MAINFRAME TO AZURE MIGRATION PATTERNS

Microsoft

Infosys®
Navigate your next

Table of Contents

1. Introduction

Mainframe modernization helps transform the application portfolio into agile and efficient systems while mitigating risks and lowering costs. However, mainframe migration can be complex. It needs custom approaches depending on the requirements, workloads and risk profile of the enterprise. While there are some common challenges to most mainframe migration journeys, there are also different types of migration patterns from rehosting to re-engineering that must be considered.

For instance, there are a set of repetitive architectural patterns for mainframe applications. Each architectural pattern that emerges on the mainframe needs a unique strategy to host it on Azure/SQL data platform. This whitepaper describes how to map existing architectural patterns and components to their equivalent components on Microsoft Azure.

2. Mainframe challenges and cloud migration platforms

The common challenges of mainframe systems are high licensing cost, lack of technology experts and the evolving needs of the digital world. Today's enterprises need the right platform that can off-load mainframe batches and migrate data processing to the new cloud platform in a seamless and risk-free manner.

Microsoft Azure cloud computing platform is a successful and proven target environment for transitioning from mainframe workloads to the cloud. With robust security features and the ability to scale services on demand, Azure offers a complete operational environment to support mainframe workloads that have been migrated to the cloud. In addition, Azure drives innovation of application portfolios that previously resided within the inflexible mainframe computing models, thereby improving the productivity of application developers and support personnel.

3. Choosing the right modernization approach

There are four modernization approaches which can be followed while migrating Mainframe applications to Azure:

- **Rewrite** – Here, applications are reengineered to a different language for next-gen architecture such as cloud, DevOps, etc.
- **Retain** – Here, application functionalities and existing platforms are not modified/changed. However, in order to enhance application functionalities, features such as API or microservices can be enabled on top of the applications

- **Rehost** – Applications are migrated to an emulation platform in Linux/Windows without altering the user experience and application functionality
- **Retire** – Applications are shut down and business-relevant functionalities are moved to other applications

4. Understanding application patterns

The figure below illustrates the key mainframe application patterns along with the workloads and possible use cases. It is important to note that some applications may have a combination of these patterns.

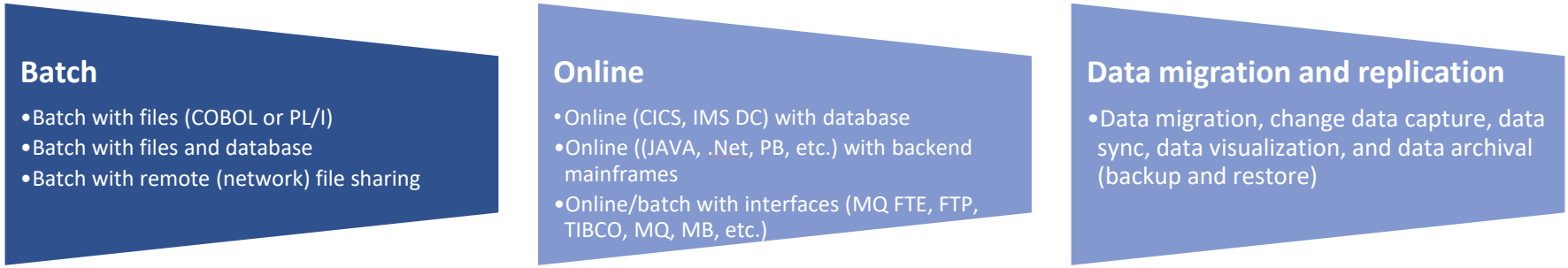


Fig 1: Type of mainframe application patterns

5. Batch with files and database

In this section, we will look migrating Batch applications to Azure

5.1. Architecture on mainframe: Batch with files and database

A typical batch workflow in mainframes involves components like:

- Jobs – There may be multiple jobs running on the mainframe. These jobs may have successor/predecessor dependency on each other

- Scheduler – This controls the job dependencies and triggers the flow based event/time/dependency.
- Data layers – This includes direct access storage device (DASD) and TAPE
- Integrations – This includes:
  - File transfer protocol (FTP) to receive and send files in batch
  - MQ to send/receive data from other external systems in batch
  - Email as a communication channel
  - Printer as a physical print medium

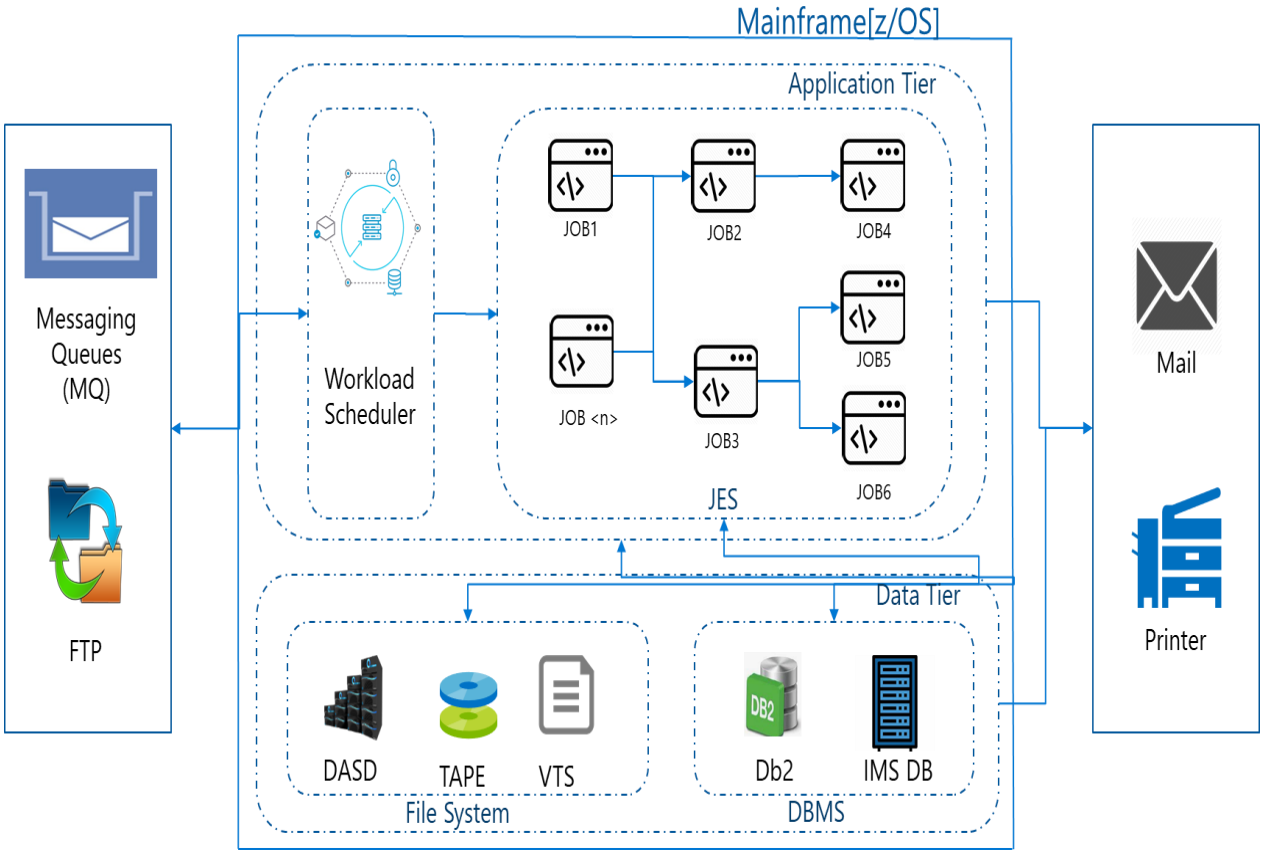


Fig 2: Mainframe pattern for batch with files

5.2 Azure target state architecture

Figure 3 depicts the reference architecture to migrate Batch applications to Azure.
The key steps in reference architecture for batch with files and database are:

- Reengineering the app tier – Application logic in mainframes can be reengineered using Azure service like Azure logic apps, Azure functions etc.
- Upload files to Azure storage – Files are uploaded to Azure Storage through Azure Data Factory

- Data load for processing – Mainframe data from DB2/IMS DB is loaded into Azure data stores
- Ingestion and orchestration – Azure Data Factory is used to manage the data in Azure Storage
- Data sync – Data is synchronized between the mainframe and Azure
- Download files – The files from Azure Storage can be downloaded to the mainframe storage in the appropriate format using services/utilities

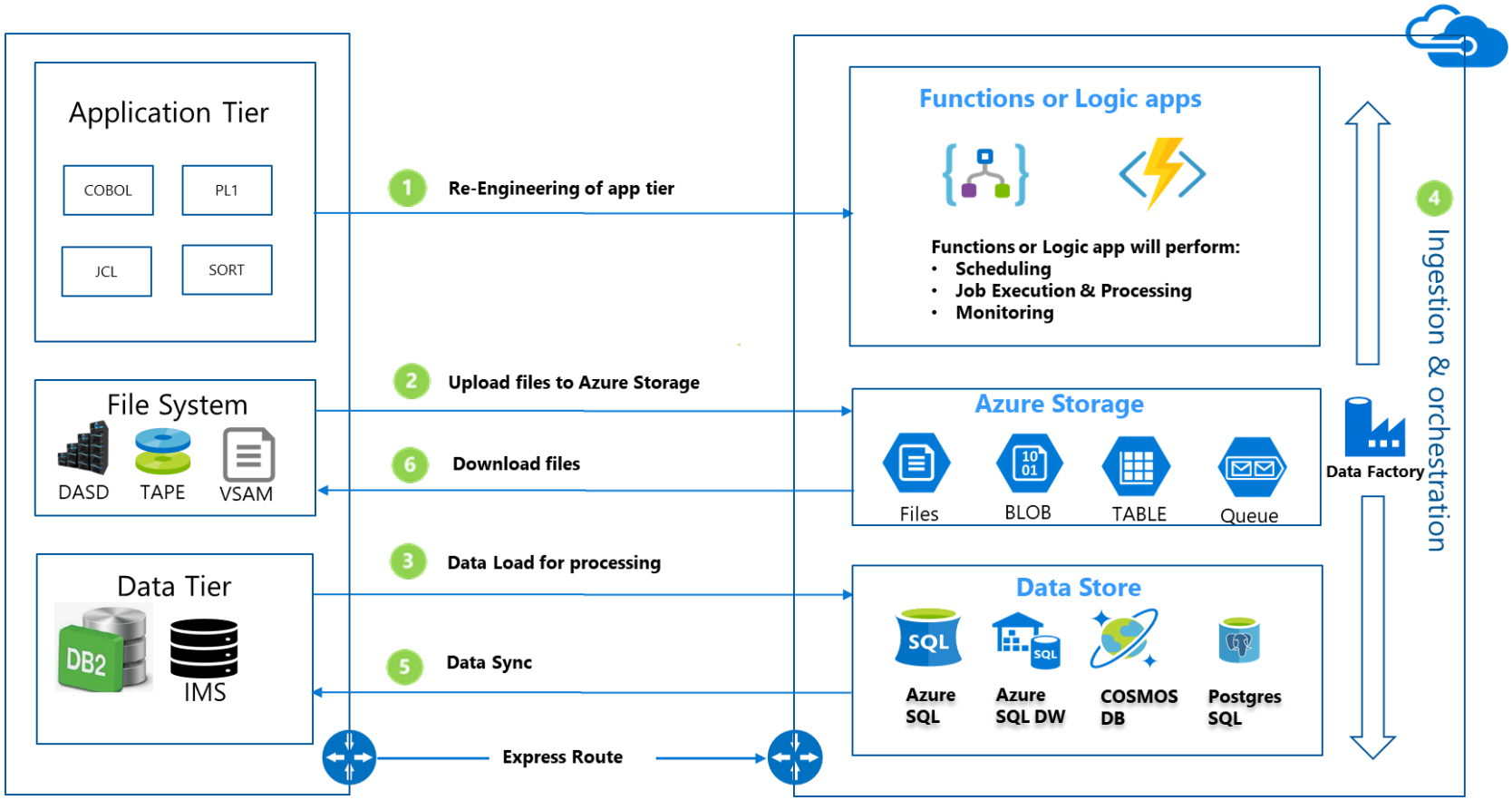


Fig 3: Reference architecture for batch with files and database

5.3 Data processing strategies for batch and real-time data

Figure 4 depicts reference architecture of Data processing strategies in Azure. In the diagram, Data moves through the pipeline through the following stages:

- **Data sources** – Data may reside in various on-premises source systems like sequential files, databases, applications, MQ, and logs
- **Ingestion** – Source data is ingested into Azure in Event Hub, Blob Storage and Azure Data Lake
- **Transformation** – Data is normalized and cleaned according to the transformation rules
- **Analysis** – Data is statistically analyzed to monitor logs and gain application insights
- **Data layers** – Data is loaded into Azure databases for consumption and business intelligence/visualization
- **Consumption** – Data is used to generate alerts, operational metrics, reporting, and visualizations

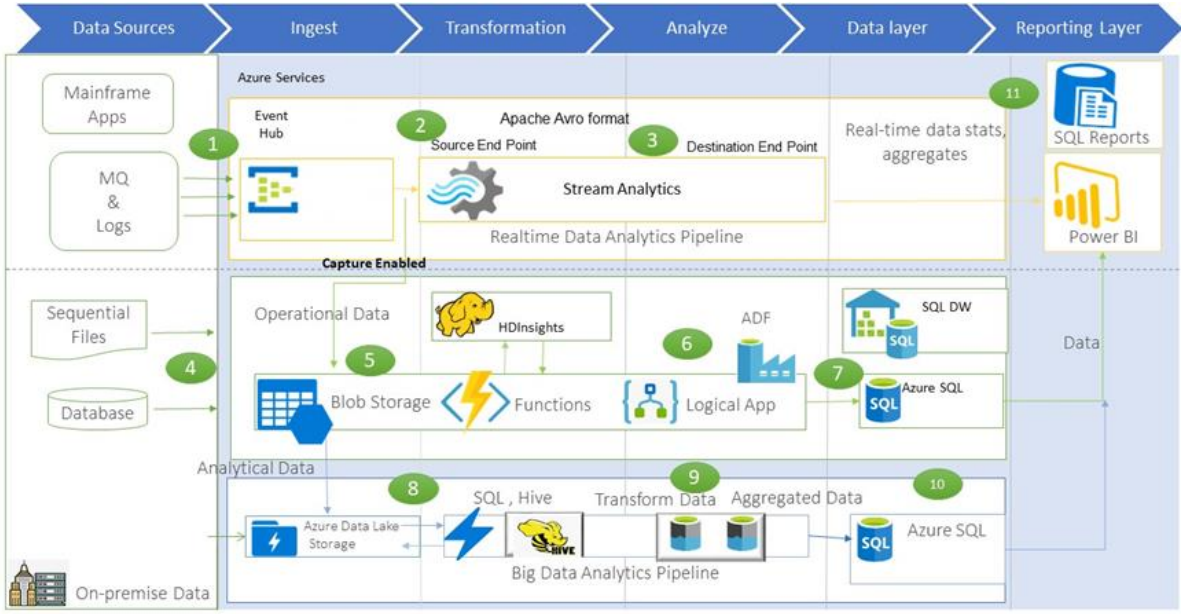


Fig 4: Typical architecture for real-time and batch data processing

Here is how Azure can be leveraged to migrate data within the data pipeline:

1. Data from mainframe applications, MQ and logs is ingested into Azure Event Hub. Event Grid, an event routing service, can also be used to respond to events from apps and services
2. Users can define source and destination points for stream analytics. This aggregates event data in near real-time
3. Real-time data along with data at rest is captured and ingested into the storage layer
4. Once the data is in Azure, data is processed using services like Azure logic apps, functions and HDInsight
5. Azure Data Factory is used to orchestrate the data pipeline, and schedule and monitor jobs
6. The target data is loaded into SQL DB or Azure SQL data warehouse
7. Analytical data is stored in the Azure data lake store
8. Raw data is transformed to aggregated data based on metrics and KPIs
9. The target data is loaded in SQL DB
10. Data visualization tools like Power BI are used for creating reports based on the processed data

6. Online applications

One of the main types of applications seen on mainframes are online applications. These are fundamental business transactions (like creating orders) that require higher processing speed.

This section examines how legacy online applications can be reengineered to the Azure platform.

6.1. Architecture on mainframe: Online with CICS/IMS

A typical online workflow in mainframes involves components such as:

- Transaction manager – A customer information control system (CICS) or information management system (IMS) may act as the transaction manager
- Transaction – The logical application that has the embedded application logic
- Data layers – Generally, database or VSAM files act as backend data layers
- Interface – Screens that users can access for business functionalities

- Integrations – This includes:
  - MQ or payloads that come into mainframe and trigger a transaction
  - Email as a communication channel
  - Printer as a physical print medium

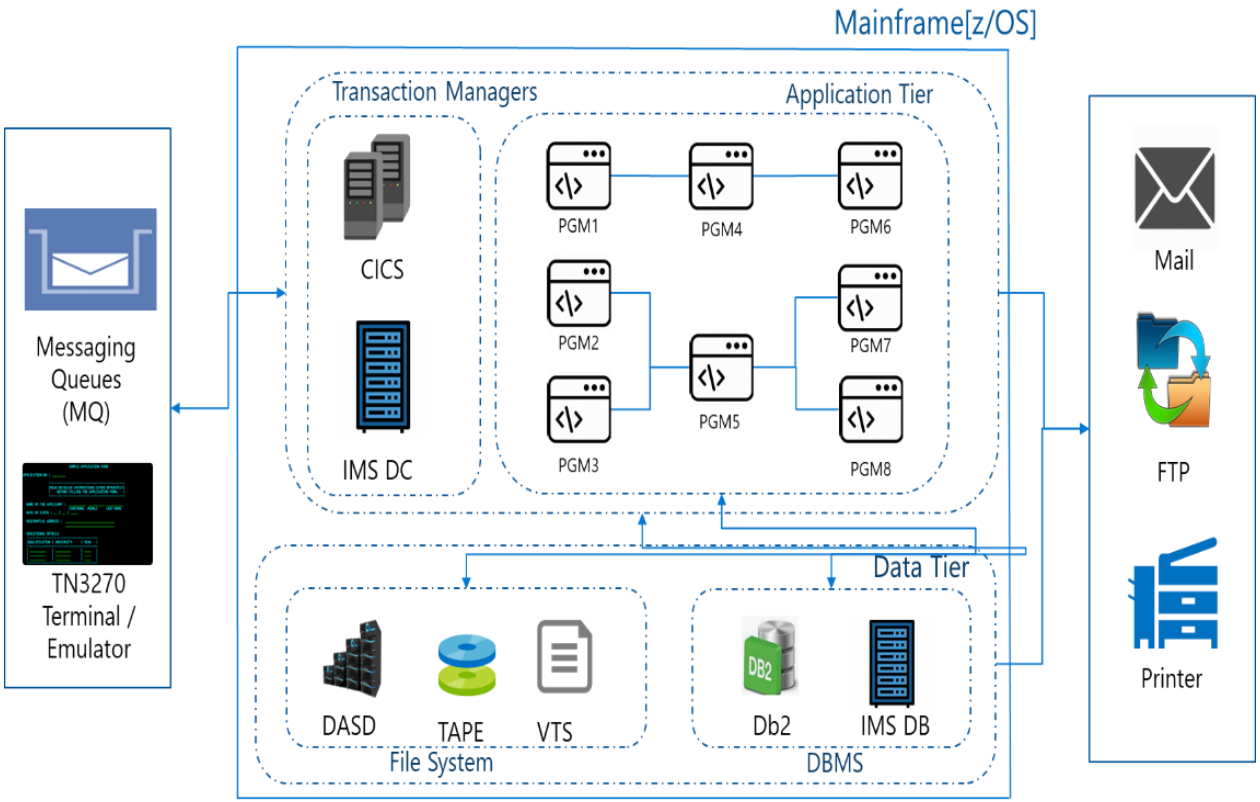


Fig 5: Mainframe online architecture

6.2. Azure architecture online – Rehost and re-compile

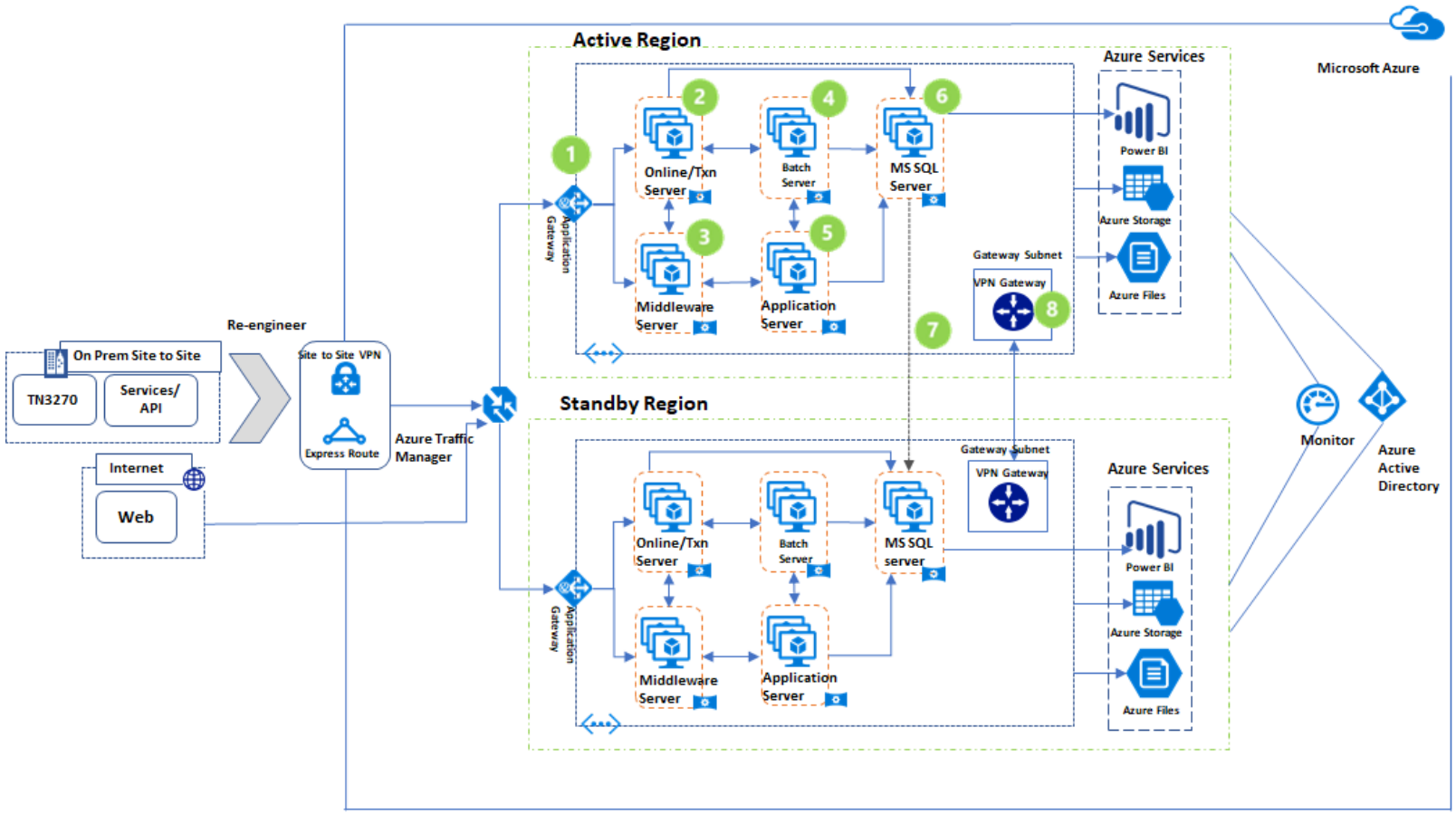
Mainframe applications can be rehosted without major structural changes using lift and shift process. There are third-party ISVs that can re-host platforms without modifying the existing source code.

Mainframe applications can also be recompiled as-is to cloud using cloud-native compilers or to cloud-native languages such as Java or .NET using a translator. There are third-party ISVs that recompile mainframe applications to the cloud. This helps modernize mainframe applications to new technologies in a scalable manner based on the requirements.

Rehosting and recompiling of mainframes is beyond the scope of this whitepaper. The following section examines how to reengineer mainframe application onto the Azure platform using Azure virtual machine (VM) and Azure Kubernetes Service (AKS).

6.3. Azure architecture online – Reengineering on VM

Mainframe applications can be reengineered using Azure VM whereby each server can be scaled up or down depending on the request. Applications are reengineered such that online, batch, application, middleware, and database servers reside in different VMs that can access the Azure services within VNET. The primary and secondary networks are connected through a VPN gateway that aides in disaster recovery.



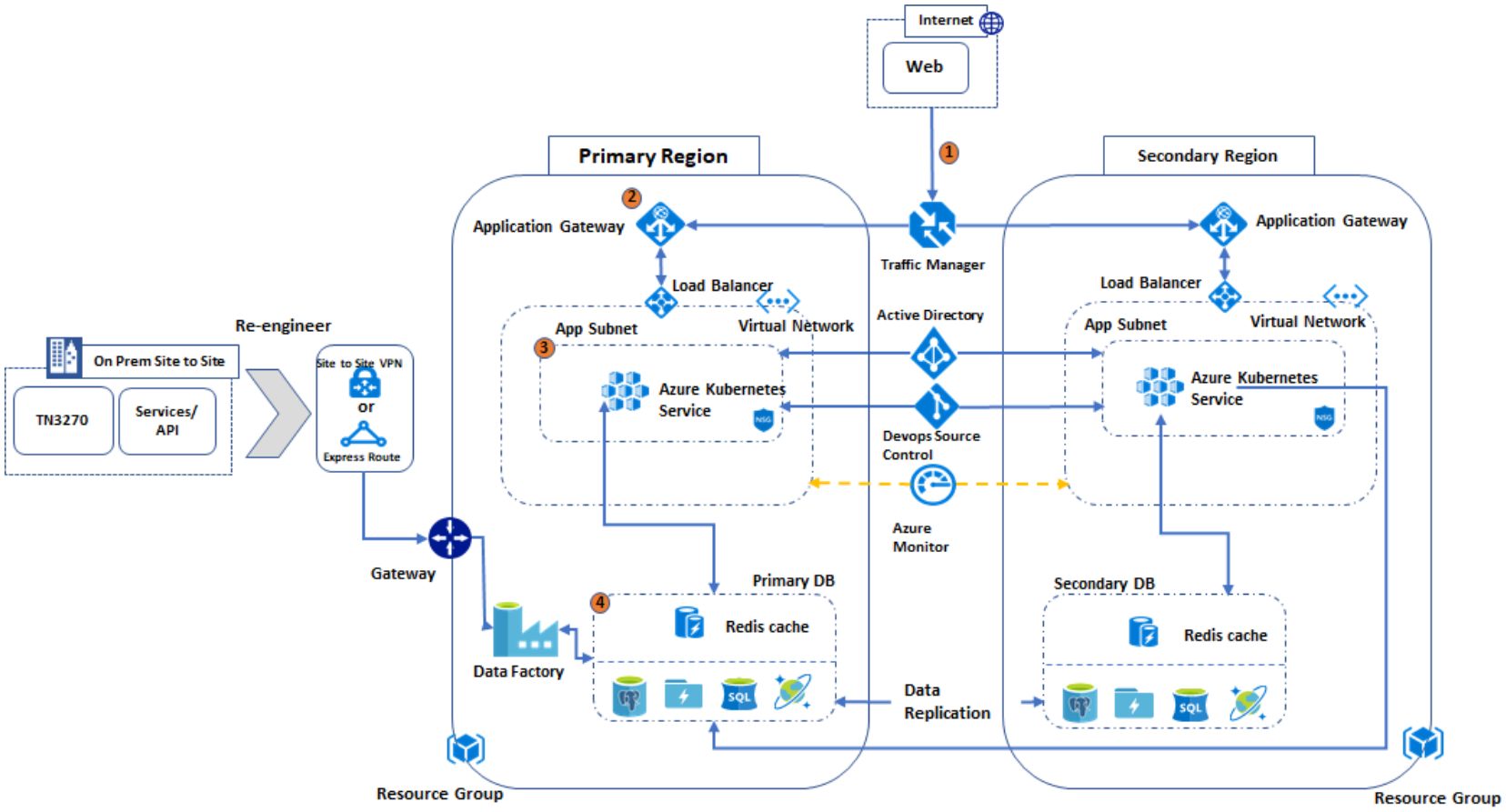
The reference architecture on Azure VM has the following components/services:

- An **application gateway** that manages traffic for the applications and routes traffic to the right destination server
- An **online/transaction server** that acts as the front-end to receive input transactions like the web server
- A **middleware server** that handles the message queues and routing mechanisms for the applications
- An **application server** that contains the business logic and processes data
- A **batch server** that contains the scheduler to process the jobs

- An **SQL DB server** that provides the database services and stores data
- A highly available **SQL server** that leverages synchronous commits for automatic failover
- A **gateway subnet** that connects the two regions through VPN

6.4. Azure architecture online: Reengineering on AKS

Mainframe applications can be re-compiled/translated to cloud-native languages and containerized using Azure Kubernetes Service (AKS). These containerized applications reduce the infrastructure cost and are scalable based on traffic. AKS can be used to connect to different database nodes and operating platforms.



The reference architecture on AKS has the following components:

- A **traffic manager** that directs the requests/traffic to specific endpoints, thereby balancing load optimally
- An **application gateway** that manages traffic for the applications and routes these to the destination server
- An **application subnet** that handles requests from the application gateway and interacts with databases. The AKS consists of applications and online/batch processes

- A **primary database** that has a Redis cache for caching database requests and supports primary databases in Azure
- **Azure Data Factory** is used to orchestrate the services and build data pipelines
- **Azure Kubernetes Service (AKS**) manages the hosted Kubernetes environment, making it quick and easy to deploy and manage containerized applications without container orchestration expertise. The fully managed AKS offers serverless Kubernetes, an integrated CI/CD experience and enterprise-grade security and governance

7. Data Migration

This section focuses on mainframe data tier migration and replication, different data migration use cases and third-party tools and accelerators to expedite the migration and adoption of Azure/SQL data platform.

Mainframe data workload migration has the following use cases for file and database systems:

1. One-time data migration for rehosting, reengineering and re-platforming scenarios
2. Data replication/sync for co-existence during migration phase and ease of DevOps
3. Change data capture for co-existence and hybrid scenarios

7.1. One-time data migration

This use case is for migrating data, schemas and programmable objects from mainframe database to the Azure data platform in a single and seamless manner. This migration approach includes 3 steps

- **Pre-migration** – The goal of pre-migration is to discover the existing data sources and assess details about the features that are being used. This will help gain a better understanding and plan for the migration. Once the databases are assessed and the discrepancies addressed, the next step is to execute the migration process.

- **Migration –** This step focuses on schema and data migration. It involves two steps:
  - Convert and publish the schema and programmable objects
  - Migrate data from the source to the target environments

  Once the migration has been successfully completed, there is a series of post-migration tasks to be executed to ensure that everything functions smoothly and efficiently.

- **Post-migration** – In this step, various post migration tasks should be run. These include:
  - Application remediation

- Data validation
- Functional testing
- Performance testing
- Optimization

7.2. Data replication and sync

All the steps outlined above in the one-time data migration approach are applicable here. However, this use case scenario is best suited for situations where the customer wants both environments to co-exist for some time. In such scenarios, apart from one-time data migration, we can also set up data replication/sync that synchronizes data between both environments in near real-time.

7.3. Change data capture

All the steps outlined in one-time data migration are applicable here. However, this use case is leveraged when the customer wants to maintain two copies of records in real time. In such scenarios, apart from one-time data migration, we also set up change data capture (CDC) processes that synchronize data between both the environments instantly. There are two types of CDC – unidirectional and bi-directional.

7.4. Data Migration Products

Various products can be used to streamline data migration depending on the use case scenario. The following table outlines Microsoft first-party and third-party products that help accelerate data migration use cases.

Products for database migration

Product name	Description
Microsoft SQL Server Migration Assistant (SSMA)	SSMA for DB2 is a tool to automate the migration from IBM DB2 database(s) to SQL Server, Azure SQL Database and Azure SQL Database Managed Instance. This product includes a client-based graphical user interface (GUI) application to manage the migration process.
Azure Data Factory (ADF)	ADF is a cloud data integration service that composes data storage, movement and processing services into automated data pipelines.

SQL Server Integration Services (SSIS)	SSIS is a component of the Microsoft SQL Server database software that can be used to perform a broad range of data migration tasks. SSIS is a platform for data integration and workflow applications.
Infosys Data Services Suite (iDSS)	iDSS is an Infosys data movement platform that helps migrate data from mainframe to Azure platforms.
Attunity (ISV)	Attunity Replicate empowers organizations to accelerate data replication, data ingestion and data streaming across a wide range of heterogeneous databases, data warehouses and big data platforms. It moves data easily, securely and efficiently with minimal operational impact.
Ispirer (ISV)	Ispirer MnMTK and Ispirer Services ensure easy, successful and automated database migration. It supports the conversion of most source and target technologies.

8. Joint assets for mainframe to Azure transformation

Microsoft and Infosys have jointly developed assets to assist mainframe to Azure transformation.

Some of these include:

Reference architecture	Microsoft and Infosys have identified multiple mainframe patterns and arrived at reference architectures to host the Mainframe workloads on Azure
Benefit realization framework	Azure TCO Calculator has been incorporated with pattern sizing (small, medium, and large) to provide the cost-benefit analysis for mainframe migration
Accelerated execution	Cloud templates have been created for all patterns to accelerate migration to Azure
Joint white papers	Microsoft and Infosys have published several white papers that provide strategic guidelines for enterprises that want to know more about mainframe to Azure modernization.

9. Conclusion

Organizations looking to migrate from mainframe to Azure must select the right architectural patterns which has been tested and proven through similar implementations. This Paper has addressed the reasons why Azure is ideal end state for mainframe migration. When migrating to cloud, it is important to understand how applications in mainframes map to their equivalent architecture on Azure. This whitepaper is an aimed at providing customers with references and aid their migration journey from the rich experience that Microsoft and Infosys have gathered from their past implementation experiences.

About the authors

Microsoft contributors:

Ashish Khandelwal: Senior Engineering Architect, Mainframe Modernization
Mukesh Kumar: Engineering Lead for Mainframe and Legacy Modernization
Pramod Vasanth: Principal Cloud Solutions Architect

Infosys contributors:

Jaydip Sanyal: Associate Vice President and Head, Legacy Modernization Practice
Rajesh Kumar Vinukonda: Technology Architect, Legacy Modernization Practice.
Ajay Jeswani: Principal Consultant, Legacy Modernization Practice
Ramanath Nayak: Lead Consultant, Legacy Modernization Practice

References

https://datamigration.microsoft.com/
https://azure.microsoft.com/en-in/services/
https://docs.microsoft.com/en-in/azure/mysql/partners-migration-mysql
https://www.infosys.com/modernization/
https://www.infosys.com/services/cloud/offerings/Pages/microsoft-azure.aspx
https://www.infosys.com/newsroom/press-releases/Pages/azure-expert-managed-services-provider.aspx