Gathering Data Projects requirement

In the last 3-5 years the variety of data and ways to onboard, process and visualise data using various tools and technologies have grown exponentially. Before 2015, if you worked on Microsoft platform data projects, you had limited tools like SQL server for storage and transformation, SSIS for integration, SSAS for modelling and SSRS for reporting. Now for each step during that data evolution journey there are multiple technologies and options are available at our disposal and hence it very important to choose them wisely. It's paramount to ask the right set of questions up front to gather low level requirements in order to select the right tool set and architecture. Polyglot persistence is a new way to design data platform architecture.

Functional requirements questions to ask:

What are we trying to create?

- On-prem data platform (SQL server, SSIS and SSRS , Power BI report Server)?
- Cloud based data platform?
- Hybrid data platform?

If Cloud, what kind of data system are we trying to develop?

- Warehouse system (MPP architecture e.g., Synapse, Databricks, Azure HD Insight etc.)?
- Operational System (Replicated data store e.g., SQL replication, Replicated Secondary Database for reporting system)?
- Lakehouse (New data management and processing palindrome)?

What kind of data project are we trying to execute?

- Migration project (Leverage Migration Assistant for online offline migration)?
- Modernisation project (Decommission legacy platform and redevelop ingestion and integration platform on cutting-edge technology)?

What kind of data processing do we need?

- Batch processing (data is loaded in batch mode incrementally from source system)?
- Streaming processing (where data is continuously loaded from IOT devices or events)?

Things to consider in ingestion and integration patterns.

- Deciding on the right Orchestration Tool
- Metadata Framework or configuration table (to easily onboard new objects during the ingestion process)
- Incremental load/Full load
- Data frequency and scheduling options
- Layers of data processing
    - Raw – staging – transform (Dim and facts)
    - Landing – Bronze – Silver – Gold
- Number of data sources
- Connectors to the data sources
- Load processing (Bulk, polybase, copy command, Delta load, etc.)

Data storage

- Data Format (csv, parquet, orc Delta format -Databricks ACID etc.)?
- Data Compression (gzip, snappy etc.)?
- Size of the file?
- Storing data in partition for fast processing data?
- Choosing the right file size to store (each file should be between 128 MB to 500 MB size max for better performance)?
- Data lifecycle management or archiving/Purging data
    - Hot?
    - Cold?
    - Archive tier?

What kind of data is being processed?

- Relational vs non-relational?
- Structured or un-structured?
- SQL vs No-SQL?
- Batch vs stream processing?
- If data science work is scoped in?
- Some of the key industry leaders for processing data are Synapse, Databricks and Snowflake. Hence it is very critical to choose the right tool based.
- Are we looking for a unified data platform?
- For stream processing are we looking for an SQL based drag and drop approach? (Azure Stream analytics)
- Are we looking for custom data processing and evaluation with data quality check and error data?

- Are we looking for schema evolution?
- Are we supposed to handle data quality issues?
- Are we supposed to handle ACID transactions?

What kind of programming language development team is comfortable with?

- SQL based (SMP and MPP databases, SQL Endpoint of Databricks)?
- Scala (Databricks)?
- Python (Databricks)?
- Drag and drop approach (Mapping data flow of data factory)?

What should the serving layer look like?

- Dimensional model?
- Lakehouse implementation?

What kind of concurrency of users accessing the system will there be? Concurrency is critical for choosing the serving layer.

Are you looking for a data virtualisation layer?

How will the data be accessed by the end user?

- Exploratory Analysis (Serverless Querying Options)?
- Via reporting?
    - Near real time reporting (i.e., Data virtualisation or Direct Query option in Power BI)
    - Import Mode (data is Loaded into the reporting layer)
- Via API (create API endpoint with Azure Cosmos and Azure Functions)?
- Will the data be accessed across different regions (considered distributed databases)?
- What kind of latency will there be to access data?
- How about consistency (Eventual or strong or any other approach)?

Non-functional requirements questions to ask:

Performance

- What are the performance parameters across multiple layers?
- How fast should the data be able to be accessed from Source to Serving layer?
- Latency for data to be accessed by the end user via the reporting platform or API layer?
- Time taken by the reporting layer to load the report which involves processing and buffer speed at the server side?
- Processing Performance has 2 factors to consider.
    - Time taken for user to access the data?
    - Time taken to load the data into the database?

Security

Infrastructure Security (Azure)

- Do we need private endpoints or service endpoints to secure PAAS services holding data? So that data cannot be accessed outside the network?
- Do we need advanced threat protection or Azure defender?
- Do we need auditing enabled?
- Do we need Azure Monitor and log analytics configured to capture all the logs on the services?
- Do we have data coming in from On-prem to Azure using Express route or Site to Site VPN?
- How keys/passwords/tokens/connection string are managed or stored? (key vault)?
- What are inbound and outbound rules to access data?

Data Security

- How data is security at rest?
- How is data secured in motion or transit?
- How connectivity between different components is secured?
- Are we handling PII data? How is it managed and maintained?
- Is data masking required?
- Do we need to implement row level security?
- Do we need to implement column level security?
- Do we need to implement role-based security?
- How about data sovereignty? Can data reside or go outside a specific region or country?

Reliability

- How reliable are the components on which the data solutions that are being deployed?
- What is the SLA of each component?

Scalability

- How easily we can scale up the solution to handle unpredicted workload, and how we can scale down the solution during non-peak hours?

Agility

- How easily we can expand the solution and on-board new data sources and systems?
- How easily we can deploy changes?
- Continuous integration and deployment and code management using DevOps tools?

High Availability and DR

- Based on RTO, RPO and MTO we need to design the DR solution?
- Can we implement a service DR or a regional DR plan.
- One way to keep solutions highly available is to configure Passive instance for the data solution or using some inbuilt methodology like:
- RA-GRS, GRS, LRS for Storage
- Geo replication or Failover groups for Azure SQL
- Automated restore points, user define restore points and Geo restore for Synapse.
- Re-deploy code-based services like ADF, Databricks from the Azure DevOps Codebase and Infrastructure as Code.

Cost

- Use PAAS services wherever possible.
- With computer and storage separated Pause Compute whenever not used for cost savings.

Governance

- Is there a centralised Data Catalog system?
- Is data classification done?
- Is there a metadata layer information for the business to consume?
- Is the business definition stored centrally?
- Is data lineage implementation done?

It is essential for your business to carefully consider all of these functional and non-functional requirements before you start a new data project to ensure that the solution/s implemented meet all business objectives and save you time and money in the long run. If your business needs help to gather all the requirements and work out the best options, please contact a Fujitsu Data & AI specialist now.

Contact
Fujitsu Data & AI
+61 3 9924 3000