**HLSD Options Analysis**

# HDLC Options Analysis - Detailed Documentation

# Team Composition

| **INITIALS** | **SURNAME** | **GIVEN NAME** | **STUDENT NUMBER** |
| --- | --- | --- | --- |
| NM | Mahagoda | Ninuri | 46592156 |
| QN | Nguyen | Quoc Hung (Alan) | 47578130 |
| AC | Chowdhury | Aasnayem Gazzali (Adam) | 46325824 |
| NK | KHAN | Noorullah | 47197404 |
| TV | Vilathgamuwa | Tashiya | 46992162 |

[**HDLC Options Analysis - Detailed Documentation 1**](#_fl1pzamu4bfa)

[**Team Composition 1**](#_h6a0mdgufdug)

[**Options Overview 2**](#_qz78uo8hxv5e)

[Option 1: 2](#_j8r4fg5iibdy)

[Option 2: 2](#_7xiiqifpjn9e)

[Option 3: 2](#_ky2ok0ad4eut)

[Option 4: 2](#_iaf4olmvt8om)

[**1. Options Ratings 3**](#_cii8cro223g3)

[Option 1: 3](#_vta1waraqnwz)

[Option 2: 3](#_7ib92exnb4a3)

[Option 3: 3](#_tvu1q22sj103)

[Option 4: 4](#_fzv6sls6oayy)

[**Options Pricing Analysis 4**](#_wenulnbmcakm)

[Option 1 4](#_8k6iq8biebno)

[Option 2 4](#_2zbr10c6vzdd)

[Option 3 4](#_y2vx4ryzk01d)

[Option 4 5](#_j3mwcx8d0385)

[**Options Explanation 6**](#_8k1ewm3f3e67)

[Dismissal of option 2: Complexity of transformation 6](#_kiofznhbuxc0)

[Dismissal of option 4: Over-engineered and cost 6](#_mbtunlrn8o9b)

[Option 1 and 3 comparison: 6](#_lkq9rnbk92vk)

[Option 1: 6](#_9e94vt5s9n2m)

[● Advantages: 6](#_8hhlziwtzxiu)

[● Disadvantages: 6](#_wr21wabos8z)

[Option 3: 7](#_qo54lsby5fmg)

[● Advantages: 7](#_ms7wybvozcw)

[● Disadvantages: 7](#_w47agh4ptrf)

[**Strategic Recommendation 8**](#_eixlmneyhf9x)

# Options Overview

## Option 1:

* **Bronze Layer:** Ingest data through Azure Data Lake Storage Gen2 REST API.
* **Silver Layer:** Azure Function to transform and store data
* **Gold Layer:** Azure Function to transform and store data
* **Orchestration:** Azure Data Factory
* **Storage:** ADLS2 for all layers, possibly MySQL database for Silver Layer

## Option 2:

* **Bronze Layer:** Ingest data through Azure Data Lake Storage Gen2 REST API.
* **Silver Layer:** Azure Data Factory’s Data Flows to transform and store data
* **Gold Layer:** Azure Data Factory’s Data Flows to transform and store data
* **Orchestration:** Azure Data Factory
* **Storage:** ADLS2 for all layers, possibly MySQL database for Silver Layer

## Option 3:

* **Bronze Layer:** Ingest data through Azure Data Lake Storage Gen2 REST API.
* **Silver Layer:** Azure Databricks’ notebook to transform and store data
* **Gold Layer:** Azure Data Factory’s Data Flows to transform and store data
* **Orchestration:** Azure Data Factory
* **Storage:** ADLS2 for all layers, possibly MySQL database for Silver Layer

## Option 4:

* **Bronze Layer:** Ingest data through Azure Data Lake Storage Gen2 REST API.
* **Silver Layer:** Azure Synapse Analytics to transform and store data
* **Gold Layer:** Azure Synapse Analytics to transform and store data
* **Orchestration:** Azure Synapse Pipelines
* **Storage:** ADLS2 for all layers, possibly MySQL database for Silver Layer

# Options Ratings

## Option 1:

**Cost**: ★★★★★

* **Explanation**: Azure Functions is very cost-effective for small to medium sized workloads with the first 400,000 GB/s of execution and 1,000,000 executions being free. As our initial dataset is small around 10,000 rows of data which is around 3MB of data for our Bronze layer. Cost will only incur if our data volume or execution amounts increase however, even then for the scale of this project the costs will be negligible.

**Utilisation**: ★★★★☆

* **Explanation**:Azure Functions is very easy to set up and deploy thus, efficient for small-scae projects like ours. Furthermore, you pay as you go so you pay for the resources as you uses them increasing cost efficiency. However, it is not suitable for advanced project when high level of requirements are needed and costs may rise. Ultimately, Azure Function provides good value for the resources utilised and needed for this project.

**Scalability**: ★★★☆☆

* **Explanation**: Azure Functions, while scalable for small to medium datasets once we grow our dataset we may face issues with scalability. The basic consumption plan only has an upper limit timeout duration of 10 minutes hence changing to another plan is more feasible however this also leads to increased costs

**Capability**: ★★★★☆

* **Explanation**: Azure Functions provides flexibility and a wide range of integration making them versatile in many use cases. However, the scalability limitations and cold start latency impact the effectiveness of the solution as the da

## Option 2:

Cost - ★★★★★

* Azure Data factory’s Data Flow is one of the cheaper options compared to the others as it includes a pay as you go costing feature. The estimated cost per run also come to around $0.31with the assumption that a run would take five minutes and that our storage comes to 1.1MB.

Utilisation - ★★★☆☆

* Data Flows offer a user-friendly interface and are effective for many common data transformation tasks. They may be less effective for very complex logic or highly customised tasks, but they cover a broad range of use cases well.

Scalability - ★★★★☆

* Data Flows can scale to handle large volumes of data, but performance may degrade with very complex transformations or high concurrency. Azure Data Factory’s integration with various compute options helps with scalability, but it may require careful management and optimisation.

Capability - ★★★☆☆

* Data Flows are powerful for visual and low-code data transformations, making them capable for most standard tasks. However, they might struggle with highly complex or specialised data engineering needs, where more advanced solutions might be necessary.

## Option 3:

**Cost**: ★★★★☆

* **Explanation**: For small datasets and less complex processing tasks, Azure Databricks and related services can be very cost-effective, often costing less than $5/month. The pricing scales with usage, so you only incur higher costs when your data volume or processing needs increase significantly.

**Utilisation**: ★★★★☆

* **Explanation**: Azure Databricks is efficient for small-scale operations as you only pay for the resources you use. However, as you move to more advanced or continuous tasks, the cost can rise, but it still provides good value for the resources utilised.

**Scalability**: ★★★★★

* **Explanation**: Azure Databricks is highly scalable, capable of handling everything from small datasets to large, complex data processing and machine learning workloads. This makes it an excellent choice for projects that may start small but need to scale up significantly over time.

**Capability**: ★★★★☆

* **Explanation**: Azure Databricks is very feasible for a wide range of projects, particularly those that require both data engineering and machine learning capabilities. The platform's cost-effectiveness for small tasks and scalability for larger ones make it a versatile option, though the complexity of Databricks might require some learning and setup time.

## Option 4:

**Cost**: ★★☆☆☆

* **Explanation**: Although the ADLS2 and Data Orchestration costs are minimal, this option is still expensive due to Synapse Analytics costs. If we eventually increase the size of the dataset, the prices will scale up as well. So, this option will not be feasible.

**Utilisation**: ★★★☆☆

* **Explanation**: Azure Synapse is very extensive with many functions and options. As our dataset is small, not all of the functions will be used. This will lead to inefficiency and the full potential of Azure Synapse will not be utilised.

**Scalability**: ★★★★★

* **Explanation**: Synapse is built to handle large amounts of data. Therefore, an increase in our dataset will not affect the performance of the system .

**Capability**: ★★★★★

* **Explanation**: As already mentioned Azure Synapse is very capable. It is a good choice for data transformation, data preparation and data analytics. Moreover, it can combine both SQL and Spark which allows us to choose between them during different instances.

# Options Pricing Analysis

## Option 1

**Cost Feasibility**

Azure Data Lake Storage Gen2

* combines the power of a Hadoop-compatible file system with an integrated hierarchical namespace with the massive scale and economy of Azure Blob Storage to help speed your transition from proof of concept to production
* Costs are billed based on the number of GB of storage needed and these costs are minimal
* Scalable - As we progress through the bronze, silver, and gold layers the amount of data also increases. Gen2 allows easy scalability of storage as we can increase for each GB of storage needed
* Storage Unlimited - $0.334 per GB

Azure Functions:

* Consumption Plan: Pay only for the time your functions run. Billing is based on number of executions, execution time, and memory used.
* The first 400,000 GB/s of execution and 1,000,000 executions are free.

## Option 2:

* **Azure Data Lake Storage Gen 2:** Above calculation (same as Option 1)
* **Cost Efficient**: Includes Pay as you go pricing option which we will only pay when used.
  + **Estimated Cost per 5-Minute Run**: Calculated based on the cost per vCore-hour for a 5-minute execution time.
    - $0.31 - This price is considerably low due to it being calculated with us having around 1MB storage
* Overall notes: For Azure Data Factory Data Flows, the pricing is primarily based on vCore-hours used during data flow execution and debugging. The cost is $0.467 per vCore-hour for General Purpose instances. Given that each Data Flow run requires a minimum of 8 vCores and typically lasts about 5 minutes, the estimated cost per run is approximately $0.31. This cost-effectiveness is beneficial for small-scale or low-complexity transformations, but it is essential to monitor usage closely to avoid unexpected costs, especially for more extensive or frequent data processing tasks.

## Option 3

* Economic Feasibility:
  + **Cost Efficiency:** For smaller datasets, the costs are minimal, making it economically feasible. However, as data volume and processing needs increase, costs could rise.
  + **Pay-as-You-Go Model:** Azure services offer a pay-as-you-go model, which means you only pay for what you use, adding to the economic feasibility.
  + Price will be under $5/month, $1/month for databricks doing infrequent processing tasks, $1/month for low-frequency orchestration of Azure Data Factory, negligible costs for ADLS2 storage costs.
  + Depending on how frequently BI analytics are generated the costs can increase but very slightly. The total size of data also might have a slight impact. However because the Azure Data Factory and Databricks notebooks are affordable it’s highly unlikely to cost more than $5 per month unless going into a very large scale implementation with several gold layers.

## Option 4

* **Azure Data Lake Storage Gen2 storage** - Storage cost for ADLS2 is minimal. We use Hot Storage which AUD$0.029 per GB. Our file is very small (potentially around 15MB). So, if we estimate 15MB, the costs will be minimal. Write operations are AUD $ 0.10942 per 4MB and Read operations are AUD $0.00873 per 4MB. If we assume 16MB, we will have 4 operations. So, the cost is again minimal. Therefore, this is cost efficient.
* **Synapse Analytics** - The price for Synapse Analytics is AUD $0.237 per vCore-Hour. Therefore, if we assume that we will use 2vCores for 3 hours everyday. That will amount up to (0.237X2X3) = $1.422 per day. Therefore, per month this amounts to AUD$ 42.66 which is very expensive.
* **Data Orchestration** - The data orchestration costs will also be minimal as the pricing is $1.531 per 1000 runs. Due to the size of our dataset, we will not even need 1000 runs, Therefore, the cost is not significant.

# Options Explanation

## Dismissal of option 2: Complexity of transformation

Data Flows are optimised for visual and low-code data transformations which works well for basic data transformation tasks. However, it might struggle with more complex logic and advanced data engineering tasks required in our solution.

Advantages

* **Visual and Low-Code Interface**
  + **Ease of Use**: Data Flows offer a user-friendly, visual design interface, making it simpler to design and manage data transformations without extensive coding.
  + **Rapid Development**: Allows for faster development of data transformations with pre-built components and a drag-and-drop interface.
* **Scalability**
  + **Handle Large Datasets**: ADF Data Flows can process large volumes of data by scaling out to multiple nodes in a cluster.
* **Integration with Azure Ecosystem**
  + **Seamless Integration**: Integrates well with other Azure services like Azure Data Lake Storage (ADLS) and Azure Synapse Analytics.
* **Cost Efficiency for Simple Workloads**
  + **Cost-Effective for Short Runs**: For short, simple data transformations, the cost can be relatively low, especially when using general-purpose clusters.

Challenges

* **Complex Logic and Advanced Transformations**
  + **Performance Issues**: May struggle with complex logic or advanced transformations due to limitations in the visual design approach.
  + **Longer Development Time**: Complex tasks might require more time to implement and test compared to coding directly in a more flexible environment.
* **Cost Considerations**
  + **Short Duration Costs**: Even short executions incur costs, and the overhead of starting and stopping clusters can impact overall cost efficiency.
  + **Minimum Charges**: Costs can add up if running multiple short tasks due to minimum billing increments.
* **Debugging and Maintenance**
  + **Debugging Costs**: Data Flow debugging time is billed, which can add to costs, particularly for complex or frequent debugging.
  + **Maintenance**: Regular updates and maintenance might be needed to ensure that data flows continue to perform well as data complexity or volume changes.

## Dismissal of option 4: Over-engineered and cost

Indeed Azure Synapse Analytics is a powerful tool designed for large-scale data warehousing and big data analytics. Using Synapse for our current initial data size (~ 10,000 rows) would be overkill. The setup and management complexity associated with Synapse would be disproportionate to the benefit we would gain at our current scale.

From a strategic standpoint, we currently have no plan to use a relational database in our solution, so we wouldn’t be able to utilise all Synapse has to offer.

**Advantages of Azure Synapse**

* **Integration of different platforms:** Azure Synapse integrates many different processes in the same platform. Data can be queried using both SQL and Spark based Analytics which makes it an efficient system. In Azure Synapse, data engineers, data modellers, and business analysts can collaborate and work on data in the same platform.
* **Scalability:** As already mentioned above, Azure Synapse can handle a large volume of data. Therefore, scaling up will not be a problem and the performance will still be excellent.
* **Quality Analytics:** Azure Synapse Analytics is a powerful tool. Different types of data transformation, data analytics, and data preparation techniques can be performed using Azure Synapse Analytics. Therefore, this is optimal for high performance.

**Challenges of Azure Synapse**

* **Higher Costs:** Compared to the other options, Azure Synapse is more expensive. Synapse Analytics is generally expensive. According to our estimates, this will cost around AUD$45 per month which is significantly more than the other options.
* **Learning Complexity:** As Azure Synapse integrates different services, it can be really challenging to maintain and manage the many different services. If the team is unfamiliar with the services, it will be more challenging to train them about the different systems and processes.
* **Underutilised:** As our dataset is relatively small, not all of the features can be used. Therefore, investing in Azure Synapse will not be fruitful as the services will not be utilised properly and the company will not get their value for money.

## Option 1 and 3 comparison:

### Option 1:

#### Advantages:

* + **Cost-effective:** Azure Functions are generally more cost-effective for small to medium scale workloads. Since we are dealing with a rather small dataset initially (~ 10,000 rows), Azure Functions help minimise costs.
  + **Simplicity and speed:** Azure Functions are easy to set up and deploy, which can lead to faster development times.

#### Disadvantages:

* + **Scalability limits:** As our dataset grows, Azure Functions might face scalability challenges. Specifically it has an upper limit timeout duration of 10 minutes. Technically we could change our plan to have dedicated resources but by then, using another platform will be more beneficial.
  + **Cold start latency:** Since Azure Functions are serverless, it will introduce latency due to cold starts, which could affect our performance as the workload increases.

### 

### Option 3:

#### Advantages:

* + **Scalability and Flexibility:** Azure Databricks is designed for large-scale data processing and can easily handle complex transformations and large datasets. This makes it a future-proof option as our data grows.
  + **Advanced Data Processing Capabilities:** Databricks offers powerful data engineering and machine learning capabilities, making it ideal for more complex workflows that may emerge as our project evolves.
  + **Seamless Integration (including Machine Learning Integration):** The integration between Azure Databricks, Azure Data Factory, and ADLS2 is designed to ensure smooth data flow and processing. This seamless integration not only supports efficient data engineering tasks but also makes Azure Databricks an excellent choice for machine learning (ML) integration. If Group 13 chooses to leverage Databricks for their ML tasks, they can benefit from having both data engineering and ML workflows within the same environment, simplifying the overall process. Additionally, Azure Databricks integrates effortlessly with other Azure services, making its implementation within the Azure ecosystem straightforward and efficient.
  + **Strategic Growth:** By starting with Databricks, we prepare for future scalability and the ability to handle more complex transformations without needing to overhaul the pipeline later.
  + **Flexibility:** The combination of Databricks for advanced data processing and Azure Data Factory for orchestration allows for flexible data transformations and workflows.
  + **Flexible Orchestration:** Azure Data Factory allows for sophisticated orchestration of data pipelines, ensuring efficient and automated data processing workflows.

#### Disadvantages:

* + **Higher Costs**: While initial costs are low for small datasets, the cost of using Azure Databricks can increase significantly as data volume and processing requirements grow, particularly with continuous use of Databricks and Data Factory. Azure Databricks tends to be more expensive than Azure Functions due to its advanced capabilities, but this cost is often justified by the performance, scalability, and integration benefits, especially for data-intensive and machine learning workloads.
  + **Complexity:** The setup and management of Azure Databricks, along with the orchestration in Azure Data Factory, can be complex and may require specialized knowledge and skills.
  + **Learning Curve**: Databricks requires more expertise to set up and manage, which could lead to a longer development time initially.

# Strategic Recommendation

* **Short-Term**: **Option 1 (Azure Functions)** is the more cost-effective and simpler solution for our current needs, given the small size of our dataset. It allows for rapid deployment and low costs while still meeting our immediate processing needs. However, Azure Functions lacks the interactive development experience that Azure Databricks’ notebooks have to offer.
* **Long-Term**: **Option 3 (Azure Databricks for Silver Layer, Data Flows for Gold Layer)** is strategically more advantageous if we anticipate significant data growth, require more complex data processing, or plan to integrate machine learning tasks. By starting with Databricks, we position ourselves for seamless scaling, advanced data engineering capabilities, and potential ML integration, making it a more future-proof solution.

Given our initial data size, it would be prudent to start with Option 1 to minimise costs and complexity. However, we should plan to transition to Option 3 as our data grows and our processing needs become more demanding, particularly if ML tasks become a focus. This approach allows us to balance immediate efficiency with long-term strategic growth.