

*8 February 2019*



**Dynamics 365 Saturday Milan**



# Introduction to Virtual Entities for .NET Developers

Giancarlo Lelli - Microsoft MVP & Technical Lead @ Avanade

Riccardo Gregori - CRM TC & Market Unit Delivery Lead @ Avanade



Sponsor Platinum



Sponsor Gold



Sponsor Gold



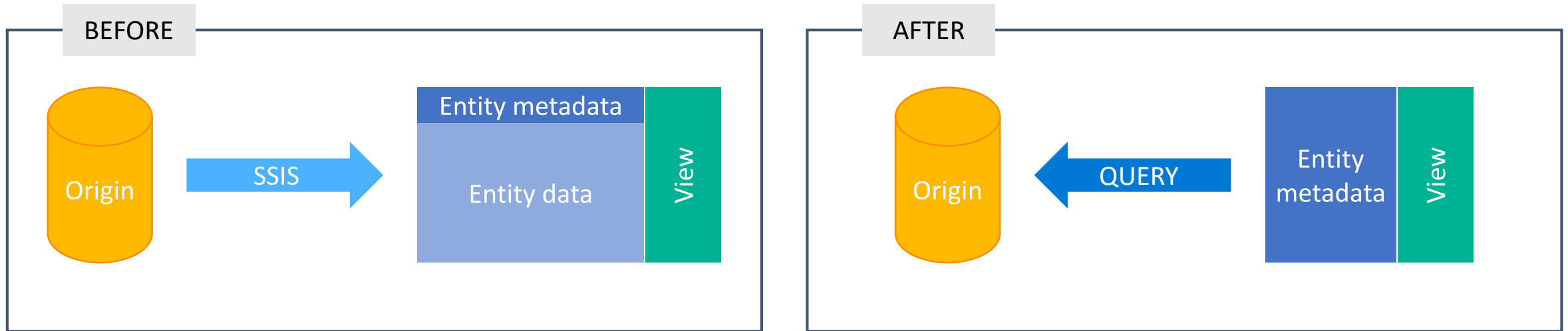
Sponsor Silver



Milan 2019



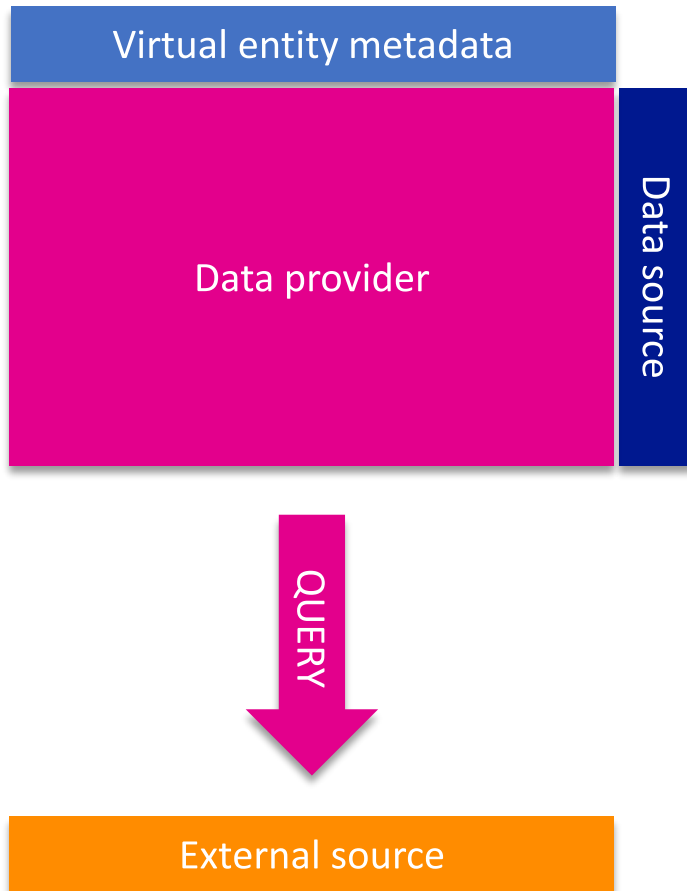
# Virtual Entities 101



Starting with Dynamics 365 9.0, Microsoft introduced **Virtual Entities**, which allows you to integrate data residing in external systems by seamlessly representing that data as entities in Dynamics 365. There are several advantages to this approach:

- You **don't have to create physical tables** for the external instances inside Dynamics 365 database
- This translates into **real-time access** to the data source
- If you have legacy on-prem CRM/ERP tools that store historical information about customers, now you don't have to spend hundreds of hours or thousands of dollars in moving and storing that information inside the Dynamics 365 database.
- And very often, your security team won't allow you to completely move/copy sensitive data outside the firewall in order to comply with security/data privacy laws such as GDPR, PCI, SOX, etc.

# How does it work?



## Virtual entity metadata

- A standard entity definition, marked with the “Virtual entity” flag.
- CRM does not create the entity table, just the entity metadata

## Data source

- A record of the EntityDataSource table containing connection info for an external data source

## Data provider

- Plugin registered on **Retrieve** and **RetrieveMultiple** events on the Virtual Entity
- Takes connection info from the related Data source, and converts the event query in a valid, external source query.

# Virtual Entities 101 - Limitations

## Features

- Read-only
- Auditing and change tracking not supported
- Export not supported
- Cannot be an activity
- No business process flows
- Cannot be enqueued
- Creating an Advanced Find that creates a join between the Common Data Service for Apps native data and the virtual entity external data isn't supported

## Security

- Only organization-owned (no default security model)
- Field-level security not supported

## Model

- Primary key should be a GUID.
- All entity properties must be represented as Dynamics 365 for Customer Engagement apps attributes.
- You must be able to model any entity relationships in Dynamics 365 for Customer Engagement apps.
- No calculated or rollup fields.
- No custom fields with the Currency, Image, or Customer data types.
- By default, virtual entities contain only a Name and Id field. No other system managed fields, such as Status or Created On/Modified On are supported



# Data providers

A data provider is a particular type of Dynamics 365 plugin, which is registered against CRUD events that occur in the platform. This initial release only supports READ operations. The following data providers ship with Dynamics 365 for Customer Engagement apps version 9.0:

- An OData v4 provider is included with the service and is installed by default.
- An Azure Cosmos DB provider is available from AppSource

OData is an open protocol to allow the creation and consumption of queryable and interoperable RESTful APIs in a simple and standard way. Microsoft has its own OData Stack: <https://odata.github.io>

***You can create your own!***

# Writing a custom Data provider

Custom data providers require substantial development resources to create and maintain.

You must have fundamental knowledge of the following areas:

- The external data source schema and data access techniques
- Dynamics 365 for Customer Engagement apps metadata schema
- Dynamics 365 for Customer Engagement apps event system
- Dynamics 365 for Customer Engagement apps plug-in architecture



# Categories of providers

There are two general categories of data provider you can create using the virtual entity data SDK assemblies.

Provider type	Description	Pros/Cons
Generic	Can be reused for all instances of this data source type. This approach is the most general but is more complicated to develop. If the schema of the data source changes, the affected virtual entities must only be remapped.	<p>↑ Flexible, configurable, reusable</p> <p>↓ Complex to develop</p>
Targeted	Only narrowly translates queries into the associated LINQ call to a known, existing data source instance. This approach is limited to a specific data source instance, but requires much less coding. If the schema of the data source changes, the data provider must be updated and rebuilt	<p>↑ Less build effort</p> <p>↓ Specific purpose</p>

# Provider / Plugin Development

There are several steps that are required to create a virtual entity data provider solution that can be imported into your Dynamics 365 for Customer Engagement apps:

1. Develop the custom data provider plug-in DLL (or set of DLLs).
2. Register the custom data provider with your Dynamics 365 for Customer Engagement apps service using the Plug-in Registration Tool
3. Create a data provider solution.
4. Customize the data source entity to reflect your data type or specific instance.
5. Export the custom data provider solution.

Because virtual entities in this release are read-only, you will write the data provider in the form of a plug-in registered on the **Retrieve** and **RetrieveMultiple** events. Each respective event will include information in the execution context which describes the kind of data to return.

- **Retrieve**: Describes which entity to retrieve as well as the attributes and any related entities to include
- **RetrieveMultiple**: Contains a QueryExpression object defining the query. The framework contains a QueryExpressionVisitor class designed to inspect different parts of the query expression tree.

# Provider / Plugin Development

If for any reason your code cannot achieve the expected result, you must throw the appropriate error, [as described here](#). See [Microsoft.Xrm.Sdk.Data.Exceptions](#).

## Plug-in registration

- Unlike an ordinary plugin, you will only use the *Plugin Registration Tool* (PRT) to register the assembly and the plugins for each event.
- Your plugin will run in stage 30, the main core transaction stage for the operation that is not available for ordinary plugin steps.
- You will not register specific steps.
- Instead of registering steps, you will configure your data provider using the [Microsoft.Dynamics.CRM.EntityDataProvider](#) and [Microsoft.Dynamics.CRM.EntityDataSource](#) entities through *Plugin Registration Tool v9*, [as described here](#).

# Building a simple Data Provider

Wide World Importers inside Dynamics 365 CE

# Microsoft Cosmos DB Data Provider

Preview feature. Do not use in production systems!

## Best practices and limitations

- When you use Azure Cosmos DB as a Data Source:
  - Each **Azure Cosmos DB Data Source** can only be associated with a single virtual entity.
  - You can connect multiple Data Sources to the same Collection in the Azure Cosmos DB.
- You can't segment the data in a collection by entity.
- Azure Cosmos DB databases do not require a schema, however **the data within the Azure Cosmos DB must be structured using a predictable schema.**
- Although the Azure Cosmos DB for SQL API Data Provider implements query translation of projection, filtering, and sorting operators, it **does not support join operations.**
- You can only **filter by a single column** with SQL API.

# Avanade Dynamics 365 Modern Data Connector

How we can integrate MS & Non-MS Data sources efficiently inside Dynamics 365 CE?

## Solution

### We didn't reinvent the wheel

- We use Virtual Entities to serve the data to the end users and maintain a consistent UX
- OData for filtering, paging, sorting & selecting data
- NHibernate for relational DB access (SQL, DB2, Firebird, Jet, MySQL, SQLite, Oracle, Postgres)

## Strengths

### Our main differentiators

- **Cross platform.** It can run on both Windows & Linux servers
- Support for **MS & non MS** database
- Extensible security model. **Not implemented by MS**
- Deterministic GUIDs. **Not implemented by MS**

## Limits

### Our upcoming challenges

- Same road blocker as the founding tech: Virtual Entities
- Upgrades distribution
- Keep feature parity
- Still some configuration needed on the CRM side

**Giancarlo Lelli** - Microsoft MVP & Technical Lead @ Avanade

**Riccardo Gregori** - CRM TC & Market Unit Delivery Lead @ Avanade

Thank you