IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Digital Object Identifier XX.XXXX/ACCESS.2026.XXXXXXX

MATE: A Methodology for Connecting Automatic Test Equipment in Industry 4.0

F. BIONDANI[1], (Student Member, IEEE), F. TOSONI[1], (Student Member, IEEE), N. DALL'ORA[1], (Member, IEEE), E. FRACCAROLI[1], (Member, IEEE), D. S. CHENG [1], (Member, IEEE),and F. FUMMI[1], (Member, IEEE)

[1]University of Verona, Department of Engineering for Innovation Medicine

Corresponding author: F. Biondani (e-mail: francesco.biondani_02@univr.it).

The work has been supported by the PRIN 2022T7YSHJ SMART-IC - Next Generation EU project and by the European Union's Horizon Europe research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 101109243. This manuscript reflects only the Authors' views and opinions; neither the European Union nor the European Commission can be considered responsible for them.'

ABSTRACT The growing complexity of intelligent systems with embedded boards has made Automatic Test Equipment (ATE) a critical component of Industry 4.0 ecosystems. However, legacy communication protocols and isolated testing processes hinder interoperability, slow down automation, and compromise data security. In this work, we propose MATE: a unified framework to integrate ATE systems into Industry 4.0, leveraging Open Platform Communications Unified Architecture (OPC UA) and Service-Oriented Architecture (SOA) principles. Building on our previous Companion Specification for ATE, we extend it to incorporate secure and automated firmware management, and we validate the combined solution in a real-world production scenario. MATE ensures secure and interoperable communication between ATE and other industrial assets, with an architecture designed to support scaling. By bridging conceptual design with practical implementation, this work provides a robust foundation for fully integrating ATE into manufacturing systems.

INDEX TERMS Automatic Test Equipment, Industry 4.0, OPC UA, Service-oriented architecture.

I. INTRODUCTION

THE rising complexity of electronic systems in automotive, aerospace, and consumer electronics has increased reliance on Automatic Test Equipment (ATE) to ensure product quality and reliability [1]. Testing is particularly critical in safety-sensitive domains, where failures can have severe consequences. As emphasized in "Chip War: The Fight for the World's Most Critical Technology" [2], semiconductor testing is not merely a technical necessity but a strategic asset in global manufacturing.

Despite their importance, traditional ATE systems often remain only partially integrated with Industry 4.0 frameworks. While some modern testers expose connectivity through proprietary APIs, interfaces remain fragmented and largely non-standard (see Figure 1). This fragmentation leads to production inefficiencies, reduced interoperability, and limited data-driven decision-making [3].

While Industry 4.0 promotes interconnected and autonomous production environments, and Open Platform Communications Unified Architecture (OPC UA) has emerged as the de facto standard for secure machine-to-machine communication, its potential has yet to be fully leveraged for ATE, leaving a critical integration challenge unresolved [4], [5].

In our earlier works, we proposed an OPC UA Companion Specification specifically tailored for ATE, establishing a standardized foundation for communication with other industrial assets [6]. Separately, we introduced a secure file transfer mechanism, presenting it as a promising approach for firmware management in testing environments [7].

However, a standardized model and a transfer protocol are insufficient on their own to transform a standalone tester into an agile industrial asset. Specifically, for an ATE to become a highly reconfigurable machine aligned with Industry 4.0 principles, a higher-level orchestration layer is required. In this paper, we move beyond these individual components to propose MATE, a framework that realizes this vision by integrating a Service-Oriented Architecture (SOA) as its core architectural element. By unifying the OPC UA Companion Specification with secure, automated firmware manage-

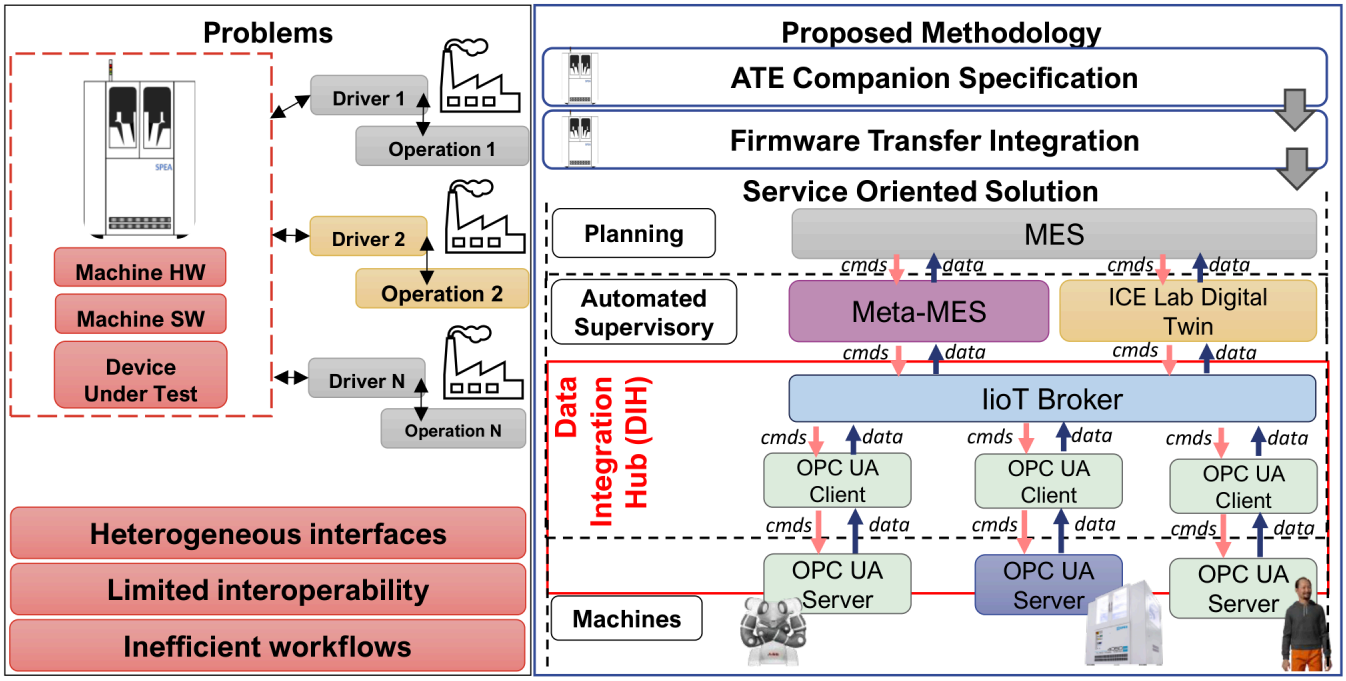

Figure 1. **Overview of the ATE problems and the MATE framework to overcome them.**

ment within a SOA context, MATE orchestrates standardized communication and data exchange to enable interoperability across heterogeneous production lines (see Figure 1). The key scientific contribution is the end-to-end validation of this integration in an operational production setting, rather than the isolated proposal of individual building blocks.

The main contributions of this work are:

1) **SOA-based Framework for ATE**: We propose and implement **MATE**, a framework that leverages a Service-Oriented Architecture to transform standalone ATE systems into reconfigurable industrial assets. This architecture enables the ATE to act as a service provider, allowing for dynamic orchestration and vendor-independent interoperability within Industry 4.0 environments.
2) **Validation of the Companion Specification**: We implement and test the ATE Companion Specification in representative testing workflows.
3) **Integration of secure firmware management**: We extend the Companion Specification to include standardized and automated firmware deployment, enabled through OPC UA, and add a technical comparison with other file transfer systems and with different firmware sizes.
4) **End-to-end validation in a production environment**: We evaluate MATE in an operational production line, where ATE participates in the creation of an electronic gadget alongside other industrial assets, with the Companion Specification and firmware transfer supporting interoperability and operational continuity.

By bridging conceptual design with real-world implementation, MATE provides a methodology for embedding ATE systems into interconnected manufacturing environments.

The remainder of this article is organized as follows. Section II summarizes the technical background on ATE, communication protocols, and SOA principles. Section III presents the proposed method and its design components, while Section IV details the concrete implementation. Section V reports the performance evaluation and system configuration results, and Section VI concludes the paper.

II. BACKGROUND

A. AUTOMATIC TEST EQUIPMENT

ATE plays a critical role in ensuring the quality and reliability of electronic components. Failures in electronic boards can cause production downtime, safety hazards, and severe reliability issues, particularly in domains such as automotive, aerospace, and consumer electronics [8], [9]. ATE mitigates these risks by performing systematic and automated testing before products enter the market [10], [11].

Functionally, an ATE system integrates test heads, which provide the physical electrical interface, handlers, which automate the positioning and loading of devices under test, and dedicated software, which controls the instruments and analyzes the results. Together, these elements enable comprehensive and repeatable testing under controlled conditions, ensuring that components meet strict quality and reliability standards.

Despite their importance, current ATE systems face several challenges:

- **Lack of standardization:** Each vendor typically relies on proprietary communication protocols and legacy

systems. This heterogeneity complicates interoperability and makes integrating ATE with other industrial assets difficult.

- **Intellectual property protection:** Firmware used during testing embodies valuable Intellectual Property (IP) [12]. Board producers who outsource testing to third-party companies must protect this IP, often relying on hardware-based solutions such as dongles. These protections are costly and fragile; if a firmware specification is incorrect, as we have experienced, production can be significantly delayed.
- **Limited collaboration in heterogeneous environments:** While recent works have advanced the performance and automation of standalone ATE systems, they rarely address scenarios where ATE must operate in close collaboration with other machines [13], [14]. In practice, modern factories demand dynamic interoperability across heterogeneous assets, yet the role of ATE in such coordinated, service-oriented settings remains underexplored.

Similar challenges concerning data security, privacy, and traceability have been noted in recent Industrial Internet of Things research [15]–[17], further emphasizing the need for standardized communication models and secure, interoperable frameworks that enable seamless collaboration between ATE and other machines in heterogeneous production environments.

B. COMMUNICATION PROTOCOLS AND SERVICE-ORIENTED ARCHITECTURES

Machine-to-Machine communication is fundamental in Industry 4.0 manufacturing, enabling interoperability, real-time data exchange, and process automation. The OPC UA, standardized as IEC 62541, has emerged as the de facto standard in this context, thanks to its platform independence, built-in security mechanisms, and extensible information modeling capabilities [18]. Unlike traditional industrial protocols, OPC UA ensures semantic consistency across heterogeneous systems while providing encryption, authentication, and fine-grained access control. For ATE, the adoption of an OPC UA Companion Specification is particularly relevant, as it standardizes test procedures, device configurations, and result descriptions, reducing fragmentation and facilitating integration into broader industrial ecosystems.

Although OPC UA provides the foundation for standardized data exchange, it is insufficient on its own to address the flexibility and scalability required in modern production environments. Managing multiple product variants, adapting to dynamic workflows, and coordinating heterogeneous assets demand an architectural paradigm that goes beyond point-to-point communication. SOA responds to this challenge by abstracting machine functionalities into modular and reusable services [19]. In manufacturing, SOA complements OPC UA by supporting integration at increasing system scale, dynamic reconfiguration of production workflows, and interoperability

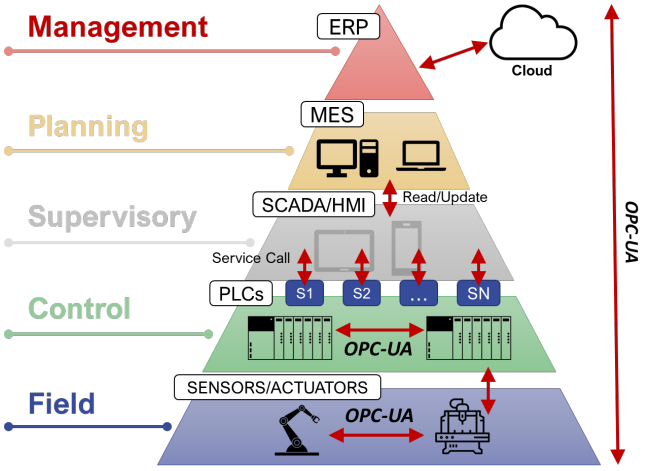

Figure 2. The Automation Pyramid and its evolution toward service-based integration.

with higher-level systems such as Manufacturing Execution System (MES) and Enterprise Resource Planning (ERP) [20].

This evolution can be understood through the Automation Pyramid (see Figure 2), which has traditionally structured manufacturing systems into hierarchical layers [21]. While effective, this rigid separation increasingly limits flexibility in Industry 4.0 environments [22].

Distributed SOA applications require fault-tolerant communication channels that can operate at increasing system scale beyond traditional client/server patterns. Publish-subscribe messaging protocols address this need through message brokers that act as intermediaries between producers and consumers, implementing delivery policies including prioritization, filtering, and persistence. Apache Kafka [23] excels in high-volume event streaming, offering guaranteed persistence through topic partitioning, making it well-suited for manufacturing environments that require high throughput and data durability. RabbitMQ [24] offers flexibility with support for multiple messaging protocols and both asynchronous and synchronous patterns, including RPC capabilities. KubeMQ [25] offers native optimization for Kubernetes environments, providing comprehensive messaging pattern support tailored to containerized microservice architectures. The selection of appropriate communication infrastructure directly impacts system responsiveness, fault tolerance, and scalability in manufacturing SOA implementations.

In conjunction with OPC UA, SOA bridges the gap between hierarchical and network-based paradigms, allowing ATE to evolve from an isolated testing machine into a secure, interoperable, and reconfigurable service within the broader Industry 4.0 ecosystem.

C. RELATIONSHIP TO PRIOR WORKS

This manuscript builds upon and significantly extends two preliminary conference publications by the authors:

Table 1. **Structured comparison highlighting the incremental contributions of the current manuscript against prior conference publications.**

Feature / Aspect	Prior Work 1: D-MATE [6]	Prior Work 2: Firmware [7]	Current Manuscript (MATE)
Core Contribution	Theoretical OPC UA Information Model for ATE.	Secure firmware transmission mechanism via OPC UA.	**Unified SOA Framework** integrating and orchestrating both concepts.
Architecture	Point-to-point client/server (Theoretical).	Point-to-point file transfer (Isolated).	**Kubernetes-based SOA** with Kafka IoT Broker and Meta-MES.
Heterogeneous Integration	None (ATE only).	None (Isolated data channel).	**Full Interoperability** with physical CNC, Vision, and Robotic systems.
Validation Scope	Conceptual design and UML modeling.	Network security penetration testing.	**End-to-End Validation** in a real-world production pilot line.
Performance Evaluation	None.	None.	**Benchmarking**:SOA computational overhead, multi-protocol file transfer performance, and application-layer chunking analysis over VPN.

- **Prior Conference Papers:** In our earlier work on D-MATE [6], we proposed the theoretical foundation of an OPC UA Companion Specification tailored for ATE. That work focused strictly on defining the information model, without physical implementation or integration with higher-level orchestration. Separately, in [7], we introduced a secure file transfer mechanism based on the OPC UA Part 20 standard. That study validated the encryption and network transmission of firmware in isolation, independent of any specific ATE workflow.
- **Newly Introduced Contributions and Extensions:** This journal article unifies these preliminary concepts by introducing **MATE**, a comprehensive SOA-based framework. The novel elements introduced exclusively in this manuscript include: (i) the practical implementation of the ATE Companion Specification into a fully operational OPC UA server; (ii) the extensive evaluation of the file transfer mechanism using various firmware sizes over a high-latency VPN, simulating distant site-to-site industrial communication; and (iii) the integration of a SOA orchestration layer that interfaces the ATE with other heterogeneous machines and dynamically manages diverse production recipes. Consequently, this work provides the first end-to-end physical validation of the complete system in a real-world production line.

Table 1 summarizes the architectural evolution and the expanded validation scope of the current manuscript compared to our previous works.

III. METHOD

MATE addresses the key challenges of interoperability, firmware protection, and machine orchestration at increasing system scale. The novelty of MATE does not lie in the development of new communication protocols but rather in the rigorous integration of established, robust standards into the ATE domain. MATE is structured into three main phases:

1) development of a domain-specific OPC UA Companion Specification for ATE
2) secure firmware transmission integrated within this specification, and
3) incorporation of a SOA to enable interoperability with heterogeneous machines and production services.

A. OPC UA COMPANION SPECIFICATION FOR ATE

The first step in MATE is the development of a dedicated OPC UA Companion Specification for ATE. This addresses the long-standing problem of vendor-specific protocols and legacy interfaces that make ATE systems difficult to unify. The specification defines a standardized information model to represent entities such as machine identification, test configurations, execution parameters, and security controls.

By structuring ATE data within the OPC UA Address Space, test workflows become interoperable with MES, ERP platforms, and other Industry 4.0 assets. Test requests can be triggered remotely, execution monitored in real time, and diagnostic data retrieved through standardized OPC UA service calls. Role-based access control ensures that only authenticated entities can initiate tests or access sensitive information. This step transforms ATE from an isolated machine into a standardized Industry 4.0 component.

B. SECURE FIRMWARE TRANSMISSION INTEGRATED WITH OPC UA

The second step of MATE responds to the need for stronger intellectual property protection in firmware handling. Many board producers outsource testing to third parties, but traditional protection mechanisms (e.g., hardware dongles) are costly and inflexible, and a misconfigured firmware specification can delay production.

To mitigate these risks, we embed a secure firmware transmission mechanism directly within the Companion Specification, adhering to the OPC UA Part 20 (File Transfer) specification. Firmware requests are authenticated, transfers occur over encrypted channels, and the data is erased automatically after use. Every operation is logged for traceability, providing guarantees of confidentiality, integrity, and version control. This approach eliminates reliance on external storage devices and ensures firmware protection is a native part of ATE integration.

While the Part 20 standard defines the theoretical model for file handling, practical implementations are often missing from mainstream open-source libraries [7]. Our implementation fills this gap. Due to the lack of existing reference implementations and documented performance studies regarding the OPC UA Temporary File Transfer mechanism, our workflow and the experimental data provided in Table 2 represent one of the first studies in this domain. This contributes to the state-of-the-art by demonstrating the practical feasibility and efficiency of the Part 20 specification for high-stakes industrial applications, such as secure firmware deployment in ATE systems.

C. SERVICE-ORIENTED ARCHITECTURE FOR SMART FACTORY INTEGRATION

The final step of MATE tackles scaling requirements and dynamic production management. As discussed in Section II-B, while OPC UA provides a standardized communication baseline, it lacks the orchestration capabilities needed for dynamic workflows. To tackle this issue, MATE utilizes the service-oriented architecture (SOA) layer by employing containerization and Kubernetes. This decision is motivated by the necessity for fault tolerance, service isolation, and self-healing mechanisms. Kubernetes automatically replaces failed containers, reschedules workloads when nodes become unavailable, and ensures the system maintains its desired state. These features are crucial in a production environment to prevent costly downtime. By leveraging Kubernetes, MATE can orchestrate the ATE together with other machines as decoupled microservices, allowing for the dynamic reconfiguration of production recipes without interrupting operations. The choice of a containerized SOA is further justified by the negligible computational overhead introduced. As demonstrated in our experimental evaluation (see Table 3), the latency added by the orchestration and message brokering layers remains below 0.21% compared to a direct point-to-point connection, a trade-off that is more than compensated by the gains in flexibility and reliability.

Together, these three phases ensure that MATE systematically addresses the main challenges of ATE integration. The Companion Specification resolves the long-standing issue of vendor-specific protocols by providing a standardized communication layer. The secure firmware transmission mechanism protects intellectual property during outsourced testing, reducing the reliance on physical hardware dongles. Finally, the incorporation of SOA enables orchestration designed to scale and the dynamic management of production recipes without interrupting operations.

IV. PROPOSED SOLUTION

A. IMPLEMENTATION OF THE COMPANION SPECIFICATION

To validate the proposed methodology, we implemented the OPC UA Companion Specification for ATE and secure firmware transmission mechanism, integrating them into a real-world smart manufacturing setup. The experimental evaluation was conducted in the Industrial Computer Engineering (ICE) Laboratory, where we tested the feasibility, security, and interoperability of the OPC UA-based ATE system in an Industry 4.0 production environment.

The implementation of the proposed OPC UA Companion Specification for ATE involves defining a structured and extensible information model that enables seamless integration with Industry 4.0 environments (see Figure 3). This section presents the key components of the model, each addressing a specific aspect of ATE operation.

1) ATE Machine Identification

This component serves as the asset management record for the ATE machine. It provides a unique identifier (MachineID) and records essential physical details like Manufacturer, Model, SerialNumber, and Location for easy tracking and asset inventory. It is a data-only component with no methods.

2) System Configuration

This handles the machine's setup and operational parameters. It manages the state of the HardwareSetup, SoftwareSetup, and NetworkConfiguration, enabling the machine to connect and function in the production environment. Its methods allow for the loading and saving of configurations.

3) Test Configuration

This defines the blueprints for testing. It manages the creation and storage of various testing processes, including TestName, TestParameters, TestSequence, and defined TestLimits. Its methods are used to create, update, delete, and retrieve specific test configurations.

4) Test Execution

This component manages the actual testing lifecycle. It tracks the test's status in real-time using attributes like ExecutionID, StartTime, EndTime, and Status. It provides operators with control methods to Start, Stop, Pause, and Resume any active test.

5) Results Reporting

This standardizes the recording and documentation of test results. It associates the outcome (e.g., PassFailStatus and detailed ResultData) with the specific test run (TestID, ExecutionID) and the Timestamp. It provides functionality to retrieve and export results for analysis.

6) Maintenance and Diagnostics

Focused on machine health, this component ensures reliability and performance. It tracks data like the MaintenanceSchedule, ErrorLogs, and PerformanceMetrics. Its methods are used to Run Diagnostics, Schedule Maintenance, and access the machine's history and reports.

7) Security Considerations: Application-Level Security

This component exposes the machine's internal user management system to the network. Distinct from standard OPC

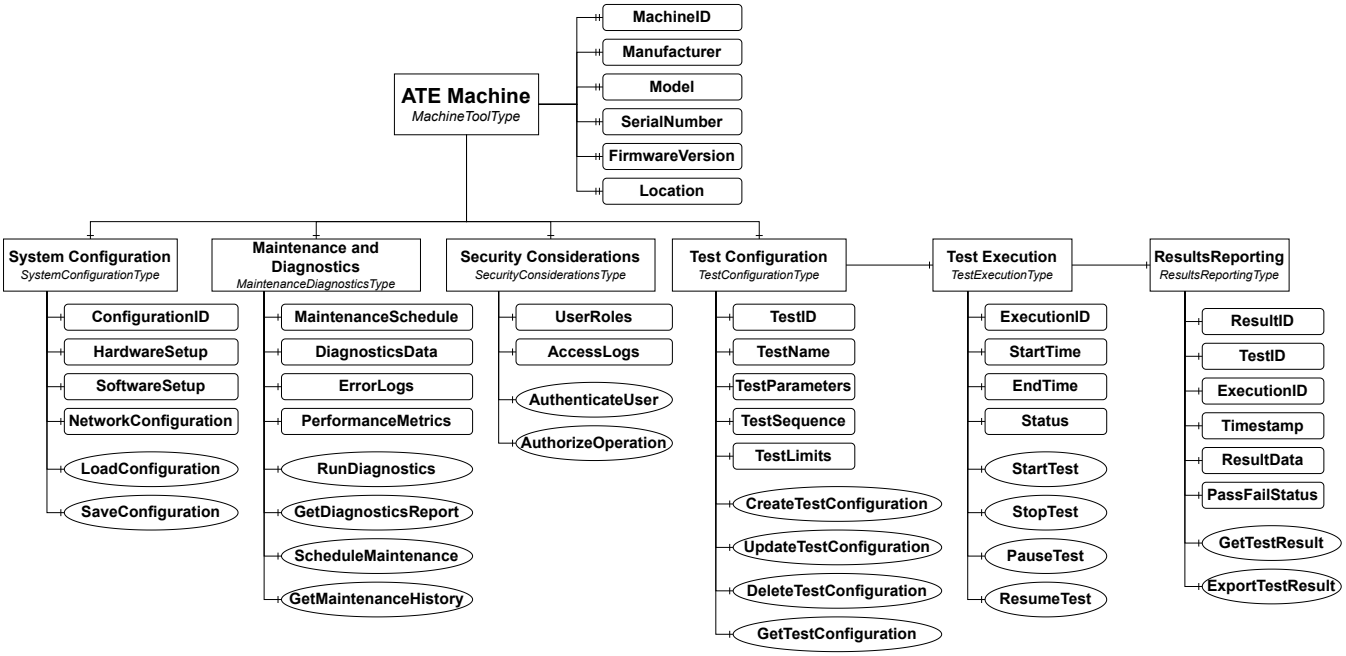

Figure 3. **Hierarchical representation of the ATE Information Model, detailing its main components and their relationships.**

UA Session authentication and the `RolePermissions` attribute, this functional object enables the management of application-level logic. The `UserRoles` variable is a custom implementation that maps a connected client to a specific operator profile within the ATE proprietary software. This allows the system to restrict sensitive operations, such as modifying test limits or bypassing safety checks based on the operator's clearance level, operating in parallel with the standard OPC UA security model.

B. OPC UA SERVER CREATION

After creating our information model, we used the model compiler [26] to convert the XML-based model into source files compatible with our OPC UA server environment. This process ensures compliance with the OPC UA standard and establishes the architecture needed for effective client-server communication. Two primary classes manage the server implementation: `RCServer` and `RCNodeManager`. The `RCServer` class manages overall server properties, node manager creation, and server lifecycle events. Specifically, it configures the server's URI, handles requests, and loads essential properties like manufacturer and software version details. Meanwhile, the `RCNodeManager` is responsible for setting up and maintaining the server's address space. It loads predefined nodes, converts them to the required typed nodes, and links methods to appropriate callbacks, enabling interaction with ATE functionalities.

Our OPC UA server design includes a custom request-handling system to manage prioritized command execution. This handler ensures deterministic management of high-priority commands, particularly those associated with safety-critical operations, by giving them precedence in processing. This system maintains operational integrity, allowing the server to efficiently manage and execute diverse client requests within the production environment.

C. SECURE FIRMWARE TRANSMISSION

Security aspects of this functionality were thoroughly analyzed in our previous work [7]. The implementation was subjected to penetration testing using the **OPC UA Exploitation Framework** [27]. The analysis confirmed the system's resilience against unauthorized access, eavesdropping, and denial-of-service attacks under industrial constraints. Building on that validated foundation, this paper focuses on integrating this secure channel within the broader Service-Oriented Architecture.

We extend the Companion Specification by adding a TemporaryFileTransferType node to the address space (see Figure 4). The implementation utilizes the OPC UA .NET stack, providing authenticated sessions, encryption, and access control with request monitoring and time-constrained execution.

V. PERFORMANCE EVALUATION

We benchmarked OPC UA against standard file transfer protocols using realistic industrial conditions. The test setup: SPEA ATE machine (client) connecting to a Windows server through VPN.

The test infrastructure specifications included:

- **Client System**: SPEA 4050 ATE machine running Windows 10 Pro with Intel Core i7-6700 processor, 16 GB RAM, and Gigabit Ethernet connectivity.
- **Server System**: Windows Server 2019 virtual machine (4 vCPU, 8 GB RAM) hosted on VMware vSphere infrastructure.

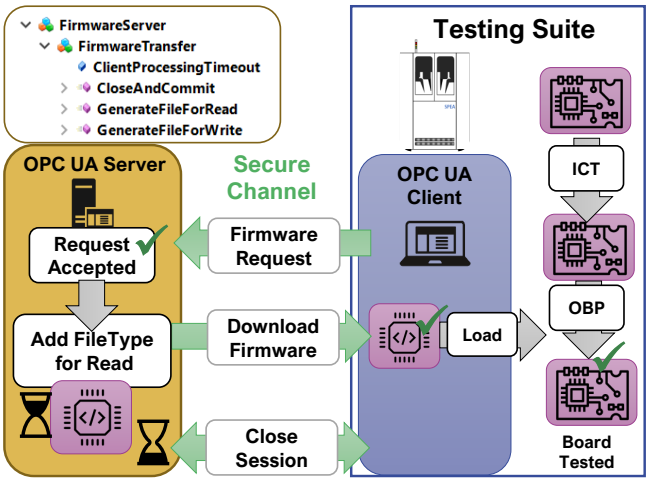

Figure 4. **Sequence of the secure firmware transfer: after establishing a secure channel and session, the ATE client requests firmware from the customer's OPC UA server. Upon validation, the server issues a FileHandle and NodeID, enabling the client to securely download and flash the firmware during the OBP phase. The session is then closed either at the end of testing or after a predefined timeout.**

- **Network Configuration**: Site-to-site IPSec VPN tunnel with 100 Mbps guaranteed bandwidth, average RTT of 45 ms, and 0.1% packet loss.
- **Security Settings**: All protocols were configured with equivalent high-security levels to ensure a fair comparison. Specifically, the OPC UA connection utilized the `Aes256-Sha256-RsaPss` security policy. To maintain parity, HTTPS, FTPS, MQTT, and gRPC were configured over TLS 1.2 using the `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384` cipher suite.

Table 2 shows average transfer times from 50 attempts per protocol and file size combination. The observed standard deviation across all configurations was approximately $\pm 7\%$, mainly due to VPN latency fluctuations.

The protocols evaluated in Table 2 encompass diverse protocols. While these protocols are designed for different operational use cases, they are frequently deployed in IoT environments. Consequently, this benchmark evaluates their relative performance when tasked with heavy payload transmission over a high-latency industrial VPN.

To standardize the evaluation, an application-level chunking strategy was applied across all protocols, baselined against the default encoding limits of the OPC UA client stack. Specifically, OPC UA defines a maximum chunk size of 65,536 bytes and a maximum assembled message size of roughly 2 MB. To mirror these limits and prevent memory allocation biases, gRPC was implemented with client-side bidirectional streaming to strictly enforce a 2 MB threshold, overriding its default 4 MB limit. Similarly, MQTT payload transfers were chunked to match this size using Quality of Service Level 1. Within the OPC UA implementation, chunking and buffering are natively managed by the `opc.tcp://` transport profile, which delegates flow control directly to the OS-level TCP sliding window mechanism without introducing application-layer bottlenecks.

Table 1 reports the mean transfer times derived from 50 iterations per configuration. The standard deviation remained within $\pm 7\%$, primarily reflecting VPN latency fluctuations. Furthermore, median transfer times deviated from the mean by less than 1%, indicating a normal distribution and the absence of extreme network anomalies.

A notable outcome of the benchmark is the performance degradation of gRPC during large file transfers. This discrepancy is attributed to gRPC's reliance on the HTTP/2 transport protocol. Over high-latency networks (45 ms RTT), the Bandwidth-Delay Product severely impacts HTTP/2's application-layer flow control, which relies on `WINDOW_UPDATE` frames. This secondary layer of flow control throttles throughput as the sender must frequently stall while waiting for window updates to traverse the VPN tunnel. In contrast, the OPC UA stack relies strictly on TCP flow control, avoiding the HTTP/2 application-layer overhead and more efficiently managing continuous, heavy-payload streams. The data highlights the distinct operational envelope of **MQTT**: while highly efficient for small telemetry packets (0.41 s at 50 KB), its performance decreases with larger file sizes due to the overhead of message fragmentation and broker handling. Conversely, OPC UA maintains operational linearity. Specifically, the use of Quality of Service Level 1, selected to ensure guaranteed delivery comparable to OPC UA, introduces additional latency due to the required acknowledgment handshakes for each fragmented packet segment. Regarding the counter-intuitive advantage over HTTPS, which is typically highly optimised for large file transfers, this result is explained by the underlying transport profiles acting over the high-latency VPN. Our OPC UA implementation utilizes the UA-Binary encoding over the TCP transport profile (`opc.tcp://`). Once the initial asymmetric security handshake is completed and the `SecureChannel` is established, data is transmitted as binary chunks. This transport natively lacks the verbose header overhead required by standard HTTP(S) streams for every packet, allowing the OPC UA stack to manage the TCP windowing constraints of the VPN slightly more efficiently.

While network conditions primarily determine transfer performance, OPC UA consistently achieves 8-15% faster transfers than state-of-the-art protocols. Furthermore, the OPC UA-based secure transfer provides a digital alternative to physical hardware dongles for intellectual property protection. Because physical dongle distribution intrinsically involves logistical transport (days), a direct quantitative performance comparison with the real-time network protocols analyzed in Table 2 was excluded from this study.

A. SERVICE-ORIENTED ARCHITECTURE INTEGRATION

The final step of the MATE methodology integrates the OPC UA Companion Specification and the secure firmware transmission mechanism within a SOA implemented as a

Table 2. Comparison of average transfer times for different file sizes across protocols in industrial VPN environment.

Protocol	50 KB	10 MB	80 MB	200 MB	500 MB
HTTPS	0.55 s	3.21 s	22.27 s	51.84 s	127.8 s
FTPS	0.61 s	4.12 s	24.78 s	59.31 s	146.1 s
OPC UA	0.48 s	**2.98 s**	**20.53 s**	**49.73 s**	**122.6 s**
MQTT	**0.41** s	5.87 s	30.12 s	70.54 s	129.9 s
gRPC	0.70 s	3.05 s	21.43 s	52.66 s	173.8 s

Kubernetes-based orchestration framework. All the elements described in this section are depicted in Figure 5.

The architecture is deployed as a Kubernetes cluster, where each component runs in a dedicated container and communicates through Apache Kafka as the IoT message broker. Apache Kafka was selected for its distributed architecture, high throughput capabilities, and built-in data persistence, which are essential for industrial environments requiring guaranteed message delivery and fault tolerance.

The Data Integration Hub (DIH) handles system-wide communication. It combines an IoT broker with OPC UA servers bound to each machine and OPC UA client nodes that maintain persistent subscriptions to machine variables. Updates are published to broker topics, while RPC requests (read, write, method invocation) are routed back to the equipment. Machine services are exposed via OPC UA methods, which abstract machine-specific logic.

The **Meta-MES** manages interaction with the MES. It extends MES functionalities by enabling dynamic reconfiguration, automated recipe execution, and advanced scheduling. The Meta-MES complies with the ISA-95 standard, facilitating integration into existing MES infrastructures.

Overall, the SOA spans from the control level, where machines and OPC UA servers operate, to the planning level, where MES functions reside. The DIH ensures reliable communication, while the Meta-MES supervises operations and coordinates production workflows.

1) Architecture Performance Evaluation

To evaluate the proposed SOA architecture, we measured execution times for four production recipes of varying complexity against a baseline architecture, referred to in Table 3 as "Baseline". **Baseline Definition:** The baseline represents the standard industry approach for connected but non-orchestrated equipment. It is implemented as a direct, point-to-point OPC UA Client-Server connection where the control logic interacts directly with the ATE, bypassing the Apache Kafka broker and the Meta-MES orchestration layer. This comparison allows us to isolate and quantify the specific overhead introduced by the message brokering and service abstraction layers.

Table 3 shows the total execution times excluding material transportation through conveyor belts, as transportation data is highly variable and influenced by physical factors beyond software architecture control. The proposed SOA architecture introduces an overhead between 0.12% and 0.21% compared to the baseline direct connection, while providing a containerized deployment and standardized OPC UA communication interfaces.

Table 3. Execution time comparison between the baseline approach (direct OPC UA connection) and the proposed SOA architecture for different production recipes. The "Overhead" column indicates the percentage increase in execution latency attributed to the SOA message brokering and orchestration layers compared to the direct connection baseline.

Recipe	# Tasks	Baseline	Proposed	Overhead
1	4	140.04 s	140.34 s	0.21%
2	7	145.03 s	145.25 s	0.16%
3	8	120.03 s	120.18 s	0.12%
4	11	300.10 s	300.69 s	0.20%

Table 4. Comparison of the isolated communication delay introduced by the proposed SOA architecture against a direct OPC UA connection.

Transport Type	Read (ms)	Write (ms)	Methods (ms)
OPC UA (Direct)	0.63	0.73	0.86
SOA (MATE)	4.02	4.22	3.30
Overhead	538.53%	579.41%	283.86%

Under the tested conditions, these results indicate that dynamic reconfiguration, secure firmware management, and standardized machine interfaces can be introduced with limited additional execution time.

To quantify the computational overhead of the SOA architecture, Table 4 compares the routing delay of the proposed solution against a direct point-to-point OPC UA connection for reading, writing, and calling methods.

While the relative overhead for these operations is high (283% to 579%), the absolute delay per transaction is only a few milliseconds (e.g., a method call increases from 0.86 ms to 3.30 ms, and a write from 0.73 ms to 4.22 ms). When these individual operations are aggregated into a complete production recipe (Table 3), the total software latency remains negligible. The cumulative millisecond delay of the software commands is completely masked by the physical execution time of the different machines. Consequently, the SOA orchestration can be deployed without impacting the actual manufacturing cycle time.

For broader context, MATE was also compared conceptually with the recent work of Ji et al. [28], who proposed an OPC UA-based hybrid architecture for integrating Cloud Manufacturing and Cyber-Physical Systems. Their study remains theoretical, focusing on high-level cloud interoperability, whereas MATE validates these principles through an operational deployment and practical comparison with alternative communication protocols, thereby substantiating their theoretical claims with empirical evidence.

B. REAL-WORLD DEPLOYMENT: ICE LABORATORY INTEGRATION

To further validate the industrial applicability and operational robustness of MATE, the complete framework was deployed

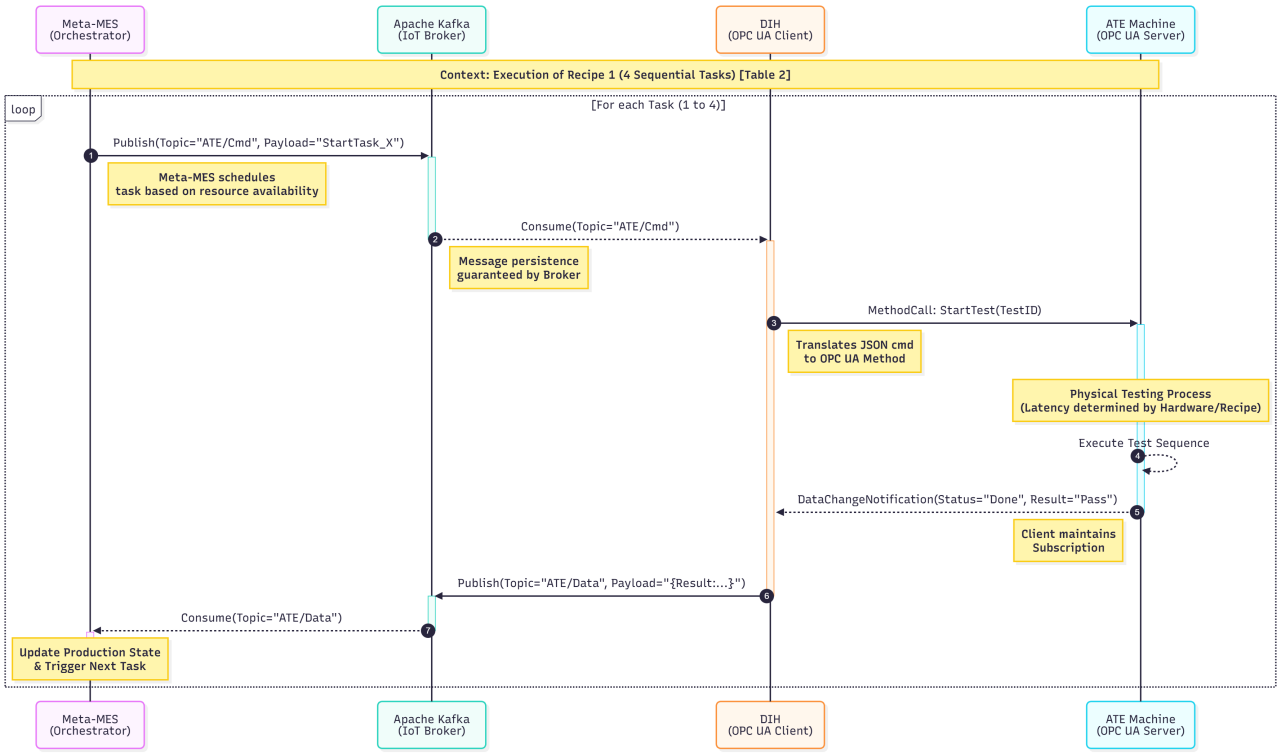

Figure 5. **Sequence diagram illustrating the interactions between the Meta-MES, Data Integration Hub (DIH), and ATE during the execution of "Recipe 1" (shown in Table 3). The diagram delineates the architectural overhead (message routing and translation) from the physical processing time, showing how the DIH bridges the asynchronous Kafka stream with synchronous OPC UA method calls.**

in the Industrial Computer Engineering (ICE) Laboratory under real manufacturing conditions. The deployment employs a *production-grade SPEA 4050 Automatic Test Equipment (ATE)* system within a heterogeneous manufacturing line, where ATE participates in the creation of an electronic device alongside other industrial assets. In addition to experimental deployment, the architecture and its OPC UA integration were reviewed by senior engineers from *SPEA S.p.A.*, one of the leading ATE manufacturers. Their assessment confirmed the feasibility of the implementation and its alignment with industrial requirements for large-scale ATE deployment. Collectively, the successful orchestration of physical assets in this representative environment, combined with expert industrial validation, advances the proposed methodology to a Technology Readiness Level (TRL) of 6.

The complete deployment and its operational workflow are demonstrated in the accompanying video[1]. As illustrated in Figure 6, the recipe comprises seven interconnected manufacturing phases orchestrated via the SOA infrastructure. Every asset mentioned is a physical commercial unit integrated into the ICE Laboratory, and no simulated data was used for the validation. Specifically:

- **Phase 1 - Warehouse Extraction & Loading:** Automated retrieval of the raw components from the vertical warehouse and robotic transfer to the production line (visible at 1:33 min).
- **Phase 2 - Visual Control:** Physical quality inspection

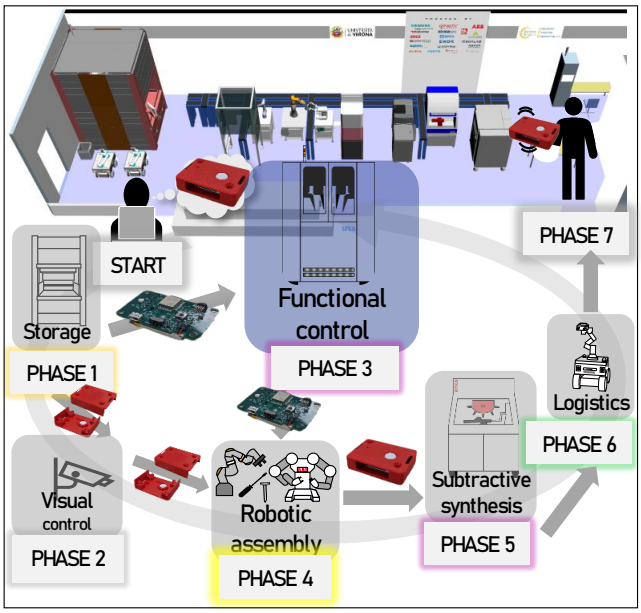

Figure 6. **The ICE Laboratory and the various phases of the recipe for the gadget creation coordination through the proposed SOA architecture.**

[1] ICE Laboratory Demonstrator.

using computer vision systems to check possible defects (visible at 5:10 min).
- **Phase 3 - Functional Control:** Automated testing of the electronic board using the ATE for product validation (visible at 4:10 min).
- **Phase 4 - Robotic Assembly:** Final gadget assembly (visible at 10:05 min).
- **Phase 5 - Subtractive Manufacturing:** CNC machining operations (visible at 13:15 min).
- **Phase 6 - Logistics:** Automated product handling and delivery (visible at 16:35 min).
- **Phase 7 - Activation:** Device initialization and functional activation (visible at 18:55 min).

Each phase operates as an independent containerized service within the Kubernetes cluster, communicating through standardized OPC UA interfaces and Kafka-based message streams. This configuration demonstrates the following capabilities:

a: Heterogeneous Equipment Integration

The architecture integrates diverse commercial equipment, from vision systems to CNC machines, using unified OPC UA companion specifications. Minimal adaptation was required for each asset to participate in the orchestrated workflow, confirming interoperability across vendors and technologies.

b: Dynamic Reconfiguration

Production recipes can be modified at runtime without halting the production line. The Meta-MES autonomously reassigns tasks based on resource availability, system status, and production priorities, ensuring continuous operation.

c: Fault Tolerance

When a subsystem experiences downtime, Kafka's persistence mechanisms ensure message delivery and enable the automatic rerouting of tasks to maintain operational continuity, thereby preventing data loss and reducing the need for manual intervention.

d: Scalability and Extensibility

The SOA-based architecture scales by instantiating new containerized services and OPC UA endpoints as needed. This design supports concurrent production recipes and variable batch sizes while preserving performance and security guarantees. Furthermore, the system handles firmware transfers of varying magnitudes, from small telemetry packets to large binary files, with minimal overhead compared to physical transport.

While the current experimental validation utilized a pilot line with a single ATE unit to demonstrate heterogeneous orchestration, the modularity of the DIH naturally supports the integration of multiple concurrent nodes. At this stage, reported performance metrics are centered on mean values, and no dedicated high-concurrency stress campaign has been executed yet. Given these current validation limits, future developments will focus on multi-ATE concurrency stress tests and broker saturation profiling to further validate throughput limits in multi-machine environments. Beyond immediate performance scaling, the framework's extensibility opens multiple research and industrial trajectories. As a future extension, predictive-maintenance workflows combining Explainable Artificial Intelligence (XAI) methods with temporal thermal analysis can be investigated; these remain out of scope for the current manuscript and are left for follow-up work. Moreover, this extensibility transcends the ATE domain. The underlying methodology, specifically the combination of a semantic Information Model with the secure `TemporaryFileTransfer` mechanism, exhibits high transferability to other industrial assets. For instance, the secure channel validated here for firmware can be seamlessly readapted for CNC machines or 3D printers to manage the transfer of sensitive G-code or CAD models, demonstrating the framework's versatility for any application requiring the secure exchange of large binary data.

VI. CONCLUSION

Automatic Test Equipment (ATE) is essential in electronics manufacturing but remains isolated from Industry 4.0 infrastructures. We introduced MATE, a methodology that unifies standardized communication, secure firmware management, and service-based orchestration to enable ATE integration from design to deployment. In the evaluated pilot-line setup, the measured execution-time overhead remained between 0.12% and 0.21%, indicating no material throughput degradation under the tested conditions. These results support the use of MATE as an integration approach for test systems in interconnected manufacturing environments.

ACKNOWLEDGMENT

The authors gratefully acknowledge SPEA S.p.A.'s assistance in acquiring and utilizing the ATE machinery employed in this research. Their contribution facilitated the execution of the study and the development of the presented solution.

References

[1] D. R. Carey, "Introduction to Automated Test Systems — Back to Basics," in *2019 IEEE AUTOTESTCON*, 2019, pp. 1–7.
[2] C. Miller, *Chip war: The fight for the world's most critical technology*. Simon and Schuster, 2022.
[3] J. Chae, S. Lee, J. Jang, S. Hong, and K.-J. Park, "A survey and perspective on industrial cyber-physical systems (icps): From icps to ai-augmented icps," *IEEE Transactions on Industrial Cyber-Physical Systems*, vol. 1, pp. 257–272, 2023.
[4] S.-H. Leitner and W. Mahnke, "Opc ua–service-oriented architecture for industrial applications," in *Softwaretechnik-Trends Band 26, Heft 4*. Gesellschaft für Informatik eV, 2006.
[5] X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, "Industry 4.0 and industry 5.0—inception, conception and perception," *Journal of manufacturing systems*, vol. 61, pp. 530–535, 2021.
[6] F. Biondani, F. Tosoni, N. Dall'Ora, E. Fraccaroli, S. Vinco, D. S. Cheng, and F. Fummi, "Special session: D-mate: A design methodology for connecting automatic test equipment in industry 4.0," in *2025 IEEE 26th Latin American Test Symposium (LATS)*, 2025, pp. 1–6.
[7] F. Biondani, D. S. Cheng, and F. Fummi, "Adopting OPC UA for Efficient and Secure Firmware Transmission in Industry 4.0 Scenarios," in

2024 IEEE 33rd International Symposium on Industrial Electronics (ISIE). IEEE, 2024, pp. 1–6.

[8] L. Mostardini, L. Bacciarelli, L. Fanucci, L. Bertini, M. Tonarelli, and M. De Marinis, "Fpga-based low-cost automatic test equipment for digital integrated circuits," in *2009 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. IEEE, 2009, pp. 32–37.

[9] K. Brindley, *Automatic test equipment*. Elsevier, 2013.

[10] W. R. Simpson and J. W. Sheppard, "An intelligent approach to automatic test equipment," in *1991, Proceedings. International Test Conference*. IEEE, 1991, p. 419.

[11] Y.-M. Bae, Y.-G. Kim, J.-W. Seo, H.-A. Kim, C.-H. Shin, J.-H. Son, G.-H. Lee, and K.-J. Kim, "Detecting abnormal behavior of automatic test equipment using autoencoder with event log data," *Computers & Industrial Engineering*, vol. 183, p. 109547, 2023.

[12] Y. Jiang, S. Wu, R. Ma, M. Liu, H. Luo, and O. Kaynak, "Monitoring and defense of industrial cyber-physical systems under typical attacks: From a systems and control perspective," *IEEE Transactions on Industrial Cyber-Physical Systems*, vol. 1, pp. 192–207, 2023.

[13] T. H. Van Nguyen and C.-F. Chien, "Semiconductor probe card proactive maintenance using graph self-supervised learning and an empirical study," *Computers & Industrial Engineering*, vol. 203, p. 110955, 2025.

[14] M. Bejani, D. Appello, M. Mauri, E. Missaglia, and S. Mariani, "Digital twin-assisted optimal sensor placement for real-time monitoring of probe cards in ews applications," in *2025 26th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE)*. IEEE, 2025, pp. 1–6.

[15] B. Han, B. Li, Y. Zhang, P. Feng, K. Wolter, H. Zhang, Y. Li, R. Jurdak, and C. Yuen, "Repeated game-based long-term incentive mechanism for blockchain-enabled reliable federated learning in iiot," *IEEE Internet of Things Journal*, 2025.

[16] B. Han, B. Li, R. Jurdak, P. Zhang, H. Zhang, P. Feng, and C. Yuen, "Pbfl: A privacy-preserving blockchain-based federated learning framework with homomorphic encryption and single masking," *IEEE Internet of Things Journal*, 2025.

[17] B. Han, B. Li, Y. Qi, R. Jurdak, K. Huang, and C. Yuen, "Dp2guard: a lightweight and byzantine-robust privacy-preserving federated learning scheme for industrial iot," *arXiv preprint arXiv:2507.16134*, 2025.

[18] J. Luth, "OPC 10000-1 UA Part 1: Overview and Concepts," *OPC Foundation*, 2022.

[19] U. D. Atmojo, Z. Salcic, I. Kevin, K. Wang, and V. Vyatkin, "A service-oriented programming approach for dynamic distributed manufacturing systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 151–160, 2019.

[20] S. Gaiardelli, S. Spellini, M. Panato, C. Tadiello, M. Lora, D. S. Cheng, and F. Fummi, "Enabling Service-Oriented Manufacturing Through Architectures, Models, and Protocols," *IEEE Access*, vol. 12, pp. 85 259–85 274, 2024.

[21] R. Drath and A. Horch, "Industrie 4.0: Hit or hype?[industry forum]," *IEEE industrial electronics magazine*, vol. 8, no. 2, pp. 56–58, 2014.

[22] W. Babel, "Automation pyramid and solutions business," in *Industry 4.0, China 2025, IoT: The Hype Around the World of Automation*. Springer, 2022, pp. 75–147.

[23] N. Garg, *Apache kafka*. Packt Publishing, 2013.

[24] J. Williams, *RabbitMQ in action: distributed messaging for everyone*. Simon and Schuster, 2012.

[25] Y. D. Dessalk, N. Nikolov, M. Matskin, A. Soylu, and D. Roman, "Scalable execution of big data workflows using software containers," in *Proceedings of the 12th International Conference on Management of Digital EcoSystems*, 2020, pp. 76–83.

[26] OPCFoundation, "UA-ModelCompiler," accessed on Feb 27, 2024. [Online]. Available: https://github.com/OPCFoundation/UA-ModelCompiler

[27] C. Team82, "OPC-UA Exploitation Framework," accessed on Oct 01, 2025. [Online]. Available: https://github.com/claroty/opcua-exploit-framework?tab=readme-ov-file

[28] T. Ji and X. Xu, "Exploring the integration of cloud manufacturing and cyber-physical systems in the era of industry 4.0–an opc ua approach," *Robotics and Computer-Integrated Manufacturing*, vol. 93, p. 102927, 2025.

FRANCESCO BIONDANI (Student Member, IEEE) received the B.Sc. degree in computer science and the M.Sc. degree in computer science from the University of Verona in 2020 and 2023, respectively. He is currently pursuing an industrial Ph.D. degree with the University of Verona in collaboration with Leonardo S.p.a., under the supervision of Prof. Franco Fummi. His research focuses primarily on digital twins and Industry 5.0.

FRANCESCO TOSONI (Student, IEEE) received the M.Sc. degree in computer science and engineering from the University of Verona in March 2022, with a thesis on multidomain fault injection in cyber-physical systems. He is currently working toward a Ph.D. degree at the University of Verona. His main research interests include system simulation and fault injection techniques to ensure functional safety in the context of industrial cyber-physical systems.

NICOLA DALL'ORA (Member, IEEE) received his Ph.D. in Computer Science from the University of Verona, Italy, in 2023. He is currently a Temporary Assistant Professor (RTDa) with the Department of Engineering Sciences at Guglielmo Marconi University, Rome, Italy. He also collaborates as a Research Consultant with the Department of Engineering for Innovation Medicine (Section of Engineering and Physics) at the University of Verona, Italy. His research interests include fault injection techniques, simulation methodologies, and the modeling of complex cyber-physical systems, with a focus on ensuring the reliability and safety of critical applications.

ENRICO FRACCAROLI (Member, IEEE) received the Ph.D. degree in computer science from the University of Verona, Italy, in May 2019. He is currently a Marie Skłodowska-Curie Global Fellow with dual appointments at the University of Verona, Italy, and the University of North Carolina at Chapel Hill, USA. His research interests include the development of new methodologies for efficiently simulating and evaluating the functional safety of embedded platforms composed of analog, digital, and network components.

DONG SEON CHENG (Member, IEEE) received the Laurea and Ph.D. degrees in computer science from the University of Verona, in 2003 and 2008, respectively, with a focus on computer vision and pattern recognition. From 2012 to 2017, he was an Assistant Professor with the Department of Computer Science and Engineering, Hankuk University of Foreign Studies, South Korea, teaching undergraduate and graduate courses. From 2019 to 2022, he was with SETECNA EPC S.r.l., an electronics company in the HVAC Industry. He has been a Research Associate with the University of Verona, since 2023. His research interest includes machine learning. He has published several journals and conference papers in his research field.

FRANCO FUMMI (Member, IEEE) received the Laurea degree in electronic engineering and the Ph.D. degree in electronic and communication engineering from the Polytechnic of Milan, in 1990 and 1994, respectively. Since March 2001, he has been a Full Professor of computer architecture with Università di Verona. He is also leading the Cyber-Physical and IoT Systems Design (CISD) Group, Università di Verona, which comprises more than 20 people and is working on hardware description languages and electronic design automation methodologies for modeling, verification, testing, and optimization of cyberphysical systems. He is also the Co-Founder of two spin-off companies: EDALab, focused on networked embedded systems design; and the automation control software company FACTORYAL.