ChatGPT

# Executive Summary

Energinet – Denmark's national electricity and gas **TSO (Transmission System Operator)** – is undergoing a cloud-driven digital transformation in a highly regulated, critical infrastructure context. Multi-cloud adoption (Azure, AWS, and GCP) has accelerated innovation but also introduced **unpredictable costs, low resource utilization, and governance challenges**. This FinOps report provides a comprehensive strategy to maximize cloud value while ensuring compliance (e.g. **EU NIS2** security directives, audit trails) and financial accountability. Key findings and recommendations include:

- **Adopt FinOps as a Cultural Practice:** FinOps (Finance + DevOps) is *not just cost-cutting* – it's about **driving business value from cloud** through cross-team collaboration and timely, data-driven decisions [1] [2]. We recommend Energinet formally embrace FinOps principles (e.g. *teams collaborate, everyone takes ownership, centralized enablement*) to instill a cost-aware engineering culture. This will break down silos between engineering, finance, and product teams and create shared ownership of cloud spend.

- **Establish a FinOps Operating Model:** We propose a **hybrid FinOps team structure** [3] [4] – a central FinOps Core Team that sets standards, provides tooling and governance, and **embedded FinOps Champions** within product/dev teams for domain-specific cost optimization. A RACI matrix clearly defines responsibilities (e.g. FinOps team accountable for tagging policy and discount management, product teams responsible for efficient resource use) [5] [6]. New governance forums (monthly cost reviews, quarterly optimization councils) will align stakeholders and accelerate decision-making.

- **Implement a Unified Cost Management Data Pipeline:** Energinet should build a **single source of truth** for cloud cost data across AWS, Azure, GCP and on-prem. This involves ingesting detailed billing data (AWS CUR, Azure Cost Management exports, GCP billing data) into a centralized data lake (e.g. Azure Data Lake or Snowflake), **normalizing it using FinOps FOCUS** (the open cost spec) [7] [8], and enriching with business metadata (tags, project IDs, cost centers). A robust tagging strategy is critical: we recommend mandatory tags (e.g. `BusinessUnit`, `Application`, `Environment`, `Owner`) on ≥90% of resources to enable accurate allocation and showback [9] [10]. This pipeline will provide timely (daily) cost visibility and support chargeback, unit cost calculations, and anomaly detection.

- **Optimize Cloud Costs Proactively:** We identified **10 key FinOps "failure modes"** currently causing waste (e.g. idle instances, overprovisioning, low RI/Savings Plan coverage, tagging decay, shared cost allocation gaps, siloed decisions). Our **Optimization Playbook** targets each with high-impact actions. For example, aggressive **rightsizing** of VMs and containers can cut compute costs ~20% by eliminating underutilization [11] [12]. Automating **shutdown of idle resources** (dev/test environments off-hours, orphaned volumes & IPs) and deleting unused storage snapshots will directly reduce waste (currently up to *32% of cloud spend is wasted* industry-wide [13]). Increasing **commitment-based discounts** (Reserved Instances, Savings Plans, Committed Use) to ~70-80% coverage of steady-state workloads can yield ~25-40% rate savings [14]. Other levers include **auto-scaling**, **spot instances** for non-critical workloads, **storage tiering** (archive cold data to cheaper tiers), **optimizing data egress** (via peering or CDN), and tuning **serverless and analytics jobs** for efficiency. Each optimization initiative is

quantified in a “Savings Backlog” with owner, expected savings, and prerequisites, to focus efforts on high-ROI items first.

- **Enhance Forecasting & Budgeting Discipline:** Today’s cost forecasting is inaccurate (>20% variance) due to dynamic usage and lack of granular data. We recommend implementing **driver-based forecasting** combining **trend analysis and unit economics**. For example, forecast cloud spend based on expected growth in data volume or transactions, with input from product roadmaps. Set up **monthly budgets and variance alerts** (using native tools like AWS Budgets, Azure Cost Alerts) so that budget overshoots trigger reviews [15]. Over time, integrate ML-based prediction (e.g. AWS Cost Anomaly Detection, Azure ML) to improve accuracy. The goal is to achieve <10% variance between forecast and actual by next year, improving predictability for Finance.

- **Leverage Automation and Policy-as-Code:** Embed cost controls into the CI/CD and cloud governance process. For example, use **Azure Policy and AWS Config** rules to enforce tagging compliance and prevent deployment of overly large instances in non-prod. Implement **OPA-based guardrails** (Open Policy Agent) or cloud-native policies to **flag unapproved resources** (e.g. a costly GPU machine launched outside of ML team) and enforce quotas. Automated anomaly detection should monitor spend patterns and alert within hours to any “cost spikes” beyond normal (e.g. a 50% daily increase) [16] [17]. Where possible, integrate cost checks into deployment pipelines (e.g. using Terraform guardrails or Infracost to show engineers the cost impact of their code changes in merge requests). This ensures cost optimization is continuous and not just a periodic cleanup.

- **Tooling Strategy – Build vs Buy:** Evaluate the **FinOps tooling landscape** to accelerate capabilities. Native cloud tools are being used today (e.g. AWS Cost Explorer is used by ~78% of FinOps practitioners [18]) but have limitations in multi-cloud analysis and customization. Third-party platforms (Apptio Cloudability, VMware Aria/CloudHealth, CloudZero, Flexera, etc.) offer advanced features: multi-cloud cost aggregation, showback/chargeback, automation, and rich reporting. We recommend a **hybrid approach**: in the short term, leverage native tools and open-source (e.g. Grafana dashboards on billing data) to prove quick wins; in parallel, initiate an RFP for an enterprise FinOps platform or augment the data lake for custom analytics. Key selection criteria include multi-cloud support, integration with our data warehouse, granular cost allocation, and compliance (on-premise or EU-hosting given GDPR). The tooling comparison matrix in Annex F provides a feature-by-feature evaluation. Any chosen tool must integrate with Energinet’s security and compliance framework (e.g. single sign-on, role-based access control for least privilege data access).

- **Key FinOps KPIs & Reporting:** To measure success, we will track a balanced scorecard of FinOps **KPIs**. Key metrics include: **Tag Coverage** (% of spend with complete tag attribution – target >95% [9]), **Waste Reduction** (e.g. % of resources with ≤5% utilization, aiming to halve this within 6 months), **Discount Coverage** (% of spend on RIs/SPs – target 70%+ of eligible spend), **Unit Cost trends** (e.g. cost per customer connection or per MWh balanced – aiming for cost per unit to flatten or decrease even as usage grows), **Forecast Accuracy** (actual vs budget variance, target <10%), **Anomaly MTTR** (time to detect and resolve cost anomalies), and **Engineer Action Rate** (what % of optimization recommendations are acted upon by teams). We will implement **dashboards** segmented by business unit and product – e.g. monthly cost reports showing each product’s spend against budget, and unit economics metrics that link cloud spend to business KPIs. Regular **showback** reports will be provided to product owners (finance transparency without forcing chargeback initially) to drive accountability through transparency. These KPIs create accountability and will be reviewed in monthly FinOps reviews.

- **12-Month Roadmap & ROI:** The roadmap is phased into **Quick Wins (0-3 months)** – establishing the FinOps team, turning on basic cost alerts, identifying "low-hanging fruit" optimizations like deleting idle resources; **Foundation (4-6 months)** – implementing the unified cost data platform, defining the tagging policy, initial training for engineers; **Acceleration (7-12 months)** – advanced automation (rightsizing bots, predictive scaling), rolling out chargeback, and refining unit economics for key services. We project **cloud cost savings of 15–25% in year one** from waste elimination and rate optimization, which on Energinet's cloud spend (~€X million/year) is approximately €Y million saved. Industry benchmarks (e.g. McKinsey) show mature FinOps practices can reduce cloud spend by *20–30%* [19] . Beyond cost savings, FinOps will deliver **non-financial value**: improved planning accuracy, faster delivery (by reallocating saved budget to innovation), and strengthened compliance (via clear audit trails of cloud expenses). The **ROI model** indicates payback within the first year – for every €1 invested in FinOps capabilities (people, tools), we expect ~€5–€7 in cost avoidance or savings.

In summary, **Energinet's FinOps journey** will align cloud investments with business value, ensure every euro spent on cloud is accountable, and build a sustainable culture of cost-efficient innovation. The following report details each aspect of this strategy, providing practical steps, governance templates, and a clear path to FinOps maturity.

# 1. Context & Objectives

Energinet operates mission-critical energy infrastructure, delivering electricity and gas across Denmark. Its IT landscape is hybrid: mission-critical SCADA systems on-premises for real-time grid operations, and a growing multi-cloud footprint for enterprise IT and data analytics (Azure-based data platform [20] with workloads also in AWS and GCP for specific applications). **Annual cloud spend** is estimated in the mid-seven figures (EUR), spread across ~200 Azure subscriptions, ~50 AWS accounts, and GCP projects – reflecting use of cloud for data processing, customer-facing apps, IoT telemetry, and AI/ML pilots.

**Pain Points:** With rapid cloud adoption, Energinet faces several challenges:

- **Cost Visibility & Allocation:** Cloud costs are currently reported in aggregate, obscuring which business units or products drive spend. Shared services (e.g. VPN gateways, data lake) are not allocated clearly, causing **chargeback disputes** and perceived inequities. Tagging is inconsistent, with only ~60% of spend properly tagged to a project or cost center, leading to "unallocated" costs that frustrate Finance.

- **Rising Cloud Waste:** Lack of governance has led to **idle and over-provisioned resources** – e.g. dev VMs running 24/7, oversized VMs at low utilization, orphaned storage volumes – contributing to an estimated 20-30% waste. This aligns with industry surveys where up to *32% of cloud spend is wasted* due to unused or overprovisioned resources [13] .

- **Poor Forecasting & Budget Overruns:** Actual cloud spend often exceeds budgets by 20% or more. Forecasting variable cloud costs (especially for new analytics and IoT projects) is difficult, leading to "surprise" invoices. This unpredictability is problematic in a regulated setting where budgets are scrutinized.

- **Low Commitment Utilization:** Energinet has enterprise agreements (Azure) and committed use discounts, but **Reserved Instance (RI) & Savings Plan coverage** is low (~30%). Many workloads run On-Demand rates with no long-term discounts, meaning lost savings opportunities on

steady workloads. For example, AWS RI utilization is patchy – some purchased RIs go unused (savings leakage), while other workloads that could benefit from RIs have none.

- **Cultural & Ownership Gaps:** Historically, cost management was seen as Finance's job, with engineers having little accountability for cloud spend. There's a cultural gap: engineers optimize for performance and reliability, often ignoring cost implications (a common industry challenge [17] [21]). This silo has resulted in **lack of cost ownership** in engineering teams and slow response to cost optimizations (FinOps Foundation surveys cite *"getting engineers to take action" as the #1 FinOps challenge* [22]).

- **Regulatory Compliance:** As a TSO, Energinet must comply with standards (e.g. NIS2, ISO 27001) that emphasize risk management and operational resilience. Cloud cost management is not directly mandated, but uncontrolled cost growth or poorly managed cloud assets can introduce operational and security risks (e.g. unmonitored shadow IT). Auditors have also asked for clearer linkage between cloud costs and services provided, to ensure prudent spending of what is essentially ratepayer-funded operations.

**Objectives:** Energinet's leadership (CIO, CFO, and COO) has set clear FinOps objectives for the next 12 months:

- **Improve Cost Efficiency by 15–25%** without compromising reliability – primarily by eliminating waste and leveraging economies of scale (enterprise discounts, optimization). This translates to saving several million DKK annually, which can be reinvested in grid innovation or returned to lower consumer tariffs.

- **Achieve 90%+ Cost Allocation Accuracy** – meaning the vast majority of cloud spend is attributed to a business unit, product, or project via tags or account structure. This enables fair chargeback and insights into unit economics (cost per service).

- **Establish Forecasting Accuracy within 10%** – implementing processes so that quarterly cloud spend forecasts versus actuals differ by <10%. This will improve budgeting for the regulated business and reduce the risk of overspend.

- **Implement FinOps Governance** – set up a cross-functional FinOps practice with defined roles (FinOps lead, cloud analysts, etc.), regular reporting to the Cloud Center of Excellence (CCoE) and CFO, and integration into change management. Success means FinOps becomes a routine part of planning new projects (e.g. every architecture design reviews cost impact).

- **Maintain Compliance & Security** – all FinOps improvements must maintain or enhance compliance. For example, any cost data platform must have proper access controls (FinOps data might reveal usage patterns and must be handled as internal sensitive data). Also, optimization efforts (like turning off resources) should not jeopardize uptime of critical systems – we aim for cost savings *without* impacting the reliability required by a TSO.

In summary, the objective is to **instill financial discipline into cloud operations** – *"cloud cost efficiency as a core competency"* – ensuring Energinet can scale digital initiatives sustainably. This sets the stage for the FinOps strategy detailed in subsequent sections.

## 2. FinOps Overview – Principles, Maturity & Comparison to TBM

**What is FinOps?** FinOps is an evolving cloud financial management discipline that brings **financial accountability to the variable spend model of cloud** through a culture of cross-team collaboration [1] [23]. In essence, it's about getting maximum business value from every krona spent on cloud. Key aspects of FinOps include:

- It is **cross-functional** – finance, engineering, product, and business teams *all* play roles, breaking down silos in decision-making [24].
- It is **real-time and iterative** – unlike annual IT budgeting, FinOps operates continuously, tracking cloud spend in near real-time and adjusting course continuously (reflecting the on-demand nature of cloud).
- It **balances speed, cost, and quality** – FinOps doesn't aim to minimize cost at all costs; rather, it helps teams make **trade-off decisions** between cost, performance, and time-to-market [25]. The goal is *efficient growth*, not just cost-cutting [23].

FinOps is guided by core **Principles** established by the FinOps Foundation [26] [27]:

- **Teams Need to Collaborate:** Cloud financial management is a team sport. Engineering, Finance, and Product must share data and align goals in near-real time (since cloud spend changes by the second) [26]. For Energinet, this principle means setting up regular communication channels – e.g. weekly FinOps huddles between platform engineers and cost analysts – to review cost reports and actions.

- **Business Value Drives Decisions:** Decisions about cloud (buying more capacity, optimizing, etc.) should be framed in terms of business value, not just technical metrics. **Unit economics** (cost per customer, per transaction, etc.) are preferred over gross spend [28]. This ensures we invest where we get the best value (for a TSO, value could be reliability or regulatory compliance, not just revenue).

- **Everyone Takes Ownership of Usage:** FinOps pushes **cost accountability to the edge** – meaning application teams and engineers who use the cloud are responsible for efficient usage [27]. This is a shift from traditional IT where Finance "held" the budget – now engineers are effectively **owners of the cost** of their services, just as they are owners of performance and security.

- **Accessible, Timely, Accurate Data:** Without good data, FinOps fails. All stakeholders should get **real-time visibility** into their cloud spend, with data that is reliable [29]. Energinet must ensure our cost reports are updated daily and accessible via self-service dashboards, so teams don't have to wait for month-end finance reports.

- **Centralized Team for FinOps Enablement:** A *central FinOps team* acts as an enablement and governance function (much like a central Security team) [30]. They aggregate cloud usage for bulk discounts, create standards, and evangelize best practices. However, they do *not* absolve others of responsibility – it's a **center of excellence**, not a centralized cost police.

- **Take Advantage of the Cloud's Variable Cost Model:** Rather than fear variable costs, FinOps encourages leveraging it – scale up when value is there, scale down when not [31]. It favors **agile, just-in-time capacity decisions** over static over-provisioning [32]. For a utility, this could mean

scaling analytics clusters up during peak calculation times (e.g. end of month billing, wind forecasting) and down afterward, to only pay for what is needed.

**FinOps Lifecycle – Inform, Optimize, Operate:** FinOps can be viewed as a continuous **cycle of three phases** [33] [34], each with specific goals:

1. **Inform – Visibility & Allocation:** In this phase, we focus on getting granular visibility into costs and attributing them accurately. This includes **cost transparency** (dashboards, reports), **allocation** of costs to teams (via tagging or accounts), budgeting, and creating KPIs. It answers "who is spending what and why?" [35] [36]. For Energinet, the Inform phase deliverables will be things like a detailed cost breakdown by product line (electricity vs gas, etc.), showback reports to each application owner, and establishing metrics like cost/MWh or cost per market transaction.

2. **Optimize – Efficiency & Cost Reduction:** Using the data from Inform, identify optimization opportunities. Two categories: **Rate optimization** (reducing unit prices – e.g. using RIs, savings plans, cheaper storage tiers) and **Usage optimization** (eliminating waste – e.g. rightsizing, shutting idle assets) [37] [38]. Teams collaborate to prioritize and implement these optimizations continuously. Examples include rightsizing VMs, purchasing commitments, or refactoring to serverless for better efficiency.

3. **Operate – Governance & Continuous Improvement:** Embed FinOps into operations so it's sustained. This involves **governance policies**, automation, and **integration into organizational processes** (like embedding cost checks into CI/CD, or having FinOps as part of release gates) [39] [40]. It's about evolving the culture – training, setting up FinOps as an ongoing service. Operate phase for Energinet means things like establishing a Cloud Cost Center of Excellence, setting policies for new projects (e.g. must project costs over 3 years), and continuous training for engineers on cost-aware development.

These phases are **iterative** – different teams might be in different phases at any given time, and we loop through Inform→Optimize→Operate continuously for incremental gains [41] [42].

**Maturity Model:** FinOps maturity typically progresses as *Crawl* → *Walk* → *Run*. In **Crawl**, an organization is reactive (e.g. manually analyzing cost after the bill arrives, ad-hoc tagging). **Walk** brings proactive elements (regular cost reporting, some optimization, budgeting). **Run** means fully integrated FinOps – cost is a factor in **architectural decisions**, engineers get real-time feedback on cost, and optimization is largely automated [43]. Based on initial assessment, Energinet is between Crawl and Walk: some cost awareness exists, but processes are not formalized. Our goal in 12 months is to reach solid "Walk" stage – e.g. monthly optimization cadence, automated reports – and set the foundation to "Run" (e.g. cost-aware design decisions) within 24 months.

**FinOps vs. Traditional IT Financial Management (TBM):** It's useful to contrast FinOps with **Technology Business Management (TBM)**, a framework Energinet's finance team might know from managing on-prem IT costs. **TBM** is about managing the business of IT, with a holistic view of all IT costs (data centers, hardware, software, labor) often using annual budgeting and allocations to apps/services via a fixed taxonomy [44] [45]. FinOps, in contrast, **"was born for the cloud"** – it deals with

highly variable, decentralized spending by empowering engineering teams [46]. Some key differences and complementary points:

- **Scope:** TBM covers *all* technology spend (servers, facilities, cloud, etc.) and often focuses on high-level cost transparency (e.g. total cost of ownership of an application). FinOps zeroes in on **public cloud consumption** and gets very granular (down to tagging an individual resource) [47] [48]. In practice, FinOps can be seen as the cloud-era extension of TBM, providing the detailed practices to manage cloud bills which TBM doesn't detail.

- **Cadence & Ownership:** TBM typically has Finance-led cycles (quarterly or monthly reports, annual planning). FinOps runs on a much faster, **continuous cadence** aligned to engineering sprints. It shifts execution to engineering teams (Finance sets guardrails and goals, but engineers decide *how* to optimize) [49]. As Apptio notes, "TBM manages the business of technology; FinOps enables orgs to realize the full business value of cloud" [50]. The two should work together – e.g. FinOps data feeds into TBM's broader cost model. Indeed, FinOps can produce granular cloud cost data and savings which TBM then allocates and ties to business value questions ("should savings be reinvested or returned?" [51]).

- **FinOps for Regulated Utility:** One might equate FinOps to *real-time cost management* vs TBM's *strategic cost management*. For Energinet, which operates under allowed revenue models, TBM ensures overall IT spending is optimized and transparent to regulators. FinOps will provide the day-to-day tactics to ensure our cloud spend (an increasing portion of IT spend) is efficient and tied to operational metrics (like DKK per customer served or per MW of capacity). FinOps will also produce data (like unit costs, variance analyses) that can feed regulatory reports or internal KPIs, complementing TBM's top-down approach with bottom-up cloud cost insight.

In summary, FinOps is the vehicle to continuously optimize and align cloud usage with business value. By adopting FinOps principles and integrating them with our existing financial governance, Energinet can **achieve the agility of cloud while staying in control of cost**, which is essential for a public-oriented enterprise. Next, we assess Energinet's current FinOps-related capabilities and gaps as a baseline.

# 3. Current State Assessment – Process, Data, Tooling, Culture

We conducted a current state analysis of Energinet's cloud financial management across people, process, data, and technology. Key observations:

**3.1 Organization & Process:** There is currently *no dedicated FinOps team or formal process*. Cloud cost management responsibilities are diffused:

- The **Cloud Platform Team** (part of IT Operations) monitors high-level Azure and AWS bills and sometimes flags anomalies, but this is ad hoc. They negotiated the Azure enterprise agreement and some AWS discounts, but ongoing optimization isn't in their KPI.
- The **Finance Controlling team** receives monthly invoices and does basic allocation of costs to general ledger accounts (e.g. by project code if provided). However, they lack granular data or cloud expertise, so they can't drill into why costs changed or whether resources are efficiently used. No regular financial report is produced at product/team level (only aggregate cloud spend).
- **Application/Product Teams** treat cloud costs as an OPEX line item but with little ownership. There's no practice of these teams reviewing their cost and usage regularly. In interviews, many

engineers were unaware of the scale of costs their applications incur, consistent with industry findings that 53% of cloud pros aren't sure engineers care about cloud cost [52].

- **Governance:** There's an **IT Steering Committee** quarterly, but cloud costs are only briefly discussed if a budget exception occurs. No specific FinOps or cloud cost review forum exists yet. Tags are not governed – there's a tagging policy document from 2022, but compliance isn't tracked or enforced, leading to "tag drift".

**3.2 Technology & Data:**

- **Cost Visibility:** Energinet relies mostly on native cloud portals for cost visibility (Azure Cost Management and AWS Cost Explorer). These are used on-demand but not aggregated. We do not have a unified multi-cloud dashboard. As a result, getting a full picture (Azure + AWS + on-prem) requires manual aggregation in Excel. This is time-consuming and prone to error – for example, reconciling Azure's and AWS's different billing periods and cost categories is non-trivial. The FinOps Foundation notes *consolidating and normalizing billing data* across clouds is a universal challenge [7], which we experience firsthand.
- **Cost Allocation & Tagging:** Approximately 60% of Azure resource costs and 50% of AWS costs carry at least one useful tag (like application name). But **tagging is inconsistent**: there are multiple spellings (`Dept` vs `Department`, etc.) and many resources (especially older ones or third-party services) are untagged. Kubernetes clusters present a challenge: containerized workloads share clusters, and **Kubernetes cost allocation** is not fully implemented (we haven't deployed a solution like Kubecost yet), so those costs appear lumped. Shared services (networking, security appliances) run in separate accounts with no chargeback mechanism – so these costs (~15% of total cloud spend) are sitting as "shared IT" with no transparency, causing occasional disputes with business units who question why they're charged overhead. This mirrors common FinOps challenges – *"tagging resources… can be hit or miss"* and untagged resources impede accurate allocation [53] [10].
- **Tools:** Aside from native cloud tools, Energinet has started using Power BI connected to Azure Cost Management data for some reports. But this covers Azure only. AWS CUR (Cost & Usage Report) is not yet ingested into our BI environment. No third-party FinOps tool is in use. This means we lack advanced features like anomaly detection or automated recommendations. CloudZero's survey found ~26% of companies still rely on homegrown or no tooling [18] – Energinet currently is in that bucket, indicating room for improvement.
- **Data Quality:** The cost data we do have has some quality issues. Azure's cost export sometimes shows costs with a delay (up to 2 days lag), and there have been instances of **data gaps** (e.g. a tagging bug where costs were not attributed properly for a new service, later fixed). Also, the mapping of cloud accounts to internal cost centers is done via a manual list that quickly becomes outdated as new projects spin up cloud resources without finance being notified. These data issues result in Finance questioning the reliability of IT's numbers, undermining trust.

**3.3 Culture & Skills:**

- Cloud cost awareness is **limited among engineering**. There is no mandate or incentive for engineers to optimize costs – their performance reviews focus on system uptime and project delivery. In absence of clear ownership, cost optimization tasks "fall through the cracks". This is a classic culture problem echoed by FinOps Foundation: *empowering engineering with cost info is vital since engineers can inadvertently make choices that raise costs if not engaged* [17] [21].
- The **finance team's understanding of cloud** is also limited. They treat cloud bills like utility bills – something to pay and allocate – without insight into how usage patterns or cloud architecture affect cost. There's a training gap here: FinOps concepts (like RI coverage, container cost allocation, etc.) are new to Finance. This misalignment is common – aligning tech and finance teams is cited as a major FinOps challenge in industry surveys [54].

- However, we note high-level support: the CIO and CFO both acknowledge cloud cost is now significant and have expressed support for FinOps. This executive buy-in is a strength to leverage (and indeed 96% of orgs acknowledge FinOps is important to cloud strategy [55]).

**3.4 Security/Compliance Posture:** Currently, **no major compliance issues** were found in cost management – but there is a lack of formalized policies. For example, **no policy-as-code** exists to enforce tagging or prevent deployment in non-allowed regions (which could have cost and compliance impact). From a data privacy view, our cloud cost data might include spend on personal-data-related services, but it's aggregate cost, not PII, so low privacy risk. One risk is **access control**: cloud cost data often reveals patterns about usage (e.g. if a team is ramping up spend, it might hint at a new project). We must ensure cost data is only accessible to authorized employees to avoid any inadvertent info leakage (particularly since Energinet is critical infra, even cost spikes might be of interest to regulators or attackers). Right now, access to cost info is not well governed – many engineers lack any visibility, while some with cloud portal access can see more than their scope.

**3.5 Gap Analysis:** Summarizing gaps between current state and FinOps best practices:

- *Governance Gap:* No FinOps governance structure (no FinOps team, no cadence for reviews). Decisions are reactive and escalated only when there's a budget issue.
- *Data & Tooling Gap:* No integrated cost data platform; manual, siloed reporting; incomplete tagging impedes cost allocation; lacking automation (like anomaly alerts). We do not meet the FinOps principle of timely, accessible data yet [29].
- *Process Gap:* Optimization is ad hoc. There's no "optimization backlog" or regular rightsizing exercise. Many savings opportunities remain untouched (e.g. low RI coverage).
- *Skills/Culture Gap:* Low awareness and ownership among engineers; finance not equipped to lead cloud optimization. Collaboration between teams on cloud cost is minimal (violating the "Teams collaborate" principle [26] in practice).
- *Policy Gap:* Lack of policies for cost control – e.g. no guardrails to prevent costly resource usage by mistake, no budget alerts.

These gaps mean Energinet currently is in the **"Crawl" stage** of FinOps maturity – reactive and partial. The next sections will address how to bridge these gaps with a target operating model, improved data strategy, and process changes.

# 4. Target Operating Model – FinOps Org Design, RACI, Governance

To embed FinOps in Energinet, we recommend establishing a **FinOps Operating Model** that fits our enterprise structure and regulated context. The model centers on a *FinOps Core Team* (as a hub) working with distributed stakeholders (spokes), under clear governance.

**4.1 FinOps Team Structure:** A **Hybrid FinOps Team model** is ideal [56] [57]. This blends a **centralized** function (for consistency and economies of scale) with **embedded** FinOps responsibilities in business units (for local context and faster action):

- **FinOps Core Team (Central):** A small dedicated team (initially 2-4 FTEs) that sits in the IT organization (under the Cloud Center of Excellence). Roles include:
- *FinOps Lead:* Acts as FinOps product owner – driving strategy, ensuring cross-org collaboration, and reporting to CIO/CFO on FinOps progress.

- *Cloud Financial Analyst(s):* Experts in cloud billing and analysis. They own the cost reporting pipeline, perform deep dives on anomalies, and identify optimization opportunities across all clouds [58] [59].
- *FinOps Engineer:* (Part-time from platform team) to automate tooling, integrate cost data, and enforce policies via code [60] [61].
- Optionally, a *Business/Unit Economics Analyst* who works between Finance and Product to define and calculate unit metrics (could be filled by the Financial Analyst role initially).

This central team's mission is to **"enable and govern"**: they provide the single source of truth for cost data, create guardrails, run training, and facilitate discussions – similar to how a Security team sets policies but expects dev teams to implement securely [30].

- **Embedded FinOps Champions (Federated):** Each major business unit or product team (e.g. Electricity Market platform, Gas systems, Data Analytics team) should appoint a **FinOps Champion** (could be an engineering lead or product manager). This person is the liaison to the FinOps Core Team. They attend FinOps reviews, ensure their team's resources are tagged and optimized, and help interpret FinOps recommendations in context. They are essentially FinOps "ambassadors" who spread cost awareness in their teams. Embedding FinOps responsibility ensures decisions are made with local context and speeds up response – aligning with the decentralized aspect where engineering has ownership. Indeed, companies like Netflix use such decentralized FinOps in mature stage [62].

This **hybrid approach** balances central governance with local autonomy and is common as organizations scale [63]. It also acknowledges Energinet's need for central control (for compliance/audit in critical infra) while leveraging domain knowledge in teams.

**4.2 RACI – Clear Roles & Responsibilities:** We define a RACI matrix (Responsible, Accountable, Consulted, Informed) for key FinOps activities, adapted from FinOps Foundation guidelines [5] [64]:

- *Tagging Policy & Compliance:* **Accountable (A):** FinOps Core Team (they define global tagging standards and monitor compliance) [65]. **Responsible (R):** Application/Dev Teams (they must apply tags in their infrastructure as they provision) [65]. **Consulted (C):** IT Security/Compliance (to align with any data classification tags), Finance (to ensure tags meet cost allocation needs). **Informed (I):** CIO/CTO on overall compliance rates.

- *Cloud Budgeting & Forecasting:* **A:** Finance (Finance director remains accountable for overall budgets), **R:** FinOps Core (prepares spend forecasts from trends) and Product Owners (bottom-up input for forecasts) – effectively a joint R where FinOps provides data, product teams provide demand estimates [66]. **C:** CFO, PMO. **I:** Engineering leads (so they know targets).

- *Optimization Recommendations (rightsizing, etc.):* **A:** FinOps Core (they govern the optimization backlog company-wide), **R:** Engineering/App Teams for executing specific optimizations (e.g. resizing a database) [67]. **C:** Product Manager (to confirm no business impact), **I:** Finance (for large savings realized).

- *Reserved Instance & Savings Plan Management:* **A:** FinOps Core (or central Cloud Procurement), as they handle enterprise commitments [68]. **R:** FinOps Core and Sourcing/Procurement together (FinOps analyzes usage, Sourcing executes contracts) [68]. **C:** App teams (to forecast their steady-state needs), Finance (for cash flow planning). **I:** CIO, CFO on major purchases.

- *Operations – Anomaly Response:* **A:** FinOps Core (owns the anomaly detection process and playbook), **R:** Potentially the specific Team where anomaly occurred (to investigate cause). **C:** Security if incident (e.g. cost spike from crypto mining could be breach), **I:** Relevant management depending on severity.

- *FinOps Training & Culture:* **A:** FinOps Lead, **R:** FinOps Champions (embedded) to conduct training in their teams, **C:** HR/Learning & Development (to include in curricula), **I:** All engineering (target audience).

We will document this RACI and share it widely so everyone knows who does what. This addresses the current confusion and ensures, for example, that engineers know it's their responsibility to act on cost optimizations (with FinOps team support) rather than waiting for someone else.

**4.3 Governance Forums & Cadence:**

- **Monthly FinOps Review Board:** Establish a monthly meeting including FinOps Core Team, FinOps Champions from each product area, plus Finance business partner and an executive sponsor (e.g. CTO). In this meeting, the FinOps Core presents last month's cloud spend vs budget, KPI scorecard, major anomalies, and optimization initiatives status. Teams discuss and agree on actions (e.g. Team A will commit to deleting unused test env by X date, Team B will evaluate a Savings Plan purchase). This creates a regular *FinOps ceremony* similar to a sprint review, bringing accountability. Escalation path: issues that cannot be resolved (like a team refusing to downsize something due to risk concerns) can be escalated here for leadership direction.

- **Quarterly Executive Update:** Each quarter, a summary goes to the CIO, CFO, and COO (and possibly the Audit Committee) highlighting savings achieved, forecast updates, and decisions needed (e.g. committing to a multi-year cloud discount deal, or policy changes). This keeps leadership engaged (avoid "out of sight, out of mind") and helps resolve cross-department trade-offs (like balancing cost vs performance where needed).

- **On-Demand War Room for Anomalies:** If a major cost anomaly hits (say spend in a project doubles overnight unexpectedly), the FinOps team will convene an ad-hoc incident response (including the owning engineering team, FinOps, and possibly security) to triage and mitigate. This process will be documented similarly to an outage incident process – important given we must respond quickly to cost spikes to avoid budget blowouts [15].

**4.4 Integration with Existing Processes:** FinOps should integrate into Energinet's existing governance:

- **Project Approval:** Include a FinOps review in new cloud project proposals. This means any new project must outline expected cloud costs and tagging structure, reviewed by FinOps Core. For example, before a data science project spins up a 100-node Spark cluster, FinOps ensures budgets are set and cost tracking (tags or separate account) is planned.

- **Change Management:** When major architecture changes are proposed (e.g. moving an on-prem app to cloud), include cost impact analysis by FinOps in the architecture review board checklist. This raises awareness of cost early (and avoids unpleasant surprises later).

- **Procurement & Finance Cycles:** Align FinOps activities with finance cycles. E.g., yearly budgeting – FinOps provides inputs on expected cloud growth or savings to incorporate into the financial plan. During quarterly re-forecasts, FinOps helps adjust cloud budgets.

- **Audit & Compliance:** Document FinOps processes for audit. E.g., keep records of chargeback calculations, anomaly investigation outcomes, RI purchase justifications. This provides evidence to internal/external auditors that cloud spend is monitored and controlled (supports compliance with any regulatory expectations of prudent financial management).

**4.5 Organizational Change Management:** Shifting to this model requires **cultural change**. We'll undertake change management actions: - Announce the FinOps initiative from top leadership (CIO) to signal its importance (executive buy-in is critical [69]). - Provide training and awareness sessions to both engineering and finance teams about how FinOps will benefit them – demystifying that it's not just about slashing budgets but enabling innovation by using funds efficiently. - Start with a "FinOps Pilot" (maybe focus on one cloud or one business unit) to demonstrate value, as recommended (show quick wins to build momentum [70]). - Recognize and reward teams that actively engage and achieve optimizations (e.g. shout-outs in company meetings or even tangible incentives like using part of saved budget for team development).

By establishing this robust FinOps operating model, we ensure **decisions rights and responsibilities are clearly assigned** and there are mechanisms to continuously drive accountability. This model mirrors how successful FinOps teams operate at scale, balancing central oversight with distributed action [71] [72]. It sets the foundation for all subsequent FinOps processes (data, optimization, etc.) to function effectively.

# 5. Data & Tagging Strategy – Cost Pipeline Architecture and Metadata

A reliable, unified **cost data pipeline** is the backbone of FinOps. We will implement a **"single source of truth"** for all cloud and hybrid cost data, enabling transparency and detailed analytics. Key elements of the strategy:

**5.1 FinOps Cost Data Architecture:**

We propose building a **Cloud Cost Data Lake** that consolidates usage and cost records from Azure, AWS, GCP (and potentially on-prem costs) into a normalized schema. This aligns with FinOps best practices of integrating data sources into one repository for analysis [73] [74]. The architecture (Figure 1) has these stages:

*Figure 1: Proposed multi-cloud cost data pipeline. Raw billing data from each cloud provider is ingested (AWS CUR, Azure usage export, GCP billing export) and processed through cleaning, normalization, and enrichment steps to produce a consolidated "final cost" dataset. This is joined with enterprise metadata (e.g. CMDB, org hierarchy) and fed to dashboards, reports, and forecasting models.*

1. **Ingestion of Raw Billing Data:** We will enable the native detailed billing exports:
2. **AWS:** Use the Cost & Usage Report (CUR) delivered to S3 daily. This contains line-item details of usage (down to resource-ID and hour) in CSV or Parquet. We'll load this into our data lake (or directly into a query engine like Athena or Snowflake).

3. **Azure:** Use Azure Cost Management exports to storage (daily export of usage details). Alternatively, since Energinet's Azure is under an EA, we can use the EA billing API or Azure's Data Connector to get cost data into Azure Data Lake (which we already use for other data).
4. **GCP:** Export billing data to BigQuery (Google's native method). We'll then either federate queries from BigQuery or extract to our central store. Possibly, we keep GCP data in BigQuery and integrate via views.
5. **On-Prem/Other:** For completeness, we might include on-prem infrastructure costs in the data model at least at summary level (e.g. monthly amortized cost of private cloud) to eventually calculate combined unit costs. Also, SaaS costs (if significant) could be added later.

Each provider's data schema is different, but includes common fields: usage amount, cost, service type, tags/labels, region, etc. The data can be **very large** (CUR can be millions of rows a month). We'll partition and store efficiently, possibly in Parquet format for query performance.

1. **Staging & Cleaning:** Raw data can be messy:
2. **Data cleaning:** Ensure numeric fields are correct types, handle missing values. For instance, sometimes credits/discounts come as separate negative line items – we'll want to merge or flag those appropriately.
3. **Time alignment:** Normalize timestamps to a common time zone (UTC) and consistent period (billing periods differ; we'll probably use calendar month as a standard period for reporting).
4. **Currency consistency:** If any cloud bills in USD, convert to DKK or EUR consistently using financial rates for reporting (so we don't mix currencies).

We will enforce data quality checks here (e.g., ensure the total in our processed data matches the official invoice total, to catch any ingestion errors).

1. **Tag Enrichment:** Join in the resource tags/labels from each cloud as columns in the data set. If some resources are missing expected tags, we may enrich using naming conventions or Cloud Resource Hierarchy info. For example, if an AWS resource lacks a `CostCenter` tag, but it's in an AWS account that is dedicated to "IT Ops", we can infer a cost center. We'll add that as an enriched field but flag it as inferred.

*Automated Tag Augmentation:* We can use mapping tables (e.g., a list mapping Azure subscription IDs to Business Unit, or mapping project names) to fill in some metadata where tags are absent. FinOps data workflows often add such derived tags to improve allocation [75] [76].

1. **Normalization (Multi-cloud):** To support combined analysis, transform provider-specific fields to a **common schema**. This is where adopting **FOCUS (FinOps Open Cost and Usage Specification)** can help [7]. FOCUS provides a unified set of columns (like `UsageAmount`, `UsageUnit`, `Cost`, `Service`, `ResourceID`, `Tag:key` etc.) across clouds. We will map Azure's terms (e.g. "resource group" to "namespace" perhaps), AWS (account to project), GCP, etc., into common terms. This reduces confusion and allows unified queries (e.g. total spend by service category across clouds). FOCUS also recommends combining **amortized costs** (spread upfront payments) in the dataset for true cost visibility [77], which we will do – e.g., if we pay upfront for a 1-year RI, we'll amortize that cost over the year's months in our data.

Normalization also includes standardizing units (ensuring CPU hours vs vCPU hours etc., if needed for unit cost calcs) and service names (e.g. label both AWS Lambda and Azure Functions as `ServiceType=Serverless Function` via a mapping [78]).

1. **Cost Allocation & Hierarchy:** After normalization, we assign costs to organizational entities:

2. Use tags like `BusinessUnit`, `Product` to allocate each cost line. Where tags are missing, use default assignments (e.g. if a subscription is untagged, attribute its costs to a default cost center or query the resource's owners).
3. Implement rules for **shared costs**: For example, a central VPN or ExpressRoute that serves multiple business units – decide an allocation rule (like 50/50 split between electric and gas, or based on number of connected systems). This can be encoded in the data pipeline: mark such costs and later apportion them in reports or create "shared cost" entries for transparency [79] [80].

4. Build a hierarchy mapping (e.g. cost center to division to company) so we can roll up costs. This likely comes from Finance's master data (we'll integrate with the ERP or CMDB).

5. **Outputs & Access:** The processed, enriched data is then loaded into:

6. **BI Tools:** e.g. Power BI or Tableau dashboards for various audiences (detailed analyst view vs executive summary).
7. **Self-service Queries:** Analysts or engineers can query the data (maybe via a read-only SQL interface or notebooks) to do custom analysis (subject to access control).
8. **Forecasting/Analytics Models:** Export the time-series data to forecasting tools (could be simple spreadsheets or a Python ML model).
9. **Showback/Chargeback reports:** We might output Excel or PDF summaries to send to business units as part of monthly reporting.

To ensure **timeliness**, we aim to update this pipeline daily (possibly more frequently if feasible with streaming). Some companies implement near-real-time ingestion (e.g. using Kafka to stream usage records) [81] – not mandatory initially, daily batch is fine, but we may explore it for quick anomaly detection.

Ensuring **accuracy** is paramount: we will reconcile our data monthly with cloud provider invoices to verify no discrepancies.

**5.2 Tagging Strategy:** High-quality tagging is *foundational* for cost allocation. Our strategy:

- **Tagging Policy:** Define a corporate tagging standard (published as a living document). It will include:
- **Required tags:** e.g. `BusinessUnit` (or `CostCenter`), `Application` (or Service name), `Environment` (Prod/NonProd), `Owner` (team or person). These must be present on all taggable resources. For example, every VM, database, bucket, etc., must carry tags like `BusinessUnit=GasOps` or `Env=Prod`. These correspond to dimensions Finance cares about.
- **Optional tags:** e.g. `Project` (for transient projects), `ServiceCategory` (maybe tagging if it's customer-facing vs internal), `Compliance` (e.g. any regulatory classification if needed).
- **Tag format conventions:** e.g. keys in PascalCase, values limited to certain list or free text if needed. We will standardize key names across clouds (use Tag Key normalization if needed – our pipeline can, for instance, treat AWS `aws:createdBy` similarly to an Azure `CreatedBy` tag by aliasing).

- **Examples:** Provide examples to developers, e.g., a VM for the Energy Data Portal might be tagged: `BusinessUnit=Digital`, `Application=EnergyDataPortal`, `Environment=Prod`, `Owner=TeamDataPlatform`.

- **Technical Enforcement:** Use cloud-native tag policies where available:

- Azure: Azure Policy can enforce certain tags on resource creation (either reject if missing or add default). We will deploy a policy that, for instance, auto-applies a default `BusinessUnit` tag based on the subscription if none provided, and flags resources without critical tags.
- AWS: Use Service Control Policies (SCPs) or tag compliance Lambda scripts. AWS can trigger Lambda on resource creation to check tags and mark or notify if non-compliant. We might not block creation (could impede agility), but we'll definitely audit.
- GCP: Enforce labels via Organization Policy constraints (GCP now supports required labels).

- In Terraform or IaC pipelines: incorporate tag requirements modules to ensure every resource has the needed tags in code.

- **Tag Hygiene Process:** Despite best efforts, tags will drift (especially in a multi-team environment). We will:

- Run **automated tag compliance reports** weekly – listing resources missing required tags or using non-standard keys. These go to respective owners for remediation.
- Use the cost data pipeline's **tag processing stage** to normalize tags (e.g. fix case differences, alias old tag keys to new standard [82]). For instance, if we find `costcenter` vs `CostCenter`, unify them in the data.
- Account for **"untaggable" costs**: Some costs (like AWS data transfer or certain managed services) can't be tagged. We will handle these by other means – e.g. allocate by proportion of other tagged usage or assign to an "overhead" bucket that is then allocated in reports based on a key (like % of usage).

- For **Kubernetes**: implement a solution (e.g. Kubecost or the open-source kube-metering) to allocate costs per namespace or label, since containerized environment doesn't use cloud tags in the usual way. This data then feeds into our pipeline (perhaps as a separate source).

- **Quality Metrics:** We will measure tag coverage as a KPI (target >90% of spend tagged with required fields). Also measure tag accuracy (via occasional audits – e.g. check if `Owner` tag is a valid current employee/team).

**5.3 Integration of Business Data:** Beyond tags, join cost data with business context: - **CMDB/Asset registry:** If we have an internal CMDB listing which service or system a resource belongs to, integrate that via resource naming or IDs. - **Budgets:** Include budget data in the warehouse, so we can compare actual vs budget in the same dashboard. - **Operational metrics:** For unit costs, pull in usage metrics (customer counts, transactions, energy volumes). For example, if we want cost per energy trade processed, we need the number of trades from another system. We plan to link those metrics with cost – likely in separate tables keyed by time and product. - **FOCUS extension:** The FOCUS spec is extending to allow non-cloud costs and even on-prem [83]. We will keep an eye on that so we can incorporate future cost types (SaaS, etc.) easily.

**5.4 Data Lineage & Controls:** - The pipeline will be documented showing how data flows from source to report. We'll implement **access controls** at each stage. Raw detailed data might be limited to FinOps analysts (since it's large and granular), whereas aggregated views are shared wider. This is to avoid overwhelming users and to secure sensitive info (for instance, a resource name might reveal a project code name). - We will maintain an **audit trail**: versioning of cost data if re-statements are needed (e.g. if a tagging backfill changes allocation, we log it). This ensures trust – anyone can trace a number in a report back to raw data. - **Data Freshness SLA:** We commit that prior day's cloud usage data is processed by, say, 9 AM next day. This timeliness is important because FinOps principle calls for *fast*

*feedback loops* [84]. - We'll consider **cost anomaly detection** integration directly on the data (e.g. running queries to find deviations beyond thresholds and flagging to Slack/email).

By implementing this comprehensive data strategy, Energinet will overcome the current visibility issues. All teams will operate from a **single, consistent set of cost data**, driving a "one version of truth" culture. This addresses root causes of many FinOps challenges (like confusion over different reports showing different numbers [74] [85]). Moreover, having this infrastructure will enable more advanced analysis like **showback, chargeback, and unit economics** (next sections) with confidence.

In summary, this data pipeline and tagging strategy ensures cost data is **accurate, granular, and business-contextual** – a non-negotiable foundation for successful FinOps.

# 6. Optimization Playbook – Key Levers for Cost Efficiency

With visibility in place, the FinOps team will drive a structured **Optimization Playbook** targeting waste and efficiency opportunities. Energinet's top 10 cloud cost "failure modes" (pain points) align with common FinOps challenges [17] [79]. For each, we outline actions, expected savings, and prerequisites:

**1. Idle/Underutilized Compute Resources** – *"Cloud waste"* due to low utilization was cited by 88% of hybrid cloud FinOps survey respondents [86]. In our case, many VMs and containers run at <10% CPU most of the time. - **Action: Rightsize or turn off idle instances.** Use tools (Azure Advisor, AWS Rightsizing recommendations) to identify servers with sustained low utilization. Rightsizing means choosing smaller instance types or less vCPUs/RAM for VMs that are oversized. Also, implement **schedule to shut down** dev/test VMs after hours (e.g. using an automation script or AWS Instance Scheduler). - **Effort/Impact:** Medium effort (need testing to ensure performance isn't hurt). Impact: High – typically rightsizing yields **20-25% savings on average** [14]. Shutting off non-prod outside 8am-6pm can save ~65% of those instances' cost (16 hours off per day plus weekends). - **Owner:** Cloud Platform team (to implement scheduling scripts) in collaboration with App owners (to approve smaller sizes or validate idle criteria). - **Why it matters:** Eliminating waste directly frees budget. For Energinet, early scans show ~€50k/year could be saved from dev subscriptions alone by scheduling shutdowns.

**2. Orphaned & Unused Resources** – Examples: unattached storage volumes, old snapshots, idle load balancers or IP addresses, orphaned NICs. - **Action: Resource cleanup drives.** Use cloud provider inventory to find unattached volumes (AWS EBS, Azure Managed Disks not attached), and either delete or archive snapshots to cheaper storage. Clean up old backups beyond retention policies. Remove unused Elastic IPs or Azure Public IPs (they incur charges when reserved but unused). - **Effort/Impact:** Low effort (scripts or even one-time effort). Impact: Medium – might save 2-5% of total spend. Flexera's 2023 report notes storage/Snapshot waste is a notable chunk of the ~30% cloud waste [13]. - **Owner:** FinOps team identifies; Ops team executes deletion after confirming with owners (to avoid accidental deletion of needed data). - **Guardrail:** Ensure backup policies and data governance are respected (don't delete snapshots needed for compliance retention).

**3. Overprovisioned Capacity for High Availability** – E.g., running N+2 redundancy where N+1 suffices, or keeping large buffer capacity always on for peak. - **Action: Embrace Autoscaling & Demand-Based Scaling.** Instead of static overprovisioning, use cloud auto-scaling groups and scale-out when needed. For example, if an app is running 10 servers at 10% CPU each (to handle spikes), configure auto-scale to run 3 servers normally and scale to 10 on demand during peaks. Similarly, use Azure Scale Sets or GCP Instance Groups for elasticity. - **Effort/Impact:** Effort can be medium (requires architecture/testing changes). Impact: potentially high for variable workloads – you pay only for actual demand. Could yield 20-50% savings on that service while still meeting reliability. - **Owner:** Application team (with support

from cloud engineers to set up autoscaling). - **Pre-req:** Workloads must be stateless or handle scale-in gracefully. Also needs testing to ensure scaling works within performance SLAs.

**4. Low Utilization of Reserved Pricing** – Not leveraging available discounts like Reserved Instances (RIs), Savings Plans (AWS) or Azure Reserved VM Instances, meaning we pay On-Demand rates even for predictable workloads. - **Action: Increase RI/CUD coverage.** FinOps team to regularly analyze usage for steady-state patterns. Purchase 1-year or 3-year RIs for instances with >60-70% steady utilization, or Savings Plans for flexible coverage. On GCP, utilize Committed Use Discounts (CUDs) for e.g. BigQuery or Compute. Leverage enterprise agreements (Azure) fully by reserving where beneficial. This central buying is a FinOps principle (central team handles rate optimization) [87]. - **Effort/Impact:** Medium effort (financial commitment and management overhead). Impact: Very high ROI – RIs can save 30-50% on compute costs [38]. If currently only 30% of workload is covered and we boost to 70%, that delta 40% of spend could see ~30% savings => ~12% total spend reduction. - **Owner:** FinOps Core Team (with Sourcing/Procurement), accountable to Finance for ROI analysis. Engineers provide forecast of usage to avoid overcommitting. - **Risk mitigation:** Start with conservative commitments (maybe 1-year terms or partial upfront) and use cloud vendor flexibility (AWS allows some RI exchanges, Azure allows instance size flexibility within a family).

**5. Untapped Spot / Preemptible Instances** – Certain non-critical or batch workloads (data processing, dev/test, container jobs) could run on spare capacity (spot instances on AWS/Azure, preemptible VMs on GCP) at ~70-90% lower cost, but we currently use on-demand for everything. - **Action: Adopt Spot for suitable workloads.** Identify workloads that are fault-tolerant to interruption (e.g. a nightly ETL job, or a Kubernetes cluster that can reschedule pods). Use Spot VM pools for those. For example, run a Kubernetes node group with spot instances for background processing jobs. AWS Spot and Azure Spot VMs can be integrated with scaling groups. Also, use GCP preemptible VMs for transient tasks. - **Effort/Impact:** Effort depends on workload (app modification might be needed to handle interruption). Impact: Huge savings on those workloads – up to **70-90% cost reduction** for the portion moved to spot. - **Owner:** Engineering teams owning batch workloads, with guidance from FinOps on how to implement. - **Note:** For critical systems, we likely avoid spot, but for many analytics tasks (like large simulations that can checkpoint), it's underused potential.

**6. Expensive Storage Tiers & Data Lifecycle** – Data stored on premium/expensive tiers that could reside on cheaper storage or be deleted. - **Action: Storage Tiering and Data Lifecycle Management.** Enforce policies: e.g. logs older than 30 days move from hot storage to archive (Glacier, Azure Archive Blob). Use lifecycle rules on object storage to transition data (especially for data lake of time-series). For block storage, review if any high-IOPS SSD volumes are overkill for the workload and downgrade to standard if possible. Also consider compressing data or deleting truly unneeded data (with business approval). - **Effort/Impact:** Low to medium effort (setting up rules, some validation). Impact: Can be significant if lots of cold data – archival storage is ~1/4 the cost of hot. If 20% of storage can go to cold at 75% savings, that's 15% of storage cost saved. - **Owner:** Data engineering team and system owners, guided by FinOps (which identifies candidates). - **Why it matters:** With IoT and telemetry, data volumes grow fast. Without lifecycle management, we pay indefinitely for stale data.

**7. Network Egress & Bandwidth Optimization** – Cloud providers charge for data transfer out (egress) and sometimes between regions. Energinet's data integrations might incur egress (e.g. sending data from cloud to on-prem or between clouds). - **Action: Architect for lower egress costs.** Options include using **CDNs or caching** for data served externally (so that repeated access doesn't hit origin egress), leveraging private links or direct interconnects (like Azure ExpressRoute or AWS Direct Connect) which can be cheaper per GB at scale, consolidating workloads in the same region or cloud to minimize inter-cloud data movement (multi-cloud data flow can double-charge egress). Also compress data in transit. - **Effort/Impact:** Architecture effort might be needed. Impact: Medium – egress can be a small portion of

spend, but for data-heavy orgs it adds up. Even a 20% reduction can save good money. Also, architecture changes can improve performance (latency) as a bonus. - **Owner:** Infrastructure network architects in collaboration with FinOps to quantify savings trade-offs.

**8. Container & Kubernetes Optimization** – We use Kubernetes for some microservices. Common inefficiencies are over-allocating CPU/memory requests leading to low pod density, and running too many cluster nodes 24/7. - **Action: Optimize K8s resource allocation and scaling.** Implement tools to monitor container usage vs requests (e.g. K8s Vertical Pod Autoscaler recommendations, or Kubecost's insights). Right-size container limits so that more pods can pack per node safely, reducing needed nodes. Set up cluster auto-scaler so if workloads drop, nodes scale down to zero if not needed (especially in dev clusters). Also consider using smaller node types or spot nodes as mentioned. - **Effort/Impact:** Medium (requires dev team involvement to adjust deployments). Impact: potentially high in aggregate if K8s adoption is large – e.g. improving pod density by 2x could halve cluster cost for some workloads. - **Owner:** DevOps / platform engineers managing K8s. - **Metric:** Track cluster utilization (if <20% utilized, lots of slack = optimize).

**9. Serverless and PaaS Tuning** – Services like Azure Functions, AWS Lambda, etc., where costs scale with usage. Ensuring configurations are optimal (not over-provisioning memory or runtime). - **Action: Rightsize serverless configs and optimize code.** For Lambda/Functions, choose memory settings that are sufficient but not excessive – too high memory alloc not only charges more but can waste if function doesn't need it. Use monitoring to see actual memory used. Also, optimize function code to execute faster (reducing GB-sec used). On databases (like DynamoDB, Cosmos DB), adjust provisioned throughput or use auto-scaling to avoid paying for unused capacity. - **Effort/Impact:** Low to medium. Impact: can be moderate – e.g. one case, reducing a Lambda memory from 1GB to 256MB saved 50% cost when it was underutilized. Multiply such wins across many functions. - **Owner:** Respective service developers, with FinOps providing analysis from logs.

**10. Software Licensing and Marketplace Spend** – (if applicable) Sometimes cloud bills include software licenses (Windows, databases) or marketplace SaaS. Optimize by switching to open-source alternatives or long-term licenses. - **Action: License optimization.** E.g., if we run many Windows VMs, consider Azure Hybrid Benefit or AWS BYOL to use existing licenses. Use license-included vs bring-your-own cost analysis. Also review if some third-party appliances on marketplace are expensive and if open source or more efficient alternatives exist. - **Effort/Impact:** Case-by-case. Impact: Low to medium depending on how much of spend is licensing. For example, using Hybrid Benefit can save up to ~40% on Windows VM cost by using existing on-prem licenses.

Each optimization is tracked in a **FinOps Savings Backlog** (Annex F has a template). We rank by ROI and implement continuously.

Importantly, we must tie these optimizations to **operational considerations**: - We will not compromise the reliability or compliance of systems. E.g., before rightsizing a production system, test in staging to ensure performance is okay. - For each, we involve system owners to weigh in. FinOps provides data (e.g. "Server X avg CPU 5%, consider downsizing from 8vCPU to 2vCPU [88] "). The team then decides and executes with our support.

We also plan to **measure the actual savings** from each action and report it (to show progress). For instance, if we commit to terminating 20 idle VMs, that's $X saved per month – track it. The FinOps Foundation encourages celebrating wins and benchmarking them [89] .

Beyond these tactical levers, we'll encourage a mindset shift: *architecture decisions considering cost.* That might include more strategic efforts like using managed services (which could be cheaper overall) or re-architecting an app to be event-driven so it doesn't run 24/7. Those are longer-term but FinOps can highlight where such refactoring might pay off.

By systematically attacking the top sources of waste, Energinet should realize significant savings – as noted, FinOps optimization can lead to **20–30% cloud expense reduction** [14] (which aligns with our goal). More importantly, this frees budget for other projects and fosters a culture of efficient resource use.

**Decision to Make:** For each category, there are choices and trade-offs: - What level of risk vs savings do we take (e.g. how aggressive to be with spot instances, how low to size capacity)? - Do we reinvest savings or reduce budgets accordingly? - Which tools to automate optimizations (do we invest in automation tools or rely on manual efforts initially)? These will be discussed in governance forums as we implement the playbook.

# 7. Forecasting & Budgeting – FinOps Financial Planning Processes

Accurate forecasting and effective budgeting are crucial for financial control, especially given Energinet's public accountability. The aim is to transition from reactive spend tracking to **proactive planning**, while acknowledging cloud's variability.

**7.1 Current Challenges:** We noted forecasting cloud spend is often off by >20%, and budgets are sometimes exceeded without early warning. This aligns with industry concerns – *forecasting variable spend* is a big challenge reported by FinOps teams [54]. Traditional IT budgeting (fixed yearly allotments) doesn't fit well with cloud's elastic model and product-driven usage spikes.

**7.2 FinOps Forecasting Approach:**

We will implement a **multi-layered forecasting method** combining *trend analysis* with *driver-based modeling*:

- **Trend and Time-Series Forecasting:** Use historical spend data (now accessible via our FinOps data platform) to project future costs. We can apply time-series models (even simple ones like exponential smoothing or more advanced if needed) on stable services. For example, if storage is growing at 5% per month historically, project that forward, adjusting for any known upcoming changes.
- **Driver-Based (Business-Driven) Forecasting:** Work with product teams to link cloud costs to business KPIs – for instance:
- If a system's cost is largely driven by number of users or transactions, forecast cost = (cost per user) * projected user count + fixed overhead. If we know a new data initiative (like integrating more IoT sensors) will double data ingestion, we predict the storage and processing costs accordingly.
- Use unit economics metrics (discussed in next section) to scale costs with business plans. E.g., if cost per market transaction is X DKK and the Market team plans a 20% increase in transactions next quarter, forecast that cost portion to increase similarly.
- **Capacity Plans from Teams:** Introduce a lightweight process where each product team provides a quarterly forecast for their cloud usage, informed by their roadmap (new features, expansions). The FinOps team can give them current run-rate and unit cost info to help. This

bottoms-up input often catches things raw trends miss (like a coming one-time load test or an upcoming data retention change).

- **Include Seasonality:** For energy sector, there might be seasonal usage (e.g., higher data processing in winter for some reason, or budget spent heavier at year-end). Ensure models consider seasonal patterns if present.

We aim to forecast at least at a **monthly granularity** for a rolling 12-month horizon. The forecast will be updated quarterly (or monthly for dynamic areas), not just once a year, to adjust to new information.

To manage uncertainty, we'll produce **scenario forecasts**: - **Base case:** expected outcome. - **Best case (low spend):** perhaps assuming aggressive optimization or lower growth. - **Worst case (high spend):** assuming higher usage or delays in optimization. Possibly correlate to business scenarios (if X project is very successful, cloud spend might be high but revenue also high).

This helps set expectation range and plan reserves.

**7.3 Budgeting Process Changes:**

Currently budgets are annually allocated per project or cost center, often based on last year + some buffer. With FinOps: - **Set Budgets by Product/Team:** We will allocate cloud budgets to the teams (showback initially, then move to chargeback with accountability). For instance, "Data & Analytics team has a cloud budget of 5M DKK for FY2025". This is derived from last year actual plus expected growth minus expected savings. The FinOps team will assist Finance in calculating these and negotiating with teams. By giving teams a target, we instill accountability – they need to manage within it, or if business needs more spend, get approval (similar to capacity planning). - **Monthly Budget Tracking:** Use the data pipeline to track actual vs budget burn-down. We'll implement **budget alerts**: e.g., if a team uses >80% of their monthly budget by day 20, send alert to team and FinOps. Cloud providers offer such alerts (AWS Budgets, Azure cost alerts) which we can configure [15]. - **Variance Analysis:** FinOps will analyze any budget variances: Was it a usage surge (more business activity) or inefficiency (something left running)? Provide commentary to Finance – bridging the gap so Finance understands *why* variance happened, not just that it did. A FinOps practice: *trending and variance analysis helps explain cost changes* [90].

- **Quarterly Reforecast:** Each quarter, adjust budgets for rest of year if needed. Cloud usage may shift more than traditional spend, so dynamic adjustment is acceptable (especially as any over/under gets attention quickly). Ideally, we maintain forecast accuracy within 10% so adjustments are minor.

**7.4 Tooling for Forecasting:** - We can start with spreadsheets/Power BI using exported data and scenario modeling. But to improve, evaluate using cloud provider tools (AWS Forecast for billing, GCP Forecasting in their console, etc.) which use ML on the spend trend. Some FinOps platforms also have forecasting modules that combine trending and input factors. - Possibly integrate forecasting into our anomaly detection – if actual deviates from forecast by a threshold, flag it (a kind of anomaly itself).

**7.5 Dealing with Uncertainty & Buffer:** Given regulatory constraints, we might still keep a central "unforeseen usage reserve" budget (maybe 5-10% of total) controlled by CFO, to cover unexpected spikes (like a sudden large project). FinOps aim is to minimize surprises so that ideally actual stays within planned range, avoiding emergency funding requests.

**7.6 Linking to Business Value:** We will also incorporate *forecasted unit costs and ROI*. E.g. if cloud spend is forecast to increase 15% because we anticipate 20% more energy data processed (value to business),

that's acceptable. But if cost increases without proportional value, that triggers deeper review. We'll produce metrics like *cloud cost as % of overall IT spend* (to watch trajectory) and possibly *cloud cost vs revenue or OPEX budget* to ensure it aligns with corporate financial plans.

**7.7 FinOps in Financial Reporting:** Over time, FinOps metrics (like savings realized, forecast accuracy) can be included in CFO reports. This can demonstrate to the board/regulators that we manage cloud spend diligently (like an internal benchmark: "we saved X DKK via optimization this quarter, and reinvested Y in new digital services").

**Forecast Accuracy Goals:** Within 6 months, aim to reduce variance by using the above processes. An interesting stat: Only 29% of cloud FinOps teams expect to meet their cloud goals fully [91] – often due to forecast misses. We want to be in that successful minority by the virtue of strong FinOps planning.

In summary, our approach moves budgeting from static to dynamic, and forecasting from guesswork to data-driven. This ensures **no surprises**: if cloud costs increase, it should either be 1) planned and justified by business growth, or 2) quickly flagged and corrected if due to inefficiency. This proactive stance is critical in Energinet's environment where cost overruns could draw regulatory or public scrutiny.

**Decisions/Dependencies:** Finance will need to adjust processes to allow iterative forecast updates and to empower tech teams with budget ownership. Also, deciding where savings go (reduce spend vs more features) is a strategic business call – FinOps provides options, leadership decides (e.g. in tight times, cut spend; in growth times, reinvest).

# 8. Tooling Strategy – Native vs Third-Party Solutions, Build vs Buy

A robust toolset is essential to implement FinOps efficiently. Energinet must decide the mix of **native cloud tools, custom-built solutions, and third-party FinOps platforms** to use. We analyze options with selection criteria:

**8.1 Native Cloud Provider Tools:** Each cloud offers built-in cost management features: - **AWS:** Cost Explorer (graphs, forecasts), Budgets (alerts), Cost Anomaly Detection, Trusted Advisor (basic optimization recommendations), and CUR data for custom analysis. AWS also has Savings Plans recommendations. Pros: Already available, no extra cost (Cost Explorer free up to a point), deeply integrated (e.g. anomaly detection leverages AWS ML). Cons: AWS-specific, limited multi-account roll-up unless using organizations, and can be clunky for detailed allocation. - **Azure:** Azure Cost Management (also origin from Cloudyn) provides similar cost analysis and budgeting across subscriptions, plus some optimization recommendations and ability to manage AWS costs too (Azure Cost Management can ingest AWS costs for multi-cloud view if configured). Pros: integrated with Azure portal, included in EA. Cons: Multi-cloud support beyond Azure is limited and requires setup, GCP not natively covered. - **GCP:** Cost Tables in BigQuery, Cloud Billing reports, Budgets & alerts, Recommender (gives insights like idle VM, rightsizing, Committed use suggestions). Pros: BigQuery export is powerful for DIY analytics, recommendations are good for GCP environment. Cons: UI less mature, no built-in multi-cloud view.

Given Energinet's multi-cloud use, relying solely on native tools means using them separately – which is a silo. However, we will absolutely leverage them in the short-term for quick wins: - Set up **Budget alerts** in each cloud (quick no-regret move). - Use **native recommendations** as input (e.g. AWS Trusted Advisor's idle resource list). - Use AWS & Azure's anomaly detection if possible to complement our own.

**8.2 Custom "Build" Approach:** This refers to building an in-house FinOps solution using general-purpose tools: - We have already envisioned a custom data pipeline (Section 5). That is part of a build approach. - We can create dashboards in PowerBI or Grafana on top of that data. Grafana especially can combine data from different sources (and we saw Energinet uses Grafana for other monitoring [92]). - We could script automation (like a Python script to purchase RIs as per policy, or Lambda functions to shut down instances, etc.).

**Pros of Build:** Full control over data (important for security/regulation – data stays in our environment), tailored to our exact needs (we can integrate energy-specific metrics easily), potentially lower cost than pricey SaaS tools (though labor costs apply). Also avoids vendor lock-in in FinOps tool.

**Cons of Build:** Development and maintenance effort, requires skilled team (which we are assembling but small), slower to get all features (third-party tools come with many features out-of-box). Risk of re-inventing the wheel for common functionalities.

**8.3 Third-Party FinOps Platforms (Buy):** A number of vendors (Cloudability by Apptio, VMware Aria Cost (CloudHealth), Flexera Optima, CloudZero, DoiT, Ternary, etc.) and even smaller tools like ProsperOps (automated RI management) exist. Features they typically offer: - **Multi-cloud cost aggregation** with one pane of glass (e.g. CloudHealth etc. connect to all clouds). - **Automation and Optimization** – e.g. some tools automatically buy/sell RIs or power off resources (ProsperOps focuses on autonomous RI optimization). - **Advanced Allocation** – map costs to business dimensions, often with nice UI to set allocation rules for shared costs, etc. - **Reporting & Showback** – can generate nice reports per team, often with customizable views for engineering vs finance. - **Anomaly detection and alerts** – usually included. - **Recommendations** – similar to native, but often aggregated and more customizable. Some incorporate community data or best practices from FinOps Foundation. - **Integration** – Many integrate with ITSM (ServiceNow) or Slack for alerts, etc., and have APIs to pull data.

**Pros of Buy:** Quick time to value – immediate insights and ready-made dashboards, less internal effort. Many have FinOps best practices baked in. For example, Apptio Cloudability has a **TBM taxonomy integration** (useful for linking FinOps to TBM). Tools like CloudZero emphasize **unit cost mapping** easily. They might provide ROI by finding things our small team may miss initially. Also support & community from vendor.

**Cons of Buy:** License cost – these tools themselves can cost a percentage of cloud spend or fixed fees (for spend say in our range, could be tens of thousands of EUR per year). That reduces net savings unless they clearly help save more. Some require sending detailed usage data to vendor's cloud – potential compliance/security concern (though vendors typically have ISO certs, but we must consider GDPR if any PII in tags, etc.). Also, lock-in and dependency on vendor roadmap.

**Market Trend:** According to CloudZero stats, *70% have deployed some cost management solution (commercial, OSS or custom)* [18] – meaning many do invest in tools rather than pure DIY. And the Gartner Magic Quadrant for Cloud Financial Mgmt tools now includes big players (Apptio, VMware, Flexera, etc.) [93]. Energinet should consider that as cloud spend grows, a third-party tool can amplify our small FinOps team's capabilities.

**8.4 Tool Selection Criteria:** In Annex F we list criteria, highlights: - **Multicloud & Hybrid Support:** Must handle Azure, AWS, GCP in one interface, and ideally allow input of on-prem or custom costs. - **Integration & Extensibility:** Needs to integrate with our data pipes and possibly output to our systems (APIs, webhooks). If it's too closed, that's a negative. E.g. can we export data to our data lake or BI? - **Security/Compliance:** Data residency (EU if possible), ability to restrict access by role (engineer sees

only their team's costs), SSO integration. - **Features vs Our needs:** Do we need automated RI buying (maybe, if we want to guarantee max savings and offload that logic – ProsperOps does that)? Do we need deep Kubernetes cost allocation (some tools like Kubecost specialized in that, though open-source Kubecost could be enough)? - **Cost of tool:** Many pricing models: some charge % of spend (not great incentive, but common ~2-3%), others flat tier. We'll model ROI – if a tool costs 50k DKK but helps save 500k DKK, it's worth it. But if our FinOps team can achieve near same saving with open scripts, maybe not.

**8.5 Possible Approach:** In the immediate 0-6 month timeframe, **maximize use of what we have and open-source**: - Expand our **custom data pipeline** – giving us core cost analytics capability in-house (which we are doing). - Use **CloudProvider native** alerts and basic dashboards to catch low-hanging fruit. - Possibly deploy **Kubecost (open source)** for Kubernetes cost visibility – it's an emerging must for K8s heavy shops and can feed data to FinOps. - Use tools like **Cloud Custodian (open source)** to write policies as code for cost (e.g. a policy to delete EBS volumes older than X days unattached).

Parallel to that, do a **tool evaluation PoC** with 1-2 leading vendors (maybe get trial access to e.g. Apptio Cloudability or CloudHealth and test how it works with our data). The 2024 Gartner MQ places Apptio, VMware, and Microsoft (with Azure tool) as leaders; we might test those. CloudZero or DoiT might also be interesting since they emphasize engineering-friendliness.

**8.6 Build vs Buy in Long Term:** It might end up hybrid: we continue to "build" a core data repository (we want to own our raw data), but "buy" a UI or specific functionality. For example, we might feed our raw data into a platform for the UI and multi-cloud recommendations. Or use ProsperOps specifically for automated commitment management on AWS (because that tool might manage RIs better than we do manually, saving extra).

**8.7 Tools for Governance Automation:** Not to forget, beyond cost analytics, policy enforcement tools: - We might adopt **Open Policy Agent (OPA)** with policies in code for tagging and cost guardrails. Or use **Azure Policy** and **AWS Config/Service Catalog** to create guardrails (like limiting instance sizes or regions). - **CI/CD integration:** Tools like Infracost (open source) can show a diff of cost in a Git pull request (so dev sees cost impact of their code infra changes). We should trial that for Terraformed infrastructure changes – it's a great developer-friendly FinOps tool.

**8.8 Training & Adoption of Tools:** We must invest in training the FinOps team and engineers on whichever tools we choose. A fancy platform underutilized is wasted. So part of our rollout plan includes workshops on using these dashboards and interpreting them.

**Tooling Comparison Snapshot:** (Detailed in Annex F) We compare factors like **Data Handling**, **Allocation capabilities** (tag-based vs machine learning based – e.g. CloudZero can allocate even without perfect tags using machine learning on patterns [94]), **Kubernetes support**, **Automation** (some do automation vs just reporting), **Cost**.

For Energinet's context (regulated, moderate multi-cloud spend, focus on showback and granular allocation): - Cloudability (Apptio) and CloudHealth (VMware) are enterprise-grade, proven tools with TBM integration (Apptio) and good governance. CloudHealth was earlier widely used for multi-cloud cost. - CloudZero is newer, with a strong focus on unit cost and enabling engineering – might align well with our goal to engage engineers (their blog focuses on cost per product, etc.) [95] [96]. - DoiT (if we qualify as they work differently, being a reseller model with tooling included) – sometimes appealing as they don't charge directly for tool but you route cloud billing through them. - Flexera One/Optima – integrated with broader ITAM, which could be good if we want an all-in-one IT cost suite (and Flexera

has on-prem software expertise too, relevant for hybrid). - Anodot or Zesty – specialized tools (Anodot for anomalies, Zesty for automated right-sizing and elasticity in storage). - Kubecost – we likely will deploy in any case for K8s.

**Recommendation:** *In the short term*, leverage native and open tools, *in the medium term*, likely purchase a FinOps platform for advanced capabilities once basics are in control. The choice will come after some pilots and cost-benefit analysis.

This phased approach is prudent for Energinet: ensure any external tool meets our data governance standards (no sensitive data exposure, preferably EU hosted) and that it demonstrably adds value beyond what our FinOps team can do with existing tools.

By end of the 12-month roadmap, we expect to have an integrated toolset providing: automated cost reporting, real-time anomaly alerts, strong policy enforcement, and accessible views for all stakeholders, without creating security/compliance gaps.

**Decision Needed:** Which platform (if any) to invest in – to be made in Month 3-4 of the roadmap after initial evaluations. We will prepare a formal business case at that time, weighing license cost vs additional savings expected (e.g., an automated RI management tool might pay for itself by capturing 5-10% extra savings that manual might miss).

# 9. KPIs & Reporting – Measuring and Communicating FinOps Value

To ensure FinOps delivers results, we establish clear **Key Performance Indicators (KPIs)** and reporting mechanisms. These KPIs track both financial outcomes and process maturity, creating accountability and continuous improvement.

**9.1 FinOps KPI Catalog:** We propose the following KPIs, grouped by category:

- **Cost Optimization KPIs:**
- **Percentage of Waste (Idle) Spend:** Estimate of cloud spend on underutilized resources (e.g. resources with <10% utilization). Baseline from initial analysis, then target to reduce by X%. A high percentage indicates optimization opportunities. Goal: cut this waste % in half in 6 months.
- **RI/Savings Plan Coverage:** % of eligible compute spend covered by reserved/discount instruments. E.g., “On-demand spend as % of total compute spend”. Higher coverage = better cost efficiency. Target: >70% coverage [14] (from ~30% now).
- **RI/Savings Plan Utilization:** Of the RIs/SPs purchased, how much are actually used (vs sitting idle). Aim for >90% utilization on commitments (indicates right-sizing of purchases).
- **Realized Savings from Optimizations:** Sum of cost savings from implemented optimization actions (compared to baseline or previous configuration). We will track this cumulative figure quarterly (e.g. saved DKK 500k Q1). This shows FinOps tangible value in currency.
- **Allocation & Accountability KPIs:**
- **Tag Coverage of Spend:** % of total cloud spend that is tagged with required tags (or accounted via other means). Target: 95%+ [9] . This will come from our cost data (we can calculate untagged cost portion).

- **Allocation Accuracy (Showback completeness):** % of spend that is allocated to a business unit or product in reports (including shared costs allocated). Should be same as tag coverage if process works, target ~100% (with perhaps explicit "shared pool" accounted).

- **Number of Teams with Cost Ownership:** We can measure how many product teams actively review their cost reports monthly (or how many cost center owners attend FinOps reviews). Goal: 100% of relevant teams engaged (soft KPI, indicates cultural adoption).

- **Financial Management KPIs:**

- **Forecast Accuracy:** Mean absolute percentage error (MAPE) of monthly cloud spend forecast vs actual. Baseline might be 20% error; target <10% within 2 quarters. Could break down by BU as well.
- **Budget Variance (by team or overall):** Actual vs budget each month, perhaps as a percentage or in DKK. Aim to keep within 5-10%. The number of budget breach incidents is also tracked (with an aim of zero surprises).

- **Cost per Unit (Unit Economics):** e.g. cost per market transaction, cost per data point processed, cost per customer, etc. We will define at least one key unit metric for each major product. We track its trend over time (we want it flat or decreasing via efficiencies). For example, if cost per energy trade is 0.1 DKK now, try to maintain or reduce even as volume grows. Another example: cost per 1 MW of balancing capacity managed, to ensure digital costs scale efficiently with grid size.

- **Operational KPIs:**

- **Anomaly Detection & Response Time:** Measure how quickly we detect and resolve cost anomalies. For instance, MTTR (mean time to resolve) an anomaly. If currently we discover cost overruns only at month-end (30+ days), goal is to detect critical anomalies in <24 hours and resolve (or explain) in <3 days. We can log each anomaly alert time and resolution time for this.
- **FinOps Cycle Time:** Time between identifying an optimization and implementing it. If rightsizing opportunities sit for 3 months before action, that's slow. We aim to reduce this cycle – maybe track average age of items in optimization backlog or percentage of recommendations implemented within a month. This drives continuous improvement.
- **Coverage of Cost Reporting:** e.g. fraction of cloud spend under FinOps review (should be 100%). Or number of FinOps reports produced per month (ensuring all BUs get one).

- **Cloud Spend as % of IT Spend:** This might be a high-level indicator for management – showing how cloud spend is evolving relative to all IT costs (just for context and to highlight need for FinOps).

- **Cultural KPIs:** (more qualitative, but we can survey or measure proxies)

- **Engineering Engagement Score:** Possibly via surveys or count of engineers using cost dashboards (if we track dashboard logins or FinOps Slack channel activity).
- **Training Completion:** % of relevant staff who took FinOps training. A well-trained workforce likely correlates to better FinOps outcomes.

Each KPI will have: - **Definition & Formula**: precisely how measured (we will codify these, so definitions are clear and consistent). - **Target & Thresholds**: what is good vs bad (for dashboards, maybe use green/yellow/red). - **Data Source**: e.g. from our FinOps data warehouse (for spend metrics), from budgeting tool (for budgets), from monitoring (for utilization). - **Update Frequency**: many are monthly

(some can be tracked daily like anomalies, but reported monthly). - **Owner**: who is responsible to act on it (e.g. Tag Coverage owned by FinOps lead, cost per unit might be joint FinOps & product owner responsibility).

We'll maintain a **KPI Catalog** (Annex D) with all this information and possibly SQL/Pseudo-code for how to compute (so it's transparent). For example:

```
-- Example pseudo-SQL for tag coverage:
SELECT (Sum(Cost) FILTER (WHERE HasAllRequiredTags='Yes') / Sum(Cost)) * 100
as TagCoveragePct
FROM CostData
WHERE UsageDate BETWEEN ...;
```

or using whatever tool formulas.

**9.2 Reporting Structure:**

- **Executive Dashboard:** High-level, updated monthly, showing overall cloud spend vs budget, top cost areas, and key KPIs (like RI coverage, forecast accuracy, savings achieved). Aim for <= 1 page summary (could even be a PowerBI dashboard the CIO/CFO can view anytime). It will include commentary from FinOps lead ("this month's spend increased 5% due to X, but optimization Y saved Z DKK, forecast on track").
- **Operational Dashboards:** For FinOps team and engineering managers:
- **Team/Service Cost Reports:** Each product team gets a view of their services' costs and KPIs like their tag compliance and unit cost. We'll likely implement these in Power BI or a web dashboard. This is the *showback* – so teams can see what they would be charged.
- **Optimization Tracker:** Could be a simple Kanban or spreadsheet showing all identified optimizations, status, and realized savings – shared with stakeholders so everyone sees progress (and perhaps spurs friendly competition, e.g. which team saved most).
- **Anomaly Alerts:** We'll integrate with communication channels (Slack/Teams or email) for anomalies or budget thresholds. E.g., an automated Slack message to #cloud-costs if any daily spend anomaly beyond threshold.
- **Monthly FinOps Report:** A written report (slides or doc) combining the above data with insights, used in the monthly FinOps meeting. It will highlight KPI trends (maybe sparkline charts of last 6 months for each KPI to see trajectory).
- **Chargeback Statements (if implemented):** Possibly in future, an internal "bill" to each business unit showing their usage and cost. Initially, we'll do showback (informational) not actual internal billing. But formatting it as a bill can drive the point home. This might be an Annex style output where each BU gets a page with breakdown by service, plus share of shared costs (with explanation of allocation method [97] for transparency).

**9.3 Benchmarking:** We will compare some KPIs with industry benchmarks (FinOps Foundation surveys, etc.) – e.g., is our RI coverage above industry median (in SoF report)? Are we tagging better than average? Benchmarking can motivate and contextualize (the FinOps Foundation encourages internal and peer benchmark KPIs [90]). For example, "Industry average tag coverage is ~70%, we are at 95% – a leader" or "Engineering action rate 60%, FinOps Foundation says getting past 50% is good [22]."

**9.4 Communication & Visualization:** - Use simple and intuitive visualizations (stacked area chart of spend by service over time to spot growth, gauge charts for KPI vs target, etc.). Possibly incorporate infographics for exec summary (like cloud spend relative to a baseline). - Focus on *storytelling*: each

metric's trend should be annotated with cause (e.g. "RI coverage dipped this month because new workloads came before we purchased RIs; plan to correct next month"). This turns raw metrics into actionable intelligence. - For engineers, highlight metrics they influence: e.g. cost per deployment or cost per API call might resonate with them more than % of budget.

**9.5 Continuous Improvement:** KPIs will evolve. We might start with a core set and refine definitions as we mature (like adding a carbon cost metric if sustainability becomes priority, or splitting cost per unit by variable vs fixed cost components etc.). We'll regularly review if the KPIs effectively measure the outcomes we want.

One key outcome is to ensure FinOps doesn't devolve into pure cost cutting. The inclusion of **value metrics** (like unit cost vs quality) ensures context. For example, if cost per transaction goes down but customer complaints up, that's not good. So metrics must be balanced and interpreted together.

Finally, by having these KPIs visible, we enforce transparency. Good or bad, the numbers are seen by all stakeholders, creating a natural pressure to improve and a means to celebrate successes (e.g. hitting a savings target or improving forecasting).

# 10. Security, Compliance, and Auditability

In a regulated TSO environment, any new process must uphold or enhance security and compliance. FinOps, when done right, should strengthen governance – but we must manage certain risks:

**10.1 Cloud Governance and Security Alignment:** The FinOps practice will integrate with **cloud governance** policies: - **Least Privilege Access:** Ensure that cost data and FinOps tools follow least privilege. E.g., an engineer sees detailed cost only for their projects, whereas FinOps analysts can see all. We will implement role-based access in dashboards and any FinOps tool. The FinOps data pipeline will reside in a secure environment (e.g., an Azure Data Lake with restricted access). - **Data Sensitivity:** Generally, cost data is not highly sensitive (doesn't contain personal data). However, it can reveal usage patterns or strategy (like increased spend on a project might indicate something). So we treat it as **Company-Confidential**. If exporting data to any third-party FinOps SaaS, we will vet their security (ISO 27001 certification, data encryption, EU data residency if needed, etc.). Possibly include in contract that data is stored in EU data centers, aligning with GDPR and local regulations. - **Policy as Code Enforcement:** Many cost optimizations double as good security hygiene. For example, deleting idle resources reduces attack surface. Enforcing tagging helps track ownership which is useful in incident response. FinOps will coordinate with the Security team so that any automation (like shutting down something) is done safely (e.g., don't turn off an instance that's critical just because it's idle; ensure it's non-prod or confirmed). - We'll integrate with existing security policies: e.g., if a policy disallows certain open ports for security, similarly, we might disallow extremely expensive instance types unless justified. - Use cloud-native **Guardrails** that serve both cost and security: e.g., enforce resource limits (no extremely large VM by accident – reduces blast radius of cost and also ensures sizing is deliberate). - **Change Control:** Some optimizations are changes to infra – we will route these through normal change management if required for production (especially in critical systems). For example, rightsizing a production database is a change that might need CAB (Change Advisory Board) approval in a utility context. FinOps must respect that and plan optimizations into maintenance windows or approved change windows. We'll work with ITSM processes.

**10.2 Regulatory Compliance (NIS2, etc.):** NIS2 focuses on cybersecurity for critical infrastructure. FinOps data and processes should be secured akin to other sensitive systems: - We will include the FinOps pipeline and any tool in the **scope of IT general controls** and audits. For example, ensure

backup and recovery of FinOps data store (so cost data is not lost), and access logs are kept (to see who accessed cost info). - If external regulators or internal audit want evidence of cost control (not typical in NIS, but perhaps in financial audits or performance audits), our FinOps reports and saved analyses become that evidence. We'll maintain an **audit log of cost optimizations** decisions (who approved, what impact). This shows due diligence in managing costs. - **Segregation of Duties:** FinOps practitioners might get recommendations that involve changing cloud resources (like rightsizing). We will maintain separation such that the FinOps team primarily provides recommendations and maybe automation scripts, but application owners execute or at least approve them. This prevents any single person from arbitrarily terminating resources under the guise of cost-saving without oversight (which could be disruptive or abused). - **Financial Compliance:** For accounting, if some cloud costs need to be capitalized vs expensed (depending on nature, though mostly OPEX), FinOps must provide data for proper accounting. If FinOps decides to pre-pay cloud for discounts (like 3-year RIs), that's a financial decision needing CFO approval and proper accounting (prepayments). We'll ensure these are documented and align with accounting policies.

**10.3 Auditability:** Every number we present in FinOps should be **traceable**: - From a showback report line item ("Application X spent 1000 DKK on compute in Jan") one should be able to trace to the raw billing data for that app and see how that 1000 was computed (100 hours of VM type Y at rate Z, minus RI discount, etc.). Our data pipeline design with preserved detail supports that. - We will also document allocation rules (e.g., if we allocate shared costs by 50/50, that's stated). This way, if a business unit challenges a charge, we can show the basis and they can audit it. Transparency reduces disputes in chargeback [98] . - **Internal Audit** might review cloud spending. We will preempt this by engaging them early – showing our FinOps processes and controls (like "we verify invoice vs usage, we have approval for RI purchases, etc."). A control example: any cloud commitment > certain value requires CFO sign-off – we'll formalize that. - For **security audits**, if using any SaaS FinOps tool, include it in vendor security assessments.

**10.4 Business Continuity and Risk Mitigation:** Cloud spend management might not seem like a continuity issue, but consider: - If FinOps fails (say budgets are blown), it could threaten funding for operations. So from a risk perspective, FinOps is now part of enterprise risk management. We include FinOps-related risks in our risk register (Annex G), such as: - Risk of **Cloud Cost Overrun** – Mitigation: FinOps monitoring and alerts. - Risk of **Optimization causing incident** – Mitigation: testing, change control, rollback plans. - Risk of **Data misreporting** – Mitigation: reconciliation controls, dual checks between FinOps data and invoices. - Risk of **Vendor lock-in** by commit – Mitigation: careful analysis before long-term commits, diversify across clouds where sensible, and include exit clauses.

**10.5 Privacy Considerations:** Although not a primary concern here, ensure any personal data possibly in tags (e.g., if `Owner` tag has a person's name) is handled per GDPR – likely fine as internal use, but if we expose FinOps data widely, maybe mask or limit PII in reports outside need-to-know. We can use team names instead of individual names, for example, to avoid any privacy issue with employee names in broad distribution.

**10.6 Compliance with Purchasing Policies:** Committing to cloud spend (like enterprise agreements, RIs) is a form of contract. We'll route those through procurement properly. FinOps will work with procurement so that cost-saving initiatives also comply with public procurement rules if any (though cloud providers likely sole source in many cases, but large spends might require justification).

**10.7 Environmental/Sustainability Compliance:** An emerging area – as a TSO, Energinet cares about green energy. Cloud usage has a carbon footprint. FinOps can integrate with sustainability by tracking CO2 emissions of cloud usage (Azure and AWS provide carbon tracking tools). If any regulatory or corporate goal exists to reduce carbon, FinOps will align (e.g. by including carbon cost per unit metrics

or preferring efficient infrastructure). This is not a direct compliance need yet, but we foresee it as part of corporate social responsibility.

In summary, we treat security and compliance as *non-negotiable constraints* in FinOps design. Done right, FinOps will **increase auditability** of IT spend (a plus for compliance) and impose more discipline (like tagging and change control) that aligns with strong governance. Each FinOps initiative will be evaluated for compliance impact (a checklist in each optimization runbook: e.g., "does turning off this system violate any policy or SLA?").

By integrating FinOps into our compliance framework, we ensure we reap cost benefits *without* exposing Energinet to operational or regulatory risk.

# 11. Roadmap & Change Management – Phased Implementation Plan

Implementing FinOps is a journey. Below is a phased **12-month roadmap** with 30/60/90-day plans and beyond, structured to deliver quick wins and build long-term capabilities:

**Phase 0: Preparation (Month 0)** – *Right now*: - Secure executive sponsorship ( done, CFO and CIO on board). - Assign interim FinOps lead (perhaps someone from Cloud CoE) and form initial working group (IT ops, finance rep, etc.). - Kick-off meeting to align on vision and priorities.

**Days 1–30 (Month 1): Quick Wins & Foundation Setup**: - **Establish FinOps Core Team (Interim):** Identify at least 2 people who will spend significant time on FinOps (maybe reallocating existing staff initially). Begin skill training (e.g. FinOps Foundation Practitioner training) [99] . - **Turn on Basic Alerts:** Implement cloud-native budget alerts for all major accounts/projects (so any dramatic spike triggers email/SMS) [15] . - **Gather Data & Identify Obvious Waste:** Use AWS Trusted Advisor, Azure Advisor reports to make a **Top 10 easy fixes list** (e.g. remove X unattached volumes, shut Y test VM). Execute these after quick validation, realizing immediate savings. - **Tagging Policy Draft:** Publish updated tagging policy (even if enforcement isn't there yet) and communicate to all dev teams that new resources must have tags. Configure one or two Azure Policies or AWS Config rules as a pilot to test tag enforcement in a dev subscription. - **Baseline Measurements:** Record current metrics (tag coverage%, RI coverage, etc.) as baseline. Also complete the current state cost analysis (already mostly done for this report). - **Tool Assessment Kickoff:** Compile list of potential FinOps tools. Perhaps schedule two vendor demos in next 2 months. - **Communications:** Company-wide announcement of the FinOps initiative, including a brief "what's in it for you" for engineers (emphasize enabling innovation through savings, not just cost cutting).

By end of Month 1, aim to show some savings (even minor, like "we saved €5k by cleaning X") to build credibility. Also, potential cost avoidance: e.g., catch an anomaly early that otherwise would've cost more.

**Days 31–60 (Month 2): Crawl Phase Building**: - **Data Pipeline MVP:** Set up the cost data ingestion for one cloud (Azure likely, as it's primary). Build a simple Power BI report on it. This provides first integrated view for FinOps team. - **Define KPIs & Dashboard 1.0:** Finalize the KPI list and start designing the FinOps dashboard. Populate it with last 3 months data to see trends. - **Operationalize FinOps Meetings:** Hold the **First FinOps Monthly Review** at end of Month 2. Present findings and the optimization backlog. Ensure at least one actionable decision is made (e.g. "Team A will commit to rightsizing instances next sprint with FinOps help"). - **Train & Engage Teams:** Conduct an initial training session with all product teams on reading cost reports and the importance of tagging. Possibly invite an

external FinOps expert or use FinOps Foundation materials. - **Continue Quick Optimizations:** Expand to next set of optimizations (e.g. schedule shutdowns implemented via an automation script in dev). - **Select FinOps Tool (if going for trial):** Based on demos, pick one platform to do a trial/Pilot in next phase (maybe trial Cloudability on AWS costs, for example).

**Days 61–90 (Month 3): Deeper Dive and Walk Phase Initiation**: - **Multi-cloud Data Integration:** Incorporate AWS (and GCP if possible) into the cost data pipeline. Now we have a consolidated view. Validate it against actual bills for accuracy. - **Formalize FinOps Core Team:** By now, decide if we need to hire or permanently allocate personnel. Perhaps open a role for a FinOps analyst if current staff cannot cover. - **Policy & Automation Implementation:** Write a few **Cloud Custodian** or custom scripts policies focusing on high-impact items (e.g. a policy that deletes unattached disks after 14 days of no attachment, with notification to owners). - **Pilot FinOps Tool:** If we opted to trial a third-party, integrate it with read-only access to our accounts and let it collect data now that tagging improved. Evaluate its findings vs our manual ones. - **Announce Early Results:** At end of Q1 (90 days), share a brief with leadership: e.g., "FinOps has identified X DKK of annual savings, Y DKK already realized; forecast accuracy improved from 25% error to 15%; tag coverage up from 60% to 85%." Even if not all targets hit, show direction. - **Roadmap Refinement:** Based on first 3 months, adjust the plan for next quarters (maybe some things are harder, or new issues discovered).

**Months 4–6 (Q2): Expand and Solidify (Walk stage)**: - **Chargeback/Showback:** Implement **showback reports** to each business unit monthly. They get their "bill" for information. Gather feedback, work out disputes in allocation. - **Optimization at Scale:** Start tackling bigger items like RI purchases or architectural changes. By month 6, we likely execute at least one RI/Savings Plan purchase capturing a chunk of savings, guided by FinOps analysis (with CFO approval). - **Tool Decision:** By end of Q2, decide whether to procure a FinOps platform (based on pilot results and ROI). If yes, begin procurement for deployment in Q3. - **FinOps Champion Network:** Identify FinOps Champions in each department (if not already). Start a bi-weekly sync with them for knowledge sharing (like a FinOps Guild). - **Integrate FinOps into Dev Cycle:** Update DevOps processes: e.g. add cost estimation step in architecture docs, add cost checks in CI (with Infracost). - **Forecasting Cadence:** For second half of year budget, apply the new forecasting method. Provide updated forecasts to Finance and see improvement (this is test of our forecasting process). - **Celebrations & Culture:** Perhaps gamify optimization – e.g. internal recognition for "best optimization of the quarter" to encourage teams. Share success stories (like "Team X managed to double their users with only 20% cost increase – exemplary use of cloud efficiency").

**Months 7–9 (Q3): Optimize and Operate stage**: - **Advanced Capabilities:** If FinOps tool acquired, implement advanced features (like automated anomaly detection beyond our own, more granular chargeback via the tool). - **Policy Enforcement Tightening:** Move from gentle tagging reminders to enforcing compliance (e.g. resources cannot be created in prod without tags – enforce via pipeline). - **Continuous Improvement:** Revisit KPIs: any targets not being met gets focused initiatives. E.g., if forecast accuracy still off, maybe invest in better modeling or adjust processes. - **Unit Economics Deep Dive:** By month 9, produce a **unit economics report** for one key product (e.g. cost per energy data publication on our open data platform). Use it as a success example for value discussions. - **Cross-Functional Wins:** Possibly hold a **"FinOps showcase"** meeting where teams present how they optimized costs and improved delivery (embedding FinOps in pride of work). - **Adjust Budgets if needed:** If FinOps is generating lots of savings, discuss with finance whether to reduce budgets accordingly or reallocate funds to new initiatives (ensuring captured savings benefit organization goals). - **Plan Next FY:** Start planning FinOps roadmap for next year: scaling team if needed, deeper cloud cost optimization like contract negotiation with cloud vendors (if spend large enough, maybe consider enterprise discount plans that require forecast – FinOps will provide the data needed to negotiate best deals).

**Months 10–12 (Q4): Maturity and Handover to BAU**: - **Institutionalize FinOps:** Update official IT policies to include FinOps processes (like every project must involve FinOps review, tagging policy is part of definition of done, etc.). Make FinOps part of the organizational fabric. - **Complete documentation:** Build a FinOps playbook and knowledge base internally so processes are clear (especially as we likely will onboard new people). - **Evaluate Results vs Targets:** At month 12, do a comprehensive review: Did we hit the 15-25% savings? Did we get forecast accuracy <10%? Use KPIs to measure success. Prepare results for executives and possibly external stakeholders if relevant (e.g. mention in annual report that cloud cost optimization saved X million DKK, showing stewardship). - **Next Steps:** Identify remaining gaps for the next year's plan (maybe focus on optimizing new areas like AI/ML costs or deeper unit economics, or extending FinOps to other expenses). - **Broaden Scope:** Perhaps extend FinOps practices to other IT spend categories (e.g., SaaS costs management, on-prem cost optimization) – essentially merging with TBM practices, achieving a holistic view.

**Gantt Overview (see Annex G)**: We would have a timeline chart aligning tasks to months, ensuring not to overload small FinOps team. It highlights parallel tracks: *Data/Tooling*, *Process/Governance*, *Optimization Activities*, *Cultural Change/Training*. Key milestones like "First showback report delivered (Month 4)" and "Tool live (Month 7)" etc.

**Change Management Considerations:** - Frequent communication and feedback loops – after each milestone, get feedback from teams and adjust approach if there's resistance or confusion. - Manage any fear – sometimes teams fear FinOps = budget cuts. We must communicate success stories and ensure this is seen as enabling (e.g., if a team saves money, perhaps allow them to redirect some to tech debt reduction or innovation – positive reinforcement). - Provide support – FinOps team as a service to teams, not just governance. E.g., FinOps analyst sits with a dev team to help them optimize, rather than just sending them a JIRA ticket to do it. - Leadership reinforcement – ensure leaders incorporate cost awareness in their dialogues (e.g. CTO asks in project meetings "have we considered cost implications?" signaling its importance). - Possibly incorporate FinOps goals into performance objectives (softly, e.g., product owner's objectives include operating within budget, engineers' objectives include tagging compliance or addressing cost recommendations).

This phased approach ensures we **deliver incremental value quickly** (so FinOps pays for itself early via quick wins) and builds towards a sustainable practice by year-end. It also allows flexibility – at each stage, based on results, we might adjust pace or scope.

By the end of 12 months, the aim is FinOps is no longer a project but **business-as-usual (BAU)** in Energinet's operations, with continuous improvement mindset for the future.

# 12. Business Case & ROI – Value Proposition of FinOps

Finally, we quantify the expected benefits (cost savings and beyond) and outline the ROI of the FinOps investment.

**12.1 Investment Required:** - **People:** FinOps Core Team ~2-3 FTEs (if new hires or reallocated) – cost ~€200k/year (just as an estimate loaded cost). - **Tools:** If we buy a FinOps platform, might be ~2% of cloud spend or a flat €50k/year (for example). If not, we invest in development effort for the data pipeline (some cost but mostly internal effort). - **Training/Change:** Minor costs for training materials or certification (~few thousand €). - So roughly, first-year direct investment might be on the order of €250k–€300k (approx. 2 million DKK), primarily staff and possibly tool license.

**12.2 Quantifiable Savings:** - **Resource Optimization Savings:** By eliminating waste (idle, oversized, orphaned) we estimated earlier ~15-25% savings of current spend. Suppose Energinet's cloud spend is DKK 20M/year (just hypothetical). 20% of that is DKK 4M saved. - **Rate Optimization Savings:** Using RIs and Savings Plans could shave off another chunk – e.g. if 50% of spend qualifies for 30% discounts, that's 15% of spend -> DKK 3M. - **Avoided Growth Cost:** Without FinOps, cloud spend might grow ~30% YoY due to increased usage (some industry figures show that growth). With FinOps efficiencies, maybe we handle that growth with only 15% cost increase. That avoidance is savings vs what it would have been. But to be conservative, stick with actual reductions. - Realistically, not all savings are immediately cash in hand (some is cost avoidance). But often FinOps can quickly capture ~**10-15% reduction** in bills in first cycles [19], and more over time.

Given a mid-case scenario: Save DKK 3-5 million in year 1. That is roughly 10x the direct FinOps cost (if team cost ~0.5M DKK each, 2-3 ppl ~1.5M DKK + maybe 0.5M tool = 2M DKK cost vs ~4M DKK savings). So ROI ~200% within first year (i.e., benefits double the cost). This aligns with industry cases: FinOps can pay back within months. Softjourn citing McKinsey notes 20-30% savings potential [19], which is significant, and *payback in 1-3 years* for cloud investments overall [100].

**12.3 Non-Financial Benefits (Value-add):** - **Faster Decision-Making:** With better visibility, teams can make decisions quicker (e.g. choosing architecture alternatives by cost impact). This agility has value – perhaps new product features get green-lit faster because cost is known and acceptable (removing an unknown). - **Predictability & Risk Reduction:** Reducing cost surprises means less firefighting and no need to halt projects due to budget overrun. It also means stable financial performance, which for a quasi-government entity is valuable (no negative press about IT cost overruns). - **Improved Collaboration:** FinOps by nature fosters more collaboration between departments, breaking silos. This cultural improvement can have spillover benefits in other areas of IT (like DevOps culture improvements). - **Capacity for Innovation:** Money saved is money that can be reinvested. For example, if we save DKK 4M, maybe DKK 2M goes back to TSO's bottom line (benefitting ratepayers or profits) and DKK 2M put into new digital initiatives (like grid modernization tools, analytics for green energy integration, etc.). Those initiatives in turn could provide further value to society or future cost savings. - **Transparency and Trust:** Internally, FinOps increases trust between Finance and Engineering (finance sees engineers are taking accountability, engineers see finance supporting their needs). Externally, if asked by oversight bodies to justify cloud spend, we can confidently show data and actions taken, enhancing our credibility as prudent operators. - **Performance and Reliability (unexpected but possible):** Some optimizations can improve system performance (e.g. cleaning up resources reduces noise, rightsizing can actually improve performance if done properly by matching workload needs). Also, scheduling dev environments off means less chance of someone tinkering at off hours that could cause an incident – a stretch, but sometimes lean environments promote better discipline.

**12.4 Simple ROI Model:** We will track a formal business case: - Costs: FinOps team salaries, tool subscription, etc. - Benefits: monthly realized savings (tangible), plus any cost avoidance quantified. - Compute Net Present Value (NPV) if needed over 3 years, but likely ROI is high so even Year 1 justifies it.

For example, Year 1: invest DKK 2M, save DKK 4M -> net +2M (ROI 200%). Year 2 might invest similarly but savings might accumulate or even increase as more advanced optimizations come (like architectural changes yield compounding savings).

**12.5 Sensitivity:** If cloud spend growth is higher than expected, FinOps's value grows (more to optimize). If cloud spend is low or flat, FinOps still helps in absolute terms but ROI ratio lower (however, FinOps team could be smaller in that case). We will right-size FinOps effort to spend scale. FinOps Foundation suggests ~1 FinOps practitioner per $10M cloud spend as a rough idea; we are in that ratio.

**12.6 Opportunity Cost:** Without FinOps, likely cloud spend would continue to rise unchecked, and we'd either have to cut functionality or blow budgets. With FinOps, we preserve ability to grow digital initiatives within budgets. If we consider that in a regulated environment overspend might lead to regulatory sanction or need to increase tariffs (both negative), FinOps indirectly avoids those pains.

**12.7 Intangible Value:** It's worth noting what FinOps is not meant to do: it's *not about headcount reduction* or such. It's specifically optimizing resource spend. So it's a focused ROI.

We will deliver a final business case document and "Quick Start" one-pager for team leads (Annex H might contain a quick guide "What to do this quarter: 5 steps to kick off FinOps in your team").

The quick start might say: - Review your team's cloud costs in provided report. - Identify one low-hanging optimization (with FinOps help). - Ensure all your resources are tagged by end of quarter. - Set an alert for your spending. - Attend the monthly FinOps meeting to share progress.

This ensures engagement from the get-go.

**12.8 Sustaining ROI:** After initial big savings, FinOps focus may shift to **cost avoidance and value optimization**. We'll continue to show ROI by maybe linking FinOps to enabling more projects. E.g., FinOps saved 1M DKK that was used to deliver a new analytics feature that yields operational benefits (like better wind forecast – not a direct financial gain but mission gain). It might be beyond pure cost, but we can narrate that value conversion.

In conclusion, the business case for FinOps at Energinet is strong: **tangible cost savings well in excess of the investment, improved financial governance aligning with our public service mandate, and enabling continued digital innovation in a cost-effective way**. As cloud spending grows globally (~21% growth to $723B by 2025 [101]), FinOps ensures we grow efficiently, with likely ~20-30% of cloud spend being optimizable [102] which we intend to capture.

We will monitor and report ROI periodically to ensure FinOps program stays on track and has ongoing support. Based on industry surveys, *89% of stakeholders identify FinOps as key to managing cloud complexity* [103] – our business case substantiates that for Energinet, FinOps is not just an optional exercise but a necessary competency for the cloud era.

# 13. Risks & Mitigations

No transformation is without risk. We list major risks to FinOps success and how we'll mitigate them (Annex G will have a detailed risk register with likelihood/impact):

- **R1: Lack of Engineering Buy-in** – If engineers perceive FinOps as bureaucratic or "budget police," they may not cooperate (e.g., ignoring tag policies or not acting on recommendations). *Mitigation:* Emphasize collaboration and empowerment as per FinOps principles [26]. Include engineers in decision making (FinOps Champions). Make tooling self-service and helpful (not just reports to their boss). Also, show how FinOps can improve their work (like funding their projects or making their lives easier by eliminating wasted efforts). We've already planned training and cultural incentives to address this.

- **R2: FinOps Becomes Siloed (just a Finance thing)** – If the FinOps team works in isolation, the practice could revert to old silo (Finance telling IT what to do) and fail. *Mitigation:* Keep cross-

functional governance (mix of finance and tech in the team and meetings). Possibly rotate engineers through FinOps team for a stint to cross-pollinate.

- **R3: Data Inaccuracy or Gaps** – If our cost data is wrong, decisions could be wrong or trust lost. *Mitigation:* Rigorous reconciliation with cloud bills [104], iterative improvements in data quality, backup of manual tracking as sanity check early on. Also, not over-promising precision – communicate when something is estimate (like shared cost allocations).

- **R4: Savings Not Realized (FinOps "theater")** – There's a risk we identify savings but they aren't executed (maybe due to fear of impact or bureaucracy), so stakeholders don't see actual financial benefit. *Mitigation:* Tie recommendations to action owners and follow up. Escalate if necessary (e.g., if a team consistently doesn't implement, bring it up in governance to get exec support). Also prioritize easy wins first to build momentum of execution.

- **R5: Over-optimization Impacting Reliability** – If we cut too deep (e.g., scale down and then a traffic spike causes an outage), FinOps could be blamed for hurting service. *Mitigation:* Always review recommendations with service owners to assess risk. Keep reliability requirements as input (in FinOps backlog, mark items that require careful testing). Better to err on safe side initially – e.g. do phased rightsizing rather than drastic, monitor after changes.

- **R6: Cloud Price Changes or External Factors** – Cloud providers might change pricing models, which could upset our plans (e.g., introduce new savings plan types, or, say, significant energy cost changes that inflate cloud prices). *Mitigation:* FinOps stays current via community (FinOps Foundation updates, etc.) and adjusts strategy. Also, maintain some buffer in budget for unforeseen changes.

- **R7: Internal Resistance on Chargeback** – If we move to chargeback, some departments might resist being charged IT costs or gaming the system (like trying to push costs to a shared account). *Mitigation:* Ensure fairness and transparency in allocation methods, get leadership mandate for chargeback, and implement gradually with showback first to soften the ground.

- **R8: Talent/Capability Risk** – FinOps skills are new; finding or developing the right people could be a challenge. *Mitigation:* Invest in training existing staff (cheaper and retains domain knowledge). Use FinOps Foundation resources (certifications, community forum). Possibly share resources with other Nordic TSOs if any synergy (as a knowledge exchange). If needed, bring in a consultant initially to jump-start.

- **R9: Too Many Tools or Complexity** – Implementing a heavy platform might overwhelm teams or duplicating functionality (if we have our own pipeline and a vendor tool, could confuse). *Mitigation:* Keep it simple to start (our phased approach), ensure any new tool either replaces an existing process or clearly adds value, and integrate it well (one source of truth concept).

- **R10: Security Breach or Compliance Violation** – e.g., sensitive cost data (if any) gets exposed, or FinOps automation accidentally deletes something critical outside of change window. *Mitigation:* Align with security team as planned, thoroughly test automations in non-prod, use role-based permissions for any automation (e.g. an automation account that only has rights to specific actions). And ensure fallbacks – for example, if an automated cleanup script is too aggressive, have versioning or backups to recover.

Each risk is assigned an owner (often FinOps Lead or relevant manager) to monitor and enforce mitigation. We'll include FinOps in the enterprise risk register with periodic review.

**Monitoring Signals:** We will define triggers to catch risks early: - For culture risk: do an anonymous survey mid-year to gauge engineering sentiment about FinOps. - For data risk: set up alerts if pipeline data deviate wildly from last known good (could catch an ingestion error). - For execution risk: track how many recommendations are unresolved after X time – if backlog piles up, indicates execution gap.

By proactively managing these risks, we increase the likelihood of a sustainable FinOps practice delivering on its promise.

---

**Conclusion:** Energinet's adoption of FinOps will create a **financially agile, accountable, and efficient cloud operating model**. By following this detailed roadmap and recommendations, Energinet can expect significant cost savings (20-30%), improved forecasting accuracy, and a stronger culture of cross-functional collaboration – all while maintaining the high reliability and compliance standards required of a national grid operator. FinOps is a journey, but one that positions Energinet to continually do more with its cloud investments, ensuring every krone spent on cloud ultimately translates into value for Danish consumers and the energy transition mission.

**Next Steps:** Approve the formation of the FinOps function and kick off Phase 1 actions (as per Section 11 roadmap). Establish the initial FinOps review meeting and begin tracking the recommended KPIs immediately to start measuring progress.

---


1 2 23 24 43 99 FinOps Foundation - What is FinOps?
https://www.finops.org/introduction/what-is-finops/

3 4 56 57 58 59 60 61 62 63 71 72 FinOps Team Structure - Infracost
https://www.infracost.io/glossary/finops-team-structure/

5 6 64 65 66 67 68 Establishing a FinOps Decision & Accountability Structure
https://www.finops.org/framework/previous-capabilities/decision-accountability-structure/

7 8 77 83 Adopting FOCUS™, the FinOps Open Cost and Usage Specification
https://www.finops.org/wg/adopting-focus-the-finops-open-cost-and-usage-specification/

9 10 11 12 17 21 22 53 79 80 86 91 94 95 96 9 Cloud FinOps Challenges — And The Solutions To Overcome Them
https://www.cloudzero.com/blog/cloud-finops/

13 14 69 70 FinOps 101: Transform Cloud Cost Challenges Into Opportunities
https://www.spyglass.net/cloud-finops-savings/

15 FinOps for Builders: Part 4 — Sustain & Scale with Culture and ...
https://raghunitb.medium.com/finops-for-builders-part-4-sustain-scale-with-culture-and-continuous-improvement-3eb53afb55b0

16 AWS bill shock - the complete guide - Finout
https://www.finout.io/blog/bill-shock

18 52 55 100 21+ Stunning FinOps Statistics You Need To Read
https://www.cloudzero.com/blog/finops-statistics/

19 101 102 103 The State of FinOps 2025: FinOps Stats
https://softjourn.com/insights/finops-stats

20 Energinet keeps the lights on with a digital platform built on Azure
https://www.microsoft.com/en/customers/story/1720678492723992375-energinet-government-azure-en

25 26 27 28 29 30 31 32 84 87 89 90 FinOps Principles
https://www.finops.org/framework/principles/

33 34 What Is FinOps? Definition, Principles, Benefits, and More
https://ternary.app/blog/what-is-finops/

35 36 37 38 39 40 41 42 FinOps Phases
https://www.finops.org/framework/phases/

44 45 46 47 48 49 50 93 TBM and FinOps: How They Work Together - Apptio
https://www.apptio.com/blog/tbm-and-finops-how-they-work-together/

51 TBM & FinOps
https://www.tbmcouncil.org/discover/hot-topics/tbm-finops/

54 2025 - The State of FinOps
https://data.finops.org/library

73 74 75 76 78 82 85 104 How to Build and Optimize FinOps Data Workflows
https://www.finops.org/wg/how-to-build-and-optimize-finops-data-workflows/

81 FinOps in Real Time: How Data Streaming Transforms Cloud Cost ...
https://www.kai-waehner.de/blog/2025/07/03/finops-in-real-time-how-data-streaming-transforms-cloud-cost-management/

88 How to Optimize Cloud Usage
https://www.finops.org/wg/how-to-optimize-cloud-usage/

92 How Denmark's Energinet uses Grafana Enterprise to monitor ...
https://grafana.com/blog/2022/08/19/how-denmarks-energinet-uses-grafana-enterprise-to-monitor-underwater-energy-cables-and-do-detective-work/

97 Managing Shared Cloud Costs - The FinOps Foundation
https://www.finops.org/wg/identifying-shared-costs/

98 The State of FinOps - The FinOps Foundation
https://data.finops.org/