



OPEN
MAINFRAME
PROJECT

Modernization Working Group

Defining Mainframe Modernization

SEP 2026

Executive Summary

Mainframes continue to power mission-critical applications across global enterprise industries. However, industry definitions of what "modernisation" actually entails remain inconsistent and frequently misleading.

This whitepaper provides a point of view on what is mainframe modernisation through a practical three-pillar framework, evaluating the **hardware platform**, **application architectures**, and **DevOps tooling**. Furthermore, it compares modernising pathways across IBM and non-IBM vendor ecosystems, establishing essential considerations for operational resilience, agility, and long-term platform viability.

Defining "Modernisation"

Mainframes serve as the digital backbone of critical global infrastructure, powering core banking, insurance, government, retail, and aviation operations. Although frequently labeled as "legacy," this perception typically reflects static development practices and operational workflows rather than physical platform limitations. This distinction is especially important when comparing IBM Z mainframe environments, which continue to advance through regular hardware innovation, against other mainframe vendor platforms whose roadmaps have slowed or reached end-of-life.

Mainframe modernisation is the process of evolving existing applications, software development practices, and hardware infrastructure. Rather than a binary decision, modernisation exists along a spectrum tailored to organisational goals, risk tolerance, and platform roadmaps. For some enterprises, it involves enhancing capabilities via APIs and modern integration patterns; for others, it means modernising developer workflows, automating CI/CD pipelines, or adopting contemporary platform tooling.

Crucially, modernisation does not imply tearing down core architectures, rewriting entire software estates into cloud-native languages, or abandoning proven technologies like COBOL or JCL. The advent of AI code assistants further simplifies legacy code maintenance and mitigates skill gaps and workforce retirement risks. The ultimate objective of modernisation is establishing an executable roadmap that aligns platform capability with long-term enterprise technology architecture.

The Three Dimensions of Mainframe Modernisation

Mainframe modernisation can be analysed across three distinct yet interconnected dimensions: Hardware Infrastructure, Applications, and DevOps Tooling. Each dimension addresses a distinct aspect of the mainframe ecosystem, from the underlying infrastructure to the software delivery lifecycle and the business-critical applications that run on the platform. Together, these dimensions form a comprehensive framework for evaluating technical maturity and operational efficiency.

THREE DIMENSIONS OF MAINFRAME MODERNISATION

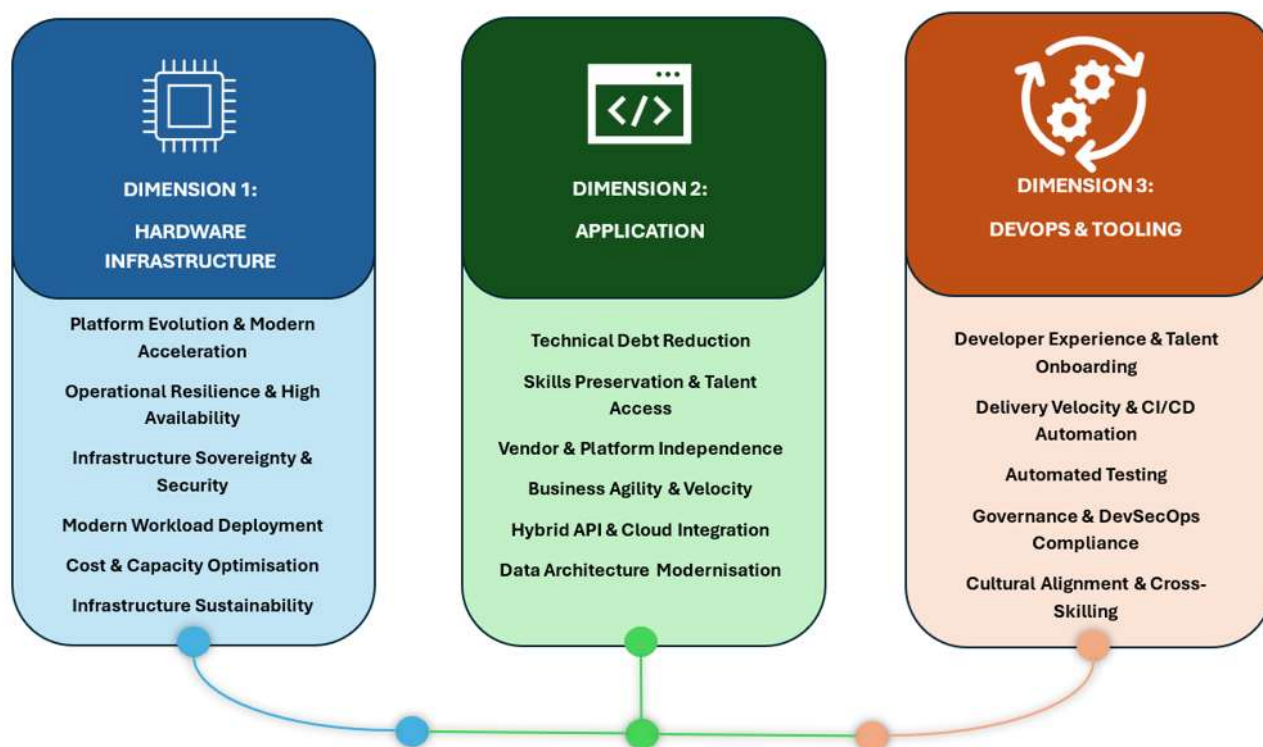


Figure 1 - The 3 Dimensions of Mainframe Modernisation and the potential opportunities

Dimension 1: Modernising Mainframe Hardware

Infrastructure modernisation focuses on hardware platform evolution, bringing in new capabilities such as Neural Processing Units (NPUs) for AI, virtualisation, or quantum-safe cryptography. Machine modernisation offers a highly predictable execution path governed by vendor product roadmaps, carrying significantly lower operational risk than application or DevOps transformations.

Upgrading the hardware platform becomes the optimal strategic choice in several critical enterprise scenarios:

- **Sovereignty & Infrastructure Control:** Highly regulated industries require strict physical data residency, operational sovereignty, and direct control over critical infrastructure.
- **Legacy Continuity with Modern Capabilities:** Organisations that need to sustain decades-old core applications while keeping the underlying platform fully supported and benefiting from modern hardware features [17].
- **New Workload Use Cases:** Enterprises looking to deploy modern workloads (e.g. real-time AI inferencing, containers, cloud-native services) alongside core transactional data and legacy applications to minimize latency [15].
- **Increased Volumes & Sustainability Targets:** Expanding processing capacity alongside improved performance-per-watt metrics, while reducing cooling demands and lowering carbon footprints [18].
- **Advanced Threat Protection:** Mitigate emerging security risks and cyber threats by deploying new hardware capabilities (e.g. confidential computing modules, pervasive encryption, and post-quantum cryptographic protection) [16].
- **Cost & Licensing Optimization:** Upgrading engines allows organisations to offload specialized tasks to dedicated processors (e.g., zIIPs, AI accelerators), significantly reducing general CPU utilization and recurring software licensing cost.

Dimension 2: Modernising Mainframe Applications

Application modernisation addresses business logic, data structures, and institutional knowledge. It represents an enterprise's highest operational risk, but may yield the greatest return in agility and innovation. Organisations should evaluate application strategies across three sequential steps.

Step 1: Identifying Application modernisation Triggers

- **Unmanageable Technical Debt:** Complex, monolithic architectures where targeted fixes are no longer viable due to performance bottlenecks, scale limits, or security risks [3].
- **Skills Depletion:** A shrinking talent pool for legacy languages where AI code assistants are either unavailable or fail to achieve required enterprise quality standards [1].
- **Spiralling Total Cost of Ownership (TCO):** Underlying compute or software licensing costs that outpace the strategic business value delivered by the application [2].
- **Vendor Risk Mitigation:** Regulatory mandates or corporate strategies to adopt open-source languages and portable frameworks, reducing single-vendor lock-in [2].

Step 2: Selecting the Application Strategy

Deciding how to modernize must be evaluated independently from where the code will run, ensuring target platforms serve as enablers rather than constraints:

- **Sustained Evolution (Modernize-in-Place):** Delivering incremental code improvements and targeted refactoring, often accelerated by AI code assistants, while keeping the core language and runtime architecture stable [5].
- **Greenfield Rebuild:** Completely rewriting applications in modern languages or frameworks when legacy technical debt prevents incremental evolution [4].
- **Commercial Replacement (SaaS/COTS):** Replacing custom, in-house code with enterprise software packages or open-source solutions when bespoke code offers no competitive advantage [4].

Step 3: Determining the Target Execution Environment

- **On-Mainframe:** Retaining workloads on existing mainframe infrastructure when tight data coupling, low latency, high transaction volume, or strict operational control are essential
- **Private Cloud:** Shifting applications to enterprise-managed private cloud environments when sovereignty is required, but the workload no longer benefits from mainframe hardware [4].
- **Public Cloud:** Offloading infrastructure management entirely to public cloud providers, enabling the enterprise to focus strictly on software feature delivery [4].

Dimension 3: Modernising DevOps Processes & Tooling

DevOps modernisation transforms how mainframe software is built, tested, deployed, and operated. Bringing modern DevOps to legacy applications frequently removes the need to embark on high-risk application replacement or migration initiatives. A 3-steps framework can be followed to evaluate and execute DevOps modernisation.

Step 1: Identifying DevOps Friction Points

- **Environment & Infrastructure Scheduling Constraints:** Lack of on-demand access to mainframe LPARs or dev/test environments, creating scheduling bottlenecks for engineering teams [9].
- **Manual Bottlenecks & Tooling Silos:** Outdated, manual build-and-deploy processes and isolated SCM tools that limit delivery speed, auditability, and cross-platform visibility [10].
- **Testing Delays:** Complex test data management, difficulties replicating upstream/downstream integration dependencies, and manual validation of batch/JCL execution
- **Developer Friction & Talent Barriers:** Steep learning curves associated with traditional green-screen interfaces, legacy programming languages and application architecture patterns, which slow early-career onboarding and reduce daily velocity [8].

Step 2: Deploying Foundational DevOps Capabilities

Once friction points are identified, organisations should assess implementing foundational operational capabilities, which can include:

- **Containerised Sandboxes & Off-Platform Prototyping:** Utilizing open-systems sandboxes, Linux-based environments, and containerized setups to build and test code off-platform before deploying back to the mainframe. This reduces LPAR resource contention and lowers compute costs [9].
- **Automated CI/CD & Pipeline Integration:** Replacing proprietary SCMs with Git-based version control, CLIs, and modern IDE integrations to standardize access and auditability across enterprise platforms [7].
- **Automated & Shift-Left Testing:** Implementing automated regression suites, service virtualization, and AI-assisted test generation to catch defects early in the lifecycle.
- **Developer Experience (DevX) Enhancement:** Modernising the developer workspace via modern IDEs, debugging suites, and AI code assistants capable of explaining legacy code structures and generating documentation [8].
- **Embedded DevSecOps & Audit Automation:** Integrating automated security scanning , vulnerability management, and automated compliance tracking directly into CI/CD pipelines [8].

Step 3: Managing Operational & Cultural Alignment

Finally, organisations must address the operational and cultural trade-offs required to sustain long-term DevOps adoption:

- **Parity & Behaviour Management:** Ensuring code built and tested in off-platform sandboxes behaves consistently when deployed to production mainframe runtimes [10].
- **Coexistence & Hybrid Transitions:** Managing operational overlap where legacy SCMs co-exist alongside Git pipelines during multi-year enterprise migrations.
- **Cultural Cross-Skilling:** Unifying mainframe and distributed engineering cultures through cross-training and shared iterative delivery practices.

Vendor Ecosystem Strategies

Mainframe vendors generally fall into three strategic camps: those continuously evolving their hardware to keep the ecosystem alive, those decoupling operating environments from proprietary hardware to preserve existing application logic, and those sunsetting physical hardware entirely to push organisations toward modern cloud or AI-driven architectures.

The table below summarises how these ecosystem patterns align with key modernisation considerations across platform direction, application strategy, and long-term sustainability.

Ecosystem Pattern	Evolutive Ecosystem	Hybrid-Decoupled Ecosystem	Transformational Ecosystem
Ecosystem Direction & Drivers	Continuous hardware evolution with long-term vendor investment (e.g., on-chip AI/NPUs, cryptographic acceleration, pervasive encryption) [15].	Decoupling legacy Operating System environments from proprietary hardware to run on private or public cloud platforms [22][23].	Physical hardware retirement driving architectural transformation (e.g., complete shift toward quantum computing and AI-driven platforms) [19][20][21][26].
Modernisation Approach	Modernise-in-place with incremental extension (e.g., hybrid-cloud API integration or containerised microservices) [24].	Re-platforming to cloud infrastructure (e.g., preserving core business logic while shifting underlying host environments to Azure or AWS) [22][23][25].	Complete application migration or replacement (e.g., automated COBOL to Java refactoring, greenfield rebuilds, COTS replacement) [20][21].
Long-Term Platform Sustainability	High platform stability with minimal migration disruption [11][15].	Software lifecycle extension without proprietary hardware lock-in. (e.g., operational continuity on standard cloud infrastructure) [22][23].	High initial transformation effort to mitigate hardware end-of-life (e.g., total elimination of mainframe footprint by 2035) [19][26].
Example Platform	IBM Z	Unisys ClearPath	Fujitsu GS21

Table 1 - Mainframe Vendor Ecosystem Patterns

Understanding these ecosystem patterns enables organisations to select modernisation pathways that align with long-term platform viability, operational risk tolerance, and strategic technology direction. By evaluating vendor roadmaps, modernisation tooling, and sustainability factors, enterprises can make informed decisions that balance stability with transformation across diverse mainframe estates.

Organisational Enablers & Change Management

Organisational factors ultimately dictate whether mainframe modernisation succeeds or stalls at enterprise scale. Regardless of whether an enterprise chooses to modernise on-platform or

migrate off it, technical transformation cannot deliver value without addressing systemic organisational constraints and executing deliberate operational changes.

ORGANISATIONAL ENABLERS & CHANGE MANAGEMENT

THE TRANSFORMATION FUNNEL: FROM CONSTRAINTS TO VALUE

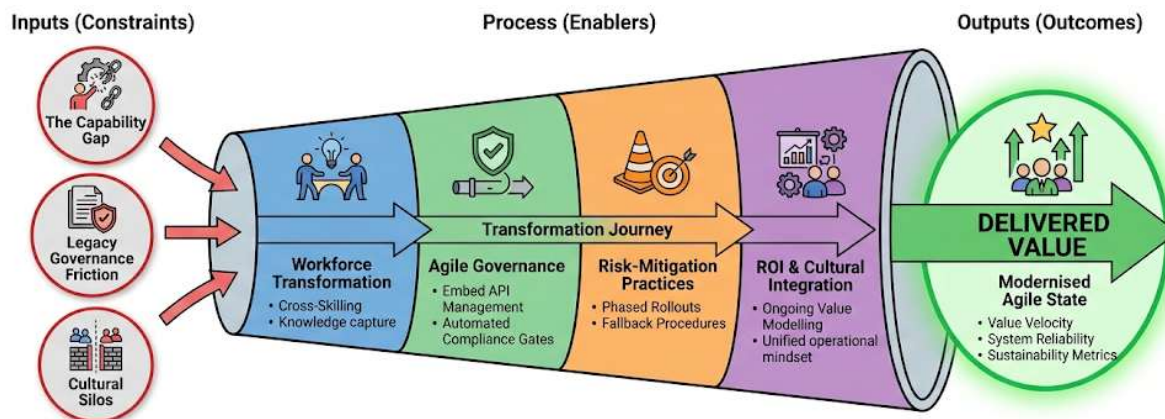


Figure 2 - The Transformational Funnel: From Constraints to Value

Primary Organisational Constraints

Modernisation initiatives routinely face three non-technical bottlenecks that impede progress:

- **The Capability Gap:** Retiring mainframe specialists alongside a scarcity of multi-skilled engineers creates institutional risk and restricts delivery velocity.
- **Legacy Governance Friction:** Outdated change-approval processes and rigid risk-management models fail to support modern APIs, automated CI/CD pipelines, and hybrid deployments [12].
- **Cultural Silos:** Deep structural divides between legacy mainframe teams and distributed engineering groups foster operational resistance, manual handoffs, and siloed mindsets [13].

Essential Organisational Changes

To remove these bottlenecks and sustain long-term transformation, enterprises must execute four core organisational shifts:

- **Workforce Transformation & Knowledge Transfer:** Replace single-skill reliance with structured cross-skilling, mentoring, and continuous knowledge-capture programs [11]. Training legacy developers on modern DevSecOps tools while upskilling cloud engineers on core mainframe concepts builds a resilient, hybrid workforce [13].
- **Agile Governance & Automated Compliance:** Modernise control mechanisms by embedding API management, automated audit logging, and static security scanning directly into pipelines. Replacing manual change boards with automated policy gates ensures compliance without sacrificing velocity [12].

- **Pragmatic Risk-Mitigation Practices:** De-risk implementation through phased rollouts, service virtualization, and parallel execution testing. Establishing clear fallback procedures protects operational stability and data integrity during structural shifts.
- **Continuous ROI Alignment & Cultural Integration:** Maintain executive sponsorship through ongoing ROI modeling that tracks cost efficiency, deployment velocity, system reliability, and sustainability metrics [13]. Aligning incentives across engineering teams transforms culture from episodic project delivery into continuous operational improvement.

Conclusion

Mainframe modernisation is an evolutionary initiative. Rather than forcing a binary choice between retaining or retiring legacy infrastructure, modernisation evaluates how the mainframe best aligns with broader enterprise technology strategy and roadmaps.

The framework presented in this paper outlines how to evaluate mainframe modernisation across three dimensions:

- **Hardware modernisation:** Enhances performance, security, and sustainability where active vendor roadmaps permit.
- **Application modernisation:** Addresses business logic and data structures along a spectrum from in-place evolution to greenfield rebuilds or commercial replacement.
- **DevOps & Tooling modernisation:** Transforms delivery pipelines to improve velocity and developer experience, frequently without requiring immediate, high-risk code rewrites

While the historical mainframe landscape was highly diverse, market consolidation over recent decades has led most vendors to retire their proprietary platforms. IBM Z remains the primary platform with an active hardware roadmap and an expanding partner ecosystem. Consequently, modernisation trajectories across the broader market fall into three distinct ecosystem patterns:

- **Evolutionary:** Continuous hardware and software innovation.
- **Hybrid-Decoupled:** Software portability across commodity or cloud infrastructure.
- **Transformational:** Complete architectural shifts away from legacy hardware.

Taking a structured approach to assessing modernisation and understanding trajectories allows organisations to select pathways aligned with vendor viability, tooling maturity, and operational risk tolerance. Ultimately, successful transformation requires technical evolution to be paired with organisational readiness. By aligning hardware, applications, DevOps practices, and workforce governance, enterprise leaders can execute a sustainable roadmap that delivers operational resilience, agility, and long-term business value.

References

- [1] Accenture: *Solving the mainframe application conundrum* 2022. [Link](#)
- [2] Companion Data Services, LLC: *From Dependency to Agility. A Proven Path to Mainframe Vendor Independence.* [Link](#)
- [3] AWS: *AWS for mainframe modernisation.* December 2025. [Link](#)
- [4] Advanced: *Mainframe modernisation Business Barometer Report* (4th annual). [Link](#)
- [5] EY: *Unlock technological progress: the power of Z unleashed.* 2026. [Link](#)
- [6] Capgemini: *Mainframe modernisation patterns for financial services.*[Link](#)
- [7] BMC: *Maximize the Benefits of Your Mainframe CI/CD Pipeline.* March 2026. [Link](#)
- [8] Octopus: *Mainframe CI/CD: core principles and 6 tips for success.* 2026. [Link](#)
- [9] Planet Mainframe: *Linux and the Mainframe: A Platform Relationship Coming Into Its Own.* 2026. [Link](#)
- [10] Konidena et al: *DevOps for Mainframe Environment, Accelerating Legacy Software Delivery through Continuous Integration and Deployment.* 2025. [Link](#)
- [11] IBM: *Application modernisation on the mainframe.* [Link](#)
- [12] TechChannel: *Data Governance and Compliance on Mainframes: Aligning With Modern Regulations and Requirements.* 2025. [Link](#)
- [13] KMPG: *From legacy to leading edge. Revolutionizing mainframe modernisation.* 2025. [Link](#)
- [14] CROZ: *One Giant Leap: How We Upgraded Seven Mainframe Versions.* 2025. [Link](#)
- [15] IBM: *The future of AI on the mainframe: 4 paths for mainframe-driven AI innovation.* 2026. [Link](#)
- [16] Rocket software: *Your Guide to Upgrading Your Mainframe's Infrastructure Resilience.* [Link](#)
- [17] Planet Mainframe: *Empowering Organizations: Achieving Continuous Availability with IBM's Unyielding Commitment to Resilience.* 2024. [Link](#)
- [18] Elnion: *Sustainability in Mainframe Operations: A Critical Path Forward.* 2025. [Link](#)
- [19] Elnion: *Fujitsu GS21 Mainframe: The Strategic Backbone for Enterprise Computing in Asia.* 2025. [Link](#)
- [20] Fujitsu: *Expansion of modernisation.* 2025. [Link](#)
- [21] Fujitsu: *MM2Agentic From Mainframe to Agentic.* 2026. [Link](#)
- [22] Elnion: *ClearPath Forward: Why Unisys Still Matters in the Mainframe Market.* 2025. [Link](#)
- [23] Unisys: *Modernize your Mainframe Ecosystem.* 2026. [Link](#)
- [24] IBM: *Mainframe Application Modernisation Solutions.* 2026. [Link](#)
- [25] Microsoft: *Unisys ClearPath MCP virtualization on Azure.* 2026. [Link](#)
- [26] Sharwood: *Fujitsu confirms mainframe biz to die in 2035, in time for quantum AI supercomputers to take over.* 2026. [Link](#)
- [27] IBM: *DevOps for IBM Z.* 2026. [Link](#)

About this paper

Our authors

This document was developed under the Open Mainframe Project's Summer 2026 Mentorship program and sponsored by the Mainframe modernisation Working Group.



Rhian Barton | Mentee @ LFX 2026 Summer Mentorship

Mainframe Infrastructure Engineer and Degree Apprentice at Barclays, currently pursuing a degree in Digital and Technology Solutions specialising in Mainframe at Manchester Metropolitan University.



Bruno Azenha | Project Mentor

I am a technology strategy and modernization professional with over 16 years of experience defining, designing, and managing large-scale transformation programs across multiple industries. As an active contributor to the Open Mainframe Project's Mainframe Modernization Working Group, I regularly support and advance thought leadership in enterprise mainframe modernization.

Special thanks

This work would not be possible without the useful suggestions and feedback of members of the Mainframe modernisation Working Group, part of the Open Mainframe Project. Special thanks to the great insights and suggestions provided by Armin Warda and Nithin Sai Jalukuru

About the modernisation Working Group

The modernisation Working Group, part of the Open Mainframe Project, intends to be a focal point for thought leadership around what it means to modernize mainframe applications.

We are an open group passionate about knowledge sharing, leading constructive discussions, challenging status quo, exploring creative solutions, and generating contents that will help the community in their journey.

Learn more about the working group at:

<https://openmainframeproject.org/our-projects/working-groups/modernisation-working-group/>