

PILLAR 02 · THE INTELLIGENCE

Causal Reasoning and Hybrid Intelligence in Mining Operations

How to combine structured flowcharts and language models so a system makes good decisions under pressure

01

01 · Executive Summary

01 · Executive Summary

72% of AI initiatives in Latin American mining fail to meet their value objectives, according to the *Mining Autonomy* study by NTT DATA and MIT Technology Review (2025). The most cited cause is not technological: it is the disconnection between what systems can recommend and what organizations are willing to execute.

This whitepaper addresses the layer immediately prior to that adoption problem: **the design of the reasoning system**. Most AI projects in mining fail not only because the organization does not adopt them, but because the system itself reasons poorly — too rigid to adapt to changing context, or too unconstrained to be reliable in operations where errors have physical consequences.

The central thesis of this document is that the question *flowchart or language model?* is the wrong question. The correct design does not choose between structure and adaptability — it combines them based on available confidence. And it adds a third capability that neither has alone: **causal reasoning**. The ability to understand the temporal sequence in which events occur — not just their co-occurrence — and to anticipate consequences before they become visible in monitoring systems.

This architecture is not the design of a better dashboard. It is the reasoning engine that runs inside **decision agents** — AI systems capable of not only recommending actions but executing them within a defined autonomy perimeter, while optimizing toward an explicit objective. Understanding that distinction — between an agent that automates a workflow and one that optimizes a decision — is the most important thing this document communicates to an executive building an AI roadmap for mining.

WHO THIS DOCUMENT IS FOR

- **Innovation and digital transformation managers:** who need to justify an AI architecture to the board and want arguments based on research, not trends.
- **AI and data architects:** who are designing or evaluating operational decision systems and need to understand the design decisions that determine whether the system will work under real pressure.
- **Mining company CTOs:** who have seen AI projects die in implementation and want to understand where the design problem was — not just the change management problem.

02

02 · The Problem the
Industry Is Solving
Incorrectly

02 · The Problem the Industry Is Solving Incorrectly

In a control room at a copper mine in Calama, the fleet optimization system had just made an impeccable recommendation: pull three trucks out of operation. The weekly production plan was already met. Keeping the full fleet running would generate more costs, more queues, more wear. The numbers were clear.

The manager looked at the screen. Looked at the trucks. And sent them to the stockpile.

The technology was right. The organization did the opposite. And no one in that room called it a failure — because the manager's KPI measured trucks in motion, not cost per tonne for the entire system.

This pattern repeats itself in flotation plants, dispatch centers, and planning rooms across the sector. But there is something prior to the adoption problem that this document wants to analyze: **the reasoning system behind that recommendation.**

Did the system understand why the manager was going to ignore its recommendation? Could it explain in operational language — not mathematical optimization terms — why its recommendation was correct? Could it present information in a way that let the manager make the decision without feeling like he was losing control?

In most cases, no. And that incapacity is not just an interface problem — it is a reasoning architecture problem.

2.1 · THE TWO WRONG CAMPS

When a mining organization faces the operational decision problem, the technology debate quickly polarizes into two positions:

THE PLAYBOOK CAMP	THE LLM CAMP
Operational procedures exist for safety reasons — they are the scar tissue of past accidents. They cannot be improvised in real time. An LLM reasoning freely in an operation where errors can cost lives is an unacceptable risk.	Procedures are too rigid. A language model can reason with dynamic context, integrate information from multiple sources, and propose solutions no static playbook anticipated.
What they get right: structure matters. Rules are not bureaucracy.	What they get right: context matters. There is data not in any system.
What they miss: context changes. A playbook written for average conditions cannot cover the real variability of an operation.	What they miss: in critical operations, unconstrained improvisation is exactly what procedures were designed to prevent.

Both positions capture part of the truth. Neither captures the complete problem. And the practical result is that projects are designed from one of the two positions, and fail because of that position's limits.

Research published by NTT at ACL 2025 — the Annual Meeting of the Association for Computational Linguistics — demonstrated empirically why single-paradigm systems fail at complex tasks. The paper *ACT: Agents with Collaborative Tasks* showed that multi-agent systems produce superior results when each agent develops knowledge about the others during the diagnostic process — rather than operating in parallel with their fragment of information without active communication.

The implication for mining is direct: the room operator, the field technician, the dispatcher, and the planner are agents with distinct knowledge domains. The AI system assisting them needs to build shared knowledge actively — not aggregate their independent outputs.

2.2 · THE THIRD PROBLEM BOTH CAMPS IGNORE

There is a problem that neither the playbook nor the LLM resolves alone: **temporal causality**. The ability to understand that signals in a mining operation are not events that occur simultaneously — they are links in a causal chain with measurable lags.

When ore is harder than expected, the sequence is predictable: first the resistance at the shovel, minutes later the larger fragmentation, then the tonnage drop in the trucks, then the longer processing time at the crusher, then the queue. Each signal is a consequence of the previous one.

A system that does not understand that chain treats signals as independent events that coincide in time. It generates late alarms. It recommends reactive interventions. And when there is contradiction between sources — the shovel operator says the face is hard but the fragmentation sensor says normal — it has no criterion to resolve the contradiction.

"The winner is not the data that is said more often. The winner is the data that occurred first in the causal chain." — Business Talk, Article 11 · NTT DATA, 2025

This limitation is not solved with a better LLM or a more detailed playbook. It is solved with a reasoning layer that explicitly represents causal relationships between operational variables — and that uses that representation to anticipate rather than only detect.

03

03 · The Principle of Causal Reasoning

03 · The Principle of Causal Reasoning

Causal reasoning in mining operations starts from a simple observation: **operational problems do not happen suddenly — they propagate**. From the original event until the problem is visible in the monitoring system, there is a sequence of intermediate signals with typical measurable and predictable lags.

3.1 · THE CAUSAL CHAIN AS AN ANTICIPATION MODEL

Take the case of hard ore. The complete sequence — from the earliest to the latest signal — looks like this:

STEP	SIGNAL	DATA SOURCE	TYPICAL LAG FROM ORIGIN
1	Anomalous resistance at the shovel	Shovel operator (direct observation)	0 min — earliest signal
2	Larger fragment size than expected	Fragmentation sensor / visual	2–8 min
3	Low tonnage per load	Dispatch system — truck scale	5–15 min
4	Longer haulage cycle time	Fleet management system	10–25 min
5	Drop in face productivity	Dispatch analytics	15–35 min
6	Longer processing time at crusher	Plant historian	20–50 min
7	Truck queue at crusher	Dispatch / camera	25–60 min — latest signal

A system that monitors the Step 1 signal and understands the causal chain can anticipate the Step 7 queue with a 25–60 minute intervention window. A system that only monitors aggregate production indicators detects it when it has already occurred.

The difference between those two architectures is not computational power — it is whether the system has or does not have an explicit representation of causal relationships between variables.

3.2 · HOW TO BUILD A DYNAMIC CAUSAL GRAPH

The technical implementation of this principle uses a **dynamic causal graph**: a data structure representing nodes (operational variables) and directed edges (causal relationships with temporal direction and estimated lags).

Graph components:

- **Nodes**: operational variables with real-time values (tonnage, hardness, availability, temperature, pH, froth level, etc.)
- **Causal edges**: directed relationships $A \rightarrow B$ representing 'A causes B' with typical lag and variability range
- **Confidence weights**: quantify how established each causal relationship is based on the operation's specific history
- **Dynamic updates**: the graph recalibrates with each shift using the learning loops (section 05)

Implementation technology:

- **Neo4j**: graph database for storing and querying causal relationships
- **PyMC**: Bayesian probabilistic modeling to estimate and update the confidence weights of each edge
- **TimescaleDB**: time series storage for the operational digital twin

Why causality and not correlation:

Correlation detects that A and B occur together. Causality establishes that A occurs before B and is its cause. The difference is critical in operations:

EXAMPLE

The correlation between 'high hardness' and 'crusher queue' exists — but if the system only detects correlation, it can only act when both signals are present.

Causality establishes that 'high hardness' **produces** 'crusher queue' with a lag of 25–60 minutes. The system can act when it detects high hardness, before the queue exists.

Additionally, causality allows resolving contradictions between sources: if the shovel operator reports high hardness but the fragmentation sensor has not confirmed it yet, the system understands that the operator's observation is earlier in the causal chain — and gives it greater weight in the current diagnosis.

3.3 · HOW FLOWCHARTS ARE BUILT: EXTRACTING DECISION LOGIC FROM EXPERT DIALOGUE

The hybrid architecture assumes structured flowcharts exist. The harder question — where do they come from? — has a technical answer backed by NTT DOCOMO research published at ACL 2025 (Findings).

Yamanaka et al. propose a method for extracting flowcharts automatically from expert dialogue data: shift handover logs, radio communications, records of operators working through decisions in real conditions. The extraction process follows four phases:

1. **Segment** dialogues into decision points — moments where the expert chose between alternatives
2. **Identify** the conditions that triggered each decision — what information the expert was waiting for before acting
3. **Map** transitions between points — if condition A then step B; otherwise step C
4. **Abstract** the flowchart from specific instances into a generalizable structure applicable to future shifts

The result captures something critical that procedure manuals miss: **decision logic the expert follows but does not explicitly articulate** — the order in which information is collected, the thresholds that exist in practice but appear in no document, the conditions that are necessary but assumed.

CONNECTION TO PILLAR 03

This extraction method is the computational equivalent of what the Knowledge Condenser methodology (Pillar 03 · The Organization) does through structured elicitation. Both address the same problem from different angles: tacit knowledge that lives in expert practice and must be converted into something a system can use. Knowledge Condenser captures physical-mechanical correlations (the machine behavior a veteran operator perceives before any sensor). Yamanaka's method captures decision logic (the sequence of questions and thresholds that operator applies when making a diagnosis).

Together, they are the complete knowledge pipeline from expert to system.

Soft vs. hard flowchart integration:

The extracted flowchart can be integrated into the LLM pipeline in two modes — a design decision that maps directly to the three reasoning modes described in section 04:

INTEGRATION MODE	HOW IT WORKS	WHEN TO USE
Soft	Flowchart serialized into the LLM's context — the model reasons within the decision space the flowchart defines, but the constraint is semantic	Mode 2 (0.55–0.75 confidence): LLM adapts parameters within the flowchart frame, can propose actions the static procedure did not anticipate
Hard	At critical decision points, the LLM's output is constrained deterministically — the model cannot generate a response that contradicts the flowchart node	Mode 1 (≥ 0.75 confidence): safety-critical steps where the procedure must be followed exactly and auditability is required

This distinction matters in operational environments: hard integration is what allows an AI system to be auditable — an operations team can trace exactly what decision node was active, what condition triggered it, and what the system was prohibited from doing. In a regulatory or incident investigation context, this traceability is not optional.

04

04 · The Hybrid Architecture: When to Use Each Mode

04 · The Hybrid Architecture: When to Use Each Mode

The central design principle of MOV — NTT DATA's operational intelligence platform — is that the reasoning mode is not fixed: **it varies based on the confidence level available for the current diagnosis**. This principle resolves the false dilemma between structure and adaptability.

4.1 · THE THREE REASONING MODES

CONFIDENCE LEVEL	ACTIVE MODE	HOW IT WORKS
≥ 0.75	Structured flowchart	The situation is recognizable and the procedure is applicable. The system executes the corresponding decision tree with its restrictions and safety controls intact. No improvisation. No free reasoning. The playbook is in control.
0.55 – 0.75	LLM + restriction policy	The situation has recognizable elements but also variables not anticipated in the procedure. The LLM reasons with current dynamic context, but within a constraint framework derived from the playbook. It can propose unanticipated actions — but cannot violate safety rules.
< 0.55	Controlled failure + human escalation	The situation is outside the known space. The system does not improvise or propose actions with low confidence. It escalates to human supervision with available information organized, active hypotheses listed, and a recommendation of what additional information is needed to make the decision.

WHY THE CONFIDENCE THRESHOLD IS THE MOST IMPORTANT DESIGN DECISION

A Mode 1 threshold that is too high (e.g., 0.90) makes the system escalate too frequently — it loses operational utility. A threshold too low (e.g., 0.60) makes the flowchart cover situations it was not designed for — it loses safety reliability.

The optimal threshold is a function of the specific operation, available data quality, and the system's historical performance in similar situations. It is established in the initial configuration and recalibrated with each slow learning loop (monthly).

In practical terms: an operation just beginning to use the system should operate with more conservative thresholds (more time in Modes 2 and 3) and gradually migrate toward lower thresholds as the reliability history is established. This is the first rung of the AI trust ladder.

4.2 · THE RESEARCH BEHIND THIS DESIGN

The combination of structure and adaptability is not a design opinion — it has published research backing.

NTT RESEARCH · ACL 2025

Paper: *ACT: Agents with Collaborative Tasks for the Design and Implementation of Complex Systems*

Conference: Annual Meeting of the Association for Computational Linguistics, 2025 — the most important academic conference in computational linguistics and natural language processing.

Central finding: multi-agent systems produce significantly superior results when agents develop active shared knowledge — each agent updates its understanding of what others know during the resolution process, rather than operating in parallel on their fragment of information.

Implication for mining: the room operator, the field technician, the dispatcher, and the geological model are agents with distinct knowledge domains. An AI system that treats them as independent data sources will produce inferior diagnostics to one that actively builds shared knowledge among them.

Connection to reasoning modes: Mode 2 (LLM + policy) is exactly the space where active collaboration between the human operator and the system produces the richest diagnostics — AI contributes data integration the human cannot process in real time; the human contributes contextual knowledge and expert judgment that no sensor captures.

WHAT THE FLOWCHART CONTRIBUTES	WHAT THE LLM CONTRIBUTES
Reproducibility and auditability of decisions	Reasoning with dynamic context not anticipated in the procedure
Incorporation of non-negotiable safety constraints	Integration of multiple heterogeneous information sources
Response speed in known situations	Ability to formulate the next most valuable question
Traceability for post-incident investigation	Communication in natural language with the operator
Operator trust through predictability	Adaptation to the specificity of each shift and face

The critical point is this: the two paradigms do not compete — they complement each other. The flowchart defines the safe action space. The LLM reasons within that space when context is too complex for the procedure. And the causal graph gives both the temporal representation they need to anticipate rather than only react.

4.3 · DECISION AGENTS: THE REASONING ARCHITECTURE IN OPERATION

The three-mode architecture described above is not the design of a smarter recommendation system. It is the reasoning engine that runs inside a **decision agent** — an AI system that does not merely suggest what an operator should do, but acts within a defined autonomy perimeter while optimizing toward an explicit business objective.

The distinction between a decision agent and a workflow agent is the most important architectural choice in an AI program for mining:

WORKFLOW AGENT	DECISION AGENT
Automates predefined sequences: reports, SOPs, approval chains, notifications	Optimizes decisions toward a technical and business objective within operating constraints
Executes the same steps faster	Determines which action is optimal at each specific moment given current operational state
Value: process efficiency within a domain	Value: outcome quality and system-level margin across domains
Dozens of platforms offer this	Gartner identifies this as the frontier differentiation in industrial AI (2026)

An operator always follows the process — but takes decisions *within* the process. A decision agent replicates that reasoning capacity: it adheres to SOPs but optimizes the decision parameters. The three reasoning modes are how it does this:

- **Mode 1** executes known-optimal procedures at speed, without decision overhead — the flowchart is the SOP, run deterministically
- **Mode 2** adapts parameters in novel situations while preserving the causal constraints that protect the operating margin — adjusting ore blend, crusher speed, or reagent dosing within the physically permissible space
- **Mode 3** recognizes when the situation is outside its competence and transfers to human judgment with a complete diagnostic — not a blank screen or a low-confidence recommendation executed anyway

The objective function:

A mining decision agent optimizes toward an explicit function:

$$\text{margin} = \text{tonnage} \times \text{grade} \times \text{recovery} \times \text{price} - \text{OPEX}$$

This makes the agent's purpose legible to any executive: not "the system made a recommendation" but "the system acted to maximize the operation's margin under current conditions, within the constraints of safe operation." Every decision the agent makes can be evaluated against that function — which is the condition for holding it accountable and for trusting the trust ladder to advance.

The causal graph as anti-hallucination layer:

A language model reasoning freely about industrial operations can generate plausible-sounding recommendations that violate physical causality. A model with no representation of causal structure might recommend increasing crusher feed rate at exactly the moment the causal graph shows a hardness surge propagating through the chain — because the correlation in its training data does not carry temporal direction.

The causal graph prevents this. It acts as a layer of unbreakable physical constraints that any LLM recommendation must pass through before reaching the operator or actuating a system. If a proposed action implies a causal direction the graph has established as incorrect — or implies acting on a consequence rather than its cause — the recommendation is rejected at the L4 reasoning layer.

This is the structural meaning of "grounded reasoning" in this architecture. It is not a philosophical property. It is enforced by the Neo4j + PyMC causal graph at inference time — and it is what separates a decision agent that can be trusted in a control room from a language model that can only be trusted in a chat interface.

DETERMINISM IN DCS ENVIRONMENTS

Decision agents in distributed control system environments must be deterministic: given the same operational state, two agents running in parallel must arrive at the same correct decision. This is non-negotiable in operations where both agents could actuate the same physical system.

This determinism is achieved through the flowchart layer: critical SOPs are encoded as structured JSON/YAML decision trees, not as free-form prompts. The LLM is not used for decisions that must be deterministic — it is used for diagnosis, context synthesis, and natural language communication with operators. The flowchart handles the actuating decisions. The result is an architecture where every actuation can be traced to a specific flowchart node and condition — the audit requirement that control system environments demand.

05

05 · The Learning Loop:
How the System Improves
in Production

05 · The Learning Loop: How the System Improves in Production

The most important differential of an operational intelligence system is not the reasoning quality at launch — it is the rate at which it improves in production. A system that does not learn fossilizes. In 12 months, the system that did not learn is less relevant than at the start, because the operation changed and it did not.

5.1 · THE THREE TEMPORAL LOOPS

MOV implements three learning loops with distinct frequencies and functions:

LOOP	FREQUENCY	WHAT IT UPDATES	TECHNOLOGY
Fast	Real-time (per event)	Causal graph weights for current shift · Confidence in active data source	Temporal.io + Redis (in-memory)
Medium	Per shift (every 8–12 hours)	Confidence threshold calibration · Incorporation of validated expert deviations	ClickHouse + Grafana (analytics)
Slow	Monthly	Structural recalibration of causal graph · Flowchart policy updates · Mode threshold review	PyMC (Bayesian refit) + PostgreSQL

5.2 · LEARNING THROUGH EXPERT DEVIATION

The most valuable — and most difficult to correctly design — learning mechanism is the capture of expert operator tacit knowledge at the moment they deviate from the system's recommendation.

An operator with 15 years at the same mine — with knowledge about the behavioral difference between crusher 2 and crusher 3 that is not documented in any system — can reach a correct diagnosis while ignoring the system's recommendation. That moment is not an error. It is the most valuable signal available to update the model.

STEP	ACTION	RESULT
1	Operator deviates from system recommendation	System records the deviation with timestamp, complete operational context, and the alternative action taken
2	System detects and classifies the deviation	Distinguishes between operator error, model limitation, or unanticipated context. Does not update automatically.
3	System formulates the cause question	At the next moment of low cognitive load: 'Why did you make that decision? Was there a condition the system wasn't seeing?'
4	Operator describes the criterion	Natural language explanation. The system does not require the operator to code the rule — only describe what they observed.
5	Human validation before updating	The shift supervisor or technical expert reviews the reported cause. Without validation, no update.
6	Model updated	The operator's criterion is incorporated as a new weight in the causal graph or as a new constraint in the flowchart. The next shift inherits that knowledge.

With the correct design, each shift a veteran operator works captures a fraction of that accumulated knowledge in the system — and in 12 months, the system knows things about that specific operation that no generic model can know.

5.3 · THE AUTORESEARCH PATTERN APPLIED TO THE DIGITAL TWIN

In March 2026, Andrej Karpathy — former AI director at Tesla, OpenAI co-founder — published a repository called *autoresearch* that reached 28,000 GitHub stars in a few weeks. The architectural pattern it proposes: autonomous agent + controlled environment + objective metric + continuous improvement loop. The human does not program parameters — they define intent in natural language.

The pattern is directly applicable to MOV's slow learning loop — with one critical difference: **in mining, the cost of a failed experiment is not compute time — it can be a crusher shutdown.**

That is why the pattern does not operate on the real operation. It operates on the **operational digital twin**: the Monte Carlo simulation model that runs in parallel to the real operation, receives the same real-time data, and allows experimenting with policy variations without physical consequences.

EXAMPLE OF EXPERIMENTATION IN THE DIGITAL TWIN

During the low-activity night shift, the system can systematically explore: 'What would have happened if the ore blend had been adjusted 20 minutes before the hardness signal appeared? What hardness threshold should automatically trigger a crusher speed adjustment?'

Each digital twin experiment feeds the real causal graph. Policies that improve the simulated margin — $\text{ton} \times \text{grade} \times \text{recovery} \times \text{price} - \text{OPEX}$ — go to human review before being activated in the field. The shift supervisor does not configure parameters: they define in natural language what to prioritize and what constraints to apply.

06

06 · Prerequisites for This
to Work

06 · Prerequisites for This to Work

The architecture described in this document can deliver real value in a mining operation. It can also learn to make mistakes faster if the prerequisites are not resolved. This chapter describes those conditions without sugarcoating.

6.1 · OPERATIONAL DATA INTEGRITY

The causal graph learns from data that operators record. If that data is contaminated by incentives — if operators document 'no critical condition was observed' when cameras showed water because reporting the real risk would affect their bonus — the system learns that lie with greater precision than any individual operator.

The NTT DATA and MIT Technology Review study (2025) identified that lack of specialized talent (27.54%) and high implementation costs (23.19%) are the most cited barriers, but the systemic contamination of data — which does not appear as a category in surveys because no one acknowledges it out loud — can be more damaging than both combined.

REQUIRED CONDITION

- **Revised incentive systems:** the KPIs that evaluate operators must align individual interest with the system's optimal — not reward not-reporting risks or not-stopping equipment when appropriate.
- **Active expert deviation culture:** the operator must have the cultural and technical permission to deviate from the system recommendation and report why. Without this permission, the learning loop never activates with clean data.
- **Separation between operational data and performance evaluation data:** what the system learns from decisions should not be visible to whoever evaluates the operator — at least in the initial implementation phases.

6.2 · OT ARSENAL AGENT-READINESS

The causal reasoning system needs to actively query the operation's systems — the SCADA historian, dispatch, maintenance system, geological model. If those systems were designed to be read by humans on screens, and not by agents that invoke them programmatically, the architecture cannot function as described.

Most mining operations are in this situation: they have good data systems, but none were designed to be consumed by an autonomous agent. Converting them requires integration work — specifically, the implementation of APIs or OPC-UA connectors that allow the system to query each data source in real time without human intervention.

REQUIRED CONDITION

- **OPC-UA connectors to SCADA/Dispatch/TIMS:** operational data integrity in real time without dependence on manual exports
- **Mobile PWA with STT (Whisper):** capture of field operator observations in natural language — the most valuable and least digitized data source
- **Validated data schema (Apache Kafka + Avro):** ensures data arriving at the causal graph has the correct format and adequate latency

6.3 · ORGANIZATIONAL WILLINGNESS TO SCALE GRADUALLY

The AI trust ladder — from copilot to autonomous orchestrator — is not climbed with budget. It is climbed with a demonstrated history of correct decisions. An organization that wants to implement the system in Mode 3 (orchestrator) from Day 1 is guaranteeing project failure.

The correct sequence is: pure Mode 1 first (the system only shows information, the operator decides everything), then Mode 2 introduced gradually in low-consequence situations, then expansion to higher-consequence situations to the extent that the reliability history justifies it.

The 72% failure rate in autonomy projects reported in the NTT DATA study (2025) has a partial technical explanation here: many projects try to implement too much autonomy too quickly, before the system has built the history that justifies that trust.

07

07 · What This Architecture Does Not Resolve

07 · What This Architecture Does Not Resolve

This whitepaper describes a reasoning architecture that works. It is also honest about its limits — because an executive who has lived through failed AI projects recognizes honesty as a differentiator more than any promise.

LIMIT	WHY AND WHAT IT REQUIRES
Does not replace autonomy governance	The reasoning architecture defines how the system decides — not who is responsible when it makes a mistake. The three reasoning modes do not resolve the legal and organizational question of responsibility in autonomous systems. That requires an explicit governance framework established before deployment — one that defines escalation paths, accountability roles, and the criteria for expanding or restricting the system's autonomy perimeter.
Does not work if the organization does not adopt the system	The most sophisticated causal reasoning produces no value if the operator ignores it. Adoption requires organizational design — process redesign, incentive alignment, active leadership — that is outside the scope of the technical architecture (see Business Talk Article 06 and Pillar 03 of this series).
Learns to make mistakes faster with contaminated data	The learning loop amplifies the quality of training data — in both directions. An organization with incentives that produce biased data will obtain a system that systematically recommends the actions its own biases favor. Data integrity is a non-negotiable prerequisite (section 06.1).
The digital twin cannot simulate what sensors do not capture	If the operation has areas without sensor coverage, unmonitored relevant variables, or legacy systems that cannot be integrated, the digital twin will have blind spots that learning experiments cannot explore. The correct PISO (Pillar 01) is the enabling condition of the intelligence system.
The trust ladder takes time — it cannot be compressed	The reliability history that justifies moving the Mode 1 threshold from 0.75 to 0.70 is built shift by shift. There are no budget shortcuts. An organization that wants Mode 3 results in 6 months is undervaluing what 'demonstrated trust' means in high-consequence operations.

08

08 · Related Reading and Next Steps

08 · Related Reading and Next Steps

All articles available at biztalksnttdata.com

RELATED TECHNICAL CONTENT IN THIS SERIES

- **Pillar 01 · The Floor: PISO Framework** — the OT infrastructure without which the causal graph does not have the data it needs to function
- **Pillar 01 · The Floor: Agent-readiness in mining** — how to convert existing systems into resources consumable by decision agents: OPC-UA connectors, Kafka streaming, the OT infrastructure that makes this architecture possible
- **Pillar 02 · The Intelligence: Data integrity in AI systems** — the incentive bias problem developed in full technical depth
- **Pillar 03 · The Organization: Tacit knowledge as critical infrastructure** — the Knowledge Condenser methodology captures the physical-mechanical knowledge that feeds the causal graph; Yamanaka et al. (section 3.3) captures the decision logic that generates the flowcharts
- **Pillar 03 · The Organization: The Bainbridge paradox in mining** — why successful automation erodes the skills it needs in its failures

NEXT STEPS

If this document describes a problem you recognize in your operation, the next conversation is technical.

1. **Maturity diagnostic:** assessment of prerequisites (sections 06.1–06.3) in your specific operation
2. **Pilot architecture:** design of the minimum scope that allows demonstrating value in 90 days without compromising operational safety
3. **MOV sandbox:** interactive demo of the three reasoning modes with synthetic mining operation data

The operator always decides. The system always learns.

NTT DATA · Mining & Resources · 2026 · biztalksnttdata.com · Pillar 02 · The Intelligence

For information about implementing MOV at your operation, contact: Jaime Rebolledo, Director - Mining · jrebolld@nttdata.com