

Embedded Intelligence in Model Driven Apps

Dynamics 365 FastTrack
Architecture Insights Series



Agenda

Adding Intelligence to Apps

- Look at the ability to add intelligence to Model Driven Apps and challenges and gaps

High Level Solution

- Look at a high-level design pattern that can be used to address the gaps and challenges and the benefits of the design

Example Solution

- More detailed look at an example solution built using the high-level design

Important Considerations

- Understanding implications of the sample design for customers/partners



Adding Intelligence to Apps

Adding Intelligence to Apps

- There are a number of out of the box ways in which makers can add intelligence to a model driven application:-

Interactive Copilot

- Customise Prompts
- Add Topics
- Add Knowledge
- Add Tools
- Add Agents

Intelligent Controls

- Natural Language in Grids
- Visualisation in Grid
- Form Fill Assistant
- Copilot in Email Draft

Intelligent Insights

- Row Level Summaries
- Prompt Columns
- Timeline Highlights

Intelligent Automation

- Agent Flows
- 1st Party Agents
- Autonomous Agents based on Triggers

While these are excellent...

User Initiated/Can't adapt

- Know the right prompt
- Know where in the app to go
- Change context to get the right insights
- Can adapt to the current stage of the business process

User Actioned

- Often provides only overviews or summaries
- User must understand the responses and then decide how to act upon the insights

Control of Costs

- Platform triggered generation can make it hard to predict cost
- Harder to quantify the value benefit to users as it can be difficult to know if they are being triggered/used

Ideal Situation

Based on the current state/stage of business process

- Allow flexibility to react and adapt to the user/customer needs as customer journey progresses

Automatically get the right insights/actions

- System should be intelligent enough to understand what needs to be asked removing the burden from the user

Display them in context to the user

- Insights should be brought to the user so they don't have to seek this out

Single click to action them

- User should be able to easily act on what they are being shown



High Level Solution

How do we do this?

Action

If the user decides to act upon the intelligence, then we need something that will execute the actions on the user's behalf

Controller

We need client/server-side logic to be able to determine what Intelligence needs to be brought forth at the right time and to display this to our users

Intelligence

We need access to an Agent or Agents which we can invoke/prompt to generate the insights that will be so useful



What are our options?

Action

Client Side (Handle Action Event):-

JavaScript or PCF Components

Controller

Client Side (Trigger & UI):-

Javascript or PCF Components

Server-Side (Trigger only):-

Agent Flows
Power Automate

Intelligence

Client/Server Side:-

Interactive Agent in the MDA

Another MCS Agent or even a 3rd Party Agent





Example Solution

Solution Example

As a salesperson I want to know what my next best actions are on an open opportunity and have the ability to act on them.

Action

Simple JavaScript handler that based on the insight selected will either update the record for the user or create the recommended email or appointment for them to edit and send

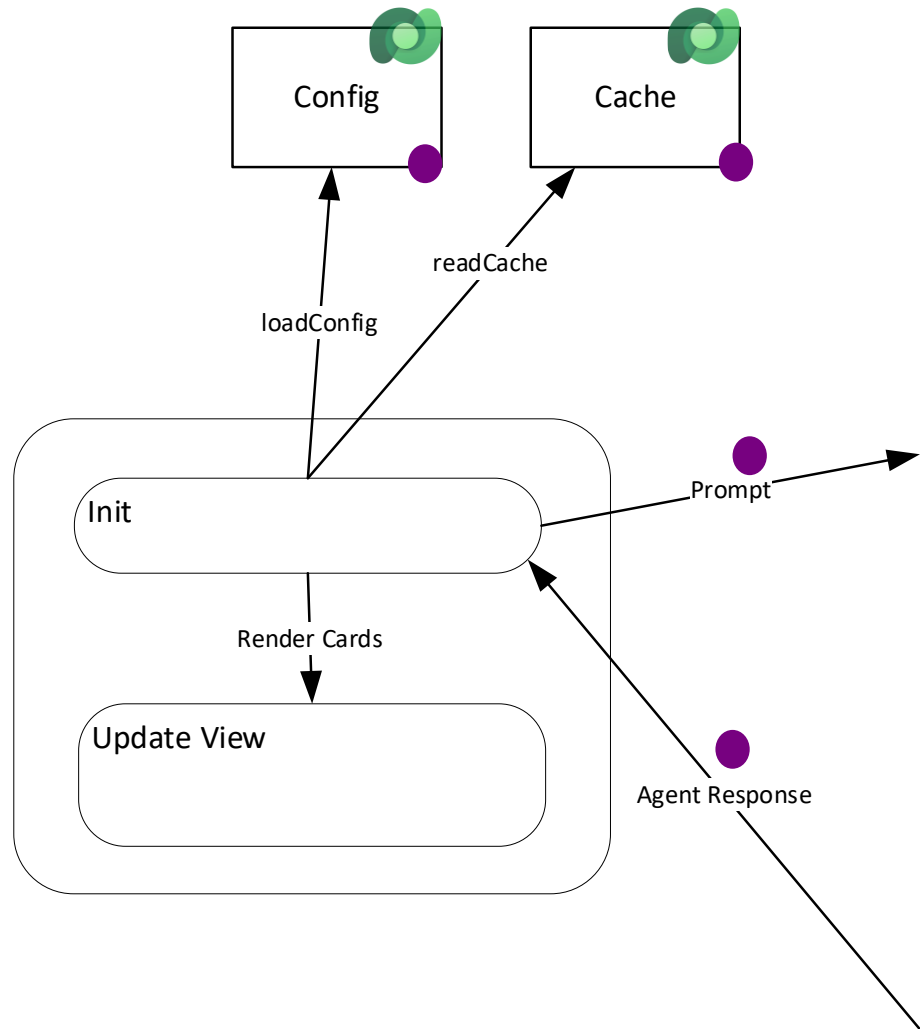
Controller

A reusable PCF that will render insights to the user on the opportunity form and with one click have those insights acted upon

Intelligence

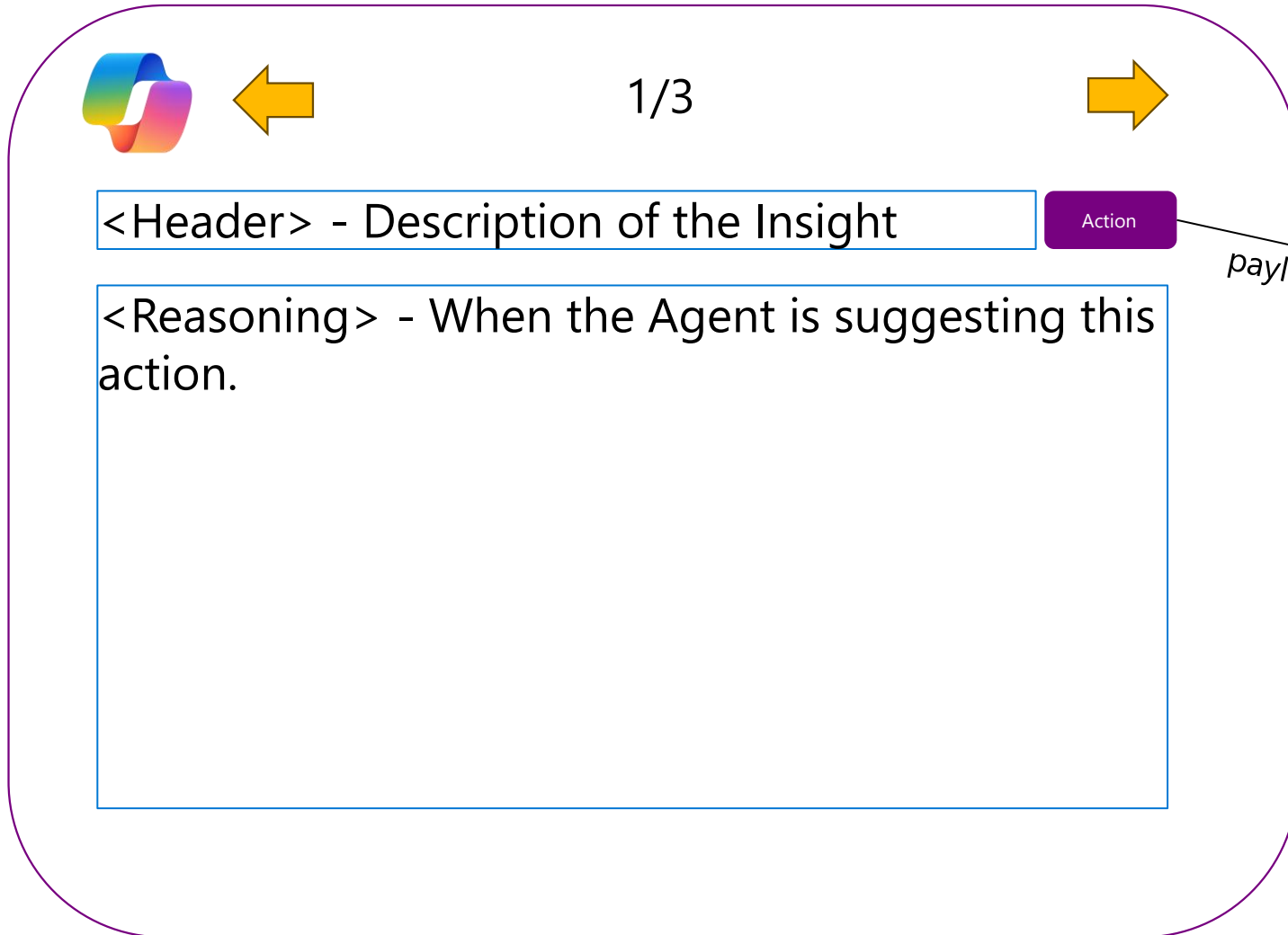
Extend the Interactive Agent that is associated with the MDA. This should get recommendations based on the current data on the record and the last 5 emails with the customer (Data Updates and Email/Appointment Recommendations)

Controller



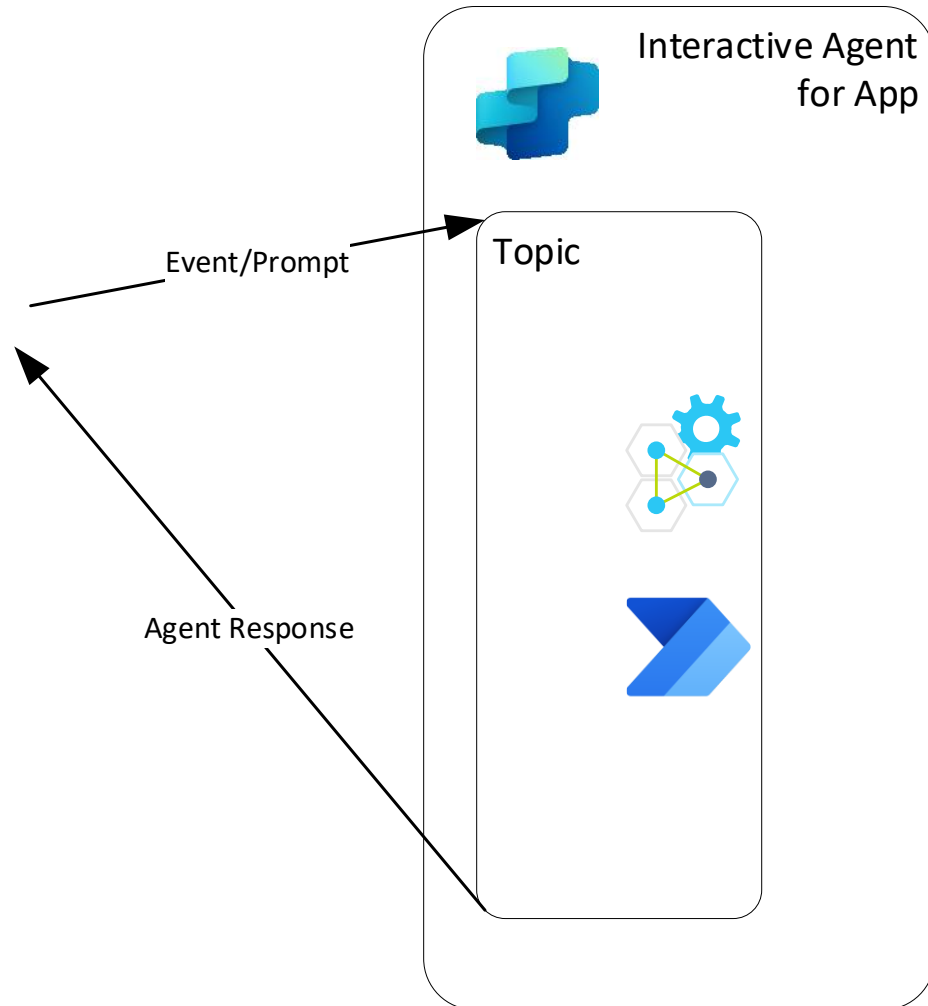
- **Config**:- Optional but reusability should be considered in a design as this component will render generic “Cards”
- **Cache**:- Recommended as this not only allows control over costs as you can check the cache first but also allows for autonomous pre generation of insights if the UI needs to be snappy. Using an Elastic table also allows us to leverage the TTL to auto clean up old cache records
- **Prompt**:- Using the OOB Copilot API’s ensure that we can easily call the Interactive Agent in the MDA and don’t need to consider Authentication. We can pass data context from the App if we need it either in parameters with the event or in the prompt. In this example we will pass a JSON payload of the current table and record.
- **Agent Response**:- To be able to deterministically render a “Card” then we would design the Agent to give back a particular JSON payload

PCF Design



```
{
  "results": [
    {
      "header": "Text",
      "text": "Text",
      "action": {
        "type": "updaterecord or createrecord",
        "payload": {
          "fields": [
            { "key": "fieldname",
              "value": "fieldvalue" },
            { "key": "fieldname",
              "value": "fieldvalue" }
          ]
        }
      }
    }
  ]
}
```

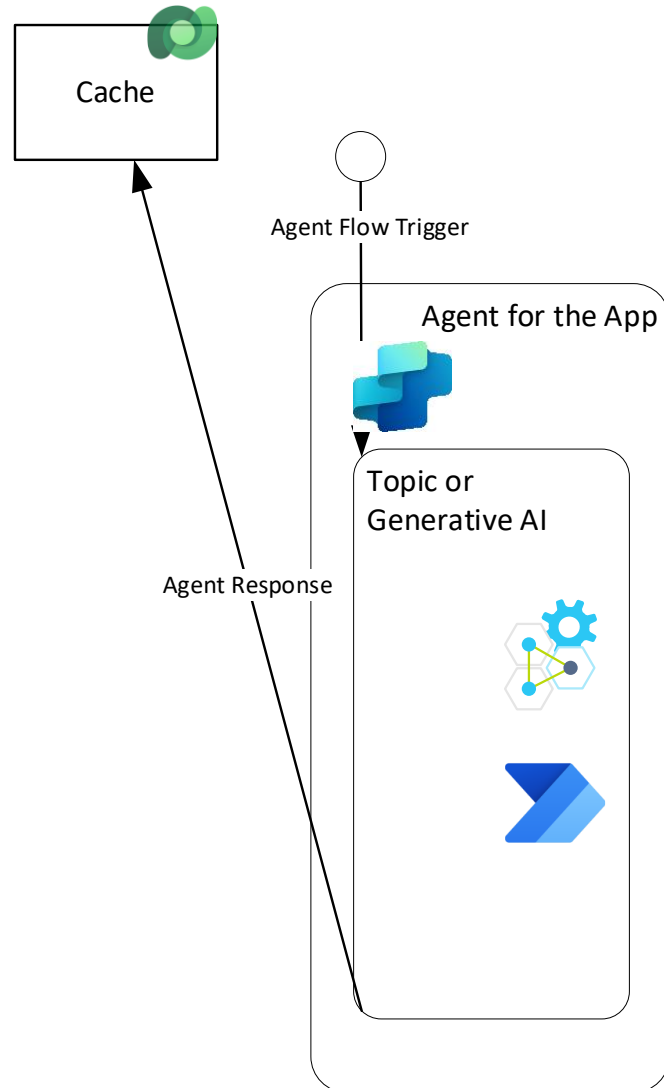
Intelligence - Topics



Directly invoking a Topic either using the Event API or a prompt works:-

- Useful when there is a deterministic order in which you want or need to contact external agents or systems
- Allows greater control of how the insights are generated
- Allows greater control over the response that is sent back to the Controller

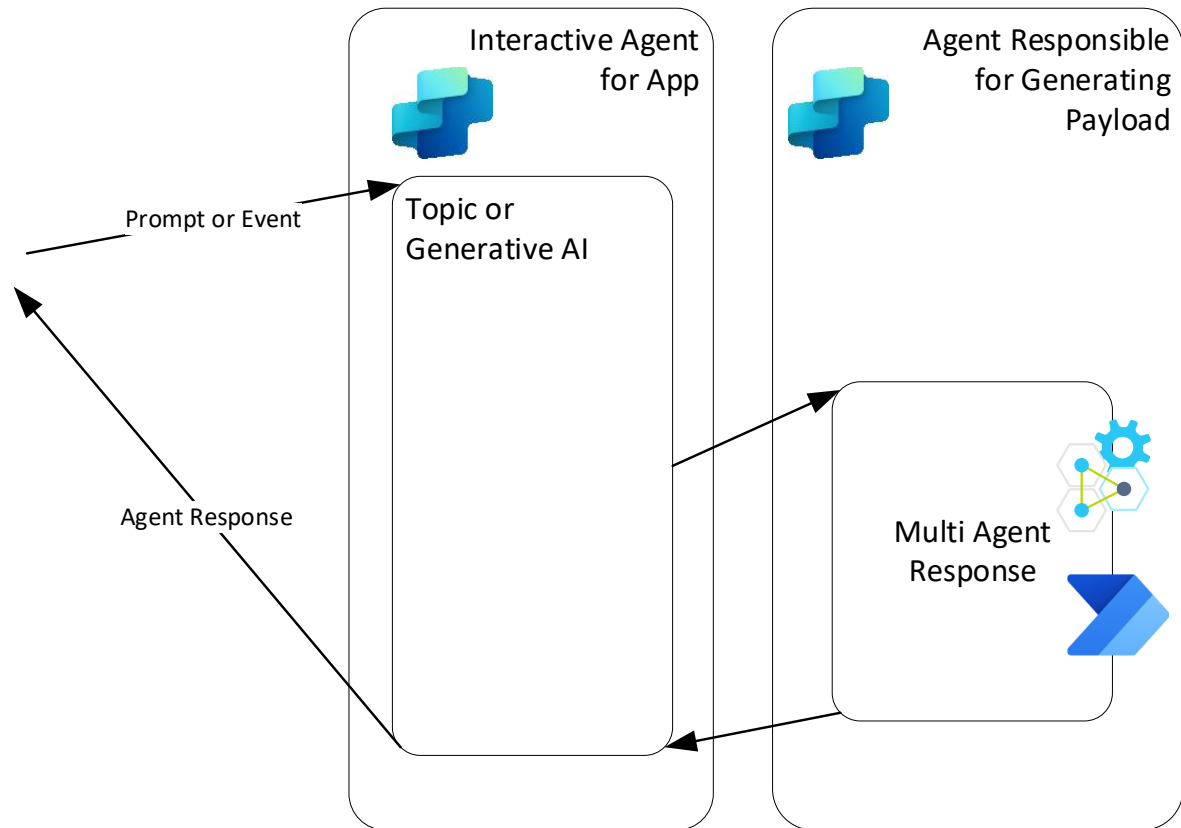
Intelligence - Autonomous



Adding the Cache to the design allows Autonomous triggers to be used to generate the insights:-

- Useful when the insights can be generated automatically based on triggers
- Useful when the insights can take a long time to generate and you don't want the user to have to wait
- Useful if the trigger doesn't come from the MDA but a 3rd Party
- Flexible for any integration scenarios also if the insights are coming from a completely separate 3rd Party Agent

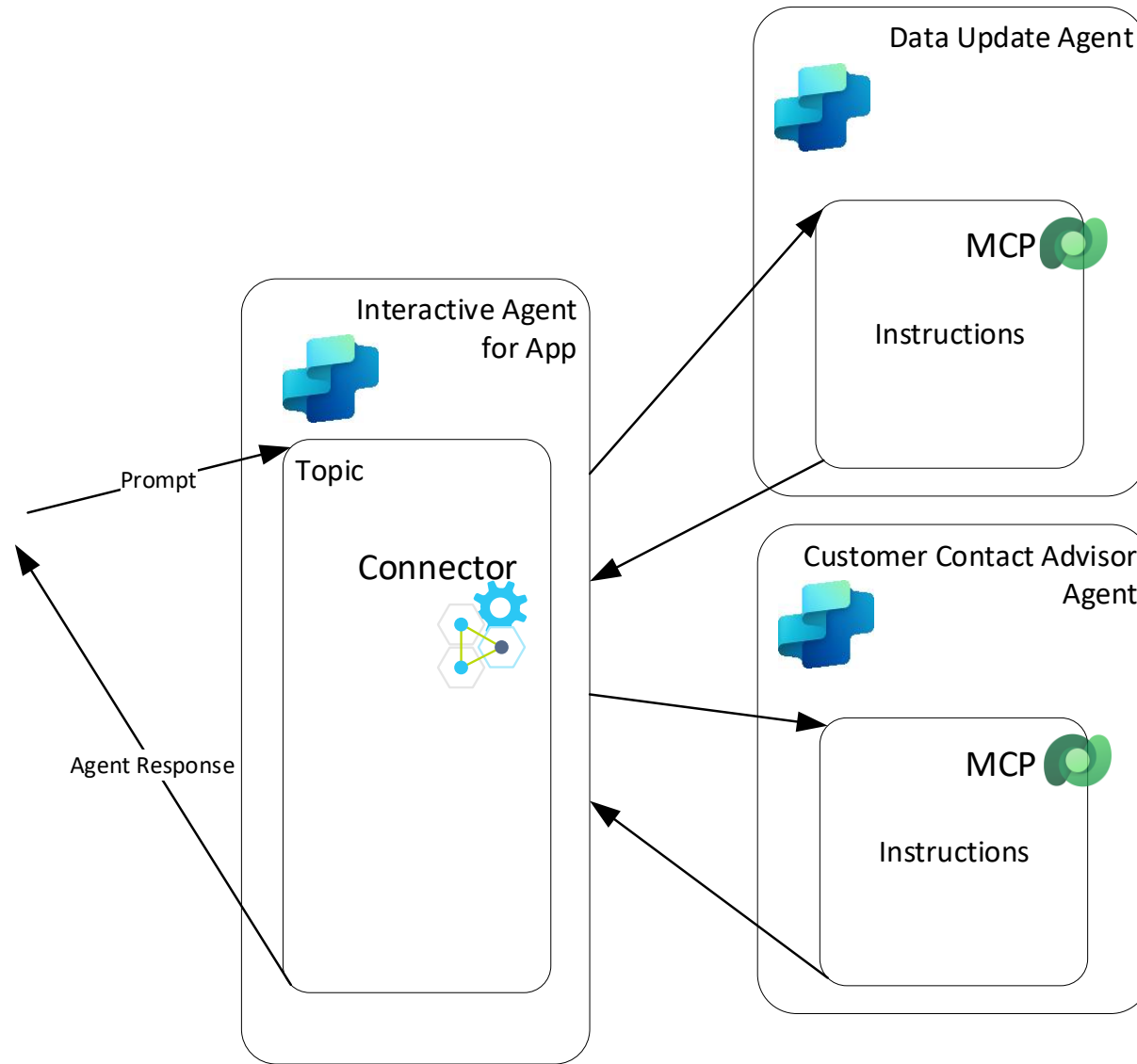
Intelligence – Agent 2 Agent



Shifting to a multi agent design allows the complexity of the Interactive Agent to be reduced so it can concentrate on Orchestration:-

- Separate Agents can be used that focus on particular insights
- Topics can be used to orchestrate the Agent response in a controlled manner or if that is not needed the Interactive Agent could generate the plan to contact the other Agents from a sufficiently good prompt/instructions
- Multiple Agents can be used to work together to produce the payload so that it is decoupled from the Interactive Agent

Example



Agents

Agent's model
GPT-4o (default) ...

Analytics

Check your agent's key performance info from the last 7 days.

Conversation sessions	Engagement	Satisfaction score
3 ↓ 91%	100% 0%	--

Open Analytics

Instructions

Cancel Save

↶ ↷

- Retrieve the top 5 emails from `Dataverse` ordered by created on date descending where the `recordid` matches the `regardingobjectid` only the subject and description are needed. Always use `TOP` dont not use `LIMIT` when trying to limit records in your query.
- Retrieve the target record (use the table name provided e.g. `opportunity`) where the primary key equals the `recordid` you should get all of the records fields to compare
- Only retrieve the emails and record once
- Compare the content of the emails with the current field values in the record do not use any other data or knowledge
- Only make recommendations if there are clear discrepancies between the email content and the value of the field that best matches the content, if the field you want to update is of type choice then you should provide the correct numeric value of the option to be set.
- Ignore fields that are not mentioned in the emails and do not provide any other recommendations on the fields in the table
- For each recommendation, provide the field schema name and the value to update in the response in this format only do not provided any other text but the `json` object. If there are no suggestions just output an empty `json` array `[]`:-

```
[[{"header": "Update Opportunity Data", //this would be a static value"text": "reasons for updating the data based on email content and data fields", //you should set this based on your reasoning"action": {"type": "updaterecord", //this would be a static value"payload": {"fields": [{"key": "fieldname",
```

2061/8000

Orchestration

Use generative AI to determine how best to respond to users and events. [Learn more](#)

Enabled

Agent's model

GPT-4o (default) ...

Analytics

Check your agent's key performance info from the last 7 days.

Conversation sessions	Engagement	Satisfaction score
3 ↓ 86%	100% 0%	--

Open Analytics

Instructions

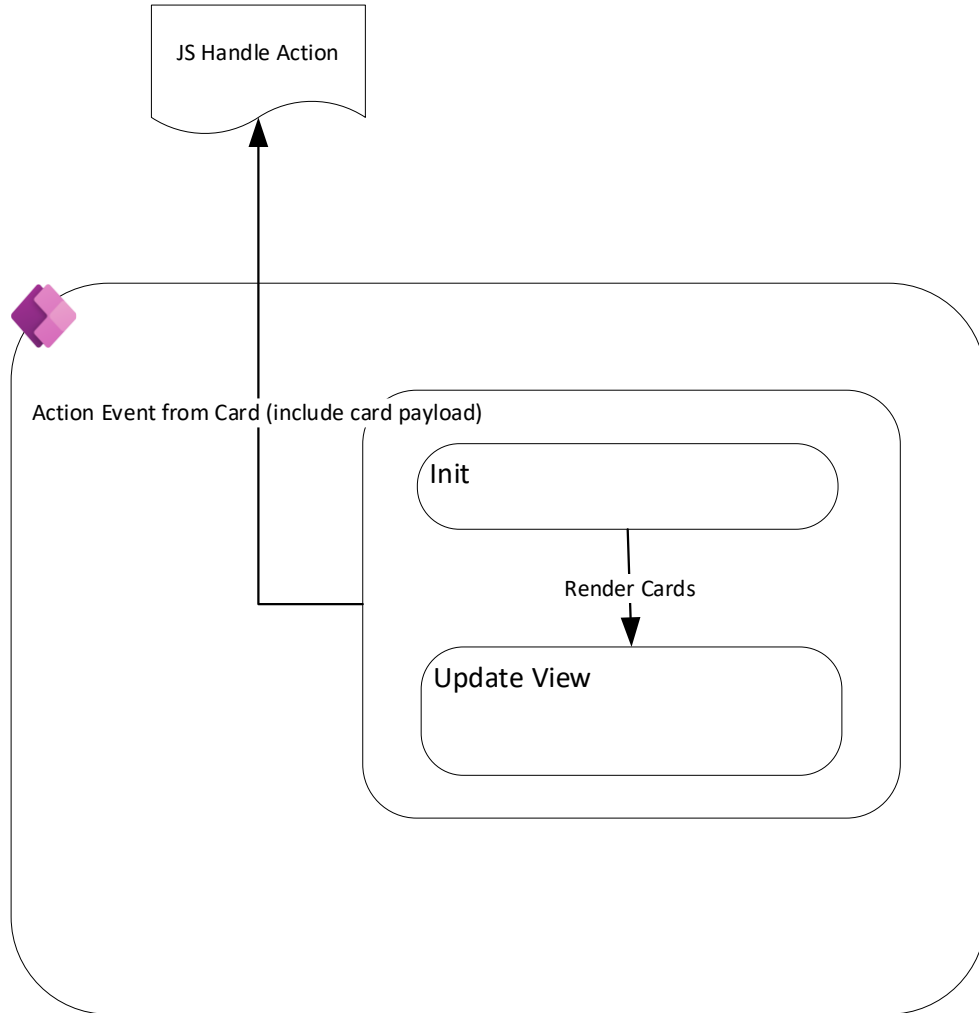
Edit

- Retrieve the top 5 emails from `Dataverse` ordered by created on date descending where the `recordid` matches the `regardingobjectid` only the subject and description are needed. Always use `TOP` dont not use `LIMIT` when trying to limit records in your query.
- Only retrieve the emails once
- Based on the all of the email content (subject and description) recommend if an appointment, email, both or neither should be required for this customer.
- Evaluate factors such as urgency, complexity, customer preferences, and previous interactions.
- Only make recommendations if there is a clear need to book and appointment and or send and email to the customer
- Do not make any changes to the data directly; only provide recommendations.
- Only use `MCP Dataverse Tool` to retrieve data do not use any knowledge sources
- For each recommendation, provide the reason for the recommendation and also suggest only the subject and description for appointment or email to be send. If there are no suggestions just output an empty `json` object `[]` :-

example appointment recommendation:-

```
[[{"header": "Schedule Appointment with Customer", "text": "The customer has indicated that they need to understand the deal structure better to be able to process with the offer, a meeting should be held with the customer", "action": {"type": "createrecord", "payload": {
```

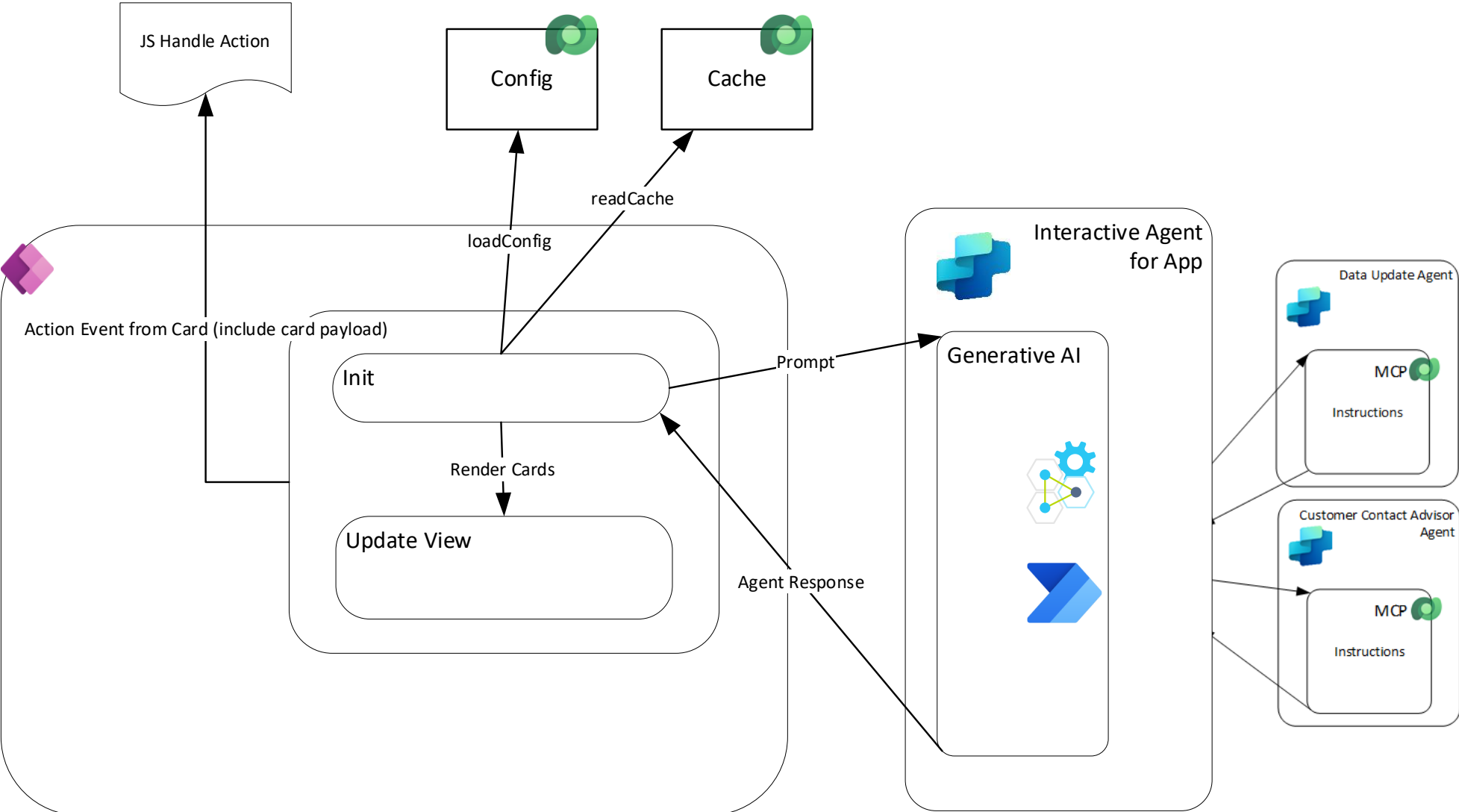
Action



Standard Javascript event handler:-

- Using PCF events and js handlers decouples the "Action" from the PCF increasing its reusability
- PCF can be used on multiple tables with different data models so you will need different fields to carry out the actions e.g. get the "customer" to add to an email or appointment changes depending on the target table
- Depending on the "Action" you cant do everything from inside the PCF context. So moving the handler to JS gives maximum flexibility to the maker.

Design Overview





Demo

- Home
- Recent
- Pinned
- New Group
- Embedded Intelligen...
- Embedded Intelligen...
- Opportunities
- ConversationTranscri...
- Accounts

My Open Opportunities

Show As Show Chart + New Delete Refresh Visualize this view Email a Link Visualize

Ask about data in this table. Owner: Me Status: Open

Topic	Potential Cu...	Est. close date	Est. revenue	Contact	Account	Probability	Rating	Email (Potential Custo...
Needs to restock their supply of Product SKU AX305; wil...	Blue Yonder Ai...	6/18/2025	\$25,000.00	Sidney Higa (s...	Blue Yonder Ai...		Hot	
Very likely will order 18 Product SKU JJ202 this year (sam...	Alpine Ski Hou...	7/21/2025	\$30,000.00	Paul Cannon (s...	Alpine Ski Hou...		Hot	
They sell many of the same items that we do - need to f...	Contoso Phar...	8/24/2025	\$26,000.00	Robert Lyon (s...	Contoso Phar...	95	Hot	
Will be ordering about 110 items of all types (sample)	Coho Winery (...)	9/22/2025	\$25,000.00	Jim Glynn (sa...	Coho Winery (...)	80	Hot	
Laptops for Youth Group	Fabrikam, Inc. ...	12/31/2025	\$12,000.00	Maria Campbe...	Fabrikam, Inc. ...	90	Hot	

Rows: 5



Important Considerations

Recommendations

1) **Controller** – Use the Copilot API's to communication with the Interactive Copilot in MDA it removes complexity of handling authentication and Agent SDK.

[Xrm.Copilot \(Client API reference\) in model-driven apps \(preview\) - Power Apps | Microsoft Learn](#)

[Bring intelligence into your components using Agent APIs \(preview\) - Power Apps | Microsoft Learn](#)

! Note for Copilot Service Workspace these API's are not currently supported so you will have to go the Agent SDK route for now !

2) **Intelligence** – Extend the Interactive Agent and use it to call purpose-built Agents (A2A design) this simplifies each of the Agents

3) **Action** – Use a separate JavaScript event handler gives maximum flexibility and decouples the Action from the Controller

4) **Add a Cache** – Adds more complexity to the design but it provides greater flexibility in the overall solution.

5) **JSON Responses** – Ensure that you provide clear instructions and examples to Agents on the format you are expecting or use a Topic to control if you are finding difficulties

6) **Topics** – When using a Topic be sure to end the conversation as sometimes with Gen AI enable it might add on another response before its sent back to the controller

Recommendations

5) **JSON Responses** – Ensure that you provide clear instructions and examples to Agents on the format you are expecting or use a Topic to control if you are finding difficult

6) **Topics** – When using a Topic be sure to end the conversation as sometimes with Gen AI enable it might add on another response before its sent back to the controller

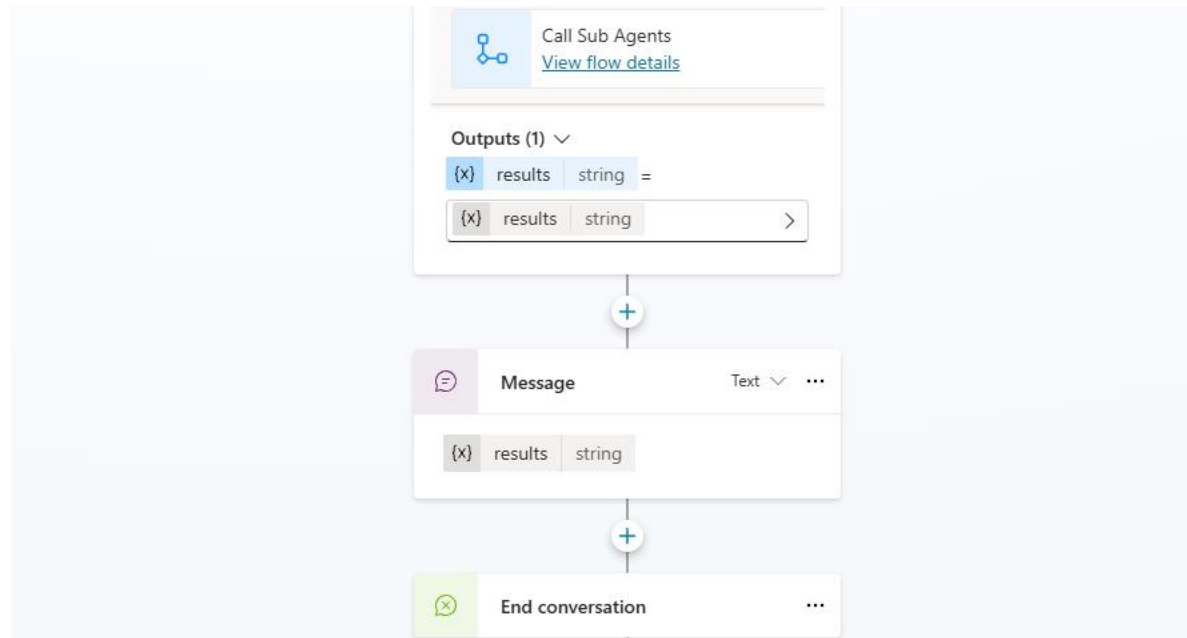
The screenshot shows the 'Data Update Agent' interface. At the top, there's a navigation bar with 'Overview', 'Knowledge', 'Tools', 'Agents', 'Topics', 'Activity', 'Analytics', and 'Channels'. Below the navigation bar, there's a status bar showing '6' with an upward arrow and '500%', '100%' with a progress bar, and '0%' with a downward arrow. The main content area is titled 'Instructions' and contains the following text:

- ignore fields that are not mentioned in the emails and do not provide any other recommendations on the fields in the table
- For each recommendation, provide the field schema name and the value to update in the response in this format only do not provided any other text but the json object. If there are no suggestions just output an empty json array []:-

```
[[{"header": "Update Opportunity Data", //this would be a static value"header": "Update Opportunity Data", //this would be a static value"text": "reasons for updating the data based on email content and data fields", //you should set this based on your reasoning"action": {"type": "updaterecord", //this would be a static value"payload": {"fields": [{"key": "fieldname", "value": "fieldvalue"}] //this should be set based on the fields you think should be updated there should always be "key" and "value" } } ]]
```

Below the instructions, there are three bullet points:

- Do not make any changes to the data directly; only provide recommendations.
- Ensure responses are clear, actionable, and concise.
- Only use MCP Dataverse Tool to retrieve data do not use any knowledge sources



Recommendations

7) **MCP** – Sometimes the Agent might get errors when querying the MCP so being more explicit in the instructions helps avoid error situations occurring at runtime

- Retrieve the top 5 emails from Dataverse ordered by created on date descending where the recordid matches the regardingobjectid only the subject and description are needed. Always use TOP dont not use LIMIT when trying to limit records in your query.

8) **API Response** – You actually get back an array of response objects including events which the Agent acted upon as well as the text response you are expecting. Ensure you dig for the right object

```
private findCorrectResponseMessage(response: any) {  
  // response is an array of responses  
  for (const res of response) {  
    if (res.type && res.type === "message") {  
      if (res.text && res.text.length > 0) {  
        if (this.isValidCardDataArray(res.text)) {  
          return res.text;  
        }  
      }  
    }  
  }  
  return "{}";  
}
```

9) Race conditions with the Controller making requests to the Interactive Copilot before it is fully loaded can sometimes cause errors or the Copilot Control to not load properly so ensure PCF waits for the form to be loaded before making the request

10) When using custom connectors in Agents the user has to accept and create connections the first time they use the Agent so ensure you cater for this during your roll out

Thank you