FROM CLOUD DASHBOARDS TO DECISIONS AT THE MACHINE

Edge AI for Industrial IoT

Executive briefing — Manufacturing & OT/IT Leadership | ~20 min

The Business Question

- AI investments are expanding, but many industrial use cases need decisions in milliseconds, not seconds
- Cloud-only architectures were built for batch analytics, not line-speed control
- This briefing frames when edge AI is the right architecture, not just a trend to chase
- Goal: align OT and IT leadership on where edge AI fits our roadmap
- Decision points at the end: pilot scope, ownership, and budget ask

Why Cloud-Only AI Breaks Down on the Plant Floor

- Round-trip latency to a cloud region can exceed the tolerance window for real-time control loops or line-speed inspection
- High-resolution camera and sensor streams consume significant bandwidth if sent continuously off-site
- Many plant and remote sites have inconsistent or metered connectivity, especially outside major hubs
- A network outage should not mean a safety system or quality gate goes blind
- Cloud remains valuable for training, aggregation, and fleet-wide analytics — the question is where inference should run

What "Edge AI" Actually Means

- Inference — the act of running a trained model on new data — executes on local hardware near the sensor or machine
- Model training typically still happens in the cloud or a data center; the trained model is deployed down to the edge
- Edge devices range from ruggedized industrial PCs to purpose-built AI accelerator boards
- Local inference means decisions happen in milliseconds, without a round trip to the internet
- Edge AI is a deployment pattern, not a different category of algorithm — the same model families apply

Where It Applies: Visual Quality Inspection

- Camera-based defect detection at line speed, where cloud latency would force the line to slow down or buffer
- Local inference keeps inspection synchronized with conveyor or robotic cycle times
- Flags parts for rework or rejection in real time rather than after the fact
- Reduces the volume of image data that needs to leave the site — only exceptions or summaries are sent upstream
- Works alongside, not instead of, existing manual inspection during initial rollout

Where It Applies: Predictive Maintenance & Safety Monitoring

- Local anomaly detection on vibration, temperature, or acoustic sensors flags early signs of equipment degradation
- Edge inference avoids streaming continuous raw sensor data to the cloud for every asset
- Worker safety monitoring (PPE compliance, restricted-zone intrusion) requires immediate local alerting, not delayed cloud analysis
- Alerts can trigger local actions — a machine stop, an audible warning — without waiting on network round trips
- These use cases share a pattern: local speed and privacy first, cloud aggregation second

Edge Hardware Options and Tradeoffs

- Industrial PCs: flexible, familiar to IT, moderate compute — good fit for lower-throughput inference
- Edge GPUs: higher compute for demanding vision workloads, higher power and cooling requirements
- Dedicated AI accelerators (purpose-built inference chips): efficient and compact, but narrower software compatibility
- Tradeoffs to weigh: compute headroom, power/thermal limits in the plant environment, ruggedization, and vendor lock-in
- Hardware choice should follow from the specific use case's latency and throughput needs, not be selected first

The OT/IT Convergence Challenge

- Edge AI devices sit on the OT network but often need IT-style management, patching, and monitoring
- Network segmentation must be preserved — edge devices should not become a bridge between OT and corporate IT networks
- Each edge device is a new endpoint and a new attack surface; cybersecurity posture has to extend to the edge fleet
- OT and IT teams need shared ownership of the edge layer, including clear incident-response responsibility
- This is as much an organizational design question as a technical one

Deploying and Updating Models Across a Distributed Fleet

- Models need a defined path from training to validation to staged rollout across sites
- Fleet management tooling (device registry, version tracking, rollback capability) is a prerequisite, not an afterthought
- Staged rollout — pilot site, then a wider wave — limits blast radius if a model update underperforms
- Monitoring model performance in production (drift, accuracy degradation) needs to happen locally and be reported upstream
- Update cadence should balance improvement velocity against the operational risk of change on a live production line

Illustrative Plant Deployment Example

- Illustrative scenario, not a verified case study — presented to make the architecture concrete
- A mid-size assembly plant deploys edge inspection cameras on two high-speed lines, plus vibration sensors on critical rotating equipment
- Edge devices run inference locally; only flagged exceptions and summary metrics are sent to a central data platform
- A local edge gateway continues operating quality checks and safety alerts even if the plant's internet link drops
- Central IT retains visibility into fleet health and model versions without being on the control-loop critical path

Handling Offline and Degraded Connectivity

- Edge inference should default to continuing operation locally when the network connection is lost or degraded
- Defined fallback behavior is essential: what happens to alerts, logging, and control actions during an outage
- Local buffering with later sync ensures data is not lost, only delayed, once connectivity returns
- Failure modes should be tested deliberately, not discovered during an actual outage
- This is a reliability requirement to design in from the start, not a feature to add later

Total Cost of Ownership: Edge vs. Cloud-Only

- Edge adds upfront hardware and deployment cost per site, plus ongoing device maintenance
- Cloud-only avoids hardware capex but can carry substantial data transfer and compute costs at scale, particularly for continuous video
- Edge can reduce recurring bandwidth and cloud compute costs once deployed, though payback period varies by use case and site count
- Total cost comparisons should be treated as directional planning inputs, not precise unnamed benchmarks, until validated against our own pilot data
- The right architecture is often hybrid: edge for real-time inference, cloud for training, aggregation, and fleet-wide analytics

Integrating with SCADA and MES

- Edge AI outputs need a defined interface into existing SCADA and MES systems, not a parallel reporting path
- Common patterns include edge devices publishing alerts or tags that SCADA already knows how to consume
- MES can log AI-driven quality or maintenance events alongside existing production and genealogy records
- Integration should preserve existing operator workflows and alarm handling rather than introducing a second system to monitor
- Early engagement with SCADA/MES vendors and integrators reduces integration risk and rework

Risk Summary Before We Proceed

- Cybersecurity exposure grows with every new edge endpoint — a security review must be part of any pilot plan
- Model performance can drift over time; ongoing monitoring is a requirement, not optional overhead
- Cross-functional OT/IT ownership needs to be assigned before deployment, not resolved after an incident
- Hardware and integration costs should be validated with a small pilot before any site-wide commitment
- None of these risks are reasons to avoid edge AI — they are reasons to pilot deliberately

Recommended Next Steps

- Select one line or one site for a scoped pilot — recommend starting with visual quality inspection given its clear success criteria
- Form a joint OT/IT working group to own network segmentation, security review, and device management for the pilot
- Define success metrics and a review checkpoint (target: 8-12 weeks) before any decision to scale
- Engage existing SCADA/MES integration partners early to scope the data interface
- The ask: approval to fund a single-site pilot and designate OT/IT co-leads within the next two weeks