CASE STUDY

From Tribal Knowledge to Smart Factory — How Lumi Built RunPilot Overnight

Proving the Agentic SDLC in Manufacturing

A CES Global Technical Case Study

2026

Ankur Nair & Anurag Sharma

CO-CREATORS OF LUMI, CES LTD

"The factory of the future will have only two employees: a man and a dog. The man will be there to feed the dog. The dog will be there to keep the man from touching the equipment." — Warren Bennis

Table of Contents

1. Introduction & Context

A Framework Made Concrete

In June 2026, CES published The Agentic SDLC: How AI is Transforming Software Delivery — a technical whitepaper establishing the theoretical and empirical foundation for a new generation of AI-orchestrated software delivery. That document made three core arguments: that traditional SDLC has a structural ceiling imposed by human throughput; that isolated copilot tools represent a first-order improvement, not a transformation; and that Lumi — CES's Agentic SDLC platform — demonstrates a multi-agent, memory-persistent architecture that genuinely changes what is possible.

This case study is the proof of the third argument, applied to one of the oldest and most complex industrial domains: manufacturing.

On the night of July 7—8, 2026, Lumi autonomously constructed RunPilot: a full edge-compute SCADA (Supervisory Control and Data Acquisition) platform for smart factories. The output was not a prototype. It was a deployable system with 110+ Python source files, 40+ passing tests, 8 PLC protocol drivers, a visual workflow builder, an on-edge AI assistant with RAG (Retrieval-Augmented Generation), 6 KPI insight applications, 11 output connectors, and a security architecture reviewed by a parallel adversarial audit that surfaced 82 findings across 65 files.

The overnight build happened because of what Lumi knew before a single line of RunPilot code was written. This is not a story about code generation speed. It is a story about **knowledge cloning** — and what becomes possible when accumulated domain expertise can be applied at machine speed across every phase of software delivery.

Why Smart Factory Implementations Fail

Manufacturing is the most rigorous test case for the Agentic SDLC thesis because it is a domain where the barriers to digital transformation are both well-documented and stubbornly persistent.

The statistics are sobering. Traditional SCADA and MES (Manufacturing Execution System) implementations cost SMB manufacturers **$500,000 to $1.5 million** for a combined on-premise deployment — and that figure excludes the domain consultant engagement that often costs an additional $300,000—$500,000 to extract operational knowledge from the people who run the plant. Hidden costs (integration, validation, training) typically represent 60—70% of total investment. Annual maintenance runs 15—25% of initial software cost — compounding forever.

The result: **75% of manufacturing companies cite high upfront costs as the primary barrier** to Industry 4.0 adoption (McKinsey/Gartner, 2024). **Only 10% of manufacturing companies are fully digitized** (PwC). And despite a decade of Industry 4.0 enthusiasm, **74% of manufacturers remain stuck in pilot purgatory** — unable to scale initial use cases beyond a single production line.

The three failure modes that explain this gap:

1. **Cost complexity.** Incumbent platforms (Ignition, Litmus Edge, HighByte, AVEVA, Siemens WinCC) charge per I/O tag, per device, per server, or via opaque negotiated contracts. Even Ignition — the disruptive pricing leader — starts at $3,280 per server and requires integrator engagement ($10,000–$50,000 per deployment) for configuration. Litmus Edge averages $12,167 per monitored asset per year.

2. **Knowledge silos.** The people who know how the factory actually works — the maintenance engineer who knows Compressor 3 runs hot on humid days, the process engineer who tightened the fill volume spec after a 2019 customer complaint, the shift supervisor who can hear when a bearing is starting to wear — are not the people who configure SCADA software. Domain knowledge and digital tooling live in separate organizations, with consulting engagements as the expensive, lossy bridge between them.

3. **Vendor dependency.** Traditional platforms require professional services for every change: a new PLC model, a new sensor type, a new analytics requirement. The plant team cannot iterate without filing a ticket with IT or waiting for an integrator's next available slot. Digital transformation becomes something that happens to a factory, not something the factory does for itself.

RunPilot was designed — and built by Lumi — to dismantle all three barriers simultaneously.

2. The Knowledge Cloning Phenomenon

What Tribal Knowledge Actually Looks Like

Every manufacturing operation sits on a reservoir of operational knowledge that exists in no document. It lives in the head of the veteran maintenance engineer. In the shift supervisor's gut feel for the machine's sounds. In the process engineer's memory of every parameter change and its consequence.

This tribal knowledge is simultaneously the most valuable and the most fragile asset in manufacturing. When the veteran retires, it walks out the door. When a new hire joins, the transfer takes months — and is never complete. When a consultant arrives to scope a digital transformation, extracting tribal knowledge is itself a $300,000—$500,000 engagement. The result is typically a static document that begins decaying the moment it is written.

Lumi's memory architecture changes this equation. Not by replacing tribal knowledge with AI — but by making accumulated expertise **operationally available:** retrievable by semantic meaning at the moment of need, composable across domains, and applicable by specialist agents without human mediation.

Lumi's 4-Tier Memory Consolidation Applied to Manufacturing

The four-tier memory pipeline described in the Agentic SDLC whitepaper (Figure 3) operates on any organizational knowledge that accumulates through work. For RunPilot, the relevant manufacturing domain expertise had accumulated over months of CES engagement with industrial systems — protocol specifications, operational patterns, market analysis, security requirements, and equipment knowledge. The consolidation pipeline made all of it operationally available:

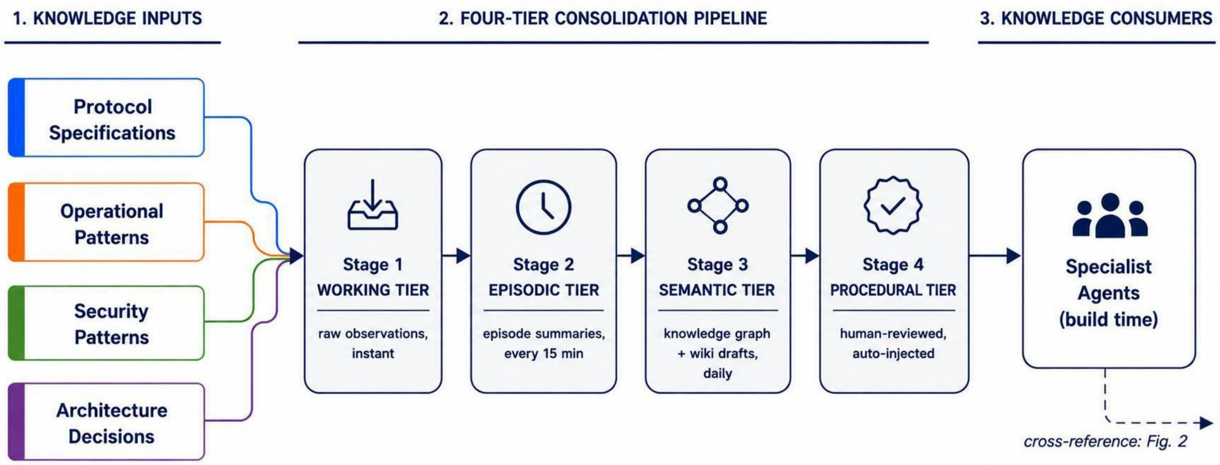

Figure 1: Manufacturing domain knowledge accumulates through Lumi's four-tier pipeline and is retrieved by specialist agents at build time.

The specific knowledge that made RunPilot possible — and that would have taken months to assemble through conventional means — included:

Protocol specifications. OPC-UA's node ID addressing formats (ns=2;s=Motor.Speed) and security policy negotiation. Modbus's register-to-function-code mapping, byte-order handling, and connection limits. Siemens S7's requirement to disable "Optimized Block Access" in TIA Portal. EtherNet/IP's symbolic tag naming convention. MELSEC's mandatory port configuration in GX Works. Omron FINS's UDP codec and node-address convention.

Operational patterns. How ISA-18.2 alarm state machines work (active → acknowledged → cleared, with hysteresis to prevent chattering). Why alarm flood suppression matters during cascade failures. How OEE is correctly calculated — the subtlety that "performance" measures actual rate against ideal rate per minute (not per hour) to capture micro-stops. What a proper shift handover requires: alarms, production counts, downtime events, KPI summaries, and the human context that data alone cannot supply.

Security patterns. Why PLC write-back requires defense in depth — a misconfigured write to an industrial controller can damage a $50,000 servo motor or cause a safety incident. The specific attack surfaces in industrial systems: eval() on config expressions, path normalization bypasses on authentication middleware, timing attacks enabling username enumeration. The OT security principle that write paths and read paths must be treated as structurally different risk surfaces.

Architecture decisions. Why edge SCADA systems use dual databases (fast time-series store for telemetry — DuckDB in RunPilot's case — combined with a relational store for configuration and metadata — SQLite). Why store-and-forward buffering is non-optional on the OT/IT boundary (airgaps, bandwidth constraints, and intermittent connectivity are features of industrial networks, not bugs). Why NATS is preferable to direct in-process queues for an edge platform that must survive individual stage failures.

The Enterprise Knowledge Clone

The traditional alternative is a discovery engagement: consultants interview subject matter experts, observe operations, review existing documentation, and synthesize findings into a requirements document — a process that takes 2–4 months and costs $300,000–$500,000. That document then serves as input to a development team that builds over 6–18 months, losing context at every handoff.

Lumi compressed both the discovery and the construction into a single session because the knowledge was already operational in its memory system. The expertise of an automation engineer, a process engineer, a security architect, and an industrial IT specialist — ingested once through the four-tier pipeline, applied in parallel across specialist agents, at machine speed.

This is knowledge cloning: not copying documents, but creating an operational replica of domain expertise that behaves like accumulated experience rather than stored text.

> **The key distinction:** A document management system stores knowledge. Lumi's memory architecture makes knowledge operationally available — retrievable by semantic meaning, composable across agents, and applicable without human mediation. The same manufacturing knowledge that built RunPilot can build the next platform, inform the next consulting engagement, or onboard the next engineer. Traditional consultant knowledge walks out the door with the consultant.

OVERNIGHT BUILD

3. Lumi Orchestrates RunPilot

From Goal to Sprint Board: Task Decomposition

The build began with a single goal statement: construct an edge-compute SCADA platform for smart factories with PLC connectivity, real-time processing, visual workflow building, on-device AI, and cloud sync.

Lumi's team lead agent decomposed this into 10 stories — each a self-contained deliverable with clear acceptance criteria, dependency ordering, and specialist assignments:

STORY	SCOPE	AGENT
1. Project Foundation & Config	Python project structure, Pydantic v2 config schema, CLI, Docker Compose, Protobuf definitions, structured logging	Backend Architect
2. PLC Driver Layer	Driver protocol interface, 8 adapters (OPC-UA, Modbus TCP/RTU, S7, EtherNet/IP, MELSEC, FINS, MQTT), connection manager with exponential backoff	SCADA/PLC Engineer
3. Data Model & Asset Registry	ISA-95 asset hierarchy, SQLite + DuckDB schemas, ISA-18.2 alarm state machine, OEE models	SCADA/PLC Engineer
4. Streaming Pipeline & Output Connectors	NATS bus, 3-stage validate/enrich/aggregate pipeline, Protobuf wire format, 11 connectors, ring buffer + WAL	Streaming Data Architect
5. Workflow Engine	YAML DAG schema, builder + cycle detection, 15+ transform types, hot-reload runtime, error policies	Backend Architect
6. On-Device AI & Knowledge	LLM fallback chain, LanceDB RAG pipeline, anomaly detection, tool-calling agent, operational skills	AI Engineer
7. Security & Compliance	bcrypt auth, 4-role RBAC, TLS support, audit logging, PLC WriteGuard with rate limiting	Security Auditor
8. Web UI & REST API	FastAPI + SSE, React SPA, device management, visual workflow builder (React Flow), dashboard tiles, AI chat	Frontend Developer
9. Edge Optimization & Deployment	Token bucket bandwidth governor, 3-tier retention policy, hardware profiles, Docker Compose, watchdog, graceful degradation	Edge Compute Engineer
10. Cloud Sync & Operations	Delta sync protocol, store-and-forward queue, MQTT/HTTPS transport, bandwidth budgeting, backup/restore	Edge Compute Engineer

The decomposition itself demonstrates knowledge cloning. The 10-story structure reflects deep understanding of SCADA architecture: the separation between driver and pipeline layers, the distinction between asset model and telemetry store, the isolation of security as a cross-cutting concern, the elevation of edge optimization to a first-class story rather than a deployment afterthought. An inexperienced architect would merge Stories 2 and 3, omit Story 9, and collapse security into "add auth.“ The decomposition here reflects the judgment of someone who has built and maintained industrial systems.

The Specialist Agent Roster

Seven specialist agents executed the build, each carrying a persona tuned to its domain:

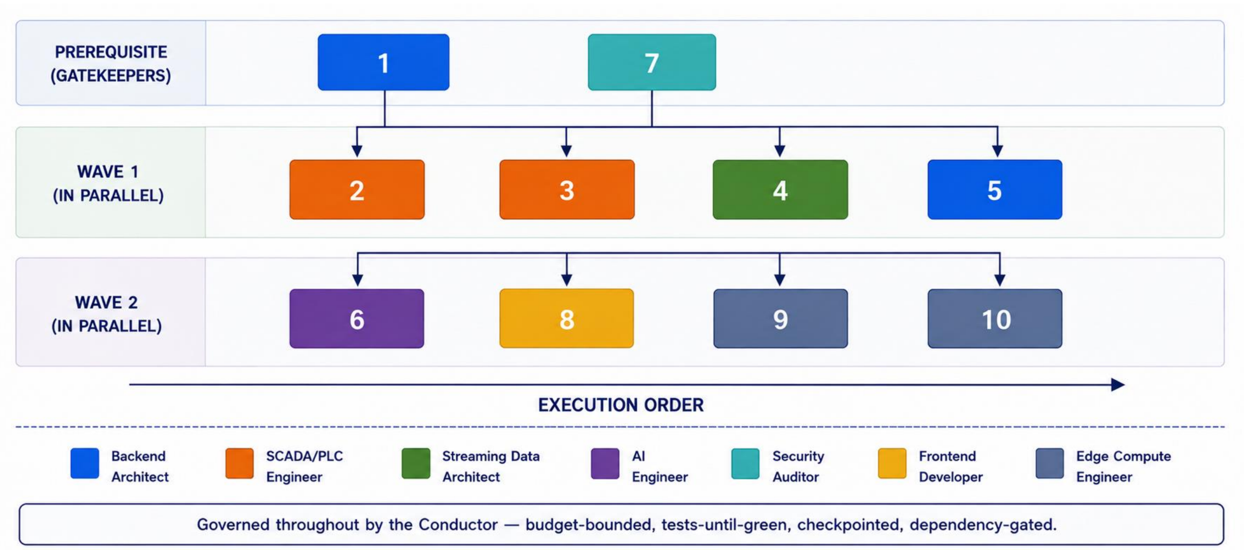

Figure 2: Multi-agent orchestration for RunPilot — seven specialists across two parallel execution waves, governed by a team lead agent.

The Conductor's Autonomous Execution

Lumi's Conductor — the autonomous execution engine described in Section 3 of the Agentic SDLC whitepaper — managed the session with four mechanisms that distinguish it from a simple task queue:

Budget-bounded execution. Every agent operated within a token and cost budget. The Conductor estimated cost before spawning each sub-agent and blocked preemptively when predicted cost would exceed the remaining budget — preventing the runaway-agent problem that plagues unbounded autonomous execution.

Tests-until-green loops. Each story included acceptance criteria expressed as executable tests. The Conductor ran verification after each major code generation step. When tests failed, failure output and logs were fed back to the agent with a bounded retry budget to diagnose and fix.

Successful fixes were recorded as verified learning in the knowledge graph. This is why the output included 40+ passing tests — not just generated code.

Checkpoint-based durability. Every step was checkpointed to SQLite. A network interruption, machine restart, or process termination would have resumed from the last checkpoint, not from the beginning. For an overnight build, this is the difference between finding a completed platform in the morning versus finding a crashed process.

Parallel execution with dependency management. Stories 1 (Foundation) and 7 (Security) were prerequisites — they ran first, establishing the shared Protobuf wire format and security primitives that subsequent stories depended on. Stories 2, 3, 4, and 5 ran concurrently in the first parallel wave; Stories 6, 8, 9, and 10 ran concurrently in the second. Total wall-clock time was determined by the longest sequential chain, not the sum of all story durations.

The Adversarial Quality Gate

The security audit that followed construction is the most striking demonstration of Lumi's self-correction architecture. A security auditor agent — operating independently from all agents that built the code — reviewed all 65 source files and surfaced 82 unique findings across 7 severity-ranked phases:

PHASE	FOCUS	FINDINGS	SEVERITY DISTRIBUTION
1. Stop the Bleeding	Exploitable vulnerabilities	10	5 CRITICAL, 5 HIGH
2. Async & Reliability	Event loop blockers, silent	8	7 HIGH, 1 MEDIUM
3. Data Integrity	Input validation, edge cases	14	2 HIGH, 12 MEDIUM
4. Security Hardening	Defense in depth	15	4 HIGH, 11 MEDIUM
5. Docker & Deployment	Container security	7	1 HIGH, 6 MEDIUM
6. Observability & Cleanup	Dead code, logging gaps	6	1 HIGH, 5 LOW
7. Test Coverage	Missing test categories	9	2 HIGH, 7 MEDIUM

This is Raskolnikov's Psychological Loop (RPL) — Lumi's adversarial review system — applied post-construction at file-by-file granularity. Critical findings included:

- **RCE via eval() on config-supplied expressions** — trivially escapable sandbox (CRITICAL)

- **Auth bypass via path normalization** — trailing and double slashes bypassing exact-match PUBLIC_PATHS (CRITICAL)
- **Username enumeration via timing attack** — missing-user returns instantly vs. bcrypt's ~200ms for valid users (CRITICAL)
- **PLC write_tag() bypassing WriteGuard RBAC** — writes to physical equipment without permission check (HIGH)
- **Synchronous SQLite calls in async handlers** — blocking SSE heartbeats and event loop (HIGH)

The audit also documented positive findings: yaml.safe_load() used correctly (no deserialization attacks), all SQLite queries parameterized (no SQL injection), secrets.token_urlsafe(32) for session tokens (adequate entropy), and 4-zone Docker network isolation properly configured.

No single developer — regardless of expertise — reviews 65 files with this level of consistency. The adversarial architecture ensures that the reviewer is structurally incentivized to find problems, not confirm correctness. This is the same principle that makes independent code review valuable in human teams, scaled to the thoroughness of a dedicated security audit.

The Build in Numbers

From zero to a deployable platform in one overnight session:

CATEGORY	COUNT
Python source files	110+
Passing tests (backend + frontend)	40+
PLC protocol drivers	8
Workflow transform types	15+
Output connectors	11
KPI insight apps	6
Knowledge base manuals	6
Operational skills (markdown playbooks)	4
Security audit findings (actionable, phased)	82

4. The Platform

Architecture: Edge-First, Self-Contained, AI-Native

RunPilot is a single-process edge-compute SCADA platform designed to run on commodity hardware — a Raspberry Pi 4 or 5, an NVIDIA Jetson Nano, an industrial mini-PC, or any Linux system with modest resources. The architecture reflects four design principles: edge-first execution, minimal external dependencies, progressive connectivity (works fully offline, syncs when connected), and human-confirmed AI actions.

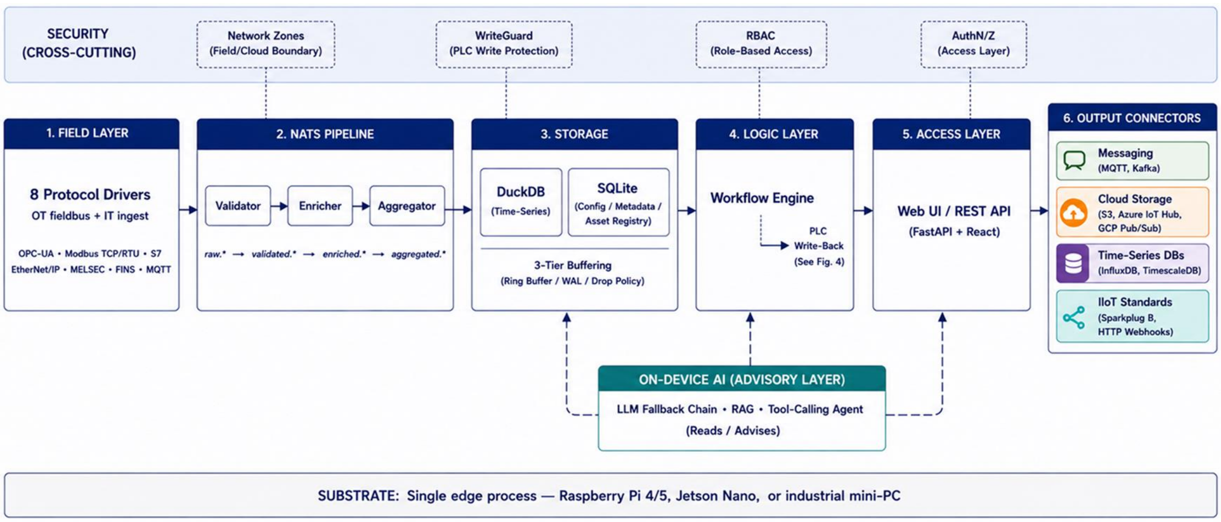

Figure 3: RunPilot system architecture — from PLC connections through the NATS pipeline to cloud connectors, all running on a single edge process.

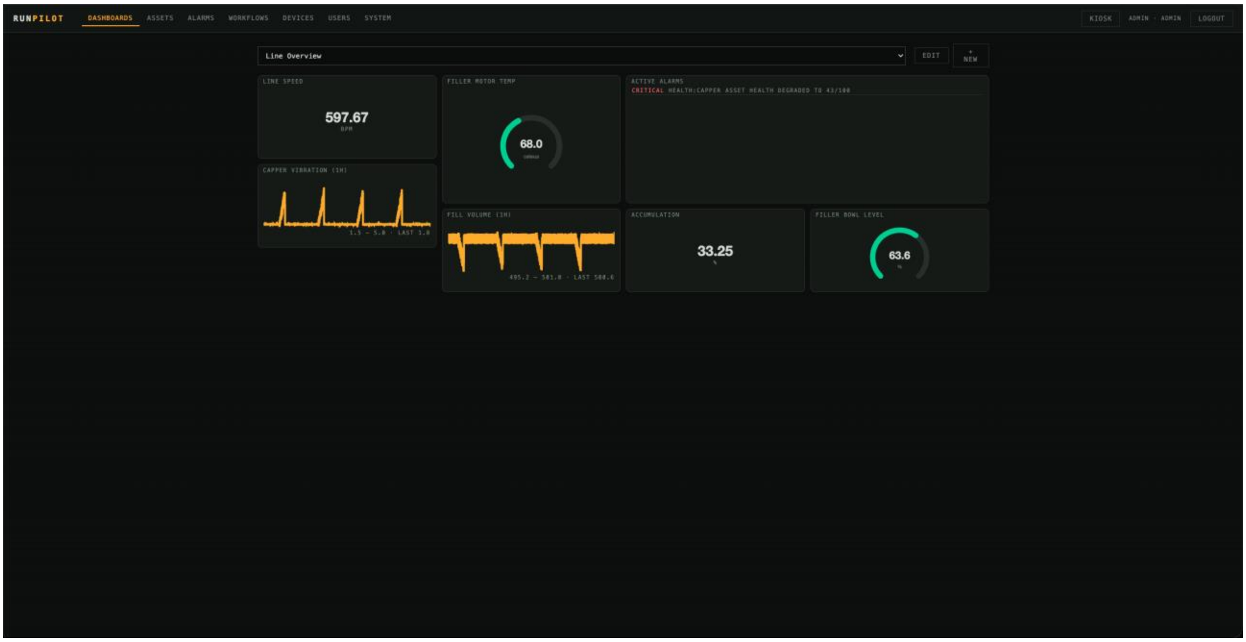

Figure 4: RunPilot Line Overview dashboard — live KPIs including line speed, motor temperature, active alarms, capper vibration trend, fill volume, and filler bowl level.

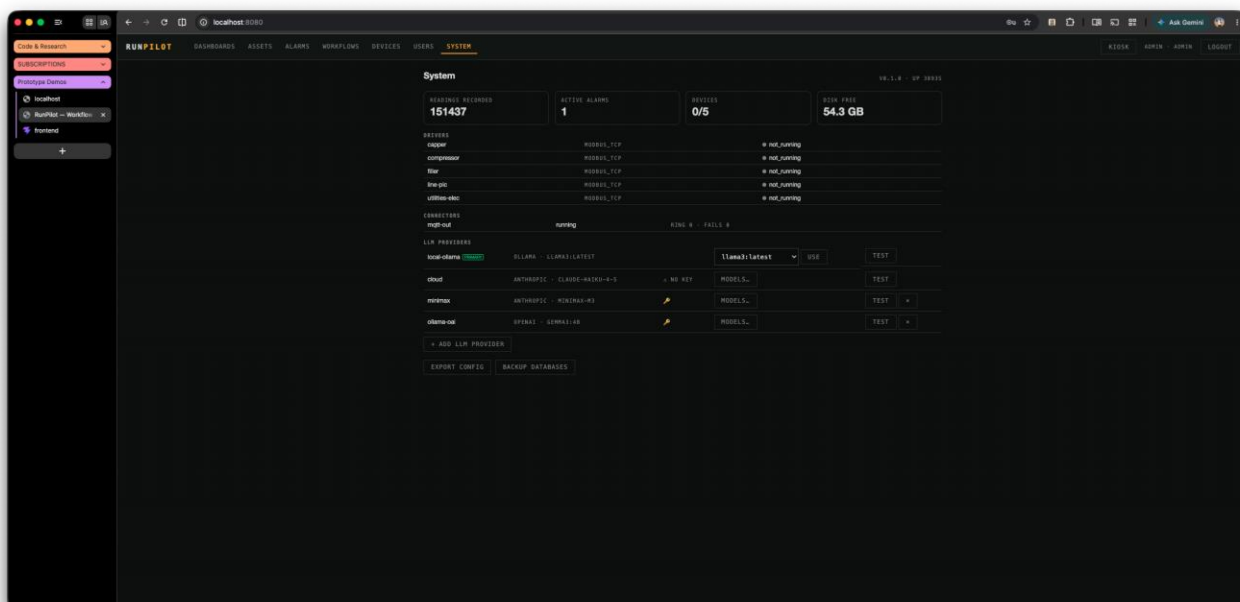

Figure 5: RunPilot System administration page — device configuration, data volumes (151K+ records), connection management, and cloud sync settings.

8 PLC Protocol Drivers

RunPilot ships eight production-grade protocol drivers — each implementing a consistent interface (connect, disconnect, read_tags, write_tag, subscribe, health_check) with exponential backoff reconnection and connection-pool management:

PROTOCOL	DRIVER ID	LIBRARY	ADDRESSING	NOTABLE
OPC-UA	opcua	asyncua	ns=2;s=Motor.Speed	Subscriptions + cert auth; security policy negotiation
Modbus TCP	modbus_tcp	pymodbus	40001 (holding), 10001 (input)	Register batching; byte-order config
Modbus RTU	modbus_rtu	pymodbus[serial]	Same as TCP	/dev/ttyUSB0; baud/parity in extra:
Siemens S7	s7	pys7 (pure Python)	DB1,R40, M54.4, IW22	S7-300/400/1200/1500; multi-var reads
EtherNet/IP	ethernet_ip	pycomm3	Program:Main.Counter, MyTag	ControlLogix/CompactLogix symbolic tags
MELSEC MC	melsec	pymcprotocol	D100, M100, X10	Q/L/iQ-R series; port mandatory from GX Works
Omron FINS	fins	stdlib UDP codec	DM100, CIO50, W10	Word areas; node = last IP octet
MQTT Ingest	mqtt	aiomqtt	topic or topic:$.json.path	Push/cache; numeric and boolean payloads

Every driver configuration is managed from the browser — connection forms, test-connection validation, tag browsing (OPC-UA and EtherNet/IP), CSV bulk import, and an AI tag mapper that converts cryptic addresses (N7:0, DB100.DBW4) into readable names and units. No YAML editing required for day-to-day operations.

Real-Time Pipeline: NATS + Protobuf, 10K pts/sec, <100ms p99

The data pipeline is a 3-stage NATS-backed architecture with Protobuf serialization:

```
raw.{device_id}.{tag_name}
  → Validator → validated.{device_id}.{tag_name}
    → Enricher → enriched.{device_id}.{tag_name}
      → Aggregator → aggregated.{window}.{tag_id}
```

Each TelemetryPoint on the wire carries: UTC timestamp, fully-qualified tag ID, value, pre-scaling raw value, quality flag (good/uncertain/bad), engineering unit, data type, source device ID, and a monotonic sequence number. Protobuf encoding compresses to approximately 200 bytes per point — efficient for bandwidth-constrained OT networks.

The aggregator produces 1-minute, 5-minute, and 1-hour windows (configurable). The Recorder subscribes to enriched.> and flushes to DuckDB at a configurable batch interval. Connection buffering uses a 3-tier strategy: ring buffer for hot data, write-ahead log (WAL) for durability during connector outages, and a configurable drop policy when the WAL reaches capacity. Backpressure propagates as exponential backoff rather than silent drops.

Visual Workflow Builder: React Flow DAG Editor

The workflow builder is a drag-and-drop canvas where process engineers configure monitoring logic without writing code:

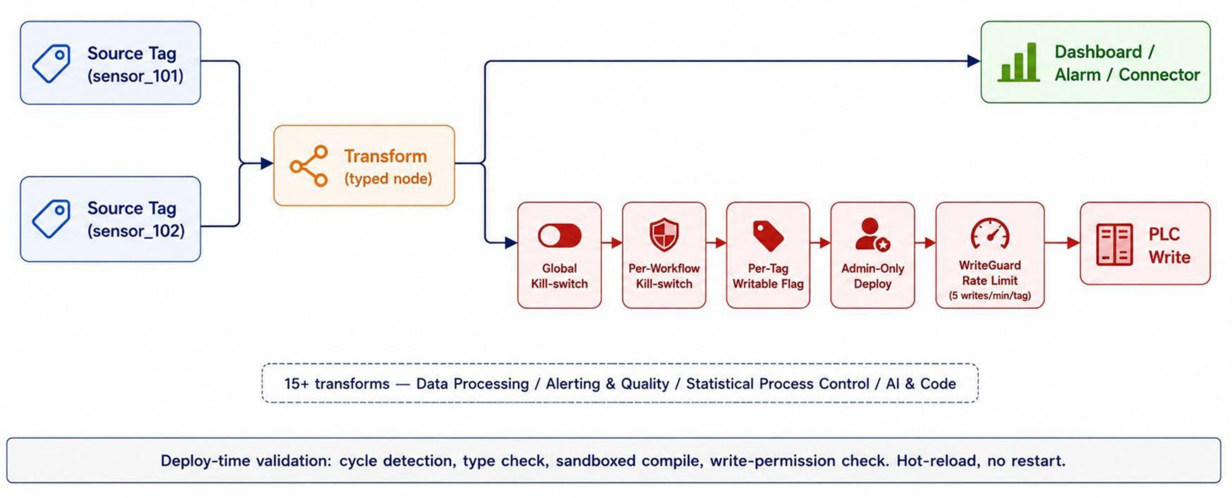

Figure 6: Visual workflow builder — source tags connect through typed transform nodes to destinations, with PLC write-back behind 5 independent safety gates.

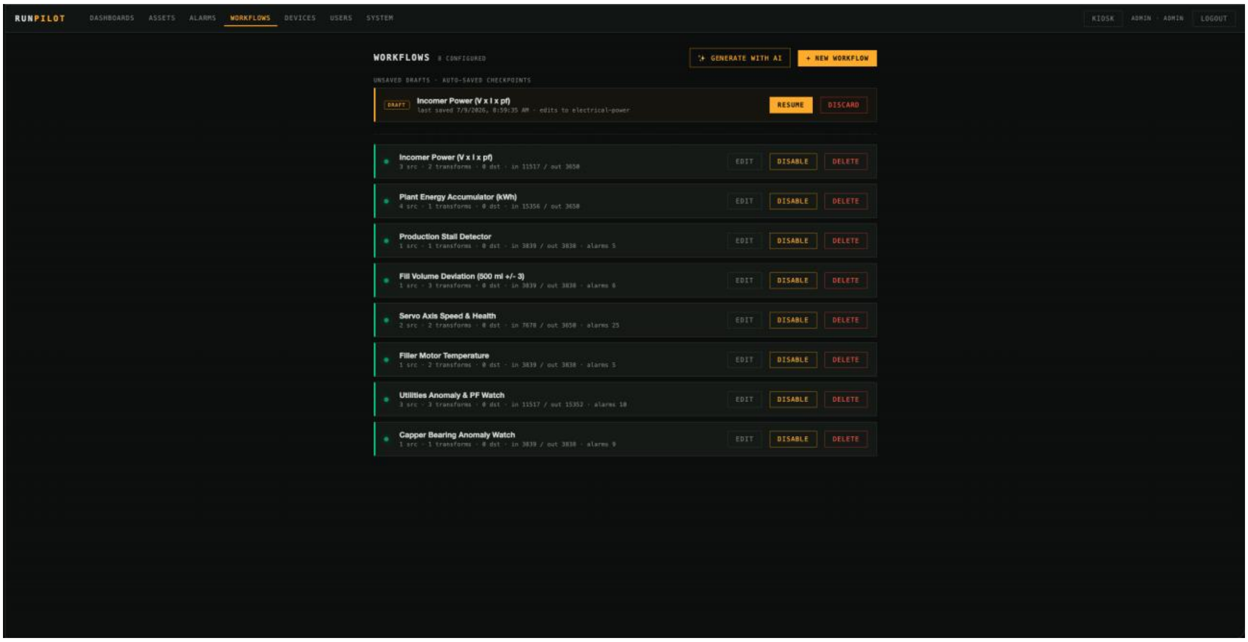

Figure 7: RunPilot Workflows list — eight production workflows including Incomer Power, Plant Energy Accumulator, Production Start Detector, and Capper Bearing Anomaly Watch, with status, run counts, and enable controls.

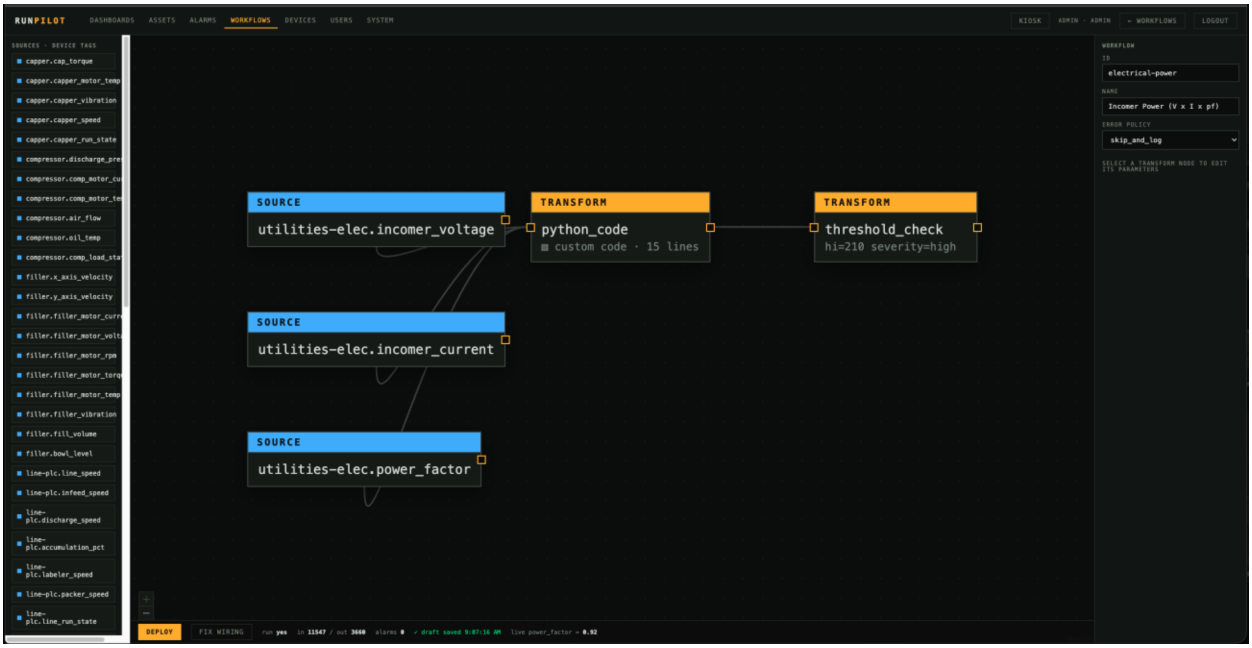

Figure 8: RunPilot Workflow DAG editor — the Incomer Power workflow showing source nodes (voltage, current, power factor) connected through Python transform and threshold check nodes, with the full device tag tree on the left.

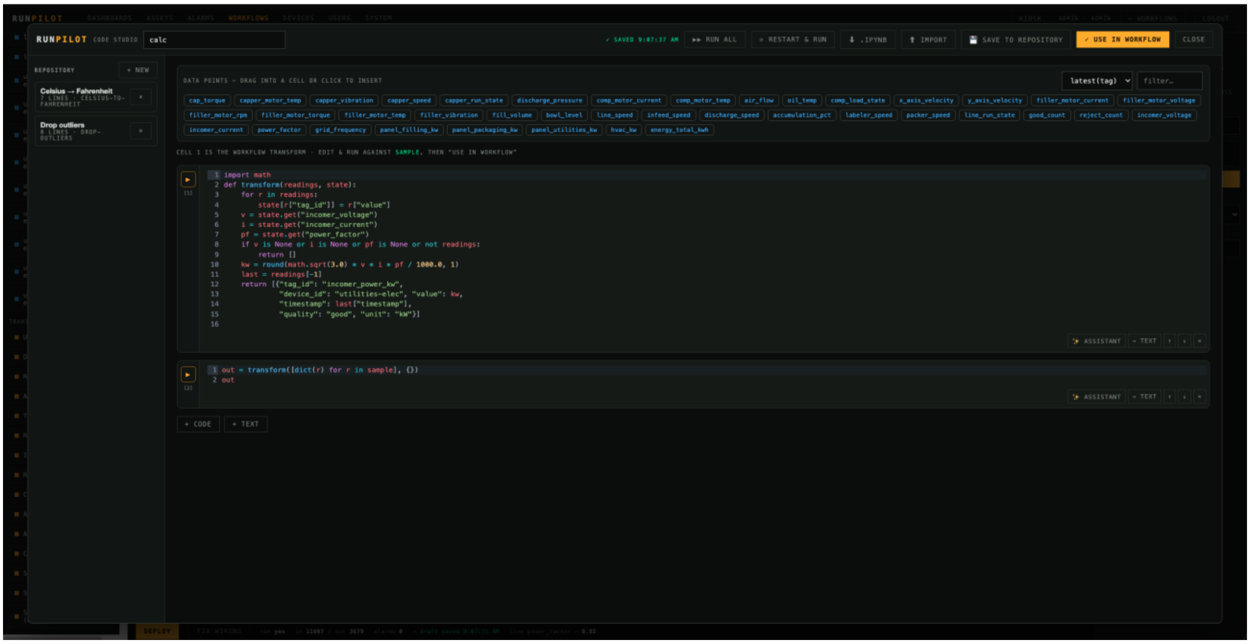

Figure 9: RunPilot Data Studio — the Python transform code editor with syntax highlighting, showing custom data processing logic for the Incomer Power workflow.

The 15+ transform types span four categories:

Data Processing: unit_convert, deadband_filter, rolling_average, aggregate, interpolate, rename, expression (safe Python via simpleeval with where-filter support)

Alerting & Quality: threshold_check, rate_of_change, anomaly_detect (Z-score + IQR statistical baseline, or IsolationForest per tag with 100-sample warmup)

Statistical Process Control: spc_xbar_r (Shewhart X-bar/R control charts), spc_ewma (EWMA drift detection), spc_cpk (rolling Cp/Cpk capability index — raises an alarm when Cpk falls below 1.33, the standard process capability threshold)

AI & Code: ai_analyze (cooldown-gated LLM analysis of batch statistics, optionally raised as an info alarm), python_code (full custom transform(readings, state) logic in a RestrictedPython sandbox with import whitelist and wall-clock timeout), run_workflow (workflow composition up to 8 nesting levels)

AI assistance is embedded at the authoring layer: **"Write with AI"** generates the transform body from a plain-English description; **"Generate with AI"** drafts an entire validated workflow graph from a sentence ("alert me when filler temperature exceeds 85°C for more than 5 minutes"). The operator reviews and deploys.

Every deploy validates the DAG: cycle detection via Kahn's algorithm, unknown transform type checking, input reference resolution, RestrictedPython compilation for code nodes, and PLC write permission verification. Hot-reload applies the new workflow without process restart.

On-Edge AI: LLM Provider Chain, RAG, Tool-Calling Agent

RunPilot's AI system is architecturally a domain-specific instance of Lumi's own intelligence layer, compressed to run on €200 hardware:

RUNPILOT AI COMPONENT	LUMI ARCHITECTURAL ANALOG
LLM fallback chain (Ollama → Claude → rule-based)	Provider-agnostic engine with ordered fallback
LanceDB vector store, all-MiniLM-L6-v2 embeddings	Local-first memory with bge-small embeddings
Tool-calling agent (timeseries, alarms, knowledge search)	Multi-agent orchestration with tool dispatch
Knowledge directory ingested at boot	Four-tier memory pipeline (ingest → index → retrieve)
Skills as markdown playbooks in knowledge/skills/	Persona system with domain-specific capabilities
Proactive advisories (anomaly → info alarm)	Conductor's proactive delivery mode

The LLM chain supports four API styles: anthropic (Anthropic, MiniMax), openai (OpenAI, Azure AI Foundry, Groq, DeepSeek, vLLM, LM Studio), azure (Azure OpenAI deployments), and ollama (fully local models). The fallback chain is user-configured: a typical edge deployment runs a local phi3:mini on-device for offline operation, falling back to Claude or GPT when internet is available.

The tool-calling agent (POST /api/ai/chat, SSE-streamed) has access to: time-series queries over DuckDB, the ISA-95 asset tree, the alarm suite, production and OEE summaries, anomaly statistics, knowledge base RAG search, workflow status — and confirm-gated mutations (alarm acknowledgement, PLC write-through WriteGuard). The agent never acts without the operator's explicit confirmation on any mutation.

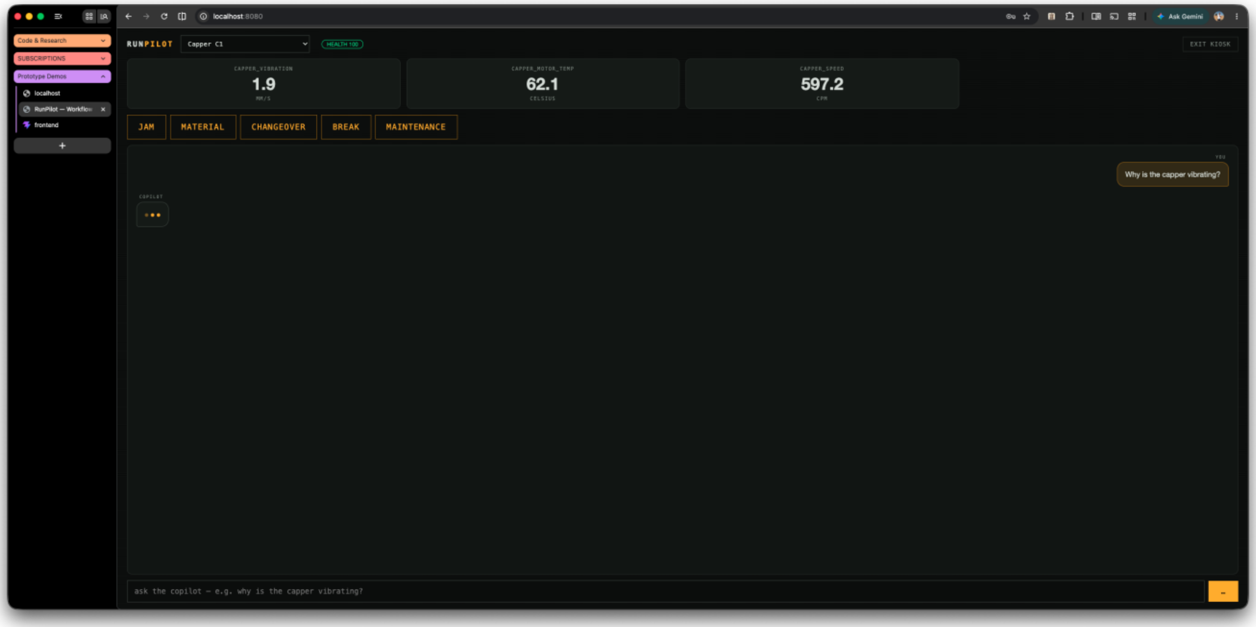

Figure 10: RunPilot Kiosk mode — the Capper C1 operator view with live sensor cards (vibration, motor temperature, speed), downtime reason buttons (Jam, Material, Changeover, Break, Maintenance), and the Copilot chat interface where an operator asks "Why is the capper vibrating?"

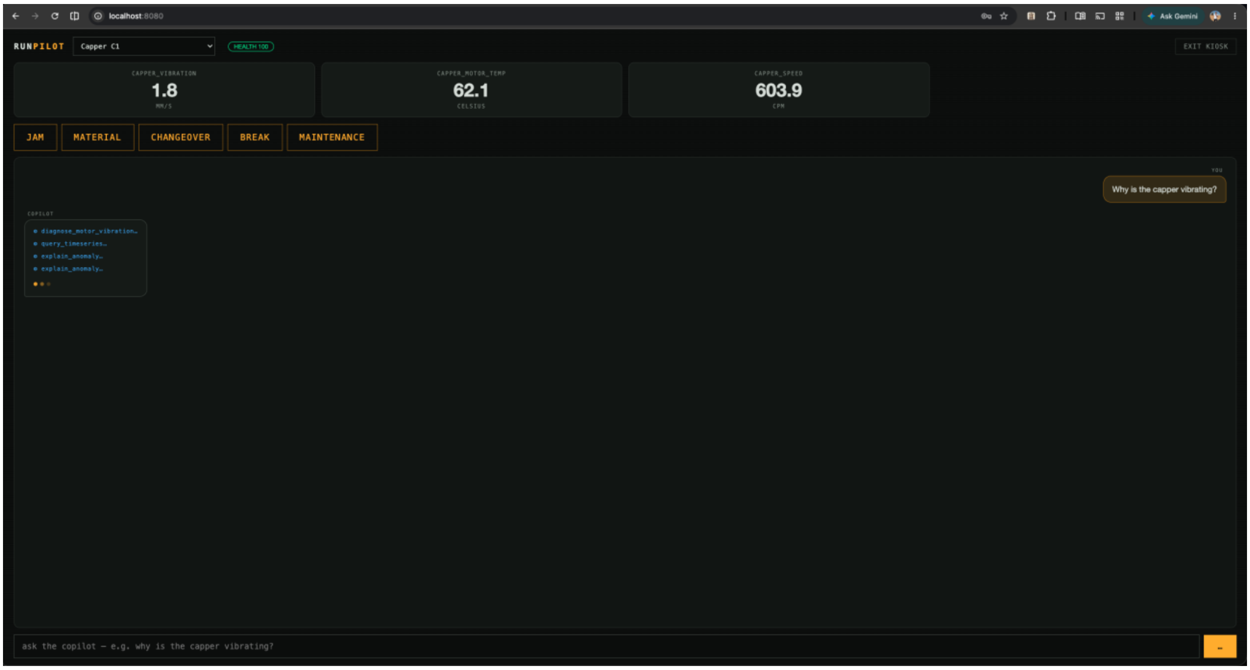

Figure 11: RunPilot Copilot responding — the AI agent invokes diagnostic tools (diagnose_motor_vibration, query_timeseries, explain_anomaly) to investigate the operator's question in real time.

Four operational skills ship as markdown playbooks: vibration triage, OEE analysis (the same skill documented in knowledge/skills/oee-analysis.md), Modbus commissioning, and alarm-flood triage. These are the manufacturing equivalent of Lumi's persona system — domain expertise packaged as retrievable, applicable knowledge.

OEE Framework: 6 KPI Insight Applications

The KPI framework binds device tags to semantic classes (run_state, good_count, total_count, actual_rate, power_kw, vibration_rms) and computes six insight applications at the edge:

OEE (Overall Equipment Effectiveness): Availability × Performance × Quality, computed per equipment and rolled up to line level. Availability = run time / planned time. Performance = actual rate vs. ideal rate per minute (granularity matters for detecting micro-stops that per-hour measurement conceals). Quality = good units / total units. World-class benchmark: 85%. Global manufacturing average: 55—60%. The platform's OEE computation targets the specification from RunPilot's Line 1 operations guide: ≥90% Availability, ≥95% Performance, ≥98.5% Quality, yielding ≥84% OEE overall.

Energy Baselines: kWh accumulation from three-phase power (V × I × power factor), with overspend alarms when consumption exceeds the configured baseline. Supports tariff-aware energy cost tracking.

Predictive Maintenance: IsolationForest on-edge (when sklearn is available) with statistical fallback always present, producing 0—100 health scores per asset. Health alarms trigger when scores fall below configurable thresholds.

SPC (Statistical Process Control): X-bar/R charts, EWMA drift detection, and Cp/Cpk capability indices. Raises alarms when Cpk falls below 1.33 — the standard manufacturing process capability threshold.

Anomaly Feed: Per-tag Z-score and IQR detection with configurable thresholds, producing a categorized alarm feed (/api/alarms?category=anomaly) with optional LLM "why" narration.

Digital Twin: Per-asset live mirror with bound values, KPIs, health badge, active alarms, and sparklines. AI contextualization proposes the ISA-95 hierarchy and tag-class bindings from a raw tag list with one click to apply.

Security Architecture

Security is a first-class architectural concern, not an afterthought — reflecting both the adversarial security audit's findings and the particular severity of industrial security failures:

Authentication & Session Management: bcrypt password hashing, secrets.token_urlsafe(32) session tokens (32 bytes of entropy), httponly session cookies, brute-force protection (5 attempts → 5-minute lockout by IP+username).

RBAC — 4 Inherited Roles:

- viewer: view_dashboard, view_data
- operator: Above + acknowledge_alarm
- engineer: Above + modify_config, manage_workflows, manage_knowledge
- admin: All permissions including manage_users, plc_write, security_settings

PLC Write Defense in Depth: Five independent gates on every physical actuator command:

- Global security.plc_write_enabled kill-switch (off by default)
- Per-workflow plc_write_from_workflows kill-switch
- Per-tag writable: true configuration flag
- Deploying user must hold the plc_write permission (admin only)
- Every write routed through WriteGuard with configurable rate limit (5 writes/minute/tag)

Network Isolation: 4-zone Docker network architecture — plc-net (internal bridge, no internet, for OT traffic), internal (RunPilot ↔ NATS), frontend (operator browser access), cloud (sync-agent outbound only).

Audit Log: Append-only SQLite log of every security-relevant action: login success/failure with IP, device/connector/workflow mutations with before/after snapshots, alarm acknowledgements, PLC writes, config imports/exports, backups.

11 Output Connectors

RunPilot's connector layer supports 11 output targets behind a shared store-and-forward buffer (ring buffer + WAL):

CATEGORY	CONNECTORS
Messaging	MQTT (incl. AWS IoT Core via mTLS), Kafka (Confluent/MSK/Redpanda/Azure Event Hubs Kafka endpoint), Azure Event Hubs (native AMQP), AWS Kinesis
Cloud Storage	S3/MinIO/R2 (gzipped JSONL batches), Azure IoT Hub (SAS auth), GCP Pub/Sub
Time-Series DBs	TimescaleDB/PostgreSQL, InfluxDB v2
IIoT Standards	Sparkplug B (NBIRTH/NDATA with aliasing, minimal Protobuf), HTTP webhooks

All connectors are managed from the browser — no integrator engagement required to change output destinations.

5. The Cost Equation

Traditional Approach: 7 Engineers × 6—18 Months

Building RunPilot's feature set through conventional means requires seven specialized roles:

ROLE	DURATION	ANNUAL LOADED COST	PROJECT TOTAL
SCADA/Automation Engineer	12—18 months	$150K-$200K	$150K-$300K
Streaming/Data Architect	6—12 months	$180K-$220K	$90K-$220K
Backend Engineer	12—18 months	$150K-$200K	$150K-$300K
AI/ML Engineerning	6—12 months	$180K-$250K	$90K-$250K
Frontend Developer	6—9 months	$130K-$170K	$65K-$128K
Security Consultant	2—4 months	$200K-$300K	$33K-$100K
Edge/DevOps Engineer	3—6 months	$160K-$200K	$40K-$100K
Development subtotal			$618K-$1.4M
Domain consultant engagement	2—4 months	-	$300K-$500K
Total	6—18 months		$918K-$1.9M

The domain consultant engagement is the hidden cost that most technology estimates omit. Building a SCADA platform without manufacturing domain knowledge produces software that connects to PLCs but doesn't understand how a factory operates. The consultant extracts tribal knowledge, validates requirements against operational reality, and ensures the platform addresses problems plant teams actually face — alarm floods during shift changes, OEE inflation from manual measurement (8—12 percentage points above automated values), micro-stop detection that per-hour measurement misses.

Lumi Approach: Knowledge Ingestion + One Overnight Session

The Lumi approach redefines the cost structure:

COMPONENTS	INVESTMENT
Domain knowledge ingestion	Months of prior CES work — protocol analysis, operational pattern capture, security research, market analysis — already operational in Lumi's memory system
Multi-agent orchestration	One overnight session — 7 specialist agents, 10 stories, parallel execution
Security audit	Included — adversarial reviewer ran post-construction, produced 82 findings + phased remediation plan
Output	110+ source files, 40+ tests, 8 protocols, React SPA, Docker deployment, knowledge base, security roadmap

The knowledge investment is non-zero — it represents months of prior work. But unlike consultant knowledge, it is **reusable at zero marginal cost**. The same manufacturing domain expertise that built RunPilot can build the next platform, inform the next factory engagement, or onboard the next engineer. The traditional consultant's knowledge walks out the door with the consultant.

RunPilot's Own Cost Disruption

RunPilot itself embodies the cost disruption Lumi used to build it:

DIMENSIONS	RUNPILOT	INCUMBENT PLATFORMS
Hardware	€200 Raspberry Pi 5 or industrial PC	Dedicated industrial gateways ($2K—$10K)
Licensing	MIT License — zero per-device fees	$50K—$150K per site (Litmus Edge); $3K—$35K per server (Ignition)
Configuration	Self-service browser UI — no integrator required	Professional services engagement, $10K—$50K
Protocol breadth	8 drivers out of the box	Protocol modules often sold separately
AI assistant	Included — any LLM vendor or fully offline	Not available on any incumbent platform as of mid-2026
Annual cost/asset	Near zero (hardware amortized)	~$12,167/asset/year (Litmus Edge all-in)

ROI: The OEE Math

The financial case for RunPilot deployment is anchored in OEE improvement. The research evidence is consistent: a 10-point improvement in measured OEE on a single production line represents **$300,000 to $2,000,000 in annual recoverable production value.**

For a high-speed bottling line running at 600 bottles/minute with a nominal OEE of 70% — a realistic starting point given that the global manufacturing average is 55—60% and manual measurement inflates self-reported OEE by 8—18 percentage points — the math is direct:

- A 4-point OEE improvement (from 70% to 74%) on a line producing $15M/year at full capacity recovers approximately **$600K in annual production value**
- RunPilot hardware cost: **€200**
- RunPilot licensing: **€0**
- Deployment time: **one afternoon** from box to browser

The payback period on hardware alone is measured in hours of recovered production. Even accounting for engineering time to configure the platform and tune the workflows, the ROI timeline is days, not the 12—24 months typical for traditional SCADA/MES implementations.

The OEE improvement mechanism is also direct: RunPilot's automatic data capture eliminates the 8—12 percentage point inflation that manual tracking introduces. When a manufacturer first switches to automated measurement, measured OEE often appears to drop by 15—20 points — not because production worsened, but because measurement became accurate. Accurate measurement is the prerequisite for genuine improvement. RunPilot provides it from day one.

6. AI as Augmentation — The Manufacturing Workforce

The Fear Is Real

The workforce anxiety around AI in manufacturing is not irrational. Worker fears are grounded in documented evidence: 40% of workers worldwide now fear losing their jobs to AI — a 12-percentage-point increase from 28% in 2024 (Mercer, n=12,000). Companies attributed 55,000 job cuts directly to AI in 2025 alone — a 12× increase from 2023. The dockworkers' strike of October 2024, the Hollywood writers' strike of 2023, the UAW action of 2023 — each was shaped, in part, by automation anxiety.

In manufacturing specifically, the fear takes a concrete form: if AI can monitor the machines, analyze the data, and tell operators what to do — what is the operator's role? If AI can draft shift handover reports and explain anomalies, what does the veteran bring that the newcomer doesn't? If AI can configure workflows from a sentence, why train process engineers at all?

These are serious questions that deserve serious answers — not reassurance. RunPilot's design is itself an answer.

RunPilot Embeds the Augmentation Thesis

The Agentic SDLC whitepaper's core claim — that AI's role is augmentation, not replacement, changing what humans do rather than whether humans are needed — is operationalized directly in RunPilot's architecture:

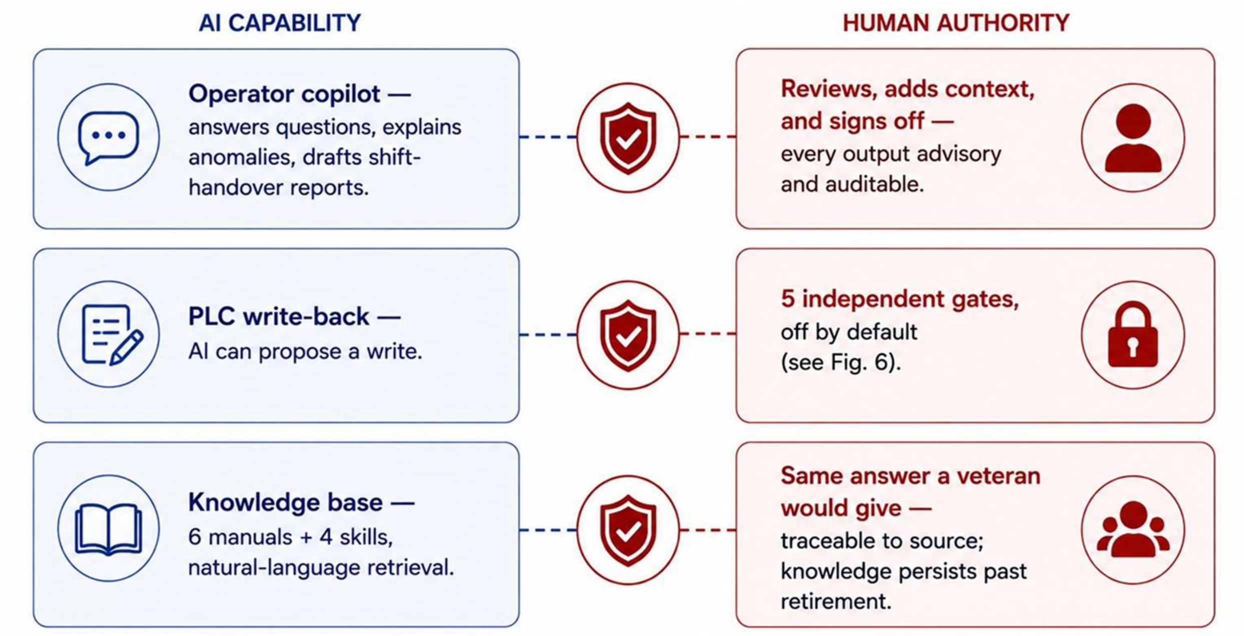

Figure 5: RunPilot's augmentation architecture — AI capabilities paired with mandatory human authority at every consequential decision point.

Operator copilot, not operator replacement. The AI assistant answers questions ("what's the capper vibration limit?"), explains anomalies ("why did energy spike at 14:30?"), and drafts shift-handover reports aggregating alarms, production counts, downtime events, and KPI summaries from the past shift. Every output is advisory — labeled, explainable, and auditable. The AI compresses 8 hours of operational data into a readable briefing. The human ensures it is accurate and adds context that data alone cannot capture.

Human-confirm gates on all critical actions. PLC write-back — the action that can physically affect equipment — is off by default and requires five independent conditions to be satisfied simultaneously. This is defense in depth for the physical world. A single misconfigured write to an industrial controller can damage a $50,000 servo motor or cause a safety incident. RunPilot's design treats PLC writes the way aviation treats control inputs: multiple independent checks, every action recorded, no unverified autonomous operation.

Knowledge democratization, not knowledge replacement. The 6 plant manuals and 4 operational skills in RunPilot's knowledge base represent tribal knowledge made accessible to everyone — not just the veteran. When the maintenance engineer with 20 years of compressor experience retires, the knowledge of operating envelopes, warning signs, and triage procedures doesn't leave with them. It persists in the knowledge base, retrievable through natural language by any operator at any hour, in any language the LLM supports.

The new hire who asks "what vibration level means the TC-200 capper needs attention?" gets the same answer the veteran would have given — immediately, with context, and traceable to the source documentation.

Factory Workers as Digital Transformation Agents

The traditional model of factory digitalization places IT departments or external integrators in control. Plant operators and process engineers — the people who understand how the factory actually works — are consulted but not empowered to build. Digital transformation becomes something that happens to a factory.

RunPilot inverts this model deliberately:

The **visual workflow builder** lets a process engineer drag sensor tags to transform nodes to alarm thresholds, deploy, and iterate — without writing code, without filing a ticket with IT, without waiting for an integrator's next available slot. The person who understands the process builds the monitoring logic for that process. Domain knowledge and digital tooling are no longer in separate organizations.

AI-assisted authoring lets an operator describe a workflow in a sentence and get a validated graph on the canvas for review. The AI handles the syntax; the human provides the domain knowledge. A mechanic who wired a new motor can describe the monitoring logic she wants in natural language and see it on the canvas within seconds.

Self-service device management — forms, test-connection validation, tag browsing, CSV import — means adding a new PLC or a new sensor requires no YAML editing, no integrator engagement, and no IT ticket. The electrician who wired the motor can configure its monitoring in the same afternoon.

This is Industry 5.0's human-centric principle made operational: technology that amplifies the people closest to the process, rather than technology that requires intermediaries between the process and its digital representation.

Addressing the Fear Directly

Worker anxiety about AI does not dissolve in the face of good design. It requires transparent demonstration.

The IPC research from MIT (2024, n=9,000 workers across nine countries) identifies the conditions under which workers view new technology as beneficial: when jobs involve complex problem-solving, and when the technology supports rather than monitors or replaces. RunPilot is designed to meet both conditions: it augments the operator's problem-solving capability (by providing context, anomaly detection, and knowledge access) rather than monitoring the operator's behavior.

The Industry 5.0 framework — championed by the EU Commission as a corrective to Industry 4.0's "lights-out factory" narrative — names three first-class outcomes: human-centricity, resilience, and sustainability. RunPilot's architecture addresses all three: operators remain in authority over consequential decisions (human-centricity), the platform degrades gracefully when connectivity is lost (resilience), and energy monitoring with tariff-aware kWh tracking creates an operational path to sustainability reporting.

The evidence that AI augmentation improves worker satisfaction is real: 72% of manufacturers expect AI to augment workers rather than replace them (McKinsey), and pilot implementations of Industry 5.0 approaches report increased worker satisfaction when employees feel technology supports rather than supplants them. The mechanism is not mysterious: when automation handles the tedious and the dangerous, humans are left with the interesting and the consequential.

7. Lessons Learned & Patterns

Pattern 1: Knowledge Cloning Is Reproducible

The most important lesson from the RunPilot overnight build is that it was not a one-off. The conditions that made it possible — accumulated domain knowledge in Lumi's memory, specialist agent personas with relevant expertise, the Conductor's tests-until-green execution model — are stable and reproducible.

The overnight build was not primarily a story about code generation speed. It was a story about the ratio of knowledge availability to build time. When domain knowledge is operational (retrievable, composable, applicable without human mediation), the bottleneck shifts from knowledge acquisition to execution — and execution is where multi-agent orchestration has structural advantages over human teams.

The pattern generalizes: any domain where expertise can be accumulated and made operational through Lumi's memory pipeline can be approached with the same architecture. Manufacturing was the test case. The pattern is domain-general.

Pattern 2: Decomposition Quality Determines Outcome Quality

The 10-story decomposition of RunPilot was not merely organizational — it was architectural. The separation between the driver layer and the pipeline, between the asset model and the telemetry store, between security and business logic, reflects design decisions that would be made by an experienced architect. An inexperienced decomposition would have produced a different build — one that might have passed initial tests but would have been harder to extend, maintain, and secure.

The implication: the quality of multi-agent orchestration is determined not just by the agents' capabilities but by the quality of task decomposition. Lumi's team lead agent produced a high-quality decomposition because it had access to the same manufacturing domain knowledge as the specialist agents — and because that knowledge included architectural patterns for SCADA systems specifically.

Pattern 3: Adversarial Review Catches What Single Developers Miss

Eighty-two findings across 65 files — including 5 CRITICAL vulnerabilities — from a security audit of code that was functionally correct and passing all tests.

This is not a failure of the building agents. It is a demonstration of why adversarial review is structurally necessary. The building agents were incentivized to make code work. The security auditor was incentivized to make code fail. These are different cognitive orientations, and both are necessary. No single developer — regardless of security expertise — maintains both orientations simultaneously across 65 files.

The RPL adversarial architecture makes this structural: the reviewing agent is architecturally independent from the building agents, explicitly mandated to challenge and probe, and operating without the sunk-cost attachment that causes reviewers who also built the code to defend their decisions rather than question them.

Pattern 4: Edge-First AI Is Practical on Commodity Hardware

The conventional assumption in industrial AI is that meaningful inference requires cloud connectivity or dedicated AI hardware. RunPilot challenges this assumption directly: a Raspberry Pi 5 running Ollama with a phi3:mini model can provide meaningful AI assistance to an operator — answering questions about plant documentation, explaining anomalies in context, drafting shift reports — with no cloud dependency and no external data transmission.

This matters for three reasons. Industrial OT networks are often airgapped or bandwidth-constrained by design — cloud-first AI architectures fail in exactly the environments where AI assistance is most valuable. Data sovereignty concerns in manufacturing (proprietary process parameters, production volumes, quality data) make cloud transmission undesirable even when bandwidth is available. And the cost of cloud inference at operator-interaction rates adds up to a per-facility subscription cost that undermines the economics of the platform.

The edge-first, cloud-optional architecture makes the AI capability genuinely deployable at scale.

Pattern 5: Self-Service Beats Vendor Dependency

The operational model of every incumbent SCADA platform assumes professional services involvement for non-trivial changes. This assumption creates a structural dependency that prevents the plant team from iterating at the speed of insight.

RunPilot's self-service architecture — browser-managed devices, connectors, and workflows; AI-assisted authoring; hot-applied changes without restarts — is not a feature. It is the product. The value is not in the platform's capabilities in isolation; it is in the fact that the people closest to the process can configure and iterate those capabilities without external mediation.

8. The Road Ahead

RunPilot's Roadmap: Three Horizons

RunPilot's development follows a three-horizon roadmap that reflects the progression from data collection to operational intelligence to human-centric optimization:

Horizon 1 (Shipped): The first-step accelerator is complete. Tags flowing within 15 minutes of connecting an edge box. Visual workflows and alarms within 30 minutes. Cloud data streaming within 60 minutes. AI assistant available to operators from day one. All six KPI insight apps operational. Self-service fleet management (store-and-forward transport shipped; central console pending second product surface).

Horizon 2 (In Progress — months): Fleet management console for multi-site enrollment and signed config/workflow bundle distribution. Predictive maintenance model registry with remaining-useful-life (RUL) estimation. Report pack with PDF generation and scheduling. Operator andon and shift log (one-tap downtime reason logging, AI-drafted shift handovers).

Horizon 3 (Quarters): Three capabilities that represent the human-centric differentiators of Industry 5.0:

Operator copilot at the line — kiosk mode (fullscreen per-asset live tiles + andon + AI chat) is shipped; voice input, multilingual support, and dedicated hardware packaging remain.

Energy & sustainability module — kWh-per-unit tracking, idle-load detection, CO_2 reporting. The energy story is often the fastest ROI in a smart-factory business case, and it is the capability most directly aligned with Industry 5.0's sustainability mandate.

Closed-loop optimization with guardrails — AI-proposed setpoint changes that always route through the confirm gate and WriteGuard, building operational trust before increasing autonomy. This is the Industry 5.0 model for AI-human collaboration applied to the physical plant: the AI proposes, the human confirms, the plant adapts.

Lumi's Expanding Manufacturing Toolkit

RunPilot is the first manufacturing-specific output of Lumi's Agentic SDLC platform — but the pattern is extensible. The same knowledge cloning mechanism that built RunPilot's SCADA platform can be applied to:

- **MES integration layers** — connecting RunPilot's edge data to existing manufacturing execution systems without vendor lock-in
- **Energy management platforms** — dedicated tools for tariff optimization, demand response, and carbon accounting
- **Quality management systems** — SPC and inspection data capture with AI-assisted root cause analysis

- **Maintenance platforms** — work order management, spare parts optimization, and predictive maintenance model management

The knowledge investment compounds. Every manufacturing engagement makes Lumi's memory system more capable of the next one. Every plant connected to RunPilot generates operational knowledge — equipment behavior, alarm patterns, OEE drivers — that feeds back into the knowledge graph and makes the next deployment more capable from day one.

The Vision: Smart Factory Acceleration, Cost No Longer a Barrier

The thesis of this case study is ultimately simple: the cost and complexity barriers that have kept smart factory capabilities out of reach for the majority of manufacturers are not technological. They are organizational — the result of a market structure that treats domain knowledge as a billable professional service and configuration as a specialist skill.

When domain knowledge is operational in an AI platform's memory, and when configuration is accessible through self-service tooling and natural language authoring, the barriers collapse. A €200 edge box becomes a smart factory accelerator. A process engineer becomes a digital transformation agent. A plant that could not justify a $1M SCADA implementation can deploy RunPilot in an afternoon and begin recovering OEE within the same week.

The 3—6% of manufacturers who currently achieve world-class OEE (85%+) are not there because they have better equipment. They are there because they have better visibility, better measurement, and faster feedback loops. RunPilot makes that visibility, measurement, and feedback accessible to the other 94%.

That is the vision: not technology as the domain of specialists, but technology as the amplifier of the people who understand their process. Industry 5.0 as an operational reality, not a conference theme.

About the Authors

Ankur Nair is Co-Creator of Lumi at CES. His work focuses on agentic systems architecture, distributed computing, and the design of AI platforms that operate as organizational infrastructure rather than point-solution tools. He brings to Lumi a perspective shaped by the practical demands of building and operating software at scale — the failures and costs that motivate the architectural decisions documented in this whitepaper.

Anurag Sharma is Co-Creator of Lumi at CES. His work spans the strategic and technical dimensions of AI adoption in software organizations — the governance, organizational design, and value realization frameworks that determine whether AI investments compound or fragment. He brings to Lumi a perspective shaped by the organizational challenges of the Agentic SDLC transition: the human roles that evolve, the governance infrastructure that enables trust, and the measurement frameworks that distinguish genuine productivity gains from self-reported optimism.

Together, Ankur and Anurag built Lumi not as a product roadmap exercise but as the infrastructure CES uses to deliver software — and, as RunPilot demonstrates, to build the platforms that serve its clients. RunPilot is proof that the Agentic SDLC described in CES's June 2026 whitepaper is not theoretical. It is operational, repeatable, and extending into every domain where accumulated expertise and parallel execution can change what is possible.

References & Sources

Primary Sources — RunPilot

- RunPilot README — runPilot/README.md
- RunPilot ROADMAP — runPilot/docs/ROADMAP.md
- RunPilot User Manual — runPilot/docs/USER_MANUAL.md
- RunPilot Knowledge Base: Line 1 Operations — runPilot/knowledge/plant/line-1-operations.md
- RunPilot Knowledge Base: OEE Analysis Skill — runPilot/knowledge/skills/oee-analysis.md

CES Internal Analysis

- CES Agentic SDLC Whitepaper (June 2026) — The Agentic SDLC: How AI is Transforming Software Delivery
- CES Internal Research: RunPilot Overnight Build Strategy Analysis (July 2026)
- CES Internal Research: Smart Factory Market & OEE Benchmarks (July 2026)

Smart Factory Market Research

- DPS Telecom — SCADA System Prices and Maintenance Costs
- Savvycom — SCADA Software Price in 2025
- GE Vernova — SCADA: A Total Cost of Ownership Analysis
- Shoplogix — Total Cost of Ownership for MES
- OEE Tools — Rough Cost Estimation of MES Implementation (2025)
- Symestic — MES Software Vendors, Features & Costs Compared 2026
- iFactory — Siemens vs Rockwell vs Schneider for Greenfield
- Tatsoft — Top 8 SCADA Platforms Compared
- Inductive Automation — Ignition Pricing
- MachineCDN — Litmus Pricing
- HighByte — HighByte Intelligence Hub Pricing
- N3uron — Pricing Modules EU

OEE Benchmarks

- Evocon — World-Class OEE: Industry Benchmarks From 50+ Countries (2023—2024)
- Fabrico — OEE Benchmark: 250-Plant European Study (2025)
- Godlan — OEE Benchmarks by Industry Vertical: 2025 Data
- Symestic — OEE Benchmarks: Realistic Values by Industry 2026
- TeepTrak — How to Calculate Real OEE (Not the One from Excel)
- Matics — OEE Value: Financial Impact
- Blue Mountain — Understanding the $ Value of OEE Improvements

Industry 4.0 / 5.0 Adoption

- McKinsey — Industry 4.0: Capturing Value at Scale in Discrete Manufacturing
- Parsec/Wakefield Research — Digital Transformation Survey (n=1,200, 2024)
- LNS Research — Industrial Transformation: What Are SMBs to Do?
- Aptean — The Manufacturing Digital Transformation Disconnect
- BusinessWire — Survey: Digital Transformation Brings Industry 4.0 Within Reach (2024)
- Deloitte — 2025 Smart Manufacturing Survey
- Manufacturing Leadership Council — Survey: M4.0 Appears Poised for A Significant Leap
- MarketsandMarkets — Industrial Edge Market Report
- Statifacts — Edge Computing in Manufacturing Market
- Automation.com — Edge Computing Adoption in IIoT

Worker Sentiment & Industry 5.0

- Mercer — Global Workforce Hopes and Fears Survey (n=12,000, 2026)
- PwC — Global Workforce Hopes and Fears Survey 2025
- Challenger, Gray & Christmas — AI-Attributed Job Cuts 2025
- WEF — Future of Jobs Report 2025
- MIT IPC — Automation from the Worker's Perspective (n=9,000, 2024)
- Taylor & Francis — Human-Centred AI in Industry 5.0: A Systematic Review (2024)
- Taylor & Francis — Human Centric Innovation at the Heart of Industry 5.0 (2025)
- McKinsey — AI Augmentation in Manufacturing Survey

- Gartner — Smart Factory and Digital Transformation (2024)
- Forbes — Why The Dock Workers' Strike Should Be A Cautionary Tale For Every Manufacturer (2024)

AI Productivity Research (referenced from CES Agentic SDLC Whitepaper)

- METR — Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity (arXiv 2507.09089)
- Anthropic — 2026 Agentic Coding Trends Report
- Gartner — 40% of Enterprise Apps Will Feature AI Agents by 2026
- Forrester — Agentic Software Development Takes the Lead (June 2026)
- PwC — Agentic SDLC Research (n=377 tech leaders)
- Larridin — Developer Productivity Benchmarks 2026

This whitepaper was compiled by Lumi, CES's AI engine, and signed off by its Co-Creators.

Anurag Sharma & Ankur Nair

CES Global LLC — *[www.cesltd.com]*