Deploying Subsurface AI Inside a Closed NOC Network: The On-Prem MLOps and Geo-Platform Integration Blueprint

Most subsurface AI fails its final exam not on accuracy but on address: the model works, but it works in a cloud the operator's security team will never sign off on. This whitepaper is the deployment blueprint for the other path — productionizing a deep-learning formation-evaluation system entirely inside an oil-and-gas operator's air-gapped corporate network, where no training byte, no checkpoint, and no log image is permitted to leave the perimeter. Drawn from a multi-phase engagement with a mid-sized Middle East carbonate operator we partnered with, it specifies the four layers that make on-prem subsurface AI real: an on-premise GPU compute tier (an Nvidia DGX A100 class server backed by a data-management server and custom GPU nodes, replacing the consumer-grade cards the R&D phase ran on); an MLOps control plane built on a custom platform with Weights & Biases experiment tracking, a Seafile data backbone, GitHub Enterprise and Ubuntu; a delivery surface that ships each model as a containerized application exposed over a LAN-only API into the operator's existing geomatics platform; and a confidentiality regime — including depth-digit masking — that lets the system retrain and serve without exposing the operator's subsurface position. We then make the decision explicit: three hosting models trade operator control against operating burden, and they are not equivalent on the one axis that matters in production — retraining latency, which ranges from minutes-to-hours under a managed on-prem arrangement to days when the operator runs the loop alone to two-to-three weeks off-premise. The handover is the acceptance test; this whitepaper is the wiring diagram behind it.

Tannistha Maiti
Senior AI Researcher

Tarry Singh
Founder & CEO

MLOPS / SUBSURFACE-AI / ON-PREM-DEPLOYMENT / DATA-SOVEREIGNTY / GEOMATICS-PLATFORM / AIR-GAPPED-NETWORK

CONTENTS

A subsurface AI programme can pass every metric on the slide and still fail the only review that decides whether it ships: the one where the operator's security architect asks where the model runs, where it retrains, and where the training data physically lives. For a national oil company, the honest answers to those three questions are usually fatal. The model runs in a vendor's cloud. It retrains by re-uploading proprietary image logs. The training data lives on someone else's disks, in someone else's jurisdiction, under someone else's incident-response plan. None of that survives contact with a board that treats its subsurface position as a national asset. The model was never the problem. The address was.

This whitepaper is the wiring diagram for the other address: a subsurface AI system that runs, retrains, and serves entirely inside the operator's own air-gapped corporate network, with no training byte, no checkpoint, and no log image ever crossing the perimeter. It is written for the people who actually have to make that real — the geomatics platform lead who owns the geo-analytics layer the model has to plug into, the IT and infrastructure lead who owns the network and the iron, and the enterprise architects and data-platform owners who have to certify the whole thing as operable. It is drawn from a multi-phase formation-evaluation engagement with a mid-sized Middle East carbonate operator we partnered with, where the brief was not "build a model" but "build a model we can own, inside our walls, after you leave."

The constraint that designs the system: the perimeter is the spec

In most ML deployments, the network is an implementation detail. In subsurface AI for a national oil company, the network is the specification. The defining requirement is not throughput or latency; it is that subsurface data — the operator's most sensitive intellectual property — must never traverse a boundary the operator does not own. That single constraint cascades into every other decision in this document. It rules out a managed cloud notebook. It rules out a SaaS inference endpoint. It rules out the convenient default of "just retrain it for them in our environment and send the weights back," because the data needed to retrain cannot leave to begin with.

So the architecture inverts. Instead of moving data to the compute, you move the compute — all of it — inside the perimeter. The GPU tier, the experiment tracker, the data backbone, the model registry, the serving layer, and the retraining loop are provisioned on-premise on

hardware the operator controls, on a network with no path to the public internet for subsurface payloads. This is more expensive and more demanding than a cloud build, and pretending otherwise is how programmes over-promise. The reason to accept the burden is singular and decisive: it is the only design a national oil company's security function will sign.

> **ON-PREM IS NOT A PREFERENCE HERE — IT IS THE REQUIREMENT**
>
> The proposal phase costed public-cloud, hybrid, and on-premise infrastructure explicitly, and the recommendation leaned to a private/hybrid posture for one reason above all: IP protection and regulatory control, with the avoidance of public-cloud cost-shock a secondary benefit. For a national oil company, "where does the data live" is answered before "what does it cost." Every layer below is a consequence of answering that question correctly.

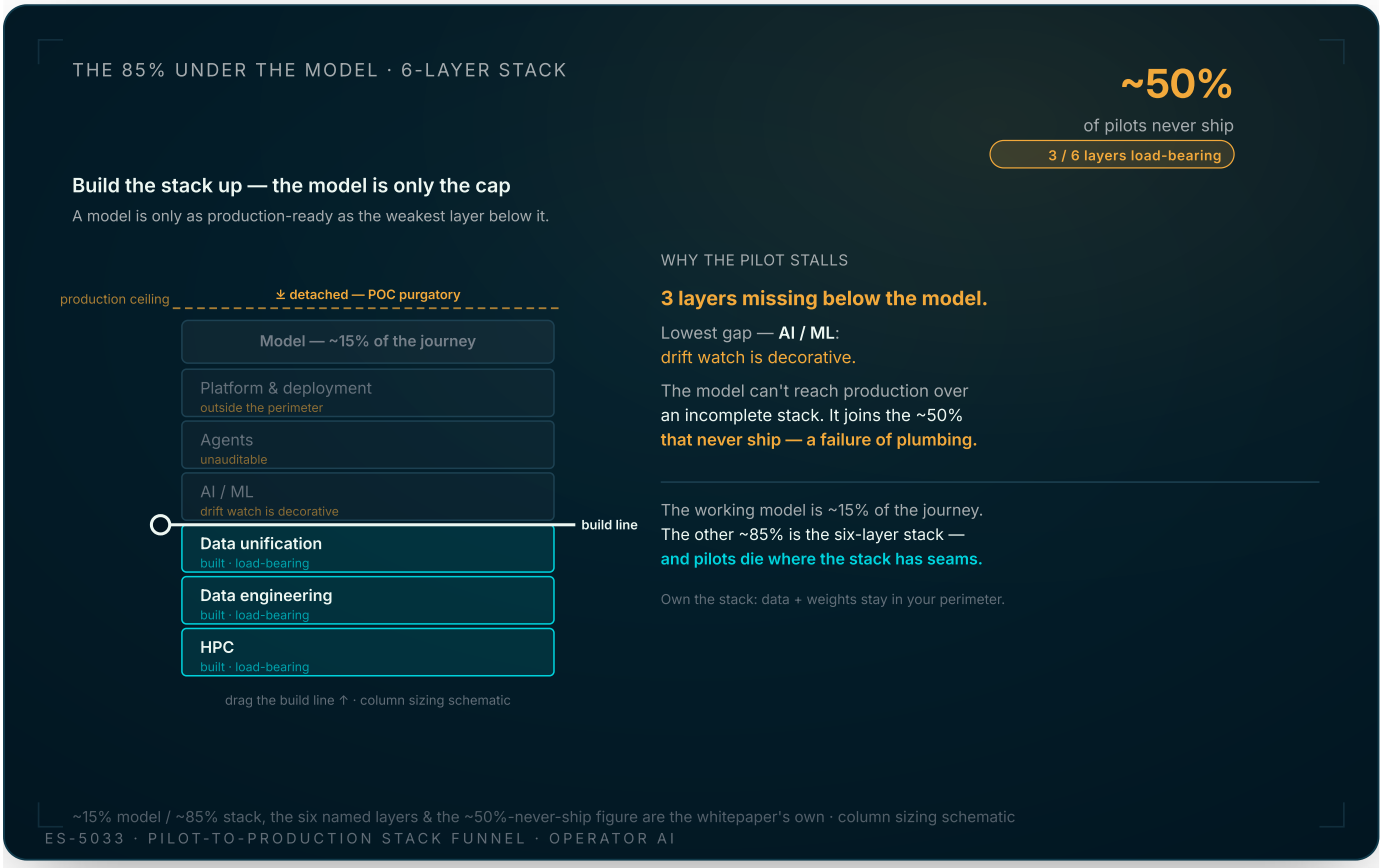
Layer one: the on-prem compute tier — from consumer cards to a DGX-class spine

The research phase of a subsurface programme can run on whatever GPUs are at hand. Ours did: the early deep-learning experiments — the unsupervised pickers, the first transformer-based sinusoid detectors on borehole image logs — were trained on a stack of consumer-grade cards offering just 8 GB of memory per machine. That is enough to prove an architecture converges. It is nowhere near enough to retrain a multi-task detection model on a growing well inventory, on the operator's own schedule, inside their building.

Productionizing therefore begins with replacing the R&D iron with a compute tier sized for the real training workload. The spine of the on-prem build is an Nvidia DGX A100-class server — four to eight A100 GPUs, up to 640 GB of aggregate GPU memory, in the 2.5-to-5 petaFLOPS AI range — provisioned with 7.7 TB of SSD, 512 GB of system RAM, and 320 GB of GPU memory. Behind it sits a dedicated data-management server (4 TB SSD, 128 GB RAM) and a pair of custom GPU nodes for parallel experimentation and serving. Where an operator's ambition runs to a fleet of models across many assets, the same line scales to a multi-node DGX A100 SuperPod tier reaching 25-to-50 petaFLOPS with 3-to-6 TB of GPU

memory over a 200 Gb HDR InfiniBand fabric — but that is a capacity decision, not an architectural one. The point for this whitepaper is that the production compute lives on the operator's floor, not in a region a cloud provider chose.

This is also the clearest illustration of why a pilot is not a product. The model that ran on 8 GB consumer cards in the lab is the same model — but the apparatus required to retrain it on demand, inside the perimeter, at production cadence, is a different class of machine entirely. The leap from R&D to production is not a better network; it is the six load-bearing layers underneath the network, and the compute tier is the first of them.



Pilots don't stall because the model is weak. The working model is only ~15% of the journey; the other ~85% is a six-layer engineering stack (HPC → Data engineering → Data unification → AI/ML → Agents → Platform/deployment), and a project ships only when every layer below the model is built to production grade. Drag the build line up the load-bearing column: with all six built the model reaches the production ceiling; with any gap below it the model detaches into POC purgatory — the ~50% that never ship. The ~15%/~85% split, the six layers and the ~50% figure are the whitepaper's own; the equal-sixths column sizing is schematic.

The funnel above is the attrition every operator should plan for and most do not. A validated metric on consumer GPUs clears exactly one gate. On-prem HPC, a provenance-bearing data layer, a unified ontology, the model itself, an auditable agent layer, and a deployment

surface inside the security perimeter are all gates between that metric and a workflow on the asset — and a programme that under-builds any layer below the model ships a checkpoint that detaches into the roughly half of enterprise AI pilots that never reach production. The discipline is to provision the full stack on-prem as a first-class deliverable, not to bolt the hard layers on after the model "works."

Layer two: the MLOps control plane, behind the firewall

A GPU server is not an MLOps platform any more than a printing press is a newspaper. The control plane is what turns raw compute into a system the operator's in-house team can run: ingestion, data versioning, experiment tracking, a model registry, and a retraining loop, all wired together and all hosted inside the perimeter.

The stack we built and handed over runs on Ubuntu 20 LTS with GitHub Enterprise for source, a custom MLOps platform as the orchestration spine, **Weights & Biases** as the experiment tracker, and **Seafile** as the data backbone — the latter provisioned as a 1 TB network store over 4 TB of redundant SSD. None of these are cloud services in this deployment; they are self-hosted instances on the operator's network. That distinction is the whole game. A cloud-hosted experiment tracker is a hole in the perimeter wearing a dashboard; the same tool, self-hosted, is a lab notebook the security team can certify.

The engineering vocabulary here matters because it is what the receiving team inherits. Every processed dataset is a content-addressed, immutable artefact named by a hexadecimal UUID rather than a human label, so a run is permanently bound to the exact bytes it consumed. This is not cosmetic: across the engagement the core training dataset grew from roughly 900 image-and-ground-truth pairs to over 55,000 — a 65-fold expansion driven by overlapping-patch generation and augmentation — so "the dataset" referred to dozens of distinct frozen sets over time. Without content-addressing, retraining inside the closed network would be irreproducible by construction. With it, the operator's engineers can point at the precise artefact behind any model on the asset.

> **DATA QC IS A VERSIONED GATE INSIDE THE PERIMETER, NOT A VIBE**
>
> Of ten wells received in one intake, two were excluded before training — both carried abnormal static-image value ranges in the binary wireline log file that fell outside the normal 0-255 band and defeated normalisation, and one of the two had additionally

> been logged with a different image-logging tool whose response was not directly comparable. Eight wells went to training; the recorded reason for the other two lives in the dataset's provenance. When that exclusion logic runs on the operator's own MLOps plane after handover, the next engineer reads the lineage rather than guessing. Silent exclusion is how irreproducibility re-enters a programme the moment the build team leaves.

Layer three: the delivery surface — a LAN-only API into the geomatics platform

A model that only a data scientist can invoke is not deployed; it is parked. The delivery layer is what puts a frozen checkpoint in front of a geoscientist who has never opened a terminal — and for a closed-network operator, it has to do that without opening a single external port.

The pattern that survives a security review is deliberately boring. Each production capability is packaged as an application — built in **Streamlit**, containerized with **Docker**, exposed on its container port (8501) — and run on an internal corporate server behind the operator's VPN. The geoscientist never sees the container; they see a link embedded directly in the operator's existing geomatics analytics platform, one link per well, sitting in that well's description card. Click it, and the model's predicted bedding and fracture regions overlay the interpreted image for the intervals a human has not yet picked. The model's output reaches the interpreter through the geo-platform they already use, over a LAN-only API, with no traffic leaving the building.

This is the integration most programmes underestimate. The hard part of "plug the model into the operator's platform" is not the inference; it is doing it as a containerized service on a LAN API into a live geomatics system, behind a VPN, with deployment artefacts the operator's ICT team can rebuild themselves. The serving surface and the geo-platform integration are co-designed with the operator's geomatics engineers, not handed to them as a finished black box — because a delivery surface the receiving team cannot rebuild is a delivery surface with a half-life.

> **THE CONFIDENTIALITY REGIME EXTENDS INTO THE MODEL'S OWN INPUTS**

> Perimeter control is necessary but not sufficient; sensitive identifiers can leak through the data itself. Depth is the obvious one — a true measured depth pins a prediction to a specific position in a specific well in a specific field. The discipline is to treat depth as a maskable channel: the model learns and predicts on relative geometry within a patch, while absolute depth digits are masked or offset wherever a result moves between contexts, so a shared artefact never carries a coordinate that re-identifies the operator's position. The instrument floor sets the tolerance for all of this — at the image resolution in play, a single image-log pixel corresponds to roughly 3 cm of depth, so the localisation tolerance that defines a true versus false positive is chosen against that physical ±3 cm floor, not against the masked label. Confidentiality and metric honesty are the same engineering discipline viewed from two sides.

Layer four: the retraining loop, and why hosting model is the real decision

Everything above is static until the system has to learn from a new well. That is where on-prem subsurface AI either lives or quietly dies — because retraining is the operation that requires the data, and the data cannot leave. The retraining loop must therefore close entirely inside the perimeter: a new well lands on the Seafile backbone, the data-QC gate runs, the dataset UUID advances, a training run executes on the DGX tier with its full configuration and metrics logged to the self-hosted tracker, the resulting checkpoint is promoted in the registry, and the serving container picks it up. No step in that loop touches an external network.

The honest decision an operator faces is not *whether* to retrain on-prem — it is *who runs the loop*, and that is a genuine trade between control and operating burden. We costed three hosting models with their real consequences rather than selling one. In the first, the operator owns and operates everything: maximum sovereignty, maximum staff load. In the second, the operator owns the stack but the loop is co-operated with the partner team: shared burden, retained control. In the third, the stack stays on the operator's iron but the partner manages the retraining service end to end: minimum operator load, maximum dependence. The investment envelope scales with that choice across roughly USD 250-350K, 650-800K, and 1.5-4M tiers of escalating platform capability — but the number that actually governs the decision is not the capital figure. It is retraining latency.

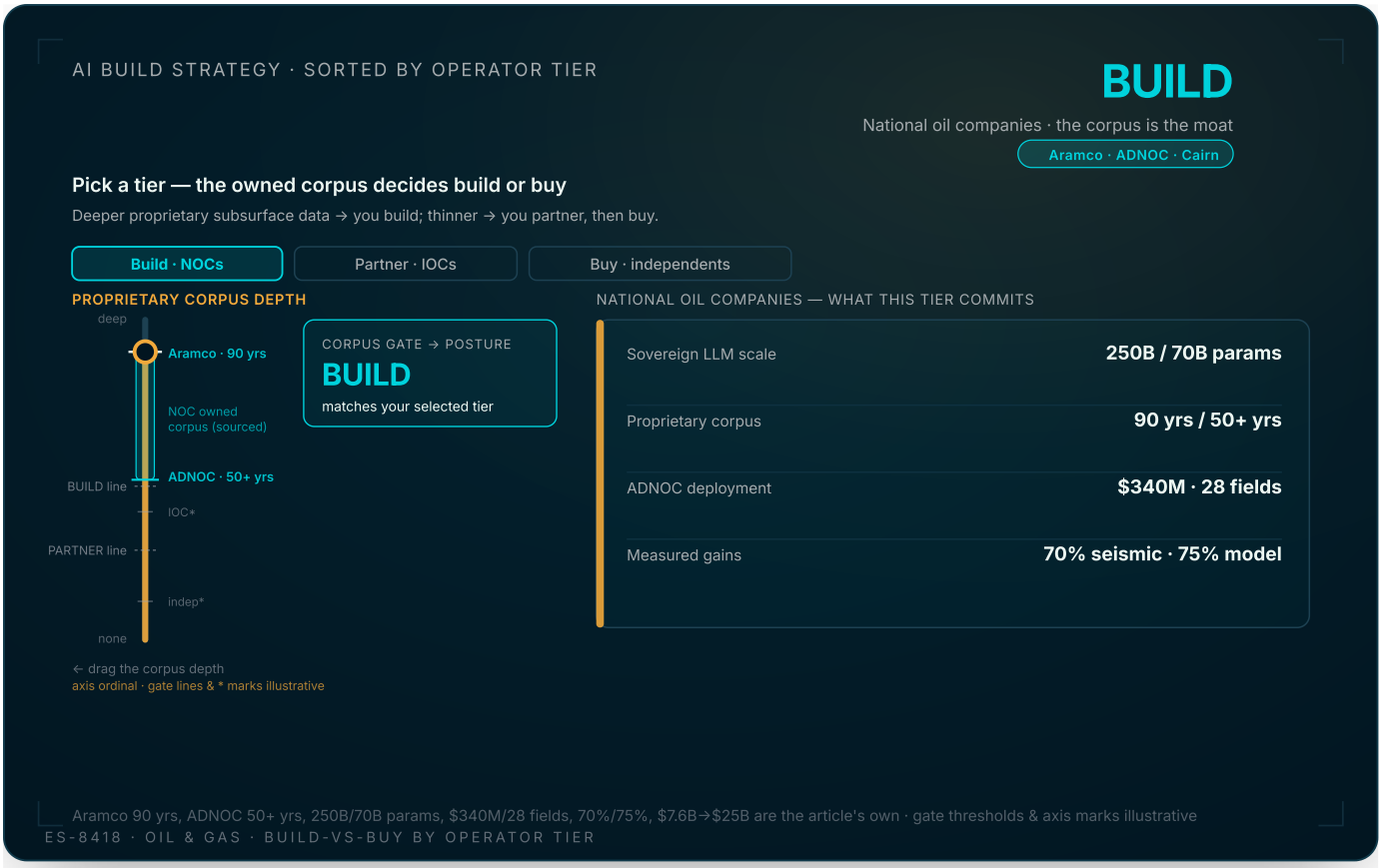
Under a managed on-prem arrangement, where the loop is tuned and operated as a service on the operator's hardware, a retrain on a new well runs in **minutes to hours**. When the operator runs the loop entirely alone — the go-on-your-own posture — the same retrain takes **days**, because the in-house team is absorbing the orchestration, the QC adjudication, and the sweep that the managed arrangement had automated. And an off-premise managed alternative, where the heavy work is handled in an external environment, runs on a **two-to-three-week** cadence, because the data-movement and scheduling friction the perimeter was built to avoid reasserts itself at the boundary. Same model, same hardware footprint — an order-of-magnitude spread in how fast the system can respond to new ground, set entirely by who operates the loop. That spread is the staleness the next exhibit makes concrete.

The case study's real argument isn't faster retrains — it's that the gap between 'model is stale' and 'model is fresh again' is where decisions ran on quietly degrading predictions. Drag 'today' across a year of operating life: under the manual queue, drift hid for months and a retrain took six-plus weeks, so the model's staleness sawtooth grows long teeth; under the agentic loop, drift surfaces in days and a retrain runs overnight, so the teeth collapse. The orange band is the silent-drift exposure the loop removes — the window in which five of 18 requeued models had drifted into a combined $4.2M of misallocated infill capital. The retrain cycle times (6 weeks → overnight, ~40×), the drift-detection cut (months → days), +22% accuracy, the $4.2M figure and ~40 assets are the case study's own; the week-by-week staleness curve shape, the year-long retrain cadence and the vertical weeks-stale scale are schematic, drawn to argue the gap rather than chart a measured series.

The ledger above is why hosting model is a production decision, not a procurement footnote. A subsurface model begins decaying the moment the formation it sees diverges from the formation it trained on, and the gap between "this model is stale" and "this model is fresh again" is exactly the retraining latency above. A minutes-to-hours loop keeps that gap to slivers; a multi-week loop lets it grow long teeth, and every day inside the gap is a day the asset team is interpreting new wells against a model that no longer reflects them. Capability transfer is a dial, not a switch — and the operator, not the vendor, should set it with the staleness consequence in full view.

Handover as the acceptance test: three deliverables, packaged to be owned

A closed-network deployment is only as real as the operator's ability to run it after the build team leaves. We treated handover not as a documentation drop but as the acceptance test for the entire architecture, and packaged the work as three self-contained production capabilities — vug detection, bedding-and-fracture detection, and well-to-well correlation. Each was handed over as a complete unit: the versioned dataset, the frozen model, the architecture specification, the output format, and the runbooks. A model without its dataset and its operating documentation is not a deliverable; it is a liability with good metrics and a maintenance contract.



In 2026 the AI build-vs-buy split in oil & gas is sorted by operator tier, and the deciding variable is the depth of the proprietary subsurface corpus an operator owns. Pick a tier — NOCs (Build), Western IOCs (Partner), mid-tier independents (Buy) — and the panel reconfigures to that tier's posture, named operators and the article's own commitments. The orange ladder is the single argument: the deeper the owned corpus (the sourced NOC band runs from ADNOC's 50+ years to Aramco's 90 years), the further toward BUILD a tier sits. Drag the corpus-depth marker — or step tiers with the chips / arrow keys — and the recommended posture snaps to the band the depth lands in. Named operators, the NOC corpus depths, model sizes ($340M / 28 fields, 250B / 70B params), the 70% / 75% gains and the $7.6B→$25B market are the article's own; the corpus-depth axis, the gate thresholds and the IOC / independent marker positions are illustrative.

The build-versus-operate gate above is the same trade the three hosting models encode, viewed from the operator's chair. And it is decided as much by people as by iron. The engagement deliberately built local capability alongside the system, training a cohort of 55 young professionals — 15 of them local nationals, drawn from regional universities — so that the judgement to run, debug, and extend the on-prem platform lived inside the region rather than departing with the delivery team. This is the part of "data sovereignty" that gets quietly dropped: a stack the operator physically owns but cannot operate is not sovereign, it is stranded. The infrastructure disciplines in this document — content-addressed datasets, a self-hosted tracker, a model registry, a containerized LAN delivery surface, a perimeter-closed retraining loop — are precisely the things that make operation teachable. A model only one person understands is not handoverable, no matter whose building it sits in.

This is not, and should not be read as, a purely textbook reference. The architecture here is the one we deployed inside a real national-oil-company network, and the same patterns generalize across the operators we have worked with in the Middle East and the United States — the perimeter constraint, the on-prem control plane, the LAN delivery surface, and the hosting-model dial are not specific to one field. What is specific to one engagement are the numbers: every quantified result in this whitepaper — the 65-fold dataset growth, the well-exclusion ratio, the retraining cadences, the trained cohort — traces to the Middle East carbonate programme, and to that programme alone.

What good looks like

For the geomatics platform lead, the IT lead, or the enterprise architect certifying a subsurface AI deployment for a closed network, the questions that decide whether it is real are not about model architecture. They are about the address:

- Does every part of the system — compute, tracker, data backbone, registry, serving, and the retraining loop — run inside the operator's own perimeter, with no path for subsurface payloads to an external network?
- Is the production compute tier sized for retraining on a growing well inventory at production cadence, not just for proving an architecture on consumer GPUs?
- Is the delivery surface a containerized application on a LAN-only API into the operator's existing geomatics platform — rebuildable by the operator's own ICT team — rather than an external endpoint or a vendor portal?
- Does the confidentiality regime extend into the data itself, so that depth and other re-identifying coordinates are masked or offset before any artefact moves between contexts?
- Has the hosting model been chosen with its retraining-latency consequence in full view — minutes-to-hours, days, or weeks — by the operator, not assumed by the vendor?

If the answer to all five is yes, the operator owns a sovereign, operable subsurface AI system. If the answer to any is no, they own a model with the wrong address — and in a closed network, the wrong address is the same as no model at all.

WHAT THIS WHITEPAPER ARGUES

1. For a national oil company, the perimeter is the specification: subsurface data must never cross a boundary the operator does not own, so the compute moves inside the walls — not the data out.
2. Productionizing means replacing R&D consumer GPUs with an on-prem DGX A100-class compute tier (7.7TB SSD / 512GB RAM / 320GB GPU RAM), backed by a data-management server and custom GPU nodes.
3. The MLOps control plane — custom orchestration, Weights & Biases, Seafile, GitHub Enterprise on Ubuntu — is self-hosted; a cloud-hosted tracker is a hole in the perimeter wearing a dashboard.
4. The delivery surface is each model packaged as a Docker/Streamlit app on a LAN-only API, embedded per-well in the operator's existing geomatics platform behind the VPN — plus a depth-masking confidentiality regime in the data itself.
5. Three hosting models trade control against burden, but the governing axis is retraining latency: minutes-to-hours (managed on-prem) vs days (go-on-your-own) vs 2-3 weeks (off-prem) — chosen by the operator, with the staleness consequence in view.
6. Handover is the acceptance test: three capabilities packaged with dataset+model+architecture+output+runbooks, plus local capability (55 trained, 15 in-country nationals) so a physically-owned stack is also an operable one.

References

International Energy Agency, 2025 International Energy Agency. Energy and AI Special Report (2025). Missing internal expertise identified as the dominant adoption barrier across the energy sector. https://www.iea.org/reports/energy-and-ai

Sculley et al., 2015 D. Sculley et al. Hidden Technical Debt in Machine Learning Systems. NeurIPS 2015. The canonical argument that the trained model is a small fraction of a production ML system. https://papers.nips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html

NVIDIA, 2020 NVIDIA Corporation. NVIDIA DGX A100 System Architecture (2020). Specification basis for the on-prem compute tier described in this whitepaper. https://www.nvidia.com/en-us/data-center/dgx-a100/

Carion et al., 2020 N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko. End-to-End Object Detection with Transformers (DETR). ECCV 2020. Architectural basis for the set-prediction detection model served by this stack. https://arxiv.org/abs/2005.12872

EARTHSCAN

Deploying Subsurface AI Inside a Closed NOC Network: The On-Prem MLOps and Geo-Platform Integration Blueprint

Published June 2025. Generated 2026-07-22.
https://www.earthscan.io/whitepapers/subsurface-ai-on-prem-mlops-geo-platform-integration