



How Dinova built a Scalable and Resilient Managed Kubernetes Service

**Empowering Cloud Providers with
Sveltos, Kamaji – and now kMetal**

The Cloud Provider Challenge: Managed Kubernetes at Scale

COMPANY SUMMARY

- **Company name:** Dinova S.R.L.
- **Website:**
<https://www.dinova.one>
- **Headquarters:** Italy
- **IT Environment:** Kamaji and Sveltos on vSphere, evolving to kMetal on bare metal
- **Industry:** Digital Innovation / Cloud Service Provider (CSP)
- **No. of employees:** ~350

HIGHLIGHTS

- Provisioning cut from 1-2 days to minutes with kMetal
- Control plane provisioned in under 16 seconds
- Zero shared resources across fully isolated tenant clusters
- Sveltos auto-restores critical resources after accidental changes
- No separate hypervisor layer – tenant clusters run on existing bare-metal servers

Dinova is a strategic innovation partner that brings together human and artificial intelligence to transform how organizations think, decide and grow. Headquartered in Bologna with additional offices in Milan, Rome and Lugano, Dinova operates as a hub of thinking, practice and reference. A Google Cloud Premier Partner since 2013, it was recognized as Google Cloud Country Partner of the Year 2026.

With over 3,500 projects delivered and more than 2,000 clients served, Dinova has established itself as a trusted partner, part of the Maggioli Group.

Dinova's comprehensive service portfolio spans four core competence centers: Cloud, Solutions (AI, data and intelligent platforms), Cybersecurity, and Experience.

This case study draws from both a 2026 interview with Dinova Cloud Architect Daniele Scattaglia and an earlier blog article by Simone Morellato, Co-Founder & CMO at Project Sveltos.



“kMetal delivers impressive flexibility and simplicity while staying true to the Kubernetes standard. Clastix support is highly responsive, offering a direct line to the developers.”

–Daniele Scattaglia
Team Leader of Cloud Native Department

CHALLENGES

As a CSP, Dinova's tenant clusters needed to be:

- **Isolated** – each customer (tenant) must have a dedicated cluster with no shared resources.
- **Transparent** – provisioning and management of nodes and resources should be seamless for the customer.
- **Resilient** – critical infrastructure resources must remain intact, even with customer actions.
- **Scalable** – deploying and managing numerous clusters efficiently is crucial.

Most traditional Kubernetes management solutions were not suitable for Cloud Providers, as they missed the key features needed to fulfill these requirements: ensuring resource consistency while preventing customer disruptions.

Dinova's managed Kubernetes service did not start with Kamaji. The team had been running customer clusters on another product, but needed to remove customer access to the control plane entirely and to standardize on fully conformant, upstream Kubernetes. Kamaji let them hide the control plane from tenants and deliver compliant Kubernetes as a service.

Tips

Management Cluster Reliability – since both Kamaji and Sveltos run in the management cluster, ensure it is highly available and resilient, with multiple control plane nodes and regular backups.

Resource Monitoring – with *ContinuousWithDriftDetection*, Sveltos can add load to the management cluster; tune its resources to handle this at scale.

Customer Education – educate customers about cluster boundaries; for example, resources like the CNI are managed by Sveltos and should not be modified.

SOLUTION

To address these challenges, Dinova adopted **Kamaji for control plane management** and **Sveltos for resource provisioning and drift detection**.

Step 1 – Kamaji for Dedicated Tenant Clusters

Kamaji is a Kubernetes Control Plane Manager that allows service providers to **run tenant cluster control planes as pods within a central management cluster**, rather than on dedicated machines per tenant. This approach, often called a "**Hosted Control Plane**," is specifically designed for CSPs/MSPs and is based on the same underlying principles used by larger hyperscalers. It offers several benefits compared with more traditional solutions:

Cost Efficiency – running control planes as pods instead of virtual machines significantly reduces the infrastructure footprint. Kamaji eliminates the need for separate control plane nodes for each tenant cluster, saving on compute resources.

Scalability – Kamaji makes it easy to deploy hundreds or thousands of tenant clusters. Benchmarks show that Kamaji can provision a control plane in under 16 seconds and reconcile 100 control planes in less than 150 seconds.

Isolation – each tenant's control plane runs in its own isolated namespace within the management cluster, ensuring hard multi-tenancy. Tenants consume their Control Plane as a Service, with no visibility into any other tenant's clusters.

Dinova installed Kamaji via Helm, following the official documentation, and configured it to use a dedicated datastore to store the state of each tenant cluster. For each customer, they create a *TenantControlPlane* resource, which Kamaji uses to spin up one or more dedicated control plane pods. The worker nodes for each tenant cluster are then joined to this control plane, **running on separate virtual machines or bare metal instances**.

This setup allows Dinova to provide each customer with a fully compliant Kubernetes cluster that feels like a dedicated instance, while managing the control planes efficiently behind the scenes.

Public Cloud

Hypervisor

Edge

Kubernetes

Step 2 – Sveltos for Provisioning Nodes and Resources

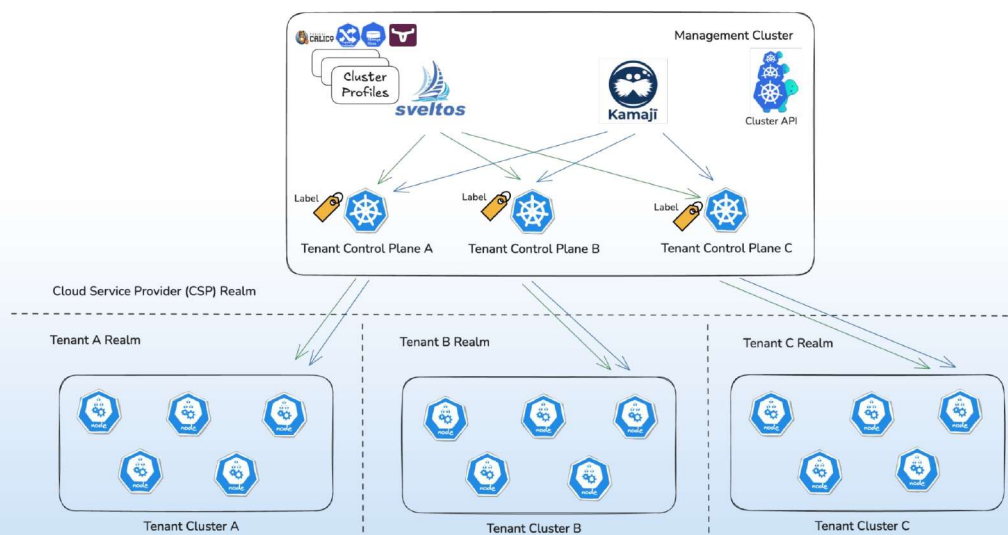
While **Kamaji** handles the control planes, **Sveltos** manages the provisioning of infrastructure resources across the tenant clusters. Sveltos is an open source Kubernetes add-on controller that runs in the management cluster and deploys resources to managed clusters (in this case, the tenant clusters created by Kamaji).

Dinova uses *ClusterSelector* labels to target specific tenant clusters – for example, labelling each tenant cluster with metadata like `customer=acme` or `env=production` – and Sveltos deploys resources only to clusters matching those labels. It integrates with Cluster API for worker-node management: Dinova uses Sveltos alongside Cluster API to provision worker nodes, and Sveltos ensures that additional resources like networking add-ons and storage solutions are deployed to those nodes.

Combined, Kamaji spins up the control plane, Cluster API provisions the workers, and Sveltos deploys the necessary resources – all without the customer needing to interact with the underlying infrastructure.

Step 3 – Sveltos for Resource Consistency

One of the biggest risks in a managed Kubernetes service is that customers might accidentally delete or modify critical infrastructure resources, such as networking components, storage classes, or RBAC policies, that are required for the cluster to function properly. Dinova relies heavily on Sveltos' *syncMode: ContinuousWithDriftDetection* feature. Sveltos defines the critical resources that must always exist in each tenant cluster via a *ClusterProfile*, continuously monitors the state of these resources, and automatically re-deploys anything a customer deletes or modifies. This lets Dinova confidently delegate cluster access to customers without worrying about accidental changes breaking their clusters. If a customer removes a critical resource, Sveltos silently restores it, ensuring uninterrupted service.



Ongoing Benefits with Sveltos and Kamaji: Unprecedented Efficiency, Transparency, and Scalability

By combining Kamaji and Sveltos, Dinova achieved several key benefits as a cloud provider:

Operational Efficiency – Kamaji's hosted control plane approach reduces the cost and complexity of managing tenant clusters at scale. Dinova no longer needs dedicated control plane nodes for each customer, which saves on infrastructure costs and simplifies operations.

Transparency – Sveltos allows provisioning resources behind the scenes, giving customers a seamless experience. They get a fully configured Kubernetes cluster without needing to understand or interact with the provisioning process.

Resilience – the *ContinuousWithDriftDetection* feature ensures that critical infrastructure resources are always present, even if a customer accidentally deletes them. This reduces support tickets and improves reliability.

Scalability – Kamaji's ability to manage hundreds of control planes and Sveltos' ability to deploy resources across a fleet of clusters make it easy for Dinova to scale their managed Kubernetes service as their customer base grows.

Security and Isolation – Kamaji's hard multi-tenancy ensures that each customer's cluster is fully isolated, with no shared resources. Sveltos' multi-tenancy features (e.g., *ClusterProfile* and *Profile* resources) further support this by allowing Dinova to manage resources for each tenant independently.

Original blog article by Simone Morellato, Co-Founder and CMO of Project Sveltos, from which part of this case study was adapted:

<https://clastix.io/post/empowering-cloud-providers-with-sveltos-and-kamaji>

THE NEXT CHALLENGE: Moving Beyond VMware

With Sveltos and Kamaji in production, Dinova had solved control-plane management and multi-tenancy. The next challenge sat one layer down – in the infrastructure itself.

Like many providers, Dinova needed to move off VMware. Continuing to build on a licensed, proprietary hypervisor no longer fit a strategy that is otherwise fully cloud-native.

Kamaji had already removed the per-tenant control-plane machines, but the worker nodes still ran on that separately licensed hypervisor, so infrastructure cost and lock-in remained.

Dinova also wanted the platform to be as automated as possible, so customers could get working environments without needing hands-on help with the clusters. In the earlier setup, standing up a single cluster took 1-2 days.

A growing share of Dinova's work is with public administration, which prefers local service providers – and private customers increasingly ask for the same. Digital sovereignty is becoming a requirement, not a nice-to-have.

THE kMETAL SOLUTION: Cloud-native, all the way down to bare metal

kMetal answers this. Built on Kamaji – the same hosted-control-plane engine Dinova already relied on – kMetal extends the cloud-native model all the way down to the infrastructure layer.

It removes the separate hypervisor layer, running tenant workloads on kernel-native KVM, so Dinova can operate a self-contained, Kubernetes-based platform and stop paying for a proprietary hypervisor. Through the KubeVirt CSI driver, Dinova can bring their own storage. And because kMetal is a bare-metal platform, it runs on the servers Dinova already operates – no hardware changes are required to adopt it, unlike a migration from VMware to another virtualization product.

The logo for kMetal by CLASTIX. The word "kMetal" is in a large, bold, sans-serif font, with the "k" in red and "Metal" in dark blue. Below it, the words "by CLASTIX" are in a smaller, dark blue, sans-serif font.

About DINOVA

Dinova is a strategic innovation partner that brings together human and artificial intelligence to transform how organizations think, decide and grow. Headquartered in Bologna with additional offices in Milan, Rome and Lugano, Dinova operates as a hub of thinking, practice and reference. With over 3,500 projects delivered and more than 2,000 clients served, Dinova has established itself as a trusted partner, part of the Maggioli Group. From cybersecurity solutions that protect identities and foster security culture, to experience design that creates authentic data-driven relationships, to cloud transformation that evolves with organizational intelligence, Dinova delivers measurable business value by turning AI into an engine of structural change and operational excellence.

About PROJECT SVELTOS

Project Sveltos is an open-source Kubernetes fleet management controller designed to simplify the deployment, orchestration, and lifecycle management of add-ons, applications, and configurations across multi-cluster and multi-tenant environments. Operating from a central management cluster, Sveltos utilizes a declarative, label-based approach to automatically target and configure managed clusters, making it an ideal extension for GitOps pipelines using tools like ArgoCD and Flux. With native support for Helm, Kustomize, and raw YAML, it features advanced capabilities including automated drift detection, event-driven remediation, and a secure pull-mode architecture tailored for edge and firewalled environments.

About CLASTIX

CLASTIX builds enterprise-grade Kubernetes infrastructure through hosted control planes and hard multi-tenancy – solving cluster sprawl, fragile isolation, and runaway infrastructure costs at scale. With deep open source credibility – including the donation of Capsule to the CNCF – the kMetal platform is purpose-built for cloud providers, MSPs, and platform teams that demand real isolation, efficiency, and scale. Customers include NVIDIA and Mistral AI.

Contact us

Please contact us to request a quote or learn more about our support offerings.



CLASTIX SRL
V. Francesco Caracciolo, 17
80122 Naples, Italy

VAT No. IT11183320966
REA: NA-1152378
© 2026 Clastix | All Rights Reserved

hello@clastix.io
www.clastix.io
<https://github.com/clastix>