ZETALY®

WHITE PAPER

What is Mainframe FinOps?

Mainframe FinOps is the discipline of applying financial operations (FinOps) practices to IBM z/OS mainframe environments — bringing real-time cost visibility, proactive budget control, and automated capacity optimization to one of enterprise IT's largest and most opaque cost categories: Monthly License Charges (MLC). For organizations running mission-critical workloads on IBM mainframe, software costs can represent tens of millions of dollars annually — and monthly software license fees alone typically account for 30–50% of the total mainframe budget. Yet most teams only see those costs at month-end, after billing peaks have already occurred and opportunities to optimize have passed. Mainframe FinOps changes that.

In this guide, we will cover

- What mainframe FinOps is — and how it differs from cloud FinOps
- The mainframe's role in global enterprise computing, by the numbers
- The IBM pricing models that make mainframe cost management uniquely complex
- How SCRT and SMF data underpin every mainframe billing calculation
- ISV software licensing and its impact on total mainframe costs
- The three pillars of a mainframe FinOps practice
- Who is involved: key roles and responsibilities
- How enterprises implement mainframe FinOps in practice
- What good results look like — and the tools that deliver them
- Mainframe FinOps during modernization and cloud migration
- Frequently Asked Questions
- References and Further Reading

1- The Mainframe Cost Problem

The IBM mainframe remains deeply embedded in global enterprise computing. It processes the majority of the world's financial transactions, runs national healthcare systems, and powers the core banking platforms of most major banks. For the organizations that depend on it, a full migration away from mainframe is a multi-year — often decade-long — undertaking. In the meantime, the costs keep running.

Mainframe software costs are notoriously difficult to manage. Unlike cloud services — where costs are visible in near real time and can be turned off instantly — mainframe billing is governed by complex IBM contracts, calculated on rolling usage averages, and often only fully understood after the invoice arrives.

The result is a familiar pattern: IT teams discover a billing overrun weeks after it happened, with no clear picture of what caused it or which workload drove the peak. Finance teams receive mainframe invoices they can't reconcile against business activities. And capacity managers make conservative decisions — either over-provisioning to avoid capping critical workloads, or accepting unnecessary cost inefficiency.

Mainframe software costs — particularly Monthly License Charges (MLC) — are one of the largest and most opaque line items in enterprise IT budgets. For large enterprises, annual MLC spending can range from $10 million to over $100 million. Traditional approaches leave teams reactive: managing costs after the fact rather than in real time.

According to Forrester's State of Mainframe, Global, 2025, mainframe footprints are growing at many organizations, and mainframe budgets are increasing even as modernization efforts continue in parallel. This reflects a practical reality: most enterprises run mainframe and cloud simultaneously for many years, not as a binary choice. That dual-running period is precisely when mainframe FinOps delivers the most value.

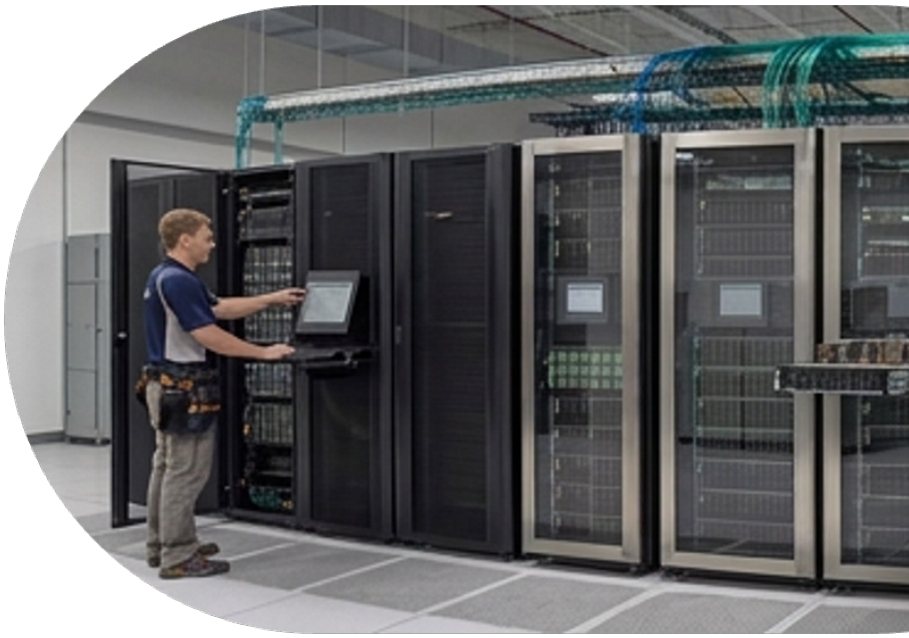
Mainframe FinOps addresses this directly. It applies the same principles that have transformed cloud cost management — visibility, accountability, continuous optimization — to the specific constraints and pricing structures of IBM z/OS.

2- The Mainframe by the Numbers

The mainframe's role in global enterprise computing is often underestimated. These figures illustrate why mainframe cost management is one of enterprise IT's most consequential financial disciplines.

A note on sources: several of the statistics below originate from IBM and Broadcom, who have a commercial interest in the mainframe's continued relevance. We include them because they are the most frequently cited figures in the industry, but we flag the source context so readers can weigh them accordingly. Where independent sources are available, we reference them directly.

- Mainframes handle approximately 68% of the world's production IT workloads and power an estimated 74% of global transactions, yet account for only about 8% of total IT costs. [Broadcom]
- 90% of global credit card transactions are processed on mainframe systems. [IBM]
- 95% of financial institutions and 71% of Fortune 500 companies rely on mainframes for core operations. [Broadcom]
- The mainframe market was valued at approximately USD 5.33 billion in 2025 and is projected to reach USD 7.54 billion by 2031, growing at a 5.95% CAGR. [Mordor Intelligence]
- IBM Z systems held approximately 63% of the mainframe market in 2025, and IBM reported its highest annual zSystems revenue in two decades for fiscal year 2025. [Mordor Intelligence; IBM Q4 2025 Earnings]
- Monthly software costs (MLC) typically represent 30–50% of the total mainframe budget, making it the single largest controllable mainframe cost category. [BMC]
- The mainframe modernization market — services, tools, and platforms for migrating or transforming mainframe workloads — is valued at USD 8.18 billion in 2025. [InsightAce Analytic]



3- What is FinOps — and Why Does Mainframe Need Its Own Practice?

FinOps, as defined by the FinOps Foundation, is «an operational framework and cultural practice which maximizes the business value of technology, enables timely data-driven decision making, and creates financial accountability through collaboration between engineering, finance, and business teams.» [FinOps Foundation]

FinOps emerged from the cloud world, where infrastructure costs became dynamic and distributed across teams, making traditional IT budgeting approaches inadequate. The FinOps Framework organizes this practice into three phases — Inform, Optimize, and Operate — and six core principles:

1. Teams need to collaborate
2. Business value drives technology decisions
3. Everyone takes ownership for their technology usage
4. FinOps data should be accessible, timely, and accurate
5. FinOps should be enabled centrally
6. Take advantage of the variable cost model of the cloud

Notably, the FinOps Foundation's 2025 update explicitly extended the framework's scope beyond public cloud to cover licensing, data centers, SaaS, and other technology cost categories — formally recognizing that mainframe and on-premises environments are now part of the FinOps discipline. The 2025 update introduced «Scopes» as a core framework element, with Data Center identified as one of the initial Scopes alongside Public Cloud and SaaS. [FinOps Framework 2025 Update; FinOps Scopes]

The 2026 Framework update, published in March 2026, went further — introducing «Executive Strategy Alignment» as a new Capability, creating dedicated Technology Category pages for Data Center, AI, SaaS, and Data Cloud Platforms, and renaming several Capabilities to be technology-inclusive rather than cloud-specific (for example, «Workload Optimization» became «Usage Optimization» and «Architecting for Cloud» became «Architecting & Workload Placement»). [FinOps Framework 2026 Update]

These framework updates reflect the practitioner community's recognition that FinOps principles apply wherever significant technology spend occurs — not only in public cloud. That said, the practical case for mainframe FinOps doesn't depend on any trade body's endorsement. It rests on a simple reality: mainframe software costs are large, complex, and almost universally undermanaged. The framework provides useful structure; the business case stands on its own.

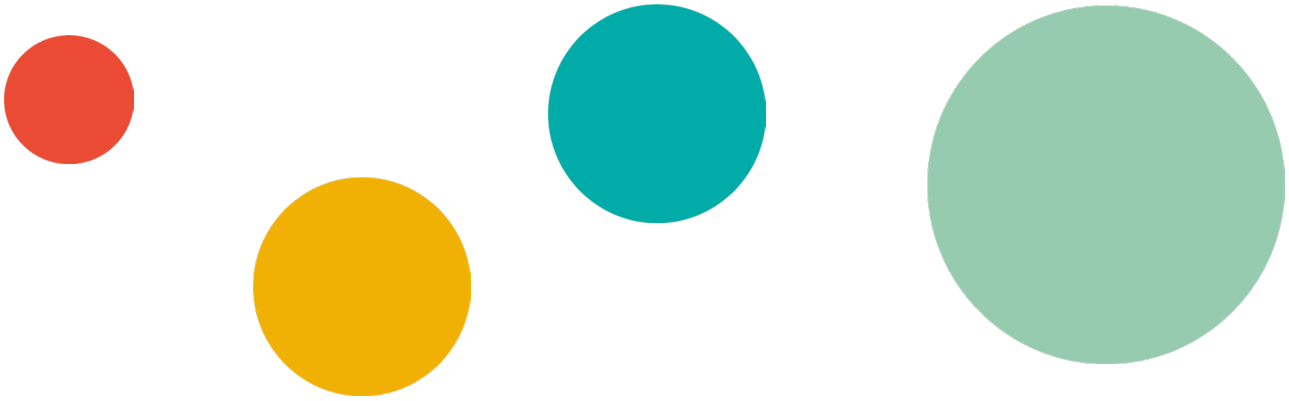
The core FinOps loop is simple: make costs visible, hold teams accountable, and continuously optimize. Cloud FinOps tooling — AWS Cost Explorer, IBM Cloudability, and similar platforms — has matured significantly over the past decade. But mainframe environments require a fundamentally different approach, because the cost structure, billing model, and optimization levers are entirely different from cloud.

Mainframe FinOps vs. Cloud FinOps

The table below illustrates the key structural differences:

Dimension	Cloud FinOps	Mainframe FinOps
Cost unit	vCPU hours, GB-months, API calls	MSU (Millions of Service Units)
Billing model	Pay-per-use, on-demand or committed	Subscription contracts (AWLC, TFP, CMP)
Key billing metric	Resource utilization, spend rate	R4HA (Rolling 4-Hour Average) or total MSU consumption
Visibility	Near-instant via cloud console	Typically day+1 or month-end via SCRT data
Primary optimization lever	Eliminate waste (rightsize, turn off unused)	Manage billing peaks, optimize LPAR defined capacity
Primary risk	Runaway spend on unused or over-provisioned resources	Billing peak overruns, contract non-compliance
Scaling model	Horizontal (add/remove instances)	Vertical (adjust LPAR defined capacity in MSUs)
Contract structure	Pay-as-you-go + optional commitments	IBM enterprise contracts with specific pricing terms
Tool ecosystem	Mature — 100+ vendors	Specialized — few purpose-built solutions
FinOps Foundation scope	Primary scope since inception	Formally included via "Data Center" and "Licensing" Scopes (2025+)
FinOps Foundation scope	Primary scope since inception	Formally included via "Data Center" and "Licensing" Scopes (2025+)

The critical distinction: cloud FinOps is primarily about eliminating waste — unused resources, unattached volumes, over-provisioned instances. Mainframe FinOps is primarily about managing billing peaks and optimizing capacity allocation within IBM's contractual pricing structures. A cloud FinOps practitioner who moves into a mainframe environment will find that their existing tools and mental models don't transfer. Mainframe FinOps requires understanding MSU consumption, LPAR defined capacity, rolling four-hour averages, and IBM-specific pricing contracts — a different body of knowledge entirely.



4- Understanding Mainframe Software Pricing: The Foundation of Mainframe FinOps

To practice mainframe FinOps, you first need to understand how mainframe software is priced and billed. This section covers IBM's pricing models, the data infrastructure that underpins billing, and the ISV software costs that complete the picture.

IBM MLC Pricing Models

IBM mainframe software pricing is divided into two categories: Monthly License Charges (MLC) and International Program License Agreement (IPLA) products. MLC is the recurring monthly charge that applies to core software such as z/OS, Db2, CICS, IMS, and IBM MQ. IPLA covers one-time charge (OTC) products plus annual subscription and support charges (ASC), typically for tools and utilities. [IBM z/OS — Planning for Sub-Capacity Pricing]

MLC billing is built on MSU (Millions of Service Units) consumption — a measure of processing capacity. But how that consumption is measured and charged depends on which IBM pricing contract your organization has. There are two main MLC pricing models available today:

AWLC — Advanced Workload License Charges

AWLC is IBM's traditional sub-capacity pricing model. Under AWLC, your software bill is calculated based on the Rolling 4-Hour Average (R4HA) — the highest average MSU consumption across any four-hour rolling window during the month. This is measured per LPAR (Logical Partition). [IBM — MLC Software Pricing; IBM Z Pricing & Licensing]

The challenge with AWLC: a single workload spike — a batch job, an end-of-month processing run — can set your R4HA for the entire month, inflating your bill even if the rest of the month runs efficiently. Managing AWLC means actively preventing those peaks.

TFP — Tailored Fit Pricing

Tailored Fit Pricing is IBM's newer enterprise pricing model, designed to deliver simplicity, transparency, and predictability. TFP offers several solutions including the Software Consumption Solution (a cloud-like, usage-based model measuring MSUs consumed), the Enterprise Capacity Solution (full-capacity licensing), and the Application Development and Test Solution (cost-competitive dev/test capacity). [IBM — Tailored Fit Pricing for IBM Z; IBM z/OS — What is Tailored Fit Pricing?]

Under TFP's consumption model, the goal shifts from peak management to total consumption control. The risk is exceeding your contracted baseline or allowing development containers to permanently expand. TFP can deliver significant predictability advantages — but only if consumption is actively managed within contractual boundaries.

IPLA — International Program License Agreement

While MLC covers core operating system and middleware software, IPLA products represent another significant mainframe software cost category. IPLA products are licensed under a one-time charge (OTC) plus ongoing annual subscription and support charges (ASC). These typically include tools for managing and monitoring Db2, CICS, IMS, and other middleware components. [IBM z/OS — Planning for Sub-Capacity Pricing]

Some IPLA products can be licensed at sub-capacity levels when the organization has contracted for sub-capacity MLC pricing. The distinction between MLC and IPLA is important for complete mainframe software cost management — a comprehensive mainframe FinOps practice must account for both.

How Billing Data Works: SCRT and SMF

Every IBM mainframe billing calculation ultimately relies on two foundational data sources: SMF (System Management Facilities) and SCRT (Sub-Capacity Reporting Tool). Understanding these is essential for practicing mainframe FinOps.

SMF is the z/OS subsystem that continuously collects system activity data. For billing purposes, the critical records are SMF type 70, subtype 1 (CPU Activity) and SMF type 89, subtypes 1 and 2 (Product Use). These records capture how much processing capacity each LPAR consumed and which software products were active. [IBM — About the Sub-Capacity Reporting Tool]

SCRT is IBM's no-charge tool that analyzes a month's worth of SMF utilization data and produces a sub-capacity report. This report shows the required license capacity (in MSUs) for each sub-capacity eligible product running on the system. SCRT reports must be submitted to IBM monthly to comply with sub-capacity pricing terms and conditions. [IBM — About the Sub-Capacity Reporting Tool; IBM SCRT PDF Documentation]

In practice, the SCRT report is the «invoice basis» — it determines what you pay. Any mainframe FinOps tool that calculates costs must work from the same SMF data that SCRT processes, applying your specific IBM contract terms to translate MSU consumption into dollar amounts.

ISV Software Costs: Beyond IBM

IBM software is not the only cost driver on a mainframe. Independent Software Vendors (ISVs) represent a significant portion of total mainframe software spending.

Understanding and managing ISV software costs alongside IBM MLC is what distinguishes a complete mainframe FinOps program from a partial one.

Every pricing model — IBM and ISV alike — has a different set of optimization levers and risk factors. Effective mainframe FinOps starts with understanding which models you're on and applying the right strategy for each model's specific billing mechanics.

5- The Three Pillars of Mainframe FinOps

A mature mainframe FinOps practice rests on three pillars: Visibility, Optimization, and Governance. These pillars align with and extend the FinOps Foundation's Inform, Optimize, and Operate lifecycle — adapted for the specific requirements of mainframe environments. Most organizations start with one pillar and build toward all three.

Pillar	What it means	Key activities
Visibility	Know where every MLC dollar comes from — by LPAR, application, business unit — in near real time, not at month-end.	Real-time cost monitoring, budget alerts, cost allocation by workload
Optimization	Actively reduce costs by dynamically managing LPAR defined capacity based on live workload demand and pricing objectives.	Automated soft capping, dynamic LPAR reallocation, billing peak management
Governance	Stay compliant with IBM contracts, track budget vs. actuals, and prevent costly overruns before they happen.	Contract compliance tracking, budget threshold alerts, cost accountability

Pillar 1: Visibility — Know Before You're Billed

The first step in any FinOps practice is making costs visible. In mainframe environments, this means moving from month-end billing reviews to near-real-time cost monitoring.
Visibility means being able to answer questions like:

- Which LPARs or applications are driving my highest MSU consumption right now?
- How much is our payment processing workload actually costing, in dollars, this month?
- Are we on track to stay within our contractual baseline, or are we trending toward an overrun?
- What's driving the cost increase I'm seeing compared to last month?

Without visibility, optimization is guesswork. Teams that achieve genuine cost transparency — down to the application and business unit level — consistently make better capacity decisions and catch potential overruns before they become billing events.

Pillar 2: Optimization — Reduce Costs Without Sacrificing Performance

Visibility tells you where costs come from. Optimization actively reduces them. In mainframe FinOps, the primary optimization lever is defined capacity management — dynamically adjusting how much processing capacity is allocated to each LPAR based on live workload demand. Traditional soft capping is static: a fixed ceiling per LPAR that limits consumption but doesn't adapt to actual workload patterns. This creates a false choice between cost control and performance: either cap LPARs and risk performance degradation, or leave them uncapped and accept uncontrolled billing.

Dynamic capacity optimization resolves this tradeoff. By continuously redistributing capacity across LPARs based on real-time demand, WLM service class priorities, and billing objectives, it's possible to absorb workload spikes on critical LPARs while staying within overall cost targets — without manual intervention.

The Role of z/OS Workload Manager (WLM)

z/OS Workload Manager (WLM) is a core component of the z/OS operating system that automatically manages system resources — CPU, memory, and I/O — based on business-defined performance goals rather than technical metrics. WLM assigns work to service classes, each with its own performance goal and business importance level. (IBM — z/OS Workload Manager)

WLM is central to mainframe FinOps because it determines how system resources are allocated across workloads — and therefore what gets billed. Dynamic capacity optimization tools work in coordination with WLM service class priorities: they understand which workloads are most business-critical and allocate capacity accordingly, ensuring that optimization never comes at the expense of service quality for high-priority work.

Enterprises implementing dynamic LPAR capacity optimization typically achieve 5–18% reduction in Monthly License Charges, with no impact on service quality or SLA performance. Performance improvements of 5–15% are also common, as workloads receive the resources they need precisely when they need them.

Pillar 3: Governance — Stay Compliant and Budget-Accountable

Governance is what turns mainframe FinOps from a technical practice into an organizational one. It means having clear ownership of mainframe costs, contractual compliance tracking, and budget accountability across teams. In practice, governance includes:

- Tracking actual MSU consumption against IBM contractual commitments in real time
- Allocating mainframe costs to specific business units, applications, or services — so that cost responsibility is shared, not siloed in IT
- Defining budget thresholds and receiving alerts before overruns occur
- Maintaining auditability: knowing why costs changed and being able to explain it to finance leadership

6- Key Roles in a Mainframe FinOps Practice

Mainframe FinOps is inherently cross-functional. It requires collaboration between technical teams who manage the platform and business teams who are accountable for the costs. This aligns directly with the FinOps Foundation's first principle: «Teams need to collaborate.» (FinOps Principles)

The key roles in a mainframe FinOps practice typically include:

- **Mainframe Systems Programmer** — Manages z/OS configuration, LPAR settings, and defined capacity parameters. Implements technical optimization changes.
- **Capacity Planner** — Forecasts future workload growth, plans hardware and software capacity, and models the cost impact of changes.
- **z/OS Performance Analyst** — Monitors system performance, WLM service class goals, and workload behavior. Identifies optimization opportunities.
- **IT Finance / Cost Manager** — Owns the mainframe budget, tracks spending against forecasts, and reports to finance leadership. Translates MSU consumption into business costs.
- **Application Owners / Business Unit Leaders** — Accountable for the cost of the workloads they run. Need visibility into what their applications cost and how their decisions affect the mainframe bill.
- **FinOps Practitioner** — Orchestrates the FinOps practice across all stakeholders. Drives adoption of FinOps processes, tools, and culture. Ensures visibility data reaches the right people and optimization actions are taken.

In many organizations, the biggest barrier to mainframe FinOps maturity is not technology — it's organizational silos. Mainframe operations and IT finance often operate independently, with limited shared visibility into costs and consumption. Breaking down these silos is the first and most important governance action.

7. How Enterprises Implement Mainframe FinOps

Most mainframe FinOps programs follow a similar maturity progression, regardless of organization size or IBM pricing model:

Stage 1: Reactive — Managing Costs After the Fact

At this stage, mainframe costs are reviewed monthly when the IBM invoice arrives. Teams lack real-time visibility and respond to billing surprises after they've already occurred. Cost allocation is limited or non-existent, and optimization is handled manually and infrequently.
This is where most organizations start — and where significant budget is silently lost.

Stage 2: Aware — Gaining Visibility

The first step toward mainframe FinOps maturity is achieving near-real-time cost visibility. Teams at this stage can see MSU consumption trends, monitor R4HA or container usage, and receive alerts when consumption approaches contract thresholds.
Cost data starts to be shared with finance and business unit leaders, creating shared accountability for mainframe spending. Billing surprises become less frequent.

Stage 3: Active — Optimizing Continuously

At this stage, capacity management is automated rather than manual. LPAR defined capacity is adjusted dynamically in response to workload patterns, pricing model constraints, and WLM service class priorities. Teams have moved from managing costs monthly to managing them at each billing interval.
This is where the measurable savings appear: 5–18% MLC reduction is typical for organizations that move from reactive capacity management to active automated optimization.

Stage 4: Strategic — Mainframe FinOps as a Competitive Advantage

At full maturity, mainframe FinOps informs strategic decisions: contract negotiations with IBM, mainframe modernization roadmaps, application refactoring prioritization, and capacity forecasting for future workload growth.

Organizations at this stage can answer not just «what are our mainframe costs today?» but «what will they be in 12 months if we make this architectural change?» — turning mainframe finance from a cost center into a strategic input. This kind of cost intelligence is also invaluable for migration planning: knowing exactly what each workload costs on mainframe is the foundation for any credible business case for moving it to cloud.

8. What Good Results Look Like

Mainframe FinOps outcomes vary based on starting point, IBM pricing model, and the complexity of the environment. But across enterprise deployments, a clear pattern of measurable results emerges:

Cost Reduction

- 5–18% reduction in Monthly License Charges — without service quality impact
- Up to 20% MLC reduction in CMP environments transitioning to TFP
- Elimination of billing peak overruns through proactive capacity management
- Reduced over-provisioning as teams gain confidence in dynamic capacity management

A note on these figures: the 5–18% MLC reduction range reflects results reported across Zetaly's customer deployments. As with any vendor-reported outcome, actual results will vary by environment. Organizations considering mainframe FinOps tooling should request customer references and, where possible, a proof-of-concept against their own SMF data before committing to a platform.

Performance Improvement

- 5–15% performance improvement as workloads receive precisely the capacity they need
- Elimination of unnecessary capping on critical production LPARs
- Faster response to unexpected load spikes through automated capacity reallocation

Operational Improvement

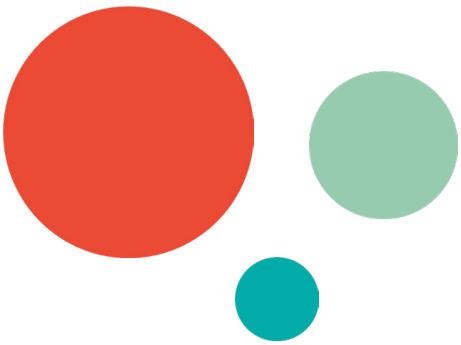
- Near-real-time cost visibility replacing month-end billing reviews
- Clear cost allocation by business unit, application, or service
- Proactive budget alerts replacing reactive billing surprises
- Reduced manual capacity management effort — freeing capacity engineers for higher-value work

In Practice: Orange

«With Zetaly Automated Capacity, Orange has found a game-changing solution. We can now navigate the price increases of our software portfolio for the next two years without any impact on performance. Not only that, but our mainframe activity runs more smoothly than ever before and we have become more responsive, ensuring optimal operations and improved customer experiences.»
Stéphane Rousset,
Mainframe Manager, Orange

9. Tools for Mainframe FinOps

Mainframe FinOps requires purpose-built tooling. Generic IT financial management platforms — built for cloud or on-premise x86 environments — don't understand MSU billing, IBM pricing contracts, or LPAR capacity management. Effective mainframe FinOps tools need to be native to the IBM z/OS environment.
A complete mainframe FinOps toolset addresses all three pillars: visibility, optimization, and governance.



For Visibility: Zetaly Data Platform (ZDP) and Zetaly Service Intelligence (ZSI)

Zetaly Data Platform (ZDP) is the foundation of mainframe observability. It uses collectors and connectors to continuously gather operational data from across the mainframe environment, then enriches that data with metadata, basic calculations, cross-source correlations, and business-unit filters so it becomes accessible and actionable. No mainframe expertise is required to consume the output — ZDP handles the translation. ZDP runs on Linux (zLinux, zCX, or cloud), which keeps its own footprint off z/OS and supports hybrid mainframe-and-cloud environments. ZDP applies to both AWLC and TFP customers: under AWLC, it provides the workload-level cost data needed to inform

optimization and capacity decisions; under TFP, it delivers the same visibility without inflating the customer's own SCRT baseline.
Zetaly Service Intelligence (ZSI) sits on top of ZDP as the analytics and Application Performance Monitoring (APM) layer. ZSI translates raw mainframe operational data into business-application views — showing what each application actually consumes in MSUs, which workloads drive cost, and where inefficiencies sit. This is what shifts the FinOps conversation from «which transaction ran» to «what does this product or business unit cost to run,» and it lets technical, finance, and business stakeholders all read the same data. ZSI is the layer that surfaces budget trends and the cost impact of business activity, so teams can act before consumption becomes a billing event.

For Optimization: Zetaly Automated Capacity (ZAC)

ZAC is the only mainframe soft capping solution that dynamically adjusts LPAR defined capacity in real time — distributing resources across LPARs based on live workload demand, WLM service class priorities, and billing objectives. Rather than applying static capacity ceilings, ZAC continuously rebalances capacity so that critical workloads always have what they need while overall MSU consumption stays within cost targets.

ZAC deployment takes one week and delivers measurable cost and performance improvements from day one, without requiring changes to application code or z/OS configuration.

What to Look for in a Mainframe FinOps Tool

When evaluating mainframe FinOps tooling, the following capabilities are most critical:

- Native z/OS integration — The tool should operate directly within your mainframe environment, not as a layer of abstraction requiring data export.
- SCRT data fidelity — Cost calculations must align with what IBM's SCRT tool produces. Any divergence creates reconciliation problems.
- Contract-aware pricing — The tool should apply your specific IBM pricing contract terms (AWLC, TFP, CMP), not generic billing models.
- Real-time visibility — Month-end reporting is not enough. Meaningful cost management requires near-real-time data.
- Dynamic capacity management — Static soft capping is a blunt instrument. Dynamic optimization that responds to live workload demand is the basis of measurable savings.
- Multi-LPAR and multi-system support — For organizations with complex environments, the tool must operate across all LPARs and CPCs in scope.
- Business unit cost allocation — The ability to allocate costs to applications and business units is what makes FinOps organizational rather than purely technical.

10. Mainframe FinOps During Modernization and Cloud Migration

A common question from IT and finance leaders: «If we're planning to migrate some or all of our mainframe workloads to the cloud, does mainframe FinOps still make sense?»
The short answer is yes — and often more urgently than for organizations with no migration plans at all.

The Reality of Mainframe Migration Timelines

Mainframe-to-cloud migrations are among the most complex IT undertakings an enterprise can attempt. Organizations that have successfully completed them — including Meliá Hotels (60% compute cost savings after migrating a COBOL-based reservation system to AWS) (Meliá Hotels AWS Case Study) and Commonwealth Bank of Australia (which migrated its full core banking platform to AWS in 2025, supporting 16 million customers and 40% of Australian payment flows) (Commonwealth Bank) — describe multi-year programs requiring extensive planning, specialist expertise, and close regulatory coordination.

Commonwealth Bank's core banking migration took 18 months with 54 testing crews and a partner ecosystem that included SAP, Accenture, AWS, and Red Hat. CBA's technology lead described it as requiring «extensive modelling upfront on what this would look like» — including FinOps modelling for the cloud workload costs. (Commonwealth Bank)

TSB Bank's 2018 migration attempt — moving 5.2 million customer accounts from a Lloyds Banking Group platform to a new system — offers a cautionary counterpoint. Despite three years of planning and 85 specialized subcontractors, the migration failed on go-live day, locking customers out of accounts for weeks. The result: £62 million in regulatory fines [TBC: FCA Final Notice 2022 records the combined FCA+PRA fine as £48.65m — please reconcile], over £32 million in customer compensation, and a CEO resignation. The UK's Financial Conduct Authority found that TSB's timeline had been set «based on very little information» before requirements were fully understood. (FCA Final Notice — TSB Bank, 2022)
The lesson is not that migration is impossible — it clearly is not, as Meliá and Commonwealth Bank demonstrate. The lesson is that it takes longer, costs more, and requires more preparation than most organizations initially plan for. During that preparation and execution period — which commonly spans three to seven years for complex mainframe environments — the mainframe costs keep running.

Why FinOps Matters Most During Migration

Organizations in the middle of a mainframe modernization journey frequently have their highest software costs, not their lowest. They are running parallel workloads: existing mainframe production systems continue operating while cloud alternatives are built, tested, and validated in parallel. That dual-running period can last months or years.

Mainframe FinOps during this period serves three specific purposes:
1. Fund the migration. Every dollar saved through mainframe cost optimization is a dollar available to invest in cloud build-out.
Organizations that implement mainframe FinOps before or during migration consistently find that the savings help fund the transition itself.

2. Inform migration decisions. Cost visibility at the workload and application level — which ZCC provides — is exactly the data needed to answer «which applications are worth migrating?» A workload that costs $50,000 per month on mainframe is a very different migration priority than one that costs $5,000. Without that visibility, migration sequencing is based on guesswork rather than economics.

3. Avoid paying for workloads twice. Without active cost management, organizations can end up paying full mainframe costs for systems that have partially migrated, where residual dependencies keep the mainframe running at full capacity even though half the workload has moved. FinOps governance catches these situations before they become expensive surprises.

When Migration Makes Sense — and When It Doesn't

Not every mainframe workload is a good cloud migration candidate, and intellectual honesty about this is important:

- **Strong candidates for migration** include workloads with well-documented business logic, high peak-to-average compute ratios (where cloud elasticity delivers real cost benefits), applications undergoing significant redesign, and new products where cloud-native development is faster and cheaper.
- **Workloads that often stay on mainframe** include high-throughput transaction processing systems where mainframe throughput/cost economics are genuinely competitive, systems with deeply entrenched COBOL or Assembler code where refactoring risk is high, and regulated environments where mainframe's security and auditability characteristics are difficult to replicate.

Mainframe FinOps is not an argument against migration. It is an argument for managing costs responsibly during the years it takes to complete one — and for ensuring that migration decisions are based on accurate cost data rather than incomplete estimates.

To learn more, download our white paper

Mainframe to Cloud Migration:

Pros, Cons, and What Actually Happens

at Zetaly.io/mainframemigration

11. Frequently Asked Questions

What is mainframe FinOps?

Mainframe FinOps is the application of financial operations (FinOps) practices to IBM z/OS mainframe environments. It combines real-time cost visibility, automated capacity optimization, and budget governance to bring financial accountability to Monthly License Charge (MLC) spending — one of enterprise IT's largest and most complex cost categories.

How is mainframe FinOps different from cloud FinOps?

Cloud FinOps focuses on variable compute costs on public cloud platforms, where resources can be turned off immediately and costs are visible in near real time. Mainframe FinOps addresses IBM-specific pricing contracts (AWLC, TFP, CMP), rolling-average billing mechanics, and LPAR defined capacity management — a fundamentally different cost model that requires specialized tools and expertise. Cloud FinOps tools do not apply to mainframe environments.

What are Monthly License Charges (MLC) on IBM mainframe?

Monthly License Charges (MLC) are IBM's recurring fees for mainframe software, calculated based on MSU (Millions of Service Units) consumption. The specific calculation method depends on the pricing contract: AWLC uses the Rolling 4-Hour Average (R4HA) per LPAR; TFP uses total MSU consumption against a contracted baseline; CMP aggregates consumption across multiple data centers. MLC costs can represent 30–50% of an enterprise's total mainframe budget.

What is an MSU?

An MSU (Millions of Service Units) is IBM's standard unit of measurement for mainframe processing capacity. MSUs represent the amount of useful work a processor can perform. The MSU rating of a mainframe system determines the maximum processing capacity available, and actual MSU consumption is the basis for most mainframe software billing calculations.

What is an LPAR and why does it matter for FinOps?

An LPAR (Logical Partition) is a virtualized instance of an IBM mainframe, each with its own defined capacity allocation measured in MSUs. Mainframe billing is typically calculated at the LPAR level, which means how capacity is distributed across LPARs directly determines your software costs. Effective mainframe FinOps requires active management of LPAR defined capacity to prevent unnecessary billing peaks while maintaining performance for critical workloads.

What is the difference between AWLC, TFP, and CMP?

AWLC (Advanced Workload License Charges) bills based on the highest rolling 4-hour average MSU consumption per LPAR during the month — making peak prevention the primary optimization goal. TFP (Tailored Fit Pricing) offers consumption-based or fixed-capacity pricing against a contracted MSU baseline, with separate pricing solutions for different workload types. CMP (Country Multiplex Pricing) consolidates multiple data centers into a single billing unit, allowing workload sharing across sites. Each model requires a different optimization strategy.

What is the difference between MLC and IPLA?

MLC (Monthly License Charges) are recurring monthly fees for core mainframe software (z/OS, Db2, CICS, IMS, IBM MQ). IPLA (International Program License Agreement) covers products with a one-time charge plus annual subscription and support — typically tools and utilities for managing mainframe middleware. Both contribute to total mainframe software costs, and a complete FinOps practice accounts for both.

What is SCRT and why does it matter?

SCRT (Sub-Capacity Reporting Tool) is IBM's no-charge tool that processes a month's worth of SMF utilization data and produces the sub-capacity report that determines your MLC charges. SCRT reports must be submitted to IBM monthly. The SCRT report is effectively the invoice basis for sub-capacity pricing — understanding what data feeds into SCRT and how it calculates billing is foundational to mainframe FinOps.

What is soft capping on mainframe?

Soft capping is the practice of limiting an LPAR's defined capacity (measured in MSUs) to control mainframe software billing. Traditional soft capping sets a fixed ceiling per LPAR. Dynamic soft capping continuously adjusts these ceilings across all LPARs based on live workload demand, allowing the system to absorb load spikes on critical LPARs while staying within overall billing targets. Dynamic soft capping eliminates the performance vs. cost tradeoff of static approaches.

How much can mainframe FinOps reduce software costs?

Results vary depending on starting environment, IBM pricing model, and the maturity of existing capacity management practices. Across 30+ enterprise deployments, organizations typically achieve 5–18% reduction in Monthly License Charges through dynamic LPAR capacity optimization, with no impact on service quality. Organizations transitioning from CMP to TFP often see reductions of up to 20%. Performance improvements of 5–15% are also common as workloads receive capacity precisely when needed. These figures are drawn from Zetaly customer data — prospective customers should validate estimates against their own environment data before committing.

How long does it take to implement mainframe FinOps tooling?

Modern mainframe FinOps tools are designed for fast deployment. Optimization tools like ZAC deploy in one week and deliver measurable cost and performance improvements from day one. Visibility tools like ZCC typically provide cost data within the same billing period. Full organizational maturity — including cost allocation by business unit and proactive governance — develops over one to three months as teams build processes around the new data.

Do I need both visibility and optimization tools for mainframe FinOps?

Most organizations benefit from both. Visibility tools tell you where costs come from and alert you to budget risks. Optimization tools actively reduce costs by dynamically managing LPAR capacity. You can start with either — many teams begin with visibility to establish a baseline understanding of their costs, then add optimization once they know where the savings opportunities are. However, the greatest impact comes from the combined approach: see the full cost picture, then automatically optimize it.

Is mainframe FinOps relevant if we are modernizing or moving to cloud?

Yes — and for most organizations, more urgently so. Mainframe-to-cloud migrations take years to complete, and during that period the mainframe costs continue to run — often at their highest levels, as parallel environments are maintained. Mainframe FinOps serves three functions during modernization: it funds the migration by generating savings from the existing environment, it informs migration sequencing by providing accurate per-workload cost data, and it prevents «dual-paying» by catching residual mainframe costs after workloads have nominally migrated.

12. References and Further Reading

FinOps Foundation

- What is FinOps?
- FinOps Framework (Principles, Phases, Domains)
- FinOps Principles
- FinOps Scopes
- FinOps Framework 2025 Update (Cloud+ scope including data center & licensing)
- FinOps Framework 2026 Update (Executive Strategy, Technology Categories, Converging Disciplines)

IBM — Mainframe Pricing & Documentation

- IBM Z Pricing & Licensing (official overview)
- IBM — Tailored Fit Pricing for IBM Z (content solutions hub)
- IBM z/OS — What is Tailored Fit Pricing?
- IBM z/OS — Monthly License Charge (MLC) Software Pricing
- IBM z/OS — Planning for Sub-Capacity Pricing (PDF, covers AWLC, R4HA, CMP)
- IBM — About the Sub-Capacity Reporting Tool (SCRT)
- IBM — Using the Sub-Capacity Reporting Tool (v30.1.0, PDF)
- IBM — Are You Using the Right 4-Hour Rolling Average? (PDF)
- IBM — What is FinOps?
- IBM — z/OS Workload Manager

Industry Research & Analysis

- Mordor Intelligence — Mainframe Market Size, Growth & Outlook (2026)
- Forrester — The State of Mainframe, Global, 2025
- BMC — AMI Cost Management

Migration Case Studies Referenced

- Commonwealth Bank — Core Banking Cloud Migration Case Study (March 2026)
- Meliá Hotels — Mainframe Migration to AWS Case Study
- FutureCIO — Lessons from Failed Mainframe Migration (TSB Bank)

Open Mainframe Project (Linux Foundation)

- Open Mainframe Project — About
- Open Mainframe Project — Open Source on the Mainframe

Zetaly Resources

- How to Optimize Your TFP Environment for Maximum Efficiency
- Maximizing Savings: Transitioning from CMP to TFP
- Transitioning from AWLC to TFP: A Guide for Mainframe Optimization
- Free ZAC Simulation — Discover Your Mainframe Savings Potential

Start Your Mainframe FinOps Journey

Whether you're looking to understand your mainframe costs for the first time, or ready to actively reduce your MLC spend, Zetaly's FinOps platform covers both needs.

Explore ZAC for Automated LPAR Capacity Optimization and MLC reduction

Get a free ZAC Simulation
a personalized report showing your potential savings based on your actual environment data. No commitment required

Explore ZSI for application-level cost insightnal data without z/OS overhead

Explore ZSI to see what each application costs

ZETALY®

CONTACT US

zetaly.io

contact@zetaly.io

Zetaly/In

+33 (0)2 22 06 88 60

Document 10010 RevA 20260525