Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira, Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria
 Published by Research Publishing, Singapore.
doi: 10.3850/ESREL2026061419_esrel26-p27806-cd

Prototype of a multi-agent architecture for real-time monitoring of personal protective equipment using an RFID-based smart gate

Federico Paolucci
Department of Engineering, University of Perugia, Italy. E-mail: federico.paolucci@dottorandi.unipg.it

Paolo Bosetti
Department of Industrial Engineering, University of Trento, Italy. E-mail: paolo.bosetti@unitn.it

Tiberio Truffarelli
Department of Engineering, University of Perugia, Italy. E-mail: tiberio.truffarelli@dottorandi.unipg.it

Roberto Marsili
Department of Engineering, University of Perugia, Italy. E-mail: roberto.marsili@unipg.it

Anna Carla Araujo
Institut Clément Ader, Université Toulouse III – Paul Sabatier, France. E-mail: araujo@insa-toulouse.fr

Gianluca Rossi
Department of Engineering, University of Perugia, Italy. E-mail: gianluca.rossi@unipg.it

Ensuring operator health and safety in complex industrial environments requires continuous verification of Personal Protective Equipment (PPE) usage. This work presents a prototype of a distributed monitoring framework based on the MADS-NET multi-agent architecture, integrating heterogeneous RFID sensors for real-time PPE detection and authentication at a smart gate. The gate implemented using a Raspberry Pi 5, hosts dedicated agents that collect data from High-Frequency (MFRC522) and Ultra-High-Frequency (UHF) RFID readers, operating at different sampling rates and detection ranges. These agents locally assess PPE compliance against predefined thresholds and communicate via a wireless broker with temporal synchronization, enabling a scalable and modular design that can be extended to multiple gates. A dedicated node handles real-time visualization of detection events and signal strength metrics, while a logger node stores data in a no-SQL database for auditing and historical analysis with configurable logging frequency. The prototype demonstrates low latency in data exchange, high repeatability in detection, and high adaptability to different industrial layouts. Performance indicators (including response time, read rate, and false positive/negative rates) confirm the measurement reliability of the system. Preliminary results validate the feasibility of the proposed decentralized approach with quantitative accuracy metrics, paving the way for future deployment of a fully distributed, multi-gate safety monitoring system in intelligent manufacturing environments.

Keywords: *Multi-agent system, Personal Protective Equipment (PPE), Real-time monitoring, Industrial safety, Heterogeneous RFID sensors*

.

1. Introduction

In complex industrial environments, worker safety critically depends on proper use of Personal Protective Equipment (PPE). Traditional manual inspections are time-consuming, subjective, and error-prone (Carbonari et al., 2011; Giretti et al., 2009; Li et al., 2015). As manufacturing shifts toward Industry 4.0 (Hermann et al., 2016; Lee et al., 2015), there is growing need for automated, real-time PPE monitoring across workers and zones while ensuring low latency, reliability, and adaptability. Centralized systems pose single points of failure and scalability limits, making them unsuitable for safety-critical applications. This work proposes a distributed multi-agent prototype based on the MADS-NET framework for real-time PPE monitoring using smart RFID gates. Heterogeneous RFID technologies detect different PPE types, with specialized agents processing data to provide immediate feedback. The main contribution is a distributed multi-agent approach enabling graceful degradation: unlike monolithic architectures, performance degrades progressively, maintaining partial functionality when components fail, crucial for safety-critical systems. A dynamic windowing technique for RFID acquisition adapts debounce intervals based on signal stability, improving accuracy and reducing false readings. Existing PPE monitoring approaches include vision-based systems, wearable sensors, and RFID-based access control. Vision methods, though non-intrusive, suffer from occlusion and lighting variability (Fang et al., 2018; Mneymneh et al., 2019; Park & Brilakis, 2012). Wearables require battery management and can be uncomfortable (Valero et al., 2016). RFID solutions balance accuracy and usability (Kelm et al., 2013) but mostly rely on centralized architectures. Unlike these, the proposed system targets real-time distributed safety monitoring. Multi-agent systems have proven effective in manufacturing for scheduling, quality control, and logistics (Barenji et al., 2016). Their distributed nature allows modularity, fault tolerance, and scalable expansion, yet real-time safety applications remain underexplored. RFID technologies operate across multiple bands: High-Frequency (13.56 MHz) systems like MFRC522 enable short-range detection (under 10 cm) suitable for gloves, while Ultra-High Frequency (860–960 MHz) systems allow longer ranges (up to meters) but are sensitive to metal interference (Goodrum et al., 2006). This work combines HF and UHF RFID to create a comprehensive, adaptable PPE monitoring solution.

2. System Architecture

2.1. MADS-NET framework overview

The system is built on MADS-NET (Multi-Agent Distributed System for Networked Environments), a lightweight framework designed by the authors for implementing distributed agent networks. MADS-NET provides several advantages over alternatives like ROS2: significantly smaller footprint, gentler learning curve, and better cross-platform compatibility including deployment on resource-constrained devices like Raspberry Pi. The framework uses ZeroMQ as its communication layer, offering flexibility in network topology and message patterns not available in DDS-based systems. Unlike monolithic architectures, MADS-NET employs a plugin-based approach where agents can be implemented in Python for rapid prototyping or in C++ for performance-critical operations. Communication between agents occurs through a lightweight broker that handles message routing without imposing rigid coupling between components. Before implementing the smart gate, the system was tested on the initial hardware architecture shown in the figure 1.

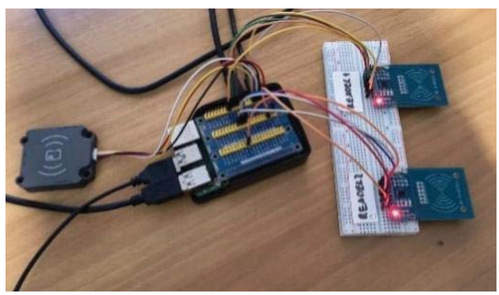

Fig. 1. Initial hardware architecture: Raspberry Pi 5 + 1 UHF RFID reader + 2 LF RFID readers

2.2. Smart gate implementation

The smart gate (figures 2,3,4) serves as a controlled passage for operator transit for a CNC workstation, built around a lightweight frame approximately 2 meters tall. The gate frame is made of steel, and the operator platform is wooden to avoid NFC interference, as detailed in Section 3.1. At its core, a Raspberry Pi 5 coordinates two MFRC522 modules for gloves and shoes, and a UHF reader for shirts and safety glasses. Operator positioning is guided by the wooden platform with foot indicators to ensure optimal sensor coverage. Cables are routed internally, and a 7-inch touchscreen provides immediate visual feedback on authorization status. Power is supplied via a single 230V cable, while wireless communication with the backend server (broker) is handled through a dedicated WiFi router using the MADS-NET protocol over ZeroMQ/TCP.

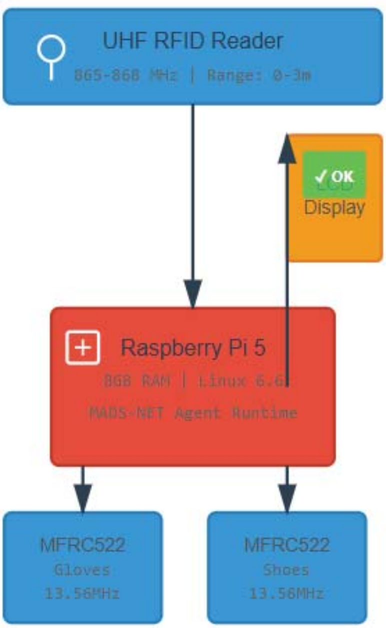

Fig. 2. Smart gate hardware architecture, showing the arrangement of NFC readers at hand and foot level, the UHF antenna at the top, and the Raspberry Pi 5 control unit

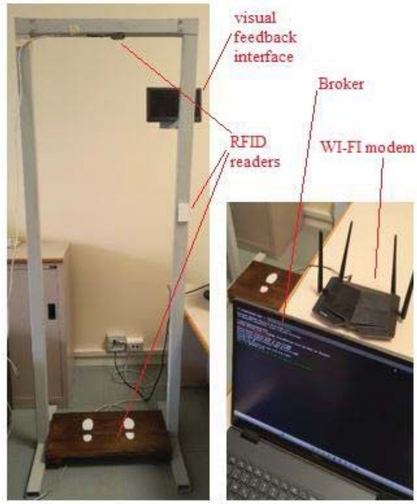

Fig. 3. Physical smart gate prototype with a steel frame featuring MFRC552 RFID readers at hand (left) and foot (right) levels, a UHF antenna on the top bar, and a wooden base platform for operator positioning. The Raspberry Pi 5 controller is housed in the white enclosure (left), with an LCD display mounted at eye level (right).

Fig. 4. Final installation of the smart gate in an industrial environment, showing the complete assembly with structural frame, sensors, display, and operator platform

2.3. Agent architecture

The system (figure 5) employs five specialized agent types, each responsible for a specific aspect of the monitoring workflow.

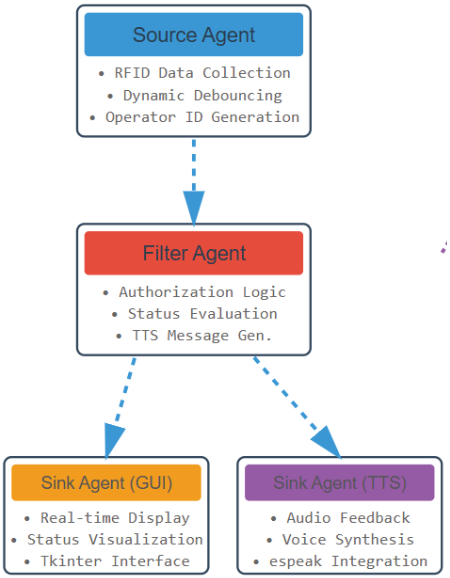

Fig. 5. Agent architecture and data flow: the five specialized agents (source, filter, visualization, TTS, logger) communicate through the MADS-NET broker via ZeroMQ pub-sub patterns

The source agent (*source.py*) interfaces directly with RFID hardware, implementing a novel dynamic debounce mechanism for UHF readers. Traditional fixed-window debouncing can miss rapid state transitions or introduce unnecessary delays. The implemented approach adjusts buffer length dynamically: when consecutive readings differ, the buffer expands to capture the transition more accurately; when readings stabilize, the buffer contracts to reduce latency. For MFRC522 readers, the agent uses an optimized detection strategy: if at least one of the last two readings indicates tag presence, the system reports presence. This asymmetric approach reduces false negatives caused by momentary signal loss while maintaining acceptable false positive rates, following approaches used in experimental studies on real-time measurement uncertainty (Capponi et al., 2023). The agent generates a unique operator ID when PPE is first detected and maintains this ID throughout the operator's presence. Critically, when no PPE is detected, the system implements a four-second grace period before clearing the operator ID, preventing spurious resets when operators briefly step back from the gate. The output is a JSON structure containing operator ID, individual PPE status flags (gloves, shoes, shirt, glasses), hostname for multi-gate installations, and a monotonically increasing timecode for temporal synchronization across distributed components. The filter agent (*filter.py*) receives data from source agents and determines authorization status. An operator is considered authorized only when all four PPE items are detected. The agent maintains state across invocations to track authorization transitions and trigger appropriate responses. To prevent notification fatigue, the filter implements rate limiting for text-to-speech messages. The agent tracks the last authorization status change and time of the last announcement. Messages are generated immediately upon status changes but are rate-limited to a configurable interval (default 10 seconds) for repeated notifications of the same status. When no operator is present (*operator_id* is null), the system enters a waiting state, producing output but marking authorization as undefined. This distinction prevents false alarms when the gate is idle. The filter enriches the data stream with a *is_authorized* boolean flag and, when appropriate, a *tts* (text-to-speech) field containing the text to be spoken. This separation of concerns allows the text-to-speech subsystem to operate independently without understanding authorization logic. The visualization agent (*sink.py*) creates a full-screen graphical interface (figure 6) using Tkinter that displays real-time operator status. The interface uses colour coding (green for authorized, red for unauthorized, grey for waiting) and emoji-based iconography for intuitive status comprehension. The GUI runs in a separate thread to prevent blocking the agent's main data processing loop. Updates are marshalled to the GUI thread using Tkinter's *after()* method to ensure thread safety. The interface includes a timestamp of the last update and a shutdown button for operator-initiated termination. Individual PPE items are displayed in a grid layout with status indicators, allowing supervisors to quickly identify which specific equipment items are missing when an operator is not authorized.

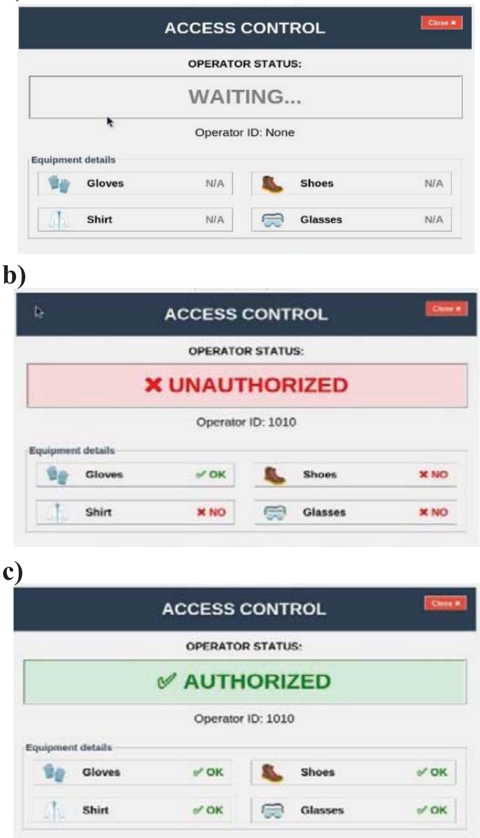

Fig. 6. The visualization agent (sink.py) uses a Tkinter-based GUI to monitor safety compliance through three states: (a) Waiting (grey) when no operator is detected; (b) Unauthorized (red) when the operator is identified but lacks PPE; (c) Authorized (green) when all safety requirements are met. The multi-threaded interface provides real-time updates on operator ID and protective equipment status.

The text-to-speech agent (*say.py*) provides auditory feedback to operators. When the filter agent adds a *tts* field to the data stream, this agent extracts the message and uses the espeak speech synthesis engine to generate Italian-language audio output (it’s also possible to choose English or French). The agent implements robust error handling to gracefully manage cases where the TTS field is absent or malformed. By using simple command execution rather than complex audio libraries, the implementation remains lightweight and reliable. Audio feedback is critical in industrial environments where visual attention may be focused on tasks rather than displays. The immediate auditory notification of authorization status allows operators to respond without diverting their attention from work. The temporal management of sensor data and the reduction of uncertainty in real-time measurements follow established approaches in dynamic systems. All agents communicate through MADS-NET's broker-based architecture (figure 7) using ZeroMQ pub-sub patterns. The broker handles message routing without requiring agents to know about each other's locations or even existence. This loose coupling enables the system to scale horizontally by adding monitoring gates without reconfiguring existing components. Each message includes a timecode that allows agents to sequence events correctly even when network delays occur. The system does not require precise clock synchronization between components, as the timecode provides logical ordering sufficient for the application's needs.

Fig. 7. MADS-NET broker: centralized message routing hub enabling loose coupling between agents via ZeroMQ pub-sub patterns

The plugin architecture allows new capabilities to be added without modifying existing code. For example, a logging agent could subscribe to the data stream and record events to a MongoDB database for audit purposes and historical analysis, all without changes to other agents. Configuration is managed through an mads.ini file that specifies agent parameters, update periods, Python virtual environment paths, and module search paths. This declarative approach separates configuration from code, facilitating deployment across different sites with varying requirements.

3. Implementation

3.1. Hardware and software setup

The gate hardware is described in detail in Section 2.2. Regarding materials, tests were conducted on Aluminium, polypropylene, ceramic, steel, polyethylene, and polycarbonate to assess potential RFID interference. UHF performance was unaffected by any material, whereas NFC detection is sensitive to Aluminium — hence the steel frame and wooden operator platform. NFC tags are embedded in flexible supports or shoe liners, and UHF tags are laminated and attached to clothing or eyewear; PPE–tag associations remain fully configurable in software. The Raspberry Pi 5 was chosen for its dual SPI buses, enhanced I/O capabilities, and processing power sufficient to run multiple Python agents concurrently on Linux with PREEMPT-RT patches, supporting low-latency operations while tolerating millisecond-scale timing variations. The system runs Python 3.11 and standard libraries for RFID communication, serial interfaces, and GUI management. MADS-NET enables deployment of modular agents, each running in a dedicated virtual environment for isolation and flexible updates.

3.2. Data flow and architecture

System operation follows a modular processing pipeline:

(i) A source agent polls RFID readers at 100 ms intervals and publishes raw detection data.
(ii) A filter agent evaluates authorization logic and enriches messages with status flags and optional audio feedback.
(iii) Visualization and TTS agents subscribe to processed data to update the GUI and provide spoken feedback.
(iv) Optional logging agents can record events for offline analysis.

This loosely coupled architecture allows agents to be added, removed, or restarted independently, ensuring robustness and ease of maintenance.

3.3. Deployment

Deployment consists of installing Python modules and dependencies within virtual environments and configuring agents via a central `mads.ini` file. Agents are launched as system services, enabling automatic startup and independent testing prior to full integration. So, each agent runs as a separate process, started with commands like:

```
mads python -n source_agent -m source -p100
```

This launches the agent named *source_agent* using the *source* Python module with a 100ms update period. A typical `mads.ini` configuration includes sections for each agent:

```
[source_agent]
venv = "/home/pi/project/.venv"
python_module = "source_rfid"
search_paths = ["/home/pi/project/python"]
```

4. Discussion

The experimental campaign enabled the extraction of measurable safety-relevant indicators suitable for supervisory and occupational safety evaluation. Real-world data from preliminary prototype trials were used for initial validation of system behavior. Since this work targets architectural validation rather than statistical inference, a synthetic dataset generated from real observations is methodologically sufficient: the authorization logic is deterministic and all possible scenarios are finite and fully represented. A synthetic dataset was then generated from these observations and used exclusively to compute the quantitative performance metrics reported here. The system was assessed not only for functional correctness but also through operational metrics characterizing authorization reliability, detection robustness, temporal responsiveness, and architectural safety behavior in a real-time industrial monitoring context. The key performance indicators are summarized in Table 1. Authorization reliability, measured through the False Authorization Rate (FAR), showed that no access events were granted in the absence of full compliance with all required PPE items (see Eq. (1)).

$$\text{FAR} = \frac{N_{\text{unauthorized granted}}}{N_{\text{total attempts}}} \quad (1)$$

Detection robustness varied across PPE categories; safety glasses showed the highest detection rate, while gloves exhibited the lowest. Detection rate is represented by the Positive Detection Ratio (PDR), which measures the proportion of positive RFID readings over the total monitoring period, including intervals in which the PPE was not worn.

$$\text{PDR}\ (\%) = \frac{\text{positives}}{\text{total}} \times 100 \quad (2)$$

Temporal performance (see Eq. (3)), evaluated as authorization latency, remained below one second under the 10 Hz monitoring frequency, confirming the system's suitability for real-time enforcement.

$$AL = t_{\text{full compliance}} - t_{\text{first PPE detected}} \quad (3)$$

The system provides continuous monitoring (see Eq. (4)) with timestamped compliance verification and enforces fail-safe operation by immediately revoking authorization if any PPE is missing or a sensor fails. Variations in Positive Detection Ratio (PDR) across PPE categories mainly reflect operator usage, not RFID reader reliability, which is confirmed by the zero False Authorization Rate.

$$MCR = \frac{N_{\text{compliance checks}}}{\Delta t} \quad (4)$$

Decisions are deterministic and independent of non-critical subsystems, prioritizing safety in case of uncertainty. Overall, the system generates audit-ready compliance logs, quantifies detection reliability per PPE category, and supports identification of technological blind spots, offering objective documentation suitable for occupational safety oversight. This capability may support compliance with Directive 89/391/EEC (European Council, 1989) and related national legislation. While no false authorization events were observed under controlled testing, larger-scale field validation is required to assess long-term reliability under environmental interference.

Table 1. Characteristics of the PPE monitoring system.

Characteristic (unit)	Value
False Authorization Rate (FAR)	0
Positive Detection Ratio – Safety Glasses (%)	82.7
Positive Detection Ratio – Safety Shoes (%)	66
Positive Detection Ratio – Safety Gloves (%)	50.7
Positive Detection Ratio – Protective Shirt (%)	55.3
Authorization Latency (s)	<1
Monitoring Continuity Rate (checks/s)	10
Fail-Safe Enforcement (–)	Immediate revocation if any PPE missing or sensor failure

5. Conclusions and Future Work

This work presents a distributed multi-agent architecture for real-time PPE monitoring using heterogeneous RFID technologies. The prototype demonstrates that lightweight agent-based frameworks can support safety-critical monitoring while maintaining modularity, scalability, and fault tolerance through the MADS-NET infrastructure. The integration of HF and UHF RFID readers with specialized Python agents validates the feasibility of the approach for industrial deployment. Future work will focus on experimentally validating RFID-based PPE detection against vision-based systems. Field tests will assess detection latency, read reliability, and false positive/negative rates under realistic industrial conditions, using a vision-based system based on pre-trained deep learning models (e.g., YOLOv8). This will enable systematic evaluation of RFID performance, including false authorizations from PPE being carried rather than worn, providing metrics to inform design choices for safety-critical monitoring. The current validation represents a preliminary experimental phase. Future extensions will expand the dataset through a simulation-based digital twin of the smart gate, enabling large-scale scenario generation, stress testing, and reliability assessment under controlled perturbations. Further developments will explore extending the software architecture to multi-gate implementations, enabling coordinated PPE monitoring across multiple control points with intuitive dashboard visualization. Integration with localization-based systems (Paolucci et al., 2025) could allow continuous PPE compliance monitoring beyond the gate, while thermal imaging (Cluni et al., 2025) could be used for environmental monitoring to detect potential anomalies around the gate. A PPE dispensing system at the gate entrance could ensure correct operator-equipment association prior to transit.

Acknowledgments

This work was conducted within the PRIN 2022 “Mirror World Miroscic” project, funded by the Italian Ministry of University and Research (MUR) under the Programmi di Ricerca di Rilevante Interesse Nazionale (PRIN).

References

Carbonari, A., Giretti, A., & Naticchia, B. (2011). A proactive system for real-time safety management in construction sites. *Automation in Construction*, 20(6), 686-698.

Giretti, A., Carbonari, A., Naticchia, B., & De Grassi, M. (2009). Design and first development of an automated real-time safety management system for construction sites. *Journal of Civil Engineering and Management*, 15(4), 325-336.

Li, H., Lu, M., Hsu, S. C., Gray, M., & Huang, T. (2015). Proactive behavior-based safety management for construction safety improvement. *Safety Science*, 75, 107-117.

Hermann, M., Pentek, T., & Otto, B. (2016). Design principles for industrie 4.0 scenarios. *49th Hawaii International Conference on System Sciences (HICSS)*, 3928-3937.

Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3, 18-23.

Fang, Q., Li, H., Luo, X., Ding, L., Luo, H., Rose, T. M., & An, W. (2018). Detecting non-hardhat-use by a deep learning method from far-field surveillance videos. *Automation in Construction*, 85, 1-9.

Mneymneh, B. E., Abbas, M., & Khoury, H. (2019). Vision-based framework for intelligent monitoring of hardhat wearing on construction sites. *Journal of Computing in Civil Engineering*, 33(2), 04018066.

Park, M. W., & Brilakis, I. (2012). Construction worker detection in video frames for initializing vision trackers. *Automation in Construction*, 28, 15-25.

Valero, E., Sivanathan, A., Bosché, F., & Abdel-Wahab, M. (2016). Musculoskeletal disorders in construction: A review and a novel system for activity tracking with body area network. *Applied ergonomics*, *54*, 120-130.

Kelm, A., Laußat, L., Meins-Becker, A., Platz, D., Khazaee, M. J., & Costin, A. M. (2013). Mobile passive Radio Frequency Identification (RFID) portal for automated and rapid control of Personal Protective Equipment (PPE) on construction sites. *Automation in Construction*, 36, 38-52.

Barenji, A. V., Barenji, R. V., & Hashemipour, M. (2016). A multi-agent RFID-enabled distributed control system for a flexible manufacturing shop. *International Journal of Advanced Manufacturing Technology*, 83(9-12), 1747-1758.

Goodrum, P. M., McLaren, M. A., & Durfee, A. (2006). The application of active radio frequency identification technology for tool tracking on construction job sites. *Automation in Construction*, 15(3), 292-302.

Capponi, L., Tocci, T., Marrazzo, M., Marsili, R., & Rossi, G. (2023). Experimental investigation on hardware and triggering effect in tip-timing measurement uncertainty. *Sensors*, *23*(3), 1129.

European Council. (1989). *Council Directive 89/391/EEC of 12 June 1989 on the introduction of measures to encourage improvements in the safety and health of workers at work*. Official Journal of the European Communities, L183, 1–8.

Paolucci F, Landi L, et al. Industrial warning system with active devices for signal reception and dynamic noise attenuation using artificial intelligence algorithms. 2025. doi:10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P7599-cd.

Cluni F, Moretti M, Paolucci F, et al. Stress distribution in additive manufactured TPMS beams: FE simulation and thermoelastic stress analysis. 2025. doi:10.1080/15376494.2025.2601274