

Fundraising and Engagement

for Dynamics 365 Sales, a Microsoft solution built with MISSION CRM

Architecture Overview

**Published**: October 9, 2020

**Release Version**: v1.0

Contents

[Overview 2](#_Toc52969553)

[Security 4](#_Toc52969554)

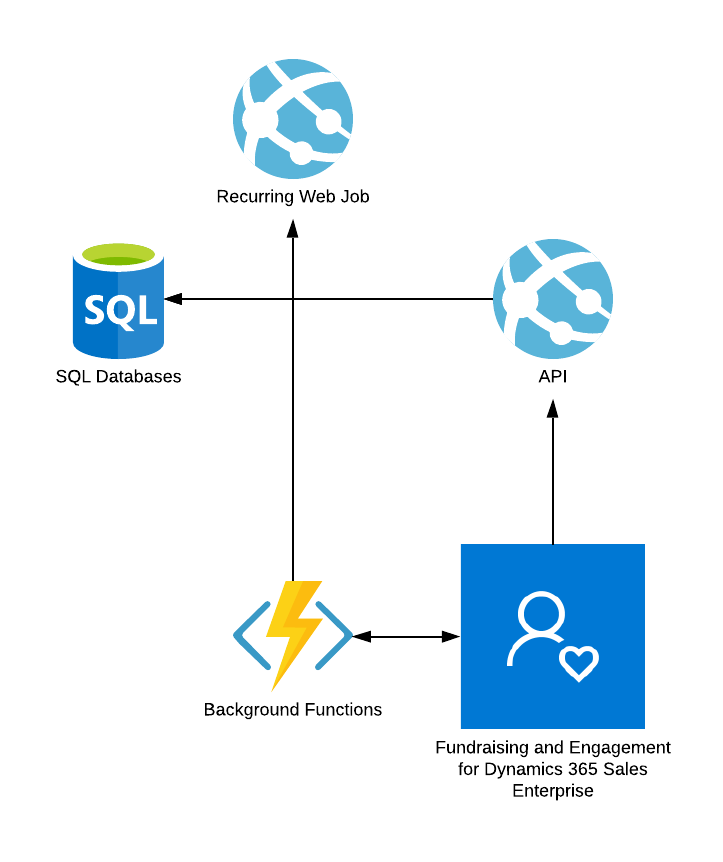
[Recurring Donation Process 6](#_Toc52969555)

[Azure SQL Database 8](#_Toc52969556)

[Learn More About Azure and Extending Dynamics 365 10](#_Toc52969557)

# Overview

Fundraising and Engagement for Dynamics 365 Sales, a Microsoft solution built with MISSION CRM, has been designed to operate with a single tenant Azure environment. A complementary Azure environment is necessary for specific functional areas of the Fundraising and Engagement solution to work as intended, including payment processing, recurring giving processing, data summaries, and key performance indicator generation as well as general data synchronization. This document outlines the Azure components required to interact with and support the Fundraising and Engagement solution.



## Util**izes .NET Core**

The dependent Fundraising and Engagement Azure components have been built to leverage the latest advancements in Azure and utilize .NET Core 3.1, built to manage modern applications purpose-built to run within the Azure architecture.

## Background Services Function App

The solution leverages a Function App which synchronizes data back to Dynamics 365, calculates summary values, and produces Bank Run submission files. This application manages the background services needed for Fundraising and Engagement and is the only application that communicates directly with Dynamics 365 utilizing an application registration.

## Recurring Engine Web Job Manages the Recurring Gift Process

The Azure application that manages recurring donations reads values from within the Azure SQL database. It assumes that any Payment Schedules created in Dynamics 365 reside within this local Azure SQL database. A timer triggers the process every 24 hours, defaults to 07:00 AM local time, and processes the recurring records yet to be billed.

## Operates When Dynamics 365 is Down

The recurring engine configuration is stored locally on the Azure SQL table, ensuring that the Azure Web Job is capable of processing transactions even if Dynamics 365 is not available. The details of successful and failed transactions are stored locally in an Azure SQL database until the logs are written back to Dynamics 365 via the Background Services Function App.

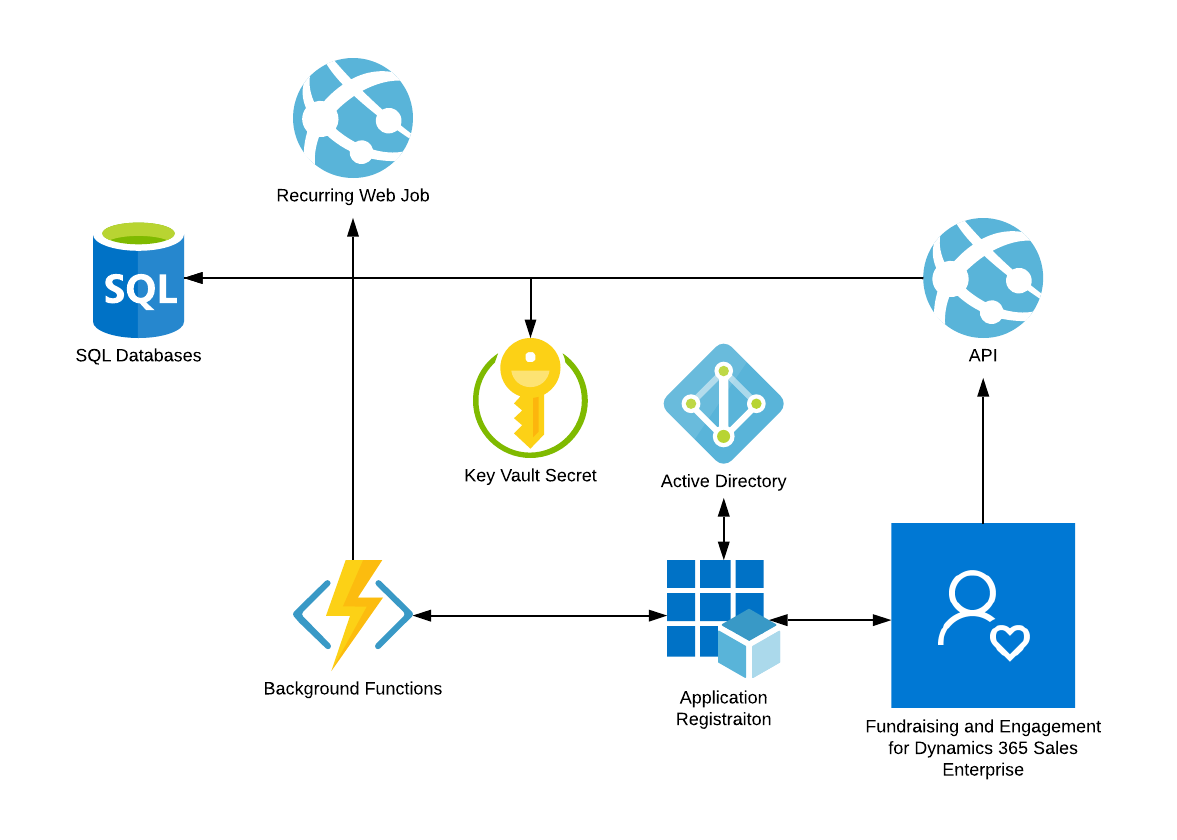
## Utilizes the Dynamics 365 Rest API

The connection with the Background Services Function App leverages Dynamics 365’s rest API and not the traditional Dynamics 365 SDK due to its incompatibility with .NET Core.

# Security

The Azure environment utilizes Server-to-Server security between the Background Services Function App and the Dynamics 365 environment. This allows the application to write successful transactions to Dynamics 365 using a specific application user that the Azure App will impersonate. This application users’ permissions within Dynamics 365 are dictated by its related security roles. This application user should only write records and cannot read any of the data in Dynamics 365. Application users do not consume a Dynamics license.

The Recurring Donation Engine is not externally accessible and is only visible to the internal Azure environment.



## Use of Key Vault Secrets

Sensitive connection information, such as SQL connection strings and users, are stored as secrets within the key vault. Whether it is the client key used to write data back to Dynamics 365 as the application user or the API that needs to process updated records from Dynamics and write them to the Azure SQL database, key vault secrets are used to retrieve sensitive information required to perform critical application functions. This protects sensitive information and ensures that it is not available directly from the applications in Azure.

## Azure AD Application Users (S-2-S Authentication)

The use of an Azure AD Application Registration allows the environment to make use of the impersonation rights available with Azure Active Directory. This process is sometimes referred to as Server-to-Server authentication, or passwordless authentication.

## Azure SQL Server Encryption at Rest

A default feature of Azure SQL server, data is encrypted at rest and this is the default setting for a Fundraising and Engagement deployment. Learn more about [Transparent Data Encryption](https://docs.microsoft.com/en-us/azure/azure-sql/database/transparent-data-encryption-tde-overview?tabs=azure-portal) (TDE).

# Recurring Donation Process

The recurring donation process is timed to run every 24 hours to capture any payment schedules that are expected to be billed on a particular day. The process is engineered to bill any records that have yet to be billed if a recurring process was missed due to an Azure environment error such as unexpected downtime. The payment method which stores the tokenized representation of the credit card or bank account is used to present the relevant information to the payment gateway along with the amount to be billed.

The recurring donation process steps are as follows.

## Recurring donation process starts

The process is on a timer, which triggers the process every 24 hours and defaults to 07:00 AM local time. Local time is dictated by the Azure environment and not necessarily the Dynamics 365 user.

## A list of applicable recurring donations is retrieved

A list of applicable recurring donations to process is retrieved by the engine and the following criteria must be met by the recurring donation for it to be included in the process:

1. The status must be active.
2. The next donation must be equal to or less than the time of when the process starts.
3. The payment method must be present.
4. The frequency must exist (without this the system would not understand when the next donation would be and therefore cannot trust that the next donation date set on the record can be trusted).
5. The next donation date must not be greater than ten days.

## Recurring donation processing completes

The recurring process passes the donation amount value and the payment method to the Payment Gateway and, based on the response, the following occurs:

1. For Successful Transactions a Successful (Completed) record is created. This record is synced back to the Dynamics 365 Fundraising and Engagement solution.
2. For Failed Transactions, a Failed record is created. This record is synced back to Dynamics 365.
3. The payment schedule is updated to record the following:
   1. Next donation date
   2. Last donation date
   3. Number of concurrent failures
   4. Number of successful charges.

NOTE: The above updates are posted regardless of a failed or successful transaction.

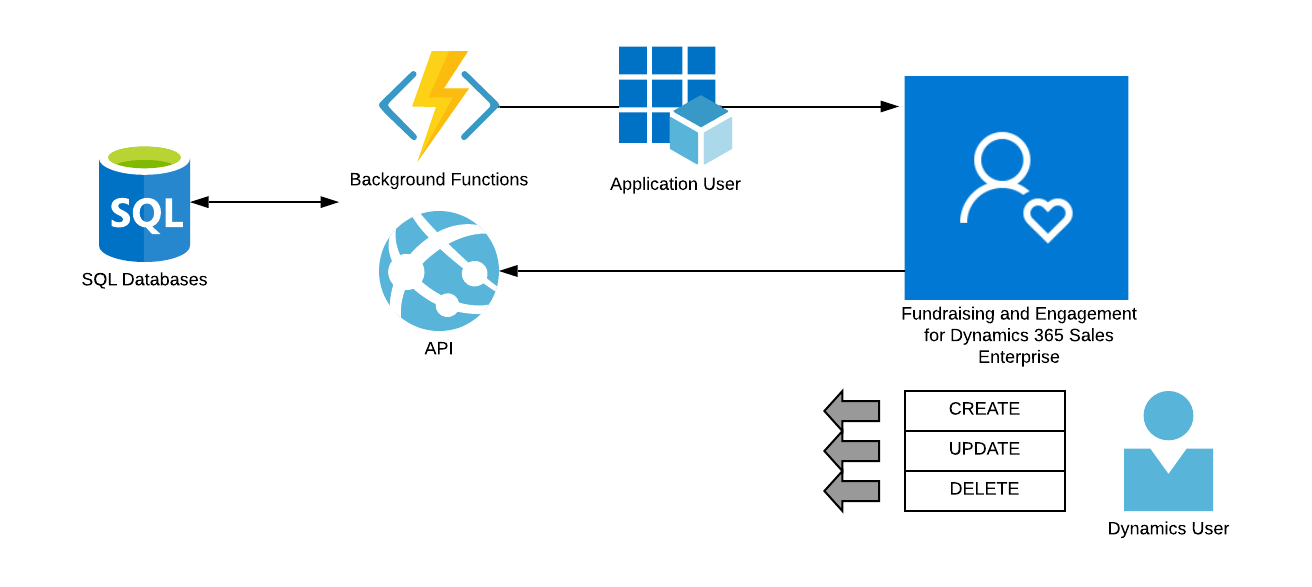
## Background Services App

The Background Services App syncs data back to Dynamics 365 every 30 minutes and captures all items that have been created or updated that have yet to sync back to Dynamics 365. When viewing records in the Dynamics 365 Fundraising and Engagement solution, the information presented to the user will be the most up-to-date data available within the last 30 minutes.

Synchronization is triggered by a flag that is present on each table labeled ‘SyncDate’. The flag is a SQL datetime column which indicates to the app that a record requires syncing. Upon a successful synchronization, the flag is updated with the date and time of when the synchronization occurred.

# Azure SQL Database

The Azure environment is dependent on a limited working set of data local to the Azure environment. This data is a paired down set of tables which contain key information on every Fundraising and Engagement custom entity in Dynamics 365. This data assumes it was synchronized with the production data at go live and is expected to contain a perfect reflection of the data in Dynamics 365.



The Azure SQL Database and Dynamics 365 are continuously kept in sync using the following processes:

## Dynamics 365 to Azure SQL Plug in

A plug-in is triggered when a user or process updates records in Dynamics. The plug-ins call the API depending on how the data is being altered in Dynamics:

* **On creation of a record**: The plug-in calls the create end point signaling that the API should create this record in Azure SQL.
* **On change of a record**: The plug-in calls the update end point signaling that the API should update existing data.
* **Upon deletion of a record**: The plug-in calls the delete end point which flags data in Azure SQL as being deleted but does not automatically delete the entry, and instead performs a soft delete.

## Azure to Dynamics 365 Data Sync Process

The process of syncing updated data or data generated by the Azure environment (such as transactions created from the recurring engine) is managed by the synchronization application based on the API. The process connects to Dynamics using the Application User and is dependent on the application user having the correct privileges to create and update the records passed from the Azure environment. The process includes:

1. Polling for changes every 30 minutes.
2. Records in Dynamics that are shown to have been created or updated by the application user.
3. The application user is established during the initial build of the system and is not user configurable.

# Learn More About Azure and Extending Dynamics 365

This document assumes users provisioning Fundraising and Engagement Azure environments have a solid understanding of both the Azure components that will be created and configured as well as the Dynamics 365 Fundraising and Engagement environment which will utilize this functionality.

The following courses and certifications paths are recommended prior to utilizing this guide:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Educate** | **Accredit** | **Discussion** |
| Azure Fundamentals | [Learning Path](https://docs.microsoft.com/en-us/learn/paths/azure-fundamentals/) | [Exam](https://docs.microsoft.com/en-us/learn/certifications/exams/ai-900) | [Community](https://powerusers.microsoft.com/t5/PowerApps-Community/ct-p/PowerApps1) |
| Dynamics 365 Fundamentals | [Learning Path](https://docs.microsoft.com/en-us/learn/paths/dynamics-365-fundamentals/) | [Exam](https://docs.microsoft.com/en-us/learn/certifications/exams/mb-900) | [Community](https://powerusers.microsoft.com/t5/Microsoft-Flow-Community/ct-p/FlowCommunity) |