**Technical Design Document: SQL On-Premises Database Migration to Azure Blob Storage and Dataverse with Key Vault Integration Using Self-Hosted Integration Runtime (SHIR) Private Endpoint and Express Route/Private Peering**

**Table of Contents**

1. **Introduction**
2. **High-Level Architecture**
3. **Prerequisites**
4. **Data Migration to Azure Blob Storage**
5. **Data Transformation**
6. **Azure Key Vault Integration**
7. **Data Migration to Dataverse**
8. **Security Considerations**
9. **Monitoring and Logging**
10. **Conclusion**

**1. Introduction**

This technical design document outlines the process for migrating an SQL on-premises database to Azure Blob Storage and subsequently to Dataverse. The migration process incorporates the use of a Self-Hosted Integration Runtime (SHIR) with a private endpoint for secure data movement and Express Route or private peering for private network connectivity between on-premises SQL Server and Azure services. Additionally, Azure Key Vault is integrated for secure credential management.

**2. High-Level Architecture**

**Components:**

* **On-Premises SQL Server:** The source SQL database hosted on-premises.
* **Self-Hosted Integration Runtime (SHIR):** Installed on an on-premises machine, SHIR facilitates data movement using a private endpoint.
* **Express Route or Private Peering:** Provides a private and dedicated connection between the on-premises network and Azure.
* **Azure Blob Storage:** Used as an intermediate storage location for data during migration.
* **Azure Data Factory:** Manages data pipelines, including data movement and transformation.
* **Azure Key Vault:** Securely stores client credentials (client ID and secrets).
* **Dataverse:** A Microsoft Power Platform service used for storing and managing transformed data.

**3. Prerequisites**

Before proceeding, ensure the following prerequisites are met:

* Access to an Azure subscription.
* An Azure Virtual Network (VNet) set up for private connectivity.
* A Self-Hosted Integration Runtime (SHIR) installed and configured on an on-premises machine.
* Azure Blob Storage account set up.
* Azure Data Factory instance created.
* Azure Key Vault instance configured for storing client credentials.
* Dataverse environment set up.

**4. Data Migration to Azure Blob Storage**

Utilize Azure Data Factory for data migration from the on-premises SQL Server to Azure Blob Storage using a private endpoint:

* Configure Linked Services:
  + **SQL Server Linked Service:** Set up a connection to the on-premises SQL Server.
  + **Azure Blob Storage Linked Service:** Configure connection details for your Azure Blob Storage account.
* Create a Pipeline:
  + Employ the Copy Data activity to move data from the on-premises SQL database to Azure Blob Storage.
  + Ensure data is copied securely over the private network connection.

**5. Data Transformation**

Perform data transformations as needed, which may include data cleansing, enrichment, or format changes. Store the transformed data within Azure Blob Storage in a suitable format.

**6. Azure Key Vault Integration**

Securely store client credentials (client ID and secrets) in Azure Key Vault:

* Create an Azure Key Vault instance.
* Add secrets for client ID and secrets.
* Configure appropriate access policies to allow Azure Data Factory and SHIR to access the Key Vault securely.

**7. Data Migration to Dataverse**

Utilize Azure Data Factory or Power Platform tools to migrate data from Azure Blob Storage to Dataverse:

* Create tables/entities in Dataverse that match the structure of the transformed data.
* Use Power Query or Power Automate to load data from Azure Blob Storage into Dataverse.
* Configure authentication in Power Query or Power Automate to use the credentials stored in Azure Key Vault.

**8. Security Considerations**

* Ensure proper firewall rules and security configurations on the on-premises SQL Server, SHIR machine, and Azure services to safeguard data in transit and at rest.
* Implement Role-Based Access Control (RBAC) and Azure Policy to control access to Azure resources.
* Regularly update and rotate credentials stored in Azure Key Vault.
* Monitor and audit Azure resources for any security breaches or anomalies.

**9. Monitoring and Logging**

Implement comprehensive monitoring and logging solutions to track the migration process and ensure data integrity and security:

* Configure Azure Monitor and Azure Security Center for real-time monitoring and threat detection.
* Enable diagnostic logs for Azure Data Factory, SHIR, Azure Key Vault, and Express Route/Private Peering.
* Set up alerting based on predefined thresholds and anomaly detection.

**10. Conclusion**

This technical design document provides a structured approach to migrating an on-premises SQL database to Azure Blob Storage and Dataverse with the use of a Self-Hosted Integration Runtime (SHIR) private endpoint and Express Route/Private Peering for private network connectivity. Integration with Azure Key Vault enhances security by safeguarding client credentials. By following these guidelines, you can ensure a smooth, secure, and efficient migration process while maintaining data integrity and compliance with best practices.

**Application Architecture Document: SQL On-Premises Database Migration to Azure Blob Storage and Dataverse with Key Vault Integration Using Self-Hosted Integration Runtime (SHIR) Private Endpoint and Express Route/Private Peering. Azure Virtual Network VM with SHIR. HTTPS over TLS 1.2**

**Table of Contents**

1. **Introduction**
2. **High-Level Architecture**
3. **Components Overview**
4. **Application Flow**
5. **Key Components**
6. **Security Measures**
7. **Prerequisites**
8. **Deployment Steps**
9. **Monitoring and Logging**
10. **Conclusion**

**1. Introduction**

This application architecture document outlines the design of an application responsible for migrating an SQL on-premises database to Azure Blob Storage and Dataverse. The architecture incorporates several critical elements, including Self-Hosted Integration Runtime (SHIR) with a private endpoint, Express Route/Private Peering for secure network communication, Azure Virtual Network VM with SHIR, and secure data transmission using HTTPS over TLS 1.2. Additionally, Azure Key Vault is integrated for secure credential management.

**2. High-Level Architecture**

**Components:**

* **On-Premises SQL Server:** The source SQL database hosted on-premises.
* **Self-Hosted Integration Runtime (SHIR):** Installed on an Azure Virtual Network VM, SHIR facilitates data movement via a private endpoint.
* **Azure Virtual Network (VNet):** Provides a secure network environment for hosting the Azure VM with SHIR.
* **Express Route/Private Peering:** Ensures a private, dedicated, and secure network connection between the on-premises network and Azure services.
* **Azure Blob Storage:** Used as an intermediate storage location for data during migration.
* **Azure Data Factory:** Manages data pipelines, including data movement and transformation.
* **Azure Key Vault:** Securely stores client credentials (client ID and secrets).
* **Dataverse:** A Microsoft Power Platform service used for storing and managing transformed data.

**3. Components Overview**

**3.1. Self-Hosted Integration Runtime (SHIR)**

* SHIR is installed on an Azure Virtual Network VM.
* It provides the capability to securely move data between the on-premises SQL Server and Azure Blob Storage.
* Communication to SHIR is via a private endpoint for enhanced security.

**3.2. Azure Virtual Network (VNet)**

* Hosts the Azure VM running SHIR.
* Provides network isolation and security for the SHIR VM.

**3.3. Express Route/Private Peering**

* Establishes a private and dedicated network connection between the on-premises network and Azure.
* Ensures data transmission occurs securely and privately.

**3.4. Azure Blob Storage**

* Serves as an intermediate storage location for data during the migration process.

**3.5. Azure Data Factory**

* Manages data pipelines, including data movement and transformation.
* Orchestrates data flow between various components.

**3.6. Azure Key Vault**

* Securely stores and manages client credentials (client ID and secrets) used for authentication during data migration and integration.

**3.7. Dataverse**

* A Microsoft Power Platform service used for storing and managing transformed data.

**4. Application Flow**

1. Data is extracted from the on-premises SQL Server.
2. SHIR on the Azure VM securely transfers data over Express Route/Private Peering to Azure Blob Storage.
3. Data is transformed as necessary.
4. Azure Data Factory orchestrates the movement of transformed data from Azure Blob Storage to Dataverse.
5. Secure client credentials are retrieved from Azure Key Vault for authentication during data migration and integration.

**5. Key Components**

**5.1. Security**

* **TLS 1.2:** All data transmissions occur over HTTPS using TLS 1.2 encryption for secure communication.
* **Azure Key Vault:** Client credentials (client ID and secrets) are securely stored and accessed from Azure Key Vault.
* **Express Route/Private Peering:** Provides a private, dedicated, and secure network connection between on-premises and Azure resources.

**5.2. Data Transformation**

* Azure Data Factory facilitates data transformation as needed, including data cleansing, enrichment, and format changes.

**6. Security Measures**

* Proper firewall rules, security configurations, and network ACLs are set up on the on-premises SQL Server and Azure services.
* Role-Based Access Control (RBAC) and Azure Policy control access to Azure resources.
* Regular credential rotation and security audits are conducted.

**7. Prerequisites**

* Access to an Azure subscription.
* Configuration of an Azure Virtual Network with Express Route or Private Peering.
* Creation of an Azure Key Vault instance for storing credentials.
* Installation of Self-Hosted Integration Runtime (SHIR) on an Azure Virtual Network VM.
* Azure Blob Storage account setup.
* Azure Data Factory instance creation.
* Configuration of Dataverse environment.

**8. Deployment Steps**

1. Deploy an Azure VM within an Azure Virtual Network with SHIR installed.
2. Configure the SHIR private endpoint.
3. Establish Express Route/Private Peering between the on-premises network and Azure.
4. Set up Azure Blob Storage.
5. Create an Azure Data Factory instance.
6. Configure a Dataverse environment.
7. Securely store client credentials in Azure Key Vault.
8. Develop and deploy data migration and transformation pipelines in Azure Data Factory.

**9. Monitoring and Logging**

* Implement Azure Monitor and Azure Security Center for real-time monitoring and threat detection.
* Enable diagnostic logs for Azure Data Factory, SHIR, Azure Key Vault, and network monitoring.
* Configure alerting based on predefined thresholds and anomaly detection.

**10. Conclusion**

This application architecture provides a secure and efficient solution for migrating an on-premises SQL database to Azure Blob Storage and Dataverse. By incorporating a Self-Hosted Integration Runtime (SHIR) with a private endpoint, Express Route/Private Peering, secure data transmission using HTTPS over TLS 1.2, and integration with Azure Key Vault, this architecture ensures data integrity, privacy, and compliance with best practices throughout the migration process.