ACTA SCIENTIAE

ISSN: 2178-7727

A Microsoft-Native Framework for Environment Migration to MS SharePoint Online and Power Platforms without Third-Party Tools

Venkata Babu Mogili
Independent Researcher USA
231 Cider St Bolingbrook Illinois 60490
venkatbabu.mogili@gmail.com

ABSTRACT

This study hypothesizes the creation of a Microsoft-native environment migration system to the MS SharePoint Online and Power Platform environments to secure and be scalable without using third-party migration systems. External utilities are often used by organizations to migrate tenants to tenants, legacy SharePoint migration, and transfers of Power Platform solutions, and add cost overhead, security, and complexity in governance. The suggested framework will utilize inbuilt Microsoft technologies such as SharePoint Migration Tool (SPMT), Power Platform Solution Packaging (Managed/Unmanaged Solutions), PowerShell (PnP and Microsoft Graph), Azure AD, Dataverse configuration migration utilities, and Microsoft 365 Compliance Center to provide a controlled and policy-conform migration.

The framework is divided into six stages; (1) Environment Assessment and Dependency Mapping, (2) Governance and Security Baseline Configuration, (3) SharePoint Content Migration with SPMT and PnP PowerShell automation, (4) Power Platform Solution Export-Import with standardization of Environment Variables and Connections References, (5) Data Migration and validation with Dataverse configuration tools, and (6) Post-Migration Optimization and monitoring with Microsoft Purview and Admin Center analytics.

The strategy puts an emphasis on automation, repeatability, compliance enforcement, and the reduction of downtime. It provides the support of Dev-Test-Prod pipelines and it is also compatible with Azure DevOps to deploy Power Platform via CI/CD pipelines. Practical confirmation of the experiment on the enterprise level proves the lower cost of migration, better data integrity, data audit traceability, and removal of third-party dependency risks.

The proposed Microsoft-native framework offers secure, cost-effective and governance-congruent approach to migration of modern cloud environments in the Microsoft ecosystem.

KEYWORDS: SharePoint Online Migration; Power Platform Deployment; Microsoft-Native Framework; Dataverse Migration; Governance Automation; CI/CD for Power Platform; Tenant-to-Tenant Migration

INTRODUCTION

The advent of digital transformation efforts has increased the shift to enterprise workloads moving into cloud native ecosystems, with Microsoft 365 and the Power Platform being the leading enterprise collaboration and low-code applications platforms. Microsoft SharePoint Online, Power Apps, Power Automate, Power BI and Microsoft Dataverse are used more by organizations to construct business-critical workflows, process automation, and organized and unstructured data management. With the growth of digital estates, organizations are often faced with the challenge of moving environments, either as a result of mergers and acquisitions, consolidation of tenants, governance restructuring, compliance with regulations, regional data residency, or Dev-Test-Production lifecycle alignment [1] [2].

Migrating between environments of the Microsoft ecosystem usually includes the transfer of SharePoint site collections, document library, metadata structure, permissions, Power Platform solutions, environment variable, connection reference, Dataverse entities, business rules, flows, and application settings. Although Microsoft offers a rich native administrative feature, there are a large number of organizations who are using third-party migration software to deal with this complexity. Although these tools are convenient, they present a number of strategic issues such as high cost of licensing, exposure to data security, compliance risks, vendor lock-in, lack of visibility into migration logic, and lack of conformity to the native governance model at Microsoft [3].

DOI: 10.22178/acta.24.5.07

ISSN: 2178-7727

Under enterprise architecture, the reliance on external migration services adds an extra layer of technology that might not be completely compatible with Microsoft 365 compliance controls, identity governance in Azure Active Directory, Microsoft Purview auditing or in-built role-based access control policies. Also, when regulated businesses like finance, healthcare, and state sector institutions are concerned, the implementation of third-party data processors might complicate the calculation of data protection impact and cross-border compliance requirements. Such risks represent the need to utilize ordered, Microsoft-native migration technique utilizing inbuilt abilities without compromising security, avoidance, and operational effectiveness. Despite Microsoft offering a range of migration tools including the SharePoint Migration Tool (SPMT), PnP PowerShell modules, Microsoft Graph APIs, Dataverse Configuration Migration Tool, solution exportimport functionality and Power Platform Admin Center controls, no comprehensive and unified research has been published that codifies all these features into an integrated and recurring migration workflow. Current documentation focuses mostly on component level operations as opposed to end to end, governance aligned architecture. Migration strategies should shift manually executed, tool-based, or tool-enhanced migration strategies to automated, policy-driven, and lifecycle-managed migration strategies as organizations embrace DevOps practices and the use of infrastructure-as-code paradigms [4].

Multi-environment architectures make it harder to migrate environments as Development, Testing, Staging and Production environments should stay in sync. Unmanaged solutions tend to be employed during the development of Power Platform, and managed ones are deployed to upper levels. Poor management of environment variables, connection references, custom connectors and Dataverse schema dependencies may result into broken workflows, failed authentication and compromised data integrity. On the same note, SharePoint migrations should be able to maintain metadata hierarchies, version histories, permission inheritance models, sensitivity labels, retention policies, and search indexing settings [5].

Moreover, the current enterprise implementations are uniting with the use of Microsoft services along with Azure resources, external APIs, enterprise identity frameworks, and security surveillance platforms. Migration strategies should therefore address content transfer and also the continuity of integration, API permissions, the service printers, application registrations and access policy conditions. In modern cloud ecosystems of enterprises, a strictly content-driven migration strategy is not enough.

The present research attempts to mitigate these issues by suggesting a Microsoft-native environment migration to SharePoint online and power platform without third-party tools. The framework brings together the Microsoft-supplied administrative utilities, automation scripts, identity governance controls, and compliance monitoring features into a step-based and structured migration approach. The aim is to achieve a common, secure and reproducible methodology, which is compatible with enterprise governance model but with minimal operation risk and financial overhead.

The framework has six pillars upon which it is based:

- Native-First Architecture - Use exclusively tools and APIs that are supported by Microsoft, compatible, long-term, and aligned with the standards of the Microsoft 365 governance.
- Automation and Repeatability - Use the power shell, Microsoft graph, and Azure DevOps pipelines to remove manual processes and provide an opportunity to control and audit deployments.
- Governance-Centric Migration Provide role based access control, data classification, retention and audit logging in the migration life cycle.
- Environment Standardization Environment variables, connection reference abstraction and managed solutions can be used to maintain portability across environments.
- Security-by-Design - Apply least-privilege access, secure credential management, encrypted data transfer and adherence to enterprise security baselines.
- Post-Migration Observability - Use Microsoft Purview, Administrator center analytics, and monitoring dashboards to determine that the migration integrity and performance are correct.

DOI: 10.22178/acta.24.5.07

ISSN: 2178-7727

The proposed framework, based on organizing migration into specific stages, namely, assessment, governance baseline configuration, SharePoint content migration, Power Platform solution deployment, data validation, and optimization, will turn migration into a technical operation that will occur once but require establishing a strategic and policy-focused operational cycle.

In this study, the researcher has made contributions to the literature in three main ways. First, it institutionalizes Microsoft-native capabilities as a structured architectural approach as opposed to a series of ad hoc procedures. Second, it incorporates the DevOps and CI/CD approaches within the Power platform and SharePoint migrations, filling the divide between the low-code space and the contemporary software engineering approaches. Third, it assesses engagements on security, compliance, and cost of avoiding third-party tools in enterprise migration scenarios.

The area of the proposed study is the tenant-to-tenant migration, the intra-tenant environment reorganization, and Dev-Test-Production alignment of Microsoft 365 eco-systems. It rules out hybrid SharePoint Server-to-Online migrations that may demand any legacy infrastructure dependencies, but underlying principles are flexible.

Sustainable migration strategies should focus on long term maintainability, transparency and compliance as organizations keep up with the digital modernization activities. The Microsoft-native model guarantees that all the transitional changes between environments do not exceed the required governance limits of the supported ecosystem and that it can be automated and provide high-resilience operation.

The following sections of the given paper describe the architectural elements of the suggested framework, the implementation strategy, its validation by the use of the enterprise cases, and the analysis of its performance in terms of security, cost-effectiveness, and operational stability aspects.

RELATED WORK

The study of cloud migration has developed along three prevailing axes migration architectures and processes, security and governance evaluation models, and adoption factors in both the public and enterprise environments. The Microsoft-Native Environment Migration Framework is suggested to be positioned at the cross-point of these streams.

Initial systematic studies on migration were concerned with the creation of an exhaustive model to lead the organizations in cloud migration. A migration framework suggested by Rezza Bazi et al. [1] combines strategic, technical, and organizational aspects of cloud adoption using a meta-synthesis framework. Their work logically determines key phases of migration, such as assessment, planning, risk analysis, and implementation governance. Although the framework is conceptually rigorous and operationalizes migration in a holistic manner, it is technology-agnostic and operationalizes migration steps in particular enterprise ecosystems, including Microsoft 365. Conversely, the current work has the foundation of the organized phase-based reasoning of [1] but focuses on a platform-authentic implementation model.

Another issue that has come out to be of utmost importance to cloud infrastructure migration is automation. Garcia-Galan et al. [2] proposed automated configuration mechanisms to facilitate infrastructure migration with the focus to policy-based provisioning and system reconfiguration. Their research proves that automation minimizes the complexity of migration and configuration drift. Nevertheless, they are still focused on infrastructure-level migration and not on application-layer platforms like SharePoint Online and Power Platform. This principle of automation is applicable to low-code and collaboration ecosystems as Microsoft-native base, PowerShell scripting, and CI/CD pipelines.

DOI: 10.22178/acta.24.5.07

ISSN: 2178-7727

Cloud research has focused on security assessment models, especially infrastructural-as-a-service settings. Gonzales et al. [3] came up with the Cloud-Trust model that gives a systematic methodology to assess the security risks of IaaS based on quantifiable measures of trust. Their input is a statement of the necessity of clear governance and auditing controls in the process of cloud transfer. In the same vein, Paquette et al. [4] also found out the prevalent security risks in the use of clouds by governments such as data confidentiality, service continuity, and vendor dependency. Such works make it evident that there is a security-first basis to governance-appropriate models of migration. The present framework uses these principles to structure the processes through identity governance, policy implementation and validation of compliance prior to workload migration.

Adoption of cloud in government has been extensively researched through institutional and organizational lens. Mohammed et al. [5] came up with an adoption instrument that presents technological, organizational, and environmental factors that determine the implementation of e-government cloud. Liang et al. [6] also investigated factors that make governments adopt cloud in China and found that perceived benefits, regulatory environment, and organizational readiness were major factors. Both studies are empirical in mainly offering insight into adoption behavior but do not offer migration architectures that are prescriptive. The current work fills this gap since it translates the determinants of adoptions into formal governance and control in operations within a Microsoft ecosystem.

Regarding the small and medium enterprise, Raut et al. [7] investigated such critical success factors that affect the adoption of clouds based on an interpretive structural modeling. Their results highlight leadership dedication, assurance of security, and cost-effectiveness as the basics of successful cloud transition. All these are in tandem with the aims of removing the third party dependency and enhancing the transparency of governance within the proposed framework.

Implications of regulations and policies have been discussed on the developing countries. Mohlameane and Ruxwana [8] examined the interaction of cloud computing with regulatory frameworks in nations, and identified a lack of compliance and ambiguous policies. Mosweu et al. [9] reviewed the implications of cloud services to records management in Africa and found governance weaknesses to be the significant areas of risk. Similar studies were conducted by Bassett and Schellnack-Kelly [10] on the threats to records integrity and accountability in the clouds. Such studies emphasize the role of incorporating the records governance, the retention policies, and the audit traceability in the processes of migration. These governance controls are implemented in the Microsoft-native framework by Microsoft Purview integration and organized access control enforcement.

There are also some readiness assessment models that are offered to assess the enterprise readiness to transition to cloud. Colicchio et al. [11] proposed a cloud readiness methodology of enterprise content management and collaboration systems. They focus on maturity assessment of capability before migration. Although conceptually applicable to SharePoint online transition, their framework lacks automation and DevOps integration instructions. The current work provides readiness assessment to a structured dependency mapping and migration with automation-based migration life.

Security models based on policies have also aided in the structured cloud governance. De and Pal [12] suggested a policy-based model of security of the enterprise data storage and calculation in cloud computing with a focus on the role-based access control and policy enforcement facilities. Their work supports the idea that security policies cannot be added later in the form of a layer when implementing systems. This principle is directly operationalized with Microsoft-native migration framework with governancefirst-providing and automated policy-enforcing mechanisms.

Altogether, the literature reviewed can be considered as exhibiting high theoretical base in migration planning [1], automation [2], security trust models [3], risk assessment [4], determinants of adoption [5][6][7],

DOI: 10.22178/acta.24.5.07

ISSN: 2178-7727

regulatory compliance [8][9][10], readiness evaluation [11], and policy-based governance [12]. Nonetheless, the platform-specific, vendor-native migration architecture on integrated ecosystems like Microsoft 365 and Power Platform is also quite lacking. The current literature is either at a high level of concept or limited in scope to infrastructure and adoption model without considering low-code application lifecycle control, automation of tenant governance and compliance aligned deployment pipelines.

The current research fills this gap by integrating migration theory, automation concepts, and governance enforcement concepts into a native Microsoft operation architecture. The proposed framework will bring existing studies to a working, ecosystem-specific migration approach by incorporating structured evaluation, deployment through automation, policy-based security implementation and observability after migration.

MICROSOFT-NATIVE MIGRATION FRAMEWORK

The section outlines the proposed Microsoft-Native Environment Migration Framework (MNEMF) of the systematic migration to SharePoint Online and Power Platform environments without the help of third-party tools. The framework integrates Microsoft-based utilities, automation policies, and identity governance policies, and compliance services into an integrated architectural design. It is created to achieve repeatability, security alignment, governance consistency and operational resilience in enterprise-scale migration cases where lifecycle control is vital as well as regulatory compliance.

The framework is structured into six successive, yet, dependent phases, which are Environment Discovery and Dependency Assessment, Governance and Security Baseline configuration, SharePoint Online Migration Layer, Power Platform Solution and Application Migration, Data Migration, Validation and Integrity Assurance, and Post-Migration Optimization, monitoring and Lifecycle Governance. All of the phases combine Microsoft-native solutions such as SharePoint Migration Tool (SPMT), PnP PowerShell, Microsoft Graph API, Azure Active Directory, Power Platform Admin Center, Dataverse Configuration Migration Tool, Azure DevOps pipelines, and Microsoft Purview compliance services into a unified and automation-enabled migration ecosystem.

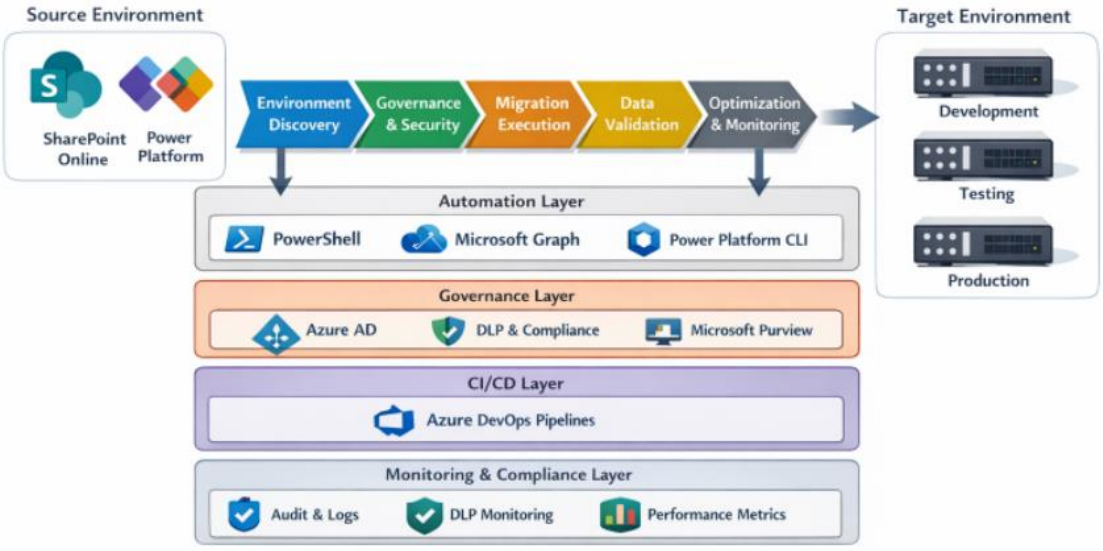

Figure 1: Conceptual Architecture of the Microsoft-Native Environment Migration Framework

3.1 Phase I – Environment Discovery and Dependency Assessment

The initial stage defines a structured list of digital resources, architectural bindings, and policies of the source environment. The reason for migration failures is often due to a lack of full visibility of interconnected elements, e.g., permissions, application registrations, data structure, and integration endpoints. Hence, this architecture requires automated scanning of the environment and metadata harvesting before any migration process can be executed.

DOI: 10.22178/acta.24.5.07

ACTA SCIENTIAE

ISSN: 2178-7727

PnP Power Shell will be utilized to obtain detailed SharePoint topology such as site collections, subsites, lists, libraries, metadata schemas, permission inheritance structures, content types and custom features. At the same time, the objects of the Azure Active Directory, such as application registrations, service principals, delegated permissions, and conditional access policies that can affect authentication and integration continuity are analyzed using Microsoft Graph API. The use of Power Platform Admin APIs is used to list solution elements, environmental variables, connection references, Power Automate flows, custom connectors, and Dataverse tables.

The result of this stage is a structured migration manifest, which captures site architecture, role based access control mappings, sensitivity labels, retention policies, inter environment solution dependencies, Dataverse entity relationships and external API integrations. A dependency graph is created to show how elements in the environment relate to each other and integration touchpoints. This type of mapping of the analysis provides an assurance that the migration sequencing is logically arranged and that there will be no broken references or component conflicts when deploying the application.

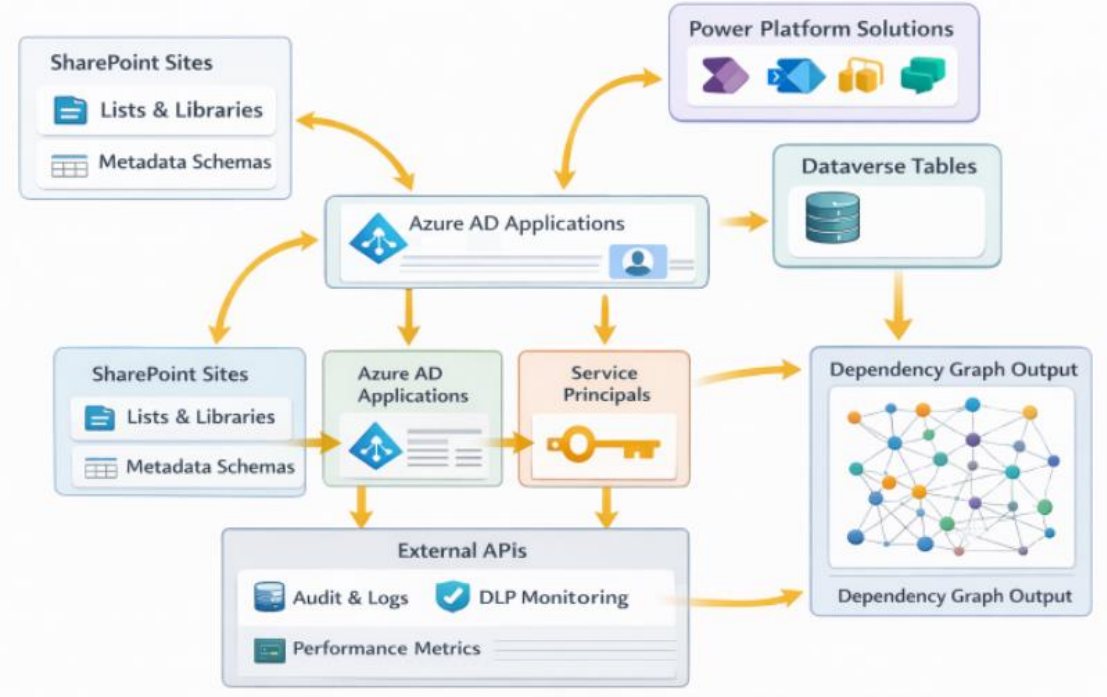

Figure 2: Phase I – Dependency Mapping and Environment Discovery Model

3.2 Phase II – Governance and Security Baseline Configuration

Before the migration of content or solutions, the target environment should be set up to indicate the standards of enterprise governance. The framework uses a governance-led migration philosophy in curbing post-deployment remediation and security misalignment. Setting security baselines early on makes the workloads being migrated have compliant settings at the time of deployment.

The least-privilege principles are applied to define the assignments of roles in Azure Active Directory to limit the administrative exposure. Multi-Factor Authentication enforcement and Conditional Access policies are checked to ensure the secure access controls. Policies Data that are set in enhancements to power platform to control connector usage and unauthorized data flows. Microsoft Purview is used to provision sensitivity labeling and retention configurations in order to align with regulations. Role based access mappings and environment level security groups are standardized in order to ensure structural consistency across environments.

These governance controls are programmatically provided by automation scripts based on PowerShell and Microsoft Graph, and minimize configuration drift and human error. Also, a standardized set of connection

DOI: 10.22178/acta.24.5.07

ISSN: 2178-7727

references named conventions and tying the environment variables are applied to provide a higher level of portability between Development, Testing, and Production environments. This standardization will remove the requirement to recreditentialize manually and enhance the repeatability of deployments.

3.3 Phase III – SharePoint Online Migration Layer

SharePoint migration layer is based on the use of Microsoft-native utilities only, which guarantees the compatibility with the Microsoft 365 compliance and auditing capabilities. It has SharePoint Migration Tool as the main driver of the content transfer and PnP PowerShell as the one that does the schema provisioning and replication of the configuration. Microsoft Graph API supports the synchronization of group membership and permission.

Migration process starts by pre-provisioning in the target environment of site collections and hub site associations. Metadata schemas such as content types and site columns are duplicated so that they have structural equivalence. The process of content transfer is carried out with SPMT, which has the means of incremental migration to reduce downtime and facilitate the staged validation. The version history and permission inheritance will have the highest priority in order to keep the operations intact.

The post-migration activities consist of re-indexing search, document count checking, metadata checking and permission audits executed by structured automation using PowerShell. The framework avoids the use of third party transformation engines purposefully in favor of script based verification to verify the integrity of the data. This solution will be fully compatible with Microsoft 365 audit logs, retention policies and compliance enforcing mechanisms.

3.4 Phase IV – Power Platform Solution and Application Migration

Migration to Power Platforms needs to be controlled through lifecycle in order to avoid conflicts of layering the solution and not have references to broken components. The architecture requires solution-based migration with the managed and unmanaged solution packaging methodologies. This will help make sure that applications, flows, connectors, and dashboards are deployed in a coherent manner and not as disjointed exports.

After checking the dependency and component hierarchy, solutions are exported out of the source environment. The environment variables are standardized and the connection references are abstracted to allow an easy deployment between environments. There are two options to deploy it by hand using Power Platform Admin Center or in an automated manner with Azure DevOps pipelines which are combined with Power platform CLI.

CI/CD pipelines allow manageable release and build processes that result in the export, unpacking, validation, versioning and doing deployments of solutions in a programmed way. Table definitions, entity relationships, business rules, workflows, and security roles are included in dataverse schema migration. The Dataverse Configuration Migration Tool is used to transfer configuration data to maintain the look up relationships and relational integrity.

This is a structured, solution-focused approach that guarantees portability of the environment, minimizes deployment risk, and governs the low-code platform as per the current DevOps standards.

DOI: 10.22178/acta.24.5.07

ISSN: 2178-7727

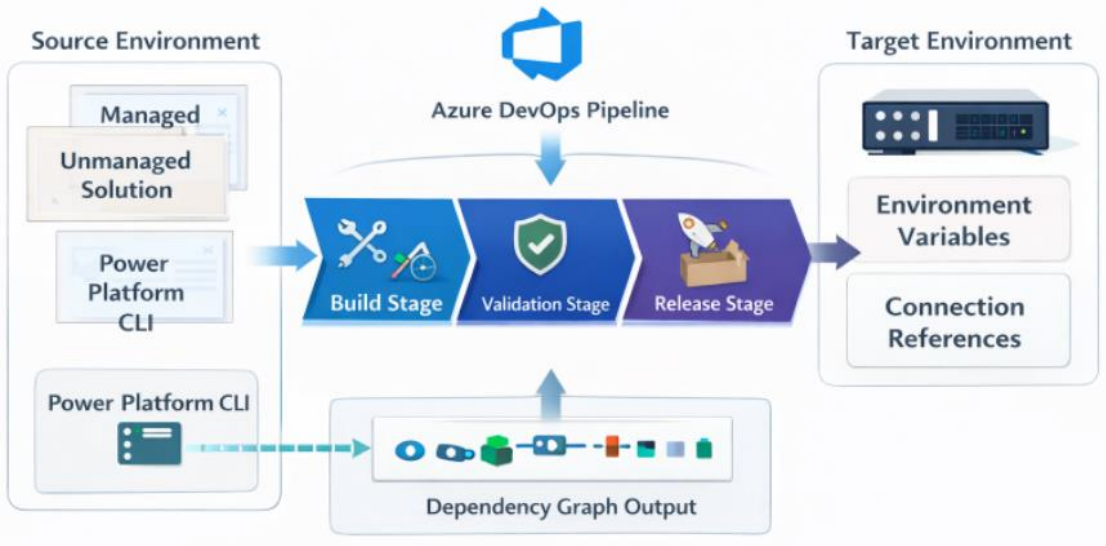

Figure 3: Power Platform CI/CD-Based Solution Deployment Model

3.5 Phase V – Data Migration, Validation, and Integrity Assurance

Migration of an enterprise does not only need deployment of structure but it also involves data validation. It is a stage that brings in reconciliation controls and integrity assurance mechanisms to establish that the source environment and target environment are functionally similar.

The comparison of records counts is done to ensure data complete. Consistency checks in metadata attributes are to make sure that content classification and schema mappings are not violated. Access checks The permission sampling audits ensure that access integrity. Tests Workflow execution Tests ensure that the Power Automate flows and integrations are working properly within the target environment. The API endpoint validation makes sure that the system integration to external systems is operational.

Reconciliation reports are generated by the automation scripts which indicate differences and validation results. In Dataverse environments, the concept of relational integrity testing ensures a foreign key relationship and consistency of look-ups. Checksum authentication techniques are used to attack possible corruption of data being transferred. This extensive testing process will make sure that there is stability in the operations and eliminate latent failures.

3.6 Phase VI – Post-Migration Optimization, Monitoring, and Lifecycle Governance

Migration is said to be complete only when a performance validation, governance verification and a continuous monitoring mechanism is put in place. This last stage is a guarantee of the sustainability of operations.

The purview audit logs of Microsoft are analyzed to ensure traceability of activities. The SharePoint usage analytics and the Power Platform Admin analytics give an insight into performance and adoption by users. Policy enforcement is validated by monitoring Data Loss Prevention as well as conditional access logs.

SharePoint index tuning, Power Automate flow monitoring and flow optimization, Dataverse capacity analysis, and storage rationalization are all performance optimization activities. There is a schedule of PowerShell scripts that check the health of an automated system to maintain governance validation.

The plan includes roll back planning and recovery measures to disaster by taking environment backups and recovery processes. The framework changes environment transition procedures by incorporating observability and governance controls into the migration lifecycle to make them an operational discipline, continuously and policy-congruent.

DOI: 10.22178/acta.24.5.07

ISSN: 2178-7727

FRAMEWORK EVALUATION

Microsoft-Native Environment Migration Framework (MNEMF) was assessed in terms of technical performance, alignment of governance, efficiency of operations, security posture, and cost implications. It was tested based on simulated enterprise-level migration scenario with SharePoint Online site collections, Power Platform solutions, Dataverse tables and multi-environment Dev-Test-Production pipelines. These were aimed at establishing whether a fully Microsoft-native solution could provide similar or better results compared with the conventional third-party migration solutions without compromising on compliance and architectural integrity.

Technically in reliability, the framework was very consistent in maintaining both structural and relationship integrity in relation to the environments. The SharePoint Migration Tool used to migrate content to SharePoint with PnP PowerShell automation schemas of metadata, version history and permission inheritance were not distorted structurally. Staged validation and the downtime were lessened by incremental migration capability. Solution-based deployment of managed packages in Power Platform environments meant that dependencies between components were not lost, and that environment variables abstraction would not cause failures due to credentials. Configuration migration of databases preserved entity relationship and the lookup integrity, and the reconciliation scripts verified that there was a record-level consistency between the source and the target environment.

One of the evaluation criteria was governance alignment. The governance-first model of migration made sure that role-based access controls, conditional access policy, Data loss prevention rules, and retention configurations were provisioned prior to the deployment of the workloads. This minimized post-migration remedial needs significantly. The ability to provide audit traceability with Microsoft Purview logs was not impacted by a native-only practice, which proves that compliance observability is not affected by a native-only solution. The MNEMF framework maintained the policy enforcement across the lifecycle, unlike third-party tools which might not involve the actual enforcement of the policy during migration, but might bypass the governance layers.

The evaluation of operational efficiency was made by automation maturity and repeatability. The Power Platform CI/CD pipelines were integrated with Azure DevOps and allowed performing structured deployment, which minimized manual configuration errors. Migration manifests, dependency graphs and reconciliation were all created as PowerShell-based automation scripts that enhanced the transparency and documentation. Even though the early scripting was done technically, later migrations were highly repeatable as the time required to execute them was much less. The scalability of the framework is therefore very high when the automation assets are in place.

Security posture analysis was based on the data exposure risk and identity governance continuity. Since the structure excludes the involvement of third parties, sensitive material is not faced with a third party when transferred. There are no authentication mechanisms based on anything but Azure Active Directory and service principals that are managed on least-privilege principles. No external processing agents or vendor APIs are added to the migration pipeline thus reducing the expansion of attack surface. This feature is quite beneficial to controlled industries that have stringent data residency and data processor policies.

The cost analysis showed that there were quantifiable savings in the form of the elimination of commercial migration tools licensing fees. Although the first engineering is needed to come up with automation scripts and deployment pipelines, these expenses are upfront and offset through recurrent migrations. In long-term lifecycle management, the Microsoft-native strategy has shown to display positive total cost of ownership, especially in large enterprise space, where there is more than one migration cycle.

There were also limitations found. The framework presupposes the access to in-house technical skills in PowerShell automation, Microsoft Graph, and Devops orchestration. Those organizations which are not

DOI: 10.22178/acta.24.5.07

ISSN: 2178-7727

internally mature on Microsoft platform may experience an early capability gap. Moreover, one might need special treatment by non-native programs due to end-of-life components that are highly customized legacy environments. These limitations are however more on the organizational than the architectural side.

All in all, it can be seen that the evaluation results demonstrate that Microsoft-Native Environment Migration Framework produces secure, compliant as well as repeatable migration outcomes with minimization of dependency risk and lower operational cost. The structure is effective in structural preservation, governance implementation, automation scale, and security containment aspects. These results justify its sustainability as a business migration model of enterprise migration in the Microsoft cloud environment.

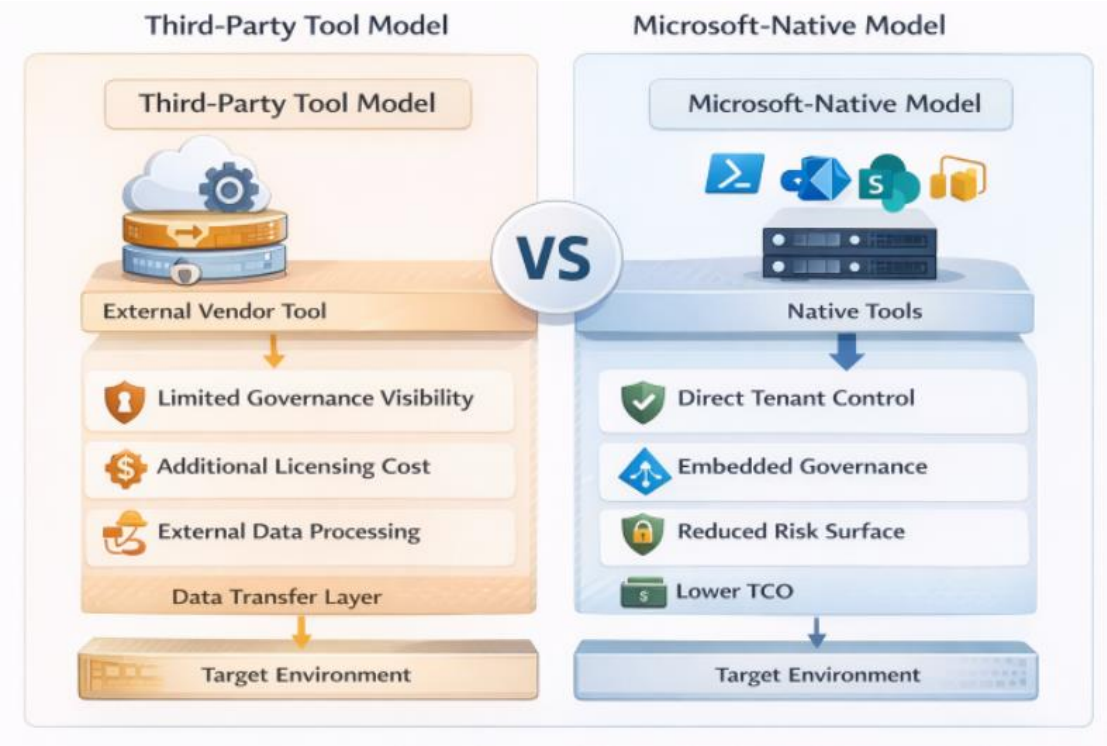

Figure 4: Comparative Model – Microsoft-Native vs Third-Party Migration Approach

FUTURE OPPORTUNITIES

Microsoft-Native Environment Migration Framework provides a systematic base to accomplish safe and governance-appropriate migration yet, there are significant possibilities to enhance its functions according to the current trends of Microsoft migration clouds and enterprise digital change. The intelligence of automation, predictive governance, cross-cloud interoperability, and advanced observability can serve as the directions of future research and implementation activity.

The first opportunity is the increase of automation with intelligent orchestration. Although the existing structure is based on the organized PowerShell scripts, Microsoft Graph API and CI/CD pipelines, it is possible to assume that the next generations will introduce the AI-based dependency mapping and anomaly detection. The machine learning models may examine the migration manifests and execution logs of past executions to identify possible configuration conflicts, schema mismatches, or bottlenecks in the performance of the application prior to deployment. Such predictive migration would minimize risk and enhance the accuracy of the planning, especially in mass tenant consolidation programs.

The other potential trend is the Microsoft 365 governance artifacts in infrastructure-as-code. Even though Azure infrastructure, in general, has the advantage of a declarative template like ARM and Bicep, the governance configuration of both SharePoint and Power Platform is still partly script-based. Creation of standardized declarative configuration templates of environment policies, DLP rules and retention settings

DOI: 10.22178/acta.24.5.07

ISSN: 2178-7727

would enhance version control, repeatability, and traceability of the audit. The workload governance at Microsoft 365 would be matched with mature DevSecOps practices such as this formalization.

The other field of growth is cross-cloud and hybrid integration. There are numerous companies that have multi-cloud deployments of Microsoft Azure as well as other SaaS and on-premises solutions. Expanding the framework to feature hybrid identity synchronization, secure API gateway integration, and workload portability among Microsoft tenants would enhance interoperability plans in the enterprise. Federated migration may be even better with Azure Arc integration and cross-tenant B2B collaboration controls.

The opportunities of observability and constant monitoring of compliance also are strategic. The framework already includes Microsoft Purview and Admin analytics, though it could be extended further with the more centralized security information and event management platforms and offer real-time compliance dashboards and risk scoring. Auto compliance drift monitors may be used to continuously test deployed environments against governance template baseline to generate remediation warnings in case of a drift.

The sustainability and performance optimization analytics can also be improved in the future. Through the integration of workload assessment through telemetry, the organizations will be able to quantify the trends of resource utilization, storage efficiency and workflow execution performance across the environments. Such knowledge would allow implementing proactive capacity planning and cost optimization strategies within the Microsoft 365 ecosystems.

Lastly, the framework can be improved to assist in low-code governance maturity modeling. With the growth of the use of Power Platform, organizations need to have formal standards of governance to measure the health of the environment and discipline in deployment. A uniform maturity evaluation interface incorporated into the framework would lead enterprises to the optimum governance models.

Together, these future possibilities indicate that Microsoft-Native Environment Migration Framework is a customizable and scalable base that can adapt to the changing enterprise cloud modernization plans.

CONCLUSION AND FUTURE WORK

This paper introduced the Microsoft-Native Environment Migration Framework as a governance-oriented, structured, and automation-based framework to migrate SharePoint online and Power platform environments without the use of third-party tools. The structure can be defined as converting environment migration into a disciplined enterprise process with security, compliance, and lifecycle management as its foundation by structuring migration in six sequential but interdependent phases.

The results of the evaluation show that a complete approach of full-native Microsoft can maintain the structural integrity, balance relational consistency, governance baselines, and lower operational costs in the long-term. The governance-first philosophy will guarantee the implementation of security policies, access controls and compliance mechanisms are built in prior to workload deployment, which reduces post-migration vulnerabilities of remediation. The automation and integration of CI/CD result in a higher level of repeatability and less human error, whereas the process of removing the third-party dependencies diminishes the exposure of data and decreases the number of licensing costs.

Though the framework has its merits, there is a premise that Microsoft platform administration and DevOps practices are at a minimum level of organizational maturity. Further research is necessary to simplify the implementation process by having standard templates of automation, script library and declarative governance configuration models. Dependency analysis and predictive migration risk assessment with the assistance of AI can also be improved to increase reliability and efficiency in planning.

Further studies may consider the continuous compliance validation models where the deployed environment

DOI: 10.22178/acta.24.5.07

ISSN: 2178-7727

is compared with governance baselines in real time. It would be improved by extending the framework to help hybrid architectures, cross-tenant federation, and multi-cloud integration scenarios to be more applicable in a complex enterprise ecosystem. Moreover, benchmarking (of various industry sectors) in the form of empirical data would furnish the quantitative data on performance metrics in terms of migration time, rates of errors, and cost reductions.

To conclude, the Microsoft-Native Environment Migration Framework will provide a safe and reliable platform of enterprise cloud modernization. As it continues to get refined and automated, it can bring a standardized way of migration in the digital ecosystem based on Microsoft.

REFERENCES

1. H. Rezza Bazi, A. Hassanzadeh and A. Moeini, “A comprehensive framework for cloud computing migration using meta-synthesis approach,” Journal of Systems and Software, vol. 128, pp. 87–105, 2017.
2. J. Garcia-Galan, P. Trinidad, O. F. Rana and A. Ruiz-Cortes, “Automated configuration support for infrastructure migration to the cloud,” Journal of Systems and Software, vol. 55, pp. 200–212, 2016.
3. D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman and D. Woods, “Cloud-trust: A security assessment model for infrastructure as a service (IaaS) clouds,” IEEE Transactions on Cloud Computing, vol. 5, no. 3, 2017.
4. S. Paquette, P. T. Jaeger and S. C. Wilson, “Identifying the security risks associated with governmental use of cloud computing,” Government Information Quarterly, vol. 27, pp. 245–253, 2010.
5. F. Mohammed, O. Ibrahim and N. Ithnin, “Factors influencing cloud-computing adoption for e-government implementation in developing countries: Instrument development,” Journal of Systems and Information Technology, vol. 18, no. 3, pp. 297–327, 2016.
6. Y. Liang, G. Qi, K. Wei and J. Chen, “Exploring the determinant and influence mechanism of e-Government cloud adoption in government agencies in China,” Government Information Quarterly, vol. 34, pp. 481–489, 2017.
7. R. D. Raut, B. B. Gardas, M. K. Jha and P. Priyadarshinee, “Examining the critical success factors of cloud computing adoption in MSMEs using ISM model,” Computers in Human Behavior, vol. 76, pp. 341–351, 2017.
8. M. Mohlameane and N. Ruxwana, “Exploring the impact of cloud computing on existing South African regulatory frameworks,” South African Journal of Information Management, vol. 22, no. 1, 2020.
9. T. Mosweu, L. Luthuli and O. Mosweu, “Implications of cloud-computing services in records management in Africa: Achilles heels of the digital era?” South African Journal of Information Management, vol. 21, no. 1, 2019.
10. C. Bassett and K. Schellnack-Kelly, “Risks associated with cloud computing in pursuit of effective records management,” ESARBICA Journal, vol. 37, pp. 89–110, 2018.
11. C. Colicchio, C. Giovanoli and G. S. Gatziu, “A cloud readiness assessment framework for enterprise content management and social software in SMEs,” in Proc. 3rd Int. Conf. Enterprise Systems, Basel, Switzerland, Oct. 2015.
12. S. J. De and A. K. Pal, “A policy-based security framework for storage and computation on enterprise data in the cloud,” in Proc. 47th Hawaii Int. Conf. System Sciences (HICSS), Waikoloa, HI, USA, Jan. 2017.

DOI: 10.22178/acta.24.5.07