A Simulation-Based Architecture Blueprint for Usage-Based Modular Devices Using Asset Administration Shells (AAS)

Research Question:
How can a standardized Asset Administration Shell (AAS) architecture be designed and evaluate to support plug-and-produce modular devices with usage-based operation in distributed manufacturing environments?

Research Topic Overview:
This research project aims to design and evaluate an architecture blueprint for modular production devices that leverages Asset Administration Shells (AAS) to enable usage-based operation and billing (e.g., cost per cycle or runtime).

The approach supports emerging industrial models such as shared, rentable devices (e.g., laser engravers or robotic arms), where equipment is billed based on usage instead of permanent ownership or fixed contracts.

This simulation-based study will focus on modelling AAS sub models that track real-time usage metrics and expose them in a standardized format to external systems such as billing or workflow engines.

Why This Research Matters:

- Modular, plug-and-produce devices are growing in industrial importance, especially for short-term deployment or rental.
- Current AAS literature lacks standardized usage-tracking and cost-mapping models.
- This research contributes a reusable architectural blueprint that aligns with Industry 4.0 goals of flexibility, modularity, and data transparency.

Planned Activities:

1. Literature Review
   - Review current AAS specifications and implementations.
   - How Pay-per-Use is used in the industry to showcase the necessity for the research
   - Identify gaps in submodel standardization for usage tracking and billing.
2. Architecture Blueprint Design
   - Define an AAS structure with a custom UsageTrackingSubmodel:
     - cyclesCompleted (int)
     - hoursUsed (float)
     - costPerCycle (float)
     - usageCost (float)

- Support with UML diagrams (Class, Sequence, Component) for clarity.

3. Simulation & Evaluation

   - Simulate the AAS behavior using Python (in Google Colab).
   - Use JSON to model submodel updates (e.g., incrementing usage counters).
   - Demonstrate billing calculation logic and simulate typical usage scenarios.

4. Optional (If time allows):

   - Add LocationSubmodel as a future enhancement.
   - Visualize usage and cost metrics in a mock dashboard (e.g., Figma or Simplifier).

Deliverables:

- A complete AAS usage-based architecture blueprint.
- Simulated runtime data and evaluation results.
- UML diagrams supporting the architectural design.
- A research paper summarizing findings, limitations, and future work.

Tools & Technologies:

- Google Colab (Python, JSON simulation)
- UML design tools (e.g., draw.io, Lucidchart)
- Optional: Eclipse BaSyx for AAS mockup
- Optional: Simplifier/Figma for dashboard mockup

Expected Outcome:

- A well-structured, reusable AAS architecture supporting usage-based billing.
- A lightweight simulation demonstrating feasibility and practical application.
- A publishable or academically valuable master’s thesis based on current industry trends.