Fractal Cloud

Use Cases for System Integrators

Accelerating customer success while improving delivery efficiency and profitability

www.fractal.cloud

Accelerated and Consistent Customer Onboarding

Provisioning the foundational infrastructure of a new enterprise customer (landing zone, baseline services, observability, security, etc.), across every engagement.

Today

- IaC modules forked and adapted for each new customer
- 4 to 8 weeks from contract to a production ready environment
- The 20th landing zone is not the same as the 1st: variants accumulate over time

With Fractal Cloud

- Each recurring pattern becomes a Fractal: versioned, parameterized, ready to instantiate
- The next engagement starts from the catalogue, instead of from a fork of the previous customer
- Each engagement adapts the Fractal through parameters, within guardrails defined by the catalogue

Value

- **Efficiency**: foundational setup goes from weeks to days
- **Consistency**: every onboarding meets the same standard, regardless of who runs it
- **Reuse**: the reference architecture is built once and improved over time, instead of being rewritten on each engagement

Day-2 Managed Services Across a Customer Portfolio

Recurring infrastructure work (CVE patches, version bumps, policy updates, drift control, etc.) executed consistently across every customer account, on time, without eroding margin.

Today

- IaC repository per customer, pipeline per customer, ticketing on top
- No consolidated portfolio view, no continuous drift detection, no governed rollout across customers
- Per customer variants tend to accumulate, and know how on each customer concentrates in whoever built it, slowing rotation and peer review

With Fractal Cloud

- A versioned Fractal catalogue replaces per customer IaC sprawl
- Continuous reconciliation detects and remediates drift, including changes made manually in the customer's cloud console
- Governed rollouts across the portfolio (canary, batch, full) from a single view

Value

- **Efficiency**: the same patch rolled out across the portfolio in a fraction of the time
- **Margin**: effort per customer becomes more predictable, so managed contracts can be priced with smaller risk buffers
- **New offerings**: SLAs on drift response and recurring governance reports become deliverable

Delivery of Regulated Services

Delivering infrastructure to customers operating under regulatory frameworks (NIS2, DORA, PCI, ISO 27001, etc.), where the rules shape the architecture from the start.

Today

- Compliance is handled as a separate process: review boards, manual checklists, retrofits after provisioning
- Compliance work concentrates on a handful of specialists, and their availability caps how many regulated customers the SI can serve in parallel
- Audit evidence is gathered manually after the fact, by querying cloud consoles, reading logs, and reconciling spreadsheets

With Fractal Cloud

- Compliance controls (encryption, RBAC, network policies, audit logging, tagging, etc.) are encoded inside the Fractal
- Every Live System inherits the standard automatically, and deviations are blocked at provisioning
- Continuous reconciliation keeps environments aligned after go live, and evidence is queryable from the platform

Value

- **New market**: the SI can take on regulated customers with the standard delivery team, instead of relying on a specialist one
- **Margin**: compliance overhead per customer drops, because the controls are pre built and enforced by the platform
- **Credibility**: the SI brings into every new customer the controls already validated on previous engagements; experience compounds in the catalogue instead of being rebuilt each time

Workload Migration Across Clouds, On-Prem & Sovereign

Moving customer workloads from one cloud to another, from on prem to cloud, or to a sovereign cloud, with a repeatable methodology rather than as a custom project every time.

Today

- Each cloud target requires its own IaC, because the managed services on AWS, Azure, GCP, OCI and others are different products with different APIs
- A migration project therefore means rewriting the modules for the new target, rather than simply redirecting them
- The SI carries specialists per target, and the effort to support a new destination grows with each one

With Fractal Cloud

- One Fractal definition resolves to different cloud specific implementations through the Component, Service, Offer model
- The same architecture deploys on Azure, AWS, GCP, OCI, sovereign providers or on prem
- Migration becomes instantiating the same Fractal on a different target, instead of rewriting it

Value

- **New service line**: repeatable migration projects, including cloud to sovereign
- **Margin**: the same expertise is reused across targets, and the SI avoids maintaining parallel IaC stacks
- **Customer outcome**: migration projects become predictable in time and cost

Modernizing the Existing IaC Estate

The SI has years of IaC (Terraform, Pulumi, etc.) in production. The investment is preserved, and governance, drift control and lifecycle are added on top without rewriting it.

Today

- Years of IaC, often forked across customers and engineers
- No central catalogue, no continuous drift detection, no controlled rollout across customers
- The automation handles provisioning well, but does not handle governance, lifecycle, or changes that need to be applied to many customers at once

With Fractal Cloud

- Existing IaC (e.g. Terraform) is imported into Fractals: the code stays, the governance layer is added
- Versioned catalogue, continuous reconciliation, controlled propagation across customers
- The SI keeps the IP they built and operates it from a single platform

Value

- **Efficiency**: the existing investment is retained, with no rewrite cost
- **New capability**: applying a change to many customers at once becomes a single controlled rollout instead of N separate apply runs
- **Phased adoption**: Fractal Cloud can be introduced one customer at a time as the catalogue grows