OPEN **MAINFRAME** PROJECT

Modernization

WORKING GROUP

The Hybrid Cloud Mainframe

Opening up to cloud-native developers

September 2025

Executive summary

In 2025, enterprise IT landscapes continue to evolve rapidly towards agility and cloud-native operations, with hybrid cloud strategies emerging as the dominant model. Gartner forecasts that 90% of organizations will have adopted a hybrid cloud model by 2027 [1]. Within this context, modern mainframes remain critical: according to IBM's Institute for Business Value (IBV), 78% of executives agree that mainframe-based applications will continue to have an important role in digital transformation [2].

Given the continued relevance of mainframes and the accelerated adoption of cloud computing across enterprises, this paper presents one of the several options that enable bridging mainframe and cloud development practices. Specifically, it discusses how using Linux operating systems on IBM Z, containerization technologies, DevOps pipelines, and other open-source tools extensively used in cloud environments, enables creating unified application development and delivery practices across different processor architectures (x86_64, ARM, and s390x). The proposed approach reduces cognitive overload, accelerates delivery time, and unlocks compute cost savings while continuing to leverage the mainframe hardware. For example, in the mainframe modernization business barometer report, 92% of organizations are considering modernization and only 6% believe the mainframe will be replaced in the near future [3].

Adopting this hybrid cloud strategy enables a smoother learning curve, encourages collaboration between development teams, enables application portability and dynamic workload placement, and improves the efficiency of software delivery. This is achieved without compromising the legendary benefits that traditional mainframe processes and tools have provided over the decades.

Evolution of the Mainframe Landscape

Despite their reputation as "legacy," mainframes remain foundational in global IT, powering the majority of mission-critical workloads in industries such as financial services, government, and telecommunications. However, the number of mainstream platform vendors with active advancement and support has sharply declined. In 2025, IBM stands as the major vendor with a clear, ongoing roadmap for mainframe hardware evolution covering both legacy applications and runtimes, along with cutting-edge AI support and hybrid cloud capabilities [4]. Other mainframe platform vendors have shifted mostly to maintenance and legacy support. Annex I provides a view on mainframe vendors and the outlook for their platforms.

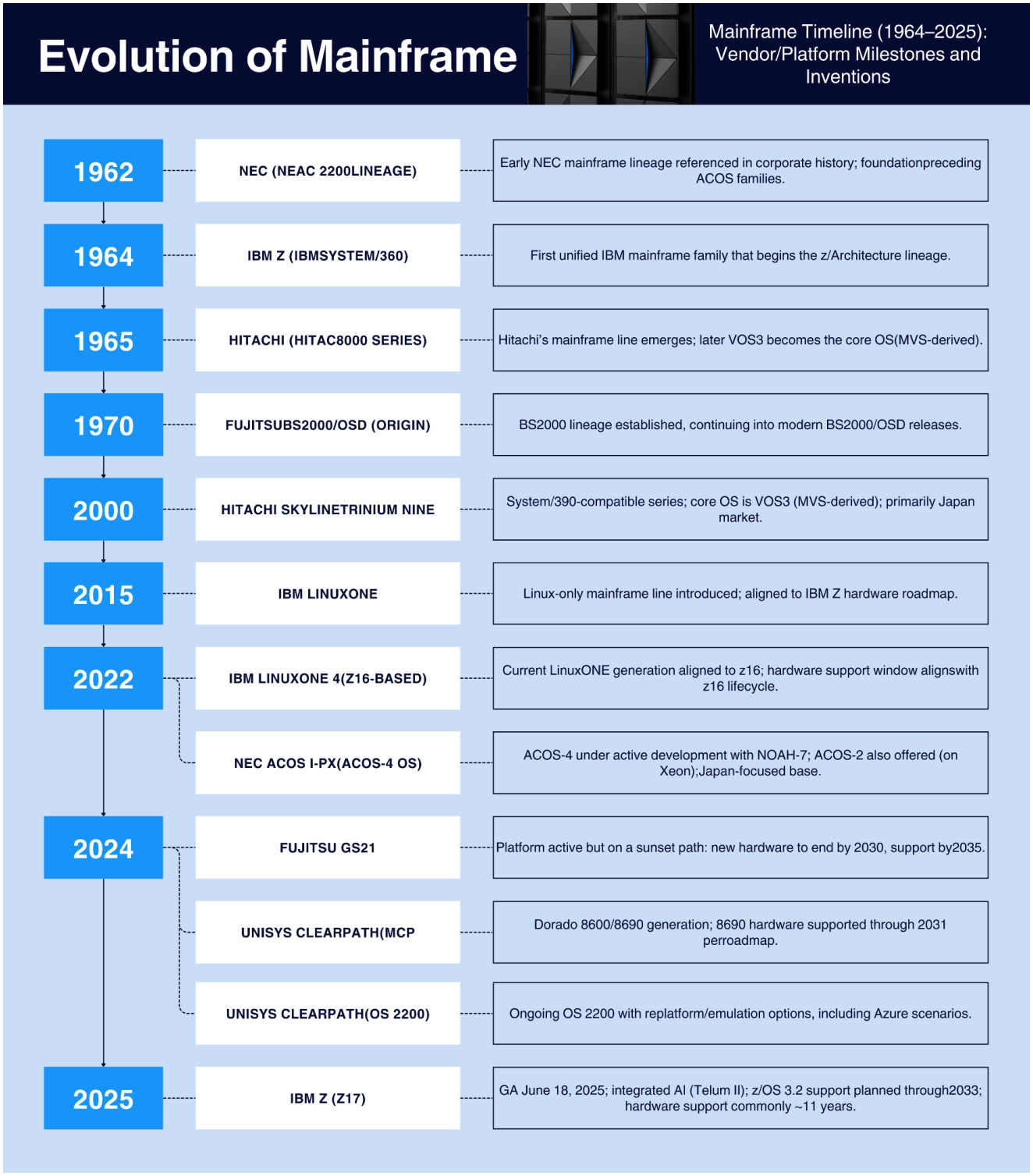

Fig.1 - Mainframe landscape evolution

Continued investment by IBM and its partners in the Z series and LinuxONE mainframe platforms has fostered a significant expansion of open-source software and community-driven initiatives. This has brought popular technologies to the platform, not only to the long-standing z/OS operating system but also through popular Linux distributions like SUSE, Ubuntu, and Red Hat Enterprise Linux [5].

Modernizing and evolving legacy mainframe applications does present challenges with regard to application delivery strategy and skills. Using common development teams for disparate systems — for example, mainframe and cloud — faces significant obstacles, including a steep

learning curve and the constant switching between different platform environments, abstraction layers, frameworks, and tools. This typically impacts development speed and increases delivery complexity. Conversely, maintaining dedicated development teams for different platforms can lead to developer underutilization or bottlenecks, given the cyclical nature of changes across systems. The availability of Linux environments on the mainframe, along with enterprise grade open-source software and community-driven initiatives, presents a new option for bridging the gap between traditional mainframe technologies and cloud technologies [6].

The hybrid mainframe technology blocks

Enterprises looking to simplify application modernization and software delivery across hybrid environments can nowadays leverage enterprise grade open-source technologies that span mainframes and public/private clouds. Let's look at how to unify application delivery across different compute architectures (s390x for mainframes and x86_64 for clouds).

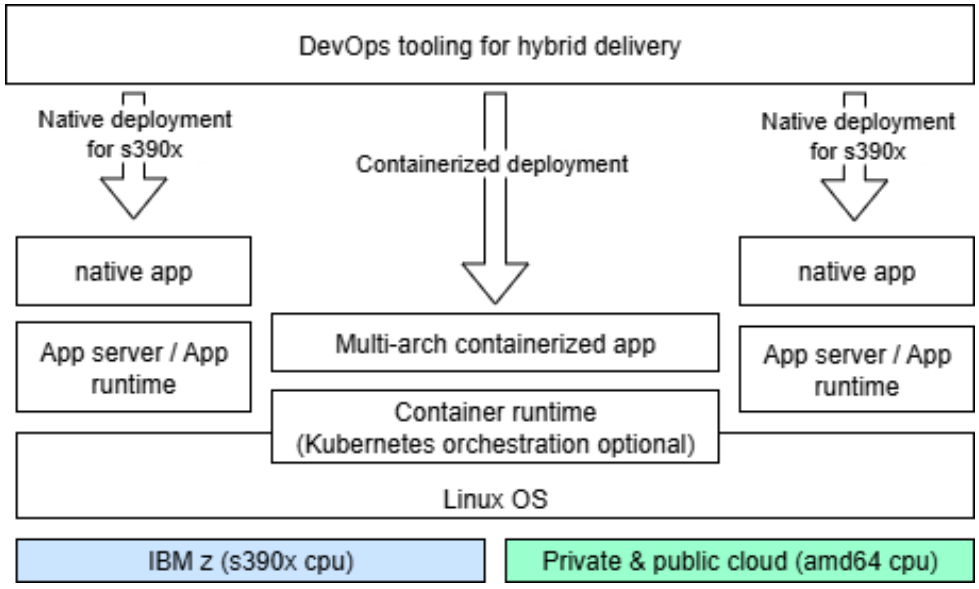

Fig.2 - Building blocks for hybrid cloud application delivery

Looking at Fig.2, key building blocks for unified hybrid application delivery are:

- **Linux Operating System (OS):** choosing a Linux distribution that is available and supported for both mainframe and cloud is the first step to reduce retraining and cognitive load for developers. It also unlocks optimization opportunities by allowing, for example, developing and running initial tests on cloud environments, with pre-production and production running on the mainframe [7].
- **Applications deployment model:** deciding whether to build applications natively for the Linux OS and application runtime or to wrap them inside a multi-architecture container. Each model has its strengths, and both can coexist if needed [8].
- **Container runtime and Kubernetes (K8s):** the container runtime is what allows running containerized applications in the same manner anywhere (mainframe or cloud). When

dealing with complex and high-volume application portfolios, using Kubernetes to manage containerized applications simplifies management at scale with greater speed, security, and stability [9].

- **DevOps tooling:** selecting tooling that enables building and deploying applications on different architectures or creating multi-architecture containers is essential. The pipelines need to be designed to automatically test and deploy apps on different platform architectures (s390x for mainframes and x86_64 or ARM for clouds), which will ensure accelerated delivery and increased developer efficiency. Using infrastructure as code (IaC) as part of the pipelines is also essential to remove potential slowdowns and blockers around making infrastructure components available to deploy applications [10].

The benefits of this integrated hybrid approach are tangible: enterprises report delivery time reductions of up to 40% [10], reduced context switching for developers, and improved consistency across environments.

The technology options

Let's dive further into popular technologies currently available that enable a common hybrid technology stack. Figure 3 provides an overview of some technologies commonly used on cloud environments and that would enable a seamless transition to mainframes.

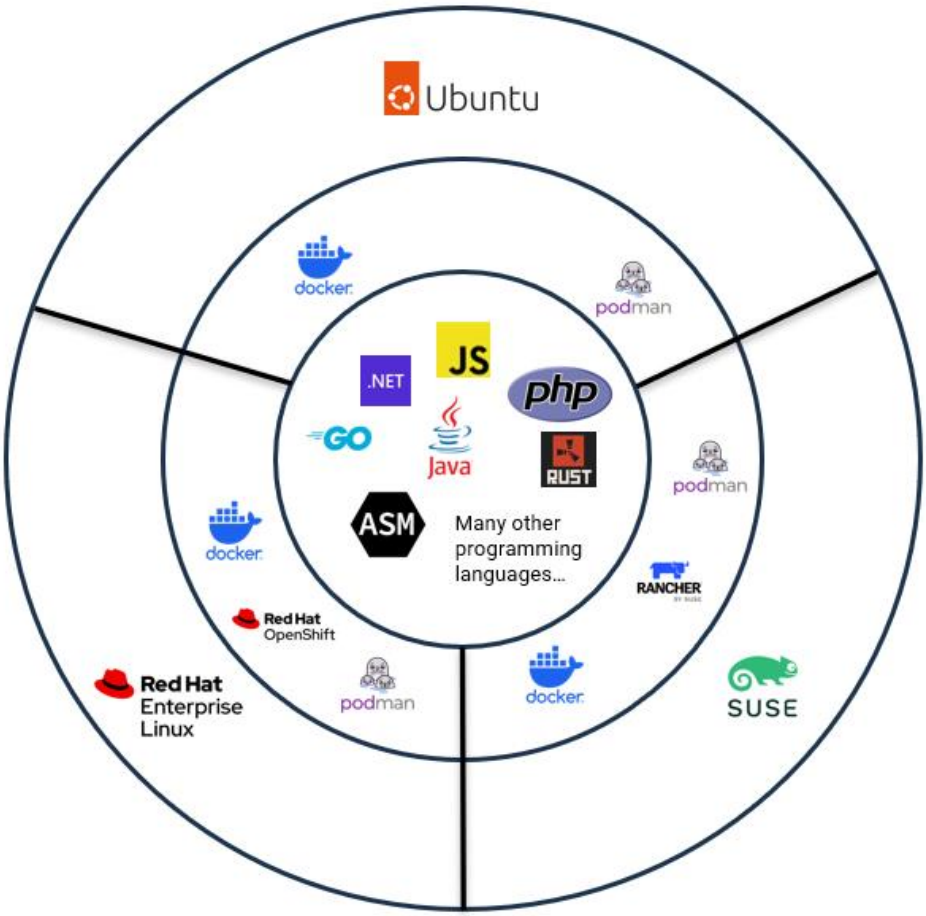

Fig.3 - Key technology options for hybrid application delivery

The OS layer

Linux for cloud environments and IBM Z/LinuxONE mainframes is available through three major enterprise distributions: Canonical Ubuntu (Ubuntu), SUSE Linux Enterprise Server (SLES), and Red Hat Enterprise Linux (RHEL) [11]. The principal benefit of these distributions is operational parity with x86_64/ARM: identical package repositories, kernel capabilities, SELinux/AppArmor profiles, systemd, and automation stacks, enabling uniform platform engineering and SRE practices across heterogeneous estates. Co-locating Linux on mainframe hardware provides data adjacency, high I/O throughput, strong isolation, and vertical scaling, which together support dense consolidation and resilient operations [12].

The containerization layer

While having Linux environments running on a mainframe is a good starting point for enabling hybrid cloud, using containers instead of deploying applications natively to Linux offers several key benefits, primarily revolving around consistency, efficiency, and portability. Containers package an application and all its dependencies, which leads to fewer issues when redeploying across heterogeneous environments. Containers are also more lightweight than VMs, which leads to higher density, faster startup times, and overall resource usage efficiency [13].

For simple, single-container applications, running them on top of a container runtime like Docker or Podman may suffice. However, enterprise-grade applications typically benefit from, and even require, a full-fledged Kubernetes container orchestration platform. Rancher by SUSE and OpenShift by Red Hat are currently the only two distributions with enterprise support for mainframe (s390x) and traditional compute and cloud environments (x86_64) [14]. Ubuntu does have a Kubernetes distribution but supporting x86_64 environments only. Ultimately, the goal of building multi-arch container images that target the s390x and x86_64 CPU architectures is to have a single container image that can be used on the cloud and on the mainframe, leveraging the exact same container runtime on both environments to make application portability seamless [15].

The application layer

Thanks to communities and software vendors' efforts, all popular programming languages and frameworks (e.g., Rust, Go, Python, and others) typically have feature parity between the s390x (mainframe) and x86_64 (cloud) distributions [16].

The question then becomes whether to use native deployment, which involves installing the application directly onto the Linux operating system and relying on its libraries and configurations, or to use a containerized deployment.

Using a direct (native) application deployment is typically well suited for scenarios like the following:

- Availability of deep OS-level expertise: If a team is highly skilled in Linux administration and is comfortable managing dependencies and configurations directly on the OS, native deployment can be straightforward [16].
- Application style: Large monolithic applications not designed to be broken into smaller components and services may not benefit from containerization.

- Need for tight OS integration: Some applications, such as low-level system services or those that need to directly manipulate the operating system or hardware, are better suited for native deployment.

Containerized deployment is well suited when facing the following scenarios:

- Implementing a hybrid cloud dev strategy: when developing on the cloud and deploying on the mainframe, containers provide the portability needed to seamlessly move applications across different infrastructures and architectures (x86_64 and s390x) [17].
- Delivery agility: With containerization, provisioning environments becomes easier and quicker, allowing teams to prioritize resources and release environments when no longer needed.
- Developer agility: Containers allow developers to work independently and switch easily between cloud and mainframe environments because they leverage the same containerization platform [18].
- Maximize resource utilization: Containers share the host's kernel and are lightweight, allowing higher density on mainframe Linux VMs than individual native applications, reducing cost and improving efficiency.

Multi-Architecture DevOps

Regardless of deploying applications natively or using containers, selecting DevOps tools that can build, deploy and test applications in different architectures is essential. Fig. 4 illustrates some of the key stages to be considered in the DevOps pipelines.

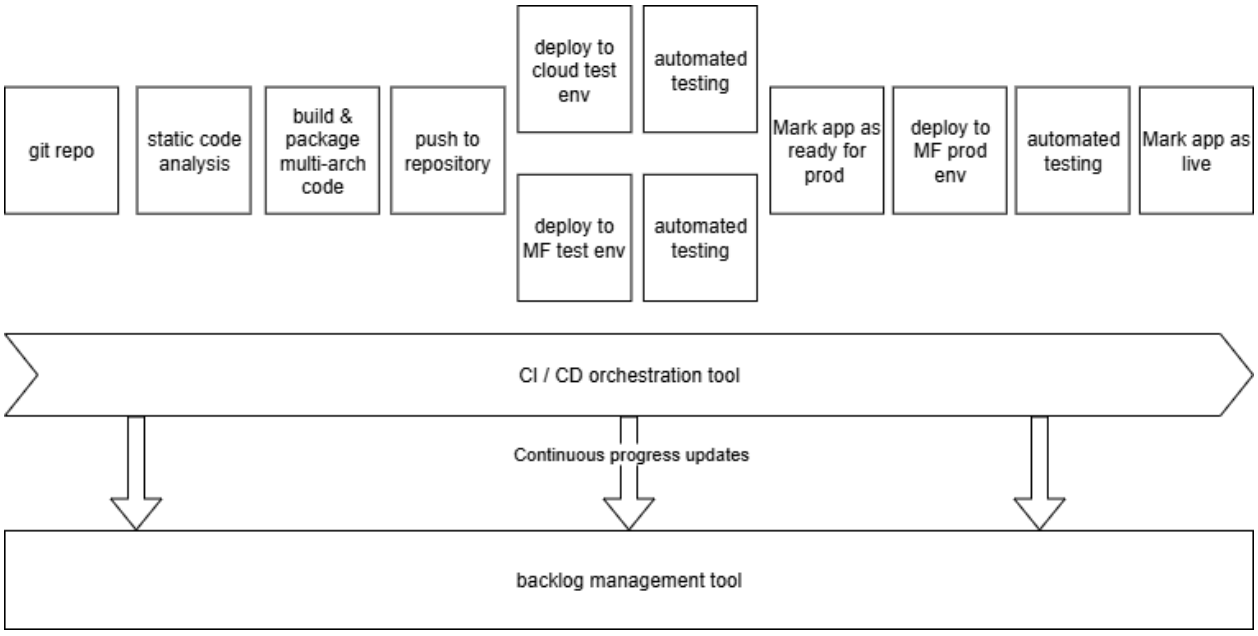

Fig. 4 - Key DevOps stages that enable hybrid application delivery

Generic CI/CD tools like Jenkins, GitHub Actions, GitLab CI/CD, and Tekton can target s390x by adding dedicated runners or build nodes. Maintaining performance tests, synthetic probes, and parity checks to confirm consistent behaviors across architectures reduces the risk of latent issues or discrepancies at the OS and hardware level [19].

Conclusion

Hybrid IT has become the pragmatic target operating model for enterprises that must balance agility with uncompromising reliability, security, and compliance. In this context, the mainframe continues to serve as the backbone for business-critical processing, while Linux on IBM Z/Linux ONE provides a cloud-native substrate that aligns with modern engineering practices. Treating Linux on s390x as a first-class platform — complete with enterprise distributions, containers, Kubernetes, and multi-architecture DevOps — removes historical friction and enables consistent pipelines and governance across heterogeneous estates.

A developer-first approach makes the difference. Standard languages and frameworks (Java, Python, Go, Rust, Node.js), container packaging with multi-arch images, GitOps-driven promotion, and policy-enforced supply chains bring the speed and simplicity of cloud to mainframe-adjacent workloads. This approach preserves data proximity and mainframe qualities of service while enabling incremental modernization through APIs and event-driven patterns, not disruptive rewrites.
Strategically, the mainframe platform has consolidated across three pillars. First, Linux as the operational fabric for cloud-native workloads running close to systems of record. Second, Kubernetes to unify containerized applications across hybrid environments. Third, multi-architecture DevOps as the default.

The path forward is clear and actionable. Start with a canonical Linux-on-mainframe architecture, deploy GitOps from day one, and prioritize high-impact, low-risk modernization patterns — API enablement, container adjacency to data and core applications, and selective legacy apps-to-container consolidation. Measure progress with engineering and reliability KPIs that span mainframe and the broader estate. This blueprint elevates the mainframe from a perceived legacy constraint to a modern, integral node in the hybrid cloud—combining resilience and scale with developer velocity and operational consistency.

References

[1] Gartner: 90% of organizations will adopt Hybrid Cloud through 2027. CIO Dive, Nov 2024. Link
[2] IBM Institute for Business Value: *Mainframes as mainstays of digital transformation*. October 2024. Link
[3] Advanced: *Mainframe Modernization Business Barometer Report* (4th annual). Link
[4] IBM Hybrid Cloud Roadmap 2025. Link
[5] Linux on IBM Z and LinuxONE Open-Source Software Report (July 2025). Link
[6] Kyndryl's 2025 State of Mainframe Modernization Survey Report. Link
[7] IBM LinuxONE Open-Source Software Report (July 2025). Link
[8] IBM Z Modernization and Containerization Survey 2025. Link
[9] IBM Kubernetes on Mainframe Adoption Report, 2025. Link
[10] IBM / IDC Report: *Accelerated Application Delivery with Hybrid Mainframes*, 2025. Link
[11] IBM LinuxONE Platform Overview, 2025. Link
[12] IBM Redbooks: *Linux on IBM Z Performance and Scalability*, 2025. Link
[13] IBM Containerization and Cloud-Native Strategy for Mainframes, 2025. Link
[14] SUSE Rancher and Red Hat OpenShift for s390x Deployment Guide, 2025. Link
[15] IBM Multi-Architecture Container Strategy, 2025. Link
[16] IBM Developer Community: Programming Language Support on IBM Z, 2025. Link
[17] IBM Hybrid Cloud Modernization Best Practices, 2025. Link
[18] IBM Developer: Containers for Mainframe Agility, 2025. Link
[19] IBM DevOps Solutions for Multi-Architecture Pipelines, 2025. Link
[20] Scribd: IBM z17 Release Details, 2025. Link
[21] Scribd: IBM Z Lifecycle and EOS Information, 2025. Link
[22] Scribd: IBM LinuxONE 4 Release Notes, 2025. Link
[23] IBM: LinuxONE for Cloud-Native and Virtualization, 2025. Link
[24] Wikipedia: NEC SX Series History, 2025. Link
[25] Wikipedia: ACOS Mainframe Status, 2025. Link
[26] Wikipedia: Hitachi HITAC 8000 Series, 2025. Link
[27] Wikipedia: Hitachi Skyline Trinium Nine Support, 2025. Link
[28] Fujitsu BS2000/OSD History, 2025. Link
[29] The Register: Fujitsu Mainframe Sunsetting, 2025. Link
[30] Unisys ClearPath History, 2025. Link
[31] Unisys ClearPath Dorado 8600 / OS 2200 Release Notes, 2025. Link
[32] Learn Microsoft: OS 2200 Azure Emulation, 2025. Link

About this paper

Our authors

This document was developed under the Open Mainframe Project's Summer 2025 Mentorship program and sponsored by the Mainframe Modernization Working Group.

Amrutha Rajsheker is a final-year Computer Science and Engineering (IoT) student at Saveetha Engineering College, Chennai. She is the global winner of the IBM Z Datathon 2024 and an IBM Z Student Ambassador. She has experience in AI, IoT, and cloud technologies, with a strong passion for learning, research, and development in mainframes and Linux-based platforms.
She founded and leads a student community “Z Tech Enterprise Hub” focused on mainframe learning, development.

Bruno Azenha is an experienced professional in technology strategy and modernization, with over 16 years’ experience defining, designing and managing the delivery of large transformation programs across multiple industries.
He is an active contributor to the Mainframe Modernization Working Group, part of the Open Mainframe Project initiative, and regularly supports advancing the thought leadership around the mainframe modernization topic.

Special thanks
This work would not be possible without the useful suggestions and feedback of members of the Mainframe Modernization Working Group, part of the Open Mainframe Project. Special thanks to Steven Dickens for his guidance and perspectives on the mainframe world.

About the Modernization Working Group
The Modernization 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/modernization-working-group/

Appendix I - Major Mainframe Vendors

The table below summarizes major mainframe technologies background, their supportability status and outlook as of July 2025.

Table1. Mainframes Technology Landscape in 2025

Vendor/ Platform	First Mainframe Release (Name, Year)	Latest/Last Mainframe Release (Name, Year)	Actively Supported in 2025?	Status Summary and Outlook
IBM Z (z/Architecture)	IBM System/360 (1964)	IBM z17 (Jun 2025) [20]	Yes. z/OS 3.2 support through 2033 z17 HW support expected till 2036 (typically 11 years)	Ongoing cadence of hardware and OS; LinuxONE aligned to z16/z17; clear lifecycle/EOS schedules; hybrid cloud focus continues [21]
IBM LinuxONE (z-based)	LinuxONE (2015)	LinuxONE 4 (z16-based WHEN??) [22]	Yes. Linux OS (RHEL/SLES/Ubuntu) as per vendor policies LinuxONE 4 HW supported till around 2029	Positioned as secure Linux mainframe for cloud-native and consolidation; shares z HW roadmap; OpenShift and virtualization emphasized [23]
NEC SX series (Japan)	NEAC 2200 (1962) [24]	ACOS i-PX with ACOS-4 OS (2022) [25]	Yes. Hardware support unclear ACOS-2 and ACOS-4 still sold as of 2025 with support expected till 2030 ACOS-4 under active development	NEC returned to proprietary NOAH for ACOS-4 (NOAH-6, then NOAH-7 in 2022); ACOS-2 runs on Intel Xeon; Japan-focused installed base [25]

Hitachi (Japan)	HITAC 8000 Series (1965) [26]	Hitachi Skyline Trinium Nine Series with System/390-compatibility (2000) [27]	Unclear Scattered information site ongoing extended support in Japan	core operating system is VOS3, an MVS-derived fork; Japanese market focus; modernization path often involves migration to IBM Z [27]
Fujitsu BS2000/OSD (Japan/EMEA)	BS2000 lineage (1970s) [28]	GS21 (2024) [29]	Yes, with constraints. All support will cease by 2035 following announcement to sunset the mainframe business	Fujitsu emphasizes gradual sunsetting while maintaining limited OS/application support [29] hardware sales to end by 2030
Unisys ClearPath Forward (Dorado 2200/MCP Libra)	UNIVAC (1950s), ClearPath (1992) [30]	ClearPath Dorado 8600 Series hardware (2024), ClearPath OS 2200 (2023) [31]	Yes. Dorado 8690 hardware supported through 2031; OS/app support also available through emulation for OS 2200 operating system and apps	OS 2200 has an option for replatforming/ emulation into Azure [32]