

Thinking outside of the Power BI box

I've had a recent experience with a client. We've stood their first Azure modern data platform. They've data streaming into the cloud, stored in a lake, transformed into various business data layer and presentation layer objects. They have strong data assets which the business analysts are eager to get their hands on, but there is an issue with Power BI reporting layer.

Coming from an on-premise scenario, where they had SSRS connecting directly to source databases, the client had a lot of customer reports that they wanted to migrate. When I say a lot, I mean over 800 active customer reports. The previous SSRS reporting framework was metadata driven and produced reports as excel attachments that were emailed to customers. There was a master report template that contained approximately 160 fields. And the customers were given the opportunity to select which of these columns they wanted to include, in which sequence they'd like them to be displayed as well as the opportunity to amend the table header names. The reporting framework took this metadata at runtime and dynamically created SQL queries that connected directly to the source database. The customers could schedule their own report delivery time, but as you can imagine, some scheduled times were quite popular and resulted in many concurrent queries taking up valuable resources from the source database.

Hands down, a modern data platform solution would resolve the resource contention issues they were facing with their current customer reporting framework. But the tricky part was how to deploy these 800+ reports into the Power BI service.

Power BI purists may argue that the proper way to do reporting in the Power BI service is with creating interactive reports. Ok, so with PowerBI premium we could create individual customer workspaces, we could drop in their requested version of the interactive report, and we could package it up into their own Apps. Then either the customer or an internal staff member could subscribe the customer to the report. But there are several reasons why this was not a good option for this client:

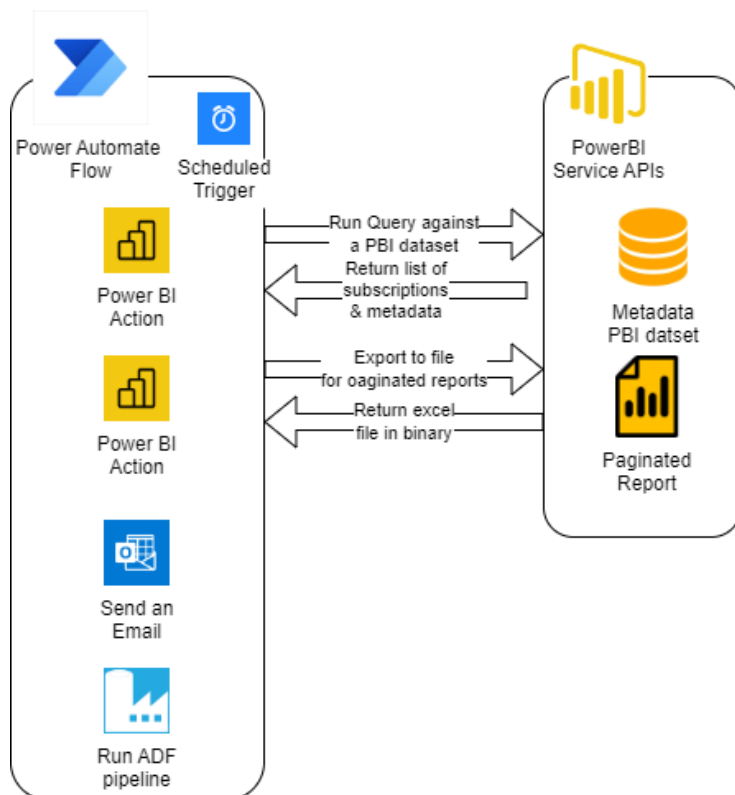
- Customer buy in: the customers are used to receiving the report as an email attachment, some would be ingesting this file into their own data warehouse. They may not have the staff with Power BI experience.
- Outputting excel documents from Power BI interactive reports is not perfect. Whilst it is possible, it requires a work around and it doesn't work seamlessly with the Power BI service subscription service.
- Power BI subscription maintenance: currently the Power BI subscription service cannot be data driven. There are no subscription APIs that can be called to update subscriptions, it all

needs to be maintained from the PBI service User Interface. So there is a requirement that an internal staff member with a Power BI admin privileges needs to maintain the customer subscriptions

- Initial report building efforts are substantial and will require dedicated ongoing maintenance. Although the report builders could start with a template, they would still be required to drag and drop the columns into the correct order on the tablix visual, rename the columns if required and apply the required filter criteria.
- Security: It would not be advisable to apply the customer code filtering inside the Power BI report that a customer has access to. Users should not be able to view another customer's data. So row level security would need to be applied at the data source level. Which would require additional work to setup and maintain.

The Solution

The reason for the title of this article, is because not all the best solutions fit nicely inside Power BI interactive reports and the Power BI service. For this number of reports, and at this point of time, I would not recommend creating them all as individual reports. I also would not recommend maintaining the subscriptions inside the power BI service. Once the PowerBI subscription service becomes data driven, I'll probably rethink my stance here.



So my solution elements in a nutshell:

- Power Automate flow
 - Scheduled Trigger
 - Run Query against an PowerBI dataset

- DAX query against a PowerBI dataset to return a list of active subscriptions and some metadata to execute the report
- Export to file for paginated report
 - Executes a paginated report with provided parameters and returns the binary contents of the excel file
 - The paginated report includes:
 - A Tablix with generic column placeholders (i.e. col1, col2, ..., coln)
 - a metadata dataset that contains the customer specific DAX query
 - a metadata header dataset that contains customer specific header sequences
 - a single reportID parameter
 - a report dataset whose query is driven by the selected reportIDs dax query
- Send an Email
 - Using metadata response from PowerBI dataset, creates an email distribution list, body and subject
 - Using the binary contents of previous activity, adds the excel file as an attachment
- Run ADF Pipeline
 - ADF pipeline to update the metadata database and sunset processed subscriptions

Without going into too much detail, the purpose of this article was to demonstrate that a traditional Power BI interactive report solution may not always be the best solution. That you can leverage the Power BI service APIs with a Power Automate flow and produce a flexible solution that is still tightly coupled with Power BI. Best of all, your solution can always be data driven. Chances are, if you already have metadata that's running your existing/outdated report framework, that this can be repurposed into a modern data platform. Also, that initial development costs and ongoing maintenance costs can be minimalised whilst at the same time enjoying the benefits of the Azure modern data platform.

If your business needs help with building a Power BI reporting solution, contact a Fujitsu Data & AI specialist now.

Caveat: as a Data Engineer for modern data platforms, I see constantly evolving technologies that provide new opportunities to provide better solutions. Later this week, the Microsoft Build event should see the announcement of much anticipated integration of the next generation Copilot with the Power Automate service. I suspect that this article will require an update, or at least a follow on article to discuss how this new integration can benefit our clients.

Contact

Fujitsu Data & AI
+61 3 9924 3000

© Fujitsu 2022. All rights reserved. Fujitsu and Fujitsu logo are trademarks of Fujitsu Limited registered in many jurisdictions worldwide. Other product, service and company names mentioned herein may be trademarks of Fujitsu or other companies. This document is current as of the initial date of publication and subject to be changed by Fujitsu without notice. This material is provided for information purposes only and Fujitsu assumes no liability related to its use.