAI Architecture
Business View
Domain Model
[Enterprise Architect]
The Domain Model provides a view of all the objects that make up an area of interest, and their relationships. It is used to capture the significant objects and types within a system, an organization, or any target domain.
?? what about the workflows ??
Actor Model
[Enterprise Architect]
The Actor model gives an overview of all stakeholders, denoted as human actors, which are actively involved in a (development) project or whose interest might be affected as a result of the project.
reuse
Functional View
Requirement Model
[Enterprise Architect]
The Requirements model specifies what the system is supposed to do in form of (functional) requirements and the relationships between requirements.
transforms
Use Case Model
[Enterprise Architect]
The Use Case model defines the system's intended functionalities (use cases) and the relationship between them from a user's viewpoint.
Note: corresponds to "Scenario" of the 4+1 view model [Kru95].
System Context View
System Context
The System Context view facilitates a common understanding of
The ECOMAI Architecture shows the layered approach as well as the envisaged structure for model-based development in ECOMAI inspired by the C4 methodology and based on different IoT-PML viewpoints.
The layered approach has the following advantages:
Users are enabled to see the big picture at any time through identifying (also external) neighbors.
Element types indicate the level of abstraction.
Traceability dependencies allow to keep track between internal and external parts of the system through all layers.
Tools used for modeling and developing the different artifacts are shown in brackets: The modeling tool Enterprise Architect provides appropriate diagrams for each layer (or view) and toolboxes for these diagrams, containing modeling elements and relationships with own semantics TBD MATLAB, Simulink, TVM, ...
? TBD - maybe later: add basic information for each level /main diagram (in element Notes):
scope OK (visible note)
(diagrams)
primary elements
supporting elements

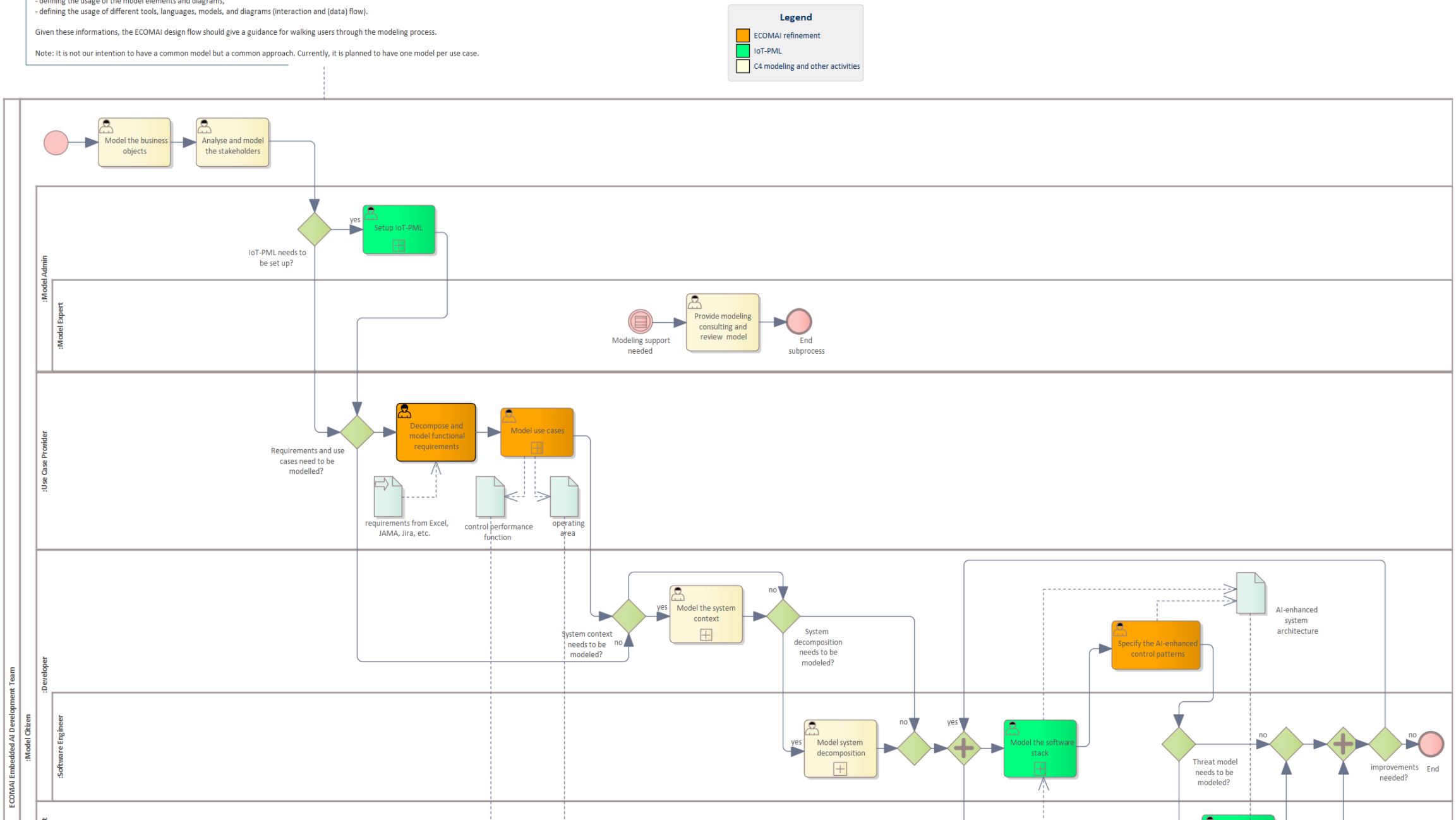
Business Process AI-PML

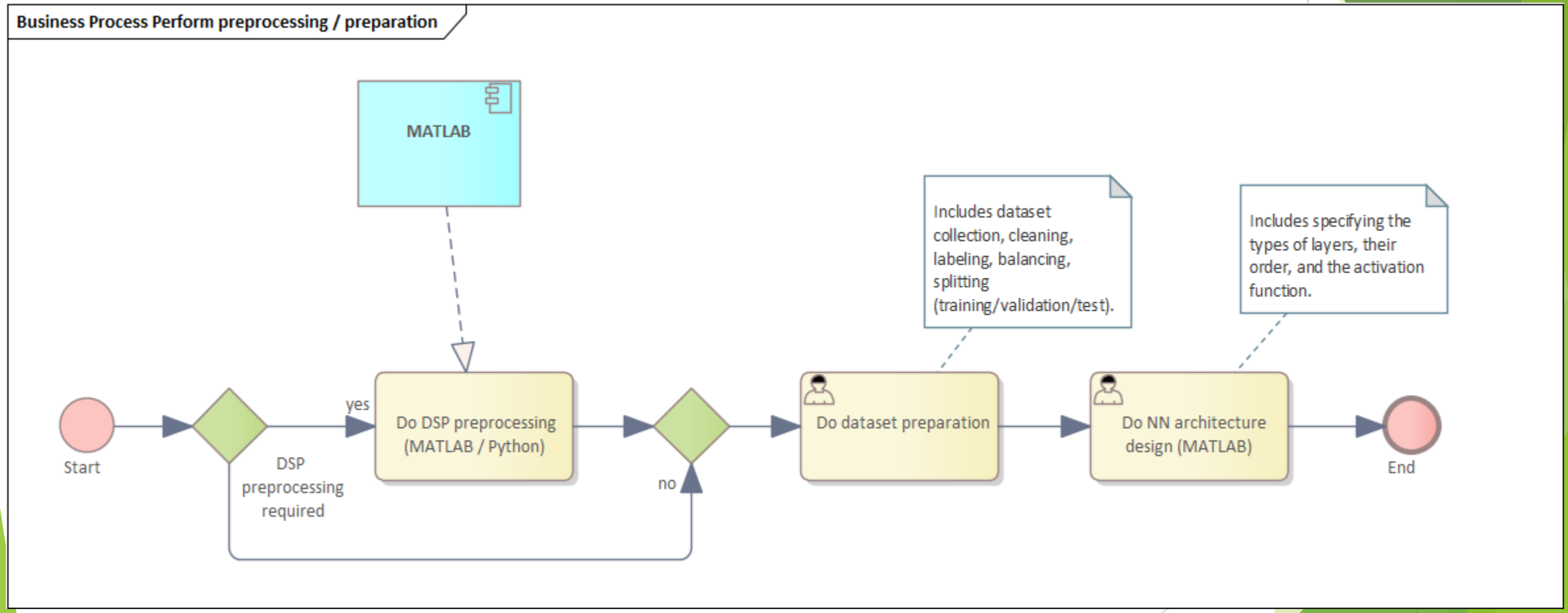
Describes the top-down ECOMAI design flow on how to build models using the "AI-PML" (methodology).

The goal of this sub-task is to document the ECOMAI development process, which includes
- defining the steps and responsibilities (user roles) for model-based development in the ECOMAI Design Kit,
- defining the usage of the model elements and diagrams,
- defining the usage of different tools, languages, models, and diagrams (interaction and (data) flow).

Given these informations, the ECOMAI design flow should give a guidance for walking users through the modeling process.

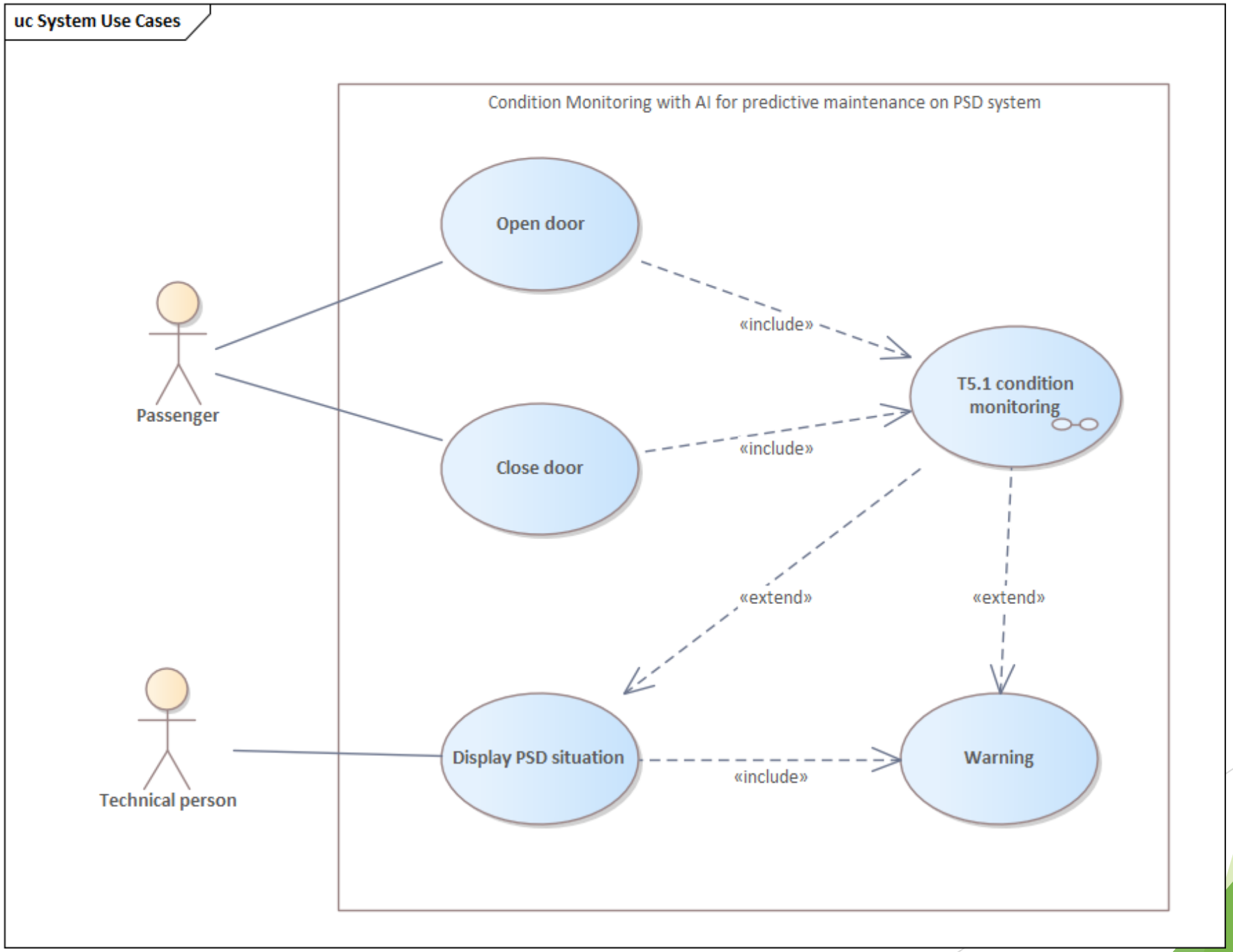
Note: It is not our intention to have a common model but a common approach. Currently, it is planned to have one model per use case.

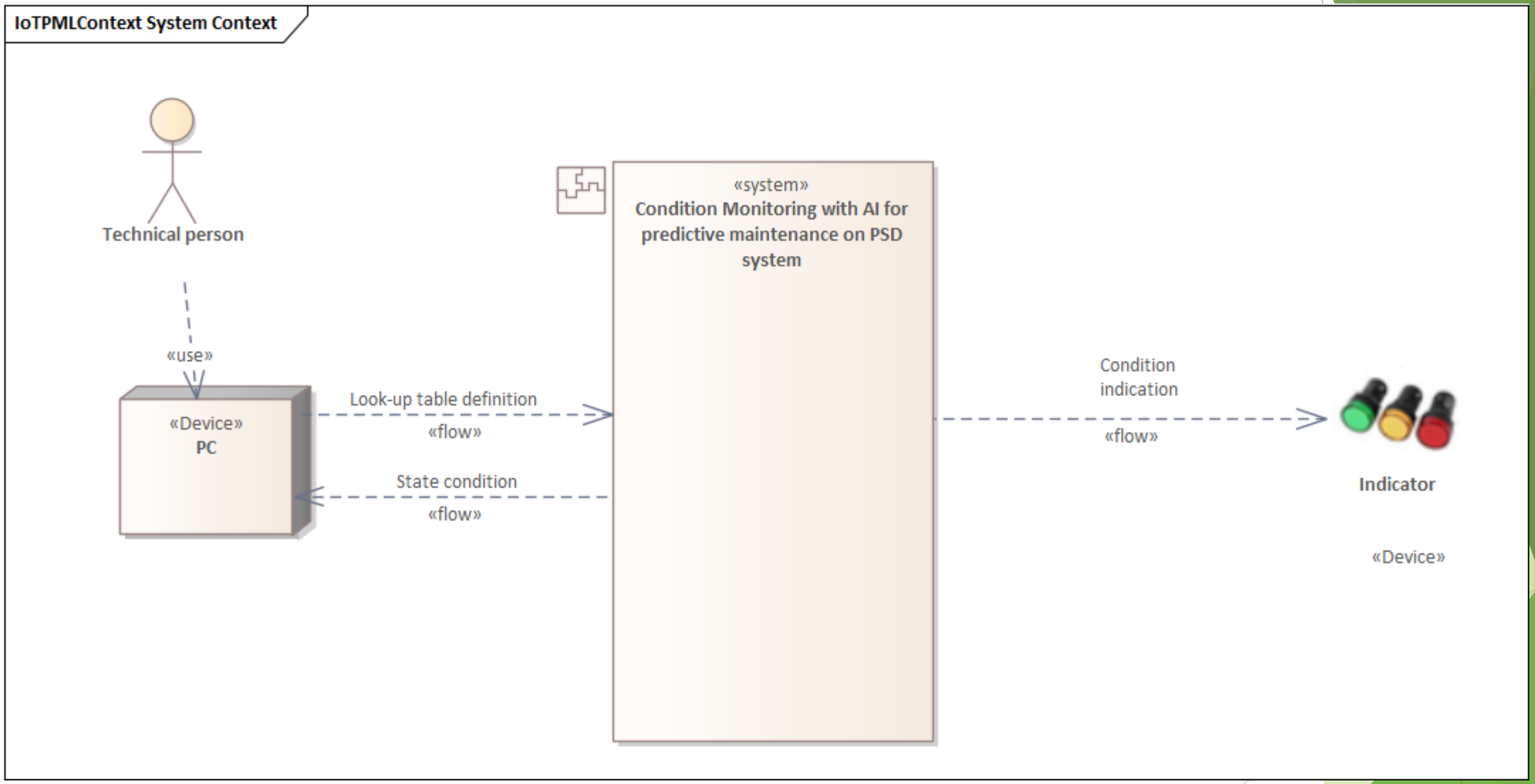


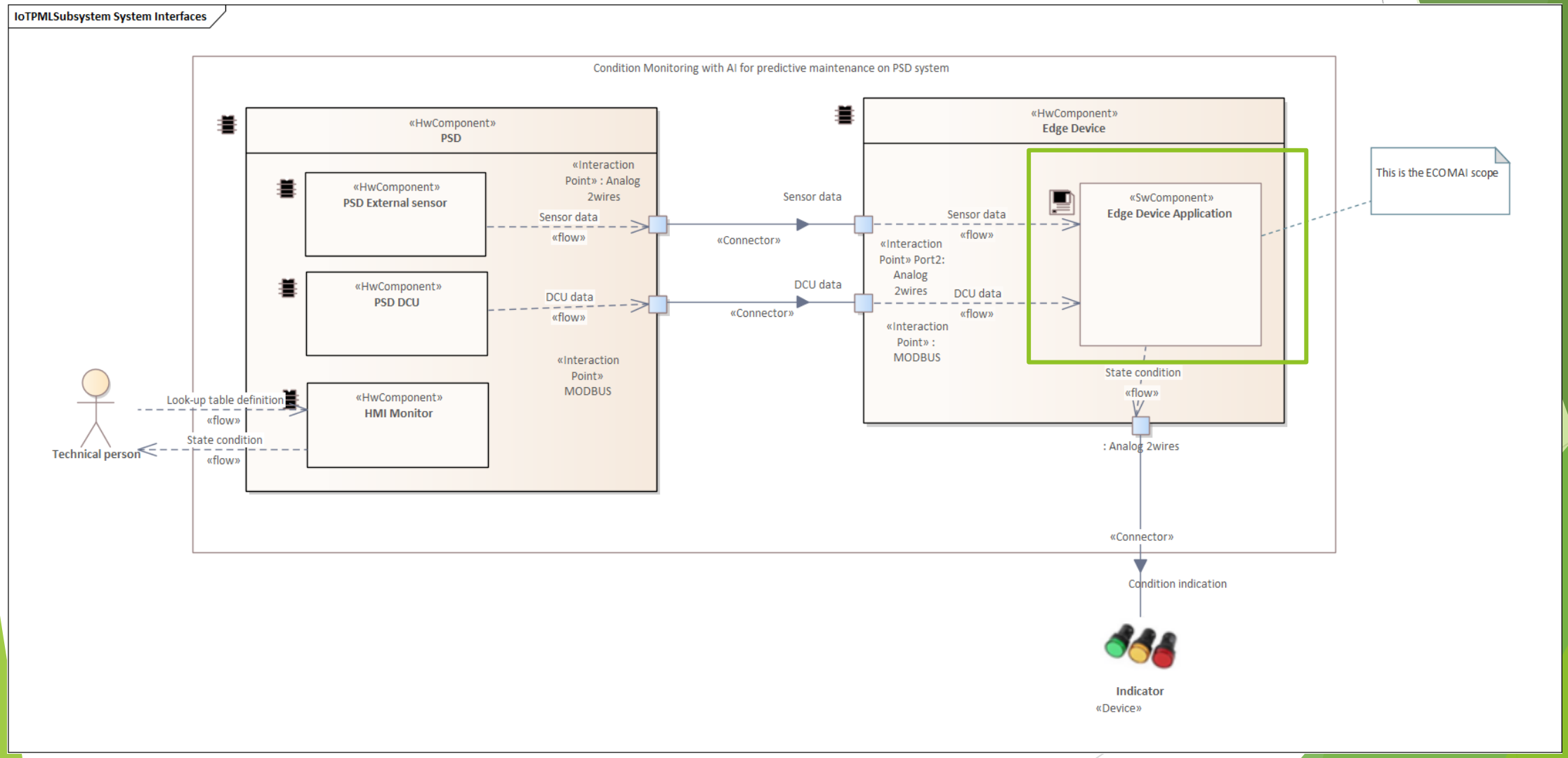

Business Process Perform preprocessing / preparation
MATLAB
Includes dataset collection, cleaning, labeling, balancing, splitting (training/validation/test).
Includes specifying the types of layers, their order, and the activation function.
Start
DSP preprocessing required
yes
Do DSP preprocessing (MATLAB / Python)
no
Do dataset preparation
Do NN architecture design (MATLAB)
End

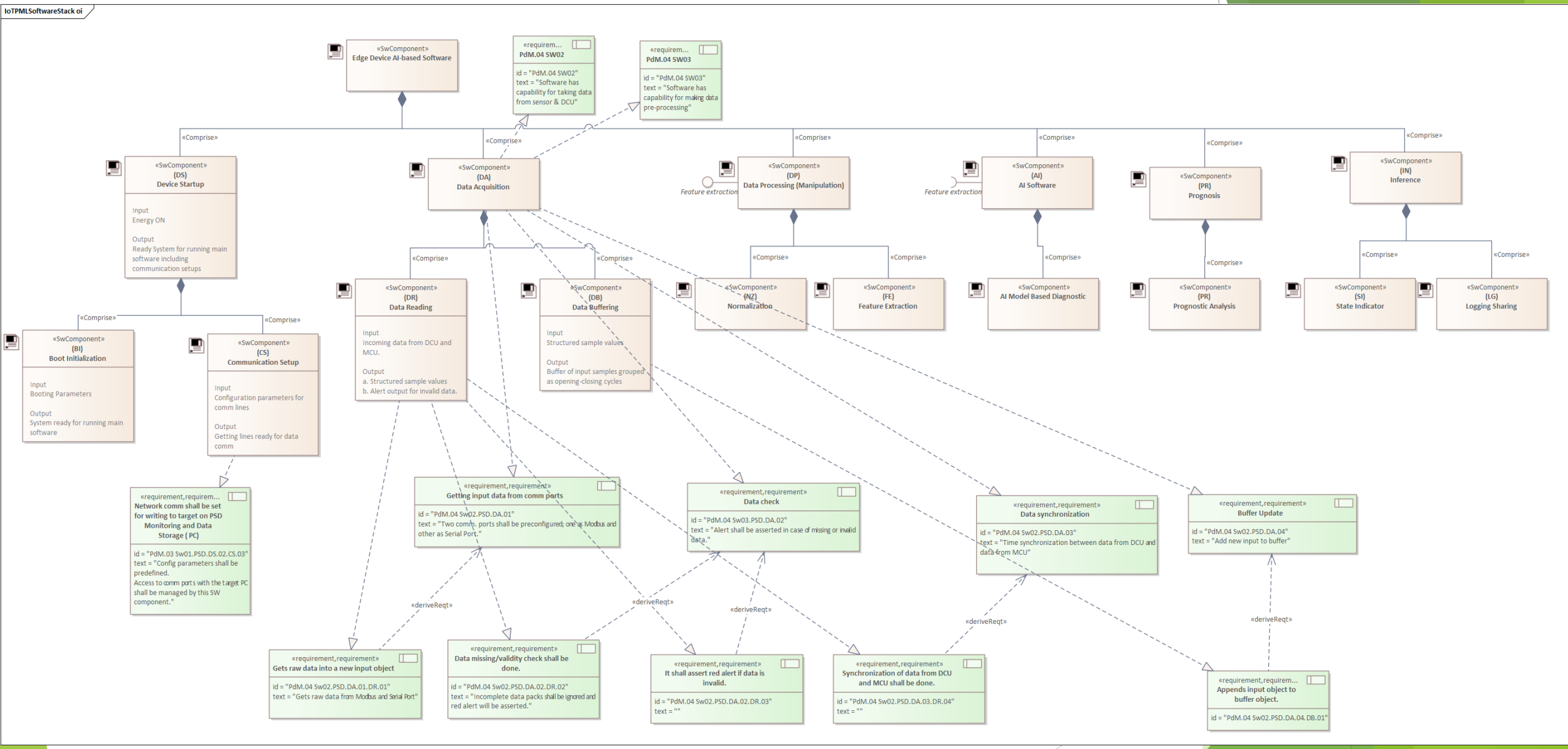
Alb

- Requirements (as SysML Requirements)
- Identification of Use Cases
- Usage of IoT-PML
- Identification of Scope of ECOMAI

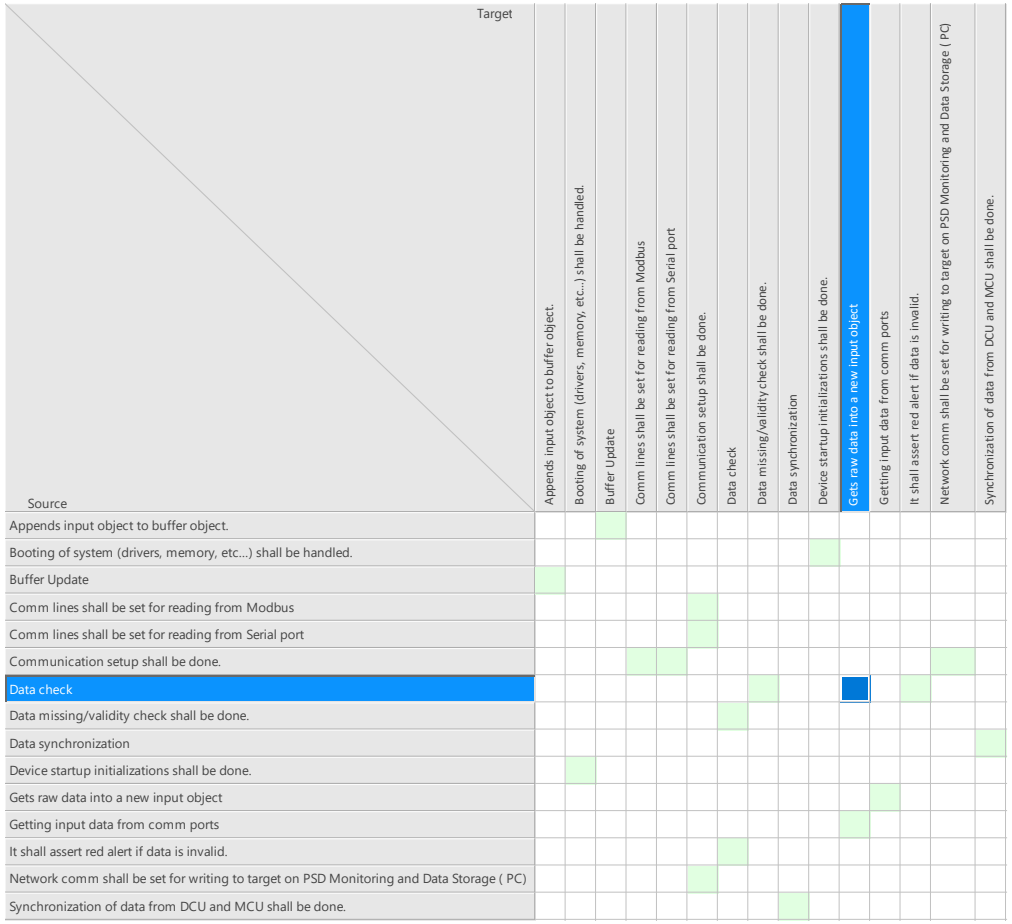

uc System Use Cases
Condition Monitoring with AI for predictive maintenance on PSD system
Open door
Close door
Passenger
«include»
«include»
T5.1 condition monitoring
«extend»
«extend»
Technical person
Display PSD situation
«include»
Warning


IoTPMLContext System Context
Technical person
«use»
«Device»
PC
Look-up table definition
«flow»
State condition
«flow»
«system»
Condition Monitoring with AI for predictive maintenance on PSD system
Condition indication
«flow»
Indicator
«Device»

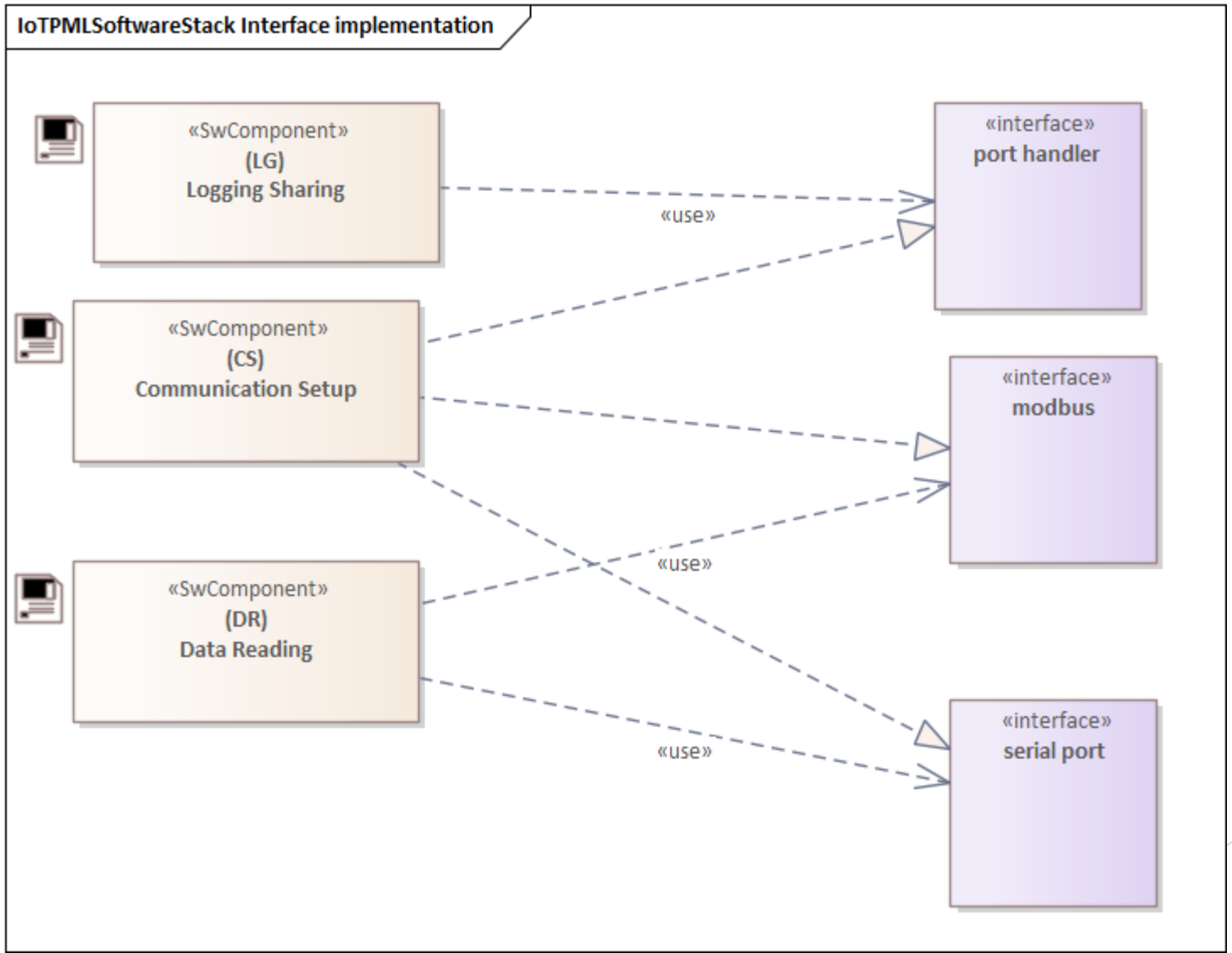

IoTPMLSubsystem System Interfaces
Condition Monitoring with AI for predictive maintenance on PSD system
«HwComponent»
PSD
«HwComponent»
PSD External sensor
«HwComponent»
PSD DCU
«HwComponent»
HMI Monitor
«Interaction Point» : Analog 2wires
«Interaction Point» MODBUS
Sensor data
«flow»
DCU data
«flow»
Look-up table definition
«flow»
State condition
«flow»
Technical person
Sensor data
«Connector»
DCU data
«Connector»
«HwComponent»
Edge Device
«Interaction Point» Port2: Analog 2wires
«Interaction Point» : MODBUS
Sensor data
«flow»
DCU data
«flow»
«SwComponent»
Edge Device Application
State condition
«flow»
: Analog 2wires
«Connector»
Condition indication
Indicator
«Device»
This is the ECOMAI scope


IoTPMLSoftwareStack oi
«SwComponent»
Edge Device AI-based Software
«requirem...
PdM.04 SW02
id = "PdM.04 SW02"
text = "Software has capability for taking data from sensor & DCU"
«requirem...
PdM.04 SW03
id = "PdM.04 SW03"
text = "Software has capability for making data pre-processing"
«Comprise»
«SwComponent»
(DS)
Device Startup
Input
Energy ON
Output
Ready System for running main software including communication setups
«SwComponent»
(DA)
Data Acquisition
Feature extraction
«SwComponent»
(DP)
Data Processing (Manipulation)
Feature extraction
«SwComponent»
(AI)
AI Software
«SwComponent»
(PR)
Prognosis
«SwComponent»
(IN)
Inference
«SwComponent»
(BI)
Boot Initialization
Input
Booting Parameters
Output
System ready for running main software
«SwComponent»
(CS)
Communication Setup
Input
Configuration parameters for comm lines
Output
Getting lines ready for data comm
«SwComponent»
(DR)
Data Reading
Input
Incoming data from DCU and MCU.
Output
a. Structured sample values
b. Alert output for invalid data.
«SwComponent»
(DB)
Data Buffering
Input
Structured sample values
Output
Buffer of input samples grouped as opening-closing cycles
«SwComponent»
(NZ)
Normalization
«SwComponent»
(FE)
Feature Extraction
«SwComponent»
AI Model Based Diagnostic
«SwComponent»
(PR)
Prognostic Analysis
«SwComponent»
(SI)
State Indicator
«SwComponent»
(LG)
Logging Sharing
«requirement,requirem...
Network comm shall be set for writing to target on PSD Monitoring and Data Storage (PC)
id = "PdM.03 Sw01.PSD.DS.02.CS.03"
text = "Config parameters shall be predefined.
Access to comm ports with the target PC shall be managed by this SW component."
«requirement,requirement»
Getting input data from comm ports
id = "PdM.04 Sw02.PSD.DA.01"
text = "Two comm. ports shall be preconfigured; one as Modbus and other as Serial Port."
«requirement,requirement»
Data check
id = "PdM.04 Sw03.PSD.DA.02"
text = "Alert shall be asserted in case of missing or invalid data."
«requirement,requirement»
Data synchronization
id = "PdM.04 Sw02.PSD.DA.03"
text = "Time synchronization between data from DCU and data from MCU"
«requirement,requirement»
Buffer Update
id = "PdM.04 Sw02.PSD.DA.04"
text = "Add new input to buffer"
«deriveReqt»
«requirement,requirement»
Gets raw data into a new input object
id = "PdM.04 Sw02.PSD.DA.01.DR.01"
text = "Gets raw data from Modbus and Serial Port"
«requirement,requirement»
Data missing/validity check shall be done.
id = "PdM.04 Sw02.PSD.DA.02.DR.02"
text = "Incomplete data packs shall be ignored and red alert will be asserted."
«requirement,requirement»
It shall assert red alert if data is invalid.
id = "PdM.04 Sw02.PSD.DA.02.DR.03"
text = ""
«requirement,requirement»
Synchronization of data from DCU and MCU shall be done.
id = "PdM.04 Sw02.PSD.DA.03.DR.04"
text = ""
«requirement,requirem...
Appends input object to buffer object.
id = "PdM.04 Sw02.PSD.DA.04.DB.01"

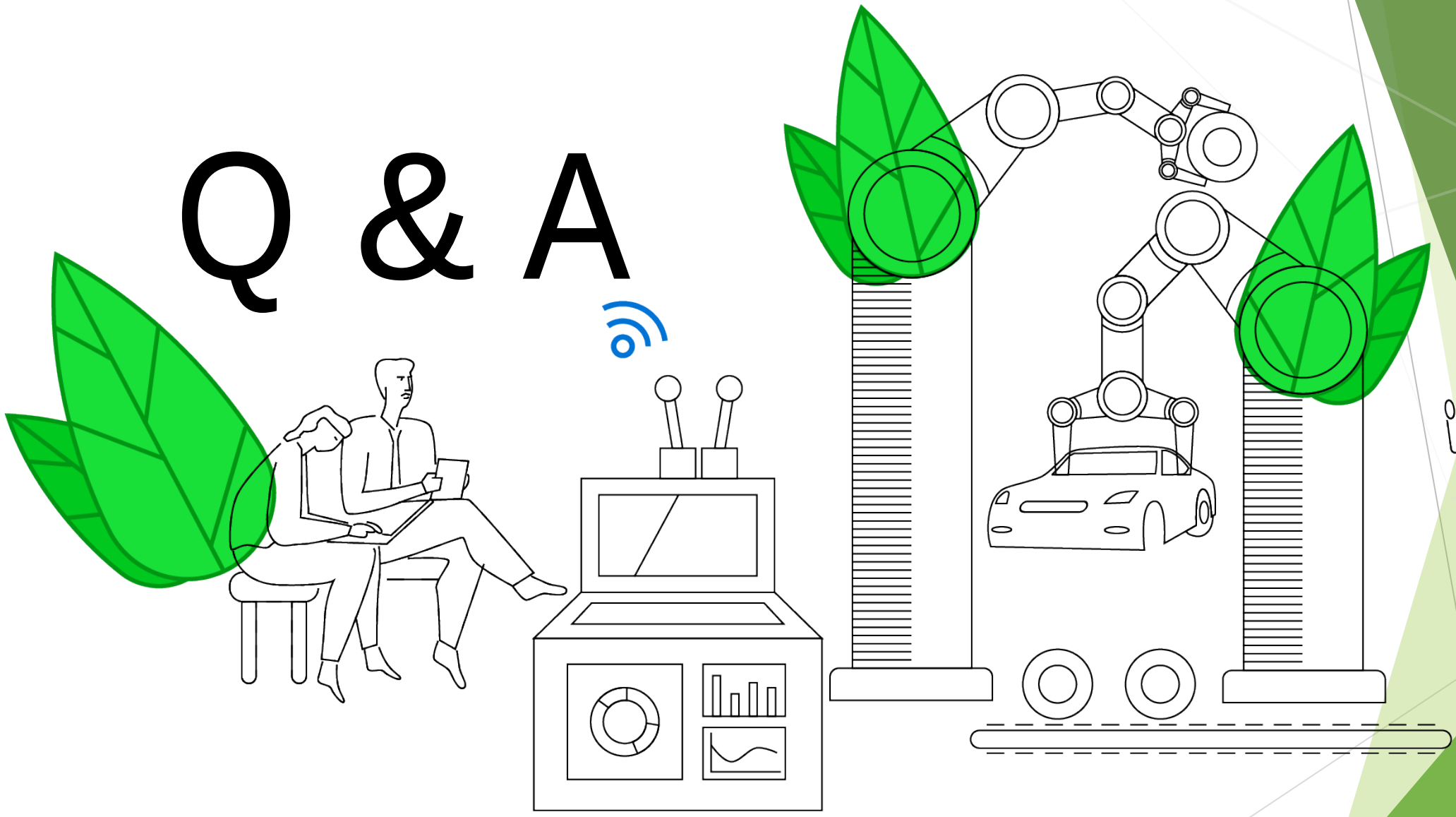
Example of REQs traceability



Interface implementation



Q & A



ECOMAI