FIREBALL INDUSTRIES

IGNITE YOUR FACTORY EFFICIENCY

WHITE PAPER

AI Belongs on the Line

Deploying AI at the edge in manufacturing: vision inspection, anomaly detection, and predictive maintenance that run next to the machine, keep working when the network doesn't, and never take the wheel from the PLC.

By **Patrick Ryan** | Fireball Industries | November 2026



EXECUTIVE SUMMARY

The model has to live where the part is.

Manufacturers do not need to be sold on AI anymore. They have seen a camera catch a defect a person missed and a vibration model flag a bearing weeks before it failed. What stops most projects is not whether AI works. It is where it runs, who owns it, and what happens to the line when it is wrong.

Much of the AI on offer to plants assumes the cloud: send the data up, run the model in someone else's data center, send the answer back. That works for a monthly report. It does not work for a reject gate that has 100 milliseconds to decide, a camera producing more data than the plant's internet connection can carry, or a line that has to keep running when the WAN goes down at two in the morning.

Edge AI runs the model on hardware at the line: in a smart camera, on an industrial PC next to the cell, or inside the controller itself. The data stays in the plant, the answer arrives in milliseconds, and the line does not care whether the internet is up. Training can still happen wherever the compute is; inference, the part that makes decisions on live product, belongs on the floor.

Running AI on the line raises the stakes, so it needs rules. The most important one fits in five words: **AI advises, the PLC decides.** Model output is an input to a control program that already enforces limits. It never sits in the safety function, and the line keeps running safely when the model is unavailable.

This paper covers why inference belongs at the edge, where AI pays for itself on the plant floor, the hardware options from smart cameras to controllers, the lifecycle that keeps a model accurate after go-live, and the governance and security a plant should expect from anyone who deploys AI on its equipment.

100 ms

Time per part on a line running 600 parts a minute. That is the whole budget to capture, decide, and act.

1.2 Gb/s

Uncompressed output of one 5 megapixel monochrome camera at 30 frames a second.

0

Production decisions that should wait on an internet connection.

SECTION I

Why Inference Belongs at the Edge

Four constraints decide where a model has to run on a plant floor. Any one of them is enough to rule out a round trip to the cloud for a live production decision.

1. Time

A line running 600 parts a minute gives each part **100 milliseconds**. In that window an inspection system has to capture an image, run the model, and tell a reject gate or a robot what to do. A round trip over a wide area network takes a variable amount of time that the plant does not control, and the worst case, not the average, is what decides whether a bad part ships. A model on the line answers in a time that can be measured, bounded, and tested before startup.

2. Bandwidth

Vision is where most plant AI starts, and vision is heavy. One 5 megapixel monochrome camera at 30 frames a second produces about **1.2 gigabits a second** of raw image data. Ten of them produce twelve. Compression helps, but it also throws away the detail an inspection model needs, and few plants have an internet connection sized to carry continuous video from every station. Running the model next to the camera turns a video stream into a verdict: pass, fail, and why.

3. Availability

Internet links drop, cloud services have outages, and certificates expire at the worst possible moment. When inspection or monitoring depends on a connection off site, a network problem becomes a production problem. A model running at the edge keeps working when the link is down, and results are sent upstream when it returns.

4. Ownership

Images of your product, the signature of your process, and the timing of your line are competitive information. Many customers, especially in aerospace, defense, and pharmaceuticals, require that production data stay on site. Edge inference keeps raw data in the plant and sends out only what the business has approved: results, counts, and the occasional image chosen for review or retraining.

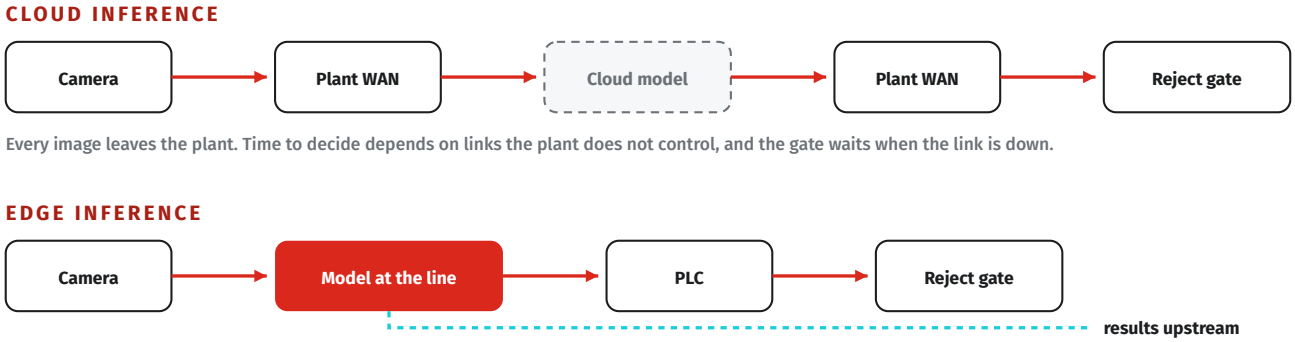

Figure 1. Cloud inference puts the network inside the decision. Edge inference keeps the decision on the line and sends results, not raw video, upstream.

SECTION II

Where AI Earns Its Keep on the Floor

The AI projects that pay back on a plant floor are rarely the most ambitious ones. They are the ones aimed at a cost the plant already measures: scrap, escapes, unplanned downtime, and labor that cannot be hired. Five applications account for most of the value we see.

APPLICATION	WHAT IT READS	WHAT IT DECIDES	WHAT CHANGES ON THE FLOOR
Vision inspection and defect detection	Images of every part, at line speed	Pass or fail, the defect type, and where it is on the part	Defects that a rules-based camera or a tired inspector misses get caught, and escapes and chargebacks fall
Vision-guided robotics	Images of parts in a bin, on a conveyor, or on a pallet	Where to pick, how to grasp, and where to place	Robots handle mixed, unsorted, or randomly placed parts that fixed programming cannot
Process anomaly detection	The values PLCs already produce: temperatures, pressures, flows, currents, cycle times	Whether the process is drifting from its normal pattern	An off-spec batch or a failing station shows up while there is still time to act
Predictive maintenance	Vibration, temperature, current, and run hours on rotating equipment	Which asset is degrading and roughly how soon it will need attention	Bearings, pumps, and motors are replaced on a planned shutdown instead of during a breakdown
Digital twins with AI feedback loops	A model of the line, plus live data from the real one	How a change in rate, recipe, or layout would play out before it is made	Changes are tested on the twin first, and the twin gets more accurate as real data flows back into it

Train on your own parts

The fastest way to a model that works on your line is to train it on your line. A defect model trained on your own good and bad parts, under your lighting, at your line speed, catches what a generic model misses, because the generic model has never seen your parts. The same holds for anomaly detection: what counts as normal is specific to each machine, and the model learns it from that machine's own history.

Start with a cost, not a technology

Every successful project we have seen began with a number: the cost of an escape, the cost of an hour of downtime on a specific line, the overtime spent on manual inspection. If the plant cannot put a number on the problem, it will not be able to tell whether the AI solved it. Our downtime calculator is a good place to start for the downtime side of that number.

SECTION III

Where the Model Runs

"The edge" covers a range of hardware. Picking the right place for a model depends on how fast it has to answer, how big it is, and how tightly it has to tie into control.

WHERE	GOOD FOR	STRENGTHS	TRADE-OFFS
Smart camera or vision controller	Single-station inspection, reading codes and labels, simple classification	Self-contained, compact, one vendor's tools from end to end	Limited compute; models and tools are usually specific to the camera vendor
Industrial PC with a GPU or NPU at the cell	Multi-camera inspection, vision-guided robotics, larger models	Serious compute next to the line; runs standard model formats	Needs to be built, hardened, and maintained like any other control-room computer
Inside the controller	Small, fast models whose output feeds control logic directly	The model answers inside the control cycle, and the result is a PLC variable	Model size is limited; the platform is tied to the controller vendor
Edge server for a line or area	Anomaly detection across many signals, several models at once, digital twins	Shared compute for many assets; easier to update and monitor centrally	Results reach the line over the plant network, so it suits decisions measured in seconds, not milliseconds

AI in the controller is already real

Controller vendors have brought inference into the PLC itself. Siemens introduced the SIMATIC S7-1500 TM NPU module in 2018: a neural processing module for the S7-1500 built around Intel's Movidius Myriad X, which runs a trained network on sensor or CPU data and hands the result back to the CPU program to evaluate.[1] Beckhoff's TwinCAT 3 Machine Learning runs trained models inside the TwinCAT real-time runtime, imported in the ONNX format, and Beckhoff cites response times under 100 microseconds for smaller networks.[2] These are vendor figures, and a plant should test any model on its own controller, but they show how far inference has moved toward the machine.

Keep the model portable

A model trained once should not be trapped on one vendor's hardware. **ONNX**, the Open Neural Network Exchange, is an open format for machine learning models launched in 2017 by Facebook and Microsoft and now a graduated project of the Linux Foundation's AI & Data foundation.[3] Models built in common training frameworks can be exported to ONNX and run on a wide range of runtimes and accelerators, including the controller platforms above. We prefer it for exactly the reason we prefer open architecture everywhere else: the plant should be free to change hardware without starting the AI over.

SECTION IV

AI Advises, the PLC Decides

Every model is wrong some of the time. The question is not whether it will make a mistake but what the mistake costs. On a plant floor, the answer has to be: very little, and never a person's safety. That is an architecture decision, and it is made before the first model is trained.

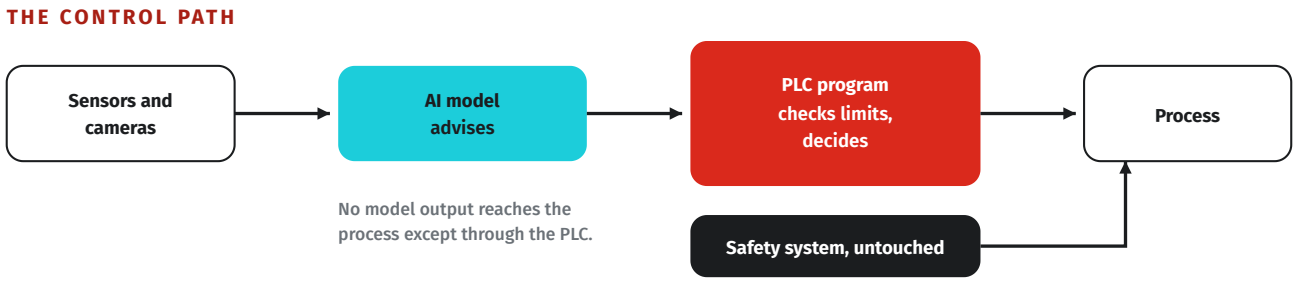

Figure 2. The model is an input. The PLC program evaluates it against limits it already enforces, and the safety system stays separate and deterministic.

Five rules we build to

1. **The model is an input, not an output.** A model's result reaches the process only as a value the PLC program reads, the same way the Siemens and Beckhoff designs above hand inference results to the control program.[1,2] The program decides what to do with it.
2. **The PLC enforces the limits.** If a model suggests a setpoint, the control program clamps it to the range the process engineer approved. A model can make the line better; it cannot push it outside its envelope.
3. **The safety function never depends on AI.** Guarding, light curtains, e-stops, safety PLCs, and their certified logic stay exactly as designed and validated. AI is never part of a safety function.
4. **There is always a fallback.** If the model is unavailable, slow, or reports low confidence, the line falls back to a defined behavior: run without it, divert parts for manual inspection, or slow down. That behavior is written down and tested at startup.
5. **An engineer signs off.** We treat a model the way we treat new PLC logic. Nothing goes to a running line until an engineer has reviewed it, tested it, and signed off, and the same goes for every update after that.

> **Shadow mode first.** Before a model is allowed to influence anything, it runs alongside the existing process and its decisions are recorded but not acted on. Comparing those decisions with what the inspectors and the process actually did is the cheapest, safest proof that a model is ready.

SECTION V

A Model Is Never Finished

A conventional machine behaves the same way on day one thousand as on day one, unless something breaks. A model can get worse without anything breaking: a new supplier's material looks slightly different, a light fixture is replaced, a product is added, or a machine wears in. That gradual mismatch between what the model learned and what the line now produces is called **drift**, and the only defense is a lifecycle that expects it.

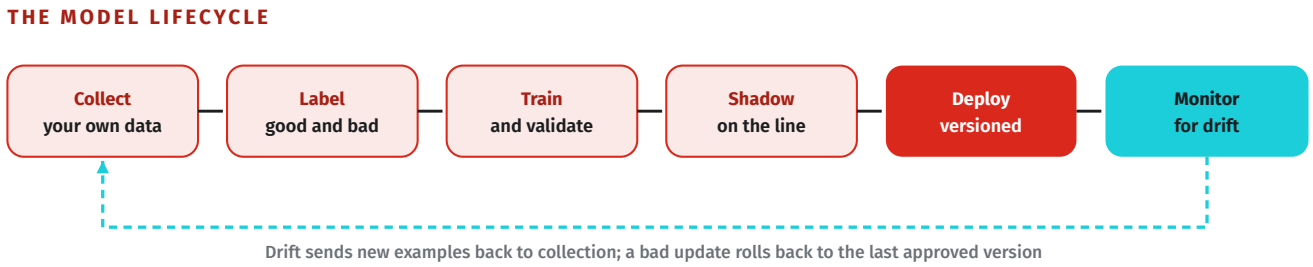

Figure 3. The lifecycle of a production model. Every step after deployment exists to catch the day the line changes and the model has not.

What the lifecycle needs

- **Versioned models.** Every model on the line has a version, a record of the data it was trained on, and the test results that approved it, the same way a PLC program has a revision history.
- **Rollback.** When an update performs worse, the previous approved version goes back on the line in minutes, not after a new project.
- **Confidence and drift monitoring.** The edge reports how confident the model is and how its inputs compare with the data it was trained on, so drift shows up on a dashboard before it shows up as scrap.
- **A feedback path.** Parts the model was unsure about, and parts a person overruled, are saved for review. They are the most valuable training data the plant will ever have.
- **A planned retrain.** Product launches, material changes, and lighting changes trigger a review of every model they could affect, the same way they trigger a review of the inspection plan.

This is also where the edge pays off a second time. Because inference runs at the line and only results and selected examples go upstream, retraining uses exactly the data that matters, and the plant never has to ship its full video history anywhere to keep a model accurate.

SECTION VI

Governance and Security

A model on the line is software that influences production, and it deserves the same scrutiny as the control system around it. Two frameworks help structure that scrutiny.

Manage AI risk on purpose

The NIST AI Risk Management Framework, version 1.0, released on January 26, 2023, is a voluntary framework for building trustworthiness into how AI is designed, developed, used, and evaluated.[4] It is organized around four functions, and each has a plain meaning on a plant floor:

FUNCTION	WHAT IT MEANS FOR A MODEL ON THE LINE
Govern	Someone owns each model, approves its deployment, and is accountable for its behavior; the sign-off rule in Section IV is part of this.
Map	Each model's purpose, inputs, the decision it influences, and what a wrong answer could cost are written down before it is built.
Measure	Accuracy is tested on held-out data from the real line, and confidence and drift are tracked after go-live.
Manage	Fallbacks, rollback, and retraining are in place, and incidents feed back into the next version.

Secure the edge like any other control-room computer

Edge AI hardware sits next to the control network, so it belongs inside the same security model as everything else there. The ISA/IEC 62443 series is the reference most plants already use.[5] In practice:

- **Segment it.** AI hardware lives in a defined zone with controlled conduits to the PLCs it reads and the systems it reports to, not on a flat network.
- **Outbound only.** Results and selected examples go out of the plant on connections the edge opens; nothing outside opens a path in.
- **Sign and verify models.** A model is code that influences production. Updates are signed, verified before they load, and logged, so nobody can swap one in unnoticed.
- **Least privilege.** The model gets read access to the data it needs and nothing else. Any write it influences goes through the PLC, as described in Section IV.

Questions to ask anyone who deploys AI on your equipment

1. Where does inference run, and what happens to the line when the network is down?
2. What data leaves the plant, and who can see it?
3. What does the PLC do when the model is wrong, slow, or unavailable?
4. Is the model in an open format we can move to other hardware?
5. How will we know when accuracy drifts, and how fast can we roll back?
6. Who signs off before a model or an update touches the line?

SECTION VII

Getting Started, and Building It with Fireball

The plants that get value from AI fastest do not start with a platform. They start with one problem, one line, and one number.

1. **Pick a cost.** An escape, a scrap rate, an hour of downtime, or a manual inspection station, with a dollar figure on it.
2. **Measure the baseline.** How often it happens today and what it costs, so success has a definition before work begins.
3. **Collect your own data.** Images or signals from the line, labeled by the people who know what good and bad look like.
4. **Run in shadow mode.** The model watches and records, and its decisions are compared with reality.
5. **Go live with limits.** The PLC evaluates the model's output, fallbacks are tested, and an engineer signs off.
6. **Prove it and scale.** Compare against the baseline, then take the same approach to the next line or the next problem.

What Fireball brings

Fireball Industries builds AI for the plant floor as part of the same engineering services we bring to controls, robotics, and integration. Because we build the controls too, the model, the PLC logic around it, and the fallback are designed together by one team. Our AI and machine learning work covers:

- **Industrial and machine vision**
- **Predictive maintenance and anomaly detection**
- **Vision-guided robotics and defect detection**
- **Digital twins with AI feedback loops**
- **Edge AI deployment on PLCs and PCs**

We are vendor agnostic and work with the platforms plants already run, including Rockwell Automation, Siemens, CODESYS, Cognex, and Keyence, and we deliver models in open formats so the plant is never locked into our tools or anyone else's.

Bring us one problem with a number on it.

We will tell you whether AI is the right tool for it, where the model should run, and what the PLC does when the model is wrong.

sales@fireballz.ai | fireballz.ai/get-a-quote

METHODOLOGY AND REFERENCES

How the Numbers Were Built

Time per part. 60 seconds divided by 600 parts is 0.1 seconds, or 100 milliseconds per part. It is an example rate chosen to show the scale of the budget, not a measurement of any particular line.

Camera bandwidth. A 5 megapixel monochrome sensor at 8 bits per pixel produces 5 million bytes per frame. At 30 frames a second that is 150 million bytes, or 1.2 billion bits, per second, before compression. Ten such cameras produce 12 gigabits per second. Real cameras vary in resolution, bit depth, and frame rate; the figure shows the order of magnitude.

Zero production decisions waiting on the internet. This is the design principle the paper recommends, not a measurement.

Vendor figures. The Siemens and Beckhoff details in Section III come from the vendors' own announcements and product material, cited below, and have not been independently measured by us.

1. Siemens AG. "Artificial intelligence for Simatic" (SIMATIC S7-1500 TM NPU, Intel Movidius Myriad X), press release, November 2018. press.siemens.com/global/en/pressrelease/artificial-intelligence-simatic
2. Beckhoff Automation. "TwinCAT 3: Machine learning for all areas of automation," April 2019, and TwinCAT 3 Machine Learning product information. beckhoff.com/en-en/company/press/twincat-3-machine-learning-for-all-areas-of-automation-2019-04.html
3. LF AI Foundation. ONNX becomes an LF AI graduate project, 14 November 2019; ONNX project site. onnx.ai
4. National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, released 26 January 2023. nist.gov/itl/ai-risk-management-framework
5. ISA/IEC 62443 series, *Security for industrial automation and control systems*, zones and conduits model.

CONTACT

Talk to an engineer.

Fireball Industries builds standard automation products and the engineering services behind them for manufacturers across the United States and Mexico, from our headquarters in Macedonia, Ohio, our engineering office in Yuma, Arizona, and our operations in Zacatecas, Mexico.

Email
sales@fireballz.ai

Web
fireballz.ai

Get a quote
fireballz.ai/get-a-quote

More white papers
fireballz.ai/resources

© 2026 Fireball Industries LLC. All rights reserved. Siemens, SIMATIC, Beckhoff, TwinCAT, Intel, Movidius, ONNX, and the other product and standard names in this paper belong to their owners and are used here only to describe compatibility and the state of the industry.