# Demo Feedback – P&G Perspective by Rodolfo

Feedback based on the demo and platform concept, with focus on:  
- How the current ‘pay only for what you use’ pricing interacts with cost governance.  
- How procurement and business controls can support long-term sustainability.

# Overall Impression

You’ve made an impressive demo so far. The platform delivers on what it promises: fast provisioning, strong (GDPR) compliance, and a developer-friendly setup. From a business and procurement perspective, your pricing model is clear and attractive. The structure (modular, transparent, and pay-as-you-go) already reduces many of the risks found in early-stage infrastructure products. Still, there are several hidden cost variables that, if left untracked, could challenge your ‘fixed or predictable pricing’ model at scale.

# Current Model: Strengths & Implications

|  |  |  |
| --- | --- | --- |
| Strength | Why It Matters | Impact |
| Transparent component pricing (compute, DB, cache, load balancer, bandwidth) | Supports cost traceability; each resource is priced explicitly. | Increases pricing trust and cost control. |
| Bundled bandwidth (20TB included) | Absorbs volatility from normal traffic patterns and simplifies billing. | Reduces billing disputes and improves predictability. |
| Predefined resource configurations (e.g. 2–10 replicas, 2vCPU/4GB) | Provides cost predictability for both you and the customer. | Easier to forecast cost-to-margin ratio. |
| Hetzner EU-based infrastructure | Keeps compliance costs low and ensures stable, competitive pricing. | Maintains GDPR alignment and vendor trust. |
| No minimum contract | Reduces friction for early adopters and accelerates growth. | Encourages onboarding and experimentation. |
| Auto-scaling with min/max thresholds | Balances performance with cost management. | Optimises resource use dynamically. |

# Key P&G Considerations

While major cost categories are already visible, several variable cost drivers remain that could affect long-term margin stability.

|  |  |  |
| --- | --- | --- |
| Category | Business Risk | Recommended Governance / Technical Action |
| Auto-scaling (2–X replicas) | Sustained high traffic can increase costs while pricing remains fixed. | Implement replica usage tracking per tenant; flag sustained >80% usage for upgrade recommendations. |
| Bandwidth (20/X TB included) | Heavy workloads can exceed limits and impact margins. | Add real-time bandwidth alerts at 80%; show usage in dashboard; charge for overages. |
| Storage Growth (Postgres 50 GB) | Data accumulation increases costs over time. | Automate storage tracking and trigger plan upgrades at thresholds. |
| Backup & Redundancy | Retention policies expand storage footprint invisibly. | Track backup volume per tenant; apply tiered retention (7/30/90 days). |
| Vendor Dependency (Hetzner) | Vendor price changes directly affect margins. | Sync vendor pricing via API; maintain 4–5% buffer and alert on cost shifts. |
| Support & Monitoring | More customers increase maintenance effort. | Log support time per tenant; integrate data into pricing model quarterly. |
| Configurable Resources | Customer scaling can erode margins if mapping isn’t dynamic. | Link configuration changes to internal cost tables and recalculate impact real-time. |

# Bridging Business & Technical Governance

Your next step is to make cost governance part of the technical foundation, so pricing predictability becomes a system feature.

|  |  |
| --- | --- |
| Business Goal | Technical Translation |
| Maintain predictable margins | Build a Cost Telemetry Service: track compute, storage, and bandwidth per app/config. |
| Detect cost drift early | Create alerts when usage exceeds thresholds (CPU > 90%, bandwidth > 80%). |
| Support smart scaling decisions | Log replica usage and provide visual insights for plan design. |
| Stay vendor cost aware | Store vendor SKUs and pricing in a local catalog with version control. |
| Enable procurement reporting | Auto-generate usage and vendor reports for GDPR and enterprise clients. |

# Business Summary

You’ve already completed 70% of the work: modular pricing, transparent cost drivers, and compliance integration. The remaining 30% lies in internal visibility and proactive cost governance. If these controls and telemetry layers are implemented, pricing remains simple for customers, margins stable for you, and governance readiness becomes part of the product’s DNA.

# Recommended Steps

|  |  |  |
| --- | --- | --- |
| Layer | Purpose | Example Implementation |
| Cost Mapping | Define cost baselines per template (compute, DB, cache). | Store baselines in DB/YAML and link to telemetry metrics. |
| Monitoring & Thresholds | Detect overuse or margin erosion early. | Prometheus/Grafana dashboards with usage alerts. |
| Review & Update Cycle | Sync vendor pricing and usage data regularly. | Scheduled API pulling Hetzner or similar pricing; review quarterly. |

# Closing Note

Governance is a constant, even as the product evolves. The platform is starting to feel enterprise-ready in its design and foundation. The next step is to make internal cost awareness as strong as its external simplicity, turning predictable pricing into a technical differentiator and governance into sustainability.  
  
At this stage, it’s fine to rely on your hosting partner’s certifications (Hetzner or similar) for the infrastructure layer. However, working with a compliant partner does not automatically make your platform compliant. As you scale and start handling production data, establishing your own governance layer (policies, access controls, and ISMS) will be key.

Creating the necessary documents ( ToS, DPA, NDA, SLA, Framework agreements etc.) won’t be a problem. They can all be built to align and merge seamlessly with your business plan later on. Doing so builds credibility and confidence with future investors, showing that the platform stands on security, transparency, and long-term resilience.