



Section 3. Computer science

DOI:10.29013/EJTNS-26-5-16-22



DESIGNING MODULAR AI INFRASTRUCTURE FOR STEEL AND METALWORKING OPERATIONS: SEPARATING ARCHITECTURAL INTEROPERABILITY FROM MODEL-LEVEL GENERALIZATION

Nikita Vadimovich Elagin¹

¹ Head of the scientific and technical department for automation and digital manufacturing, Industry Research Institute of Mechanical Engineering, Saint-Petersburg

Cite: Elagin N. V. (2026). *Designing Modular AI Infrastructure for Steel and Metalworking Operations: Separating Architectural Interoperability from Model-Level Generalization*. *European Journal of Technical and Natural Sciences* 2026, No 5. <https://doi.org/10.29013/EJTNS-26-5-16-22>

Abstract

Modular architectures built from containerized microservices and standards-based communication layers deliver scalability and cross-plant portability to artificial intelligence systems in steel and metalworking operations. The interoperability these architectures deliver is a structural property of the deployment layer, separate from a statistical property: whether the AI models the architecture hosts generalize to a machine, material, or lighting configuration different from the one they were trained on. Container-orchestration benchmarks, cross-machine anomaly-detection studies, and steel-surface defect-detection results converge on one conclusion: a fully functioning modular deployment can host a model whose accuracy collapses outside its training distribution, and current reference architectures for industrial AI collapse the two properties into a single, undifferentiated claim of adaptability. Adoption decisions and vendor claims in this domain require two independent checklists, one architectural and one model-specific. Existing predictive-maintenance and computer-vision platforms already embed retraining pipelines that concede the model-level limit their surrounding language papers over.

Keywords: *modular architecture, industrial artificial intelligence, predictive maintenance, computer vision, model generalization, Industry 4.0 interoperability*

Introduction

Roughly 40–50% of the sensing and control assets already installed in U.S. metal-producing facilities would need upgrading before an Industry 4.0-class platform could run on them at all, and the equipment that

remains typically spans several generations of PLCs and ERP systems built for isolated operation, each generation running its own data format (Shargaev, 2025b).

The seam between an architecture's interoperability and a model's generalization

across the machines, materials, and lighting conditions that same architecture serves is the field's central problem. Both properties share one vocabulary, adaptability and flexibility, which is exactly why a platform can be modular in every engineering sense and still fail the moment it meets a CNC cell, alloy grade, or camera angle outside its training data.

Literature Review

A microservice platform built around the MIMOSA interoperability schema decomposes predictive-maintenance functions, ingestion, feature extraction, model serving, into independently deployable services that communicate over REST interfaces, with containers that can be added or removed while the rest of the platform keeps running (Nikolakis et al., 2020). This same decomposability turns out to be compatible with a stricter requirement, formal AI governance: a 2026 reference architecture built to satisfy ISO/IEC 42001's AI-management standard defines four deployment tiers, device, edge, platform, and enterprise, and reports that this layering, validated against a production case, improved system traceability and maintenance-related downtime tracking while leaving the underlying microservice logic intact (Mateo-Casali et al., 2026).

A reference software architecture built to be framework- and technology-agnostic was instantiated separately for three industrial partners in the automotive, energy, and precision-machining domains, and the validation reported that the three instantiations produced diverging scalability and interoperability requirements despite sharing the same underlying blueprint (Haindl et al., 2022). A systematic review of container-based virtualization in real-time industrial systems narrows this to a measurable mechanism: processing latency inside an identical Docker container improved by a factor of 13.9, from 446 to 32 microseconds, simply by swapping the underlying kernel, and the standard deviation in scheduling precision under a demanding co-located workload ran two orders of magnitude larger than under a light one (Queiroz et al., 2023).

A predictive-maintenance platform for CNC-based metal-cutting systems is presented as modular because plants can begin with

a minimal sensor set and expand later, and its edge layer is described, in the same sentence, as accommodating different hardware configurations while its machine-learning pipelines retrain themselves as new datasets accumulate (Shargaev, 2025 a). The first half of that sentence, hardware accommodation and incremental sensor deployment, is an architectural claim of exactly the kind Nikolakis et al. (2020) and Mateo-Casali et al. (2026) document. The second half, pipelines that retrain themselves as data accumulates, is a different kind of claim: it concedes that the model must first see new data from the new deployment before its predictions earn trust there, which is precisely the generalization gap that Haindl et al.'s (2022) diverging site requirements and Queiroz et al.'s (2023) kernel-dependent latency locate outside the architecture layer.

Trained-model accuracy in this literature is a property of the specific dataset a model was evaluated on, once the deployment layer is held constant and only the input data varies. A one-dimensional convolutional network trained on sensor data from CNC and machining-center assets reaches 91.57% accuracy on held-out data from the same equipment, outperforming random forest, gradient boosting, and several other classifiers trained on the identical feature set (Athar et al., 2024); the result establishes a machine-matched accuracy ceiling specific to the equipment it was evaluated on. A CNN-LSTM hybrid architecture for predictive maintenance reaches 96.1% accuracy and a 95.2% F1-score specifically against the best of three named industrial sensor datasets, outperforming standalone CNN and LSTM baselines on those same datasets (Li & Li, 2025); the 96.1% figure is a property of the three datasets it was measured on, and applies to CNC or metal-cutting deployment only to the extent a new site's data resembles them. Steel-surface defect detection produces the identical pattern in a visual: a YOLOv5-based detector augmented with an attention mechanism and a bidirectional feature pyramid network reports accuracy gains on the steel surface defect categories present in its training set (Gao et al., 2024), specifically a mean Average Precision of 75.7%, a 4.2-percentage-point gain over the baseline model, at a throughput of 70 frames per

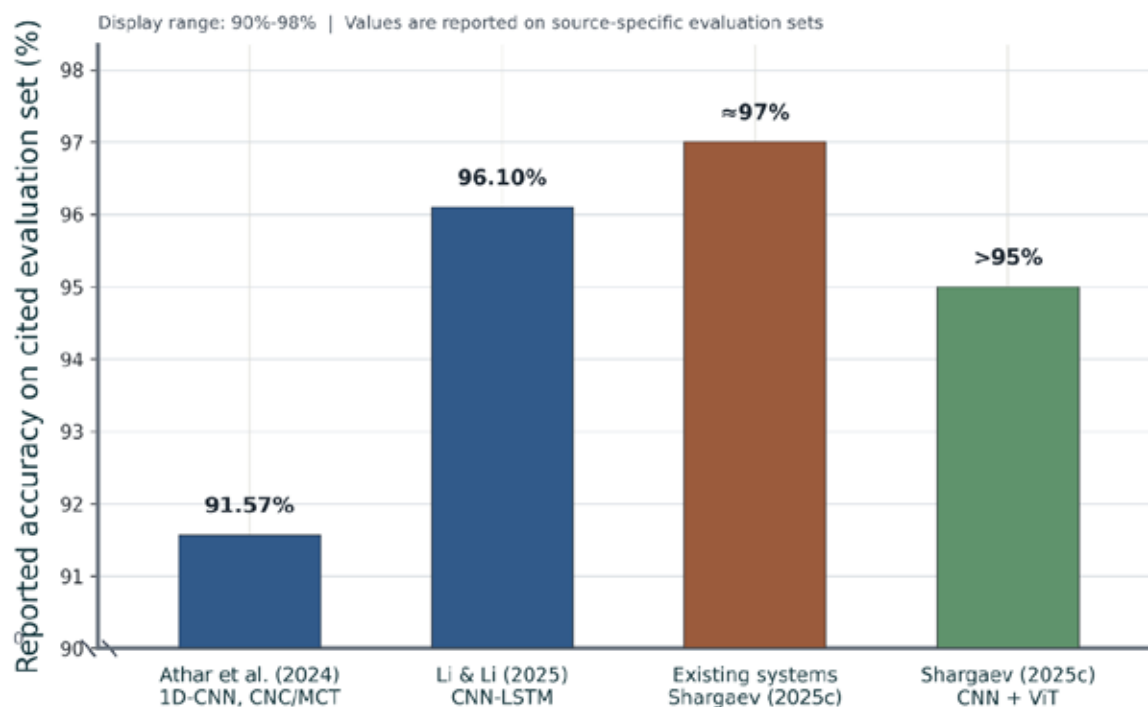
second (Gao et al., 2024); and an earlier detector built specifically to address small and structurally complex defect targets on steel surfaces reports its improvement against that same constraint, detection performance

measured relative to the defect population the model was shown during training (Zhao et al., 2021). Each of these four results is positive and reproducible, tied to a named dataset.

Table 1. Accuracy figures cited in this review, each anchored to a specific evaluation set

Source	Task / domain	Reported accuracy	Evaluation basis
Athar et al. (2024)	Anomaly detection, 1D-CNN, CNC / machine-center sensor data	91.57%	Held-out data from the same equipment
Li & Li (2025)	Predictive maintenance, CNN-LSTM hybrid	96.1% (F1=95.2%)	Best of three named industrial sensor datasets
Existing AI inspection systems (per Shargaev, 2025 c, prior-art comparison)	Metal defect detection, single-model CNN	~97%	Trained defect types only; new types require new training
Shargaev (2025 c), invention	Metal defect detection, CNN + ViT hybrid	>95%	Known defect types shown during training
Gao et al. (2024)	Steel surface defect classification, YOLOv5 + SE attention + BiFPN	75.7% mAP (+4.2 pts), 70 FPS	Defect categories present in training set
Zhao et al. (2021)	Steel surface defect detection, modified Faster R-CNN	75.2% mAP (+12.8 pts); per-class range 50.1%-90.5%	Defect population shown during training

Figure 1. Reported accuracy figures are anchored to the dataset or defect population each source evaluated



The patent for a computer-vision quality-analysis system embedded in metal-cutting lines documents this same ceiling from inside its own prior-art comparison. Existing AI-based inspection systems, the patent states, reach approximately 97% accuracy on trained defect types, but their adaptability is limited to their training data, and new defect types require new training before the system recognizes them (Shargaev, 2025 c). Read against Athar et al.'s and Li and Li's dataset-anchored accuracy figures, the patent's own architecture functions as an engineering acknowledgment of the model-generalization limit: the system retraines because every accuracy figure reported in this literature ties performance strictly to the data the model was shown, leaving new geometries, materials, and lighting rigs to be earned through the same retraining cycle.

Plants adopt modular, standards-based architecture for reasons that sit entirely on the standardization side of the ledger, independent of the generalization problem identified above; the standardization gap it closes is documented as severe enough on its own terms to justify the investment regardless of what happens to model accuracy afterward. A RESTful binding for OPC UA was proposed

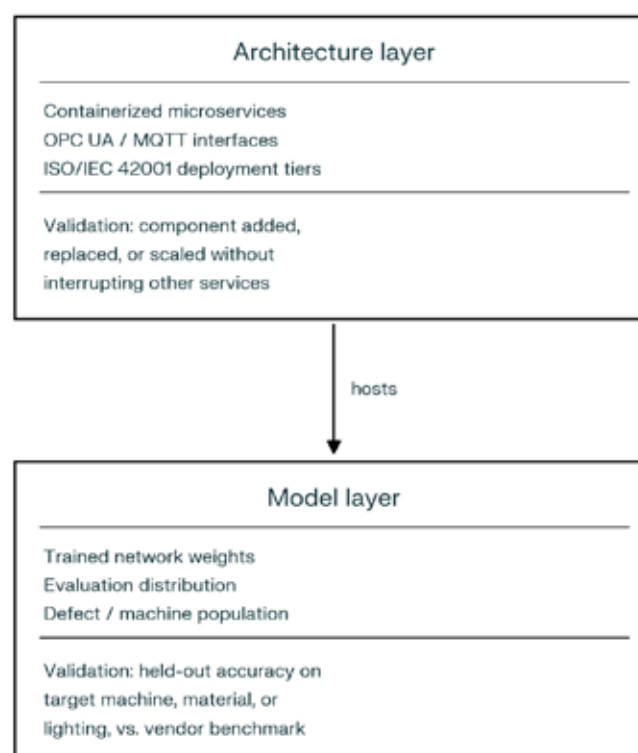
as a lighter-weight alternative to the standard's native service-oriented interface for accessing industrial data (Grüner, Pfrommer, & Palm, 2016), and the companion specifications built on top of OPC UA were designed so that devices from different manufacturers interpret the same variable without custom mapping (Schleipen, Lüder, Sauer, Flatt, & Jasperneite, 2016).

In U.S. metallurgy specifically, adoption of OPC UA and comparable standards is described as uneven, with different generations of PLCs and ERP systems installed in the same plant still failing to interoperate even where a nominal standard is present (Shargaev, 2025 b). This plant-level condition is what makes modular retrofit strategies attractive, independent of how mature any individual AI model is: plants adopt containerized, standards-based platforms because the alternative, a monolithic system assuming uniform legacy infrastructure, requires exactly the uniform starting point these plants lack.

Discussion

Modular architecture works, and the AI models tested in this literature perform at the levels their authors report.

Figure 2. *The architecture layer and the model layer*



Contrary to Haindl et al.'s (2022) framing of a framework- and technology-agnostic reference architecture as a blueprint that any manufacturing context can adopt through direct derivation, their own validation across automotive, energy, and precision-machining partners reports diverging scalability and interoperability requirements at the point of instantiation. This is a substantial finding: the framework-agnostic property holds at the

level of documentation and component naming, while the property that determines deployment cost and risk, how much site-specific data collection and model retraining each instantiation needs, is generated by the AI models the architecture hosts. A reference architecture agnostic to technology stays fully exposed to data, and the divergence Haindl et al. report across three unrelated manufacturing domains demonstrates exactly that exposure.

Table 2. *A two-checklist framework for evaluating modular AI platforms in steel and metalworking operations*

Checklist	Question it answers	Representative evidence
Architecture checklist	Can a component be added, replaced, or scaled while the rest of the system keeps running?	Nikolakis et al. (2020); Mateo-Casali et al. (2026); Queiroz et al. (2023)
Model checklist	Has the specific trained model deployed inside that architecture seen data resembling the target machine, material, or lighting condition?	Athar et al. (2024); Li & Li (2025); Shargaev (2025 c, prior-art comparison)

Under this separation, the computer-vision patent's own retraining subsystem reads as the correct engineering response to a limit the patent itself documents in its prior-art comparison: existing systems reach roughly 97% accuracy on trained defects but require new training for new ones (Shargaev, 2025c), and the invention's answer makes that retraining cycle a built-in, periodic feature of the architecture. That is a genuine, self-contained contribution to the architecture checklist. Set beside the four-source spread in Table 1, that ~97% figure sits at the same order of magnitude as Li and Li's (2025) 96.1% and above Athar et al.'s (2024) 91.57%, while the two visual-inspection results added above (75.7% and 75.2% mAP) sit roughly 20 points lower: the gap tracks task type, sensor-based anomaly detection versus image-based defect localization, at least as strongly as it tracks any architectural choice, which is further evidence that the accuracy number belongs to the model-and-task pairing.

This separation also reframes what standards-adoption barriers actually buy a plant. When OPC UA or an equivalent protocol closes the interoperability gap that Grüner et al. (2016) and Schleipen et al. (2016) address, and when Ladegourdie and

Kua's (2022) performance evaluation confirms the protocol remains feasible at the edge, a plant clears the architecture checklist: components from different vendors and different equipment generations exchange data in a common format. Solving the architecture checklist leaves the model checklist exactly where it was: the persistent obstacles that Hamdi Alaoui et al. (2026) identify under the heading of model robustness and cross-condition heterogeneity belong to the model checklist on their own terms, independent of how much standardization the plant has achieved. Conflating the two lets a genuinely modular, standards-compliant platform earn the label adaptive for its plumbing alone, as the CNC predictive-maintenance case above shows, while the label quietly expands to cover the model running inside it.

Conclusion

The distinction between an architecture's interoperability and a model's generalization, the seam separating two literatures that currently share one vocabulary, resolves a specific practical ambiguity in how modular AI infrastructure for steel and metalworking operations should be evaluated before purchase or deployment. A plant evaluating a predictive-maintenance or computer-vision platform

gains little by confirming only that the platform's services are containerized, standards-compliant, and independently scalable; the evidence reviewed here shows that property is now achievable and documented across two independent architectural traditions spanning 2020 to 2026. The same evidence stops short of supporting one further step: treating that achievement as a proxy for whether the specific model inside the platform performs on the plant's own machines, alloys, and lighting rig before first collecting site-specific data and running it through a retraining cycle of the kind the patented computer-vision system described above already builds in as a first-class feature. A vendor evaluation checklist built on this distinction requires accuracy figures reported separately for the target site's own defect population, material grade, and

lighting configuration, standing apart from the single aggregate figure inherited from the vendor's original training set. Testing whether such site-specific figures diverge from the vendor's published benchmark, and by how much, is the next empirical step the field owes its adopters.

Every accuracy figure in Table 1, 75.2% to 97%, comes from the same hardware, defect population, or sensor rig the model trained on, including Athar et al.'s (2024) 23.3-percentage-point spread across ten classifiers. Two adoption risks in this field already have numbers attached: roughly 40–50% of legacy sensing and control assets need upgrading before an Industry 4.0 platform runs at all (Shargaev, 2025b), and swapping a container's kernel alone can shift latency by 13.9x (Queiroz et al., 2023).

References

- Athar, A., Mozumder, M. A. I., Abdullah, et al. (2024). Deep learning-based anomaly detection using one-dimensional convolutional neural networks (1D CNN) in machine centers (MCT) and computer numerical control (CNC) machines. *PeerJ Computer Science*, – 10. – e2389. URL: <https://doi.org/10.7717/peerj-cs.2389>
- Gao, Y., Lv, G., Xiao, D., Han, X., Sun, T., & Li, Z. (2024). Research on steel surface defect classification method based on deep learning. *Scientific Reports*, – 14. – 8254 p. URL: <https://doi.org/10.1038/s41598-024-58643-1>
- Grüner, S., Pfrommer, J., & Palm, F. (2016). RESTful industrial communication with OPC UA. *IEEE Transactions on Industrial Informatics*, – 12(5). – P. 1832–1841. URL: <https://doi.org/10.1109/TII.2016.2530404>
- Haindl, P., Buchgeher, G., Khan, M., & Moser, B. (2022). Towards a reference software architecture for human-AI teaming in smart manufacturing. In *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: New Ideas and Emerging Results* (P. 96–100). URL: <https://doi.org/10.1145/3510455.3512788>
- Hamdi Alaoui, A., Meddaoui, A., & Hain, M. (2026). AI-driven predictive maintenance for industry 4.0: A systematic review of models, methods, and challenges. *The International Journal of Advanced Manufacturing Technology*, – 143. – P. 4861–4876. URL: <https://doi.org/10.1007/s00170-026-17531-w>
- Ladegourdie, M., & Kua, J. (2022). Performance analysis of OPC UA for industrial interoperability towards Industry 4.0. *IoT*, – 3(4). – P. 507–525. URL: <https://doi.org/10.3390/iot3040027>
- Li, W., & Li, T. (2025). Comparison of deep learning models for predictive maintenance in industrial manufacturing systems using sensor data. *Scientific Reports*, 15, Article 23545. URL: <https://doi.org/10.1038/s41598-025-08515-z>
- Mateo-Casali, M. A., Maaßen, M., Heymann, H., Boza, A., Fraile, F., Leyendecker, I., Grunert, D., & Schmitt, R. H. (2026). Reference architecture for the design and implementation of AI systems in manufacturing in conformity to ISO/IEC 42001. *International Journal of Computer Integrated Manufacturing*, – P. 1–23. URL: <https://doi.org/10.1080/0951192X.2026.2664187>
- Nikolakis, N., Marguglio, A., Veneziano, G., Greco, P., Panicucci, S., Cerquitelli, T., Macii, E., Andolina, S., & Alexopoulos, K. (2020). A microservice architecture for predictive analytics

- in manufacturing. *Procedia Manufacturing*, – 51. – P. 1091–1097. URL: <https://doi.org/10.1016/j.promfg.2020.10.153>
- Queiroz, R., Cruz, T., Mendes, J., Sousa, P., & Simões, P. (2023). Container-based virtualization for real-time industrial systems: A systematic review. *ACM Computing Surveys*, – 56(3). Article 59. – P. 1–38. URL: <https://doi.org/10.1145/3617591>
- Schleipen, M., Gilani, S.-S. Bischoff, T., Pfrommer, J. (2016). OPC UA & Industrie 4.0: Enabling technology with high diversity and variability. *Procedia CIRP*, – 57. – P. 315–320. URL: <https://doi.org/10.1016/j.procir.2016.11.055>
- Shargaev, V. (2025a). Smart predictive maintenance platform for CNC machines. LAP LAMBERT Academic Publishing.
- Shargaev, V. (2025b). Digital transformation in metallurgy: Challenges and opportunities. *Open Access Research Journal of Engineering and Technology*, – 9(2). – P. 78–83. URL: <https://doi.org/10.53022/oarjet.2025.9.2.0098>
- Shargaev, V. (2025c, December 17). Computer vision system for quality analysis in metal production (U. S. Provisional Patent Application No. 63/943, 256). United States Patent and Trademark Office.
- Zhao, W., Chen, F., Huang, H., Li, D., & Cheng, W. (2021). A new steel defect detection algorithm based on deep learning. *Computational Intelligence and Neuroscience*, 2021, Article 5592878. URL: <https://doi.org/10.1155/2021/5592878>

submitted 15.09.2026;
accepted for publication 29.09.2026;
published 30.09.2026
© Elagin N. V.
Contact: n.v.elagin.science84@gmail.com