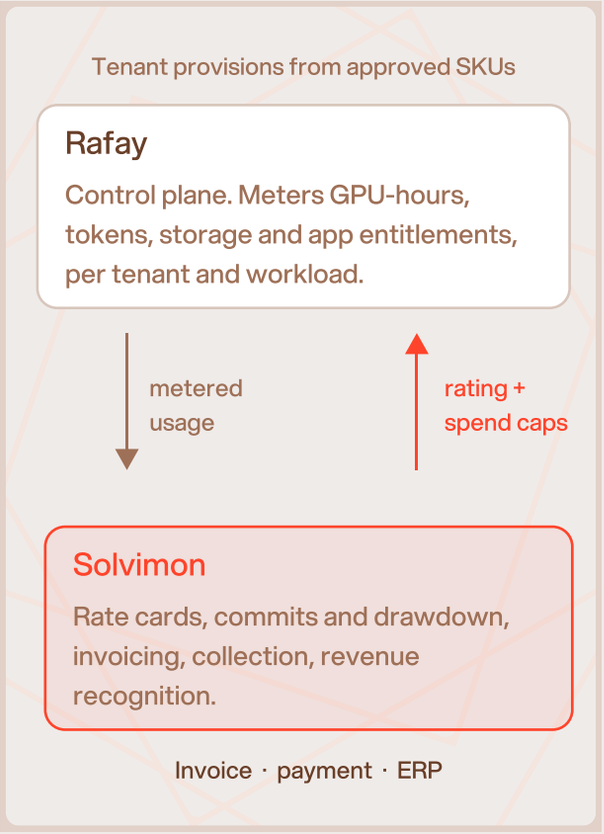
solvimon

The commercial layer for GPU clouds

Rafay runs the control plane over your GPU fleet and meters what every tenant consumes. Solvimon turns that into rate cards, invoices, and cash. The two connect over APIs, so an operator configures the commercial model rather than building one for it.

Solvimon is the billing layer for companies with complicated pricing and more than one entity. Quoting, metering, rate cards, invoicing, and cash flow on one platform, built on technology that processed more than €1 trillion each year.



€1T+
yearly volume processed on the tech behind Solvimon

2-way
usage flows in, rate cards and spend caps flow back

1 invoice
per entity and currency, every SKU on it

Why this matters for GPU clouds

Getting GPUs into a rack is now the easy part. Getting paid accurately for what runs on them is where operators stall. Capacity goes live, tenants start consuming, and the commercial model ends up in a spreadsheet maintained by whoever has the most patience.

Most operators are metering for capacity planning rather than for billing. The data exists, just not in a shape anyone can invoice from. That stays a small problem right up until someone asks what the cluster earned last quarter.

If these sound familiar, you need Rafay and Solvimon

- “Our rate card is dollars per GPU-hour, and that’s the only thing we can compete on.”
- “Customers prepay two-year commits and we track the drawdown in a spreadsheet.”
- “We sell bare metal, Kubernetes, and managed inference, and each bills differently.”
- “We’re standing up a second region and a second legal entity next quarter.”
- “Finance asks what the cluster earned and the answer takes three weeks to assemble.”

solvimon

Where Solvimon fits in GPUaaS

If you're already metering with Rafay

Rafay captures usage per organization, tenant, and workload across bare metal, VMs, Kubernetes, and SLURM, plus tokens from the serverless inference API. Solvimon reads that through the Rafay APIs, rates it against each customer's contract, and produces the invoice. Nothing new to instrument.

If you're selling commits and prepaid capacity

Prepaid commitments burn down against mixed consumption across SKUs, with rollover, expiry, and overage handled in the ledger rather than a side sheet. The same delivery data produces the ASC 606 recognition schedule, so deferred revenue converts on evidence instead of on someone's spreadsheet.

Division of labour

What Rafay meters, and what Solvimon does with it.

What you sell	Rafay meters…	Solvimon handles…
GPU compute	Utilized GPU-hours per SKU and org	Rate cards per generation and tier, commit drawdown
Bare metal	Instance hours, profile, quota	Reserved vs on-demand, minimums
Kubernetes	Job and namespace consumption	Chargeback per team, internal or external
Inference	Tokens in and out, per model	Per-token rate cards, credits and wallets
Storage	Capacity and tier per tenant	Metered lines on the same invoice
AI apps	App usage rights and entitlements	Subscriptions, seats, revenue share
Contracts	Org and tenant boundaries	Quoting, multi-entity, multi-currency, e-invoicing
Payments	—	Any PSP, dunning, cash application, ERP

Every row above lands as a line on the same invoice, per entity, in the customer's currency.

solvimon

What Rafay brings

Self-service consumption from approved SKUs

Rafay overlays the fleet with a portal, API, and CLI so tenants provision GPU environments from standardized, version-controlled SKUs. A SKU defines the compute profile, GPU slicing including MIG, storage, policy, and lifecycle, with quota enforced in real time. No tickets, no ad-hoc workflows.

Multi-tenancy, governance, and metering

Hierarchical RBAC, namespace isolation, policy enforcement, and centralised audit logging let one fleet be shared across teams and tenants. Consumption is captured per organization, tenant, team, and workload on bare metal, VMs, Kubernetes, and SLURM, alongside one-click NVIDIA NIM, NeMo, and Run.ai and a token-metered serverless inference API.

What's automated

The integration runs in both directions over APIs.

Step	What happens	Direction
Catalog	Compute profiles and SKUs defined once, created in Rafay programmatically	Solvimon → Rafay
Accounts	Each Rafay organization maps to a customer account and contract	Rafay → Solvimon
Usage	Metered consumption pulled per profile, organization, and workload	Rafay → Solvimon
Rating	Usage validated and rated against each customer's contract terms	Solvimon
Entitlements	Rate cards, commit balances, and spend caps written back	Solvimon → Rafay
Invoicing	Invoice per entity and currency, e-invoicing where required	Solvimon
Collection	Payment on any PSP, dunning, and cash application	Solvimon
Accounting	Recognition schedule from committed and delivered amounts	Solvimon → ERP

None of this involves a CSV export or a month-end reconciliation.

solvimon

You need Rafay and Solvimon

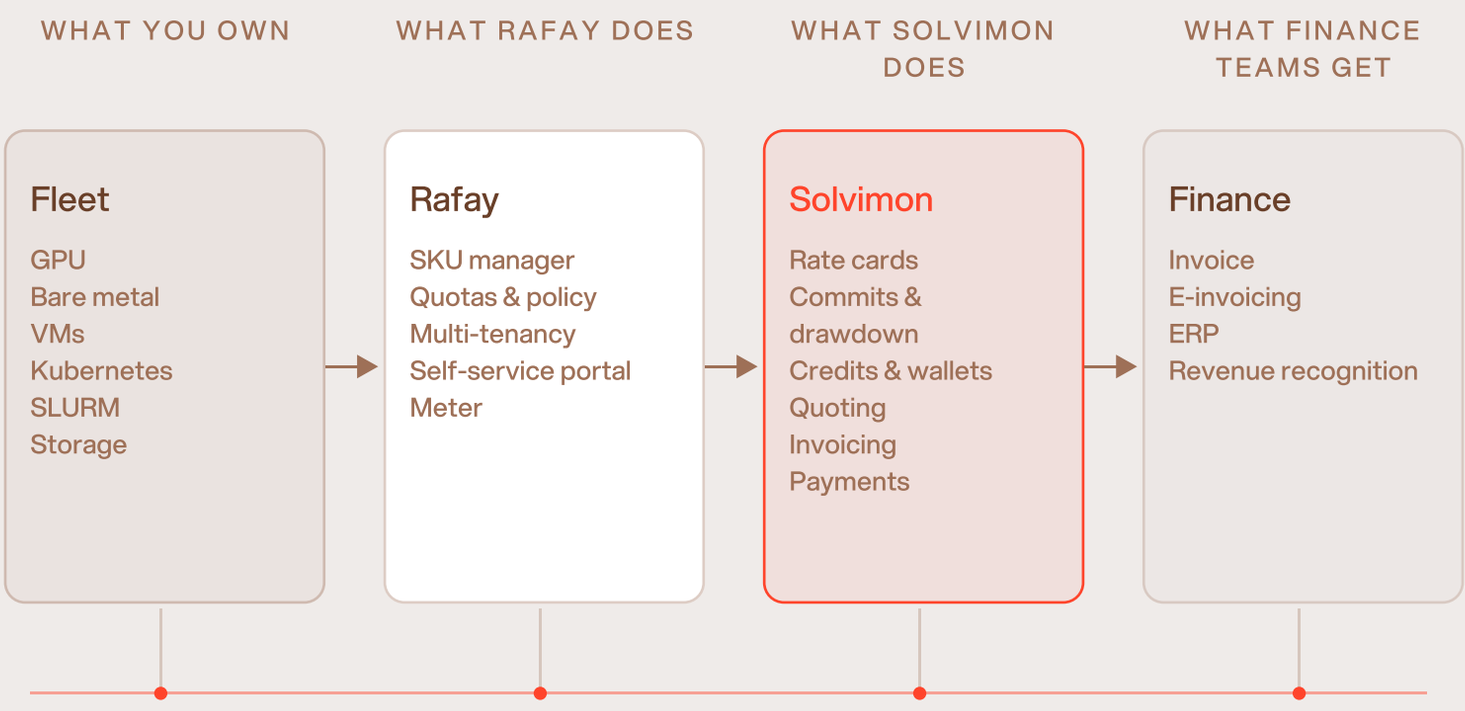
The same metered event drives the invoice, the entitlement in Rafay, and the recognition schedule. Because nothing has to be reconciled between three versions of what a tenant consumed, every invoice line traces back to the workload that produced it.

Lots of room for expansion

If you're starting on bare metal and later moving to managed inference, model hosting, or marketplace apps, you can price the new layer on the same system. With Solvimon and Rafay, getting a new SKU to its first invoice is a configuration change rather than an engineering project.

From GPUaaS to cash

With Solvimon and Rafay, one workload can be traced from the fleet to the invoice line.



an H100 job → utilized GPU-hours → rated against a two-year commit → one line on one invoice

Rafay runs the fleet. Solvimon runs the commercial model.
rafay.co · solvimon.com

Mark Vermeulen
mark.vermeulen@solvimon.com

RAFAY

solvimon