EON AI VENTURES · THE DISTILLERY

EON Delta

Owning the Trained AI Model

How we train our own models — and why the trained model, not the base model, is the asset.

Part 1 Plain English **Part 2** The Theory

First deployment: Field IQ Edge

1

Plain English

What we are building, with no mathematics.

THE PROBLEM

A genius who has never been in your plant

Hire the smartest 22-year-old in the world. Drop them into a gas facility on Monday and say “keep this safe.” They will be useless — possibly dangerous.

Not because they are stupid. Because they have no map, no rulebook, nobody checking their homework, and no way to learn whether what they just said was right.

A raw model is raw capability. Everything that turns capability into a reliable worker sits outside the model.

No map

It has never seen your equipment, tags, or layout.

No rulebook

Your procedures and standards are not in it.

No scoreboard

It cannot find out whether it was right.

This is the gap that scaffolding — and a trained model — exists to close.

THE CORE IDEA

The professor and the apprentice

You cannot put the professor in a backpack. So you don't.

P The professor

- Brilliant, general, expensive
- Needs a building full of GPUs
- Answers thousands of questions once

Frontier / large open-weight model

→

A The apprentice

- Studies only the checked answers
- Learns one job, extremely well
- Fits in a backpack, runs offline

Small distilled model + our adapter

The apprentice ends up as good as the professor at that one job — because it only ever had to learn one job.

THE QUESTION EVERYONE GETS WRONG

Knowledge can live in three places

1

THE WEIGHTS

Their brain from school

Everything baked in before they met you. Changing this is fine-tuning.

Stores: Behaviour and judgment

2

THE CONTEXT WINDOW

What's on their desk

The drawing you just handed them. Gone tomorrow.

Stores: This conversation only

3

THE DATABASE

The filing cabinet

Every procedure, incident, sensor log — looked up on demand.

Stores: All facts

The rule: facts go in the database. Judgment goes in the weights. Never put facts in the weights — they change, you cannot explain them, and you cannot delete them.

Three reasons, each one fatal on its own

1

Facts change

A plant adds a valve.

Database: edit one row, one second.

Weights: retrain the model.

2

You must explain

A regulator asks “why did it say that?”

Database: point at the document, page 14.

Weights: the fact is smeared across a trillion numbers.

3

You must delete

A customer leaves and says “remove our data.”

Database: DELETE.

Weights: you cannot. It is cooked in.

Any one of these ends an energy-sector procurement conversation. Together they are decisive.

THE MECHANISM

What a LoRA actually is

You do not rewrite the model. You freeze it and bolt a small adapter on the side.

BASE MODEL — FROZEN

8 billion parameters. Downloadable by anyone, including your competitors. Never modified.

Commodity. Same file everybody else has.

OUR ADAPTER — THE EON DELTA

~30 million trained parameters. A few hundred megabytes. Built from our verified data. Nobody else has it, and nobody can make it without doing our work.

0.4%	of parameters actually trained
~300 MB	adapter file size on disk
$300–900	cost of one training run
4–8 hrs	on a single rented GPU

Cheap enough to run monthly. Small enough to ship one per customer.

THE MECHANISM, IN WORDS

LoRA = Low-Rank Adaptation

Adaptation

You are adapting a model that already exists, rather than building one. That is the easy part.

Low-Rank

"Rank" measures how much genuinely independent information sits in a grid of numbers. A very large grid can have a low rank — meaning most of it repeats, and it can be rebuilt from far fewer numbers.

The insight: the change needed to specialise a model is low-rank. Not a complicated change — a simple one applied in a great many places.

Write it LoRA. "LORA" collides with LoRaWAN, an unrelated radio standard our pipeline and utility customers meet constantly.

The everyday version

A 500-page book that is the same paragraph repeated with small variations does not need 500 pages of storage. Store the paragraph and the variations. Same information, a fraction of the space.

16,777,216	numbers in one full grid
131,072	numbers in the two thin strips
0.78%	of that grid actually stored

This one mathematical fact is why an adapter is 300 MB and not a terabyte, why training costs hundreds and not hundreds of thousands, and why we can hold one adapter per customer.

THE 80% OF THE WORK

Where the training examples come from

1

Records we already have — like Situation Room

Past inspection reports, work orders, incident write-ups. Input is the situation; output is what the expert actually wrote.

Free. Already proven correct by reality.

2

Professor-generated, engineer-checked

The big model drafts thousands of pairs from our documents; engineers correct them. Correcting is five times faster than authoring.

This is how we reach volume.

3

Digital-twin generated

Run the scenario in the twin, capture situation → correct action.

Nobody else on earth can produce these.

Target: 500–1,000 verified pairs to start. Quality crushes quantity — 500 excellent examples beat 50,000 sloppy ones.

Green, yellow, and the third colour

GREEN

Verified fact

Timestamped sequence, measurements, physical mechanism, corrective action taken.

KEEP

YELLOW

Narrative and drama

Blame, defensiveness, corporate hedging, speculation about motive.

DISCARD

NEW

Knowable at the time

Separate what was known before the event from what was only learned after it.

LABEL SEPARATELY

Why the third colour matters. Post-incident reports suffer from hindsight bias — they make the cause look obvious. Train on the after-the-fact framing and the model becomes confidently certain about things that were genuinely ambiguous. Train only on what was knowable before the event, labelled with what happened after.

DO THIS FIRST — IT COSTS NO TRAINING RUN

Use the incidents as the exam, not the textbook

Hundreds of documented failures is small for training. It is perfect as a benchmark.

Turn each incident into a test: given only what was known 48 hours before this event, does the system flag it?

> **“Tested against 300 of your own historical incidents, our system flagged 214 of them in advance.”**
>
> *Illustrative. The sentence no competitor can say.*

Customer-specific

Scored on their own history, which they already accept as true.

Zero training cost

It is an evaluation, not a model.

Becomes the gate

Nothing ships until it beats this number.

Order of work: benchmark → pre-mortem product → training set

THE TRAP THAT KILLS WARNING SYSTEMS

Train only on disasters and it will cry wolf

A model that has only ever seen things going wrong believes everything is about to explode.

- Over-warning produces alarm fatigue.
- Alarm fatigue means nobody reads the alarms.
- A warning system nobody reads is worse than none — it manufactures false confidence.

The fix. Include normal days — cases where conditions looked similar and nothing happened — at a realistic ratio. This is the single biggest technical trap in the whole project.

The arithmetic

1 in 10,000

true incident rate per inspection-day

99%

detector accuracy — sounds excellent

~100

false alarms for every real catch

~1%

of alarms are real

Accuracy is the wrong metric for rare events. Full derivation in Part 2.

WHAT IT TAKES

Eight weeks and a few thousand dollars

Weeks	Work	Output	Cost
1–2	Define the one job. Write the exam.	Job statement + 150 held-out test cases	Internal time
3–6	Build 1,000 checked examples. Green/yellow pass.	Verified training set (JSONL)	Internal time — the real cost
7	LoRA fine-tune, quantize, score against the exam.	Adapter + benchmark score	$300–900 GPU rental
8	Deploy to hardware. Walk the site with it.	Working Field IQ Edge prototype	Device cost

The bottleneck is never the training.
It is whether we will do the unglamorous dataset work in weeks 3–6. That is the whole ballgame.

Two months to a real answer, on a few thousand dollars.
The risk is organisational, not technical.

ARCHITECTURE PLACEMENT

Distillery is a factory. Field IQ Edge is a destination.

What we have described — verified data in, trained model out — is manufacturing, not deployment. Build it once, inside Distillery.

DISTILLERY

The factory

Ingests verified data. Runs the green/yellow pass, the training run, and the benchmark.

→

EON DELTA

The asset

Base Delta (ours) + Tenant Delta (theirs), each with its dataset and benchmark score.

→

FIELD IQ EDGE

The destination

Where a Delta gets deployed. First customer of the factory — not the owner of it.

One factory, two product lines.
Small edge adapters (cheap, offline, narrow) are the learning vehicle. Large sovereign adapters on a big open-weight base, deployed on the customer's own hardware, are the high-value line. Same pipeline, same data discipline, different output size — which means line one de-risks line two.

WHAT WE OWN, WHAT THEY OWN

Every Delta has two halves

The split is not a compromise. It is what makes the data rights standard executable in a model.

TENANT DELTA

The customer's half

- Trained inside their tenant, on their records
- Knows their asset tags, sites, local procedures
- Never leaves the building
- Returned or destroyed if they leave us

Their property — a Derived Insight (B.3.1, B.5.6)

BASE DELTA

EON's half

- Trained only on de-identified patterns
- Nothing enters unless seen at two or more unaffiliated organisations
- Survives termination
- Licensed back to every participant free, in perpetuity

EON property — a Generalized Learning (B.4.1)

On the device they stack. The Base Delta supplies judgment learned across an entire industry; the Tenant Delta supplies everything specific to the site. Their half never leaves. Ours improves every month from every sector we serve.

Three tiers decide who owns what

Tier	Contents	Ownership
1 · Customer Data	Raw capture; personnel identities and records; site, asset and unit identifiers; equipment configurations; their SOPs, permits and incident reports as written	Customer. Stays in their tenant. Not licensed to EON for any purpose, including model training.
2 · Derived Insights	Site-level analytics; named-crew competency assessment; facility-specific risk and maintenance models — anything attributable to a location, asset or person	Customer.
3 · Generalized Learnings	Failure modes and precursor signatures by equipment class; procedural deviation patterns; human-error taxonomies; model weights derived from these	EON, subject to the four conditions below. Licensed back to every participant, royalty-free, in perpetuity.

A pattern qualifies for Tier 3 only if all four hold

Multi-source

Seen at two or more unaffiliated organisations

Non-attributable

No participant, site, asset or person recoverable

Outside the list

Competitively sensitive categories excluded by definition

Reviewed

Thirty-day customer window with a right to withhold

WHY THE MODEL FORM MATTERS

We promised both. Only a model does both.

Give them everything

Every participant receives the entire body of generalized learnings — including patterns contributed by others — royalty-free and in perpetuity.

Never sell the corpus

We sell the model and the product trained on it. The moment EON is a data broker, the safety framing collapses and every operator gets nervous.

A data file is a pipe. A model is a teacher.

A file

Can be read line by line, searched, cross-referenced and copied. Someone who knows the industry looks at an “anonymous” entry and names the plant. Once it is out, it is out forever.

A model

Answers questions about the situation in front of a technician. There is no list inside it to flip through — and you can test whether it leaks.

That is the difference between becoming the channel between competitors and simply teaching both of them.

THE STRATEGIC POINT

We do not own the brain. We own the pattern.

The base weights are the same file a competitor can download this afternoon. What they cannot copy is the Base Delta — the de-identified judgment of an entire industry, gathered under a standard operators will actually sign, and provable against their own history. Copying it means redoing years of real work inside real plants, with their permission.

Cheaper, better models make our patterns worth more — not less.

Base model

Commodity — anyone can download it

Tenant Delta

The customer's — never leaves their tenant

Base Delta

Ours — survives termination

The benchmark

Ours — how we sell, and how we gate

2

The Theory

Weights, objectives, low-rank decomposition, quantization, and the metrics that actually matter.

FOUNDATIONS

What a “weight” actually is

A **transformer is a stack of layers**. Each layer is mostly **matrix multiplication**. A weight is one number in one of those matrices.

```
y = W x + b              W ∈ ℝ^(d_out × d_in)

Attention per layer:     W_q , W_k , W_v , W_o
MLP per layer:           W_gate , W_up , W_down
```

For an 8B model with hidden dimension **d = 4096 and 32 layers**, each attention projection alone is 4096 × 4096 = 16.78 M parameters. **“Training” means adjusting those numbers so predictions improve.**

Pre-training sets all of them by reading the internet: ~10^{25} FLOPs, months, hundreds of millions of dollars. We never do this.

Scale, concretely

d_model	4,096
Layers	32
Attn matrices / layer	4
MLP matrices / layer	3
Total parameters	~8×10^9
FP16 size	~16 GB
Bytes per parameter	2 (FP16)

FOUNDATIONS

The objective: make the wanted answer more likely

Supervised fine-tuning is next-token prediction on our own input→output pairs. The loss measures surprise; gradient descent reduces it.

Token-level cross-entropy over a target sequence y given prompt x:

$$L(\theta) = -\Sigma_t \log P(y_t \mid y_{<t}, x; \theta)$$

Gradient step, learning rate η:

$$\theta \leftarrow \theta - \eta \cdot \nabla_\theta L(\theta) \qquad \text{(AdamW in practice)}$$

Loss is masked on the prompt

Only tokens in the desired output contribute to L. The model is not trained to reproduce the question.

Typical hyperparameters

η ≈ 1e-4 to 2e-4 for LoRA, 2–4 epochs, effective batch 8–32, cosine decay with warmup.

Overfitting is the failure mode

With 1,000 examples, more than ~4 epochs memorises. Watch held-out loss, not training loss.

LoRA: the update is low-rank, so store only the factors

Hypothesis: the weight change needed to specialise a model has far lower intrinsic rank than the weight matrix itself. So do not store ΔW — store two thin matrices whose product is ΔW.

$$h = W_0 x + \Delta W x = W_0 x + (\alpha / r) \cdot B \cdot A \cdot x$$

$$W_0 \in \mathbb{R}^{d \times k} \text{ frozen} \qquad A \in \mathbb{R}^{r \times k} \qquad B \in \mathbb{R}^{d \times r} \qquad r \ll \min(d,k)$$

$$\text{Init: } A \sim \mathcal{N}(0, \sigma^2),\ B = 0 \Rightarrow \Delta W = 0 \text{ at step } 0 \text{ (training starts from the base model)}$$

$$\text{Trainable parameters: } r(d + k) \text{ versus } d \cdot k \text{ for a full update}$$

$$\text{After training, merge: } W = W_0 + (\alpha/r)\, B\, A \Rightarrow \text{zero added inference latency}$$

Rank r

8–64. Higher r = more capacity and more overfitting risk. r = 16 is a sound default for behaviour tuning.

Scaling α

Decouples effective step size from r. Keeping α/r constant makes runs comparable across ranks.

Where to apply

Attention q,k,v,o first; add the MLP projections when the task needs more expressive capacity.

PARAMETER ARITHMETIC

Why 0.4% of the parameters is enough

Quantity	Expression	Value
One attention projection, full	d × k = 4096 × 4096	16,777,216
Same projection, LoRA r = 16	r(d + k) = 16 × 8192	131,072
Share of that matrix trained	r(d+k) / dk	0.78%
Adapter over q,k,v,o × 32 layers	4 × 32 × 131,072	~16.8 M
Adding MLP projections	7 × 32 × r(d + k_mlp)	~30 M
Share of the full 8B model	30 M / 8 B	~0.4%

Figures are illustrative arithmetic for a d = 4096, 32-layer model; MLP widths differ by architecture.

What follows from this

- A ~300 MB file, not a terabyte.
- One adapter per customer and per task, swapped over one shared base.
- Hours on one GPU, not weeks on a cluster.
- Fully reversible — the base is untouched.
- Re-trainable monthly as the dataset grows.

The adapter is the unit we ship, price, and version.

WHY THIS IS AFFORDABLE

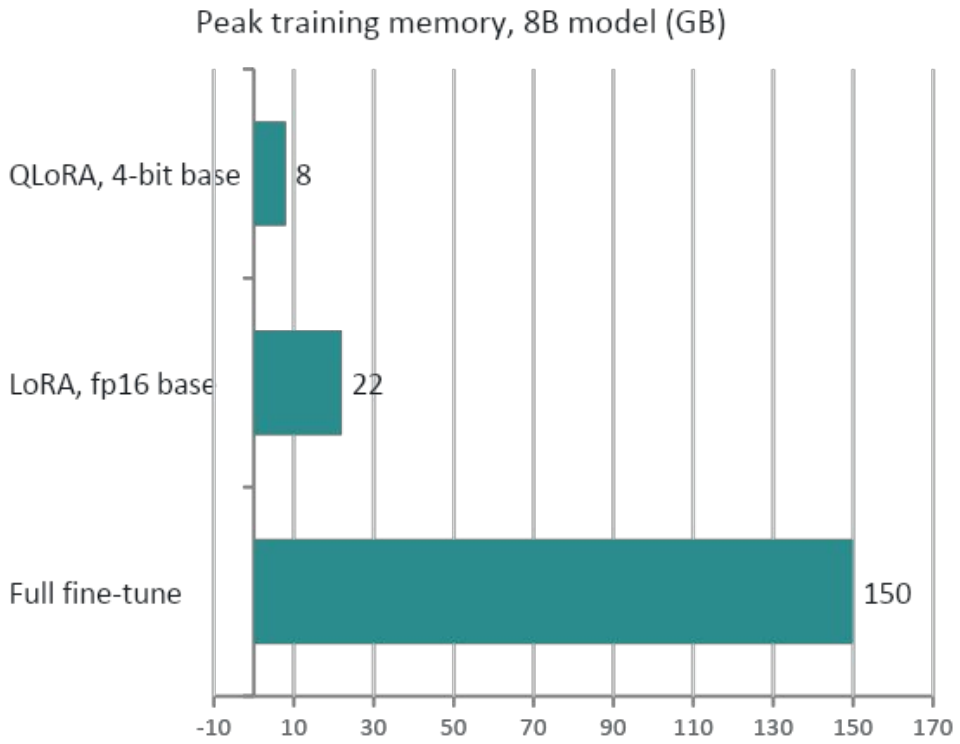
Memory arithmetic decides what hardware you need

```
Full fine-tune, AdamW:
  params(fp16) + master(fp32) + 2 moments(fp32)
  ≈ 16–20 bytes/param  →  8B ≈ 128–160 GB

LoRA on a 4-bit frozen base (QLoRA):
  base ≈ 0.5 B/param ≈ 4.5 GB
  + optimizer states on ~30 M trainable only
  ≈ 6–10 GB total  →  one 24 GB GPU
```

Gradients and optimizer state — not the weights — dominate training memory. Freezing the base removes almost all of it. That single fact is why a $300 run replaces a $300,000 one.

Gradient checkpointing trades ~30% speed for a further large memory reduction when needed.



Approximate. Exact figures vary with sequence length, batch size and kernel implementation.

Quantization: round the dials, keep the behaviour

Weights are stored as high-precision numbers. Quantization stores them with fewer bits — the model shrinks roughly linearly and degrades far less than linearly.

```
size(bytes)  =  N_params × bits / 8

Affine quantization of a weight block:
   w ≈ s · q + z ,      q ∈ {0 … 2^b − 1}
   s = (w_max − w_min) / (2^b − 1)          (scale)
   z = zero-point                            (offset)

Group-wise scales (e.g. 64–128 weights per scale) preserve
accuracy far better than one scale per tensor.
```

Precision	8B size	Quality
FP16 / BF16	~16 GB	Reference
INT8	~8 GB	Near-identical
INT4 (Q4_K_M)	~4.5 GB	Small, usually acceptable loss
INT3 and below	~3.5 GB	Noticeable degradation

Two distinct uses.
QLoRA quantizes the base during training to fit one GPU. Post-training quantization shrinks the merged model for the device. Always re-run the benchmark after quantizing — never assume the score survived.

TEACHER TO STUDENT

Distillation, formally

Classical distillation matches the student's output distribution to the teacher's. With API teachers we cannot see logits, so we distil on verified sequences instead.

```
Logit-level (Hinton) — requires access to teacher logits z_T:

   p_i^(T)  =  softmax( z_i / T )                    T = temperature
   L  =  α · L_CE(y, p_S)  +  (1 − α) · T² · KL( p_T^(T) ‖ p_S^(T) )

Sequence-level (what we will actually do):

   1.  Teacher generates candidate outputs ŷ for each input x
   2.  Engineer or twin verifies / corrects  →  y*
   3.  Student trains with plain cross-entropy on the verified pairs (x, y*)
```

The T^2 term

Rescales gradients so soft-target magnitude does not shrink as temperature rises.

Why sequence-level wins here

Verification is the value-add. A corrected answer carries our judgment; a teacher logit does not.

Rejection sampling

Generate k candidates, keep only those passing the twin or a rule check. Cheap way to raise dataset quality.

THE METRIC TRAP, DERIVED

Why 99% accurate is worthless for rare events

Let $\pi = P(\text{incident}) = 10^{-4}$ per inspection-day

sensitivity $P(+|I) = 0.99$

specificity $P(-|\neg I) = 0.99$

Per 10,000 inspection-days:

$TP = 10{,}000 \cdot \pi \cdot 0.99 \approx 0.99$

$FP = 10{,}000 \cdot (1-\pi) \cdot 0.01 \approx 99.99$

$\text{Precision} = TP / (TP + FP) \approx 0.0098 \rightarrow \sim 1\%$

Roughly one hundred false alarms per genuine catch. The detector is “99% accurate” and operationally useless.

Bayes makes this unavoidable: when π is tiny, precision is dominated by the false-positive rate, not by accuracy. No amount of model quality escapes it — only a change of objective does.

What to do instead

1. **Rank, don't alarm**

   Return the top-k highest-risk assets per shift. Bounded human cost.

2. **Let it abstain**

   “Insufficient data” is a valid, valuable output.

3. **Report precision@k**

   Not accuracy. Ever.

4. **Calibrate**

   A stated 30% risk must be right ~30% of the time.

MEASUREMENT

The metrics that decide whether we ship

Metric	Definition	Why it matters here
Teacher-relative score	student score ÷ professor score on the held-out set	Ship at 85–95%. Below that, add examples — not parameters.
Precision@k	fraction of the top-k flagged items that are genuine	Matches how a crew actually consumes output: a bounded worklist.
Recall on the incident set	share of historical incidents flagged from pre-event data only	The customer-facing headline number.
Brier score	$BS = (1/N) \Sigma (p_i - o_i)^2$	Calibration. Punishes confident wrongness, which is the dangerous failure.
Abstention rate	share of cases returning “insufficient data”	Should be non-zero. Zero means it is guessing.

The held-out set is written before the first training run and never touched again. Tune on it once and every number above becomes fiction.

KNOWN FAILURE MODES

What goes wrong, and the mitigation

FAILURE MODE	WHAT HAPPENS	MITIGATION
Catastrophic forgetting	Tuning hard on a narrow task degrades everything else — including instruction-following and refusal behaviour.	Low rank, low η, few epochs. Keep 5–10% general-purpose examples in the mix. Adapters isolate damage: unload and the base is intact.
Hindsight leakage	Any post-event field in the input teaches the model to read the answer off the label.	Enforce a temporal cutoff per example: condition only on the information set available at t, label with the outcome at $t + \Delta$.
Base-model drift	Every base upgrade invalidates the adapter — LoRA factors are tied to specific W_0.	Keep the dataset and the benchmark as the durable assets. Re-running the fine-tune is hours; rebuilding the dataset is months.
Memorisation of sensitive text	Verbatim recall of names or privileged report content.	De-identify before training, cap epochs, and probe for regurgitation as part of the benchmark.

The controls were written for data. A model needs its own.

Clause	What it requires	What that means for a trained model
B.5.1 Multi-source	Pattern seen at two or more unaffiliated organisations	Must govern training-set composition, not just the pattern list. Cap the share of examples any one contributor supplies.
B.5.2 Non-attribution	Not identifiable directly “or by inference”	Extraction and membership-inference testing. The regurgitation probe becomes a release gate, not a QA step.
B.5.4 Temporal	Minimum three-month lag	A hard cutoff enforced in the dataset builder, not a reviewer's judgement.
B.7 Review window	Thirty-day item-level customer veto	You cannot surgically remove a vetoed example from a trained adapter. A veto means retraining — so batch reviews before training runs.
B.11 Audit	Customer may audit the separation controls	Auditable training manifests: which examples, which tier, which qualification status, which adapter version.

A model is a lossy, queryable compression of its training data. If a Base Delta can be prompted into revealing something customer-specific, every control above was applied one layer too high.

FIELD IQ EDGE

Sizing the student to the backpack

Params	4-bit footprint	Runs on	Capable of
1–4 B	0.7–2.5 GB	Phone, Jetson Nano/Orin Nano	Identification, extraction, procedure lookup, form filling
7–8 B	4–5 GB	Jetson Orin NX/AGX, mini-PC	Light reasoning, drafting records, ranked warnings
12–14 B	7–9 GB	Orin AGX 64 GB, small dGPU	Stronger reasoning — battery and thermals become the constraint
30 B +	17 GB +	Rack GPU	Not a backpack. This is the base-station tier.

Recommendation: 7–8 B at 4-bit.
Enough capability to draft and reason, small enough for a wearable power budget. Runtime: GGUF via llama.cpp, or TensorRT-LLM on Jetson.

Split the workload.
The backpack handles ~90% of routine work fully offline. Hard cases queue locally and escalate to the large sovereign model when connectivity returns.

WHY THIS COMPOUNDS

The loop is the product

1

Deploy

Adapter ships to the device. Runs offline.

2

Defer

Low confidence → “I'm not sure” → the technician decides.

3

Capture

The correction is logged locally with full context.

4

Sync

Corrections return when connectivity allows.

5

Retrain

Next Distillery run folds them in. Benchmark re-scored.

Every field day makes the next version better.
A competitor can download the same base model this afternoon. They cannot download the correction log, and they cannot shortcut the years of real work inside real plants that produced it.

NEXT STEPS

Three decisions, then eight weeks

1

Name the one job

One sentence describing exactly what the Field IQ Edge model does. Narrow wins; broad fails.

2

Write the exam before the model

150 held-out cases from Orland's documented incidents, using pre-event data only. This is the gate and the sales artifact.

3

Confirm the Base / Tenant split with counsel

That the two-layer model maps onto Schedule B.3, B.4 and B.5.6 as drafted — plus written permission for training use of incident data. Week one, not week seven.

Distillery produces EON Deltas.

Base Delta — ours

Tenant Delta — theirs

+ dataset + benchmark

One per customer, one per job.

The customer keeps the story. EON keeps the pattern.