

- 
- +
- 

# **LOW-CODE APP & EXTRACT-TRANSFORM-LOAD (ETL) AUTOMATION ON MICROSOFT POWER PLATFORM**

Transforming Solar PV Engineering with a  
Centralized Data Hub that Saved 50% in Data  
Acquisition Time

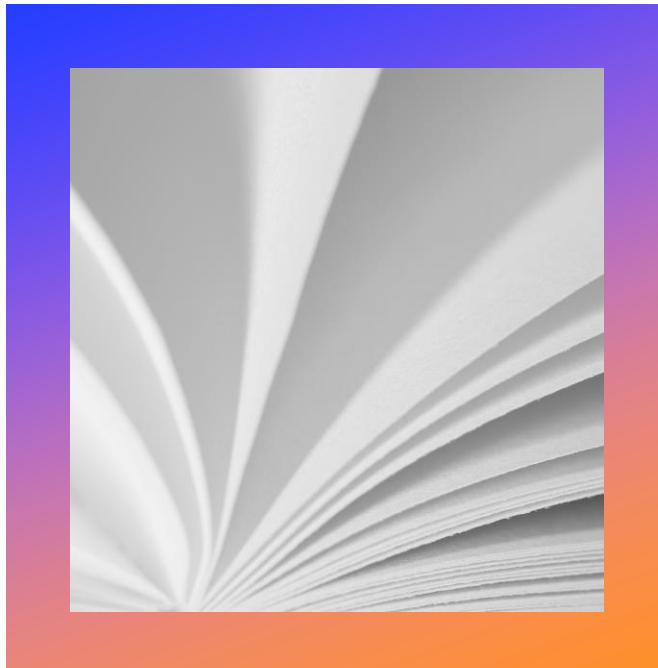
Ayokunle Samuel Adenigba  
Analytics Engineer | Power Platform Expert  
[ayo.adenigba@leitway.com](mailto:ayo.adenigba@leitway.com)  
+49 178 1854390



# PROJECT INTRODUCTION



# EXECUTIVE SUMMARY



## Low-Code ETL Pipeline

Developed a low-code app and ETL pipeline using Power Platform to automate solar meteorological data collection from multiple data providers and process them effectively.

## Technology Integration

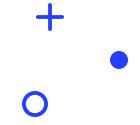
Leveraged Power Apps, Power Automate, SQL Server, and Power BI to streamline data acquisition, transformation, and analytics.

## Efficiency Improvements

Achieved 50% reduction in data acquisition time and eliminated redundant purchases through centralized data management.

## Engineering Empowerment

Provided engineers with reliable, on-demand datasets and actionable insights to enhance simulation accuracy and project turnaround.



# EXISTING SITUATION



Solar PV teams and engineers situated at different locations must collaborate on projects development and system simulations. A fundamental requirement is the historical meteorological data for the proposed site. The existing process of collection, analysis and storage of this data followed a manual and fragmented approach, therefore providing little to no data governance, traceability and visibility, which are necessary for simulation accuracy, data integrity, and to avoid cost of redundant data request.

Data description for a project: Up to 30 years of timeseries data on a 15-minute, hourly, daily or yearly interval from 3-4 different sources.

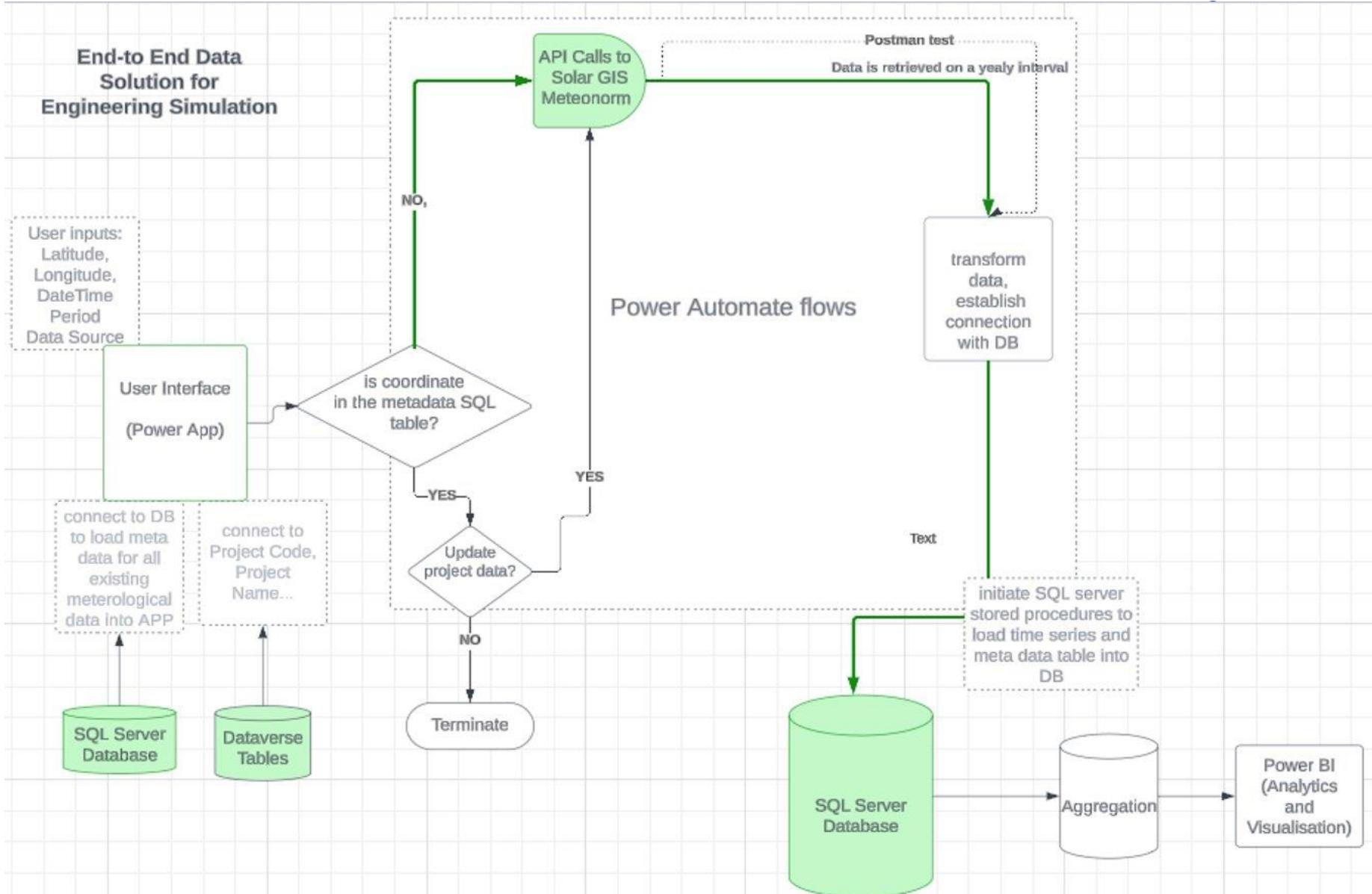
Development of PV system can take over a year depending on project size and complexity, so data update and traceability is paramount.

# ARCHITECTURE AND WORKFLOW



# ARCHITECTURE AT A GLANCE —

## DATAFLOW DIAGRAM



# SOLUTION ARCHITECTURE – LOW-CODE APP & ETL PIPELINE

## User Data Intake

Power Apps provides an easy-to-use interface for submitting data with validation and deduplication.

## Workflow Orchestration

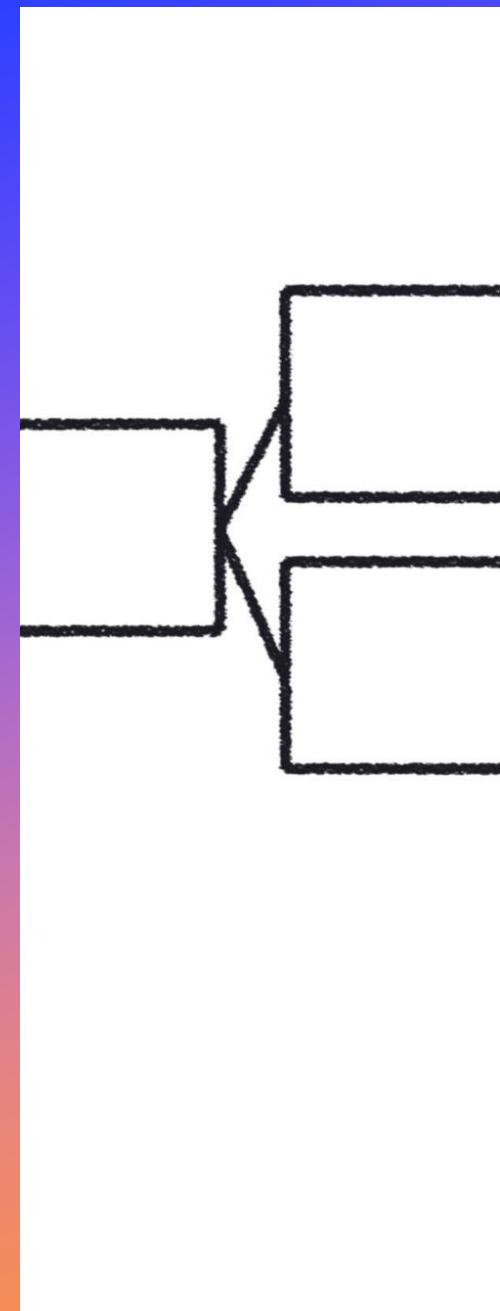
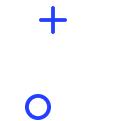
Power Automate manages data parsing, transformation, API calls, and asynchronous workflow polling.

## Data Processing and Storage

SQL Server executes stored procedures to transform and store data into consistent, auditable formats.

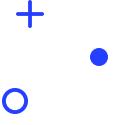
## User Notifications and Reporting

Notifications via Outlook and Teams keep users updated, while Power BI delivers interactive analytics dashboards.



# POWER APP INTERFACE





# BUSINESS LOGIC AND INTAKE

FEATURE	DESCRIPTION
<b>Project Selection</b>	Users select existing projects from Dataverse to associate with data requests.
<b>Location Input</b>	Supports single and multiple coordinate entries for data acquisition.
<b>Resolution Validation</b>	Ensures users select appropriate data resolution before submission.
<b>Duplicate Check</b>	Alerts users if similar data already exists in the database.
<b>Trigger Flow</b>	Initiates Power Automate flow upon successful validation.

**1. Select the project.**

GBR0062 X

GBR00756 X

**2a. Latitude**  
52.4544

**2b. Longitude**  
34.56

**3. Select data source(s)**

Solargis

Select data Resolution

Meteonorm

TMY\_Monthly

**Submit Data Request**

(Climate data in ibvogt SQL Database) Hide table

**Single Coordinates Request**

Insert multiple coordinates

site1:56.763,45.8976  
site2:34.6752,35.672

Site_ID	Latitude	Longitude	Meteonorm Data Resolution	Solargis Resolution	Start date	End date
site1	56.763	45.8976	TMY_Hourly	MIN_15	01.01.1994 <input type="button" value="▼"/>	01.12.2024 <input type="button" value="▼"/>
site2	34.6752	35.672	TMY_Hourly	MIN_15	01.01.1994 <input type="button" value="▼"/>	01.12.2024 <input type="button" value="▼"/>

Meteonorm  
TMY\_Hourly

Solar\_GIS  
MIN\_15

Start Date  
1/1/1994

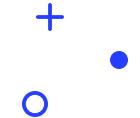
End Date  
12/1/2024

**Submit multiple Request**

# POWER APP INTERFACE

Activity: Selection of a project from dataverse. (Image - left side)

change images here to single and multiple screen request



1. Select the project.

GBR0062 X

GBR0062 Worthing

GBR0028 BESS Feeder Road

GBR0029 Brecks Farm

GBR0054 Barnsley

GBR0055 Warley

GBR0056 Walpole 1

Toggle off to proceed with request

Submit Data Request

(Climate data in ~~Project~~ SQL Database) Hide table

Site GBR0107	Meteonorm TMY_Monthly	Lat: 53.5328 Lng: -0.7741	Start Date: 01.01.2005 End Date: 01.12.2005
Site GBR0110	Meteonorm TMY_Monthly	Lat: 51.6816 Lng: -1.9759	Start Date: 01.01.2005 End Date: 01.12.2005
Site GBR0116	Meteonorm TMY_Monthly	Lat: 53.9116 Lng: -0.9642	Start Date: 01.01.2005 End Date: 01.12.2005
Site GBR2257	Meteonorm TMY_Monthly	Lat: 50.1902 Lng: -5.3319	Start Date: 01.01.2005 End Date: 01.12.2005
Site GBR2257	Meteonorm TMY_Monthly	Lat: 50.1902 Lng: -0.3319	Start Date: 01.01.2005 End Date: 01.12.2005
Site GBR2259	Meteonorm TMY_Monthly	Lat: 51.661 Lng: -2.032	Start Date: 01.01.2005 End Date: 01.12.2005
Site GBR2261	Meteonorm TMY_Monthly	Lat: 53.561 Lng: -0.177	Start Date: 01.01.2005 End Date: 01.12.2005
Site GBR2263	Meteonorm TMY_Monthly	Lat: 51.5869 Lng: 0.58	Start Date: 01.01.2005 End Date: 01.12.2005
Site	Meteonorm	Lat: 54.8076	Start Date: 01.01.2005

Power FX code within the app

```
If(CountRows(SolargisTMYmultiRequest)> 0,  
ForAll(SolargisTMYmultiRequest,  
Patch(ResponseTable, Defaults(ResponseTable),  
{ResponseDetails: SolargisTMYnewDataRequest.Run(JSON(ThisRecord))});  
  
/*  
If(CountRows(SolargisTMYmultiRequest)> 0,  
ClearCollect(ResponseTable,  
ForAll(SolargisTMYmultiRequest,  
{ResponseDetails: SolargisTMYnewDataRequest.Run(JSON(ThisRecord))});*/  
  
If(CountRows(MeteonormMultiRequest)>0,  
ForAll(MeteonormMultiRequest,  
Patch(ResponseTable, Defaults(ResponseTable),  
{ResponseDetails: MeteonormNewRequest.Run(JSON(ThisRecord), varU  
Set(showLoadingImage, false);  
Navigate(Request_StatusScreen);  
d|
```

# POWER APP INTERFACE

## Intelligent Time & Cost

### Savings:

- Proactively notified users if data for a site already existed, preventing redundant API calls and saving cost.

(Image: arrow pointers)

- Trigger power Automat flow and inform user of a successfully initiated request.

1. Select the project.

Site Code

2. Enter the coordinates

2a. Latitude

2b. Longitude X

3. Select data source(s)

Solargis

Select data Resolution

Meteonorm

Select data Resolution

Meteo-data exists for similar coordinates. please check

Toggle off to proceed with request

Submit Data Request

(Climate data in ibvogt SQL Database) Hide table			
Site DEU0249	Solargis MIN_15	Lat: 52.5552 Lng: 12.7926	Start Date: 01.01.1994 End Date: 30.03.2023
Site DEU0249	Solargis TMY_Hourly	Lat: 52.5552 Lng: 12.7926	Start Date: End Date:
Site DEU0396	Solargis MIN_15	Lat: 53.5081 Lng: 8.4679	Start Date: 01.01.1994 End Date: 21.01.2022
Site DEU0396	Solargis TMY_Hourly	Lat: 53.5081 Lng: 8.4679	Start Date: End Date:
Site DOM0001	Solargis MIN_15	Lat: 18.3504 Lng: -70.4355	Start Date: 04.01.1998 End Date: 30.03.2023
Site DOM0001	Solargis TMY_Hourly	Lat: 18.3504 Lng: -70.4355	Start Date: End Date:
Site ESP3817	Solargis MIN_15	Lat: 40.9131 Lng: -4.2361	Start Date: 01.01.1994 End Date: 31.12.2021
Site FRA9051	Solargis	Lat: 48.2873 Lng: 5.0807	Start Date: End Date:

Similar coordinates in the database

Notify user: data exist for close location (coordinate)



## Update Data

IBV\_ClimateDB APP

[Search Site in database](#)

AUS2506      Source: Solargis      Lat: 48.6123  
Resolution: HOURLY      Lng: 20.8263 >

AUS2506      Source: Solargis      Lat: 48.6123  
Resolution: MIN\_15      Lng: 20.8263 >

DEU2330      Source: Solargis      Lat: 48.613  
Resolution: DAILY      Lng: 20.826 >

DEU2330      Source: Meteonorm      Lat: 48.613  
Resolution: Monthly\_TMY      Lng: 20.826 >

ESP3823      Source: Meteonorm      Lat: 45.834  
Resolution: Hourly\_TMY      Lng: 24.567 >

DEU2330      Source: Meteonorm      Lat: 48.613  
Resolution: Monthly\_TMY      Lng: 20.826 >

ESP3823      Source: Meteonorm      Lat: 45.834  
Resolution: Hourly\_TMY      Lng: 24.567 >

Site/Project

ESP3823

Latitude

45.834

Data Provider

Meteonorm

Longitude

24.567

Resolution

Hourly\_TMY

Start date

January 1, 2005 1:00 AM

End date

January 1, 2006 12:00 AM

 Update date range Add a new resolution

Monthly TMY

Start date

January 1, 2005 1:00 AM

End date

January 1, 2006 12:00 AM

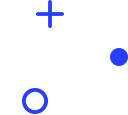
Maintain date range

 Yes     No

Update weather data for site

# POWER AUTOMATE ORCHESTRATION

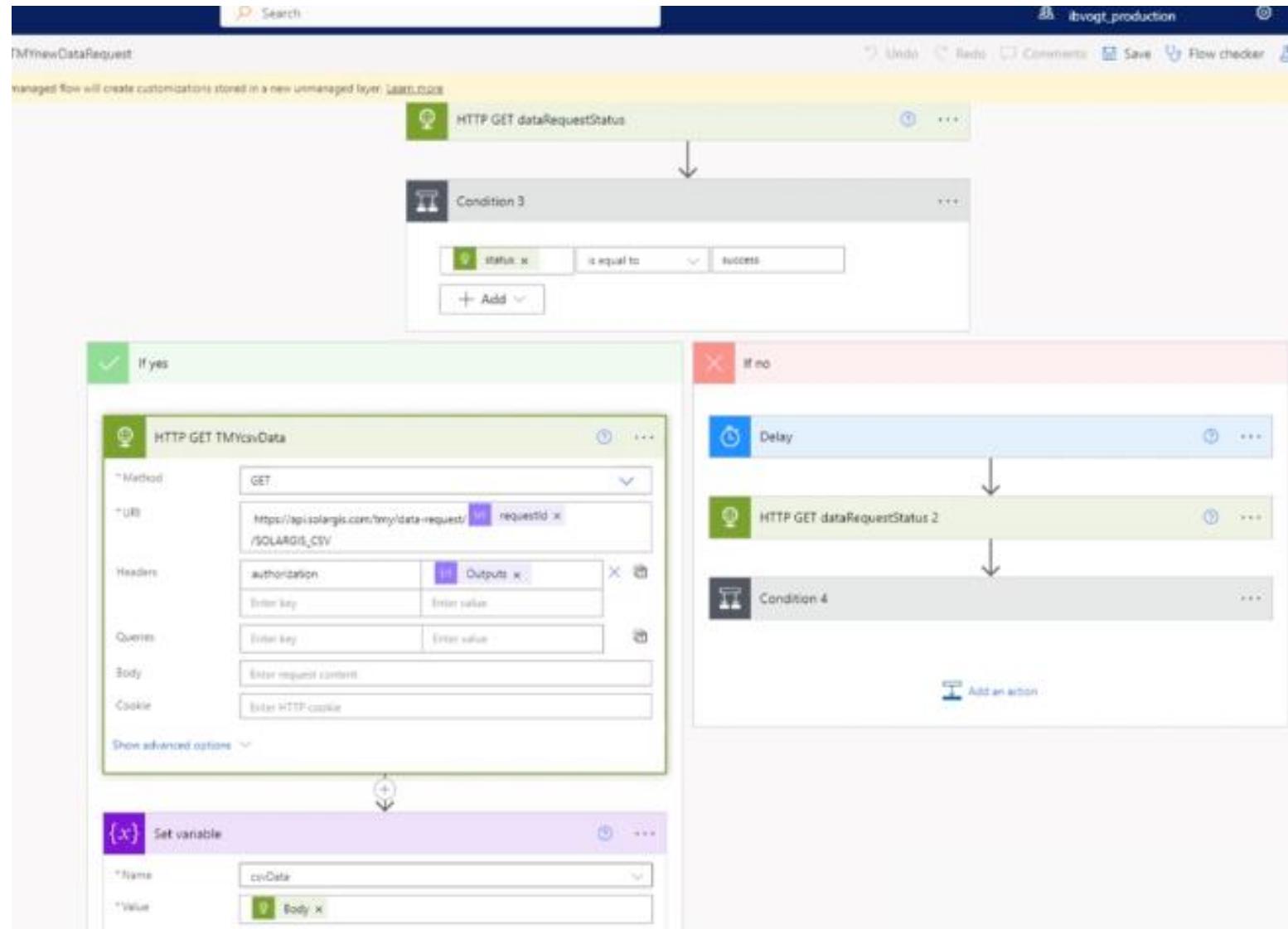
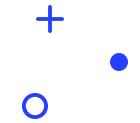




# DATA EXTRACTION AND TRANSFORMATION

ACTION	PURPOSE
HTTP POST	Submit data request to external APIs.
Polling	Check status of asynchronous jobs until completion.
Condition Checks	Validate response success before proceeding.
Error Handling	Log and notify users of failed requests.
HTTP GET	Collect actual data (Extraction)
Data Transformation	Align API response with SQL schema. (JSON, XML processing)

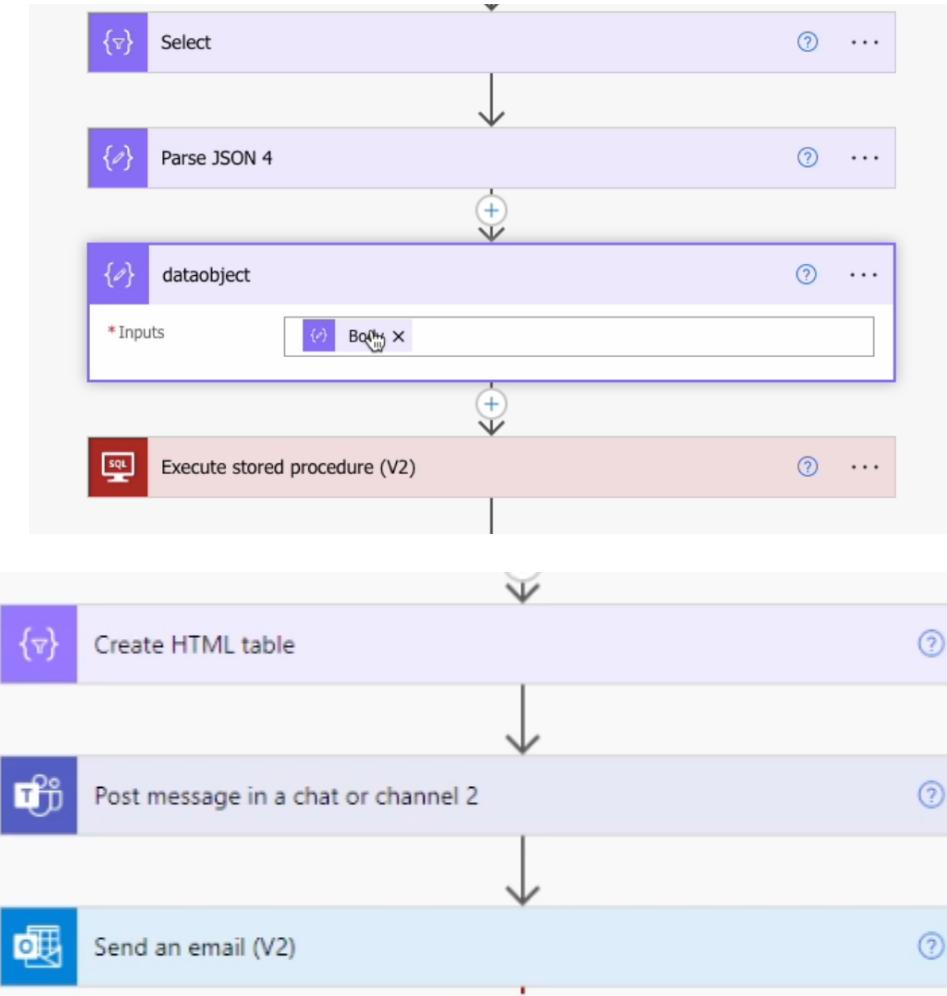
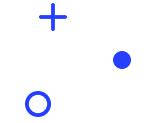
# POWER AUTOMATE FLOW SNIPPET



# DATA LOADING AND NOTIFICATIONS



# ETL - DATA LOADING

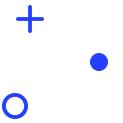


```
-- Description: <Description,>
-- =====
ALTER PROCEDURE [dbo].[importSolargisTMYjson]
    -- Add the parameters for the stored procedure here
    (@JSON_main nvarchar(MAX), @project_code NCHAR(20),
     @data_resolution NCHAR(15), @lat_alias NCHAR(12), @lng_alias NCHAR(12))

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    --DECLARE @OBrangefrom datetime, @OBrangeTo datetime
    --IF (@data_resolution = 'PT60M') --for TMY_monthly data
        INSERT INTO dbo.Solargis_TMYdata (ProjectORSite_ID, Data_resolution, Lat_alias,
                                         Lng_alias, [Unit_solarRadiation], [Date_time], [GHI], [DNI], [DIF], [Sun_elevation],
                                         [Sun_azimuth], [Temperature], [Wind_speed], [Wind_direction])
    SELECT
        @project_code AS ProjectORSite_ID, @data_resolution AS Data_resolution,
        @lat_alias AS Lat_alias, @lng_alias AS Lng_alias, 'Wh/m2' AS [Unit_solarRadiation],
        CONVERT(DATETIMEOFFSET, dt1.value, 127) AS [Date_time], d2.value AS [GHI], d3.value AS [DNI],
        d4.value AS [DIF], d5.value AS [Sun_elevation], d6.value AS [Sun_azimuth], d7.value AS [Temperature],
        d8.value AS [Wind_speed], d9.value AS [Wind_direction]
    FROM
        OPENJSON(@JSON_main, '$.DATETIME') AS dt1
        --CROSS APPLY OPENJSON(@JSON_main, '$.GHI') AS d1
        --WHERE dt1.[key]=d2.[key]
        CROSS JOIN OPENJSON(@JSON_main, '$.GHI') AS d2
        CROSS JOIN OPENJSON(@JSON_main, '$.DNI') AS d3
        CROSS JOIN OPENJSON(@JSON_main, '$.DIF') AS d4
        CROSS JOIN OPENJSON(@JSON_main, '$.SUN_ELEVATION') AS d5
        CROSS JOIN OPENJSON(@JSON_main, '$.SUN_AZIMUTH') AS d6
        CROSS JOIN OPENJSON(@JSON_main, '$.TEMP') AS d7
        CROSS JOIN OPENJSON(@JSON_main, '$.WS') AS d8
        CROSS JOIN OPENJSON(@JSON_main, '$.WD') AS d9
    WHERE dt1.[key] = d2.[key]
    AND dt1.[key] = d3.[key]
    AND dt1.[key] = d4.[key]
    AND dt1.[key] = d5.[key]
    AND dt1.[key] = d6.[key]
    AND dt1.[key] = d7.[key]
    AND dt1.[key] = d8.[key]
    AND dt1.[key] = d9.[key]
```

The screenshot shows the 'importSolargisTMYjson' stored procedure in SQL Server Management Studio. The procedure takes a JSON string (@JSON\_main) and several parameters (@project\_code, @data\_resolution, @lat\_alias, @lng\_alias) and inserts data into the 'dbo.Solargis\_TMYdata' table. The data is extracted from the JSON using OPENJSON and converted to various data types like DATETIMEOFFSET and numeric values.



# SQL STORED PROCEDURES AND USER ALERTS

PROCEDURE	FUNCTION
<b>Execute Stored Procedure</b>	Load data into SQL tables.
<b>Update Row</b>	Modify existing entries based on new data.
<b>Remove XML Header</b>	Clean API response for SQL compatibility.
<b>Notification</b>	Inform users of success or failure via email/Teams.
<b>Audit Logging</b>	Track request ID, rows ingested, and errors.

# ANALYTICS AND INSIGHTS



# POWER BI DOWNSTREAM ANALYSIS



## Data Insights

Power BI provides engineers with insights by connecting directly to SQL Server tables and views using Import modes.

## Data Quality and Analysis

DAX measures analyze data quality, availability, and engineering constraints to ensure reliable and accurate insights.

## Visualization of Key Metrics

Dashboards visualize metrics such as data coverage, resolution distribution, and request status for comprehensive monitoring.

## Integration with Simulation Tools

Power BI supports export interfaces and dataformat to simulation software(PVSyst), enabling seamless engineering workflow integration.

# POWER BI DASHBOARD

File Home Insert Modeling View Optimize Help

Sensitivity Publish ...

**Temperature\_2m WMA by data base [°C]**

Month	NASA	PVGIS_TMY	PVGIS-CMSAF	PVGIS-COSMO	PVGIS-ERA5	PVGIS-SARAH
January	2,42	3,34	3,29	3,07	3,19	3,19
February	2,97	5,40	3,10	2,82	2,90	2,90
March	5,56	7,84	5,78	5,69	5,62	5,62
April	9,31	8,24	9,93	9,95	9,81	9,81
May	13,18	14,08	13,40	13,32	13,41	13,41
June	16,69	15,61	16,07	16,14	16,18	16,18
July	18,28	17,97	18,18	18,52	18,51	18,51
August	17,88	18,58	17,72	17,35	17,43	17,43
September	14,88	13,67	14,72	14,85	15,09	15,09
October	10,87	13,64	10,70	11,32	11,19	11,19
November	6,50	6,69	7,08	7,37	7,22	7,22
December	3,27	4,02	3,71	3,82	3,86	3,86
Total	10,15	10,76	10,31	10,35	10,37	10,37

**Database** NASA PVGIS\_TMY PVGIS-CMSAF PVGIS-COSMO PVGIS-ERA5 PVGIS-SARAH PVGIS-SARA...

This slicer is synced with the slicer on the previous page

**Temperature\_2m WMA by data base [kWh/m<sup>2</sup>]**

Month	PVGIS_TMY	Total
January	3,34	3,34
February	5,40	5,40
March	7,84	7,84
April	8,24	8,24
May	14,08	14,08
June	15,61	15,61
July	17,97	17,97
August	18,58	18,58
September	13,67	13,67
October	13,64	13,64
November	6,69	6,69
December	4,02	4,02
Total	10,76	10,76

**Year count per database**

NASA	PVGIS-CMSAF	PVGIS-COSMO	PVGIS-ERA5	PVGIS-SARAH
24	10	11	12	12

**Check TMY scaling factor effect on Temperature**

**Corrected Monthly Data Export to PVsyst**

**Albedo Summary**

**Design\_Constraint: Min Temperature at high...**

**Temperature\_2m WMA by data base [°C]**

**Temperature\_2m WMA by data base [°C]**

Data Visualizations

Search

- Ambient\_Temperature...
- Coordinates\_retrieved
- Database
- Date\_Table
- Desired String Length
- Hour
- Hourly\_Data\_Repository
- LT\_Monthly\_Data\_Rep...
- Module Temperature ...
- Month\_Table
- pvPlanner\_Meteonor...
- Quarter\_Table
- Site Data
- TMY\_Data
- TMY\_month\_year
- VoC\_STC
- Year



# TECHNICAL DESIGN AND OUTCOMES



# ETL DESIGN CHOICES AND BUSINESS IMPACT

## Robust ETL Architecture

The design uses staging tables, schema-aligned transformations, and MERGE upserts for reliability and scalability.

## Idempotency and Polling

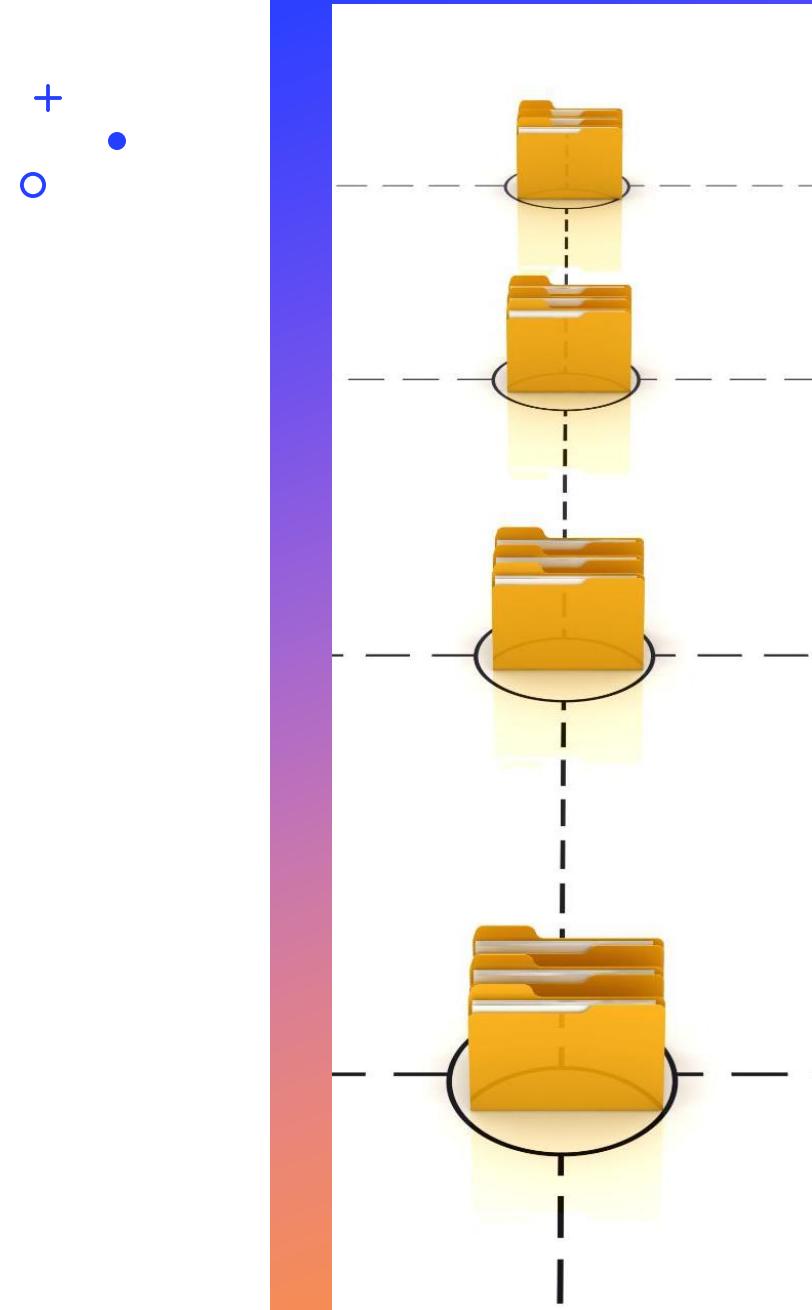
Composite keys ensure idempotency; polling with exponential backoff manages asynchronous API jobs efficiently.

## Logging and Security

Comprehensive logging tracks IDs, latency, and errors; secure connectivity is maintained via gateways and connection references.

## Business Impact and Future Plans

ETL improvements reduced acquisition time by 50%, eliminated redundant purchases, and enhanced simulation accuracy.



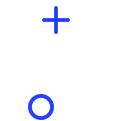
# TEAM WORK & RESPONSIBILITIES

## My responsibilities

- Development of the Power App user interface and Logic functions
- API integration with Solar GIS, Meteonorm.
- Development of Power Automate flows and logics.
- Data modelling , design and implementation in SQL Server, including stored procedures.
- Documentations

## Collaborative effort and other aspect of the development

- Solution architecture and design
- Development of Power BI dashboards for data visualization and analysis.
- Testing and deployment and documentation of the solution.



# RECOMMENDATION



**Abhishek Gawde** · 1st

Driving Business Growth through Digital Transformation | Microsoft Power Platform Expert | Automation & AI

|

I had the privilege of working with Samuel Adenigba, a seasoned professional whose expertise in solar PV project analysis and energy performance optimization is truly exceptional. Samuel's contributions have been instrumental in driving the success of solar projects, through critical milestones such as project sales, commissioning, and operational performance.

Samuel is a master of leveraging data and technology to solve complex challenges. His proficiency in tools like Microsoft Fabric, Power Platform, SQL, and Python enables him to build scalable data solutions and automate processes that significantly enhance business intelligence and asset performance. His work in orchestrating big data pipelines, developing automation tools, and optimizing PV plant designs demonstrates his innovative approach to driving efficiency and value.

What sets Samuel apart is not just his technical acumen but also his ability to bridge the gap between technical and non-technical teams. His ability to communicate requirements and align stakeholders ensures due diligence across all phases of solar projects.

If you're developing a solar project or tackling the challenges of underperforming assets, Samuel is the expert you need. His hands-on experience in energy simulations, financial analysis, and engineering design, combined with his deep industry knowledge, makes him an invaluable asset to any team or organization.

It's been an absolute pleasure collaborating with Samuel, and I highly recommend him for his unparalleled expertise and dedication to excellence in the solar energy sector.

