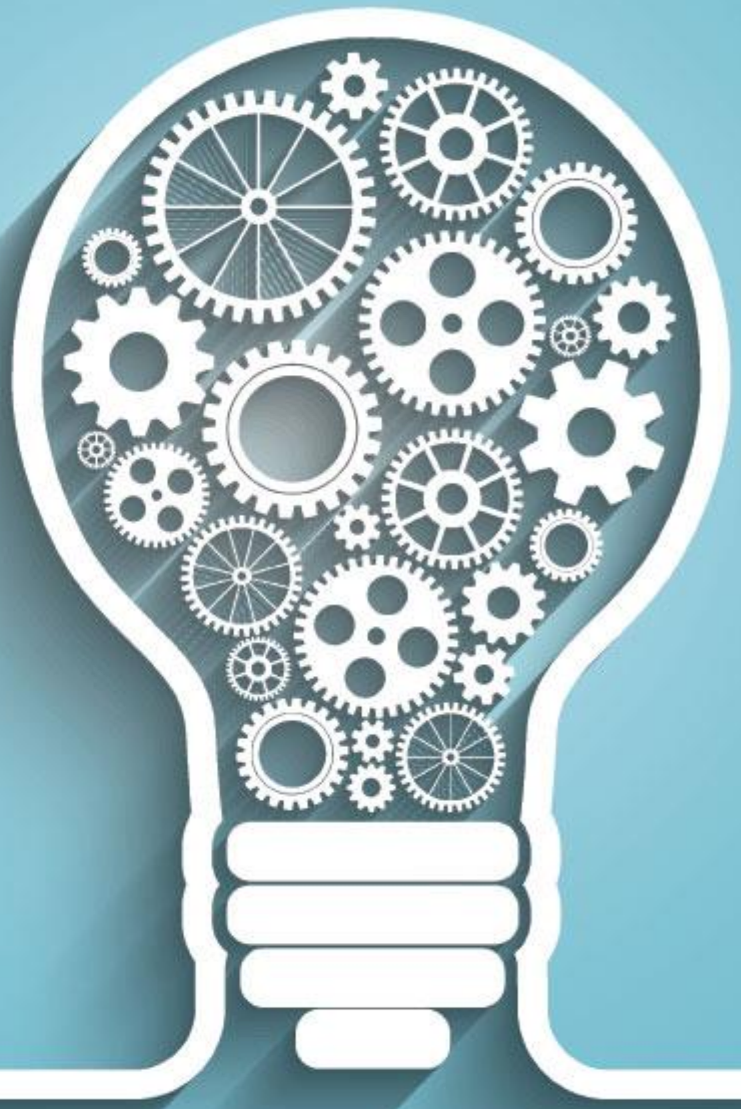
MICROSOFT DYNAMICS 365

Azure Integration Patterns



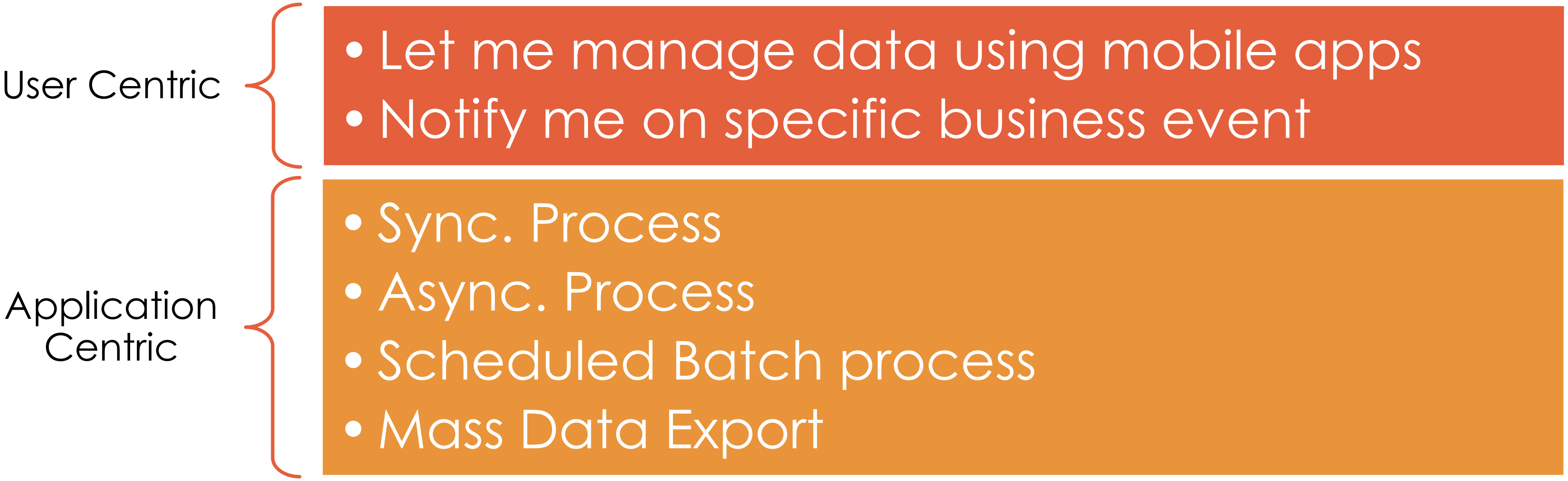
DYNAMIZE

Yaniv Arditi
Microsoft Dynamics Specialist
yaniv@dynamize.onmicrosoft.com

Agenda

- Common Integration Scenarios
- Power Platform
- Dynamics 365 & Azure
- Solution Patterns

Common Integration Scenarios



User Centric

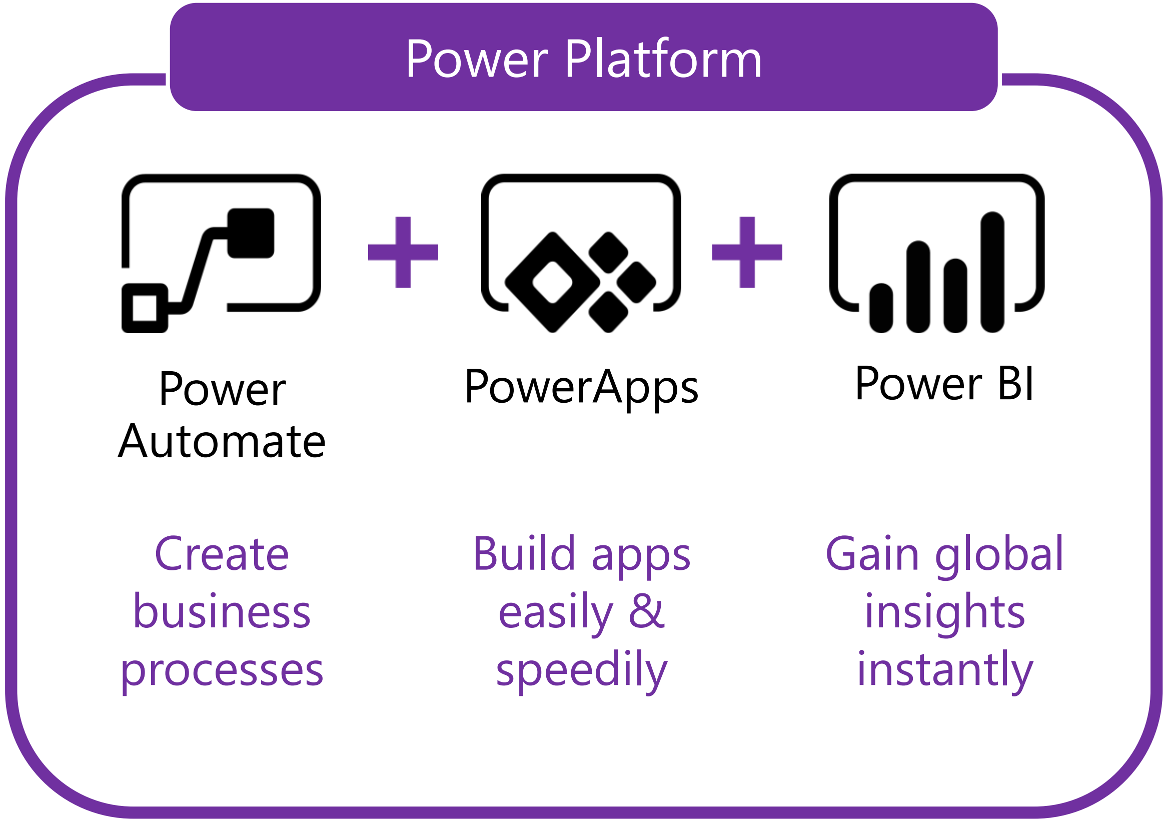
- Let me manage data using mobile apps
- Notify me on specific business event

Application Centric

- Sync. Process
- Async. Process
- Scheduled Batch process
- Mass Data Export

Power Platform

- No/low code business applications platform
- Harness data from multiple sources
- Built-in/custom connectors exposing triggers & APIs



Power Platform

Power Automate + PowerApps + Power BI

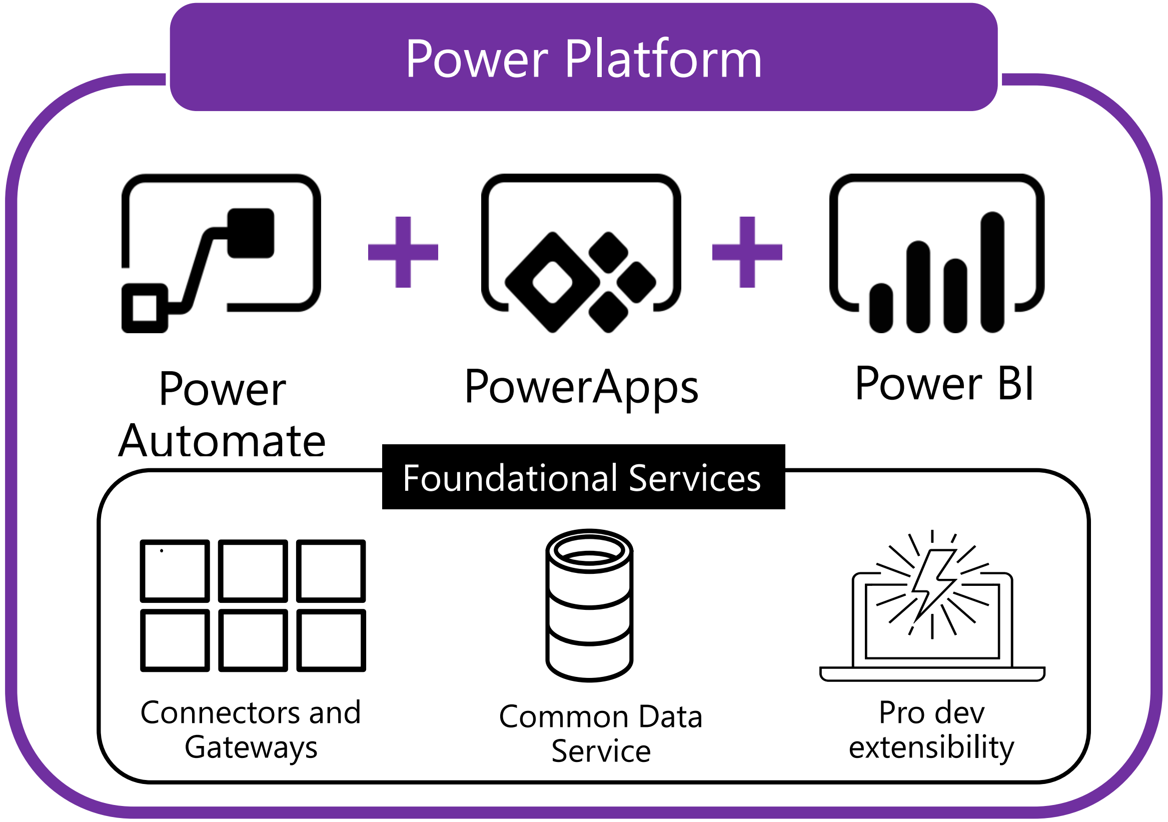
Power Automate: Create business processes

PowerApps: Build apps easily & speedily

Power BI: Gain global insights instantly

Power Platform

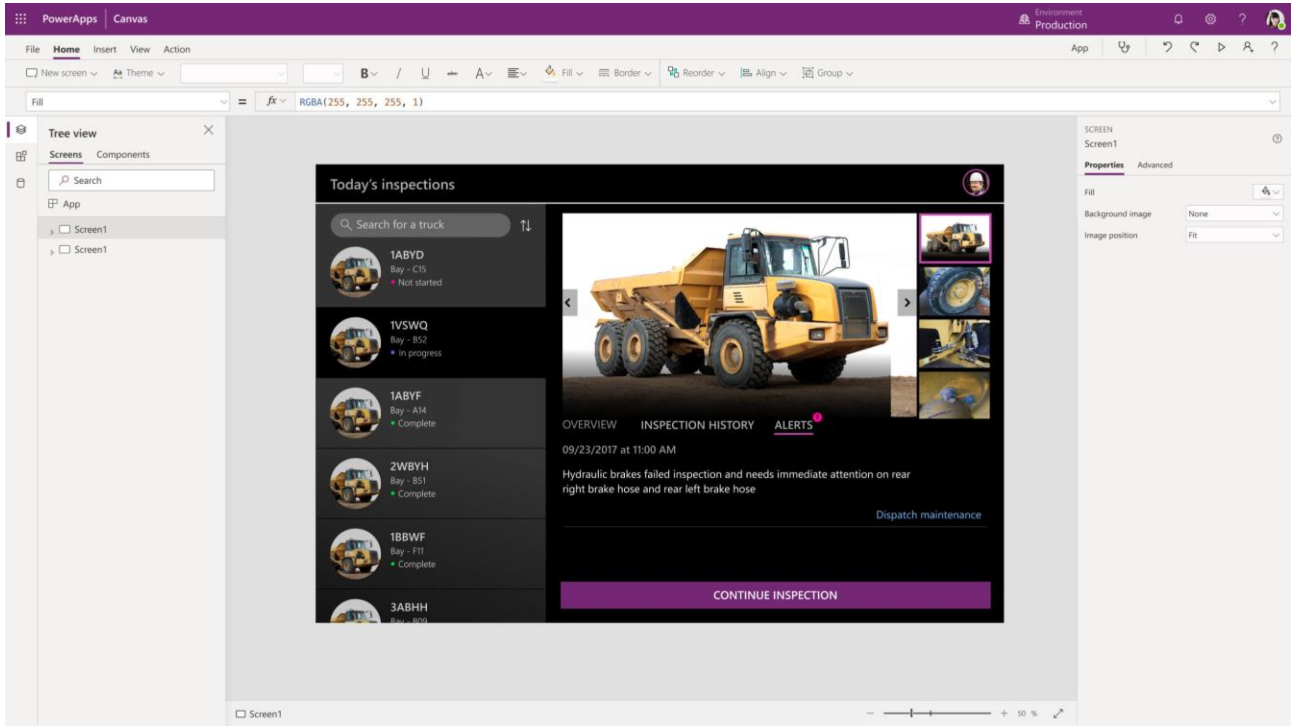
- No/low code business applications platform
- Harness data from multiple sources
- Built-in/custom connectors exposing triggers & APIs



Power Apps

- Build Model Driven Apps
  - Create reusable components & services
  - Build enterprise applications

- Build Canvas Apps
  - Extend model-driven components/apps by creating different UX
  - Create easy task-based apps available on any device



Fill = RGBA(255, 255, 255, 1)

Fill

Background image: None

Image position: Fit

Today's inspections

1ABYD
Bay - C15
Not started

1VSWQ
Bay - B52
In progress

1ABYF
Bay - A14
Complete

2WBYH
Bay - B51
Complete

1BBWF
Bay - F11
Complete

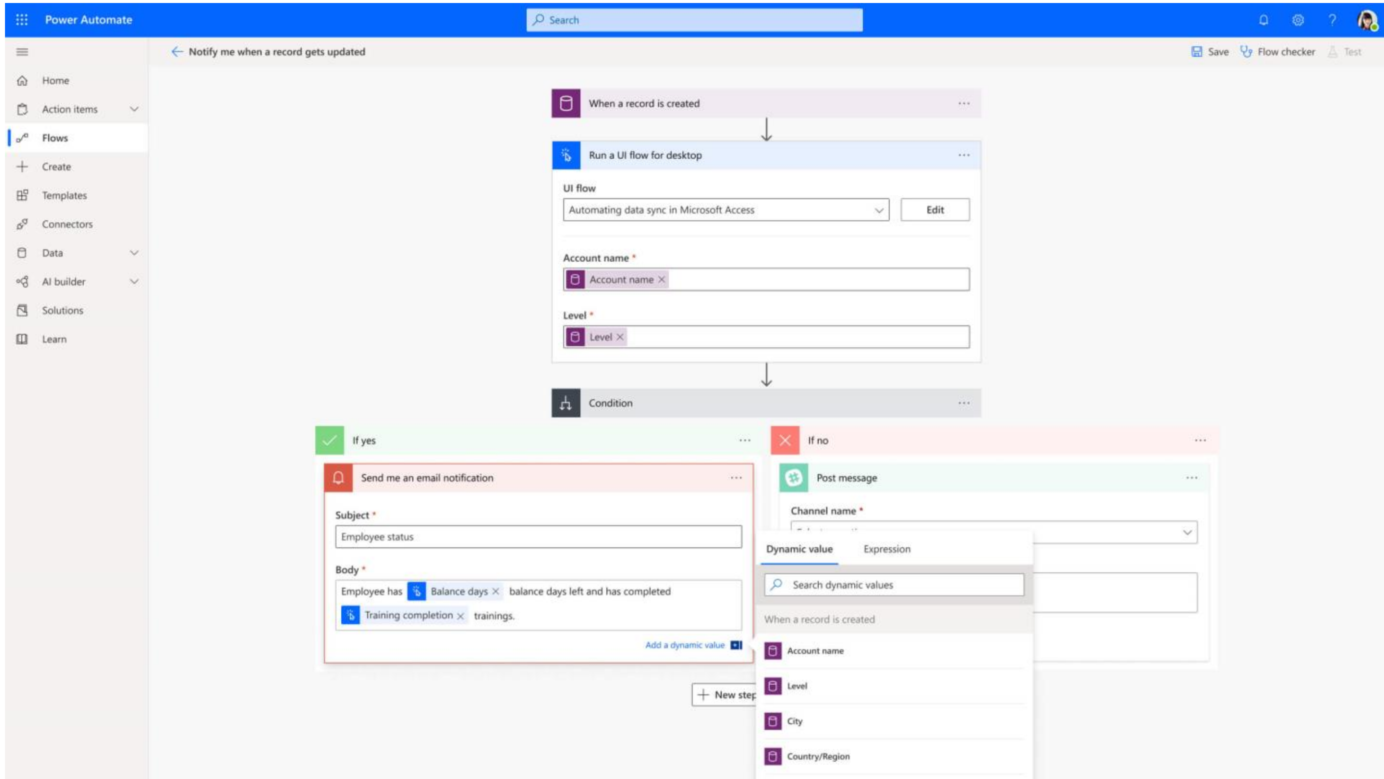
3ABHH
Bay - B09

09/23/2017 at 11:00 AM

Hydraulic brakes failed inspection and needs immediate attention on rear right brake hose and rear left brake hose

Power Automate (aka Flow)

- Process as a service
- Automate business scenarios with
  - Triggers
  - Steps
  - branching conditions
  - Built-in/custom connectors



Notify me when a record gets updated

When a record is created

Run a UI flow for desktop

UI flow: Automating data sync in Microsoft Access — Edit

Account name *: Account name

Level *: Level

Condition

If yes

Send me an email notification

Subject *: Employee status

Body *: Employee has Balance days balance days left and has completed Training completion trainings.

If no

Post message

Channel name *

When a record is created

Account name

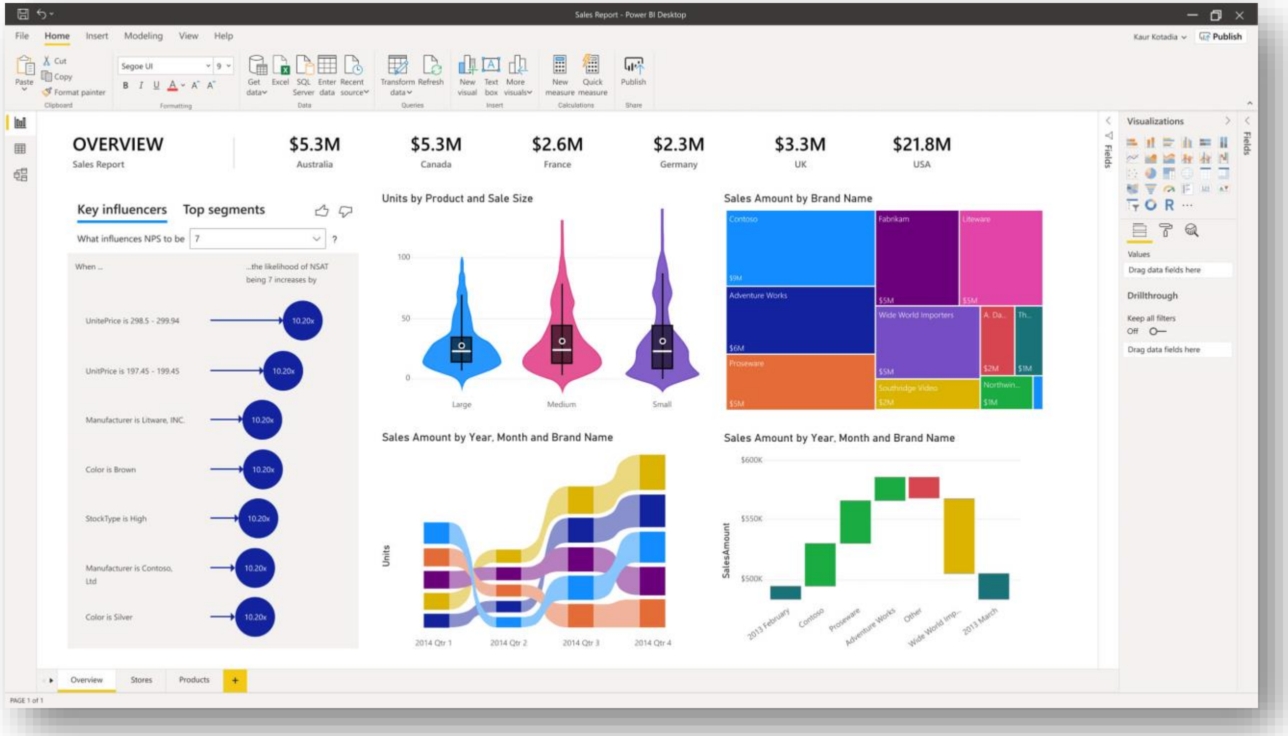
Level

City

Country/Region

Power BI

- Business analytics solution
- Connect to hundreds of data sources
- Visualize data and share insights
- Embed Dashboard and reports in apps, websites & Power Apps



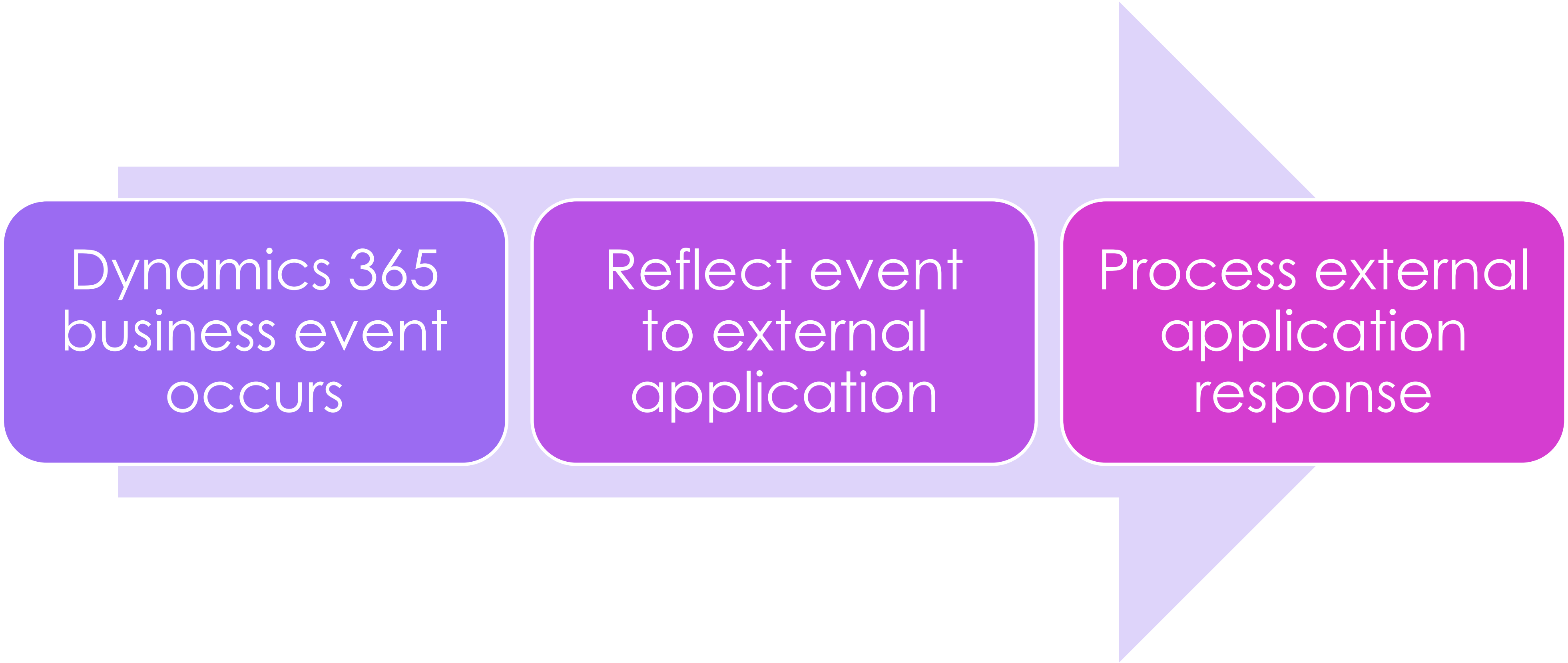
Dynamics 365 & Azure

- Many Dynamics 365 features are built using Microsoft Azure components
- Azure AD manages Dynamics 365 identities
- OOB ability to publish Dynamics 365 events to Service Bus, Azure Function, Logic Apps & Event Hub
- Many Azure components are building blocks for integrating Dynamics 365 with other systems

Solution Patterns

- Sync. Process
- Async. Process
- Scheduled Batch process
- Mass Data Export

Outbound Sync. Process



Dynamics 365 business event occurs

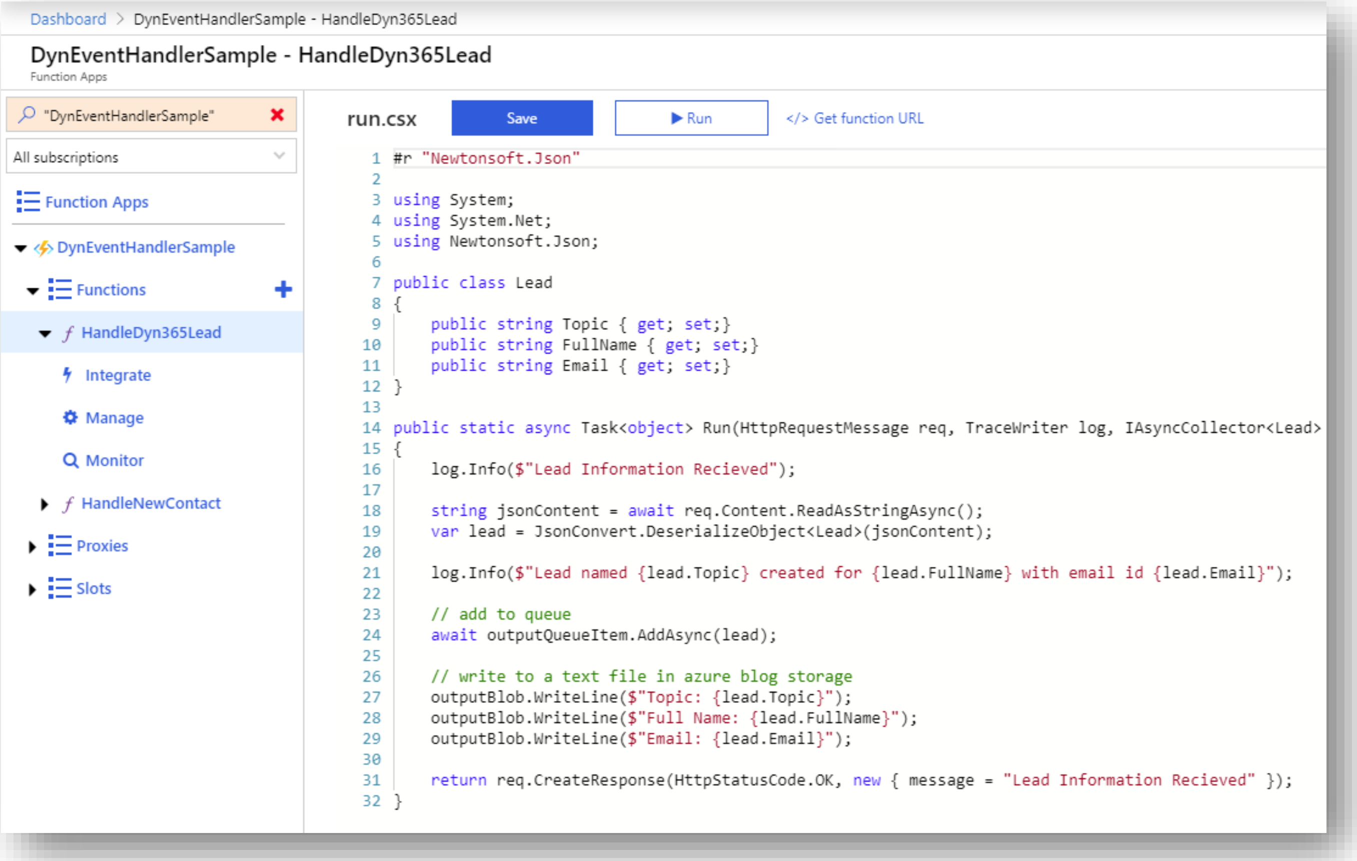
Reflect event to external application

Process external application response

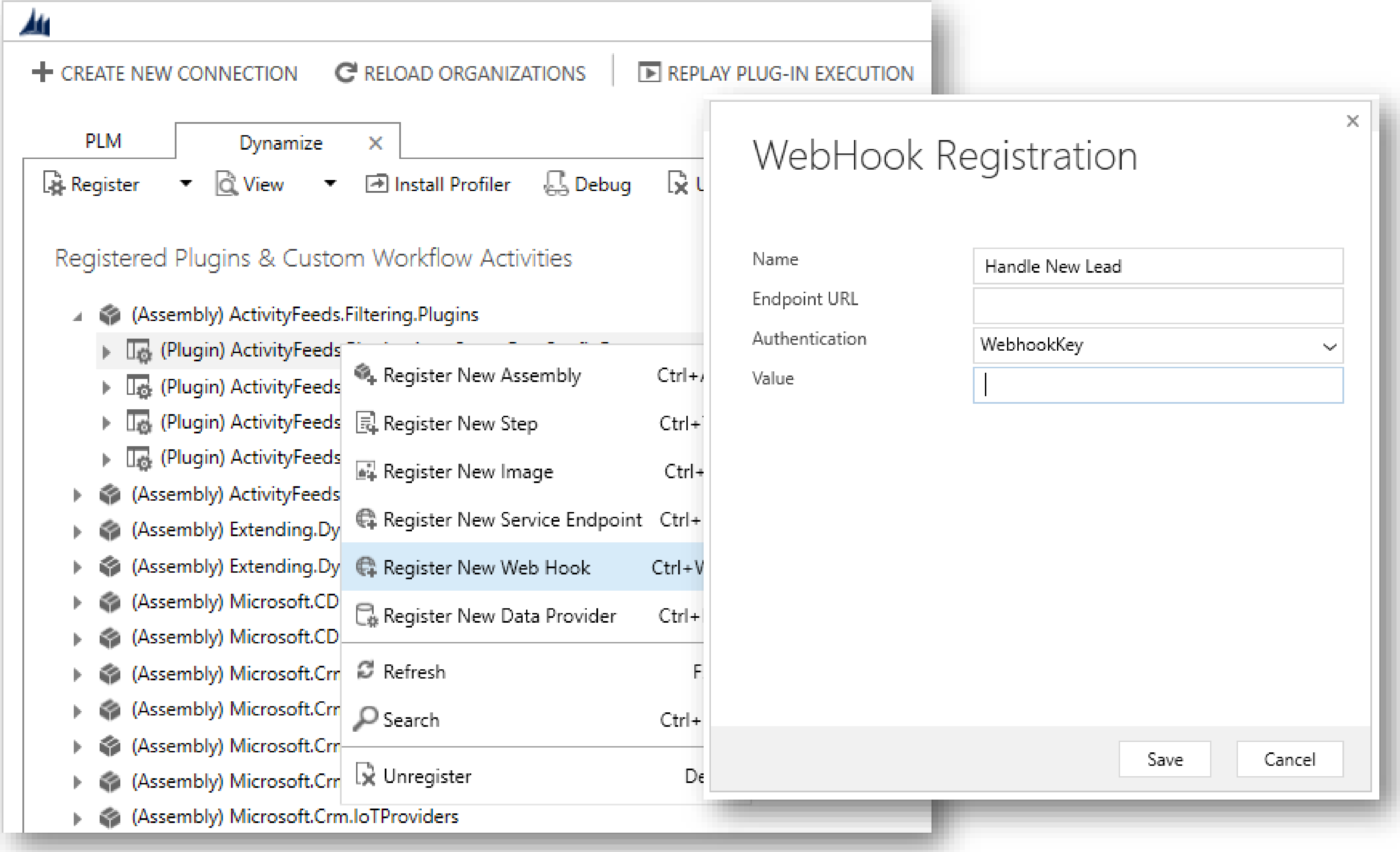
Outbound Sync. Process

Azure Function App

- Serverless computing
- Running small pieces of code
- C#, Java, JavaScript, PowerShell & Python
- Variety of triggers: Webhook, Schedule, Queue etc.
- Logging & tracing



```
#r "Newtonsoft.Json"

using System;
using System.Net;
using Newtonsoft.Json;

public class Lead
{
    public string Topic { get; set;}
    public string FullName { get; set;}
    public string Email { get; set;}
}

public static async Task<object> Run(HttpRequestMessage req, TraceWriter log, IAsyncCollector<Lead>
{
    log.Info($"Lead Information Recieved");

    string jsonContent = await req.Content.ReadAsStringAsync();
    var lead = JsonConvert.DeserializeObject<Lead>(jsonContent);

    log.Info($"Lead named {lead.Topic} created for {lead.FullName} with email id {lead.Email}");

    // add to queue
    await outputQueueItem.AddAsync(lead);

    // write to a text file in azure blog storage
    outputBlob.WriteLine($"Topic: {lead.Topic}");
    outputBlob.WriteLine($"Full Name: {lead.FullName}");
    outputBlob.WriteLine($"Email: {lead.Email}");

    return req.CreateResponse(HttpStatusCode.OK, new { message = "Lead Information Recieved" });
}
```



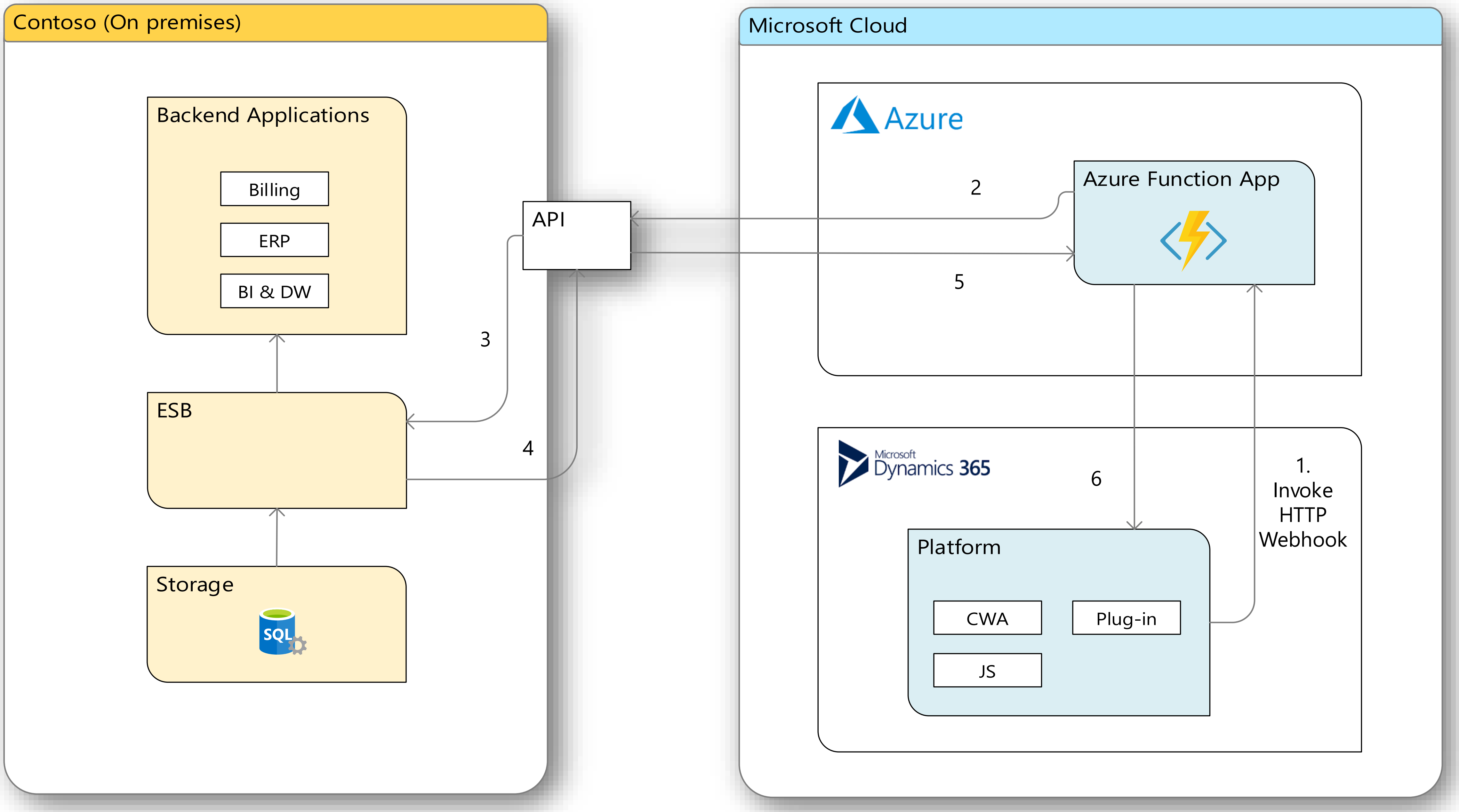
WebHook Registration

Name: Handle New Lead

Endpoint URL:

Authentication: WebhookKey

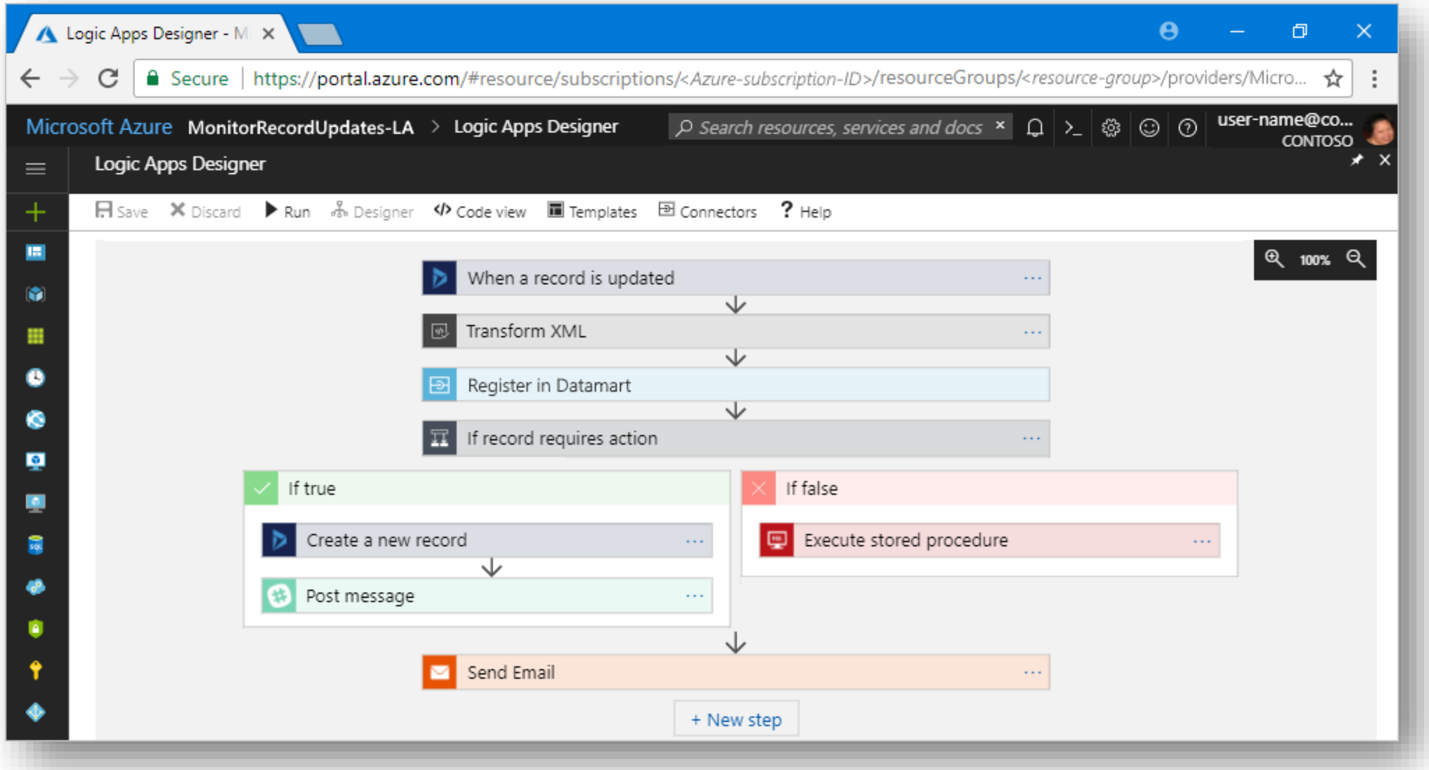
Value:


Contoso (On premises)
Backend Applications
Billing
ERP
BI & DW
ESB
Storage
SQL
API
3
4
Microsoft Cloud
Azure
Azure Function App
2
5
Microsoft
Dynamics 365
6
1.
Invoke
HTTP
Webhook
Platform
CWA
Plug-in
JS

Outbound Sync. Process

Logic Apps

- Declarative process
- Built-in/Custom triggers & connectors
- Exception handling & retry
- Logging & Tracing



When a record is updated

Transform XML

Register in Datamart

If record requires action

If true

Create a new record

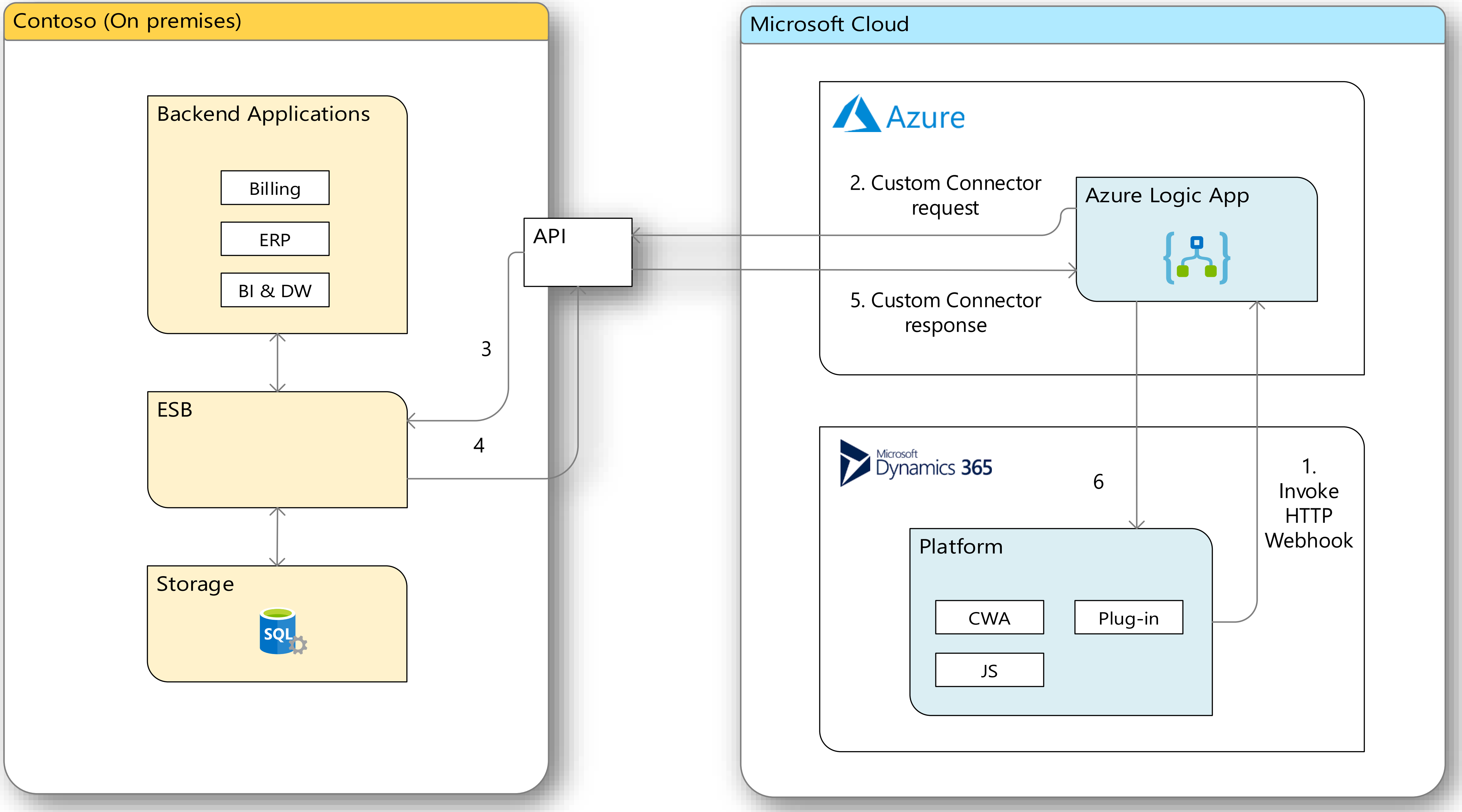
Post message

If false

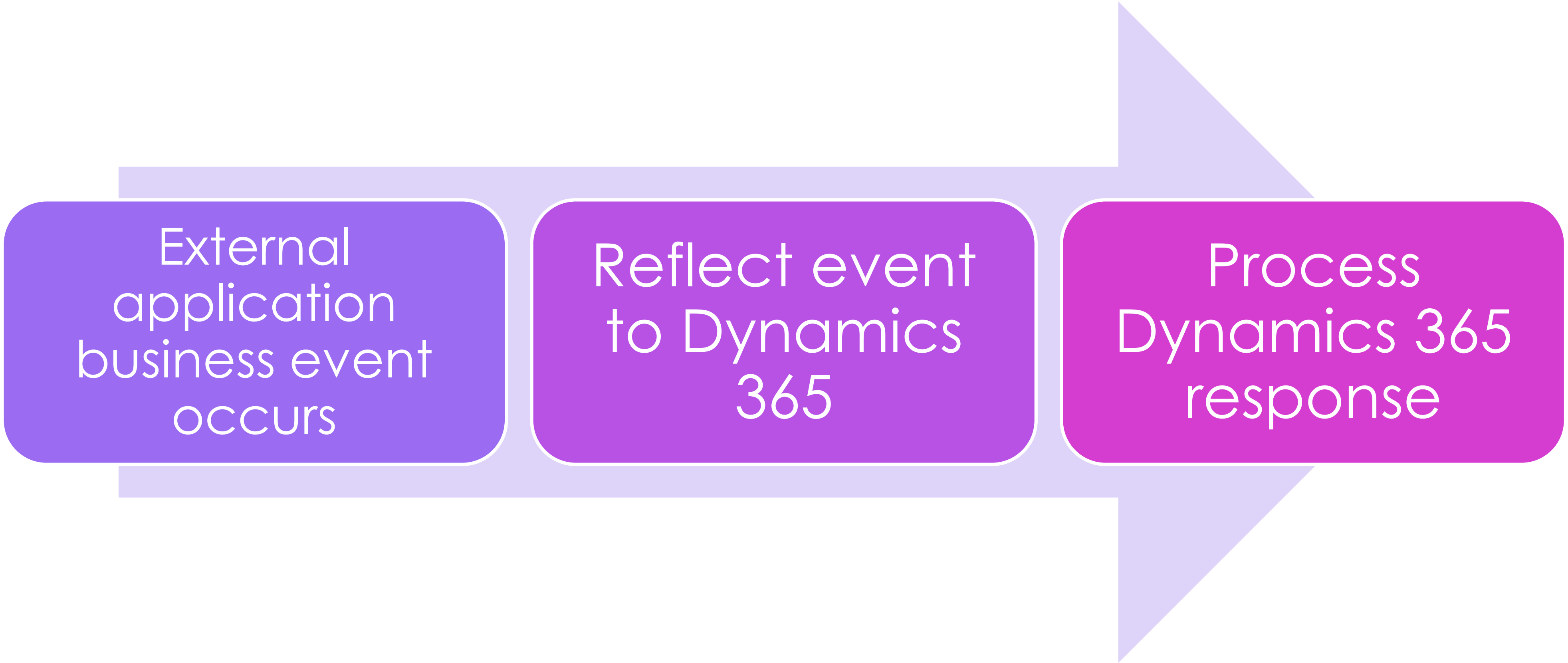
Execute stored procedure

Send Email

+ New step


Contoso (On premises)
Backend Applications
Billing
ERP
BI & DW
ESB
Storage
SQL
API
3
4
Microsoft Cloud
Azure
2. Custom Connector request
Azure Logic App
5. Custom Connector response
Microsoft Dynamics 365
6
1. Invoke HTTP Webhook
Platform
CWA
Plug-in
JS

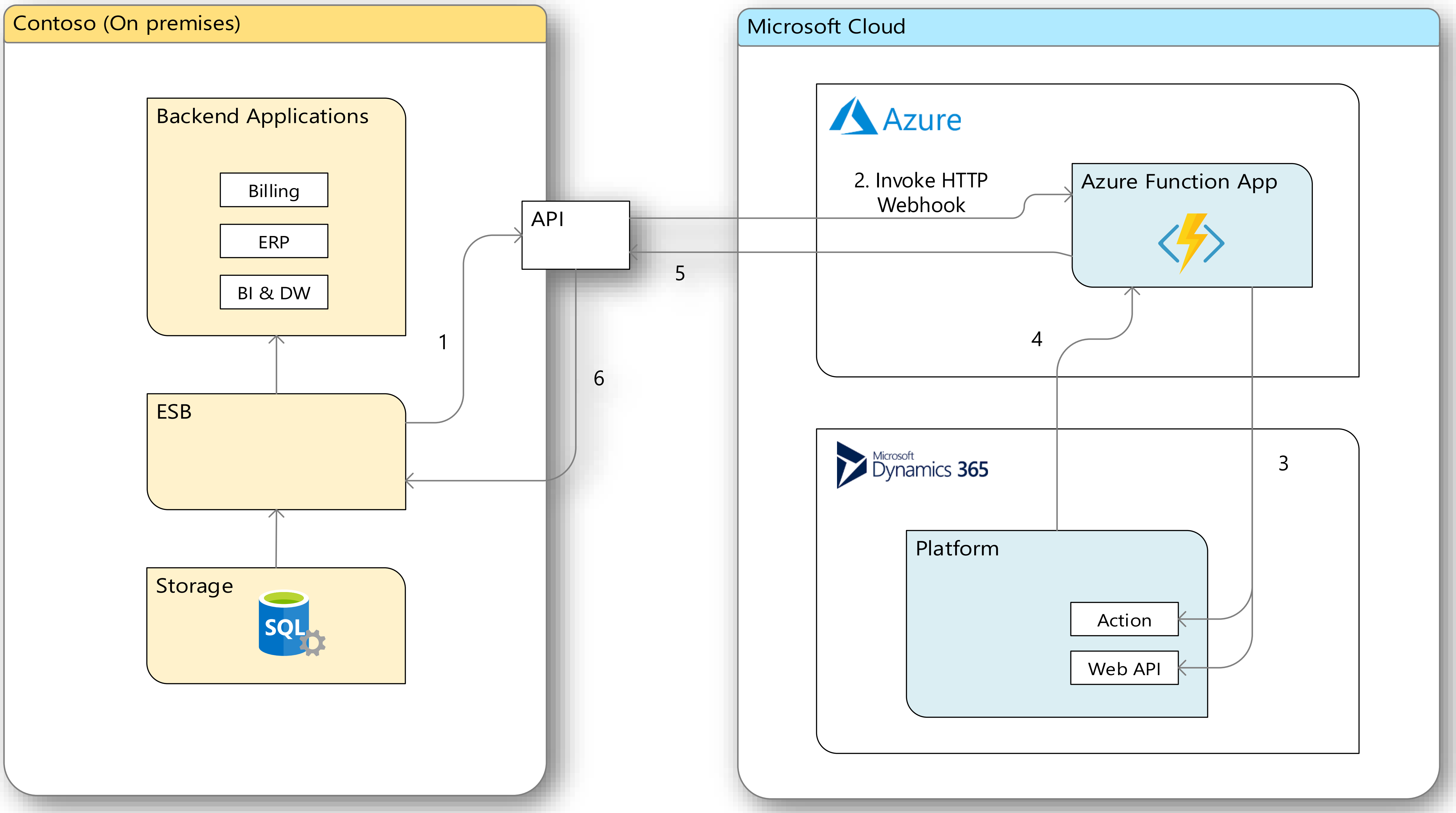
Inbound Sync. Process

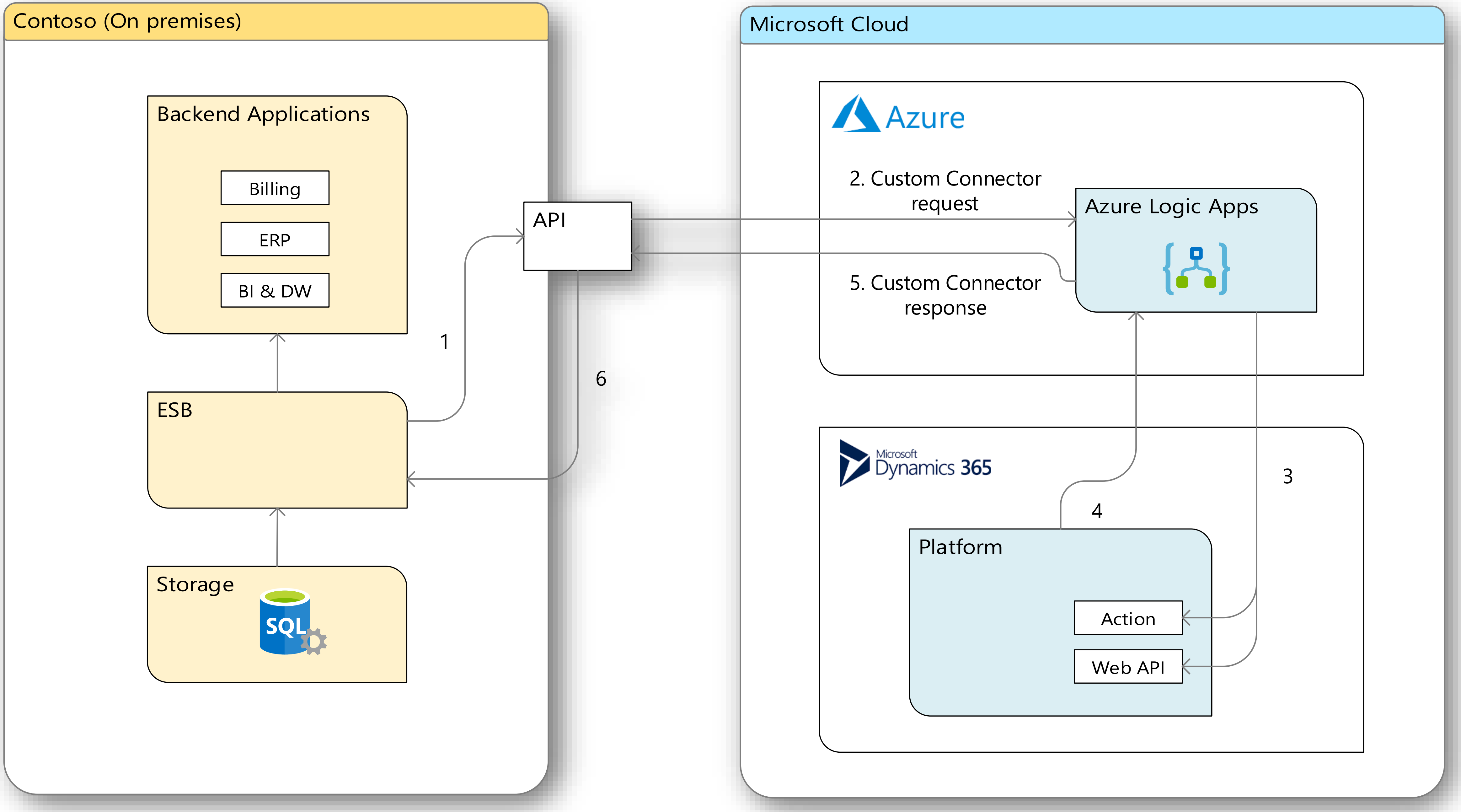


External application business event occurs

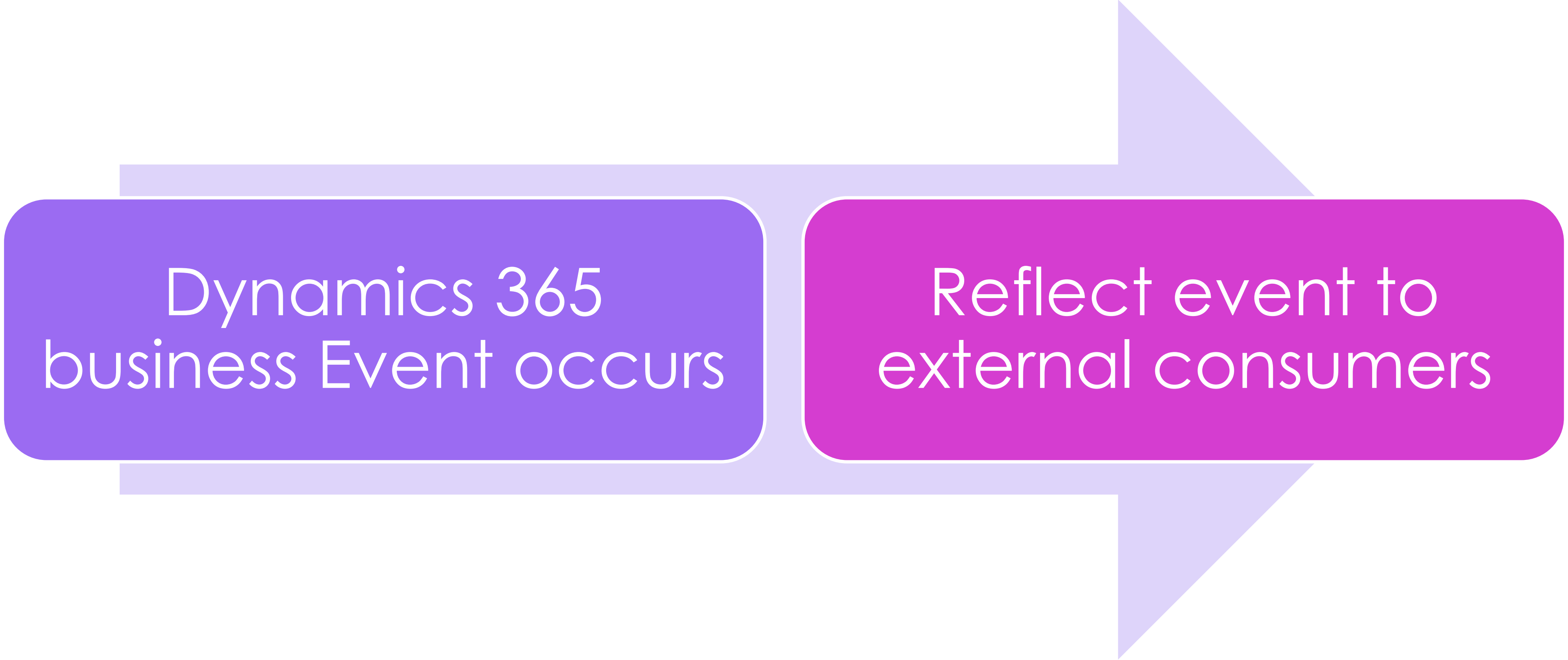
Reflect event to Dynamics 365

Process Dynamics 365 response


Contoso (On premises)
Backend Applications
Billing
ERP
BI & DW
ESB
Storage
SQL
API
1
6
5
Microsoft Cloud
Azure
2. Invoke HTTP Webhook
Azure Function App
4
3
Microsoft Dynamics 365
Platform
Action
Web API


Contoso (On premises)
Backend Applications
Billing
ERP
BI & DW
ESB
Storage
SQL
API
1
6
Microsoft Cloud
Azure
2. Custom Connector request
Azure Logic Apps
5. Custom Connector response
Microsoft Dynamics 365
3
4
Platform
Action
Web API

Outbound Async. Process



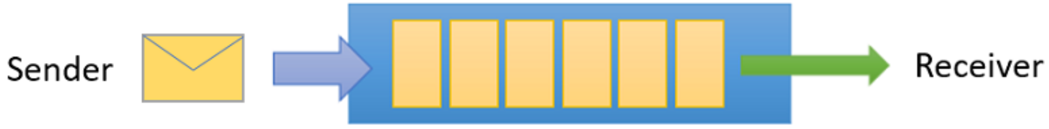
Dynamics 365 business Event occurs

Reflect event to external consumers

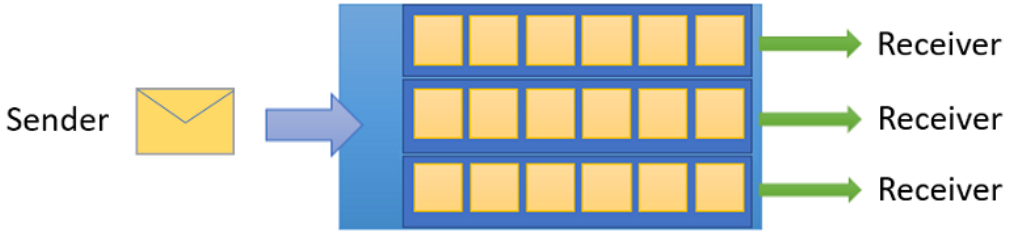
Outbound Async. Process

Azure Service Bus

- Decouple applications and services
- Enterprise integration message broker
- Out of the box integration with Dynamics 365 Plugin
- Diagnostic Logs

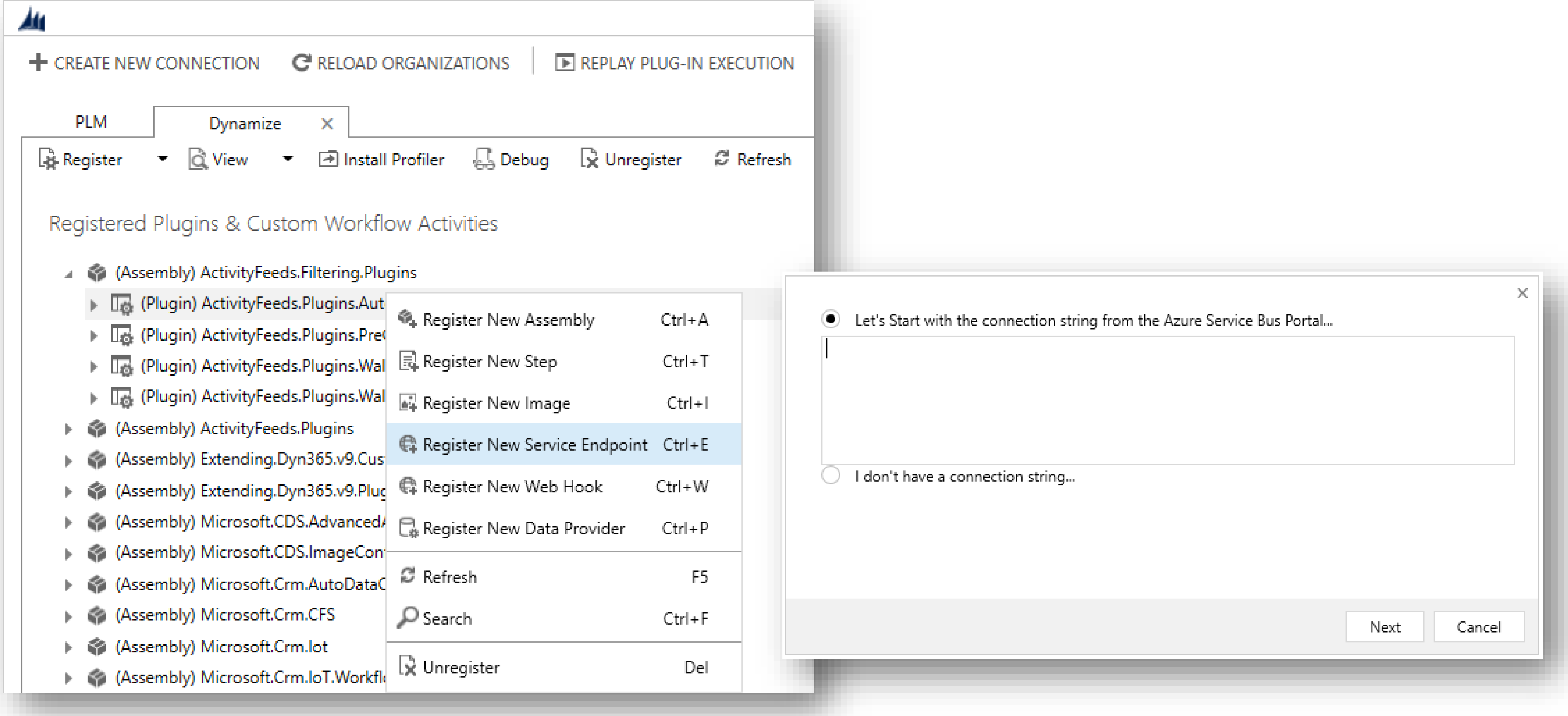


Message Queue with Messages



Topic with three Subscriptions with Messages

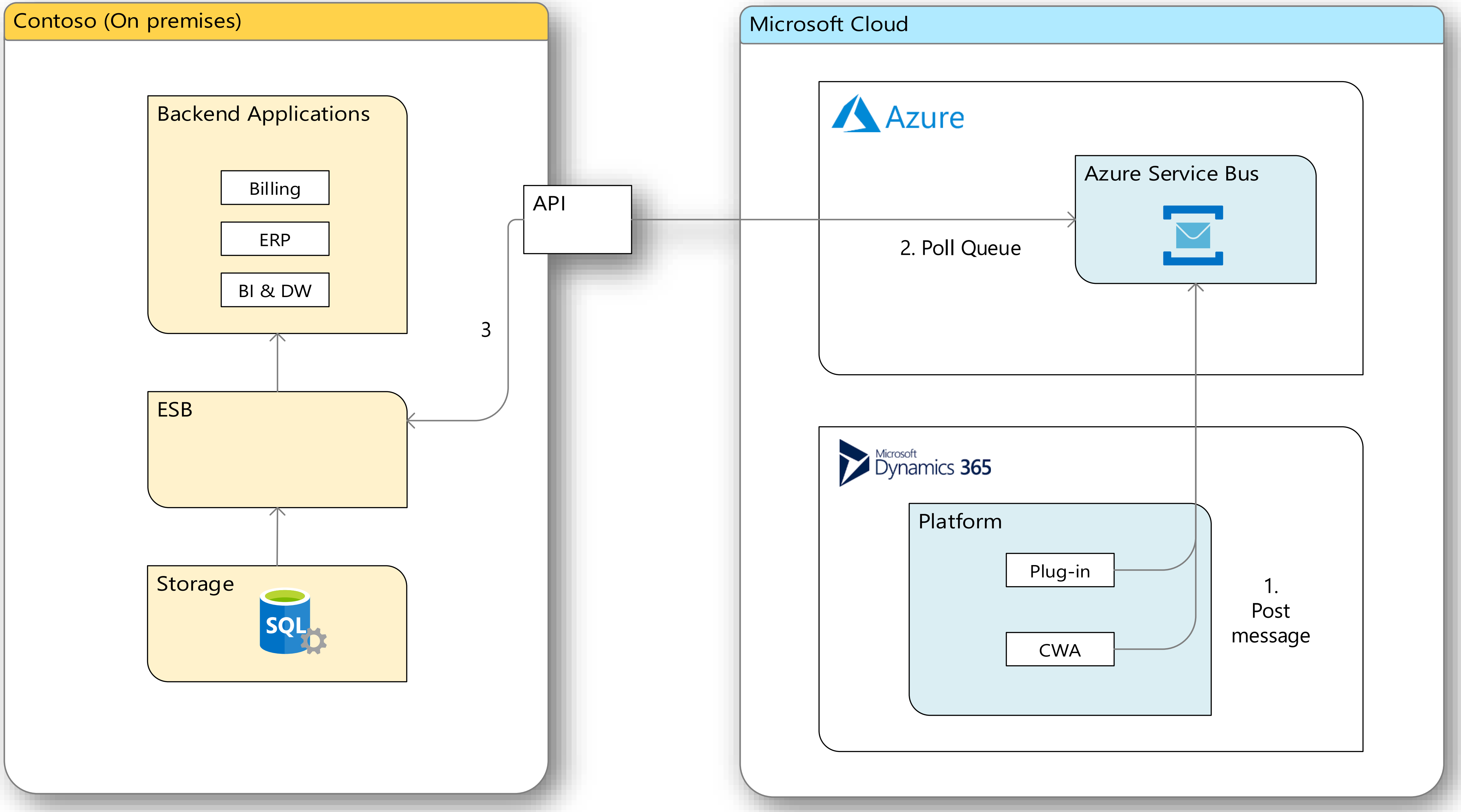
Outbound Async. Process



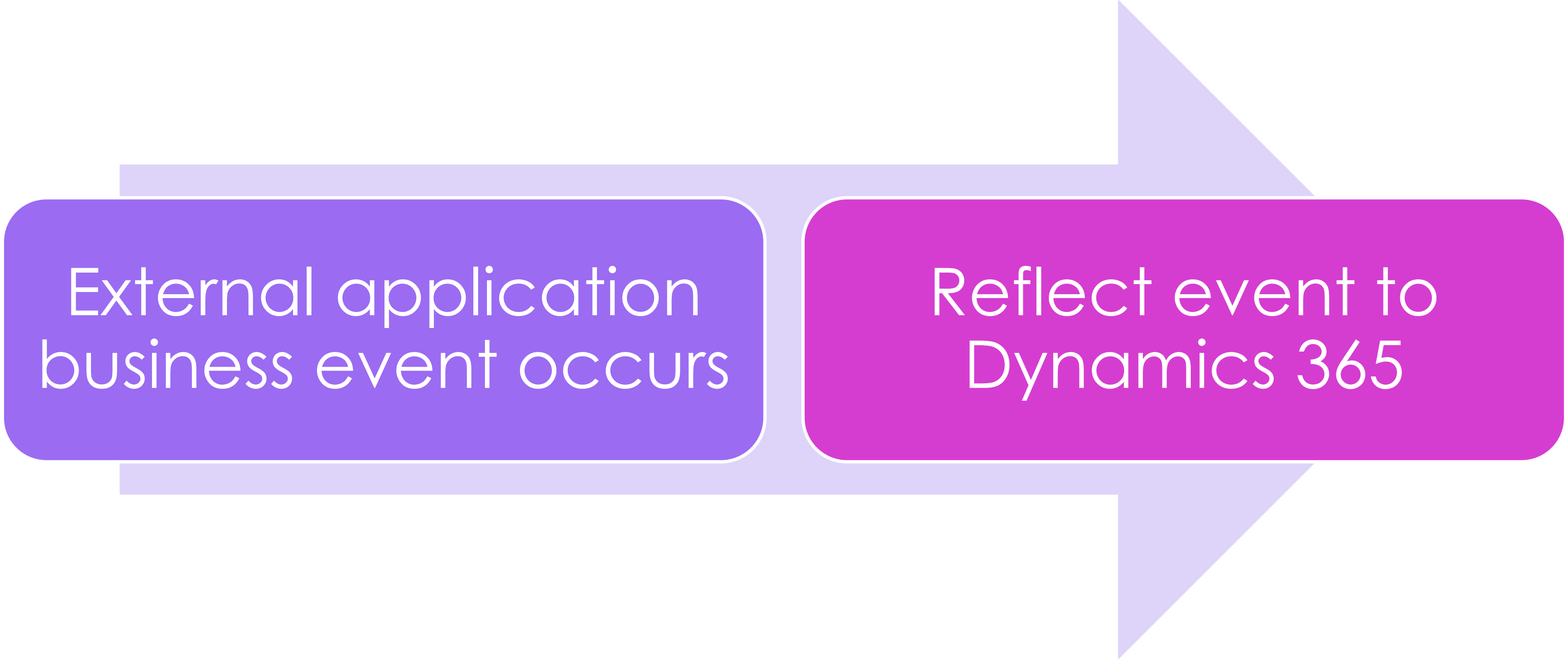
CREATE NEW CONNECTION
RELOAD ORGANIZATIONS
REPLAY PLUG-IN EXECUTION
PLM
Dynamize
Register
View
Install Profiler
Debug
Unregister
Refresh
Registered Plugins & Custom Workflow Activities
(Assembly) ActivityFeeds.Filtering.Plugins
(Plugin) ActivityFeeds.Plugins.Aut
(Plugin) ActivityFeeds.Plugins.Pre
(Plugin) ActivityFeeds.Plugins.Wal
(Plugin) ActivityFeeds.Plugins.Wal
(Assembly) ActivityFeeds.Plugins
(Assembly) Extending.Dyn365.v9.Cus
(Assembly) Extending.Dyn365.v9.Plug
(Assembly) Microsoft.CDS.Advanced
(Assembly) Microsoft.CDS.ImageCon
(Assembly) Microsoft.Crm.AutoData
(Assembly) Microsoft.Crm.CFS
(Assembly) Microsoft.Crm.Iot
(Assembly) Microsoft.Crm.IoT.Workfl
Register New Assembly Ctrl+A
Register New Step Ctrl+T
Register New Image Ctrl+I
Register New Service Endpoint Ctrl+E
Register New Web Hook Ctrl+W
Register New Data Provider Ctrl+P
Refresh F5
Search Ctrl+F
Unregister Del
Next
Cancel

● Let's Start with the connection string from the Azure Service Bus Portal...

○ I don't have a connection string...

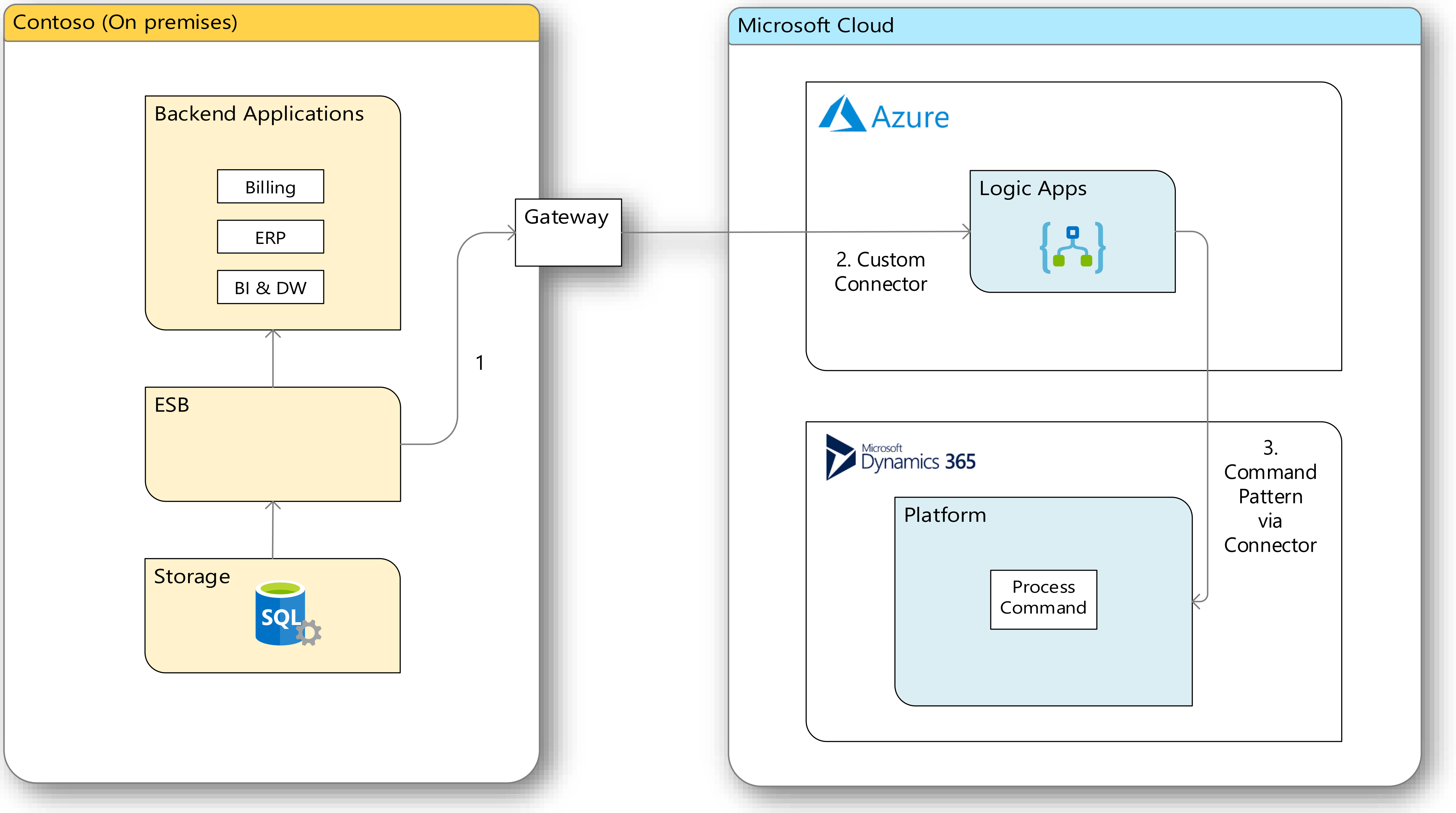

Contoso (On premises)
Backend Applications
Billing
ERP
BI & DW
ESB
Storage
SQL
API
3
Microsoft Cloud
Azure
Azure Service Bus
2. Poll Queue
Microsoft
Dynamics 365
Platform
Plug-in
CWA
1.
Post
message

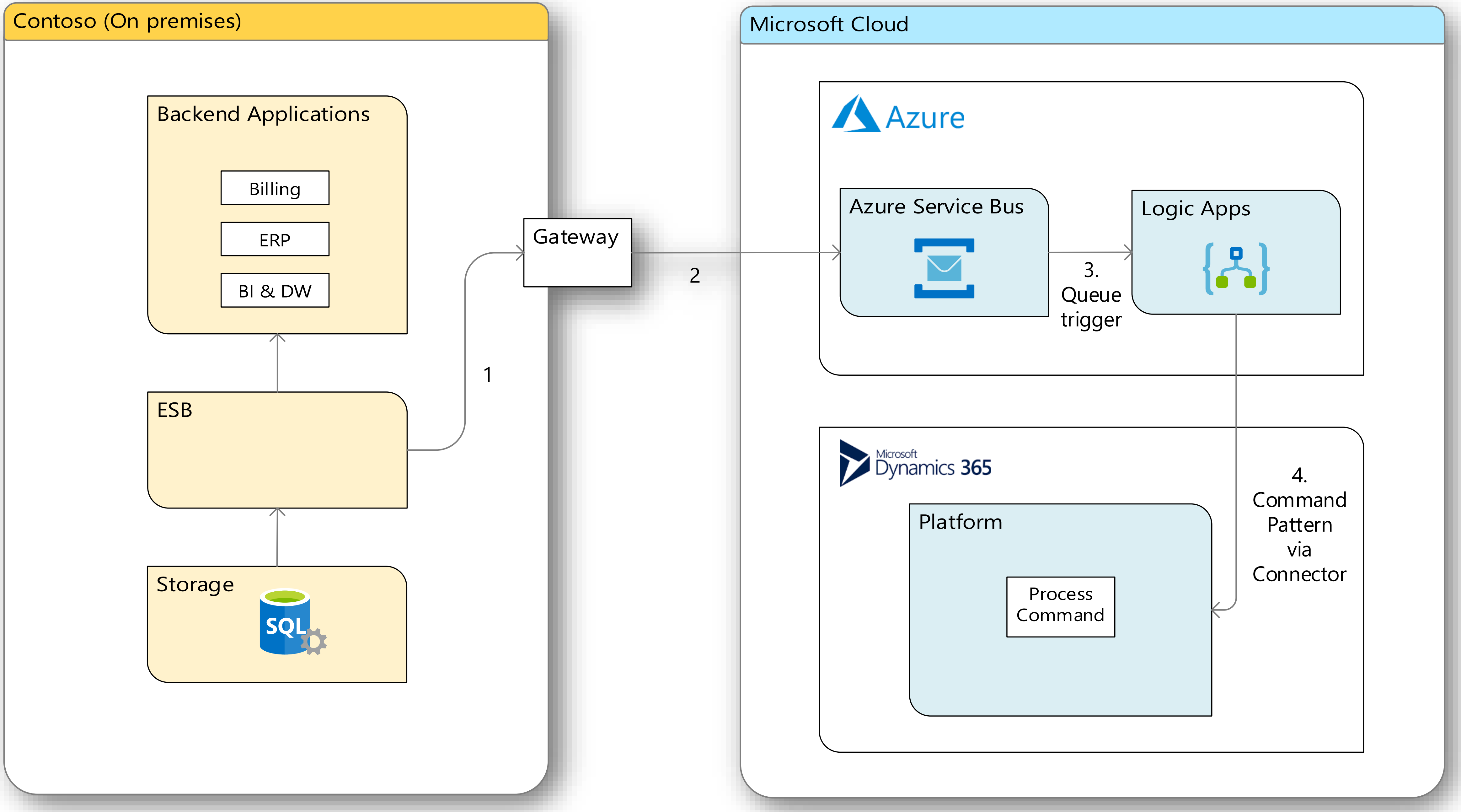
Inbound Async Process

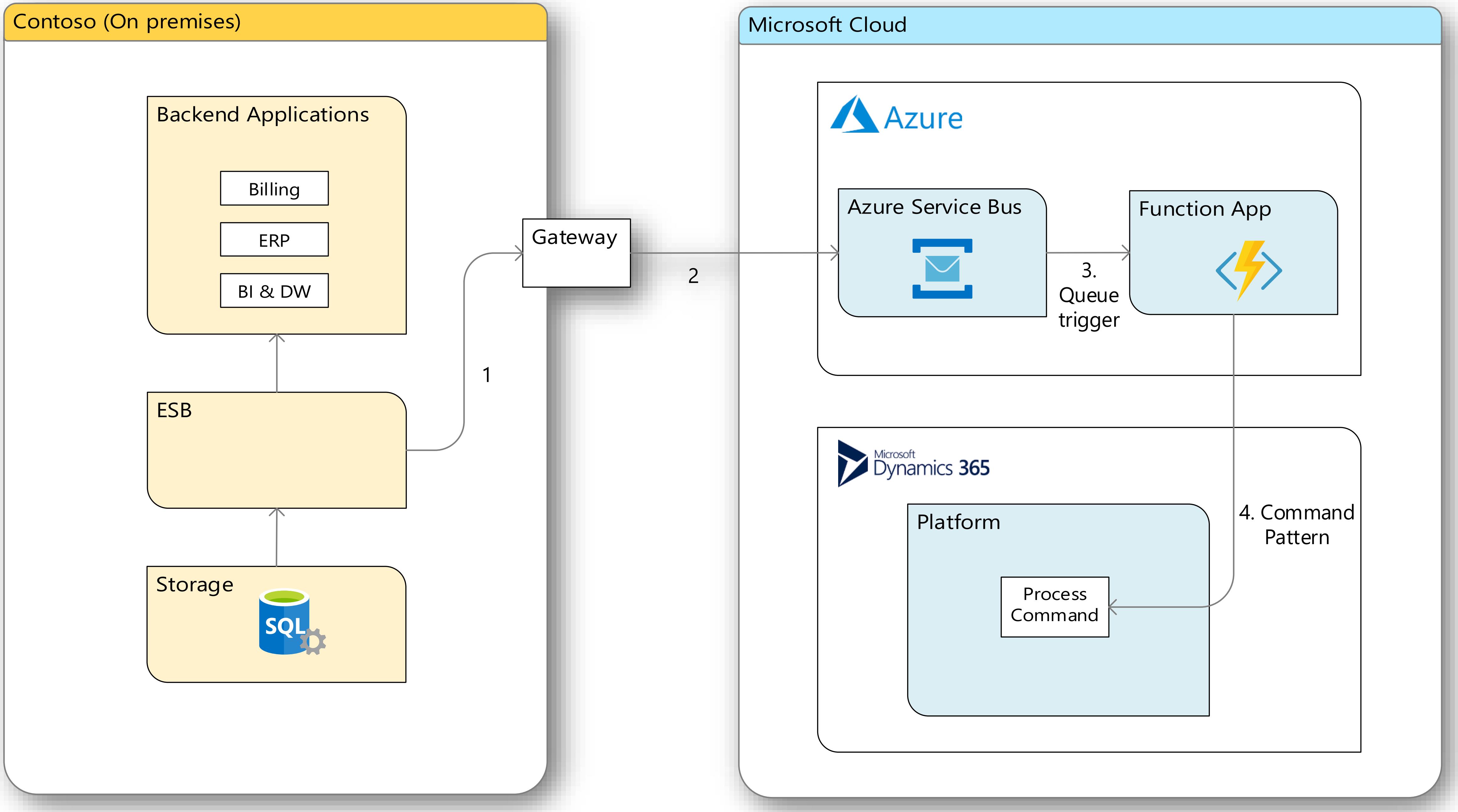


External application business event occurs

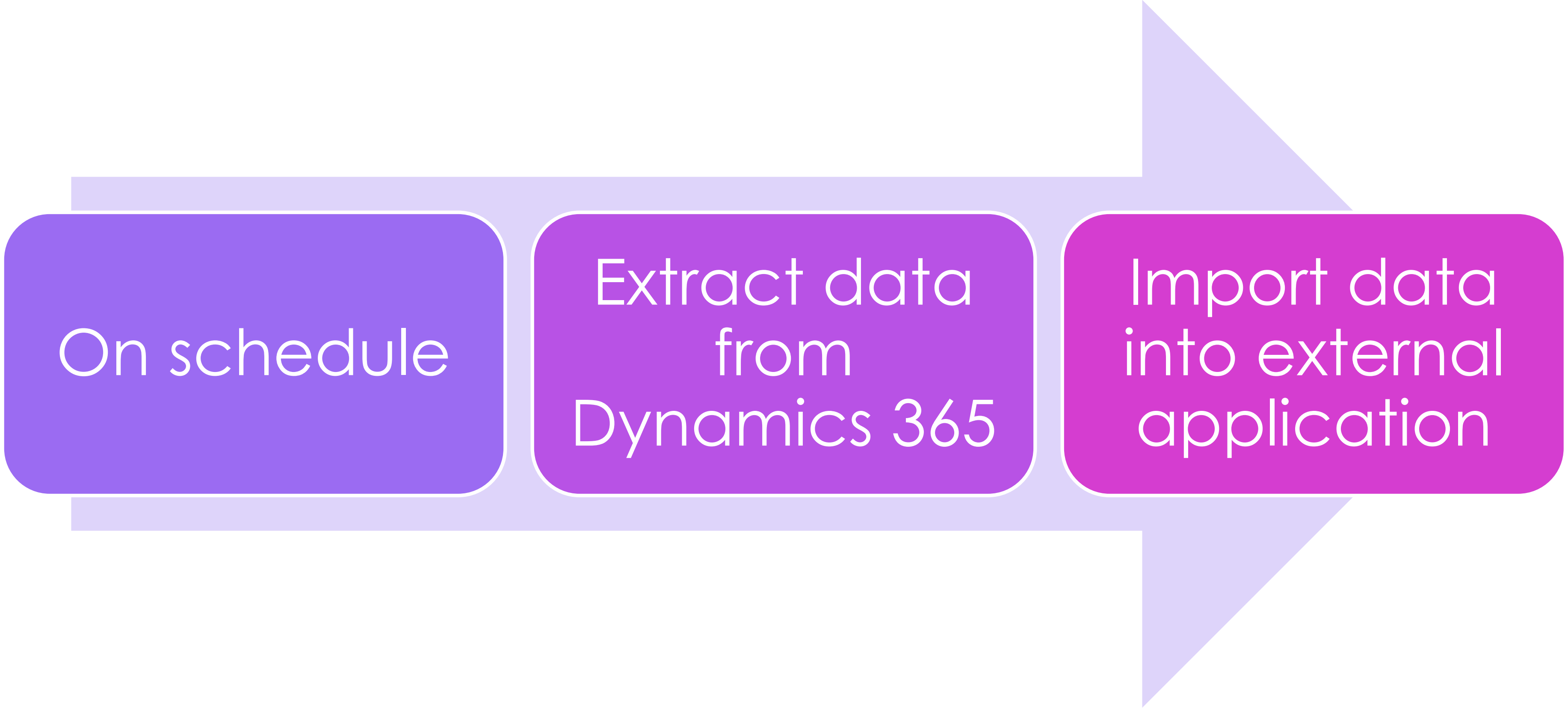
Reflect event to Dynamics 365


Contoso (On premises)
Backend Applications
Billing
ERP
BI & DW
ESB
Storage
SQL
Gateway
1
Microsoft Cloud
Azure
Logic Apps
2. Custom Connector
Microsoft Dynamics 365
Platform
Process Command
3. Command Pattern via Connector


Contoso (On premises)
Backend Applications
Billing
ERP
BI & DW
ESB
Storage
SQL
Gateway
1
2
Microsoft Cloud
Azure
Azure Service Bus
3.
Queue
trigger
Logic Apps
Microsoft
Dynamics 365
Platform
Process
Command
4.
Command
Pattern
via
Connector


Contoso (On premises)
Backend Applications
Billing
ERP
BI & DW
ESB
Storage
SQL
Gateway
1
2
Microsoft Cloud
Azure
Azure Service Bus
3.
Queue
trigger
Function App
Microsoft
Dynamics 365
Platform
Process
Command
4. Command
Pattern

Outbound Scheduled Batch Process



On schedule

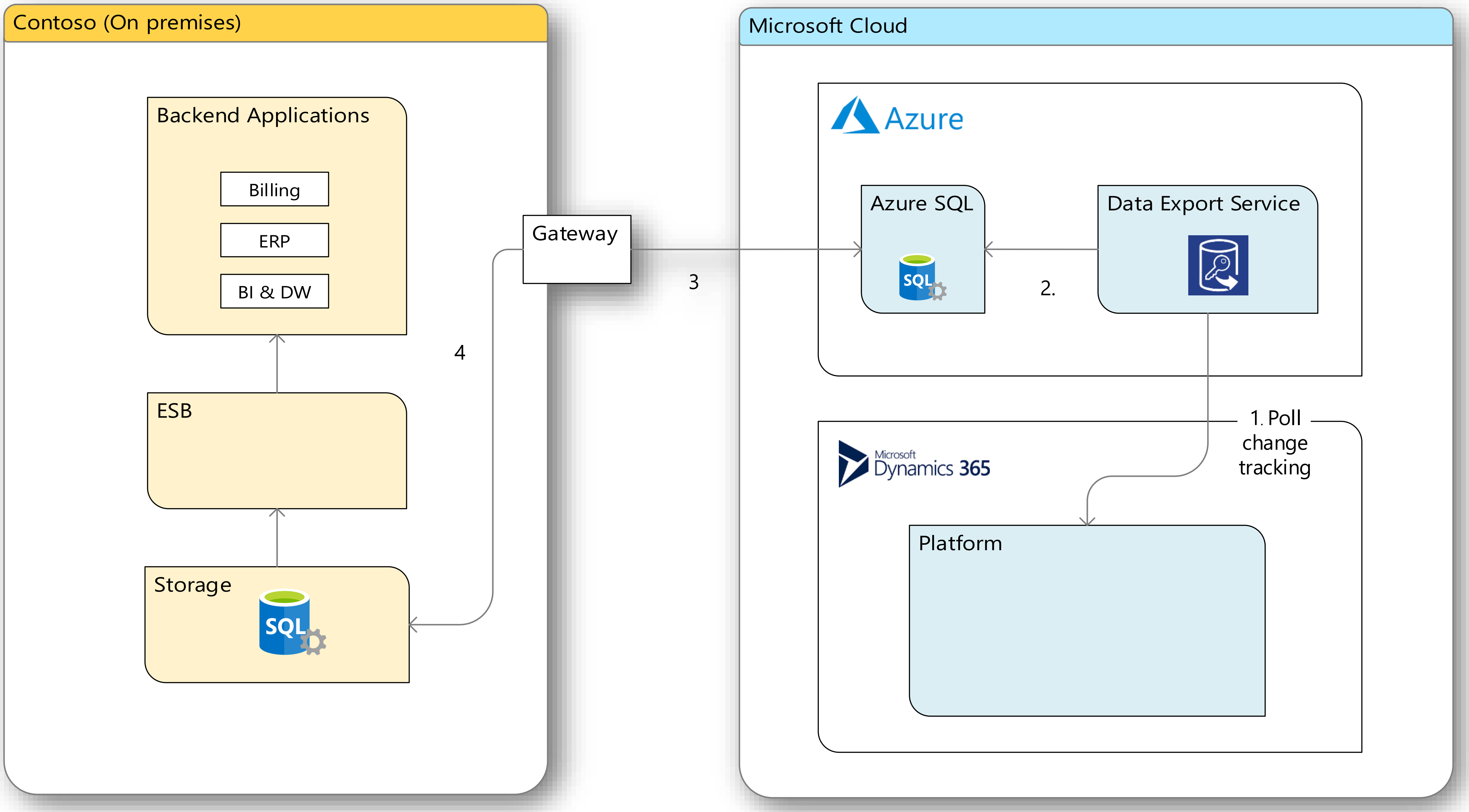
Extract data from Dynamics 365

Import data into external application

Outbound Scheduled Batch Process

Azure Export Service

- Replicate CDS data to a Microsoft Azure SQL store
- Polls on Dynamics 365 using change tracking
- Incremental data pushed in ‘minutes’
- Improve Dynamics 365 performance - reducing analytics queries load
- Azure SQL can be required separately


Contoso (On premises)
Backend Applications
Billing
ERP
BI & DW
ESB
Storage
SQL
Gateway
4
3
Microsoft Cloud
Azure
Azure SQL
SQL
Data Export Service
2.
1. Poll change tracking
Microsoft Dynamics 365
Platform

Questions?

DYNAMIZE

Yaniv Arditi
Microsoft Dynamics Specialist
yaniv@dynamize.onmicrosoft.com