LF EDGE

Below 15 Microseconds: Real-Time Edge AI on Open Horizon and FDO

EdgeBrain: Decentralized AI for Industrial Edge, Built on LF Edge

LF Edge Project(s) Used:
Open Horizon
FDO
(FIDO Device Onboard)

50 sensors
on one shared time base

<15 µs
acquisition jitter

<10 MB
models

15+ year
production-line service life

Organization Overview

de Jonge GmbH is a deep tech hardware company founded in Nuremberg, Germany, in 2024. The company builds EdgeBrain, an industrial edge AI module built on programmable hardware, designed for retrofit into production lines already in operation and for direct integration by machine builders.

The starting point is an observation rather than a technology: every machine has a heartbeat. Vibration, current draw, and acoustics change the moment something starts to wear. Most operations don't hear it in time, not until a batch is scrap or a line has stopped. The signals are usually already present, on a bus the machine has carried for years. What's missing is a way to read them at the resolution the process actually happens at, without touching the machine.

EdgeBrain attaches to that existing data path, evaluates it locally, and turns raw signals into process information on the machine itself. Operation is local and offline-first; an encrypted connection is optional rather than a precondition, which matters in plants where production data is not permitted to leave the site.

Series manufacturing runs at a contract manufacturer in Germany. de Jonge is an IBM Embedded Solution Partner and an active participant in the **LF Edge Open Horizon project** ecosystem.

The Problem

Edge computing is usually argued over bandwidth and cost. In manufacturing, the harder constraint is time, and not in the way it's normally framed.

Take injection molding, where a plastic part is formed by forcing melt into a mold under pressure. The cycle runs about twenty seconds, but what determines whether the part is good doesn't stretch across those twenty seconds. The switchover from injection to holding pressure is set on cavity pressure within a window measured in single milliseconds, and the signatures that announce a drifting process, a worn check valve, or a cavity filling unevenly play out on that same scale. Sampled once per cycle, none of this exists. The measurement reports a good part, and by the time anyone knows otherwise, the shot is already scrap.

The reason these patterns rarely reach an analysis isn't bandwidth, it's that they're destroyed before transport. Vibration, load, and speed usually travel separate acquisition paths, each with its own clock, its own sampling interval, and its own aggregation step, and the values are commonly averaged to the cycle before they're ever stored. What arrives at the central system is three time series that no longer share a time base. A vibration pattern means nothing without matching load and speed values, and if their relative timing is uncertain by even a few milliseconds, the correlation that carried the information is gone, not filtered out, not too small to see. Gone, and indistinguishable from noise.

"This is the part that doesn't improve with a faster link or a larger cluster. Whatever is lost at acquisition cannot be recovered downstream," said Dr. Andreas de Jonge, CEO of de Jonge

The Solution/Technical Approach

EdgeBrain taps the data stream between the field level and the host without becoming part of it, so the machine, its controller, and its safety architecture stay untouched. Nothing is inserted into the control path, which is what makes the module acceptable in a brownfield plant, where any modification would trigger a re-qualification of the machine.

On one machine, EdgeBrain currently follows fifty digital sensors in parallel, with bandwidth to spare. The fieldbus provides the deterministic base and carries everything the machine already has on it, including signals an IO-Link master aggregates from the sensor level. All of it is timestamped against a single hardware clock running at 300 MHz, with acquisition jitter below 15 µs in the bare-metal configuration. The bus cycle limits how often a value is seen; the jitter limits how reliably two values can be placed against each other, and it's the jitter that decides whether a pattern survives. The fifty signals share one time base, so a correlation between them is a property of the process rather than an artefact of the measurement.

MQTT, OPC UA, and REST contribute context, placed on the same timeline but never treated as a real-time source. Signals are fused and time-aligned at the point of origin, then turned into meaningful process information before anything leaves the machine. Models stay small because they're fed correlated features rather than raw streams, and production data never leaves the shop floor.

50
sensors

300 MHz
clock

<15 µs
jitter

The octopus principle

de Jonge calls the resulting split the octopus principle. The octopus doesn't solve the problem by having no brain, it keeps two-thirds of its neurons in

its arms, so a tentacle reacts in milliseconds without asking, while the central brain does what a brain is for: learning, coordinating, deciding what to do next. Neither part does the other's work.

EdgeBrain splits the same way. What happens in milliseconds is measured and evaluated where it happens, on a time base that holds. Everything that gets better with context and time goes to the central system: trend analysis across machines and sites, correlation between lines, model training, fleet management, and the deployment of updated models back to the edge. The central system gets better data this way, because it receives evaluated events instead of a raw stream nobody can afford to transport.

Three layers, separated in hardware

The module is built in three layers. The hardware layer carries the acquisition path and the model. The firmware runs alongside it, and the two are combined into one image while remaining independently updatable. Linux runs on a processor core of its own, with its own memory device, and serves as the management layer, hosting the Open Horizon agent.

The firmware is bare metal or FreeRTOS, chosen per use case. Where timing carries the argument, it runs bare metal, the configuration the 15 µs jitter figure refers to. Where an application needs concurrent tasks more than it needs the last few microseconds, FreeRTOS is the better trade, and the figure moves accordingly. Both sit below the management layer either way, so neither depends on Linux to do its work.

LF EDGE PROJECT ROLE

Open Horizon hosts the device's management layer, running on its own isolated Linux core.

The separation is physical rather than a matter of process isolation, and it's what produces a clean IT/OT boundary. The management layer is the part that faces the network, and therefore the part that carries an attack surface. If Linux is compromised or damaged, the hardware layer and the firmware keep running, and the machine keeps being watched. The layer that talks to the outside world cannot take down the layer that does the work, a property a plant engineer will ask about before an IT architect does.

Hardware and software

EdgeBrain	Specification
PLATFORM	Efinix Ti375 FPGA SoC
ACQUISITION JITTER	Below 15 µs (bare-metal firmware); firmware is bare metal or FreeRTOS depending on use case
PROTOCOLS	Raw fieldbus, MQTT, OPC UA, REST
DEPLOYMENT	Passive tap between field level and host, no interruption of the control path
POWER	Below 15 W at 5 V DC
ENCLOSURE	Full aluminium, passive cooling, 90 × 75 × 25.5 mm, 300 g
OPERATING RANGE	−40 °C to +70 °C
MODELS	TensorFlow, written into programmable hardware. Classical, neural, or hybrid, typically under 10 MB. Activated on a customer-released reboot
ARCHITECTURE	Chip image and firmware as one bitstream, independently updatable. Linux management layer on a separated RISC-V core with its own memory device
UPDATE ARTEFACT	Containers carrying chip image, firmware binary, or Linux layer, individually or together
MANAGEMENT	Open Horizon agent on the Linux layer; FDO onboarding

How the models are built and run

Models are built in TensorFlow, then written into the programmable hardware itself rather than executed as software on a processor. The evaluation happens in the circuit, with no inference runtime and no operating system in the path between signal and result, so timing is a property of the hardware, not of load on a processor. Depending on the task, de Jonge deploys classical methods, neural networks, or hybrids of the two, and a model is typically under 10 MB. The size isn't a constraint worked around, it follows from what the device is fed. A model that receives correlated features on a common time base doesn't need the capacity of one that has to find structure in raw streams first.

Training runs on de Jonge's own infrastructure, and on customer premises where production data isn't permitted to leave the plant. In most cases, no historical archive is required, the model learns the signature of the individual machine from the signals that machine produces, which turns a deployment into a matter of days rather than a project waiting on a data lake.

LF EDGE PROJECT ROLE

Open Horizon distributes model, firmware, and Linux-layer updates to the installed fleet; every update still waits on a customer-released reboot.

Updated models are distributed to the installed base through Open Horizon. Because the model is part of the hardware configuration, a new one becomes active on reboot, and the reboot is released by the customer. The change therefore lands in a planned stop rather than in a running process, nothing goes live on a machine without the plant deciding it should.

Why LF Edge Projects

Decentralized intelligence solves the timing problem and creates an operational one. Workloads spread across many machines, sites, and customers still have to be deployed, updated, and audited reliably. Open Horizon covers that for de Jonge, and FDO handles onboarding, which matters because these devices are commissioned by machine operators in brownfield environments rather than by IT staff. The person plugging in the module is often the same person running the press.

"The person plugging in the module is often the same person running the press."

LF EDGE PROJECT ROLE

Open Horizon = fleet deployment, updates, and audit across machines, sites, and customers. FDO = zero-touch onboarding for operators without IT involvement.

Open Horizon deploys containers for de Jonge, as it would for any other workload. What the containers carry isn't an application but the components of a device update: the chip image, the firmware binary, the Linux layer, individually or together. They run on the Linux management layer, place each component where it belongs, and the new configuration becomes active on the reboot the customer releases. The model is part of the hardware configuration, so a model update is a piece of hardware that travels the standard software path, with the versioning and the record of what's running where that comes with it. Because the agent is separated in hardware from the layers it manages, an update in progress cannot interfere with acquisition. Keeping the model in hardware and its lifecycle in a standard, vendor-neutral management layer is what lets de Jonge have determinism at the machine and fleet management across sites at the same time.

Which deployment strategy fits best is still being worked out against the pilot installations. Pattern-based deployment matches the current phase, where a known set of machines receives a known configuration. Policy-based deployment is the more interesting fit for where this is going, because the properties that should decide which model a device receives are machine properties: the bus it sits on, the sensors it sees, the process it runs. Getting that mapping right is worth more to de Jonge than deploying quickly, so both are being evaluated before committing.

The same applies to onboarding. FDO is the right mechanism for devices commissioned by machine operators, and the open question isn't the protocol but the ownership chain. A module passes from the contract manufacturer through de Jonge, in some cases through a machine builder, and only then to the plant that will own it. Deciding where the ownership voucher is created and handed on, and who operates the rendezvous service, determines whether the operator can commission the device without ever involving their IT department. That's the outcome de Jonge is designing for.

Why Open Source/Vendor-Neutral Foundation

The open foundation was a deliberate choice. Production lines stay in service for fifteen years and longer, and de Jonge's customers are rightly cautious about tying infrastructure on that timescale to a single vendor's platform. A vendor-neutral management layer is what makes decentralized intelligence acceptable to them at all.

The argument is sharper in this segment than in most. A mid-sized manufacturer will not accept a dependency whose commercial terms can change while the machine it manages is still halfway through its service life. An open, foundation-governed management layer removes that objection before it's raised, and it lets the customer keep operating the fleet even if the relationship with the supplier ends.

LF EDGE PROJECT ROLE

LF Edge governance is the reason customers will accept a 15-year infrastructure dependency in the first place.

Results/Impact

The figures that can be published at this point are properties of the system rather than of any one installation. In the pilot configuration, the module follows fifty digital sensors on a single machine in parallel, on one shared time base, with acquisition jitter below 15 µs and bandwidth still in reserve. Models reach the device through Open Horizon and become active on a reboot the plant releases, keeping every change inside a planned stop.

Process outcomes from the pilot sites, scrap rate and tool utilization among them, are covered by customer confidentiality and are not published here.

50 sensors	<15 µs	<10 MB	15+ year
on one shared time base	acquisition jitter	models	production-line service life

Community Engagement

de Jonge participates in the LF Edge Open Horizon project ecosystem and is an IBM Embedded Solution Partner. The company is currently evaluating Open Horizon's pattern-based and policy-based deployment models against its pilot installations, and is working through FDO ownership-voucher and rendezvous-service design as part of its onboarding architecture, both areas where deeper engagement with the respective working groups could be mutually useful.

Next Steps

Pilot programmes convert into rollouts across further machines during 2027. In parallel, integration by machine builders is the primary growth path, with EdgeBrain shipped as part of the machine itself rather than retrofitted to it.