Washington Gas Light Company

eServices Portals Re-platforming

**Solution Blueprint**

**Template Version**

|  |  |
| --- | --- |
| **Version** | **1.1** |
| **Template Release Date** | **October 25, 2024** |

Document Owner

Document Author

The primary contact for questions regarding this document is (Ravi Soni).

Disclaimer

The information contained in this document is the proprietary and exclusive property of AltaGas Ltd. except as otherwise indicated. No part of this document, in whole or in part, may be reproduced, stored, transmitted or used without the prior permission of AltaGas Ltd.

Privacy Information

This document may contain information of a confidential and sensitive nature. This information should not be distributed to any stakeholders other than those involved in this project eService Portals or without the permission of <Lindy Ybarra>.

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author/Editor | Revision Description | Date |
| 1.0 | Ravi Soni | Initial draft | November 06 ,2024 |
| 1.1 | Ravi Soni | Initial draft with review updates | November 26 ,2024 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Approvals

We the undersigned, confirm that we have reviewed the content within this document and our signatures signify approval of the OCM and Communications Strategy for <insert Project Name>.

*Check the Artifact RACI to identify the appropriate roles for developing, contributing, reviewing, and approving this document. Verify this artifact is necessary based on the output of the Artifact Tailoring Guide in the PDM Tailoring tool.*

|  |  |  |
| --- | --- | --- |
| Name & Title |  |  |
| **Role** | **Signature** | **Date** |
|  |  |  |

The following reviewers have reviewed the document for completion and accuracy:

|  |  |
| --- | --- |
| **Name** | **Title** |
| *<name>* | *<Title* |
|  |  |
|  |  |

Table of Contents

[1 Introduction 4](#_Toc183427962)

[1.1 Purpose 4](#_Toc183427963)

[1.2 Scope 4](#_Toc183427964)

[2 Target Architecture 4](#_Toc183427965)

[2.1 Front-End Application Architecture 4](#_Toc183427966)

[2.2 Back-End Application Architecture 6](#_Toc183427967)

[2.3 Infrastructure Architecture 7](#_Toc183427968)

[3 Data Migration Approach 12](#_Toc183427969)

# Introduction

## Purpose

The aim of this document is to define the high-level solution blueprint for the re-platforming of WGL eService portal and the detailed design document will be shared during the design phase.

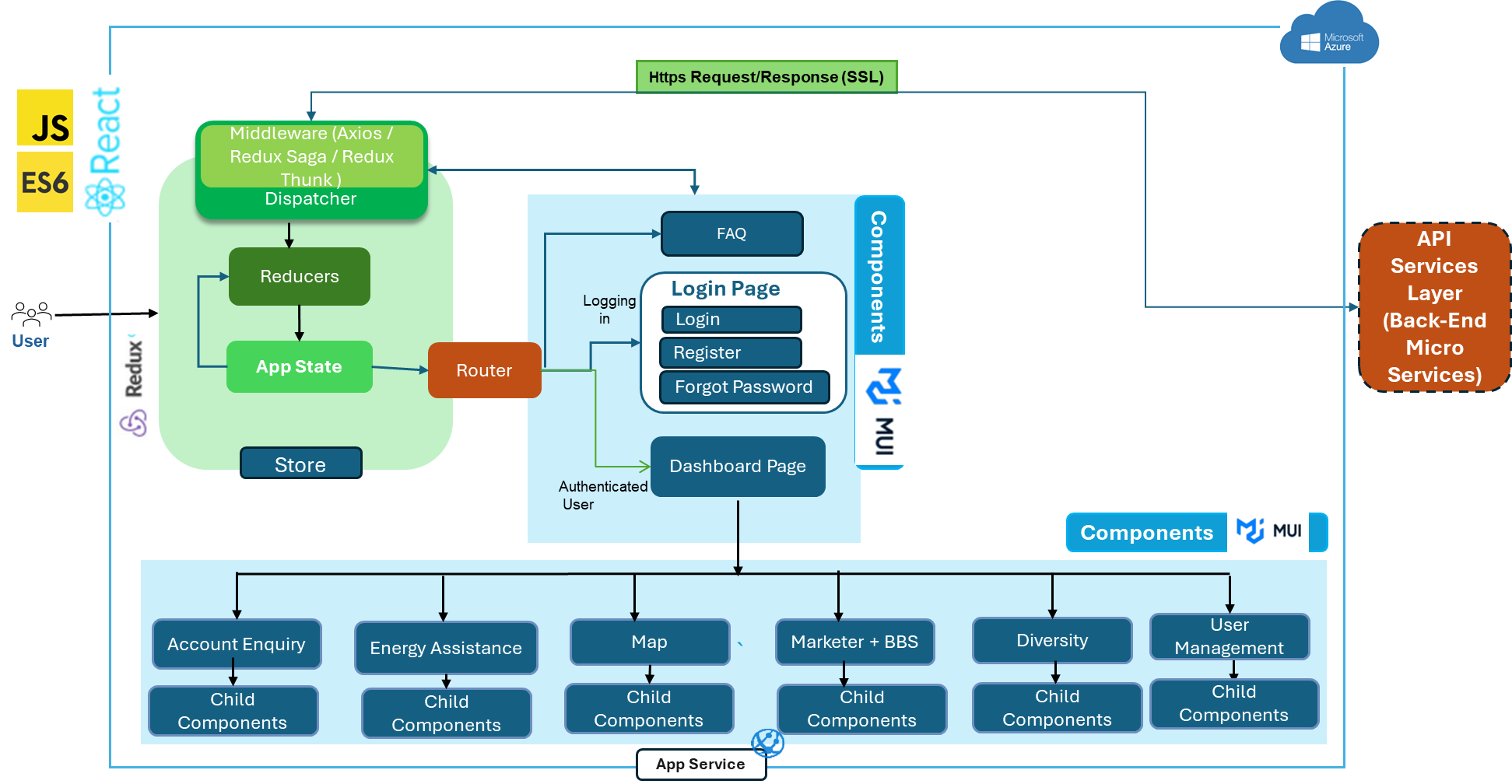
## Scope

Re-platforming the WGL eService portal using with the latest technology and to deploy the eService portal in the Azure cloud.

# Target Architecture

## Front-End Application Architecture

Front-end application architecture is the blueprint for a user interface that defines how its components interact with each other, and how the application performs and functions. It's a set of techniques and patterns that guide the design and development of an application.

**

* ***React***

*React.js is a popular JavaScript library for building user interfaces, especially single-page applications where you need a fast, interactive user experience. Developed and maintained by Facebook, react is known for its component-based architecture, which makes it easier to build and manage complex UIs.*

* ***Router***

*It handles routing in single-page applications, React Router provides declarative routing that allows you to navigate between different views or pages.*

* **State**

*State is a way for components to keep track of information between renders. Unlike props, state is managed within the component and can be changed with methods like setState in class components or the useState hook in functional components.*

* **Components**

*React applications are built using components, which are reusable pieces of code that return a React element (describing what should appear on the screen). Components can be either class-based or function-based.*

* **App Service**

*App service is the utility class that handles application-level logic or provides an interface to interact with external services, APIs, or other parts of the application. These services help manage and encapsulate functionalities like API calls, authentication, data manipulation, or any other business logic that is not directly related to the UI.*

* **Redux**

*Redux is a popular state management library often used with React, though it can be used with any JavaScript framework or library. It helps manage the state of an application in a predictable way, especially in large applications with complex state requirements.*

* **Store**

*The central repository for the application's state. There is only one store in a Redux application, and it holds the entire state tree.*

* **Dispatcher**

*A method used to send actions to the store. When an action is dispatched, Redux will pass it to the reducer to calculate the new state.*

* **Redux Thunk**

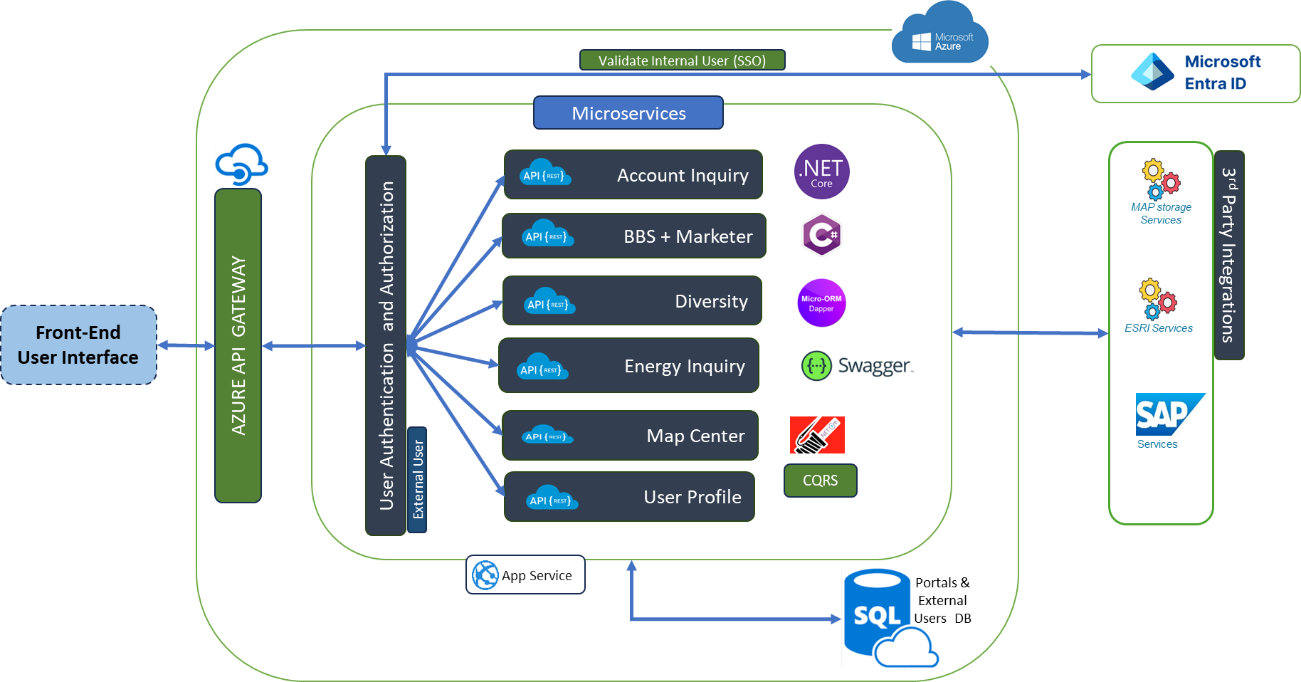
*It is a middleware for managing asynchronous operations in Redux applications. It allows you to write action creators that return functions instead of plain action objects. This is useful for handling side effects like data fetching, which cannot be done directly within reducers because reducers are pure functions and should not contain side effects.*

* **MUI**

*Material-UI (now known as MUI) is a popular React component library that implements Google's Material Design system. It provides a set of customizable and accessible components that follow Material Design principles, which can help speed up development and ensure a consistent look and feel across your application.*

## Back-End Application Architecture

*Back-end application architecture is the blueprint for a business logic that act as a separate service for each feature that defines how its components interact with each other, and how the application performs and functions. It's a set of techniques and patterns that guide the design and development of an application.*

**

* ***Azure API Gateway***

*The API Management gateway (also called data plane or runtime) is the service component that's responsible for proxying API requests, applying policies, and collecting telemetry. Specifically, the gateway: Acts as a facade to backend services by accepting API calls and routing them to appropriate backends.*

* ***MS Entra ID***

*It is a cloud-based identity and access management (IAM) provider that develops a unified access management (UAM) platform for enterprise-level businesses and organizations.*

* ***Internal User Authentication (MS Entra ID)***

*Authentication is the process of verifying a user's identity before they are given access to a system, network, or resource. It's often a prerequisite for accessing information systems. Internal user will be authenticated using MS Entra ID.*

* ***External User Authentication (MS SQL)***

*MS SQL will be used to store external user credentials and will be used to authenticate the external users.*

* ***Authorization (Internal and External users)***

*Authorization is the process by which a server determines if the client has permission to use a resource or access a file. Both internal and external users’ authorization details are maintained in MS SQL Database.*

* ***Microservices***

*Microservices are a software development approach that breaks an application into smaller, independent services that work together to achieve a business goal. Each service is responsible for achieving a domain specific functionality and communicates with other services using an API.*

* ***Dot net Core (8.0)***

*Dot NET Core generally offers better performance for web applications due to its optimized runtime and modular architecture. This can lead to faster response times and improved scalability for web services and applications.*

* ***C#***

*C# (C-Sharp) is a modern, open-source, object-oriented programming language used to build applications for a variety of platforms. C# is known for its type safety, generics, pattern matching, async, and records features.*

* ***ORM Framework (Dapper)***

*Dapper is an open-source object-relational mapping (ORM) library for .NET and .NET Core applications. The library allows developers quickly and easily access data from databases without the need to write tedious code.*

* ***Swagger***

*Swagger is an Open-Source set of rules, specifications, and tools for developing and describing RESTful APIs. The Swagger framework allows developers to create interactive, machine and human-readable API documentation.*

* ***Serilog***

It was built for structured logging, which allows developers to record custom object properties and output logs as JSON. Serilog supports a variety of logging destinations, including the console, files, and logging services like Datadog.

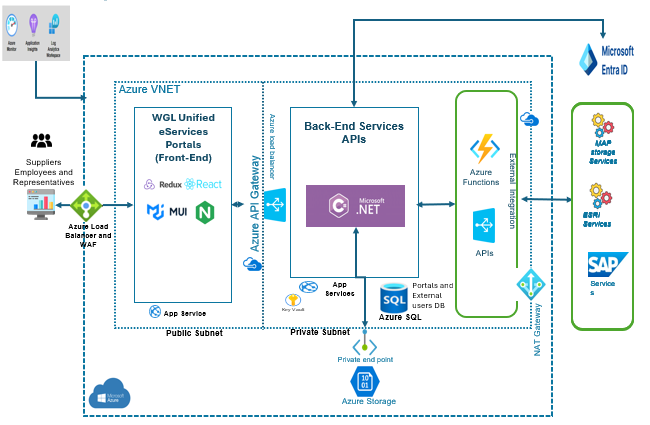
* ***Azure App Service***

*Azure App Service provides a framework for developing and running apps in the cloud. PaaS providers host and maintain the platform's servers, networks, storage, and other computing resources.*

* ***MS SQL Server***

*Microsoft SQL Server is a relational database management system. As a database server that stores and retrieves data as requested by other software applications on the same computer or a remote computer using the client-server model. We will use this MS SQL Database to store portals related data, external user credentials and internal & external users authorization details (Role Based Access Details).*

## Infrastructure Architecture



*Server details for front end, back end and database, network/firewall,*

*Above architecture outlines a highly secure, scalable, and monitored infrastructure with Azure services like App Service, Azure API Gateway, Azure SQL, and Azure Functions. It integrates strong security measures (Azure Key Vault, Private Link, NSGs, Azure Firewall), utilizes monitoring and insights tools, and incorporates high availability with geo-replication and regular backups. This structure effectively balances user accessibility, backend security, and seamless integration with SAP and other external systems using MS Entra ID for authentication.*

***Frontend:***

*The front-end is hosted on Azure's App Service, which manages and scales web apps automatically.*

* + ***Environment Setup:***

*1.Dev/QA: Premium (Premium V2) App Service Plan for testing scaling and functionality.*

*2.UAT/Prod: Premium (Premium V3) plan with autoscaling enabled to handle production traffic efficiently.*

*3. DR:* *Premium (Premium V3) plan with geographically redundant instance is set up in a secondary Azure region, for high availability.*

* + *The front-end is in a public subnet, accessible over the internet via the* ***Azure Load Balancer****, which distributes traffic to the front-end instances.*
  + ***HTTPS-only*** *access is enforced to secure communication between users and the application.*
  + ***Application Gateway with Web Application Firewall (WAF)*** *protects against security vulnerabilities like SQL injections and cross-site scripting.*
  + ***Azure Key Vault*** *is integrated to store and manage secrets, such as API keys and certificates securely.*

***Back-End Services (APIs):***

* + ***App Service****: The back-end APIs are also hosted on Azure App Service, ensuring scalability and managed infrastructure.*
  + ***Environment Setup:***

*1.Dev/QA: Use a lower-tier plan to test functionality and API performance.*

*2.UAT/Prod: Premium App Service Plan with auto-scaling and geo-distribution for high availability.*

*3. DR:* *Premium (Premium V3) plan with geographically redundant instance is set up in a secondary Azure region, for high availability.*

* + ***.NET APIs****: The backend is built using Microsoft .NET and exposes APIs that can be consumed by the front-end or other services.*
  + *The back-end API service has a* ***private endpoint****, ensuring secure communication with the database and other Azure resources.*
  + ***SQL Database****: Azure SQL Database stores the application’s backend data and is accessed by the back-end API.*
  + ***Firewall rules*** *are applied to restrict unauthorized IP addresses.*
  + ***Private Link*** *ensures secure connections between the App Service and Azure SQL, without exposing it to the public internet.*
  + ***Transparent Data Encryption (TDE)*** *is used to encrypt database data at rest.*
  + ***Key Vault*** *stores API keys and sensitive database credentials.*

***Authentication:***

* ***Internal Users:*** *Internal Users (like; employees, suppliers, and representatives) are authenticate before accessing services. MS Entra ID is integrated to for internal users authentication and authorization, enabling Single Sign-On (SSO).*
* ***External Users:*** *External Users (like; customers) are authenticated against DB and authorization is also maintained in the DB for easy customization.*

***Azure SQL Database:***

* + ***Model/License:***

1. *Dev/QA (Azure SQL Database): Use the Standard Tier for lower costs with Basic DTUs or vCore pricing.*
2. *UAT (Azure SQL Managed Instance): Use Business-Critical Tier for enhanced performance, low latency, high availability, and replication.*
3. *Prod (Azure SQL Managed Instance): Use the Geo-Replication Business-Critical Tier for enhanced performance, low latency, high availability, and replication.*
4. *DR (Replica DB of Prod): Azure SQL Database provides active geo-replication. This creates secondary replicas in a different region that are automatically synchronized with the primary database. These secondary replicas serve as your DR solution without needing a "separate" database configuration.*
   * ***Security:***
5. *Private endpoints restrict public access.*
6. *Transparent Data Encryption (TDE) enabled.*
7. *Managed identity used for secure access with Azure Key Vault managing sensitive credentials.*
8. *Dev/QA: May use soft firewall rules for easier access, but UAT/Prod should have stricter rules ensuring only whitelisted IPs can connect with auto-failover options.*

***Authentication:***

* *MS Entra ID is integrated to handle authentication, enabling Single Sign-On (SSO) for Internal users. It connects to various services like SAP and external systems for authentication and identity management.*
* *Internal Users (like employees, suppliers, and representatives) authenticate via MS Entra ID before accessing services.*

***Integration & Middleware:***

* ***Azure API Gateway****:*
  + *This component acts as an intermediary, routing traffic between the front-end and the back-end services securely. It ensures security, load balancing, and API management.*
  + ***SAP Services*** *are connected, likely for business process integration (e.g., handling financials, HR, and supply chain data).*

***Azure Functions:***

* ***Azure Functions****: A serverless compute service is employed to execute backend logic, event-driven tasks, or scheduled tasks without managing server infrastructure, enabling highly scalable event-driven architecture.*
  + *Functions handle specific business logic, such as email notifications, database cleanup jobs, or data processing workflows.*
  + *Artifacts: Serverless code to handle event-driven tasks such as email notifications, database cleanup, data processing, or integration with third-party services.*
  + ***Environment Setup:***

*1.Dev/QA: Lower resource consumption (consumption plan), used for testing specific business logic and scheduled tasks.*

*2.UAT/Prod: Premium plan for higher throughput and long-running functions. Managed identity should be used to securely access Azure resources like Azure Storage or SQL databases.*

* + ***Managed Identity*** *is used for secure access to other Azure resources, such as storage and databases.*
  + *Functions are invoked over* ***HTTPS****, ensuring secure communication.*

***Storage:***

* + *Provides* ***blob storage*** *or file storage for handling large datasets, files, and backups.*
  + *The* ***private endpoint*** *for Azure Storage ensures secure access from within the Azure virtual network (VNet).*
  + *Use of* ***RA-GRS (Read-Access Geo-Redundant Storage)*** *enables backup and data availability across regions for disaster recovery.*

***Monitoring:***

* ***Azure Monitor and Application Insights****:*
  + *These services are used for tracking performance, logging, and diagnostics across the infrastructure, providing a centralized view for monitoring health and performance.*
  + ***Alerts*** *and* ***dashboards*** *are set up to notify admins of performance issues or failures in the system.*
* ***Log Analytics Workspace****:*
  + *Consolidates logs and performance data, offering query and alerting capabilities for deeper insights into application and infrastructure behavior.*

***Security:***

* ***Azure VPC (Virtual Private Cloud)****:*
  + *The VPC isolates and secures the resources in a private network, ensuring that backend services are protected from public exposure.*
* ***NAT Gateway****:*
  + *Manages outbound internet traffic from private subnets, ensuring security by not exposing internal resources directly.*
* ***Network Security Groups (NSGs)****:*
  + *NSGs are configured to control traffic flow within the Azure VPC, allowing only authorized traffic between subnets.*
* ***Azure Firewall****:*
  + *Provides centralized firewall protection for the entire VPC, managing and filtering both inbound and outbound traffic.*
* ***DDoS Protection****:*
  + *Ensures resilience against distributed denial-of-service (DDoS) attacks, safeguarding the public-facing services.*

***High Availability (HA) & Backup:***

* ***Azure SQL****:*
  + ***Active Geo-Replication*** *is configured for high availability, ensuring continuous database operations even in case of regional outages.*
  + ***Point-in-time restore,*** *and* ***long-term backup retention*** *are enabled for the database to recover from accidental deletions or data corruption.*
* ***Azure App Services****:*
  + *Deployed in multiple regions for high availability, with autoscaling configured to manage traffic loads during peak times.*
  + ***App Service Backups*** *are scheduled regularly to ensure recovery options are available in case of system failures.*

# Data Migration Approach

*This document outlines the best practices, procedures, and steps to successfully migrate the 3 on-premises SQL databases to Azure database (VM/SQL DB/Managed Instance). The guidelines ensure minimal data loss, high security, and proper validation throughout the migration process.*

***Scope of Migration***

* *Data Sources: The primary data sources for migration are on-premises SQL Server databases that store structured, relational data.*

***Target Environment Setup and Configuration on Azure Platform***

*Given the four distinct environments—Dev, QA, UAT, and Prod—the following approach outlines the specific migration strategy for each:*

* *This migration approach details how data will be migrated from the on-premises environments to Azure, with on-premises Dev and QA moving to Azure Dev and QA, and on-premises UAT and Prod transitioning to Azure UAT and Prod.*

***Development and QA Environments***

* *Environment Setup: Deploy using* ***Azure SQL Database*** *for cost-effective resource allocation in non-production environments.*
* *Migration Approach: Use Azure DMS for initial data migration and DDL schema updates.*
* *Performance Scaling: Utilize auto-scaling in non-production environments based on workload needs, monitored by Azure Monitor.*

***UAT and Production Environments***

* *Environment Setup: Use* ***Azure SQL Managed Instance*** *for production workloads to take advantage of near-full SQL Server compatibility and VNet support.*
* *Monitoring and Scaling: Configure SQL Insights for detailed performance tracking in the UAT and production environments.*

***Database Consolidation Strategy***

* *Rationale: Consolidate the three databases into a single schema where possible to reduce complexity.*
* *Implementation: Implement the consolidation in stages to reduce load on the database during peak times and monitor closely with Azure Monitor for potential issues.*

***Migration tool selection:***

1. *Migration Tools: We identified Azure* ***Database Migration Service (DMS)*** *tool for its minimal-downtime online migration capabilities.*

*Azure Database Migration Service (DMS) provides minimal downtime migration by continuously replicating data from your source SQL Server to Azure SQL Database or Managed Instance. DMS handles schema migration, continuous data sync, and cutover.*

* *Cost: The DMS service is free; however, Azure resources (compute, storage) used during migration incur charges.*
* *Storage Cost: Azure Blob Storage may be used temporarily for backups or staging data.*
* *Compute Cost: Based on Cores provisioned for Azure SQL.*

*Online migration enables continuous synchronization between the source and target databases, minimizing downtime, while offline migration is suitable for scenarios where downtime is less critical.*

* ***Self-Hosted Integration Runtime (SHIR)****: Installed on-premises to enable secure data transfer and communication between the on-prem SQL Server and the Azure cloud. SHIR handles the orchestration of data movement.*
* ***Azure Storage****: Used to temporarily store database backup files during migration, ensuring flexibility and scalability, especially for large datasets.*

***Security and Audit Trail***

*Security is paramount in the migration process. We implement several layers of security, including:*

* ***Azure Virtual Network (VNet)****: Ensures that data transfers between the on-premises infrastructure and Azure happen within a secure, encrypted network.*
* ***Private Link****: Provides an additional layer of security, ensuring that data is accessed privately over Azure's network infrastructure.*
* ***Azure Active Directory (AAD):*** *Manages access control, ensuring that only authorized users and services can access the databases during and after the migration.*
* ***Audit Trail****: We use SQL Auditing and Advanced Threat Protection to monitor for anomalies, log all critical database actions, and ensure compliance with security regulations.*

***Migration Validation***

*Post-migration, we ensure the data's accuracy and integrity through validation checks. This process includes:*

* ***Schema Validation****: Ensures that the migrated schema matches the source schema, maintaining referential integrity.*
* ***Data Validation****: Compares record counts, row data, and relationships between source and destination databases to confirm completeness.*
* ***Performance Monitoring****: After migration, Azure Monitor is used to track performance metrics and database health, ensuring that the system is functioning optimally in the Azure environment.*

*A diagram of a software flow

Description automatically generated with medium confidence*

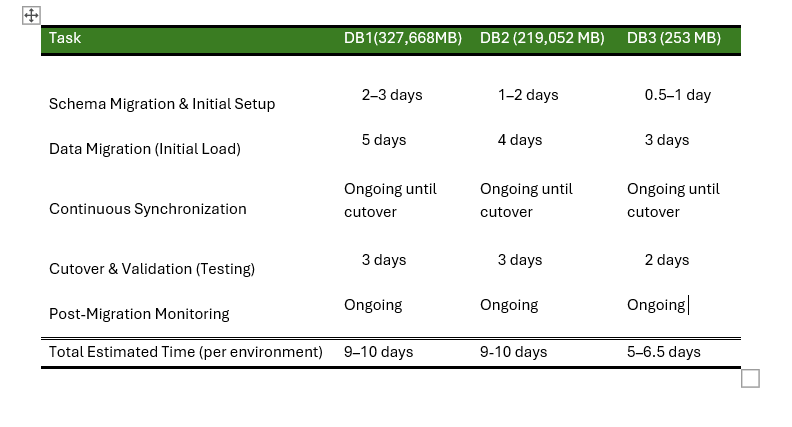
* ***On-Premises SQL Server***
  + *Source System: The SQL Server database in the current on-premises environment.*
  + *Data Storage: Hosting structured relational data with potentially high loads.*
* ***Azure Database Migration Service (DMS)***
  + *Migration Tool: The core tool used to facilitate the migration. Azure DMS connects to the on-prem SQL Server and manages the schema and data transfer.*
  + *Online Migration: Supports online migration with continuous data synchronization during the cutover phase to minimize downtime.*
* ***Azure Blob Storage (Optional)***
  + *Backup Storage: Used to temporarily store backup data during the migration, especially for large datasets or for archival purposes.*
* ***Target System (Azure SQL Database)***
  + *Destination Database: The Azure SQL Database or Azure SQL Managed Instance that will host the migrated data. This provides cloud-native scaling and performance benefits.*
  + *vCore-based Model: Provides flexible scaling options for compute and storage resources.*
* ***Azure Virtual Network (VNet)***
  + *Secure Communication: A virtual network connecting the on-premises infrastructure with Azure, ensuring secure data transfer.*
  + *Private Link: Additional layer for secure data access and communication.*
* ***Azure Active Directory (AAD)***
  + *Access Management: Manages identity and access to the Azure SQL Database. Ensures role-based access control for both the migration process and post-migration.*
* ***Monitoring and Security Services***
  + *Azure Monitor: Used for performance monitoring during and after the migration to track data flow, server performance, and database health.*
  + *SQL Auditing and Advanced Threat Protection: Ensures security compliance and detects any anomalies during the migration process.*

***Estimation Assumptions***

* *Database Migration Time: The online migration process will vary depending on database size. A larger database (Service Portal DB) may take longer than smaller ones.*
* *Environment Setup and Configuration: Setting up each environment will involve schema migration, configuration, and verification.*
* *Testing and Validation: Each environment requires thorough testing and validation, especially the production environment, which may also include user acceptance testing.*
* *Monitoring and Modifications: Post-migration monitoring and adjustments might be necessary to ensure optimal performance and data integrity.*

*To account for potential delays, I've doubled the estimated time for a more conservative approach.*

***Detailed Time Estimation Table***



***Time Estimation per Environment (Including Buffer)***

|  |  |
| --- | --- |
| **Environment** | **Total Days** |
| **Dev** | 15–20 days |
| **QA** | 15–20days |
| **UAT** | 15–20days |
| **Prod** | 15–20days |
| **DR** | 15–20days |
| **Grand Total** | 85–100 days |

***Summary***

* *Base Estimate: Approximately 60–77 days for all environments.*
* *Testing & Monitoring: Included in each environment's estimate, with additional focus on the production environment.*

*This estimation provides a conservative timeline, allowing for thorough testing and adjustments at each stage of the migration process across all four environments.*

***Approach Summary***

*This architecture and technology-driven approach ensures that the migration of SQL Server databases from on-premises environments to Azure SQL environments is secure, efficient, and validated. By leveraging Azure Database Migration Service (DMS), we minimize downtime, ensure data integrity, and adhere to best practices in cloud migration.*