Quali | TORQUE

The AI Workstation for the Intelligent Edge: NVIDIA DGX Spark

How we are bringing Agentic Intelligence to the Edge with Torque agents and the new NVIDIA DGX™ Spark

Agentic workflows are disrupting and subverting the way software is developed. In the first stage of AI adoption, companies developed agents and LLMs as separate add-on features and components. An AI chatbot for customer support, a specific agentic workflow within the application that carries out autonomous operations or an MCP server to be able to provide access to the application data by other agents.

As the potential of models to solve non-deterministic problems became apparent though, they started seeping into the core application code. Today, when implementing any core feature in SaaS or client applications, the engineering team would decide based on cost economics, and the nature of the problem which parts of the process would be automated using deterministic code and which automated with AI agents.

Code and agentic workflows become inseparable.

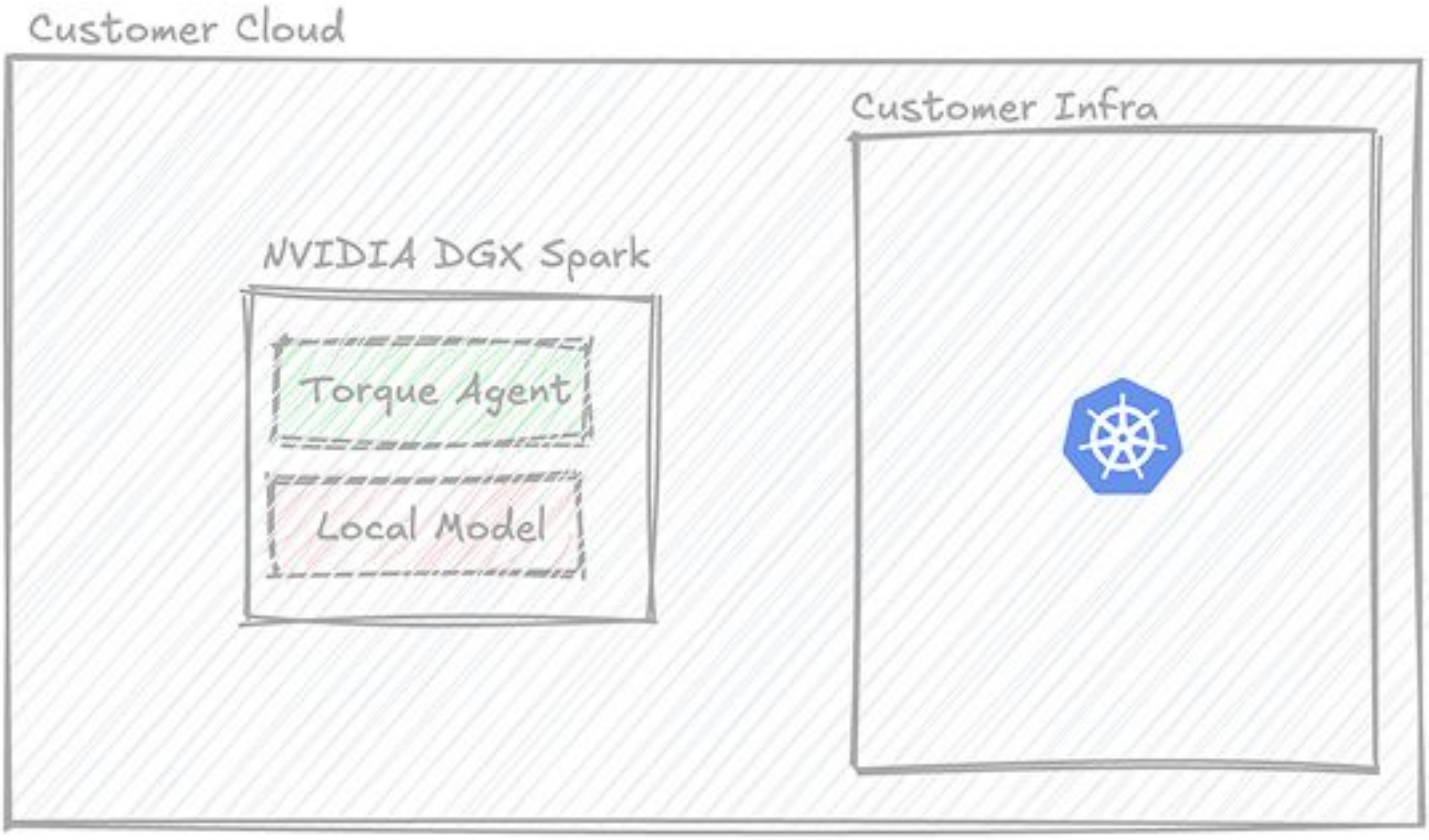
Embedding intelligence into the app creates a new set of constraints, however. The AI runs as part of your application stack, on infrastructure you control, using models you've selected and fine-tuned for your specific use cases. If the application has client-side components, it must also be available as a local capability.

Distributed Local Execution with Torque Agents

We have experienced first-hand the friction of integrating LLMs into application code with Quali's flagship product: Torque. Torque is an IAC AI-driven orchestration and automation engine. The product is specifically architected to support orchestrating distributed infrastructure in both private and public cloud.

As an environment-as-a-service platform, Torque provisions and manages resources across cloud providers, on-premises data centers, Kubernetes clusters, and edge locations, often simultaneously within a single environment blueprint. The control plane can't directly execute commands in these distributed locations. Instead, it relies on Torque agents.

These agents are lightweight processes deployed deep inside customer networks, private clouds, air-gapped facilities, and restricted environments where the central control plane has no direct access. They're the execution layer, the components that actually spin up VMs, configure networks, deploy containers, and manage infrastructure lifecycles in environments that might be separated from the control plane by firewalls, VPNs, network policies, or complete air gaps.



The agent's job is to translate orchestration intent into local action. When a blueprint requires a Kubernetes namespace in a private cluster, or a VM in an on-premises VMware environment, or a database in a customer's AWS VPC, the agent receives instructions from the control plane and executes them within its local context. It has access to the local APIs, credentials, and network paths that the control plane doesn't. It's the bridge between centralized orchestration and distributed execution.

But infrastructure orchestration increasingly requires intelligence, not just execution. When an agent provisions an environment and something goes wrong, a resource conflict, a quota limit, a networking misconfiguration, it needs to do more than report an error code back to the control plane. It needs to reason about the failure, understand the local context (what other resources exist, what constraints apply, what alternatives are available), and either resolve the issue autonomously or provide actionable guidance to users.

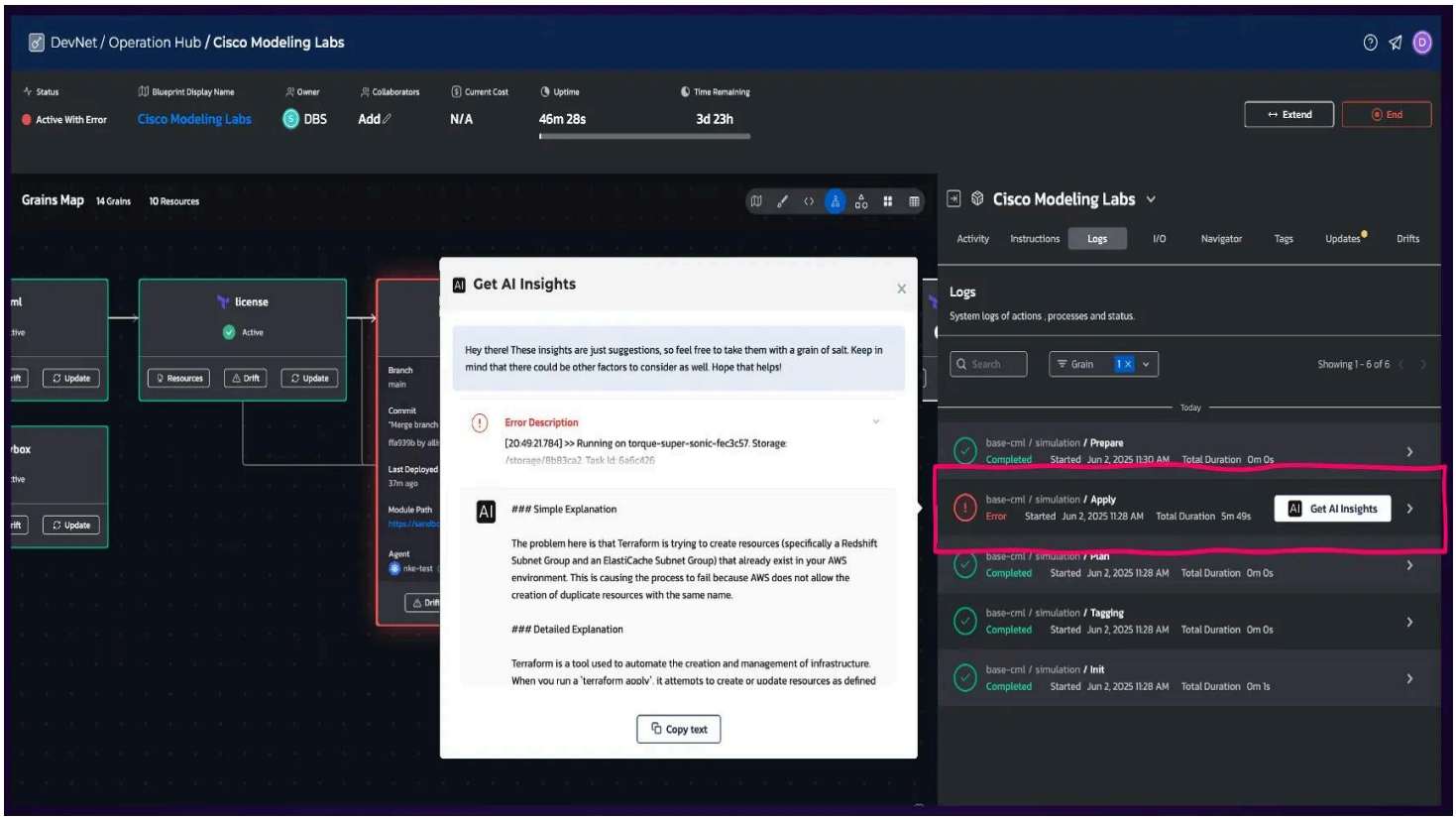
This is where embedded agentic intelligence becomes critical for Torque agents. The agent can't rely on the ability to always reliably call out to a central AI service. Latency, availability, connectivity and predictability makes this approach unreasonable as an architecture for the core application implementation. Instead, the agent needs local intelligence: a model running on the same infrastructure, with access to the same local context, capable of reasoning about infrastructure state, troubleshooting failures, and making autonomous decisions within policy guardrails.

Local applications leveraging local model inference

Deploying an ML enabled agent inside NVIDIA DGX™ Spark creates a predictable and seamless way for distributing the agent client application. Now, when a deployment fails in a private Kubernetes cluster, the agent's AI can finally leverage its full capabilities. It can query the cluster state, examine pod logs, check resource quotas, review recent changes, and diagnose the root

cause, all locally, without sending proprietary infrastructure details to external services. It can suggest fixes, automatically retry with adjusted parameters, or escalate to human operators with context-aware explanations.

The intelligence runs where the execution happens, and more importantly as an integrated part of the agent code base, with zero dependency on external connectivity and availability.



Reliable machine learning at the edge is not unique to Quali's products. In retail, agents could handle personalized customer interactions understanding queries in natural language, searching local inventory, checking real-time supply chain data, offering alternatives, processing transactions, and learning from interactions all without sending customer data to external services, with the information available on the client hand devices.

Imagine a retail chain with 500 locations, each running a DGX Spark unit hosting a local agentic assistant. Each agent handles customer interactions, inventory queries, and point-of-sale assistance using local context current inventory, regional preferences, store-specific data. The agents operate autonomously, without constant cloud connectivity, ensuring customer experience isn't dependent on network reliability.

Not every organization wants to or can legally use cloud-based AI services. Financial services firms may have regulatory restrictions on external API calls. Defense and government agencies require air-gapped deployments. Healthcare organizations face HIPAA and patient privacy constraints. Manufacturing companies worry about IP leakage when proprietary processes are visible to external inference APIs.

With agentic workflows becoming ubiquitous, applications cannot afford core functionality to depend on connectivity and security requirements that may not be possible for some accounts and industries. Standard edge devices can't handle it as model inference requires resources that are non-standard for many customers. An AI ready appliance , at the edge, in a form factor that fits the physical and operational constraints of real-world deployments can help unblock application business process access to ML models.

Distributed Agentic Architecture

With the new DGX Spark, we found a reliable medium for deploying the local version of our orchestration agent. Able to leverage our LLM models and agentic workflows in customer environments that previously did not allow it.

The future of AI isn't just smarter models in bigger data centers. It's also capable intelligence distributed to where the work happens, running locally, making autonomous decisions in real time. The NVIDIA DGX Spark makes that future deployable, that super-computer in a box, can also become a simple way to package your remote, agentic, software components.