

Optimising IT COSTS

Eliminating Waste. How do you choose your architecture to best fit your use case?

Note: Speak after everyone.

My area: use case & context important, provide examples

Eliminating waste is not necessary about choosing the right technology from the start, but instead be proactive about understanding running rates and setting up budgets.

You could optimise costs in different ways and what we need to think about Savings (difference between actual cost and budget), spend reduction (optimisation activities that cut back on cost), cost avoidance

It largely depends on the context. We need to understand business priorities and more important where or not the engineering teams understand these business priorities. For new service it could be time to market, for something that already attracts customer it could be focus on profitability. To eliminate waste require setting up budget, and make the business aware of it. There is a disconnect sometimes between what the business think about the cost and what it actually is. Make it clear what the actual cost to launch this service. Make the tech solve the business problem, and instrument the service with right observability. When in production, You will get the data insights, historical data, seasonality, running rates; And then the team could make a data driven decision and optimise architecture if needed.

Effectiveness vs Efficiency. How do you leverage observability to drive efficiency example root cause analytics?

Note: Speak second. This will likely get's into APM and profiling.

Note: My area; Add details

Worth to discuss reactive vs proactive approaches.

We use observability tooling both reactively and proactively. Reactively, it helps with root cause analysis by identifying issues quickly through logs, metrics, and traces. Proactively, we leverage anomaly detection to spot irregular patterns before they escalate into bigger problems. This balance improves **efficiency** by reducing downtime and response times, while boosting **effectiveness** by ensuring system reliability and performance over time. Gen AI and natural language processing helps us a lot. Instead of digging through tons of logs and metrics, just ask in plain english and you have potential root causes.

Effectiveness vs Efficiency. **Traditionally a lot of high-control organizations managed spending by introducing barriers for engineering teams to provision resources. What are some of the ways that you've reduced friction and maintained developer velocity while managing cloud spending?**

Note: Speak first

Note: My area; Can request resources for interfaces & environments to get approval for the budget. Could not explain back to the business on the business value of increased investment

At Holland and Barrett we have multiple models on how we control spending

Before I dive into detail, very high level answer is - we trust our engineers, automate whatever is possible, set anomaly detection for cost and unusual activities and forecasting. This is what we rely on.

I joined the company 3+ years ago, at the time we have a simple process in place with guardrails.

- **Self-Service Cloud Resource Provisioning:** This can be the most interesting approach. By default, we don't enforce any financial decisions before development begins or while teams are still in the design phase of their products—this responsibility is left to the teams. Instead, we operate with an approved, shared cloud budget (planned collaboratively with engineering leadership), and we heavily rely on automation to optimize costs. Automation helps teams manage their expenses by monitoring usage, detecting cost anomalies, and ensuring that the self-service process prevents the over-provisioning of resources. The idea is to promote golden path and self service SaaS offering where the cost model is understandable.
- **Team accountable for cloud spend.** The team is fully accountable for managing their cloud spend. They are granted admin access and the ability to provision any resources they need, whenever they need them. However, there are potential consequences to consider: the team or vertical is now responsible for managing costs, which may require expertise outside their core specialization. Additionally, blanket policies applied by the central team may no longer be relevant or effective in this scenario
- **Environments on demand.** For short-term initiatives, offering environments on demand can be the most practical approach. Once the initiative has received business approval, the team already understands the duration, high-level requirements, access needs, and budget constraints. The key benefit is that the team has the flexibility to either experiment with new technologies or stick to the established "golden path" for their work.
- **Self-Service Shared resources.**

Culture of ownership. When designing & architecting your tech stack how do you consider cost efficiencies?

Cost efficiency focus, Speak last. Note; transparent on costs.

Give teams the autonomy to choose the right tools and technologies for their needs.

The teams asks for autonomy, they design build, deploy and look after software in production. This imply that they should be accountable for costs as well. So we are moving from centralised cost model to decentralised, where teams and verticals are made accountable for they running costs. This is not just cloud costs. Consider licencing costs, 24/7 on-call support, maintenance and more. There are too many areas that needs to be considered, and the team should be aware of it.

Business is all about be profitable and burning cash isn't a common business model. If the team have no clear overview of the costs, how CIO for example could forecast IT costs in predictable manner.

- Start with basic cost tracking
- Work out the path to granular cost observability

Granular cost observability means tracking cost down the line to the tinies level. It's like playing hide and seek where you target to find all the players you not aware off. Imagine beeng able to see exactly how much a new feature cost business, every product in every region, how much each team spend from idea to operational support.

You can't be responsible for something you have no access too. And it's extremely important to make the costs visible and transparent.