

Decouple Power Apps with Azure Integration



Dataverse



Service Bus



Event Hubs



WebHook



Azure Functions



Power Automate



- Praveen

Contents

Why Azure Integration with Power Apps?

Flow diagram of Azure integration with Power Apps

High-level Steps of Azure Integration using Plugin Registration Tool

Azure Service Bus, Event Hub high level overview & differences

Demo -

Creation of Azure Service bus in Azure Portal

Installation of Plugin Registration Tool via Power Platform CLI

New Registration of Service Endpoint for Service Bus

New Step Registration against Service Bus Service Endpoint

Trigger the action using the model-driven app

Verify the action message in Azure Service bus Queue Explorer

Listen to the Azure Service bus Queue message via Azure Function App

Listen to the Azure Service bus Queue message via Power Automate

Creation of Azure Event Hub in Azure Portal

New Registration of Service Endpoint for Event Hub

New Step Registration against Event Hub Service Endpoint

Trigger the action using the model-driven app

Verify the action message in Event hub via stream analytics

Creation of HTTP Triggered Azure Function in Azure Portal

New Registration of Service Endpoint for Azure Function Webhook

New Step Registration against Webhook Endpoint

Trigger the action using the model-driven app

Verify the action message in Azure Function Webhook via monitor

Decouple Power Apps with Azure Integration

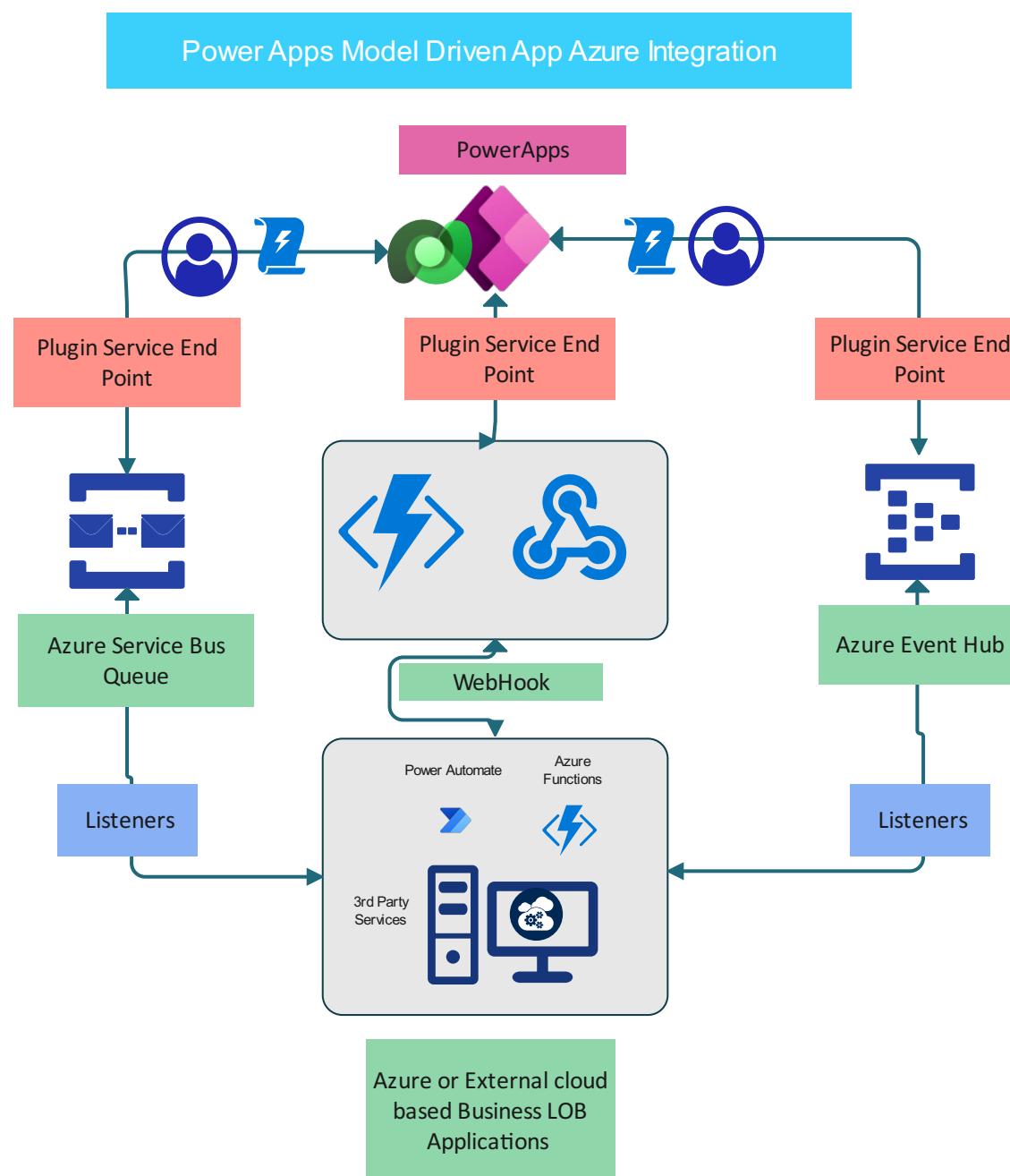


- Praveen



By integrating with Azure Service Bus / Azure Event Hub / Azure Functions.

- We are able to decouple power apps and improve the scalability and dependability of the application and services.
- Line-of-business (LOB) apps from outside the Dataverse can safely and effectively communicate with one other.
- For maintaining corporate data synchronization between numerous Dataverse systems or other Dataverse servers.



High Level Steps of integration

Create Azure Service Bus / Azure Event Hub/ Webhook

You can create an **event hub or service bus** in Azure either through API programming or interactively by using the Azure portal. Either way, after creating your event hub you must obtain a copy of the connection string and provide that string when registering the Azure service endpoint detailed in the next section.

With Microsoft Dataverse, you can send data about events that occur on the server to a web application using **webhooks**. Webhooks is a lightweight HTTP pattern for connecting Web APIs and services with a publish/subscribe model. Webhook senders notify receivers about events by making requests to receiver endpoints with some information about the events.

Register a new Service Endpoint / new Webhook

Use the Plug-in Registration tool (PRT) to register the service endpoint. When filling out the PRT registration form specify a contract type of Event Hub or service bus. For the message body format, you can choose XML or JSON. You must provide the connection string obtained when you created the event hub or service bus.

In the Plug-in Registration tool there is a new Register New WebHook option to select. When you register a WebHook you must provide three items of information - Name, Endpoint URL & Authentication (HttpHeader, WebhookKey-used for azure functions, HttpQueryString).

Register a Step & Start Listening

Dataverse needs to know the exact operation (table and message combination) that, when processed, would cause the Azure-aware plug-in to execute. Since you are creating an event hub/Service bus/Webhook, this operation would be related to the processing of Azure service bus/eventhub/functions data in particular. You must register a step for the Azure-aware plug-in in the execution pipeline.

Trigger the action

Perform an operation in Dataverse that would cause the plug-in to execute. This is the same operation (table and message combination) that you registered the plug-in step for in the previous section of this topic. You can perform the intended operation by using the power apps model driven app or web application or through application code accessing the Azure web services.

Verification

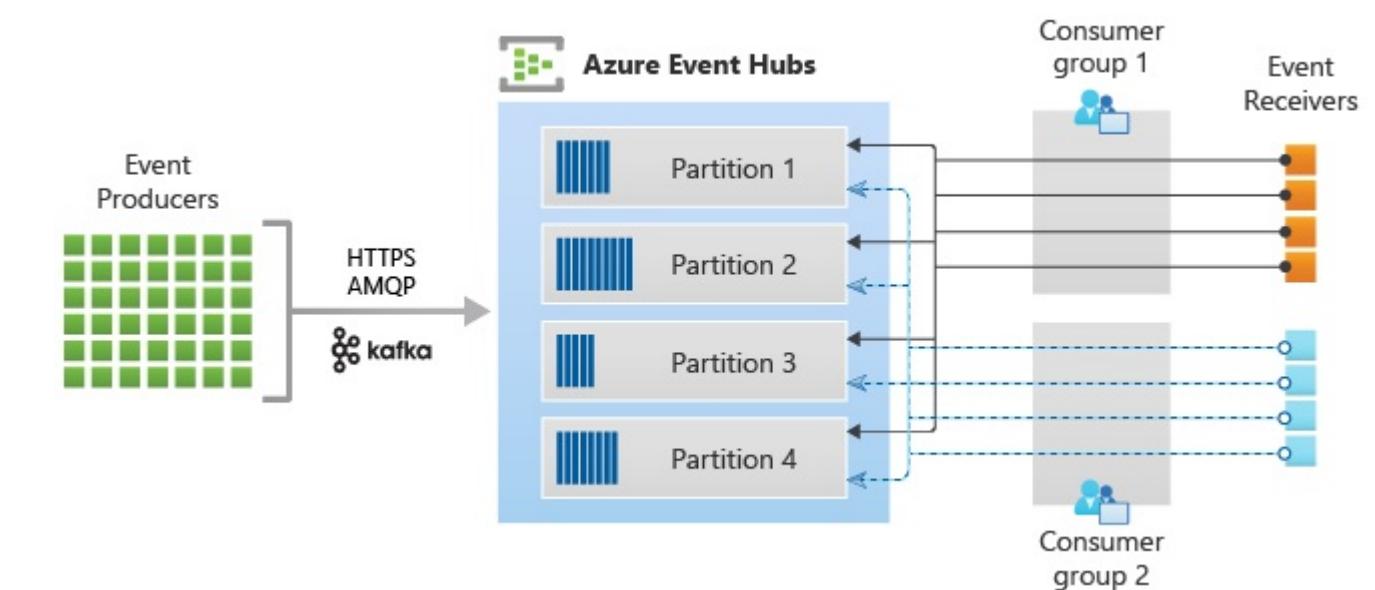
You can check the related system job in the Dataverse environment and look for a status of Succeeded. If you find a status of Failed, use the status information to identify the possible cause of the failure. You can then recheck the configurations of both systems

External cloud LOB Application Consumer listening the Actions for performing business cases

You can check the related system job in the Dataverse environment and look for a status of Succeeded. If you find a status of Failed, use the status information to identify the possible cause of the failure. You can then recheck the configurations of both systems

Azure EventHub

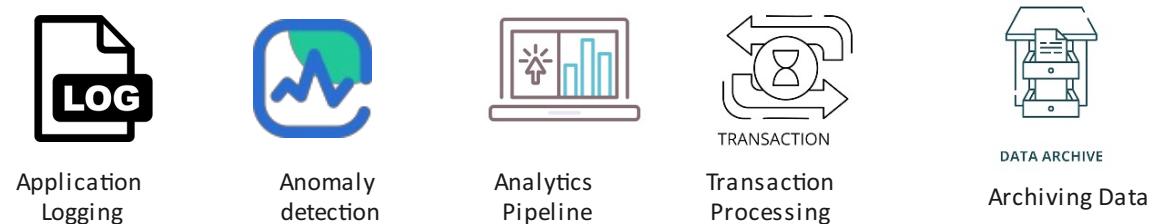
Key features



Integrations

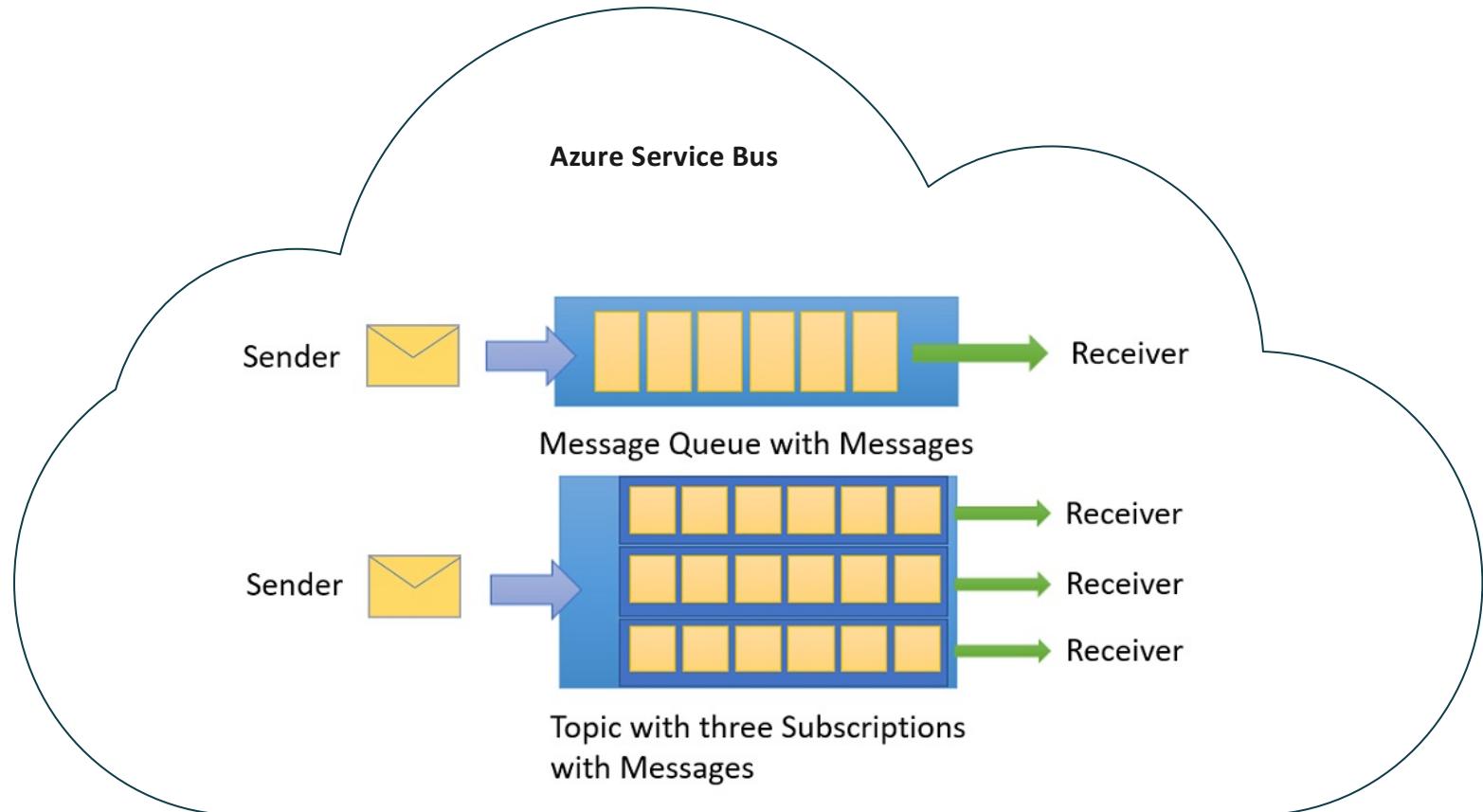


Scenarios



Azure ServiceBus

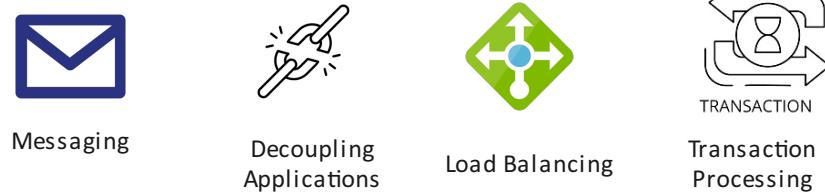
Key features



Integrations



Scenarios



When deciding between the WebHook model and the Azure Service Bus integration, here are some items to keep in mind:

- Azure Service Bus works for high scale processing, and provides a full queueing mechanism if Dataverse is pushing many events.
- Webhooks can only scale to the point at which your hosted web service can handle the messages.
- Webhooks enables synchronous and asynchronous steps. Azure Service Bus only allows for asynchronous steps.
- Webhooks send POST requests with JSON payload and can be consumed by any programming language or web application hosted anywhere.
- Both Webhooks and Azure Service Bus can be invoked from a plug-in or custom workflow activity.

	Azure Service Bus	Event Hubs
Definition	Azure Service Bus is a fully managed enterprise message broker with message queues and publish-subscribe topics	Azure Event Hubs is a big data streaming platform and event ingestion service. It can receive and process millions of events per second. Data sent to an event hub can be transformed and stored by using any real-time analytics provider or batching/storage adapters.
Scenario and usages	Messaging Decouple applications Load balancing Topics and subscriptions Transactions Message sessions	Anomaly detection (fraud/outliers) Application logging Analytics pipelines, such as clickstreams Live dashboards Archiving data Transaction processing User telemetry processing Device telemetry streaming
Key features	Queues Topics Messaging Sessions Auto-Forwarding Dead-Lettering Scheduled Delivery Message Deferral Transactions Filtering & Actions Auto Delete on Idle Duplicate Deletion Supports standard Advanced Message Queuing Protocol (AMQP) 1.0 and HTTP/REST protocols.	Fully managed PaaS Support for real-time and batch processing Capture event data - Long term retention (azure blob storage or azure data lake storage) Scalable
Integration	Event Grid Logic Apps Azure Functions Power Platform Dynamics 365 Azure Stream Analytics	Azure SQL DB Azure Storage Stream analytics Power BI Machine learning
Pricing	Basic Standard Premium more info: https://azure.microsoft.com/en-us/pricing/details/service-bus/	Basic Standard Premium more info: https://azure.microsoft.com/en-us/pricing/details/event-hubs/