Move the Question, Not the Data

A distributed manufacturing intelligence architecture for sovereign, evidence-based process improvement

Process graphs • Dual-loop edge SLMs • Deming-inspired SPC • Automated UNS • Root-cause investigation • Financial ROI

CogniaWerks Co. | Technical white paper | November 2026

Status: architectural description and proposed implementation framework. “provisional provisional patent-pending” is used as supplied by CogniaWerks; no patent application number or claim scope is represented here. Product performance, deployment maturity and savings require customer-specific validation.

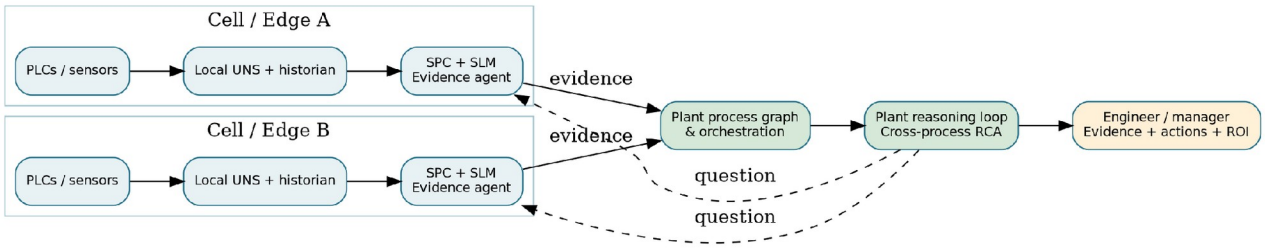


Figure 1. Local data stays in each edge domain; questions and compact evidence traverse the plant intelligence fabric.

Executive summary

Factories generate a vast volume of telemetry but often struggle to turn it into timely, explainable decisions. Centralizing every PLC tag, event and historian sample increases integration cost, bandwidth, latency and governance burden. CogniaWerks Distributed Manufacturing Intelligence (DMI) proposes the opposite operating model: send a targeted question to the relevant production data, execute analysis near the source, and return only the evidence needed for a decision.

A provisional patent-pending graph-oriented orchestration concept supplies the process context: which operation feeds another, which assets and variables are involved, and which data owners can answer a specific causal question. A first loop continuously detects variation at each edge node using statistical methods and constrained local small language models (SLMs). A second loop reasons across the process graph, requests follow-up evidence from relevant nodes, ranks root-cause hypotheses and delivers recommended actions with confidence and traceability. A financial layer then compares verified outcomes against a baseline and calculates attributable benefit.

1. The problem: data centralization is not process intelligence

A manufacturing event is rarely explained by a single alarm. A downstream dimensional defect may originate in upstream material characteristics, machine settings, thermal conditions or maintenance history. Raw tags do not encode those relationships. Traditional centralized analytics can answer some

questions, but can also create a costly data-integration project before the first production engineer receives a useful explanation.

The design objective is not “never transmit any byte.” It is to avoid routine movement of raw time-series and sensitive process data. Minimal metadata, query instructions, model/version information, evidence summaries and approved aggregates may move under explicit policy. Some investigations may require carefully authorized sample exports; that is an exception, not the default.

2. Design principles

- Plant sovereignty: retain raw operational records under plant and edge-node policy, including offline or air-gapped deployment.
- Process-first context: model production steps, material flow, dependencies, asset ownership and time alignment—not merely tag hierarchies.
- Evidence before explanation: deterministic calculations and retrieved observations precede SLM-generated narratives.
- Statistical discipline: avoid treating every fluctuation as a special-cause event.
- Human-governed action: separate recommendation from automatic control; require permissions, safety interlocks and verification.
- Measured value: claim ROI only after baseline definition, attribution and financial sign-off.

3. provisional patent-pending process graph and distributed query orchestration

The graph is the coordination layer linking sites, areas, lines, process steps, equipment, PLC endpoints, tags, batches, lots, operating modes, events, documents, analytical functions and people. Edges encode relationships such as FEEDS, PRECEDES, MEASURES, CONTROLS, CONSUMES, PRODUCES, LOCATED_AT and MAY_INFLUENCE. Each edge may carry temporal validity, direction, provenance and confidence. This is a proposed logical graph architecture; it does not assert a particular patented storage engine, novel claim or filing status beyond the supplied provisional patent-pending designation.

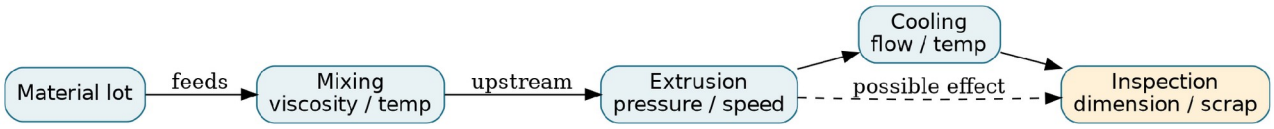


Figure 2. Example production graph: a quality excursion triggers targeted upstream investigation rather than a plant-wide data export.

3.1 Graph object and evidence contract

Object	Representative attributes	Purpose
Node	site_id, asset_id, process_id, node_id, owner, access policy	Locate computation and accountable data owner
Signal	tag_id, unit, sample rate, quality, timestamp, source	Interpret measurements consistently

Process edge	from/to, relationship, time lag, validity, provenance	Trace material and causal candidates
Evidence	event_id, time window, statistic, anomaly score, confidence, lineage	Support reproducible investigations
Model	model_id, version, prompt/tool policy, execution node	Audit AI-assisted reasoning
Action	hypothesis, recommendation, approval, expected impact, result	Close the improvement loop

3.2 Query planning and execution

An investigation begins with a natural-language or programmatic question: “Why did scrap rise on Line 2 during the second shift?” The orchestration layer resolves the affected process and time window, traverses upstream dependencies, identifies local analytical functions, sends scoped requests to authorized nodes, and assembles compact evidence. Every returned result includes source and computation lineage. Query planning should support clock-skew correction, unit normalization, late-arriving data, missing samples, sampling bias and permission failures.

Graph adjacency is not proof of causality. Candidate causes must be checked against chronology, mechanism, confounding variables, process operating state, intervention results and operator knowledge.

4. Dual-loop edge intelligence

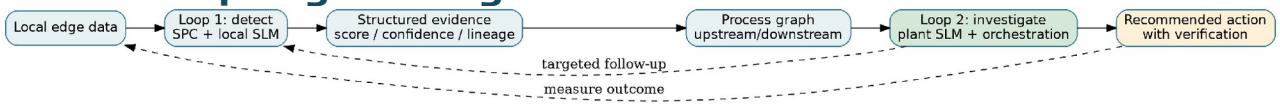


Figure 3. Dual-loop architecture: continuous local detection and selective cross-process investigation, with outcome feedback.

4.1 Loop 1 — local detection and explanation

Each edge node maintains authorized access to local PLC, historian and event data. Deterministic services compute control charts, run rules, change points, rates and contextual features. A local SLM interprets structured evidence, retrieves relevant maintenance or work-instruction context, suggests candidate explanations and generates concise, structured findings. The SLM is not responsible for inventing numerical SPC results: those are calculated by testable software functions.

A representative node response contains: node/asset/process identifiers; UTC and source timestamps; measurement units; window and sampling assumptions; control-limit method; observed rule violation; operating mode; anomaly score; confidence and uncertainty; top hypotheses; supporting and contradicting evidence; model/version; data lineage; and recommended next query.

4.2 Loop 2 — plant-level orchestration and RCA

The plant-level SLM receives structured local findings, uses the process graph to determine which upstream and downstream nodes can discriminate among hypotheses, and issues targeted follow-up questions. It compares evidence across shifts, lots and production stages; builds a ranked hypothesis set; identifies gaps; and requests further tests. Escalation to enterprise systems is optional and policy-controlled. Plant-level orchestration can run entirely on premises.

4.3 Guardrails, reliability and deployment

- SLMs use retrieval and typed analytical tools; free-form generated explanations must link to evidence IDs.
- Set latency, memory, CPU and token budgets per node; schedule inference to avoid interfering with control workloads.
- Fail closed on unavailable data, invalid units, stale timestamps, missing permissions or low confidence.
- Do not let a language model write directly to PLCs; require independent approval and controls engineering review.
- Version prompts, models, graph mappings, statistical rules and outcome measurements; support replay and audit.

4.4 Retrieval-Augmented Generation (RAG) — documentation-grounded investigations

DMI can use retrieval-augmented generation (RAG) to bring authorized technical documentation into the same evidence-based investigation as live PLC signals, SPC findings, event chronology and process-graph relationships. Relevant sources include equipment manuals, troubleshooting guides, standard operating procedures (SOPs), maintenance and calibration records, engineering specifications, quality procedures, deviation reports, work instructions and prior corrective-action records. Retrieval provides contextual evidence; it does not replace statistical analysis or establish causality on its own.

Documentation ingestion and indexing

- Ingest approved documents through read-only connectors or controlled uploads; extract text, tables, revision metadata and equipment identifiers. Preserve document owner, effective date, revision, section/page references, access restrictions and provenance.
- Chunk and index passages using keyword and semantic retrieval, linked where possible to site, line, asset, process step, recipe, fault code, material and operating mode. Keep indexes within the plant boundary when data sovereignty or offline operation requires it.
- Apply user- and node-level access control before retrieval; exclude obsolete, superseded or unapproved documents from authoritative recommendations unless explicitly included as historical evidence.

RAG-assisted root-cause and deviation workflow

- 1. Trigger: an SPC special-cause signal, quality deviation, alarm or engineer question establishes the affected asset, time window, process stage and observed symptoms.
- 2. Retrieve: use the process graph and event context to search relevant manuals, SOP limits, historical deviations, maintenance actions and documented failure modes; rank by relevance, revision validity and source authority.
- 3. Correlate: compare retrieved troubleshooting guidance and documented operating envelopes with time- aligned local measurements, upstream/downstream evidence and contradictory observations.
- 4. Hypothesize: generate ranked potential causes, distinguishing directly observed facts, document-based mechanisms, statistical associations and unverified possibilities.

- 5. Recommend: propose safe, testable inspection steps, containment actions, maintenance checks and corrective/preventive actions; include expected mechanism, required approvals and verification criteria.
- 6. Cite and learn: attach document title, revision, section/page and evidence IDs to each material recommendation; record engineering disposition and post-action SPC results for subsequent investigations.

RAG output and governance

A structured investigation record should include query and process scope; retrieved document identifiers, revision/effective dates and passage citations; relevant numerical evidence IDs; supported and contradicted hypotheses; confidence and missing evidence; proposed actions; human approval status; and follow-up verification.

Retrieval failures, stale documents, conflicting procedures and absent citations must be surfaced rather than hidden.

An SLM may synthesize the explanation, but deterministic SPC, safety interlocks, approved procedures and engineering sign-off remain authoritative.

This is a proposed architecture and requires validation with customer document repositories and permissions.

5. Deming-inspired statistical process control

W. Edwards Deming emphasized understanding variation and distinguishing common causes from special causes. DMI should operationalize that distinction before producing root-cause claims. Control limits reflect observed process behavior, whereas engineering specification limits describe acceptability; the two are not interchangeable. An out-of-spec product can arise from a statistically stable but incapable process, while an in-spec process can still exhibit a special-cause signal.

5.1 Statistical engine

The local engine selects charts by data type and sampling plan: Individuals/Moving Range for individual measurements, X-bar/R or X-bar/S for rational subgroups, p/np for defectives, and c/u for counts of defects. Phase I establishes a credible stable baseline with documented exclusions; Phase II monitors against approved frozen limits. Rule sets (e.g., points beyond three-sigma limits, sustained runs or trends) are configurable and tested for false-alarm behavior. Autocorrelation, seasonality, product mix and recipe changes may require stratification or specialized models.

For an illustrative Individuals chart, CL = mean(x), moving ranges $MR_i = |x_i - x_{i-1}|$, sigma estimate $\approx$ mean(MR)/1.128, and control limits CL $\pm$ 3 sigma. These are not universal formulas: appropriate estimators and chart design depend on process and sampling assumptions.

5.2 From SPC signal to root cause

The trigger is a statistically supported deviation, not merely a threshold crossing. DMI should compare the excursion with changes in upstream input distributions, machine states, recipe versions, maintenance, operator interventions and lot history. Hypotheses are ranked by timing, process plausibility, statistical association and corroborating evidence. Recommendations include what to inspect, what setting or input to

verify, what evidence would falsify the hypothesis, and how to confirm improvement without tampering with a stable process.

6. Automated Unified Namespace from PLC discovery



Figure 4. PLC-to-UNS workflow: automated discovery proposes a namespace; semantic validation establishes trusted meaning.

DMI can bootstrap a Unified Namespace by discovering reachable PLC endpoints and tag metadata using authorized industrial interfaces such as OPC UA, MQTT and vendor-specific adapters. The system inventories signal names, data types, units where exposed, update rates, hierarchical paths and quality flags. It then proposes a normalized site/area/line/cell/asset/signal structure and maps streams to an event/measurement contract.

A critical distinction: collecting PLC tags alone does not reliably reveal physical process flow, true engineering units, lot genealogy, ISA-95 context or causal dependencies. Automated generation should therefore produce a confidence-scored proposed UNS, followed by engineering validation and enrichment from equipment documentation, production schedules, MES, recipes and operator input. The resulting semantic graph is versioned and can evolve as new assets appear.

6.1 Example namespace

plant/atlanta/area-1/line-2/extruder-03/process_pressure {value, unit: bar, ts_utc, quality, source, asset_id, schema_version}

Asset and process identifiers should remain stable even when PLC tag paths change. Publish/subscribe transport and semantic namespace are related but separate concerns; DMI can support different transports while maintaining one logical context model.

7. Automated upstream/downstream analysis and recommendations

When inspection finds an abnormal scrap rate, the graph identifies the producing process, its immediate inputs and relevant upstream ancestors. The orchestrator aligns time windows using material transit times, machine cycle times and batch genealogy, then asks each edge node to compute local summaries. Downstream queries determine propagation, affected lots and potential containment scope. The system returns a causal investigation map rather than a single unsupported explanation.

Stage	System question	Evidence returned
Detect	Is the defect signal statistically unusual?	Chart, rule, baseline, confidence
Localize	Which process step first changed?	Aligned changes and event chronology
Trace upstream	Which input or setting preceded the change?	Scoped comparisons and lot relationships
Trace downstream	Which lots/assets may be	Exposure graph and containment

	affected?	candidates
Recommend	What action is safest and testable?	Ranked action, expected mechanism, owner
Verify	Did variation and loss improve?	Post-action SPC, quality and cost impact

8. Process Twin Simulation — graph-aware, plant-wide what-if analysis

DMI's process twin is an operational model of the production flow, grounded in the process graph and current evidence. It evaluates proposed changes to setpoints, rates, recipe parameters, cycle times, material characteristics, and operating constraints before plant personnel apply them. Unlike a standalone equipment simulator, it propagates consequences across linked stages, material and energy flows, quality dependencies, and downstream constraints.

How a proposed change propagates

- 1. Select the change: identify the site, line, asset, parameter, proposed value or range, effective time, and baseline operating condition.
- 2. Resolve the graph: traverse verified upstream inputs and downstream dependencies, including branch/merge relationships, material buffers, transport delays, batch genealogy, and shared utilities.
- 3. Build a bounded scenario: select relevant local historian windows, operating envelopes, engineering constraints, and model versions; mark missing or uncertain relationships.
- 4. Simulate the connected flow: combine first-principles rules, empirical response curves, statistical models, and validated local predictors as available. Propagate time delays, capacity constraints, and uncertainty; avoid claiming causal certainty from correlation alone.
- 5. Compare outcomes: baseline versus scenario for throughput, cycle time, WIP, scrap/rework, specification compliance, OEE components, energy consumption, and bottleneck migration.
- 6. Review evidence and approve: show assumptions, confidence/uncertainty intervals, model provenance, affected nodes, and the reason each downstream estimate changed. Never automatically write a proposed setpoint to PLC/SCADA; authorized operators use established change-control workflows.

Illustrative connected-process scenario

A plastics line considers raising extruder temperature by 5 °C to increase throughput. The graph links the extruder to cooling, haul-off, dimensional inspection, and packaging. The twin evaluates melt behavior and capacity, cooling residence time, downstream line speed, dimensional variation and potential reject rate. A faster extruder may not raise sellable output if cooling becomes the constraint; it may instead increase scrap and energy. All numerical outcomes require site-specific calibration and validation, and this example is not a claimed measured result.

Technical model and deployment

The process graph stores typed directed relationships (feeds, depends_on, shares_utility, constrains, measured_by), units, allowable ranges, lags, and confidence. Edge nodes compute local response estimates

against local historical data; the plant orchestrator joins compact scenario outputs into a site-bounded result. The simulation engine should support deterministic constraints and mass/energy balance where applicable, calibrated empirical models where available, and optional SLM explanations only after the numerical scenario is fixed. This is a decision-support twin, not a real-time PLC control loop.

Validation and acceptance criteria

- Graph coverage: demonstrate the correct affected-path traversal, including branches and bottlenecks, with explicit missing-edge warnings.
- Prediction fidelity: compare held-out historical events and approved controlled trials against predicted ranges; report error by KPI and operating regime.
- Safety and governance: enforce setpoint limits, immutable scenario audit records, versioned models, and human approval for any implementation.
- Edge economics: measure scenario runtime, memory, node count, and network traffic on the proposed inexpensive hardware.
- Business value: quantify the difference between simulated optimum and actual approved trial results, avoiding double-counting quality and throughput benefits.

Product maturity note: graph-aware scenario propagation is a product capability/roadmap emphasis; each specific physical response model and claimed predictive accuracy must be verified against the deployed code and customer data before external performance claims.

8.1 Process graph as simulation topology

Each process-graph edge is typed and directed: material flow, energy dependency, control dependency, shared utility, quality inheritance, or resource contention. Edge metadata includes direction, transport lag, capacity, operating envelope, units, and confidence. A scenario begins at the changed variable and traverses the affected subgraph; feedback loops require iterative solution or explicit exclusion. The graph limits scope and avoids treating unrelated nearby machines as causal dependencies.

8.2 Scenario engine and mathematical treatment

For each process stage i, estimate a state transition $x_i(t+\Delta t)=f_i(x_i(t),u_i(t),z_i(t),\theta_i)$, where x is process state, u is the proposed control input, z represents upstream inputs and disturbances, and θ contains calibrated parameters. The function may be a physics-based balance, engineering rule, empirical response surface, or locally trained predictor. Downstream inputs are derived from upstream outputs with material-flow constraints, yields, buffers, and transit delays. Models must not extrapolate silently outside validated operating ranges.

For branching lines, propagate mass or unit counts through each branch with yield factors; for merging lines, enforce synchronization and shared-capacity constraints. A bottleneck may shift downstream even when the modified asset increases its own output. For recirculation or feedback, run a bounded iterative solver with convergence criteria, and report failure to converge instead of inventing a stable answer.

8.3 SPC-informed scenario eligibility

Use Shewhart control-chart rules to establish whether the baseline is statistically stable. If special-cause variation is active, flag scenario estimates as conditional and prioritize investigation before optimization. Separate control limits (process behavior) from specification limits (customer requirements). Show

predicted capability and defect risk only when distributional assumptions and sampling quality support those estimates.

8.4 Distributed computation and dual-loop reasoning

Loop 1 runs local models on inexpensive edge nodes using resident PLC/historian series and returns compact scenario results: affected asset, parameter delta, baseline/scenario KPI, units, prediction interval, operating regime, source window, model version, and data-quality flags. Loop 2 traverses the process graph, resolves time lags and capacity conflicts, composes the plant-level scenario, ranks tradeoffs, and generates an evidence-backed explanation. The SLM communicates results; it does not override deterministic constraints or invent missing process relationships.

8.5 Decision report and financial attribution

Present baseline, proposed change, predicted local benefit, downstream penalties, total line throughput, first-pass yield, scrap/rework, energy per good unit, WIP, bottleneck location, and constraint violations. ROI should use incremental good units × contribution margin, avoided scrap × verified avoidable cost, and measured energy or labor savings, net of implementation and operating costs. Deduplicate overlapping benefits; include scenario uncertainty and a sensitivity range. Compare approved trials to predictions and feed error back into model calibration.

8.6 Example: speed increase that moves the bottleneck

Illustrative only: a forming station runs at 100 units/minute and downstream inspection can handle 95 units/minute. Increasing forming to 110 units/minute does not raise sustained good output beyond inspection capacity. Instead it may increase WIP, starvation/blocking cycles, or defect exposure. The process twin should report that the limiting constraint moved (or remained) at inspection and recommend investigating inspection capacity or balancing the line before approving the forming-speed change. Numerical values are hypothetical, not product benchmark results.

8.7 Governance and proof requirements

Simulations are advisory and read-only by default. Require explicit approval before changes to PLC setpoints or recipes; preserve audit trails, operator identity, model provenance, scenario inputs, and observed outcomes. Acceptance testing should include correct graph traversal, lag alignment, constraint enforcement, held-out predictive error, uncertainty calibration, node memory/CPU footprint, and reproducible comparison against actual controlled trials.

9. Automated ROI and financial attribution

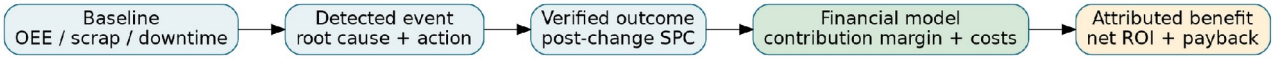


Figure 5. Financial closed loop: baseline → event → action → verified operating outcome → attributable financial value.

ROI is a measured consequence of improved manufacturing outcomes, not a model-generated estimate without supporting production and finance data. The system captures baseline performance by product, line, shift and operating mode; links events and interventions to operational KPIs; measures post-action results; and translates verified changes into financial impact using plant-approved unit economics.

9.1 Benefit categories and calculation

Benefit stream	Illustrative method	Key caution

Scrap reduction	Avoided scrap units × avoidable unit cost	Exclude recovered/reworked value twice
Downtime reduction	Recovered productive hours × contribution margin per constrained hour	Do not count unconstrained theoretical capacity
Yield / throughput	Incremental saleable units × contribution margin	Verify demand and bottleneck constraint
Energy efficiency	Normalized kWh saved × applicable marginal tariff	Normalize mix, volume and weather where relevant
Maintenance	Avoided repair/event cost net of intervention cost	Use probability-weighted avoided costs
Quality claims	Verified reduction in returns/rework/warranty cost	Avoid overlapping scrap and warranty claims

Annual net benefit = validated annual gross benefit − incremental recurring operating costs. First-year ROI = (first-year gross benefit − first-year total implementation and operating cost) / first-year total cost. Simple payback (months) = initial investment / average monthly net cash benefit, where net benefit is positive and recurring. For multi-year evaluation, use discounted cash flows, NPV and scenario analysis.

9.2 Worked example — illustrative only

Suppose a line produces 1,000,000 units annually, baseline scrap is 4.0%, and verified post-intervention scrap is 3.2%, with an avoidable scrap cost of $12/unit. The 0.8 percentage-point improvement equals 8,000 fewer scrap units and $96,000 annualized gross benefit, assuming stable volume, mix and attribution. If annualized recurring operating cost is $20,000 and initial deployment cost is $50,000, first-year total cost is $70,000 and first-year net benefit is $26,000; first-year ROI is 37.1%. Ongoing annual net benefit after the initial deployment year would be $76,000, assuming unchanged results and cost. These figures are illustrative, not a CogniaWerks performance claim.

An ROI evidence ledger should record baseline dates, excluded events, control-group or matched-period comparison, confidence intervals, approval authority, assumed capacity utilization, accounting treatment and benefit overlap. Financial validation is a joint production-and-finance responsibility.

10. End-to-end example: extrusion dimensional drift

1) The inspection node detects a special-cause pattern in diameter measurements and produces a control-chart evidence record. 2) The graph locates the extrusion, cooling and material-feed processes, with their time offsets. 3) Edge queries reveal a cooling-flow shift preceding the diameter drift; upstream material and extruder pressure remain within their validated baselines. 4) The plant loop ranks cooling instability as a leading hypothesis, with supporting and contradictory evidence. 5) An engineer inspects the cooling circuit and approves corrective maintenance. 6) Post-action SPC verifies restored stability; the ROI engine estimates scrap reduction against a comparable baseline and routes the calculation for finance sign-off.

This example demonstrates a proposed workflow, not a documented customer deployment or proof that the observed association establishes causality.

11. Security, governance and data residency

- Zero-trust node identity, mutual authentication and role-based query authorization; segregate OT control traffic from analytical services.
- Default-deny raw-data export; classify evidence fields and apply retention, redaction and aggregation rules.
- Support air-gapped deployments with local model artifacts, update controls and audit logs.
- Record source provenance, timestamps, unit conversions, model versions, operator approvals and analytical reproducibility.
- Establish human approval gates for any operational change and preserve safety systems independent of DMI.

12. Implementation and acceptance criteria

Phase	Primary deliverable	Acceptance evidence
Discover	PLC inventory and proposed UNS	Coverage %, semantic mapping review, units and quality checks
Contextualize	Validated process graph	Engineer-approved relationships and time lags
Baseline	SPC charts and rule configuration	Stable baseline, documented false-alarm behavior
Investigate	Dual-loop RCA workflow	Reproducible event replay and evidence trace
Recommend	Human-reviewed action workflow	Action ownership, permissions and verification plan
Value	ROI dashboard and benefit ledger	Finance-approved assumptions and verified outcomes

Suggested pilot metrics include: percent of critical tags mapped to trusted assets; completeness of process relationships; event detection delay; investigation time; false-positive rate; percent of conclusions with auditable evidence; SLM resource consumption; number of raw-data exports avoided; and finance-verified benefit. Targets should be negotiated per plant rather than asserted universally.

13. Differentiation and boundaries

DMI combines a process-context graph, local statistical detection, federated evidence collection, AI-assisted cross-process investigation, automated namespace bootstrapping and closed-loop financial validation. The claimed differentiation is architectural integration and workflow, not the assertion that graph databases, SLMs, SPC or UNS are individually new. A provisional patent-pending description does not by itself establish novelty, scope or freedom to operate.

DMI is not a replacement for PLC control logic, safety systems, MES, historians or quality management systems. It should complement those systems by reducing the distance between a process question, the evidence that answers it, the corrective action and the verified outcome.

14. External references and technical foundations

W. Edwards Deming Institute — Deming's System of Profound Knowledge and common/special causes
https://deming.org/explore/system-of-profound-knowledge/

NIST/SEMATECH — e-Handbook of Statistical Methods — process monitoring and control charts
https://www.itl.nist.gov/div898/handbook/

OPC Foundation — OPC UA industrial interoperability and information modeling
https://opcfoundation.org/about/opc-technologies/opc-ua/

ISA — ISA-95 enterprise-control system integration
https://www.isa.org/standards-and-publications/isa-standards/isa-95-standard

OASIS — MQTT specification and publish/subscribe messaging
https://mqtt.org/

NIST — SP 800-82 Rev. 3 — Guide to Operational Technology (OT) Security
https://csrc.nist.gov/pubs/sp/800/82/r3/final

NIST — AI Risk Management Framework
https://www.nist.gov/itl/ai-risk-management-framework

Technical and legal notes

This paper is a product-architecture proposal based on the supplied CogniaWerks concept. "provisional patent-pending graph database" is used as a positioning description; precise patent application identifiers, claims, jurisdiction and filing dates should be inserted after legal review. External standards and references are background foundations, not endorsements. All numeric ROI and process examples are hypothetical. Before customer publication, validate supported adapters, tested SLM footprints, measured latency, product availability and any security/compliance statements.

Technology and SME Deployment Addendum | November 2026

Market focus: small and mid-cap manufacturers

DMI is designed for plants with limited OT/IT staffing, heterogeneous PLCs and legacy historians, constrained capital budgets, and immediate needs to reduce scrap, downtime and troubleshooting effort. Begin with one production cell and expand after a verified economic result.

Source-aligned implementation architecture

PLC/SCADA/OPC UA/MQTT or historian → local operator tables → process/UNS discovery → graph-directed task planning → local SPC/OEE/diagnostics → evidence contract → plant-level correlation and explanation. Existing PLCs, SCADA, MES and historians remain systems of record. Read-only is the default integration stance.

Automated UNS: discovery is not certainty

Identify source tags, timestamps, units, site/area/cell associations and candidate manufacturing type. The product guide describes a 29-type catalog plus custom type, and a human confirmation step when operator identification votes conflict. Generate proposed ISA-95 hierarchy and ISA-88 batch/phase context; bind validated tags to equipment and stages. Never imply that raw tag names alone prove process causality.

Graph database and process intelligence

Model nodes for site, process, stage, asset, sensor, batch, event, document and evidence; edges for feeds, precedes, measured-by, belongs-to, affects and documented-by. Traversals are site-bounded, time-windowed and lag-aware. The provisional patent-pending claim should be tied to a reviewed application and claims, not asserted as an issued patent. The graph is a causal-hypothesis organizer, not proof of causation.

Dual-loop SLM orchestration

Fast loop: a local node performs numerical SPC, change-point detection, OEE and diagnostics and optionally uses a compact SLM to phrase findings. Deliberative loop: a plant orchestrator uses graph-directed fan-out to request only relevant evidence, joins time-aligned findings, compares alternative explanations and produces recommended actions. The model must not override deterministic SPC or fabricate evidence.

Evidence contract and guardrails

Each node response should include node/site/asset/process IDs; source timestamps and clock quality; variable/unit/quality; time window; event/anomaly score; method; confidence and limitations; model/version; source-query lineage; and access policy. Reject mismatched units, cross-site causal leaps, missing timestamps, ungrounded recommendations and uncertain namespace bindings.

Deming and Shewhart statistical discipline

Compute suitable control charts and rational subgroups, establish baseline and control limits, detect special-cause signals and avoid tampering with common-cause variation. Report capability separately from stability. Correlate upstream deviations with downstream effects using material-flow lag and batch/lot genealogy; label associations as candidates pending engineering confirmation.

Small-footprint edge economics

Offer software-only deployment on qualified customer hardware and turnkey industrial nodes with integrated acceleration where appropriate. Selected appliance configurations target hardware below $10,000, not a universal BOM promise. Benchmark CPU/NPU/GPU support, 8–16 GB memory tiers, local storage retention, thermal limits, inference latency, cybersecurity and offline operation before quoting a configuration.

Automated ROI and attribution

Baseline loss by availability, performance and quality. Estimate recoverable value using verified throughput, scrap, labor and downtime rates, then subtract deployment, support, energy and operational costs. Track realized versus projected savings, avoid double-counting overlapping losses and require plant approval of financial assumptions. ROI = (annual attributable benefit − annualized costs)/annualized costs.