Spectro Cloud

Building the university AI factory

How research computing and platform teams can turn scarce GPU budgets into self-service AI for every lab, course, and department

Executive summary

The gap between demand for AI compute on campus and your ability to supply it is widening. Researchers expect GPU access in days; procurement and integration cycles deliver it in months. Federal shared resources like the NAIRR pilot are heavily oversubscribed. Meanwhile, the GPUs you already own often run at a fraction of their capacity, locked inside one-off environments that were built by hand for single groups.

This paper makes the case for running campus AI infrastructure as a factory: a standardized, governed system that turns hardware into self-service AI environments for every lab, course, and department. We'll cover the funding and policy forces behind the shift, the five operational problems that decide whether a factory works, practical steps for getting started, and where our PaletteAI platform fits if you'd rather not do six months of integration yourself.

1. The squeeze on campus AI

If you run research computing or platform engineering at a university, you're caught between forces that don't care about each other: demand that grows monthly, funding that arrives in multi-year cycles, and hardware that depreciates whether or not anyone uses it.

Demand first. In the 2025 EDUCAUSE AI Landscape Study, 57% of institutions called AI a strategic priority, up from 49% a year earlier, and 68% of respondents said students use AI more than faculty do. The pressure comes from every direction at once. A genomics group wants to fine-tune a foundation model. The CS department needs course-scale GPU access for 200 undergraduates. The library wants a research-assistant chatbot grounded in its own collections. An expert roundtable on AI education published in 2025 put the supply side plainly: students "want to do interesting things… and we just don't have the GPU access," while existing HPC clusters are dominated by research workloads, leaving teaching to compete for leftovers.

Federal infrastructure won't absorb the overflow. The NAIRR pilot, the National Science Foundation's program for sharing AI compute with US researchers, has supported more than 500 projects and 5,000 students since January 2024, and the White House AI Action Plan has set it on the path to a permanent operations center. But Georgetown's CSET estimates the pilot's pooled capacity at roughly 5,000 H100 equivalents for the entire country, much of it on systems already near capacity. It's a valuable on-ramp, and a poor substitute for capacity of your own. Competitive allocations also tend to favor institutions that can show they already know how to run AI infrastructure.

That national-competitiveness logic is reshaping capital plans. Texas A&M put $45 million into an NVIDIA DGX SuperPOD to triple its supercomputing capacity, following similar moves at Georgia Tech, the University of Maryland, and NJIT, which expects AI to account for roughly a third of its research budget. Funding agencies, state programs, and industry partners increasingly treat demonstrable AI capacity as a prerequisite for grants and collaborations. Universities that can't offer it lose researchers, students, and money to the ones that can.

The budgets haven't followed the ambition. In the same EDUCAUSE study, just 2% of institutions said they're covering AI costs from new funding sources, and 34% of executive leaders admitted their institution tends to underestimate those costs. Capital and grant cycles run on annual or multi-year rhythms; AI demand doesn't wait for either. And the apparent shortcut (handing everyone an API key to a frontier lab or hyperscaler service) converts capital discipline into uncapped operating spend. Token bills scale with usage, agentic workloads chain many model calls per request, and your most sensitive research data can't legally leave your control anyway. Per-token APIs are a fine way to prototype. As a campus-wide strategy, they're a budget you can't forecast. Total-cost analyses of inference generally find that owning beats renting once utilization is sustained and high, which makes utilization the variable your whole strategy hinges on. More on that in section 3.

While central IT deliberates, people route around it. One 2025 survey of education employees found 78% knew of colleagues using unauthorized AI tools, while only 31% said their institution had clear policies; Ellucian's research puts AI use among faculty and administrators at 84%. Shadow AI is what a service gap looks like from the outside. When the sanctioned path takes months, people with deadlines take the unsanctioned one, and student records, unpublished results, and grant-funded IP go along for the ride. IBM's 2025 Cost of a Data Breach report found that organizations with high levels of shadow AI paid about $670,000 more per breach than those with little or none.

So that's the squeeze: demand compounding, budgets flat, national resources oversubscribed, shadow usage filling the vacuum. More heroics from a small team building bespoke environments one at a time won't close the gap. Changing the operating model will.

2. What an AI factory means on campus

"AI factory" is the industry's term, and the manufacturing metaphor is earned. It describes a production line that takes accelerated hardware in at one end and reliably ships AI services out the other: trained and fine-tuned models, inference endpoints, notebook environments, the chatbot the library asked for.

Building that line by hand means integrating the operating system, GPU drivers, Kubernetes, schedulers, ML frameworks, and serving runtimes, then keeping every version compatible and patched for the life of the system. NVIDIA estimates **six or more months** for a typical first build. That's for one environment, for one team.

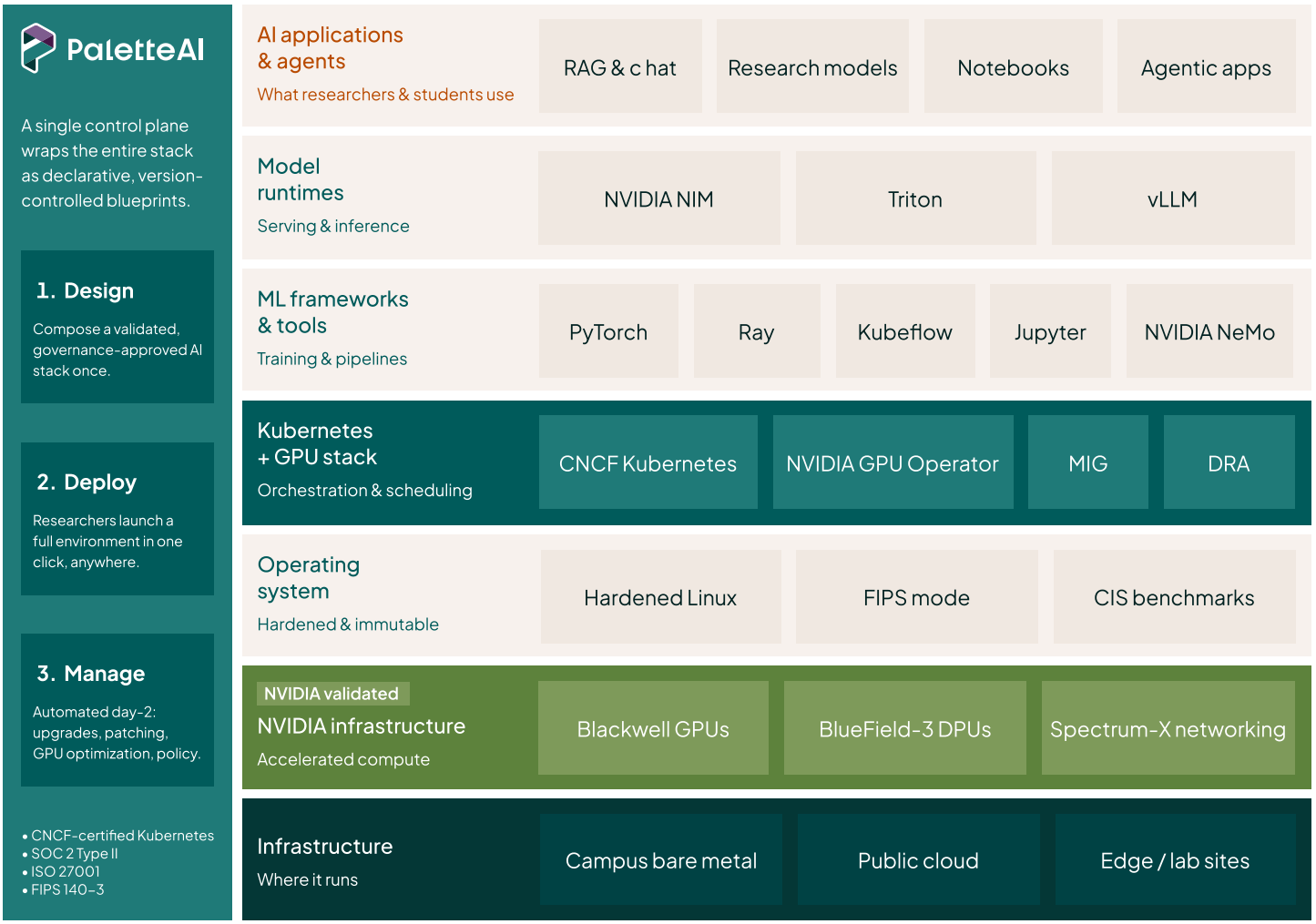
The factory model changes who does what. A platform team designs an approved stack once (operating system through model runtime, with security and compliance built in) and publishes it as a versioned blueprint. Researchers, instructors, and students then deploy complete environments from that blueprint on demand, inside quotas and guardrails the platform team controls. Day-2 work — upgrades, patches, security scans, capacity tuning — happens centrally, across every environment at once.

One question comes up on every campus: what about the HPC cluster we already have? Keep it. A Slurm-scheduled cluster is good at what it was built for: long-running batch jobs with fixed allocations. An AI factory serves a different shape of work (interactive notebooks, always-on inference endpoints, course environments that appear in week one and vanish after finals), typically on Kubernetes.

The two coexist on most campuses that do this well, sometimes drawing on a shared GPU pool, and your HPC investment becomes a complement rather than a casualty. The factory covers the workloads a batch scheduler was never designed to serve.

The university AI factory: one stack, from bare metal to model

PaletteAI assembles, deploys, and runs every layer as a single, governed system, on-premises, in the cloud, or at the campus edge.



3. Five problems that decide whether your factory works

A factory that ships environments nobody can afford, secure, or staff is just a liability with good branding. In our experience these five problems decide the outcome, in rough order of financial impact.

3.1 Utilization: stop paying for idle silicon

The single largest source of waste in AI infrastructure is the idle GPU. A **Wesco analysis of more than 4,000 Kubernetes clusters** measured average GPU utilization at just 13%; typical estimates land between 15% and 25%, and even OpenAI has put its own figure around a third. The root cause is the Kubernetes default of one workload claiming one whole GPU. A 96 GB card serving a 14 GB model wastes most of the silicon a grant paid for.

The fix is NVIDIA's sharing toolkit, matched to each class of workload. Time-slicing rapidly alternates workloads on shared memory: fine for bursty development and student notebooks, but with no isolation, so never for sensitive work. vGPU partitions at the virtualization layer, useful where VMs are mandatory.

Multi-Instance GPU (MIG) partitions at the silicon level, giving each slice dedicated streaming multiprocessors, memory, and L2 cache; tenants can't see or interfere with one another, which is what makes it safe for a physics lab, a genomics group, and an undergraduate course to share one card without trusting each other's code. And Dynamic Resource Allocation (DRA), generally available since Kubernetes 1.34, replaces "give me one GPU" with claim-based scheduling: workloads describe what they need and the scheduler finds a matching slice.

How far one card stretches

GPU	Max MIG instances	Example campus use
RTX PRO 6000 Blackwell (96 GB)	4	Four isolated 7B LLMs, one 24 GB slice each
NVIDIA H100	7	Seven inference tenants on a single training-class card
NVIDIA A100	**7**	Course labs and research endpoints side by side
RTX PRO 5000	**2**	Two isolated workloads on a workstation-class card

Utilization, before and after

Hardware-isolated GPU sharing: one card, four tenants, utilization headed toward 90%.

~15% Whole-GPU scheduling: one workload owns the card. Industry average measured as low as 13%.

▼ MIG + DRA partitioning

Many isolated tenants per card. Up to 70% more usable GPU from the same hardware. ~90%

One RTX PRO 6000 Blackwell, four tenants

96 GB GPU → 4 × 1g.24gb partitions — NVIDIA MIG

24 GB Physics lab — 7B LLM · ~14 GB	24 GB Genomics — 7B LLM · ~14 GB
24 GB CS course — 7B LLM · ~14 GB	24 GB Library RAG — 7B LLM · ~14 GB

Dedicated SMs, memory & L2 cache per slice — tenants cannot see or interfere with each other.

Sharing mode	Best for	Isolation
Time-slicing	None (shared memory)	Bursty dev /notebooks
MIG	Hardware (silicon)	Multi-tenant production

Native, out-of-the-box support for the NVIDIA GPU Operator, vGPU, MIG, time-slicing, and Kubernetes Dynamic Resource Allocation (DRA, GA in Kubernetes 1.34). A 7B-parameter model needs roughly 14 GB in FP16, so a 24 GB slice runs one comfortably.

The arithmetic favors you. A 7B-parameter model needs about 14 GB in FP16, so a single 24 GB MIG slice runs one comfortably, and FP4 quantization fits larger models into the same space. Combining MIG with DRA pushes shared-GPU utilization toward 90% against that 13–25% baseline. And as section 1 noted, sustained high utilization is exactly the condition under which owning hardware beats renting tokens. Our engineering team has published a hands-on guide to building the MIG and DRA stack, along with a field guide to the NVIDIA GPU Operator.

A typical Kubernetes GPU sits 85% idle. Palette AI partitions each card with NVIDIA MIG so many isolated tenants share it safely, pushing utilization toward 90%.

3.2 Access: researchers measure delay in careers, not tickets

A postdoc runs on a two-year clock. A course starts in week one of the semester whether or not procurement closed. When standing up an environment takes a quarter, the cost lands on careers: papers that don't get written, and faculty recruits who pick the campus where the GPUs already work. The factory answer is self-service against approved blueprints.

The platform team publishes "t-shirt sizes" (a small, medium, and large slice with matching frameworks), and a researcher goes from request to running environment in minutes through a console, Terraform, or an API. Governance happens once, upfront, instead of per request, indefinitely.

3.3 Currency: the stack you validated last year is already old

Models, frameworks, CUDA versions, and serving runtimes churn monthly, and researchers chasing publications want the new thing the week it ships. Bespoke environments handle this badly. Every upgrade is a one-off project, so most never happen, and the "AI infrastructure" calcifies into a museum exhibit. Blueprints handle it well: the platform team validates a new stack version once, publishes it, and every environment moves on a schedule, with rollback when a driver and a framework disagree. Staying current stops being a heroic effort and becomes routine maintenance, which is the only way researchers get this quarter's tooling without your team rebuilding environments every quarter.

3.4 Isolation: multi-tenancy that survives an audit

A campus GPU pool serves export-controlled engineering research, HIPAA-covered medical school workloads, IRB-governed human subjects data, FERPA-covered student records, and an undergraduate who will absolutely try to mine cryptocurrency. That mix demands more than Kubernetes namespaces: hardware isolation at the GPU itself (MIG's silicon boundary), role-based access control, per-tenant quotas so no group starves the rest, and per-tenant usage records for the audit trail. Some sponsored research adds sovereignty conditions, meaning data and infrastructure that stay on soil you control, sometimes fully air-gapped. Those terms rule out shared external services entirely and put the requirement back on your own metal. We've written more about sovereign AI patterns separately.

3.5 Headcount: thousand-node ambitions, five-person team

Research computing teams are small, and a factory multiplies what each person manages. Every node carries an OS, drivers, a GPU operator, Kubernetes, and an observability stack, each needing installation, patching, and upgrades at least three times a year, without configuration drift. Done by hand, the operational load grows with the size of the estate until it consumes the team. Done declaratively, with desired state defined in the blueprint and automation reconciling every cluster against it, the load grows with the number of blueprints instead. That's how a handful of engineers serve an entire campus. Per-tenant cost visibility belongs in this list too: chargeback (or its politically gentler cousin, showback) to grants and departments is how central IT funds the service rather than absorbing it.

4. Getting started

Whether or not you ever talk to a vendor, these practices come up again and again in AI infrastructure projects that work, across enterprises, government, and service providers as much as campuses.

Measure before you buy. Instrument the GPUs you own and collect two weeks of utilization data. If you're at the typical 13–25%, your next "capacity problem" is a sharing problem, and the budget case for partitioning beats the budget case for purchasing.

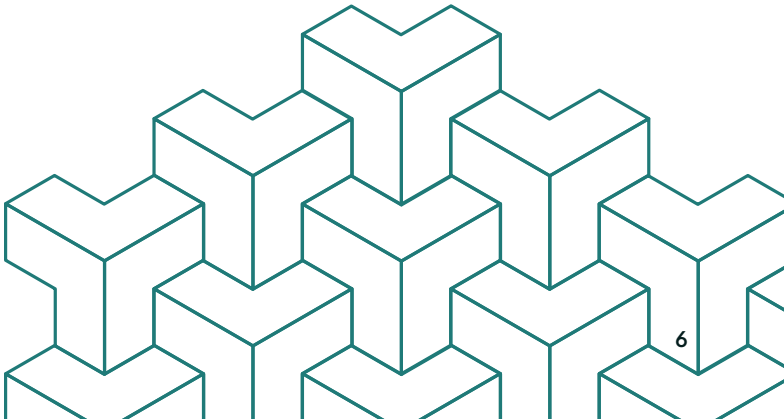
Match sharing strategy to workload class. MIG for multi-tenant inference and course labs; whole GPUs or multi-GPU nodes for serious training runs; time-slicing only where isolation doesn't matter. Publish the menu so tenants know what they're choosing between.

Design chargeback on day one. Even if you never invoice anyone, per-tenant usage data is what sponsored-programs accounting asks for, what makes your next capital proposal credible, and what turns "the GPUs are full" disputes into evidence. Start with showback; graduate to chargeback when the politics allow.

Treat blueprints as products. Each approved stack gets an owner, a version history, a test gate, and a deprecation policy. The moment blueprints become abandonware, you've rebuilt the bespoke problem with extra steps.

Make the sanctioned path the fastest path. Shadow AI follows the path of least resistance, and so does compliance. If self-service through your platform beats a personal OpenAI key on speed, governance mostly takes care of itself. If it's slower, no policy will save you.

Let the factory strengthen your funding story. Utilization data, tenant counts, and time-to-environment metrics are the evidence federal proposals, state programs, and industry partnerships ask for. A working factory compounds: it serves today's researchers and helps win tomorrow's grants.



5. How we can help

Everything above can be built by hand; the six-month estimate describes exactly that work. PaletteAI exists for teams who'd rather spend those months supporting research than doing integration. It's our enterprise AI infrastructure platform, generally available since March 2026, and it manages the entire stack this paper describes — bare metal and operating system up through Kubernetes, GPU software, ML frameworks, and model runtimes — as one governed system, built around the blueprint-and-self-service model you've just read about.

On utilization specifically, the NVIDIA GPU Operator, MIG, vGPU, time-slicing, and DRA are supported natively, out of the box. Your platform team picks MIG profiles from drop-downs, defines t-shirt sizes per project, sets quotas, and gets per-tenant usage data for showback; your researchers deploy against those sizes with no GPU expertise and no infrastructure tickets.

The NVIDIA validation is formal. Spectro Cloud is an NVIDIA Preferred Partner, and PaletteAI is included in the NVIDIA Enterprise AI Factory validated design, with coverage across the modern NVIDIA stack: Blackwell GPUs, BlueField-3 DPUs, Spectrum-X networking, NIM microservices, and NeMo. NVIDIA AI Enterprise is embedded directly into the PaletteAI experience, and validated reference architectures built with partners including WEKA, Netris, Aviz Networks, Canonical, and NVIDIA Run:ai are published on our Design Hub.

For the part of the form your CISO fills in: Spectro Cloud is SOC 2 Type II audited and ISO 27001 certified, a CNCF Certified Kubernetes provider and Certified Kubernetes AI Platform, and a member of the CNCF, OpenSSF, Agentic AI Foundation, and LF Edge.

For regulated and sovereign workloads, the VerteX editions add FIPS 140-3 validated cryptography and air-gapped deployment. Palette is rated a leader in the GigaOm Kubernetes Radar for edge to cloud, and the rest of the trophy shelf lives on our awards page.

The architecture is proven at massive scale: T-Mobile, GE HealthCare, the US Air Force, and Airbus Defence and Space run on it. And because the same blueprint deploys to a campus data center, public cloud, or an edge site, with edge clusters operating autonomously when disconnected, your factory extends to field stations and instrument labs without a second platform.

One blueprint, deployed anywhere, managed for life

PaletteAI separates what the platform team governs from what researchers self-serve, so standing up a new AI environment is a click, not a six-month project.

01 Platform team

Design

Compose a validated, governance-approved AI stack as a reusable Cluster Profile, every layer pinned and version-controlled.

- OS, Kubernetes, GPU operator, drivers
- Frameworks & model runtimes
- Security & compliance baked in

02 Researchers & students

Deploy

Self-service a full environment in one click from a console, Terraform, or API. The same blueprint runs identically everywhere.

03 Automated day-2

Manage

Operate at AI-factory scale: thousands of nodes and clusters from one pane, with policy enforced continuously.

- Upgrades, patching, security scans
- GPU optimization & quota control
- Per-tenant cost & usage visibility

Multi-tenant by design. Strong isolation, role-based access, and per-department quotas let one campus platform safely serve every lab, course, and research group. GPU-as-a-Service and Model-as-a-Service, without "shadow AI."

Bring whatever silicon you've got

PaletteAI is hardware-agnostic by design. Alongside the NVIDIA validation, the platform supports AMD GPUs, including Instinct accelerators (a strong option for inference workloads), plus Arm-based systems, Google TPUs, and AWS Graviton. The same flexibility runs down the stack: any CNCF-conformant Kubernetes distribution, and your choice of OS, storage, and networking. If your next grant buys a different vendor's accelerators, your factory doesn't start over.

Next steps

If this maps to your campus, the most useful next step is a short working session with your research computing and platform teams. Bring an inventory of one existing GPU pool and one candidate workload, and we'll map both onto a first blueprint together. **Book that conversation here.**

To go deeper first, start with our **AI factories overview**, the **MIG and DRA engineering guide**, the **reference architectures on our Design Hub**, and the **PaletteAI documentation**.

Spectro Cloud