Microsoft Dynamics 365

What's new in Virtual Table for Microsoft Dataverse

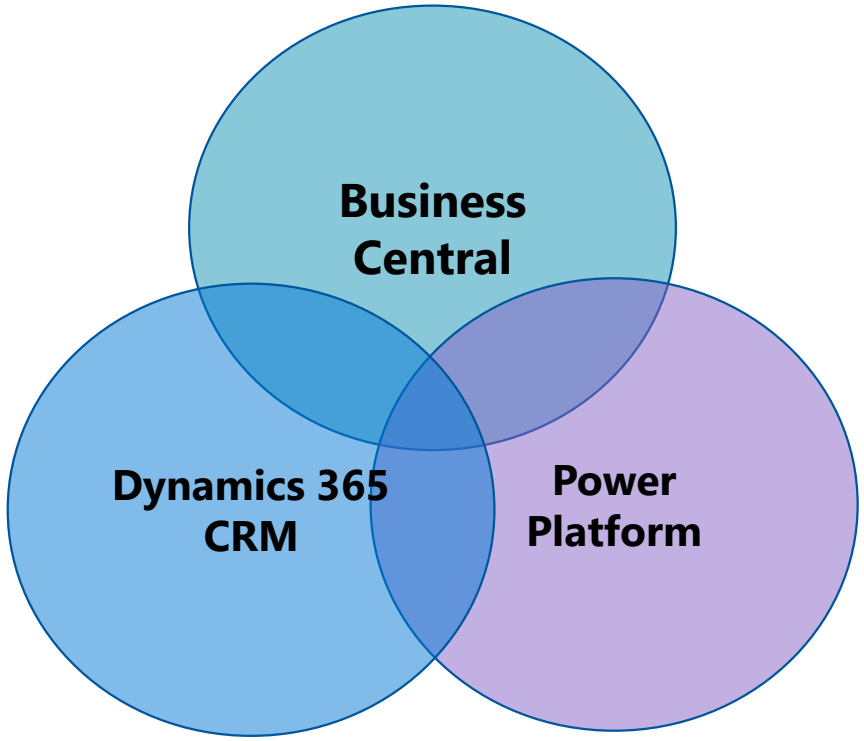
Agenda

- Review of Dataverse integration
- Virtual Table improvements
- Introducing Business Events
- Demos

Review of Dataverse integration

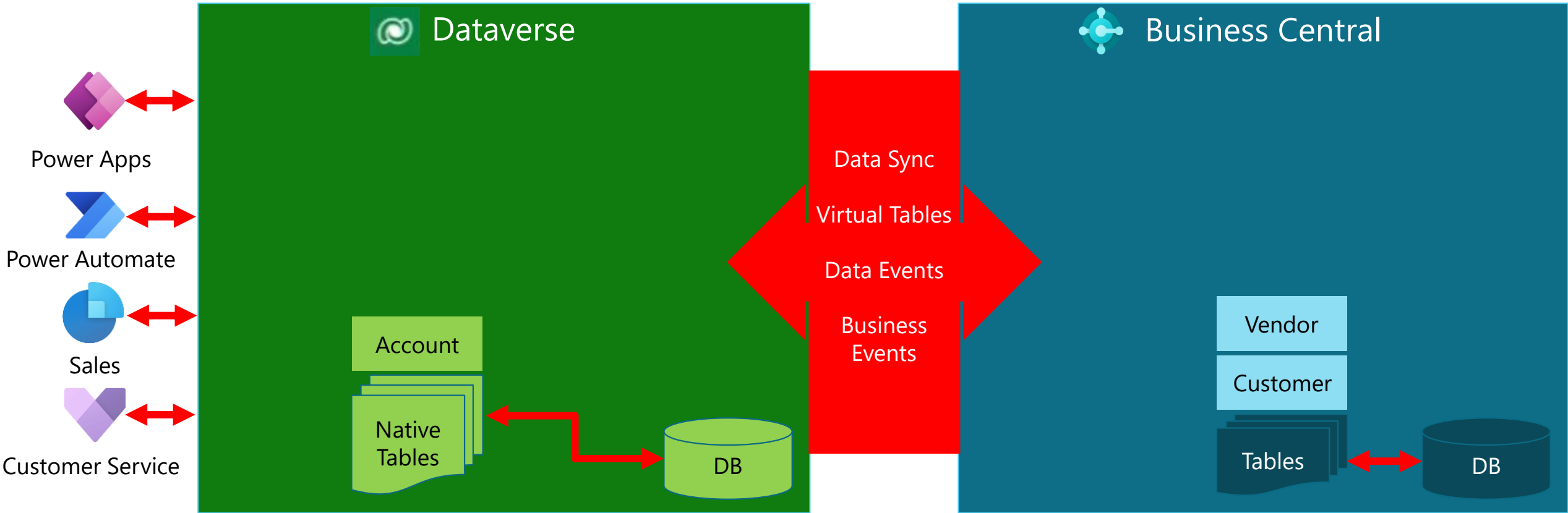
Review of Dataverse integration

- The **product mix** of Business Central, Dynamics 365 CRM (Sales/Marketing/Field Service/Customer Service), and Power Platform is often used together in complex **multi –app/role** business processes, such as Lead/Prospect/Opportunity -to-Cash
- Both Dynamics 365 CRM and Power Platform run on **Dataverse**, so integrating w/ Dataverse allows Business Central to **interact** w/ apps in its ecosystem
- The wider/deeper their interactions,
  - the stronger/tighter their integration will be,
    - the higher the value of product mix will be (“1+1+1 > 3”),
      - the stickier they will attach to each other,
        - the easier their cross-promotion will be,
          - the more customers will use them together



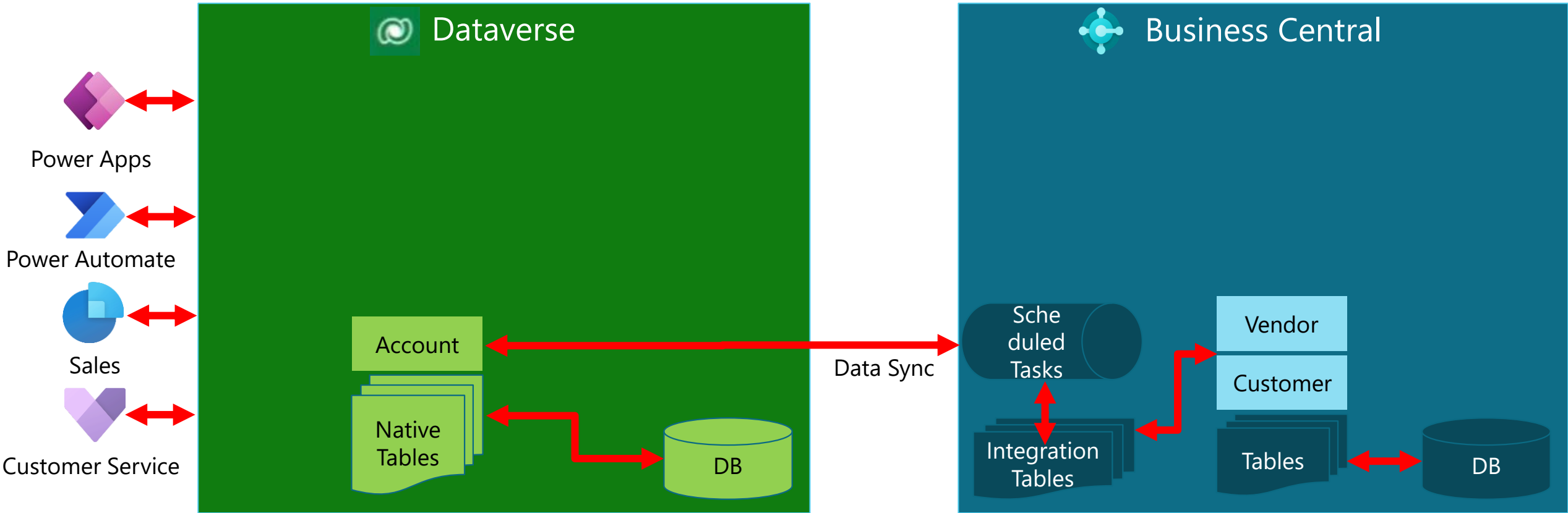
Review of Dataverse integration

- Dataverse is a **data storage & management layer** for Dynamics 365 CRM and Power Platform
  - Dataverse Integration allows Business Central to **interact** w/ apps in its ecosystem
  - There are four types of **complementary** interactions:



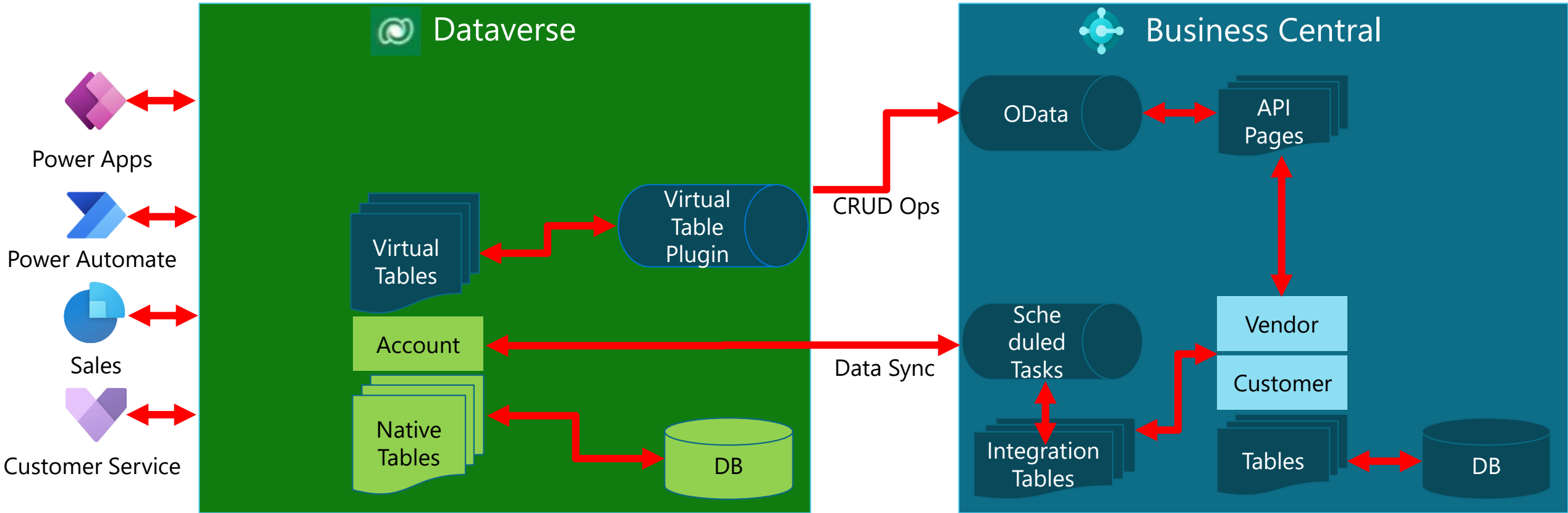
Review of Dataverse integration

1. **Data Sync** that **replicates data changes (CUD)** between **overlapping tables** in Business Central and Dataverse
   - Replication is uni/bi –directional via **scheduled** Service-to-Service calls **w/o authenticated user** and data is **duplicated** in both Business Central and Dataverse
   - Trigger **Power Automate flows** to run scheduled tasks that replicate Dataverse data changes **just-in-time** into Business Central (**21.0**)
   - Bidirectionally replicate **sales orders** between Business Central and Dynamics 365 Sales (**21.0**)



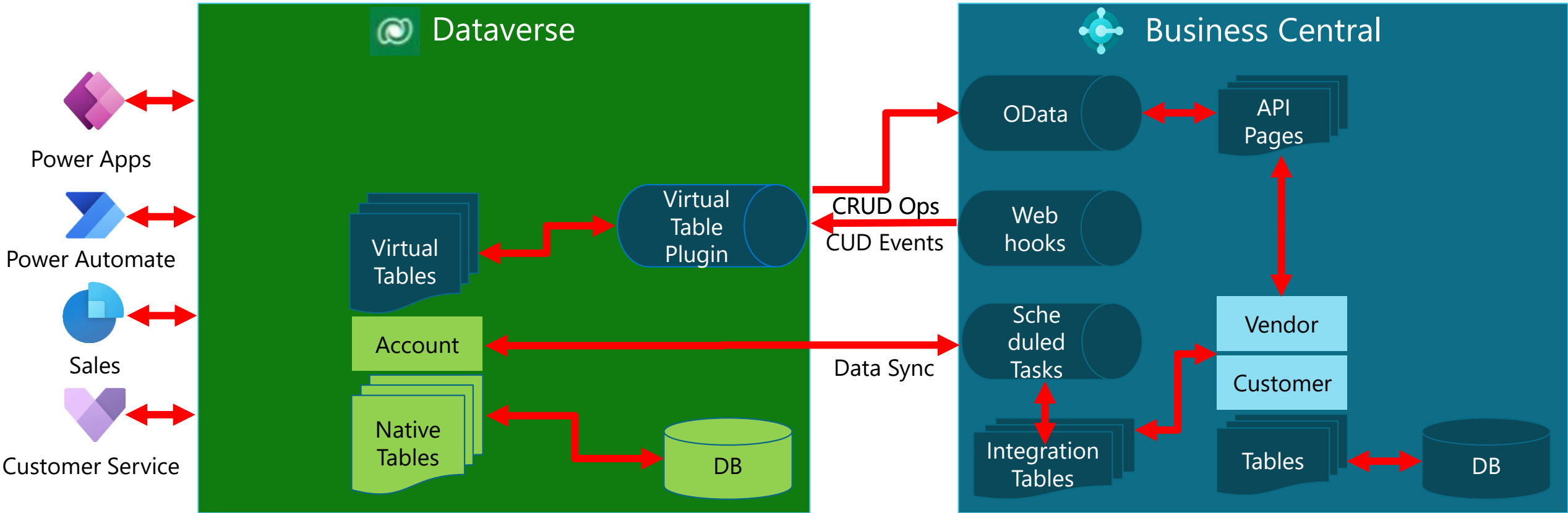
Review of Dataverse integration

2. **Virtual Tables** that enable **inbound interactions** from Dataverse into Business Central **w/o duplicating data**
   - **Virtual Table plugin** translates Dataverse CRUD API targeting virtual tables into Business Central CRUD API targeting physical tables in **near real-time** via Service-to-Service calls **on behalf of authenticated users**
   - [**New**] **Customize** virtual tables (by changing labels/icons/precisions) on model-driven Power Apps (**22.0**)
   - [**Coming Soon**] Access virtual tables via **Power Pages** (**Preview in 22.x**)
   - [**Coming Soon**] Use virtual tables in **Production** (**GA in Q2/Q3CY23**)



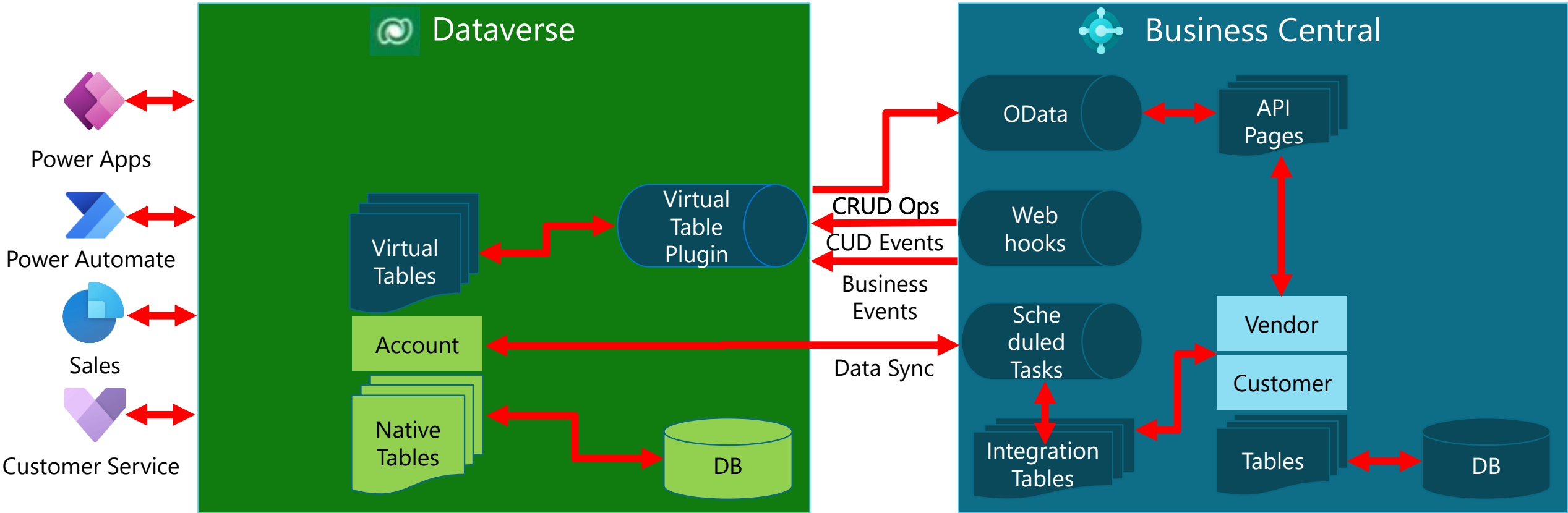
Review of Dataverse integration

3. **Data (CUD) Events** that enable **outbound interactions** from Business Central into Dataverse, where they can start **Power Automate flows** via the "**When a row is added, modified, or deleted**" trigger
   - [**Coming Soon**] Enhanced **payloads** (by including ChangedFields/pre-image/post-image) for data (CUD) events (**Preview in 22.x**)
   - [**Coming Soon**] Use data (CUD) events in **Production** (**GA in Q2/Q3CY23**)



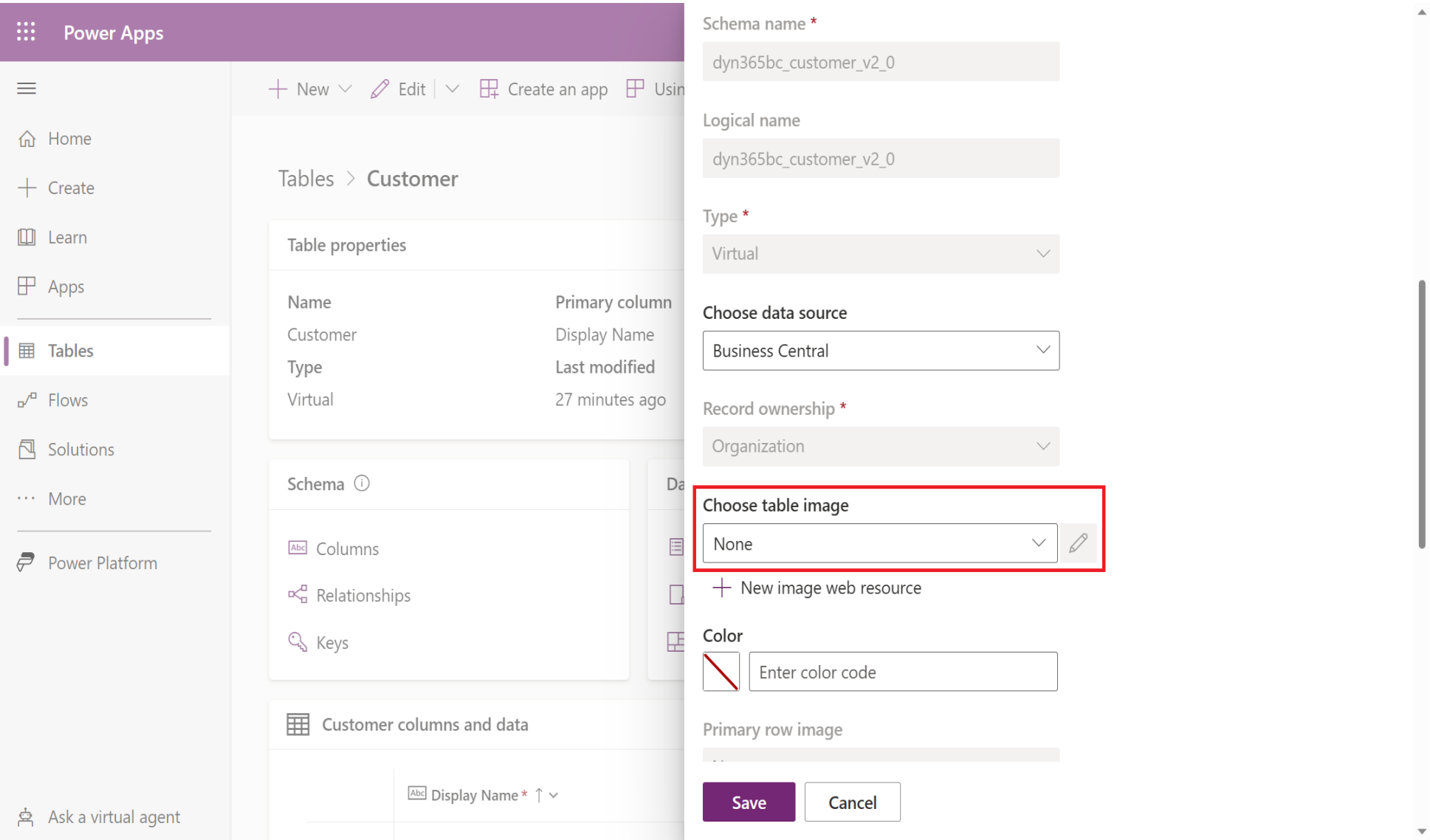
Review of Dataverse integration

4. **Business Events** that enable **outbound interactions** from Business Central into Dataverse, where they can start **Power Automate flows** via the "**When an action is performed**" trigger
   - [**New**] **Private Preview** of business events (**22.0**)
   - [**Coming Soon**] **Public Preview** of business events (**22.1/22.2**)



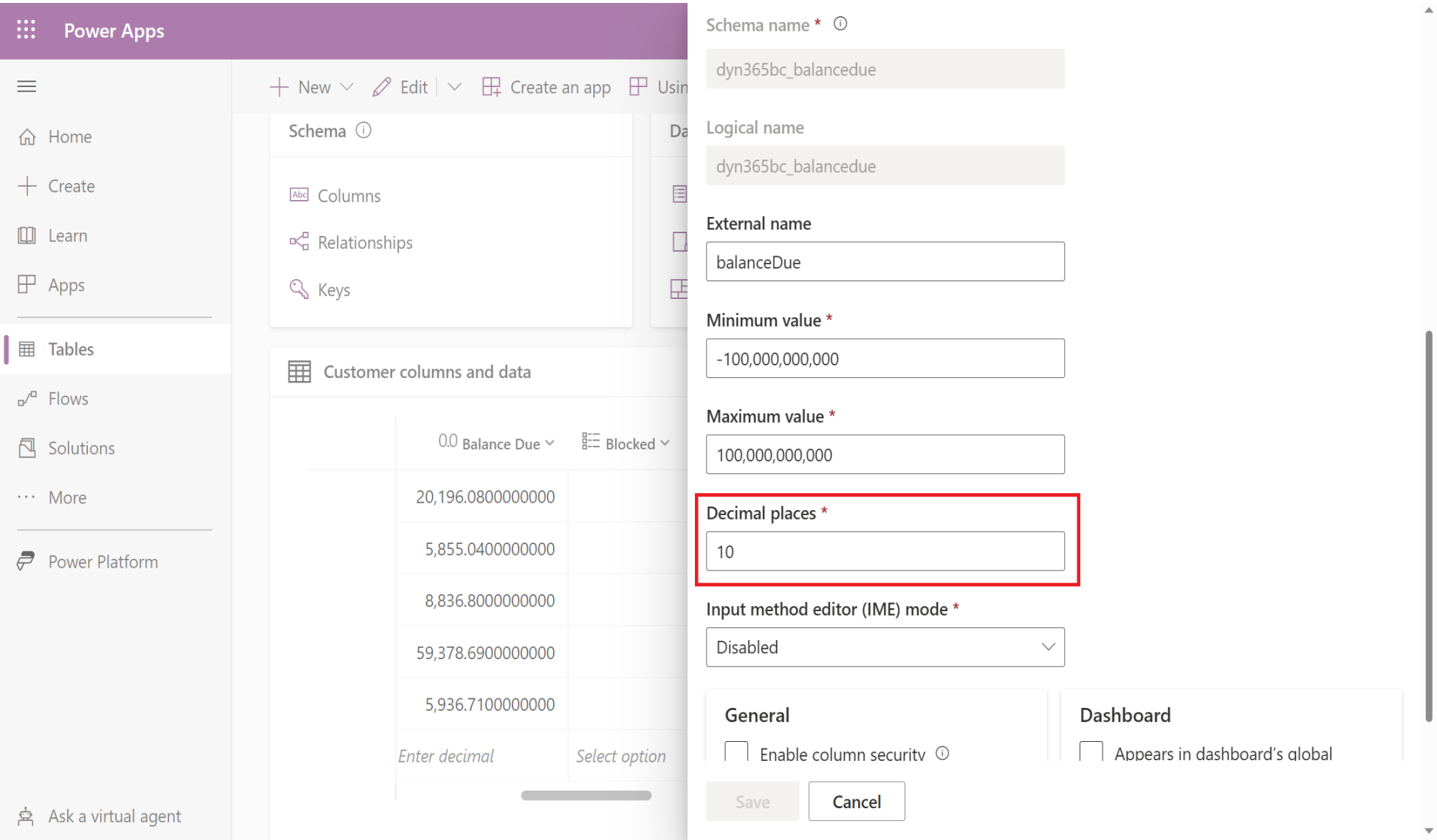
Virtual Table improvements

Virtual Table improvements – customizations



- Change icon

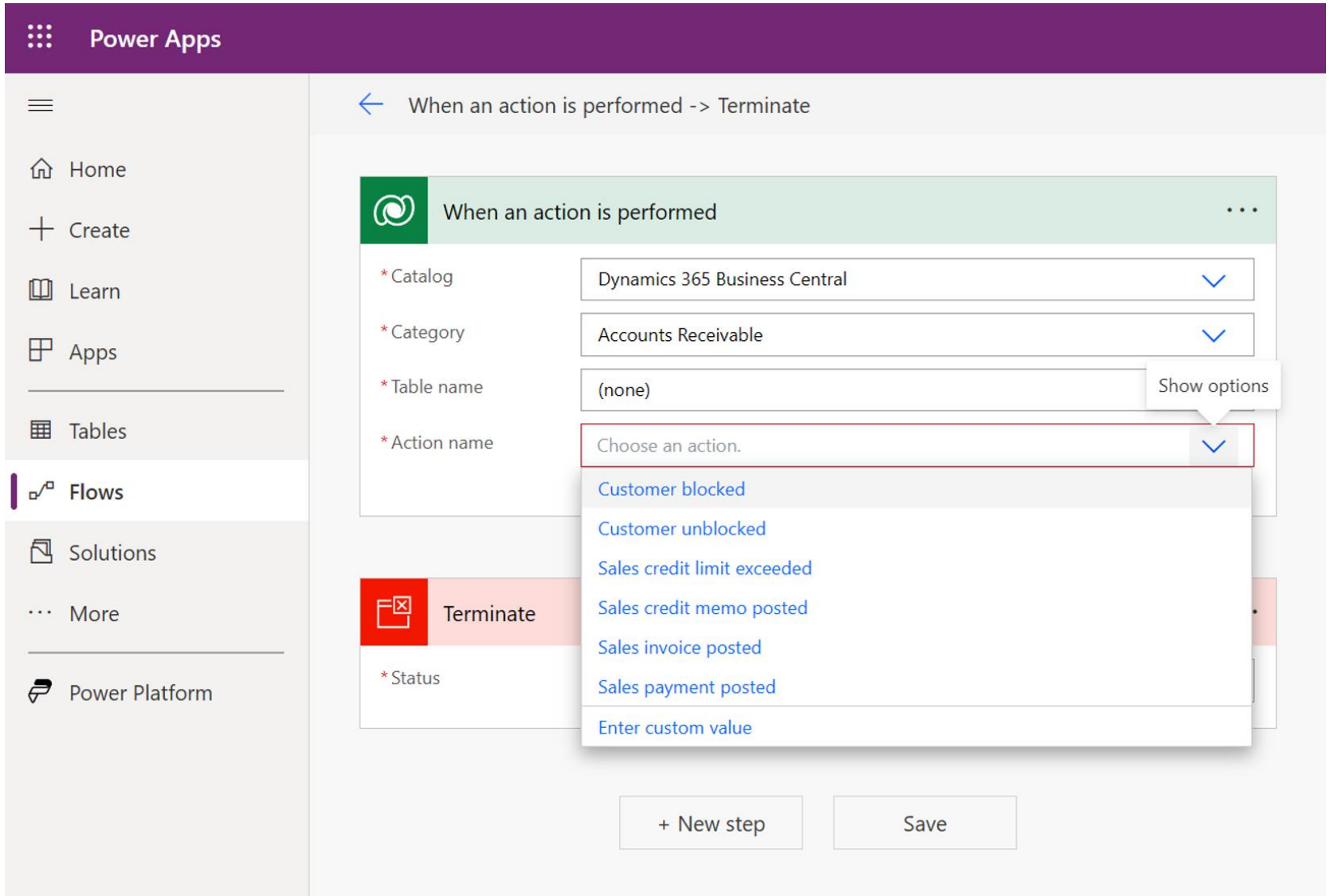
Virtual Table improvements – customizations



- Change precision

Introducing Business Events

Introducing Business Events – Power Automate trigger



Introducing Business Events – AL code syntax

```
enumextension 50101 MyEnumExtension extends EventCategory
{
    value(0; "Sales")
    {
    }
}

codeunit 50102 MyCodeunit
{
    trigger OnRun()
    begin
    end;

    [EventSubscriber(ObjectType::Page, Page::"Sales Order", 'OnPostDocumentBeforeNavigateAfterPosting', '', true, true)]
    local procedure OnPostDocument(var SalesHeader: Record "Sales Header"; var PostingCodeunitID: Integer; var Navigate: Enum "Navigate After
Posting"; DocumentIsPosted: Boolean; var IsHandled: Boolean)
    begin
    SalesOrderPosted(SalesHeader.SystemId, SalesHeader."Sell-to Customer Name", SalesHeader."No.");
    end;

    [ExternalBusinessEvent('salesorderposted', 'Sales order posted', 'Triggered when sales order has been posted', EventCategory::"Sales")]
    [RequiredPermissions(PermissionObjectType::TableData, Database::"Sales Header", 'R')] // optional
    procedure SalesOrderPosted(salesOrderId: Guid; customerName: Text; orderNumber: Text)
    begin
    end;
}
```

Introducing Business Events – get catalog

- GET api/microsoft/runtime/v1.0/externalbusinesseventdefinitions


```
"value": [
    {
          "appId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
          "name": "salesorderposted",
          "payload": "[{\"Index\":0,\"Name\":\"salesOrderId\",\"Type\":\"Guid\"},{\"Index\":1,\"Name\":\"customerName\",\"Type\":\"Text\"},{\"I
  ndex\":2,\"Name\":\"orderNumber\",\"Type\":\"Text\"}]",
          "displayName": "Sales order posted",
          "description": "Triggered when sales order has been posted",
          "category": "Sales",
          "appName": “MyApp",
          "appPublisher": "Default publisher",
          "appVersion": "1.0.0.0"
      }
]
```


- Required permissions
  - Any authenticated D365 BC user

Introducing Business Events – subscribe

- POST api/microsoft/runtime/v1.0/externaleventsubscriptions

```
{
    "appId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "companyName": null, (optional)
    "eventName": "salesorderposted",
    "notificationUrl": https://xxxxxxxxxxx.crm4.dynamics.com/api/data/V9.0/PostRuntimeIntegrationExternalEvent,
    "clientState": "" (optional)
}
```

- Required permissions
  - Read access to ExternalBusinessEventDefinition
  - Optional checks defined by RequiredPermissions attribute

Introducing Business Events – notify

- POST https://xxxxxxxxxxx.crm4.dynamics.com/api/data/V9.0/PostRuntimeIntegrationExternalEvent

```
{
    "subscriptionId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx", (Dataverse subscription only)
    "appId": " xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "eventName": "salesorderposted",
    "timestamp": "2023-02-20T10:27:35.8770000Z",
    "initiatingUserAADObjectId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "companyName": "CRONUS International Ltd.",
    "clientState": "",
     "payload": {
         "salesOrderId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
         "customerName": “Acme Insurance Co.",
         "ordernumber": "101005"
       }
  }
```

- Verifications before sending notifications
  - Subscriber must have access to ExternalBusinessEventDefinition in the current company
  - Subscriber must have access to all RequiredPermissions in the current company

Introducing Business Events – preview planning

- **Private preview (22.0)**
  - Partners will be invited to participate via Yammer
    - Selection will be based on a form that they fill out with their:
      - Scenarios for business events
      - AAD tenant + environment IDs to enable for preview
      - Email contact
  - GitHub repo for Dataverse integration (https://github.com/microsoft/d365bcdv) will be made public to share
    - Docs and sample code for business events, so selected partners can
      - Create their extension
      - Deploy it in their preview-enabled environment
      - Try out business events
      - Submit their ideas/feedbacks/questions/issues
- **Public Preview (22.1/22.2)**

Demos

Thank you