SAFEXPLAIN

Safe and Explainable
Critical Embedded Systems based on AI

Platform support to enable safety-relevant AI-based systems

Enrico Mezzetti - BSC

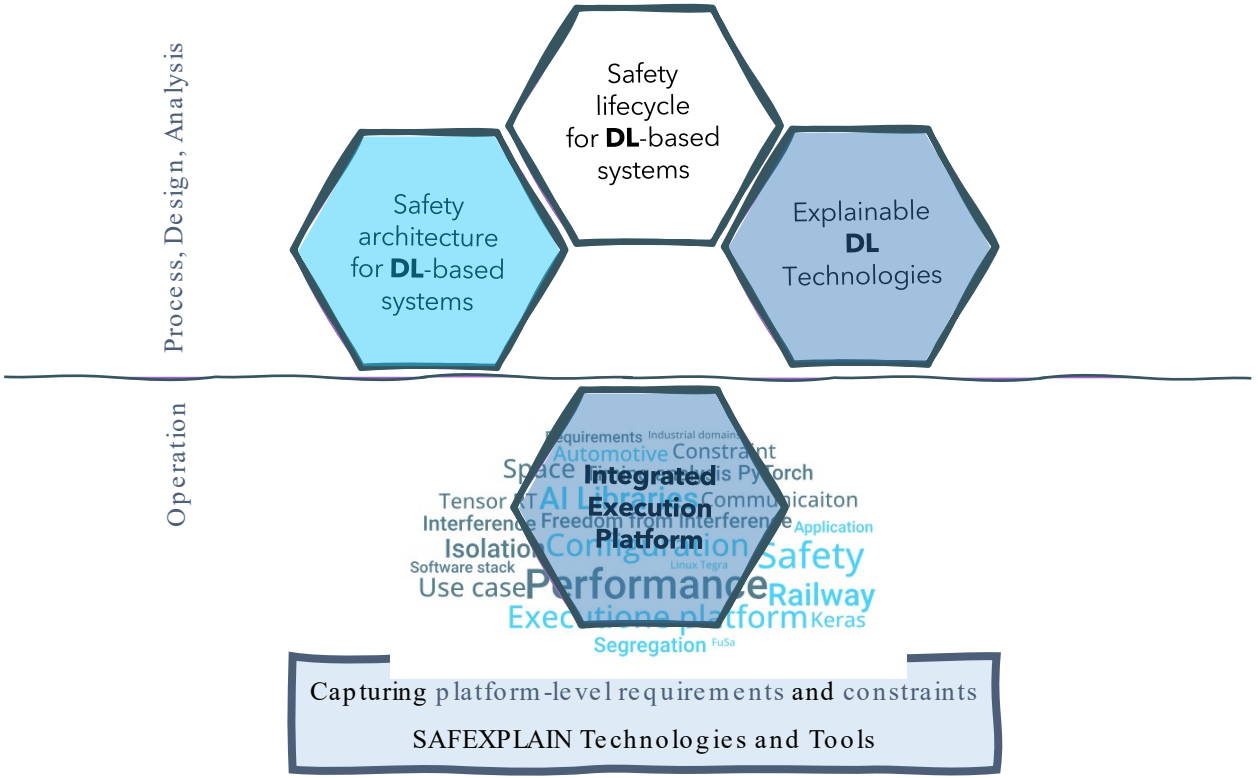
TRUSTWORTHY AI IN SAFETY CRITICAL SYSTEMS
OVERCOMING ADOPTION BARRIERS

23 SEPTEMBER 2025 | BARCELONA, SPAIN

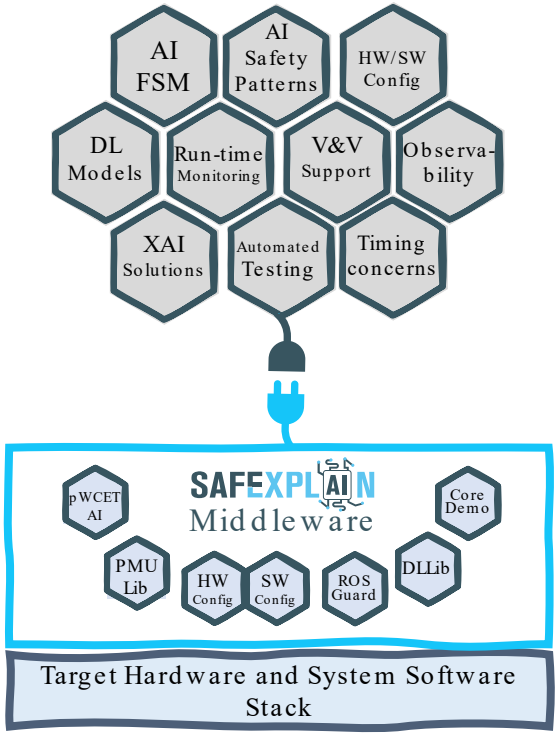
Funded by the European Union

This project has received funding from the European Union's Horizon Europe programme under grant agreement number 101069595.

SAFEXPLAIN


Process, Design, Analysis
Safety lifecycle for DL-based systems
Safety architecture for DL-based systems
Explainable DL Technologies
Operation
Integrated Execution Platform
Requirements
Industrial domain
Automotive
Constraint
Space
PyTorch
Tensor RT
AI Libraries
Communicaiton
Interference
Freedom from interference
Application
Isolation
Configuration
Safety
Software stack
Linux Tegra
Use case
Performance
Railway
Execution platform
Keras
Segregation
FuSa
Capturing platform-level requirements and constraints
SAFEXPLAIN Technologies and Tools

Platform-level Support and Tools



- **Capturing a diverse set of platform level objectives**
  - Support ***HW/SW*** *integration* of proposed solutions
    - **FuSa** SW Architecture and Patterns
    - **XAI** components
  - Provide an *abstraction layer* for system **configuration**
    - HW and SW setup for **performance** and **FuSa requirements**
  - Offer a *common layer* for SW **V&V** and FuSa mechanisms
    - Functional **testing** and **timing characterization**
    - Diagnostic and monitoring concept, health manager, etc.
  - Offer a *reference common stack* to Use Cases

SAFEXPLAIN

SAFEXPLAIN HW/System SW Platform

NVIDIA Jetson AGX Orin (32GB Dev Kit)

Computing requirements of complex AI-based systems

200 TOPS of AI performance for autonomous systems

NVIDIA Ampere GPU

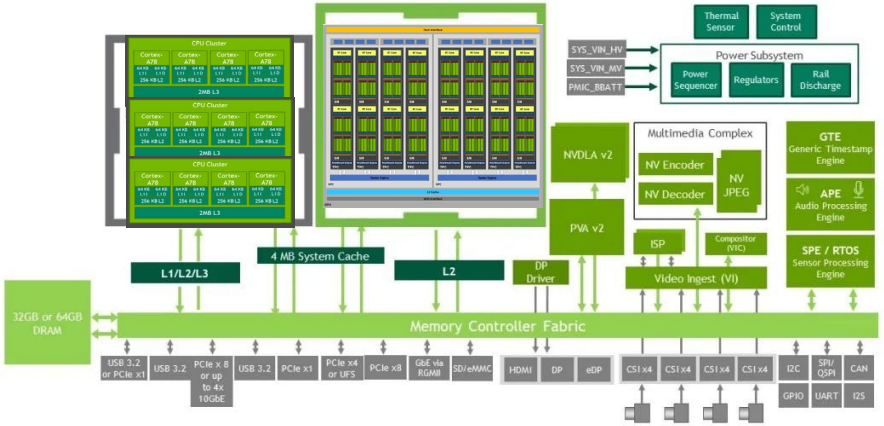
Arm® Cortex®-A78AE CPU

Next-gen accelerators for DL and Vision (NVDLA, PVA)

Video encoder and decoder

High speed I/O, 204 GB/s of memory bandwidth

32GB of DRAM + 2TB NVMe



NVIDIA reference SW stack

Jetson Linux 36.3 (Ubuntu 22.04)

Linux Tegra 5.15

JetPack 6.0.1 SDK

SAFEXPLAIN Middleware concept

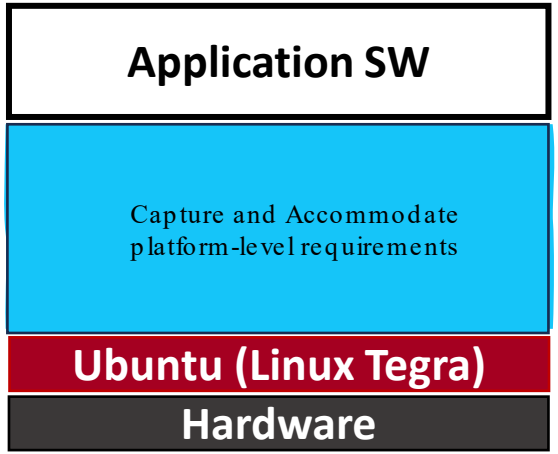
- **Fundamental enabler for SAFEXPLAIN methodology**
  - *@Design, development and testing*
    - Ensure Safety Patterns by design
    - Enable performance Monitoring
    - Support test automation and V&V support
  - *@Operation*
    - Enforce HW/SW configuration
    - FuSa mechanism
      - Monitoring and diagnostic
      - Timing interference and resource usage monitoring
    - Integrate SAFEXPLAIN XAI solutions and tools
- **Building on ROS2 concepts and support**
  - Modular design and scalability (pub-sub semantics)
  - Native support for FuSa and V&V
    - Health manager
    - Software validation (test framework, automation, assertions, …)

Performance

Other non-functional requirements

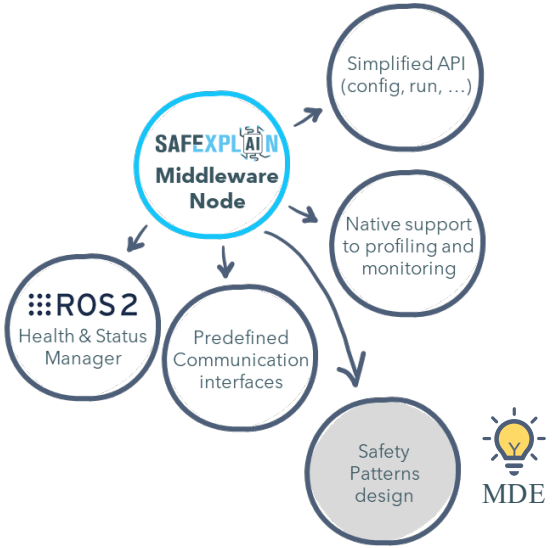
Functional requirements

XAI on DL components

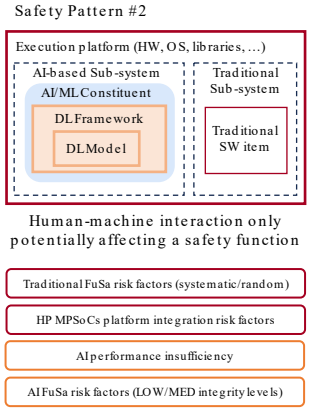


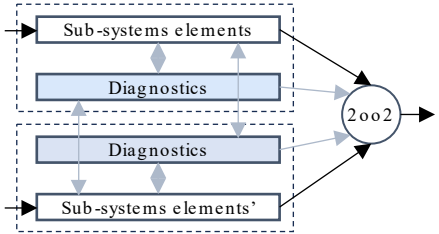
SAFEXPLAIN Middleware realization

- **Middleware nodes**
  - Set of node archetypes as specializations (wrapper) of ROS2 nodes
    - Communication patterns – data flows
    - Bespoke monitoring mechanism
    - Simplified semantics specification
- **Pivotal concept** in platform-level support
  - Compliance to Safety Patterns *by design* while preserving modularity
    - Mandatory software nodes/components
    - User-defined nodes for application semantics
    - Multi-layered monitoring and diagnostic
  - Support for deployment configurations
    - Statically pre-defined setups and configurable options
  - Test integration and V&V support
  - Seamless integration of all technologies/tools



Support Safety patterns by design (FuSa+XAI)





SAFEXPLAIN

Abstraction layer for HW and SW Configuration

- **Meet FuSa AND performance requirements**
  - *FuSa compliance with Safety Patterns*
  - *Reconciling Performance and Freedom from interference*
- **Mechanisms and deployment options**
  - *HW and OS tuning*
    - *Power models, L3 Cache partitioning*
    - *OS Kernel Parmas, Scheduling options*
  - Mapping of components to computing elements
- **Generalization**
  - Tailored to execution platform but generalizable
  - Fully controlled as middleware configuration options

Tr dq Rnes v` qd

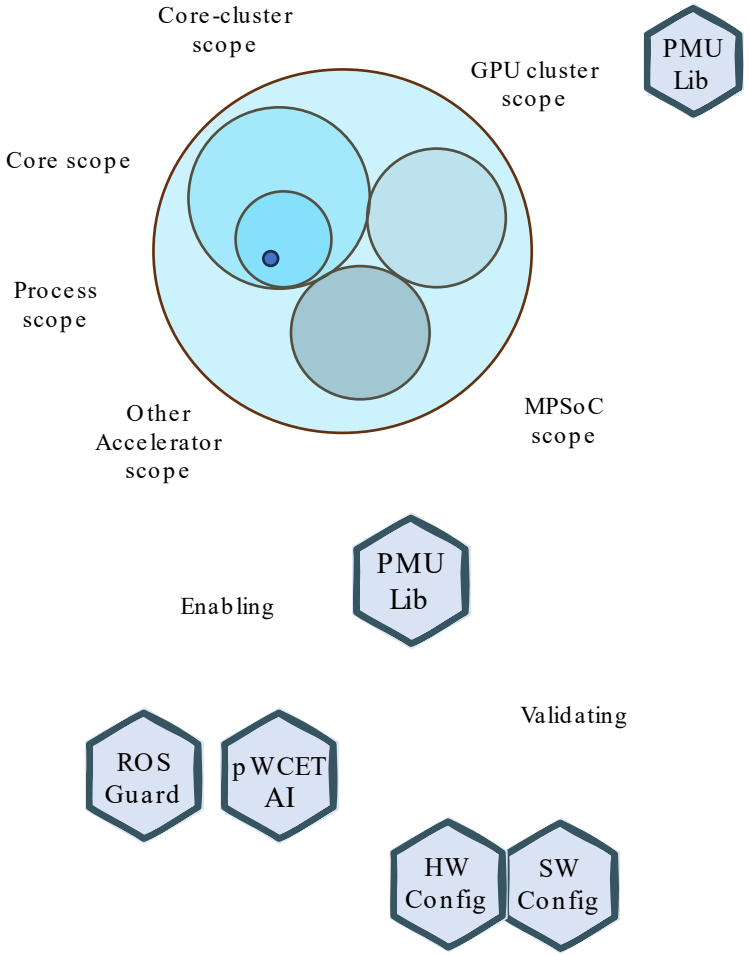
Rxr s dl
Rnes v` qd

G qcv` qd k` xdq

SAFEXPLAIN

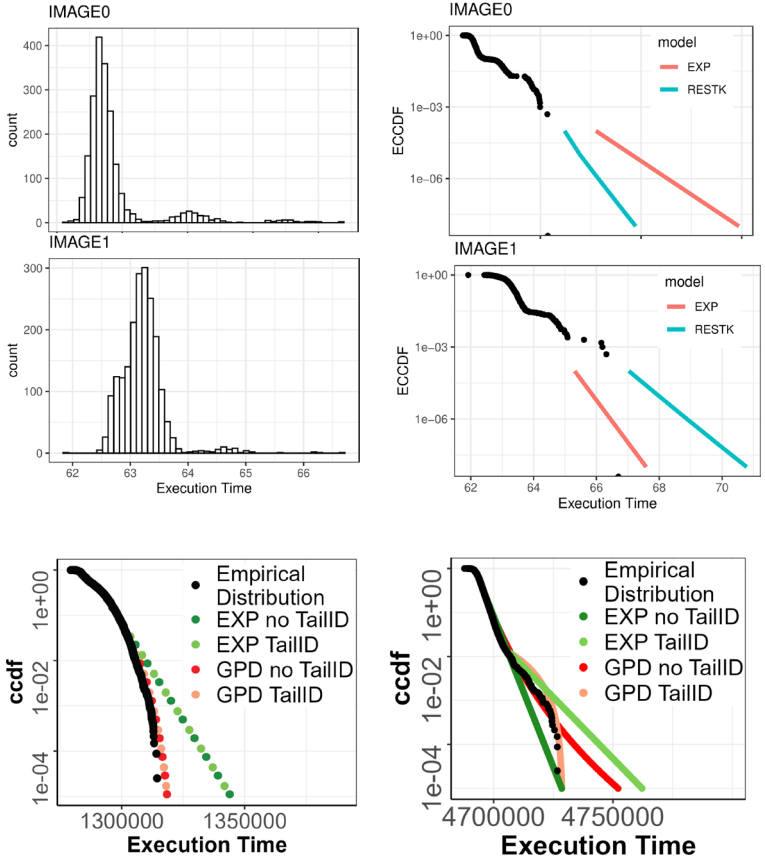
Platform-level Observability

- **Observing software behavior through Event Monitors**
  - **@Analysis time**: information gathered along program execution is exploited as supporting evidence for several analyses
  - **@Run time**: to deploy monitoring approaches to intercept and timely react to system misbehavior
    - Watchdogs, Bandwidth regulation
- **PMULib**
  - Lightweight HEM configuration/collection library
  - Support timing and interference characterization
  - Validate Safety Pattern setups
  - Integrated on SAFEXPLAIN Middleware
    - *Transparent* node-level profiling (config option)
    - User-level support (e.g. *timestamping*)
    - Multi-scopes (*process, thread, core*)



Timing characterization method for AI-based solutions

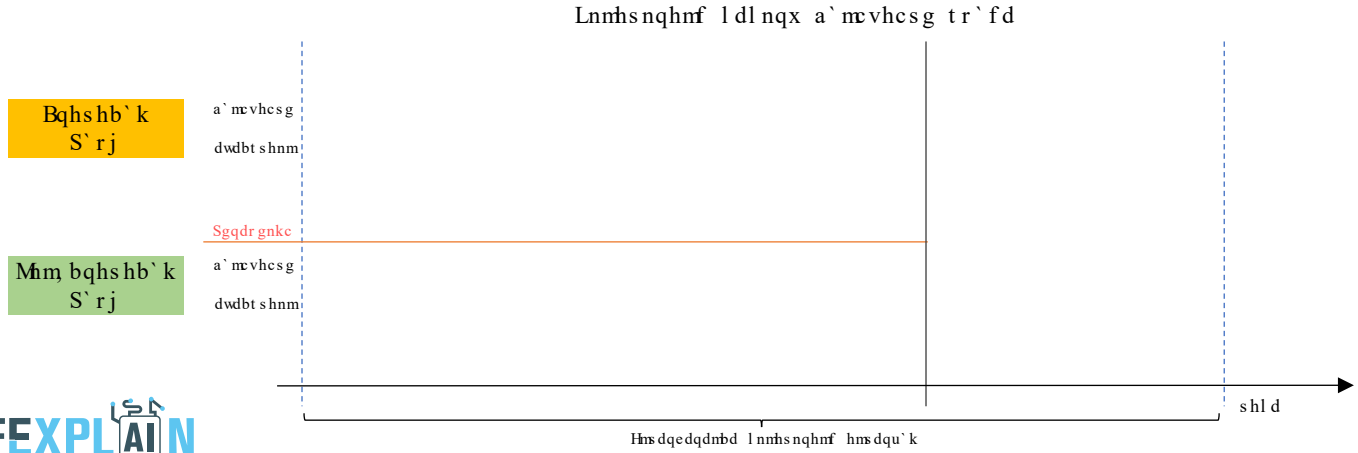
- **AI-based SW on advanced heterogeneous MPSoCs**
- **Probabilistic timing analysis techniques**
  - Increasingly considered as alternative or complementary approaches to comply with timing verification requirements in complex scenarios
  - Use of *statistical analysis tools* to derive (*p*)WCET estimates
    - Mainly based on Extreme Value theory (**EVT**)
    - Cannot properly handle **mixture distributions**
- **SAFEXPLAIN methodology**
  - **PMULib** profiling dealing w/ Inter-run Variability (**IRV**)
  - Restricted k (**RestK**)
    - Based on *Markov's inequalities* adapted to mixture distributions and integrated for on-line application
  - Intercepting lack of ID(*) in the tail (**TailID**)
    - Detecting when mixture appears in the extreme data used for estimation
    - Using confidence intervals of the Maximum Likelihood Estimator (MLE) of the GPD's Extreme Value Index (EVI)



(*) *EVT requires data to be Independent and Identical Distributed (IID)*

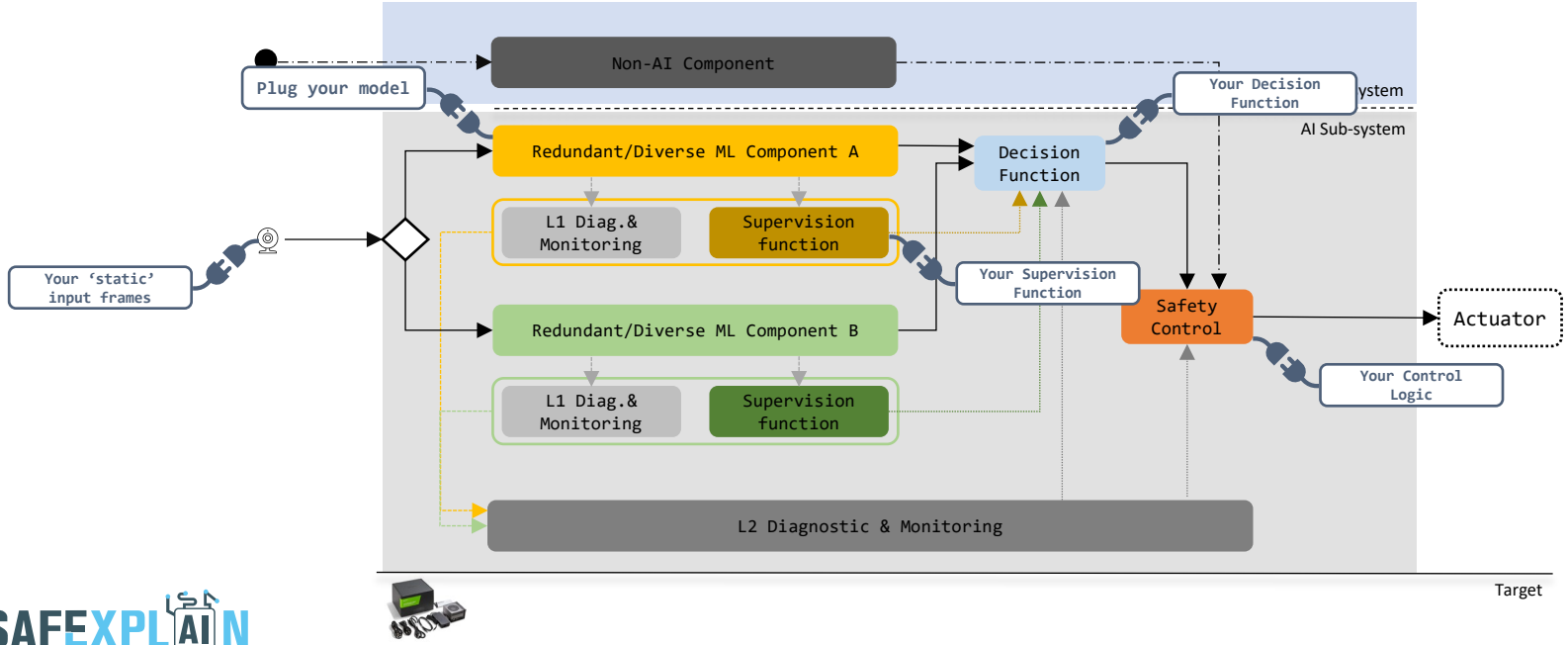
Extended support for FuSa Run-time Mechanisms

- **Execution time watchdog**
  - Capture deadline misses for critical nodes
- **ROS-based Memory bandwidth regulation**
  - Capture and control *residual* interference (after **Safety Pattern**) and HW/SW configuration
    - Non-critical nodes possibly delaying critical ones by clogging the shared resource
  - Full ***freedom from interference*** is impossible to achieve without **compromising performance**



SAFEXPLAIN CORE Demo Examples

- **Adaptable minimal cross-domain demonstrator on a Docker**
  - Small-scale, simplified, yet representative open demonstrator
    - SAFEXPLAIN technologies and tools and platform's key features
  - Replaceable building blocks to showcase specific features and scenarios

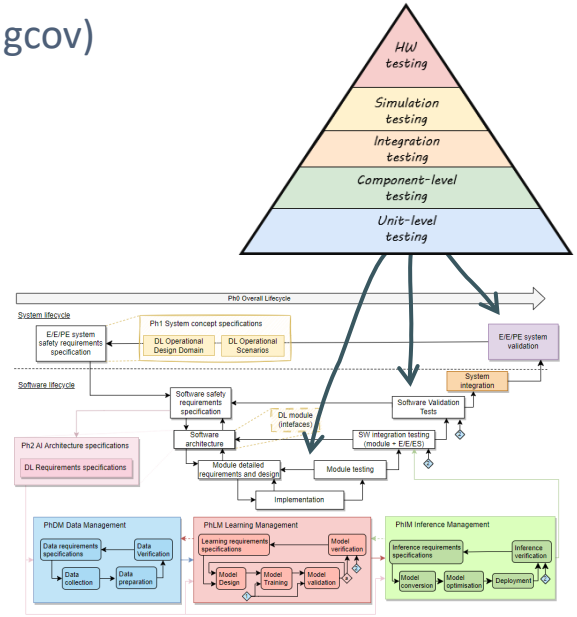


CORE Demo – Space

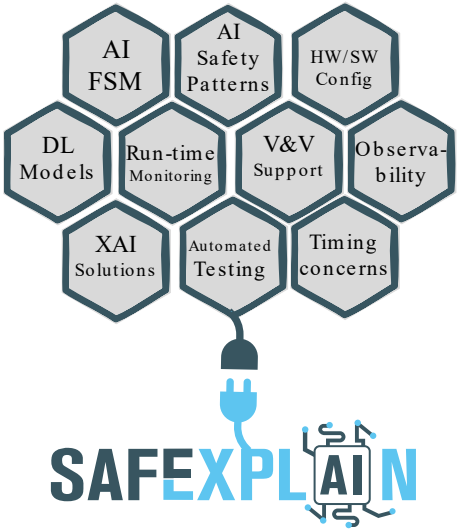
- **AIKO Open Models for Image recognition**
  - Lightweight models for object detection, trained to specifically detect the space target
  - SSDLite: a fast model using MobileNet backbone optimized for embedded environments
  - fasterRCNN: a more accurate but slower model also built on MobileNet backbone
- **Supervision function XAI**
  - Anomaly detectors (VAE): input, model activation, output
- **Decision function XAI**
  - Ensemble of model prediction (consolidated bbox)
  - Aggregate + explanations (if rejection) of supervisor outputs
    - Anomaly type and score
- **DL1 D&M Layer**
  - Input temporal consistency
- **DL2 D&M Layer**
  - Basic functionality on health and status monitoring

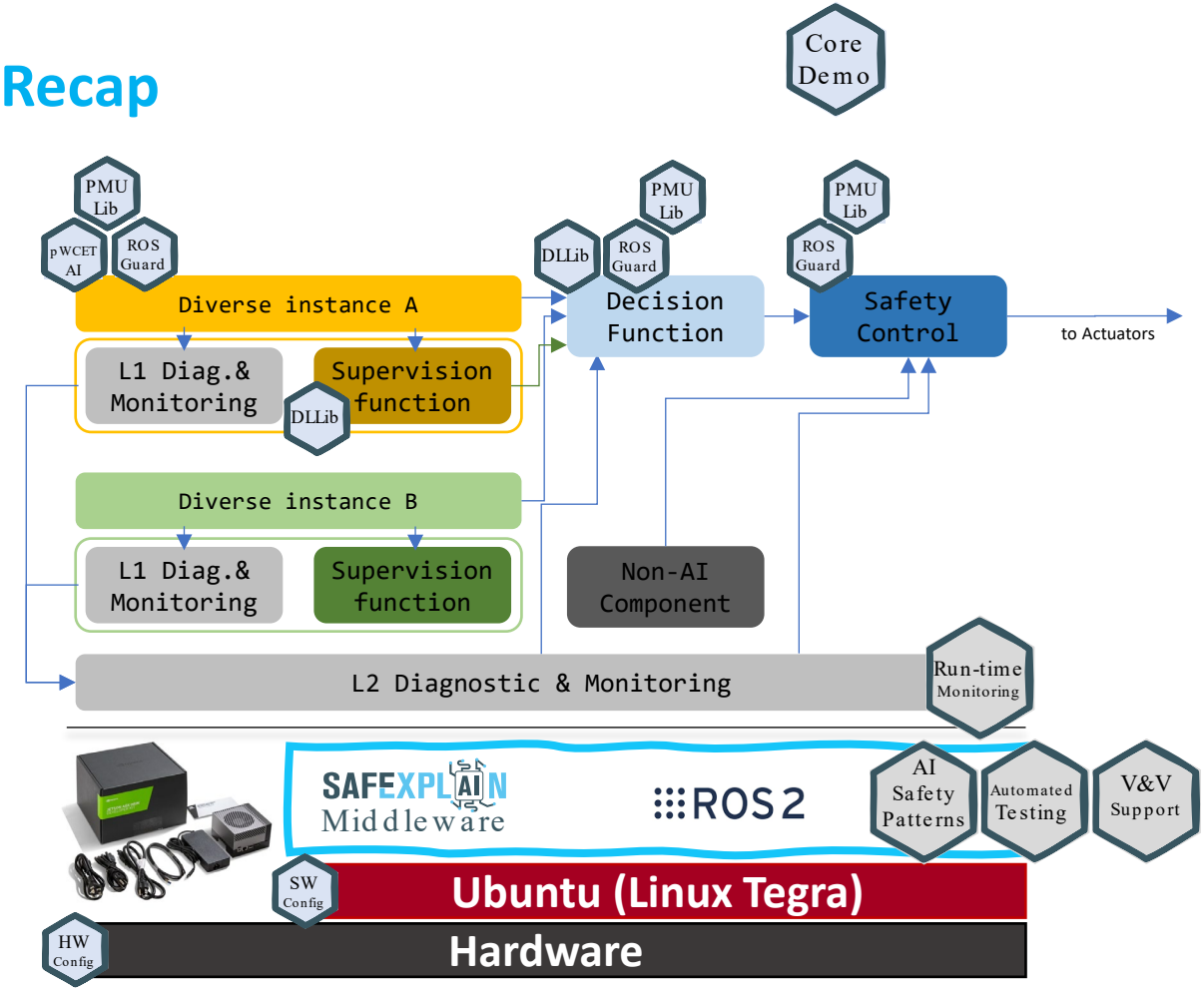
Supporting SAFEXPLAIN V&V strategy

- **Support for testing automation**
  - Unit Testing based on **gtest** (+code coverage metrics, e.g. gcov)
  - Component Testing framework
  - Integration testing based on:
    - **Roslaunch**: to setup the environment and applications
    - **Rosbag**: to replay known data
      - *Log parsing*
      - *Runtime instrumentation*
        (e.g. assertions, fault injection, etc.)
- **Support for Scenario-based strategy**
  - Triggering *selected* list of scenarios
  - Gathering information on system behavior



SAFEXPLAIN Platform Recap





THANKS!

SAFEXPLAIN

Safe and Explainable
Critical Embedded Systems based on AI

Follow us on social media:

www.safexplain.eu

Funded by
the European Union

This project has received funding from the European Union's Horizon Europe programme under grant agreement number 101069595.