



Container Platform

Capsis



Contents

1.	Overview	02
2.	Business Impact	05
3.	Core Capabilities Multi-Cluster Orchestration Helm · Add-on Management GPU · NPU Scheduling	07
4.	Observability, Security & Operations Network Policy Workload Isolation	11
5.	Works with	14

1. Overview



A container platform built on enterprise Kubernetes for AI workloads.

Capsis brings multi-cluster orchestration, governed container operations, and AI-optimized scheduling into one environment. Run Paxis agents, Metis inference services, and Maxis training pipelines without building and operating Kubernetes yourself.

01 One control surface for every cluster

Operate Kubernetes clusters spread across environments and geographies from one place. No different tooling per cluster — the same procedures and the same screens apply everywhere.

02 Governed container operations

Image management, add-on rollout, and configuration changes follow set procedures and leave a record of what was deployed where.

03 A foundation that understands AI workloads

Namespace isolation, quotas, and topology-aware placement let training, inference, and agent workloads coexist without interference.

04 Inside an environment you control

Everything runs in your data center, air-gapped networks included, under your own network policy and security controls.



Built so every AI platform runs on one shared container foundation.

Kubernetes is the de facto runtime for enterprise AI workloads. Capsis delivers that runtime as a product, so Paxis agents, Metis inference services, and Maxis training pipelines run on one shared foundation instead of each building its own.

AI workloads ask for something different: topology-aware placement, workload isolation, and traceable deployment history are built in from the start.

01 One foundation shared by every platform

Paxis, Metis, and Maxis share one execution foundation instead of each holding its own. Adding a platform does not mean rebuilding it.

02 One way to operate, however many clusters

The same procedures and the same management screens apply as clusters multiply, across environments and regions alike.

03 Execution boundaries per workload

Training, inference, and agent workloads are divided by namespace and resource quota, coexisting on one foundation.

04 Control and audit in the base build

Access control, image provenance, and deployment history are included from the start, with no extra tooling to bolt on.

2.

Business Impact



One container foundation: faster AI adoption, steadier operations.

Capsis consolidates the execution foundation Paxis, Metis, and Maxis depend on, replacing the Kubernetes each team built differently with one governed platform.

Your teams spend less on keeping infrastructure alive, and one operating standard reaches every business unit and environment.

01 Faster AI platform adoption

The container foundation is already in place: inference, training, and agent platforms skip infrastructure setup, and adding one never means rebuilding it.

02 Less infrastructure upkeep

There is no per-team Kubernetes to maintain, so platform engineers can move to service development.

03 Consistent operations

Every business unit and environment runs the same controls, network policy, and deployment procedure, so fewer failures come from environment differences.

04 Better use of AI infrastructure

Topology-aware placement and resource quotas reduce idle accelerators and keep the GPUs and NPUs you own on real work.

05 Isolation that meets regulation

Regulated workloads are separated by network and by resource, and every container operation leaves an audit trail. The same platform also covers deployments that must stay inside a government-controlled environment.

3.

Core Capabilities



Multi-Cluster Orchestration : One control surface for every cluster

Operate Kubernetes clusters across environments and regions from one control surface. Workloads route by resource availability, topology, and policy.

Unified cluster inventory

See the status, version, and resource usage of every cluster you run in a single list.

Per-cluster resource management

Select a cluster and manage its workloads, services, storage, configuration, and policies in the same screen layout.

Consistency across environments

Clusters on-premises and in the cloud follow the same operating procedures and the same management screens.

Policy-based workload placement

Resource availability, topology, and operating policy decide which cluster a workload lands on.



Helm · Add-on Management : Controlling the component lifecycle

Deploy, configure, and update platform components and the dependencies your AI workloads need, keeping every cluster on the same configuration.

Standardized installation

Install the apps and operators you need through a set input form, then review the result and the resources it created.

Manage what is installed

List what is installed on which cluster at which version, then change its configuration or remove it.

Consistent configuration across clusters

Keeping the same components at the same version on every cluster reduces failures caused by environment drift.

Traceable deployment history

Installs, updates, and removals are recorded, so you can see what changed a configuration and when.



GPU · NPU Scheduling : The right workload on the right hardware

Scheduling understands GPU and NPU topology, so training jobs, inference services, and agent runtimes land on the right hardware without manual affinity rules.

Topology-aware placement

Accelerator interconnect topology decides where communication-heavy jobs such as distributed training are placed.

Placement by workload profile

Training, inference, and agent workloads have different resource profiles, so different placement rules apply to each.

No manual affinity rules

Workloads land on suitable hardware without anyone writing node selection rules for each one.

Better accelerator utilization

Idle accelerators go down and the GPUs and NPUs you own stay on real work.

4.

Observability, Security & Operations



Network Policy : Traffic control shaped for AI environments

Define and enforce network policies across clusters, controlling ingress, egress, and east-west traffic between agent, inference, and training workloads.

Ingress control

Allow or block traffic entering the cluster from outside, service by service.

Egress control

Restrict outbound traffic from workloads so data does not leave over an unapproved path.

East-west traffic control

Traffic between Paxi agent environments, Metis inference services, and Maxis training workloads is allowed only as far as needed.

One policy standard everywhere

Rather than writing a policy per cluster, the same standard applies across every environment.



Workload Isolation : Keeping platform workloads from interfering

Namespace isolation, resource quotas, and tenant-aware scheduling let Paxos, Metis, and Maxis workloads coexist without contention or interference.

Namespace isolation

The Kubernetes namespace is the unit of isolation, dividing resources and permissions by platform and by team.

Resource quotas

CPU, memory, and GPU limits per namespace keep any one workload from taking over the resources.

Tenant-aware placement

Placement follows the tenant a workload belongs to, so work from different organizations does not collide on one node.

Resource usage visibility

Review resource usage by namespace and by cluster, and adjust the allocation policy.

5.

Works with



One container runtime foundation shared by Paxis, Metis, and Maxis.

Capsis holds the container runtime foundation in the Infrastructure layer. The Metis and Maxis platforms run on clusters Capsis provides.

CONTAINER PLATFORM (CAPSIS)

Kubernetes

PRIVATE CLOUD PLATFORM (AEGIS)

VM · Bare Metal

01 Aegis + Capsis

Infrastructure runtime — deploy alongside Aegis compute and storage in the same infrastructure layer, or run it on its own.

AI PLATFORM (METIS)

ML / LLMops Agent Ops

CONTAINER PLATFORM (CAPSIS)

Kubernetes

AI TRAINING PLATFORM (MAXIS)

End-to-end training

CONTAINER PLATFORM (CAPSIS)

Kubernetes

02 Metis + Capsis

Inference runtime — Metis model serving and inference endpoints run on Capsis clusters.

03 Maxis + Capsis

Training runtime — Maxis model development and training pipelines run on Capsis clusters.



One container runtime foundation shared by Paxis, Metis, and Maxis.

Capsis holds the container runtime foundation in the Infrastructure layer. The Metis and Maxis platforms run on clusters Capsis provides.

AGENT PLATFORM (PAXIS)

Thaki Cloud Application

CONTAINER PLATFORM (CAPSIS)

Kubernetes

04 Paxis + Capsis

Agent runtime — the Paxis Agent Builder, RAG, and MCP Tool Hub run on Capsis clusters.

SIGNUM

Unified Control Plane

05 Signum

Shared foundation — access control, operational logs, alerts and audit, security policy, and infrastructure build; included in every configuration.

CONTAINER PLATFORM (CAPSIS)

Kubernetes

06 Capsis

Capsis is available as a standalone product.