

**iLEVEL Snowflake Integration**

Architecture Review – Design Stage

The document is structured into five sections: Business Context, Ideation Stage, Design Stage, Deployment Stage, and Go-to-Market Stage. Each stage, except for Business Context, involves four review stages that are presented to the Architecture Review Board (ARB).

The Business Context section is compulsory and forms part of the initial stage, which focuses on Ideation. Our approach involves using a single document and completing the relevant sections pertinent to each stage of review.

Please fill the applicable review stage and mark the others as “Not Applicable.”

Authors:

Sandeep Khatri

Contributors:

Brian Matuszewski

Grace Li

Brian Lee

# Project Summary

## References

|  |  |  |  |
| --- | --- | --- | --- |
| Title | URL | Date | Author |
| Product Opportunity Document | [PMT-N-137 Product Opportunity - Snowflake Integration.pdf](https://spgl.sharepoint.com/:b:/r/sites/team_nam_mi_naf_m/Shared%20Documents/ARB/iLEVEL/Snowflake%20Integration/PMT-N-137%20Product%20Opportunity%20-%20Snowflake%20Integration.pdf?csf=1&web=1&e=wrBcOI) | 09-Sep-2024 | * Brian Matuszewski * Grace Li |

## Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Key Changes |
| 1 | 09-Sep-2024 | Sandeep Khatri | Initial Draft |
| 2 | 19-Sep-2024 | Sandeep Khatri | Added availability information. |
| 3 | 01-Apr-2025 | Atul Kumar Suyal | Added details in design stage section for review. |
| 4 | 06-May-2025 | Sandeep Khatri  Atul Kumar suyal | Added details for-   1. Data sync status update flow (Section - Logical Architecture) 2. Sequence diagram for data-sync process 3. Functionalities of the Snowflake Adapter (section- Component view) |

## Glossary

|  |  |
| --- | --- |
| Term | Definition |
| ARB | Architecture Review Board |
| ADR | Architecture Decision Record |
| ES | Enterprise Solutions |
| ACV | Annual Contract Value |
| CA | Client Admin |
| GA | General Admin |

Table of Contents

[Project Summary 3](#_Toc195719755)

[References 3](#_Toc195719756)

[Change History 3](#_Toc195719757)

[Glossary 3](#_Toc195719758)

[Business Context 8](#_Toc195719759)

[Requirement Summary 8](#_Toc195719760)

[Key Stakeholders 9](#_Toc195719761)

[Cost and Revenue 9](#_Toc195719762)

[Ideation Stage 11](#_Toc195719763)

[Logical diagrams / workflow concepts 11](#_Toc195719764)

[Technology choices under consideration 11](#_Toc195719765)

[New technology or Proof of Concept (POC) requests 11](#_Toc195719766)

[Initial technology cost assessment (high-level) 12](#_Toc195719767)

[High level overview of resiliency requirements 13](#_Toc195719768)

[Risks 15](#_Toc195719769)

[Design Stage 17](#_Toc195719770)

[Architecture Analysis 17](#_Toc195719771)

[Enterprise Architecture Analysis 17](#_Toc195719772)

[Project Requirement Analysis 20](#_Toc195719773)

[Logical Architecture 24](#_Toc195719774)

[Functional View 24](#_Toc195719775)

[Data View 27](#_Toc195719776)

[S&P Global Data and Content (Pre-existing and New) 27](#_Toc195719777)

[3rd Party Data 27](#_Toc195719778)

[Interfaces 27](#_Toc195719779)

[Component View 27](#_Toc195719780)

[Component Decomposition 27](#_Toc195719781)

[Service Resilience 29](#_Toc195719782)

[Business Continuity / Disaster Recovery Governance 29](#_Toc195719783)

[Architectural Dependencies and Constraints 29](#_Toc195719784)

[Physical Design 30](#_Toc195719785)

[Deployment Architecture 30](#_Toc195719786)

[Capacity Planning and Scalability 32](#_Toc195719787)

[Software Component and Device Failover 32](#_Toc195719788)

[System Management View 32](#_Toc195719789)

[Instrumentation and Monitoring 32](#_Toc195719790)

[Migration Strategy 33](#_Toc195719791)

[Assets to be Sunset or End-of-Life 33](#_Toc195719792)

[New Technologies 33](#_Toc195719793)

[Material Costs 33](#_Toc195719794)

[Risks 35](#_Toc195719795)

[Technical Risks 35](#_Toc195719796)

[Service Risks 35](#_Toc195719797)

[Architecture Scorecard 36](#_Toc195719798)

[Architecture Decision Record (ADR) 37](#_Toc195719799)

[NFR 37](#_Toc195719800)

[Deployment Stage 39](#_Toc195719801)

[Identify any changes in the project scope or design from previous stages. 39](#_Toc195719802)

[Business verification of delivery of requirements 39](#_Toc195719803)

[Instrumentation and operability 39](#_Toc195719804)

[Identified and tested promotion strategy (canary, blue green) 39](#_Toc195719805)

[Identified and tested disaster recovery / rollback strategy 39](#_Toc195719806)

[Go to Market Stage 41](#_Toc195719807)

[Ongoing maintenance and support requirements 41](#_Toc195719808)

[Proof of measurement of system performance and stability requirements 41](#_Toc195719809)

[Security controls 41](#_Toc195719810)

[Business readiness 41](#_Toc195719811)

# Business Context

iLEVEL clients today struggle to extract their data from iLEVEL to their Snowflake data warehouse, wherein they combine data from all software/data providers that they use. The process today relies on the client using a combination of the iLEVEL REST API and custom connectors, ETL or other manually coded solutions to support connectivity. It is critical we solve for this with urgency as it is becoming ever more important for our clients to centralize their data efficiently and accurately.

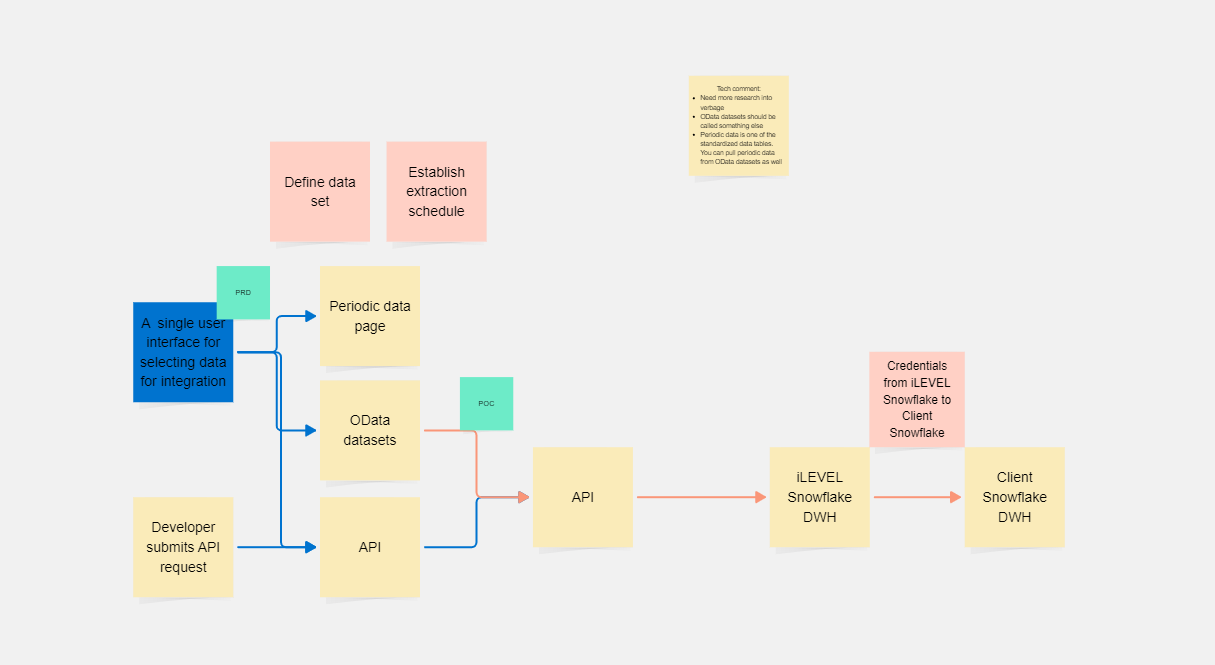
While there are other data warehouse providers in the marketplace in addition to Snowflake, our research shows that our clients use Snowflake more than any other provider. Moreover, S&P has a strong working relationship with Snowflake making this partnership and obvious starting point. That being said, we ultimately plan to support alternative data warehouses, in addition to Snowflake, in the future.

The cost of not doing this is falling behind competitors and losing the trust of our clients. Several of our competitors already support a similar out-of-the-box connectivity offering. Clients are increasingly coming to expect this capability from iLEVEL.

## Requirement Summary

There are two main, high-level requirements for this integration:

1. An iLEVEL client/user must be able to configure a dataset which they want to access in Snowflake. This also included an extraction schedule.
   1. iLEVEL has a version of this today: “Named Views” which defines a dataset that can then be shared via OData or SFTP.
   2. iLEVEL also offers some out-of-the box views, such as “Periodic Data”, that client can potentially extract also on top of the client-customized views.
2. The defined dataset needs to accessed by clients’ Snowflake data warehouse
   1. During preliminary conversations with the Snowflake team, we have explored a variety of connection options, including Snowpipe, a direct integration with each client’s specific Snowflake environment, a Native app, and more. After reviewing the iLEVEL use case closely, we honed in on Snowflake Managed Private listing most optimal method.
   2. This would involve syncing the client data first to an iLEVEL data warehouse in Snowflake instance managed by S&P. From there, data will be shared with customer via Snowflake private listing.
   3. An open question is whether there would be one singular iLEVEL Snowflake data warehouse provisioned to all the individual client data warehouses OR if there would be one iLEVEL Snowflake data warehouse per client data warehouse.
   4. See below for a high-level diagram illustrating this:



Additional information contained in:

[Product Opportunity Document](https://spgl.sharepoint.com/:b:/r/sites/team_nam_mi_naf_m/Shared%20Documents/ARB/iLEVEL/Snowflake%20Integration/PMT-N-137%20Product%20Opportunity%20-%20Snowflake%20Integration.pdf?csf=1&web=1&e=hLlT2S)

## Key Stakeholders

* Business Sponsor: Christopher Sparenberg (chris.sparenberg@spglobal.com)
* Lead Architect: Sandeep Khatri (sandeep.khatri@spglobal.com)
* Solution/System Architect: Brian Lee (brian.lee2@spglobal.com)
* Product Owner: Brian Matuszewski (brian.matuszewski@spglobal.com)
* Technical/Development Lead: Michael Stuart (michael.stuart@spglobal.com)

## Cost and Revenue

The costs of this initiative obviously include the development effort of building the integration. The initial and ongoing costs to Snowflake are seemingly low to non-existent based on discussions we’ve had. There would also be a platform cost to support the creation and syncing of datasets. It is possible some or all of these costs can be offset via a pay-as-you-use model to clients.

This integration de-risks the loss of existing client ACV (Annual Contract Value). As mentioned earlier, many of our competitors are providing this capability and our inability to do so represents a real risk in terms of client churn. Further, the clients that need this type of integration the most are some of our largest clients by ACV. If even two of these clients were to churn, we could be looking at a 7-figure loss in ACV.

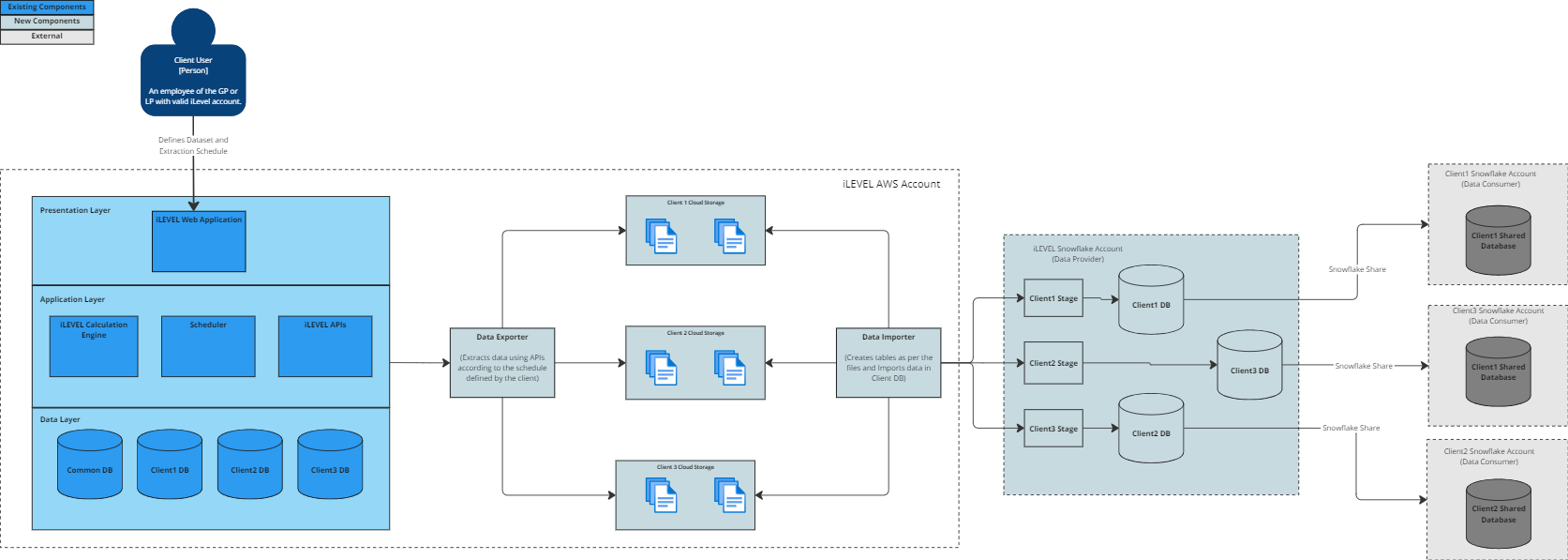
A Snowflake integration also provides a strong selling point to market prospects. In our discussions internally, it is apparent that prospects are seeking this type of integration in a portfolio monitoring product and an integration with a name like Snowflake carries significance.

Architecture Decision Record (ADR)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Title | Context | Decision | Consequences | Status | Comments |
|  |  |  |  |  |  |

# Ideation Stage

## Logical diagrams / workflow concepts



## Technology choices under consideration

Clients of iLEVEL currently extract their data from the iLEVEL platform to their Snowflake data warehouses, where they integrate information from various software and data providers. This process typically involves utilizing a combination of the iLEVEL REST API and custom connectors, ETL (Extract, Transform, Load) solutions, or other manually coded methods to facilitate connectivity.

While there are many data warehouse providers available in the market beyond Snowflake, our research indicates that Snowflake is the preferred choice among our clients. Consequently, the objective of this initiative is to achieve a seamless integration between iLEVEL and Snowflake.

## New technology or Proof of Concept (POC) requests

**Planned POCs:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PoC 1:**

*Objective* – The objective is to prove out the concept of dynamically creating/updating Snowflake table structures to match the iLEVEL widget that is created that needs to be imported. This should be done via API so there is no user interaction in the synchronization of the table structure.

The iLEVEL widget is a part of a dashboard where we see key investment data like fund performance, cash flows, and portfolio company metrics. It helps users quickly understand how their investments are doing by showing charts, tables, and summaries in a clear, interactive format.

[Detailed POC Requirement Doc](https://spgl.sharepoint.com/:b:/r/sites/team_nam_mi_naf_m/Shared%20Documents/ARB/iLEVEL/Snowflake%20Integration/PMT-N-181%20Snowflake%20POC.pdf?csf=1&web=1&e=DeWge9)

## Initial technology cost assessment (high-level)

The primary data ingested into Snowflake consists of calculated data. This calculated data will be sourced from the existing APIs. Presented below are the key items of the solution from a cost perspective.

* Snowflake Cost
* Infra upgrade Cost

At this stage, there are some uncertainties, such as common customer usage patterns and load frequencies, both of which will impact the cost. Below is a high-level cost estimate based on certain assumptions regarding customer usage patterns. This estimate will be further refined as we progress through the development cycle, particularly following the completion of the proof of concept (POC).

**Snowflake Cost:**

|  |  |
| --- | --- |
| **Storage Cost** | |
| Storage Cost Per Region (in USD) | 330 |
| No of Regions | 4 |
| **Total Storage Cost per year (in USD)** | 1320 |

|  |  |
| --- | --- |
| **Transfer Cost** | |
| Data Transfer Cost (ingress) | 0 |
| Data Transfer Cost (egress)  Primary Region to Secondary Region Transfer cost | 600 |
| **Total Storage Cost per year (in USD)** | **600** |

|  |  |
| --- | --- |
| **Data Loading Cost (Compute)** | |
| Warehouse size | M ->XS |
| Frequency Per Year | 365 |
| Time per day (in hours) | 2 |
| Warehouse Credit/hour | 4 |
| Total Credit per year | 2920 |
| **Total Cost per Year**  **(Enterprise Edition)** | **8760** |

|  |  |
| --- | --- |
| **Internal Reporting (Compute)** | |
| Warehouse size | XS |
| Frequency Per Year | 240 |
| Time per day (in hours) | 2 |
| Warehouse Credit/hour | 1 |
| Total credit per year | 480 |
| **Total Cost per Year**  **(Enterprise Edition)** | **1440** |

|  |  |
| --- | --- |
| **Customer Facing Analytics (Compute)** | |
| Warehouse size | XS |
| Frequency Per Year | 365 |
| Time per day (in hours) | 10 |
| Warehouse Credit/hour | 1 |
| Total credit per year | 3650 |
| **Total Cost per Year**  **(Enterprise Edition)** | **10950** |

**iLEVEL Infra upgrade Cost:**

The cost associated with infrastructure upgrades includes the expenses for new components within iLEVEL, as well as the costs associated with upgrading the iLEVEL API components and the calculation engine.

A cost assessment of the iLEVEL infra upgrade cost can be provided following the completion of the proof of concept (POC).

## High level overview of resiliency requirements

The proposed solution should meet or exceed the current resiliency requirements to ensure that it delivers a comparable or enhanced level of reliability and robustness.

* Fault Tolerance: Load-Balanced Multi-Zone Deployment.
* Redundancy:
  + Active–Passive Deployments. There are two deployments, one for clients in the US and another for clients in Europe.
  + Primary Region à us-east-1 (N. Virginia) and eu-west-1 (Ireland)
  + Secondary Region à us-east-2(Ohio) and eu-central-1 (Frankfurt)
* Scalability: Manual and horizontal scalable.
* Availability: 99.5%
  + RTO 8 hours
  + RPO 30 mins
* Security
  + Computational resources are shared among clients. However, for storage, separate client silos (databases) are established within iLEVEL.
  + Clients can only access their own data and are restricted from viewing data belonging to other clients. Access to data is controlled based on the permissions granted through their iLEVEL accounts.
  + Encryption – Storage Level Encryption

## Risks

None

End of Ideation Stage Section.

# Design Stage

## Architecture Analysis

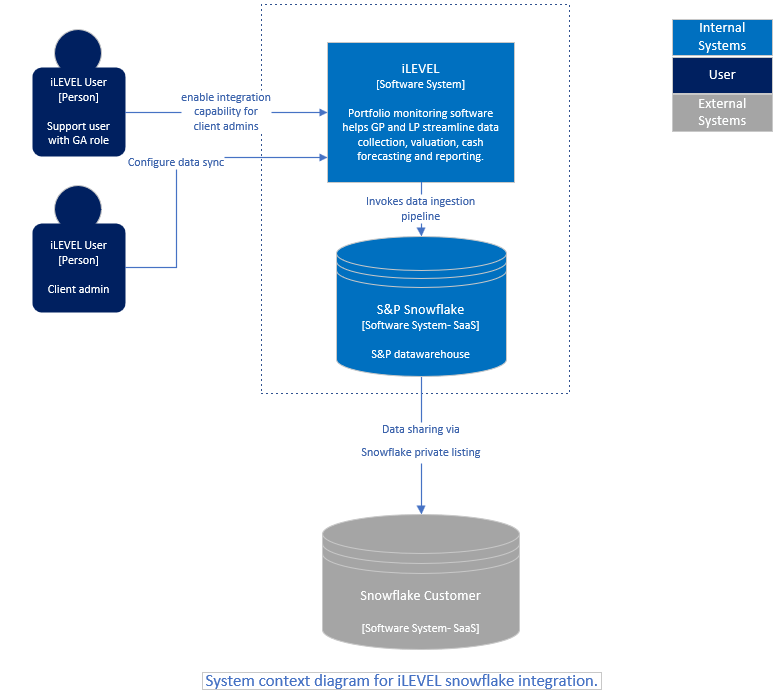
### Enterprise Architecture Analysis

The architecture leverages the capabilities of Snowflake to efficiently ingest data from iLEVEL and distribute it to Snowflake client instances. This approach loads iLEVEL data in the S&P managed Snowflake instance before distributing it via Snowflake private listing.

Support user having GA (General Admin) capability will enable the Snowflake Integration capability for a client admin. Once the capability is enabled, client admin configures a data sync process including setting a schedule for regular syncs.

Existing scheduler enging in iLEVEL System will export data according to the provided schedule and store extracted data files into S3 Storage. In S3 storage, data will be segregated logically using the prefixs (folders) for individual customer. Then Snowflake adapter (part of iLEVEL System – data ingestion pipeline) will push the data into S&P Snowflake instance using external tables. There will be separate DB in S&P Snowflake for individual customer. Finally, the data will be shared with the client’s Snowflake account via Snowflake Private Listing.

Each folder within the S3 bucket will have a unique identifier to map it to a specific customer. To control access to the S3 storage, an IAM role will be configured as part of the Snowflake storage integration, ensuring that only authorized processes can access the designated storage.

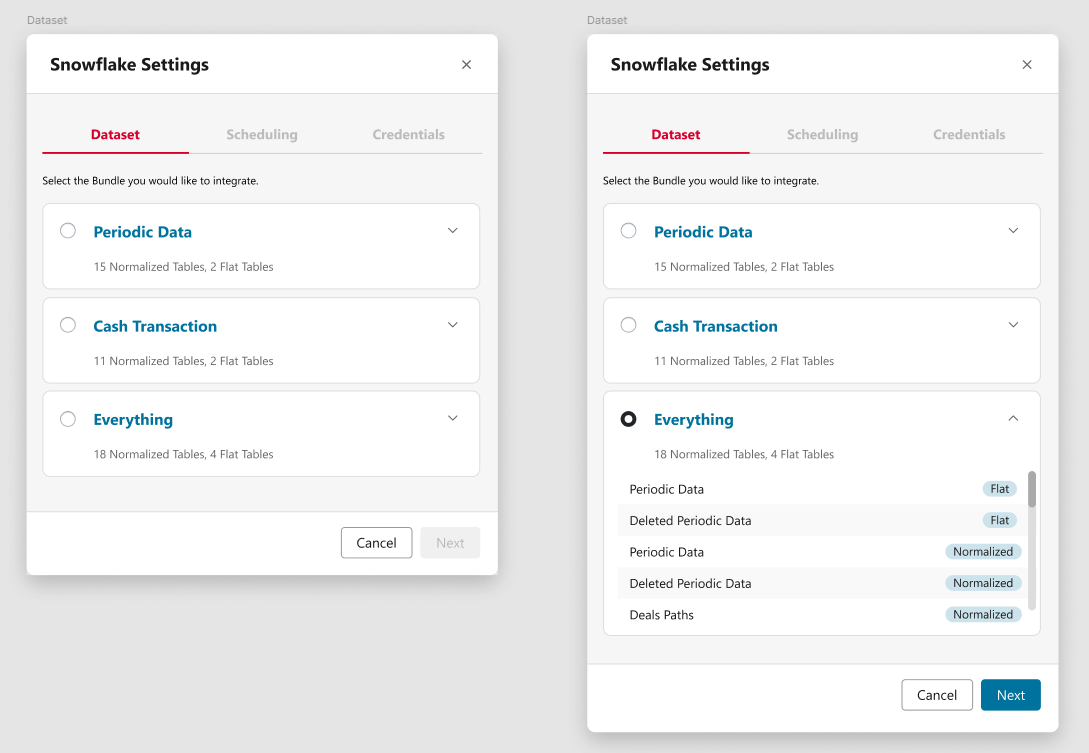


[System context visio link](https://spgl.sharepoint.com/:u:/r/sites/team_nam_mi_naf_m/_layouts/15/Doc.aspx?sourcedoc=%7B36BEA34E-65CB-4FCD-B021-A1F56A782BE3%7D&file=C4.vsdx&action=default&mobileredirect=true)

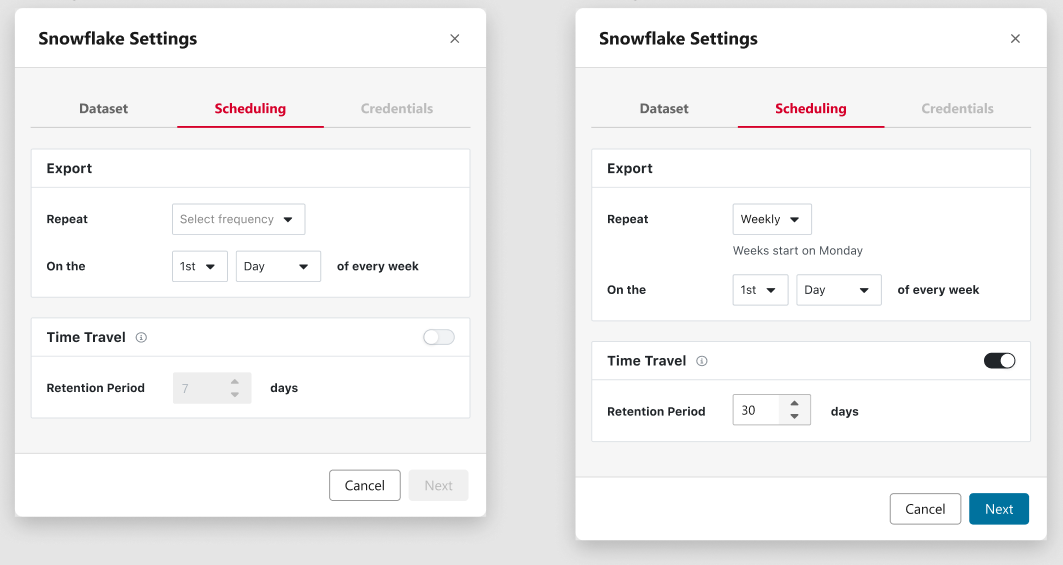
**Snowflake integration UI**

The following user interfaces in iLEVEL are used to configure the iLEVEL–Snowflake integration.

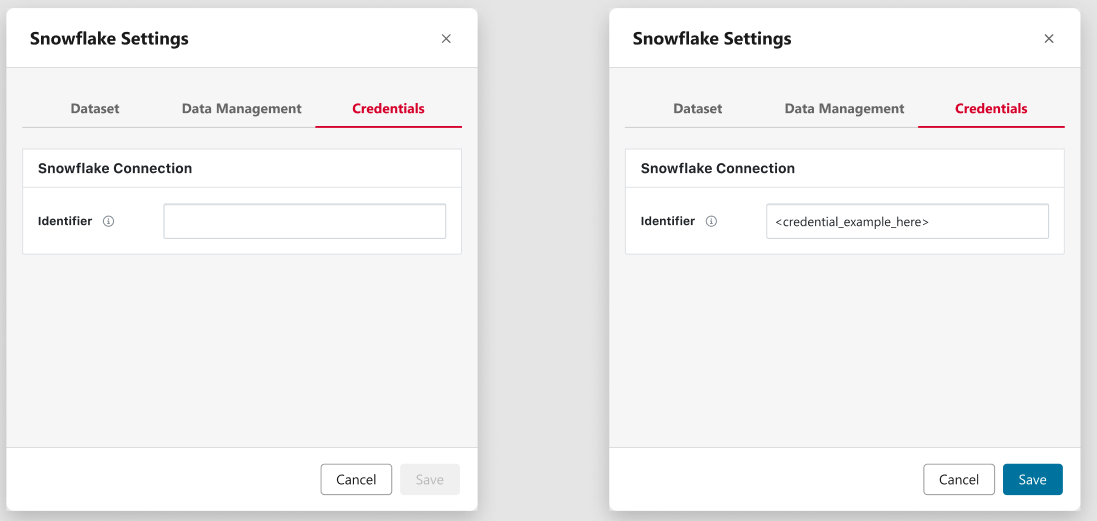
Dataset:



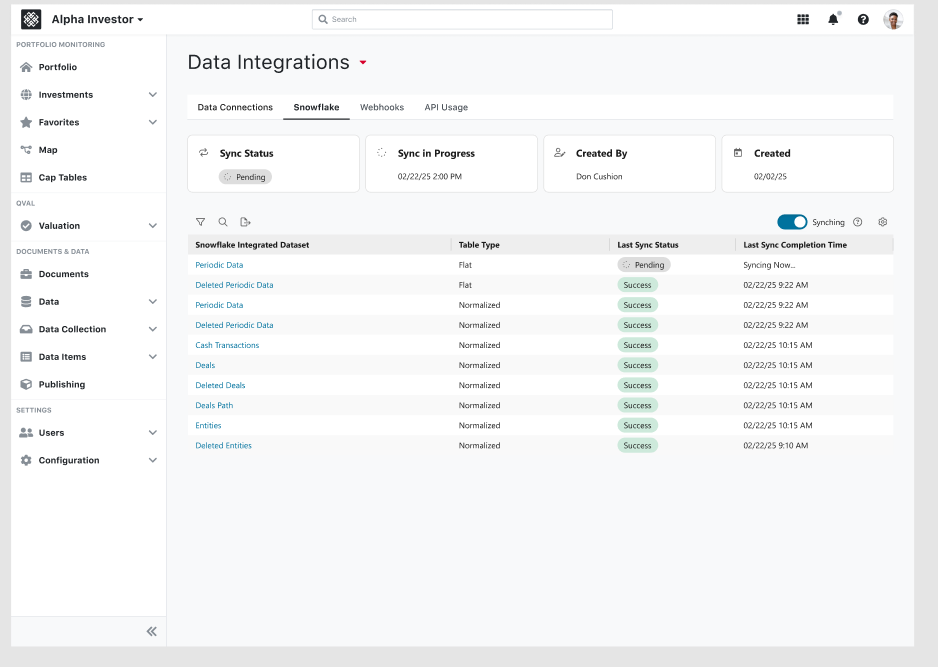
Scheduling:



Credentials configurations:



Activity logs:



### Project Requirement Analysis

#### Key Performance Indicators

* Data in Customer Snowflake needs to be 100% accurate compared to iLEVEL.

Ensure sync happens in a 100% manner between S3 storage and S&P managed Snowflake instance so that end customer can access the accurate data via Snowflake private listing.

* Syncing multiple datasets for multiple clients around the same time should not impact overall performance.

#### Capacity Forecast and Scheduled Availability

Year 1:

* 5 large clients
* 15 medium clients
* 5 small clients

Year 2 (# new clients added):

* 5 large clients
* 20 medium clients
* 5 small clients

Year 3 (# new clients added):

* 3 large clients
* 10 medium clients
* 3 small clients

Year 3 total # clients: 71 (13 large clients, 45 medium, 13 small)

Client size reference-

|  |  |
| --- | --- |
| Client Size | Periodic data points |
| XS | < 1M |
| Small | 1M |
| Medium | 1M-50M |
| Large | > 50M |

Availability- same as iLEVEL- 99.5%

#### Operating Model

Operating hours of the systems are 24 X 7.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1st Line | 2nd Line | 3rd Line | Others (Brief description) |
| **Customer Managed** |  |  |  |  |
| Third Party support |  |  |  |  |
| S&P Global support |  |  |  |  |
| **S&P Global Managed** |  |  |  |  |
| Third Party support |  |  |  |  |
| S&P Global support | iLEVEL CS | iLEVEL  Engineering | iLEVEL Engineering | Snowflake accounts are provisioned through the MI Central Team. The Central Teams may be involved in Level 3 support, if necessary. |
| **S&P Global Hosted** |  |  |  |  |
| Third Party support |  |  |  |  |
| S&P Global support |  |  |  |  |

#### Architectural Dependencies

* Snowflake: We will use Snowflake hosting on AWS in US and EU region.
* Sailpoint: Access to Snowflake will be controlled by Sailpoint.

#### Admin Requirements

* DevOps to prepare initial infra creation script for different environments. This will include all the one time activities like storage integrations and other relevant objects in the S&P managed Snowflake primary and secondary region.
* TechOps to enable the Snowflake integration capability for a client admin.
* Client admin to configure data sync process.
* DevOps and TechOps reviews application user access regularly to comply with audit controls.
* In event of data sync failure, TechOps team will analyse the issues and take the appropriate action.
* In the event of an outage in the primary region, the TechOps team will transition to the secondary region for failover and will revert back to the primary region once the outage is resolved.

#### Security

1. **Authentication and authorization:** 
   1. To control access to manage Snowflake integration settings on the UI, a new capability- “Snowflake Integration” will be added in iLEVEL. If this capability is enabled for a user, then the user will be able to see a designated Snowflake Integration Configuration UI.
   2. No client will be able to access S&P managed Snowflake account. Only DevOps and TechOps team would be able to access this account. Access to Snowflake will be controlled by SailPoint.
   3. Customer specific data will be stored in separate DBs in Snowflake and will be shared with customer using Snowflake provided “Private Listing”.
   4. There will be no changes to the authentication and authorization processes for iLEVEL.
   5. iLEVEL will use key-pair authentication to access Snowflake. Keys will be stored in the key-vault and rotated accordingly.

reference link- [Best Practices - DSE - Big Data Tech - Confluence](https://spglobal-mi.atlassian.net/wiki/spaces/DBDT/pages/291295843/Best+Practices)

[Authenticating to the server | Snowflake Documentation](https://docs.snowflake.com/en/developer-guide/sql-api/authenticating#label-sql-api-authenticating-key-pair)

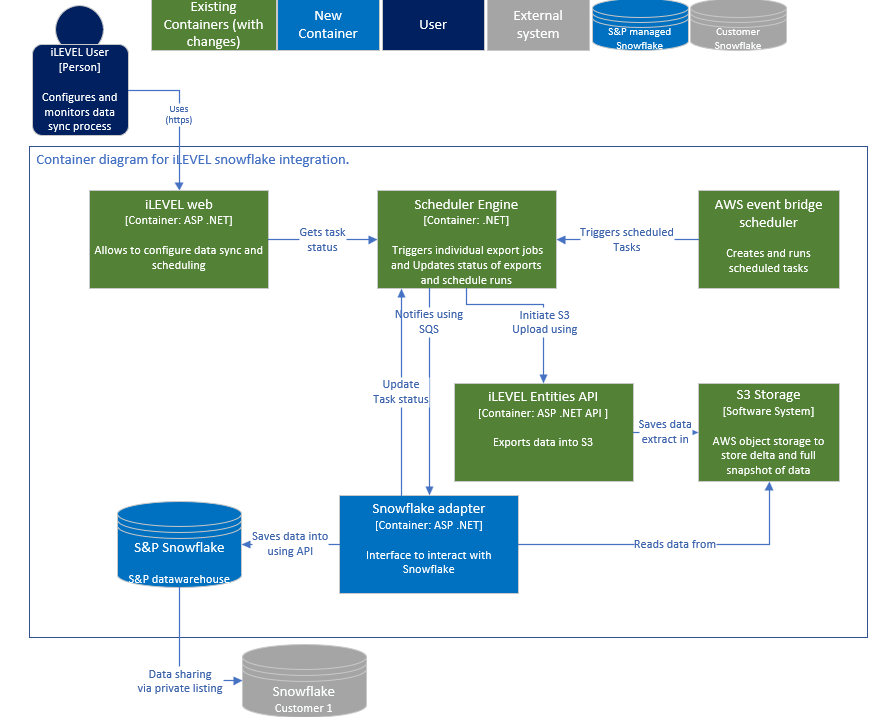
1. **Data in transit:** All communication between iLEVEL to S&P managed Snowflake instance will happen via TLS**.** Data will be shared with customer using Snowflake secure private listing.
2. **Data at rest**: iLEVEL data in Snowflake will be secured using tri-secret.

Tri-Secret Secure is the combination of a Snowflake-maintained key and a customer-managed key in the cloud provider platform that hosts your Snowflake account to create a composite master key to protect your Snowflake data. CMK will be rotated automatically on preconfigured schedule. [Reference link.](https://docs.snowflake.com/en/user-guide/security-encryption-tss)

iLEVEL data in the Snowflake external stage (S3) will be encrypted using client-side encryption. [Reference link](https://docs.snowflake.com/en/user-guide/security-encryption-end-to-end).

1. **Network security:** Network policies will be implemented in Snowflake to allow the traffic from whitelisted IPs**.** Security will be implemented as per the guidelines provided by Snowflake platform team. [Reference link.](https://spglobal-mi.atlassian.net/wiki/spaces/DBDT/pages/291295854/Responsibility+Matrix+-+Central+Team+vs+Account+Owners+Teams)
2. **Data segregation:** There would be customer specific folder in S3 bucket for data ingestion. Separate Snowflake DB will be created per customer to keep data segregated.
3. **Avoid logging sensitive data:** Keys, passwords, personally identifiable information (PII) etc. will not be logged during logging.
4. **Use key vaults to store secrets and keys:** Any secrets/keys needed by application will be stored in key vault.

## Logical Architecture



[Visio link.](https://spgl.sharepoint.com/:u:/r/sites/team_nam_mi_naf_m/_layouts/15/Doc.aspx?sourcedoc=%7B36BEA34E-65CB-4FCD-B021-A1F56A782BE3%7D&file=C4.vsdx&action=default&mobileredirect=true)

High level flow:

* An iLEVEL User with CA (Client Admin) role will communicate with iLEVEL web Portal in order to configure a data sync process.
* AWS event bridge scheduler will invoke schedular engine via SQS (scheduler queue) as per the configured schedule. Scheduler engine has a background job that will continuously poll the messages in schedular queue. Schedular engine is an existing container in iLEVEL system.
* On receiving the Snowflake data sync message in the schedular queue, schedular engine will initiate data extraction and upload in S3 in JSON format using existing iLEVEL entities API.
* After files are prepared in S3, schedular engine will notify Snowflake Adapter (using SQS) which is responsible for ingesting/syncing the data into S&P Snowflake Account.
* Once data sync is complete, Snowflake adapter will notify the status of sync process to scheduler engine.
* The scheduler engine provides an existing API to retrieve the status of scheduled tasks. The ILEVEL UI will use this API to obtain the status of the data sync process.
* Clients will access data through Snowflake Private Listing.

Exception handling and process reliability:

* Exception handling- Snowflake adapter includes a configurable retry mechanism for Snowflake API calls. If all retry attempts are exhausted without success, the sync operation will be marked as failed.
* Process reliability- Duplicate message processing will not have any adverse impact due to the use of the following two Snowflake commands:

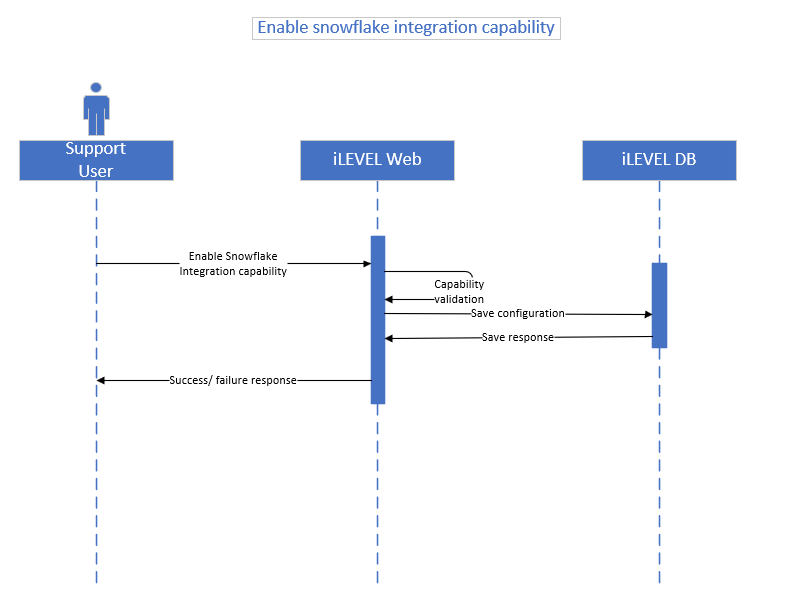
Insert overwrite – used for full snapshot loads

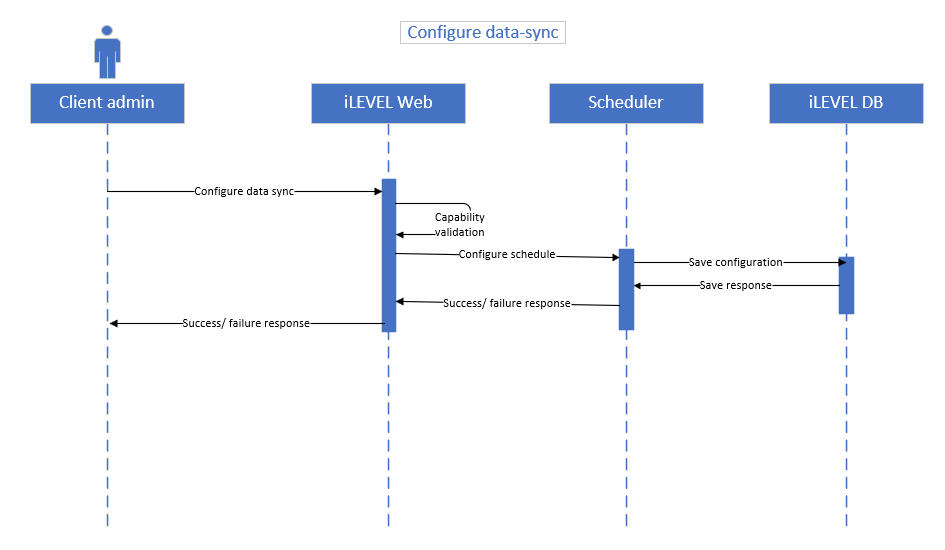
If a duplicate message is received, the entire table is overwritten with the latest snapshot. This ensures data consistency and avoids any issues.

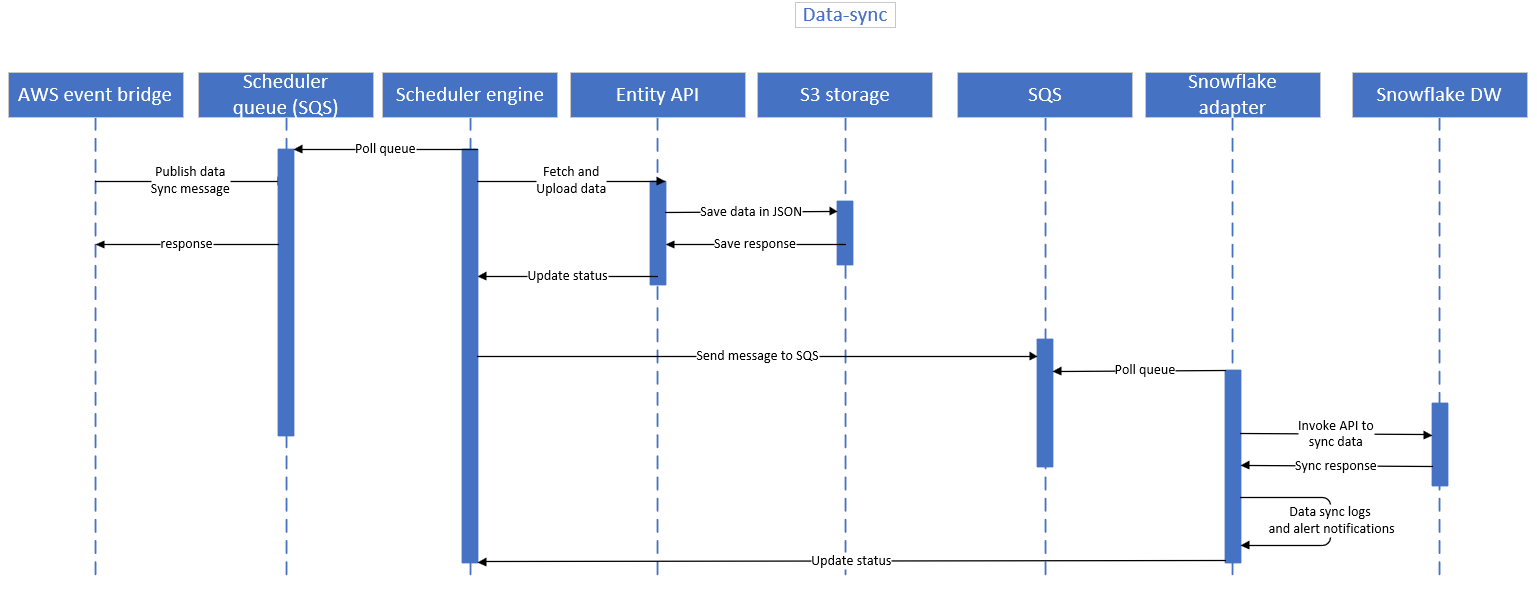
Merge– used for delta (incremental) loads

Merging is performed using a unique identifier. If a duplicate message is processed, the same record may be updated more than once, but the end state of the data remains correct.

### Functional View







[Visio link.](https://spgl.sharepoint.com/:u:/r/sites/team_nam_mi_naf_m/_layouts/15/Doc.aspx?sourcedoc=%7B36BEA34E-65CB-4FCD-B021-A1F56A782BE3%7D&file=C4.vsdx&action=default&mobileredirect=true)

* 1. Support user with GA role will enable the Snowflake Integration capability for client admins using iLEVEL UI.
  2. Client admins (having Snowflake integration capability) will configure data sync using the UI available in iLEVEL. This will include the data sync schedule and information about Snowflake account (where data need to be synced)
  3. An automatic process will sync the data from iLEVEL to S&P Snowflake and then data will be shared to the customer using private listing data sharing capability.

#### Administration View

* General Admin accounts will have the ability to enable Snowflake through the iLEVEL interface. They will also have the ability to add/update/delete Snowflake settings for each client.
* Client admin will have ability to configure the data sync process and schedule.

#### User Management Systems

In this section state which S&P Global standard systems are used for:

* Registration of new users = **No change -** **existing functionality in** **iLEVEL**
* Password management = **No change -** **existing functionality in** **iLEVEL**
* Permissions or entitlements = **No change -** **existing functionality in** **iLEVEL**
* Session Management (e.g., cookies and tokens) = **No change -** **existing functionality in** **iLEVEL**
* Snowflake access management = **SailPoint will be used to manage the access.**

## Data View

|  |  |
| --- | --- |
| Does this project create any new data / content? | No |
| Does this project consume any 3rd party data directly? | No |
| Does this project transform or normalize data? | Yes |
| Does this project introduce new or change existing relationships between data entities? | No |
| Is any data or service originating within this project to be made available either internally or externally? | Yes |
| Does this project introduce new or change existing customer-facing interfaces? | Yes |

### S&P Global Data and Content (Pre-existing and New)

Not applicable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Set Name | Source | Version | Changes in Upstream[[1]](#footnote-2) | Changes in Downstream[[2]](#footnote-3) | Public or Private |
|  |  |  |  |  |  |

### 3rd Party Data

Not applicable

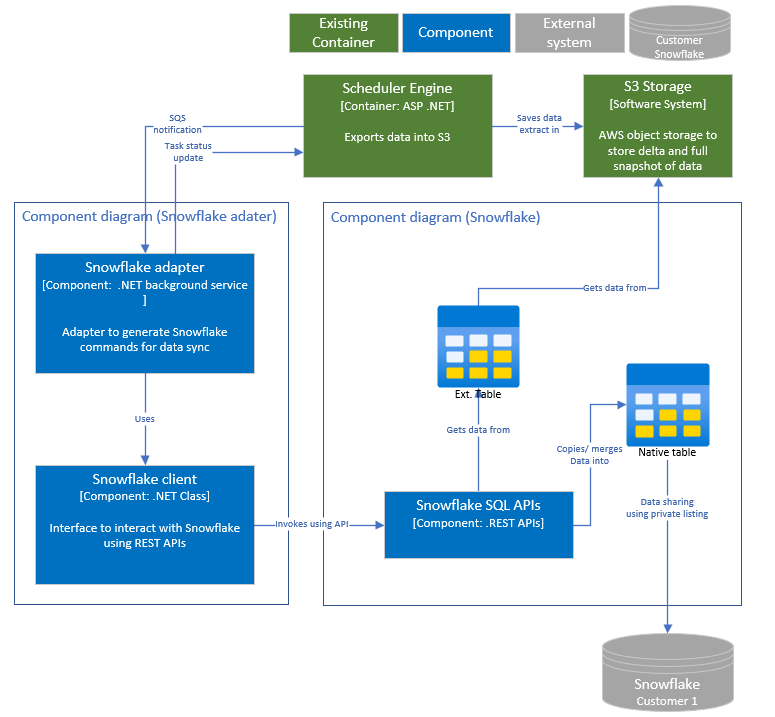
### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface Name | Description | Documentation / Metadata URL |
| iLEVEL UI | iLEVEL UI will be used to configure the data sync capability by the client admins. |  |
| Snowsight | Snowsight is Snowflake UI to manage Snowflake account and run the queries | [https://docs.Snowflake.com/en/user-guide/ui-snowsight](https://docs.snowflake.com/en/user-guide/ui-snowsight) |
| Snowflake data sharing- Private listing | Data will be shared with the customer using private listing functionality provided by Snowflake | [https://other-docs.Snowflake.com/en/collaboration/collaboration-listings-about](https://other-docs.snowflake.com/en/collaboration/collaboration-listings-about) |

## Component View

### Component Decomposition

#### Component Diagram



[Visio link.](https://spgl.sharepoint.com/:u:/r/sites/team_nam_mi_naf_m/_layouts/15/Doc.aspx?sourcedoc=%7B36BEA34E-65CB-4FCD-B021-A1F56A782BE3%7D&file=C4.vsdx&action=default&mobileredirect=true)

The Snowflake Adapter is responsible for ingesting data files from S3 Storage, prepared by the Data Export Engine, into S&P Snowflake instance.

Key components/functionalities of the Snowflake Adapter include:

* A background service that listens for new messages from the Scheduler Engine.
* Logic to determine the appropriate workflow (re-import or scheduled sync) based on the message parameters received.
* Fetches data from S3 using external tables, then performs merge or copy operations into Snowflake native tables via the Snowflake SQL API, enabling further data sharing.
* Notifies the status of data sync process to Scheduler engine.
* Generates end-to-end logs of the data sync process and populates a data-sync metadata table, which tracks details about synced tables for monitoring and troubleshooting.

In the event of a failure, the engineering team uses the logs and metadata to identify and resolve issues, then initiates a re-import or re-sync.

The main workflow for ingesting data into Snowflake contains the next steps:

1. Create a Snowflake native table and external table using Snowflake SQL API
2. Get the data in Snowflake external table from S3 using using Snowflake SQL API
3. Merge data into Snowflake native table using Snowflake SQL API

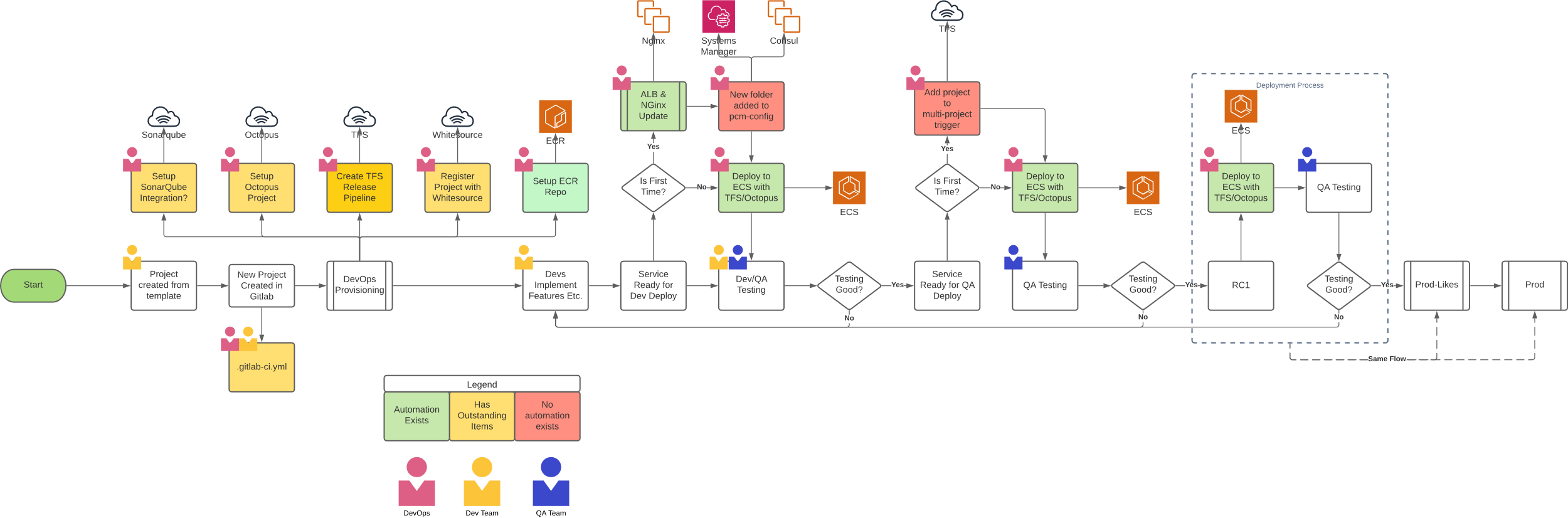
## Service Resilience

### Business Continuity / Disaster Recovery Governance

|  |  |  |
| --- | --- | --- |
| Number | Question | Status |
| 1 | Service Site Failover | Yes |
| 3 | Primary Data Center or Region | us-east-1 (N. Virginia) - US  eu-west-1 (Ireland) - EU |
| 4 | Secondary Data Center or Region | us-east-2(Ohio) - US  eu-central-1 (Frankfurt) - EU |
| 5 | Availability RPO[[3]](#footnote-4) | iLEVEL RPO: 30 mins  Snowflake Integration RPO: 30 mins |
| 6 | Availability RTO[[4]](#footnote-5) | iLEVEL RTO: 480 Mins  Snowflake Integration RTO: 24 hrs |
| 7 | Availability MTF[[5]](#footnote-6) | Not Available. |
| 8 | Availability Model | active/passive |
| 9 | Replication Method | Object – S3, Event-Snowflake |
| 10 | Central SCM[[6]](#footnote-7) | gitlab |

### Architectural Dependencies and Constraints

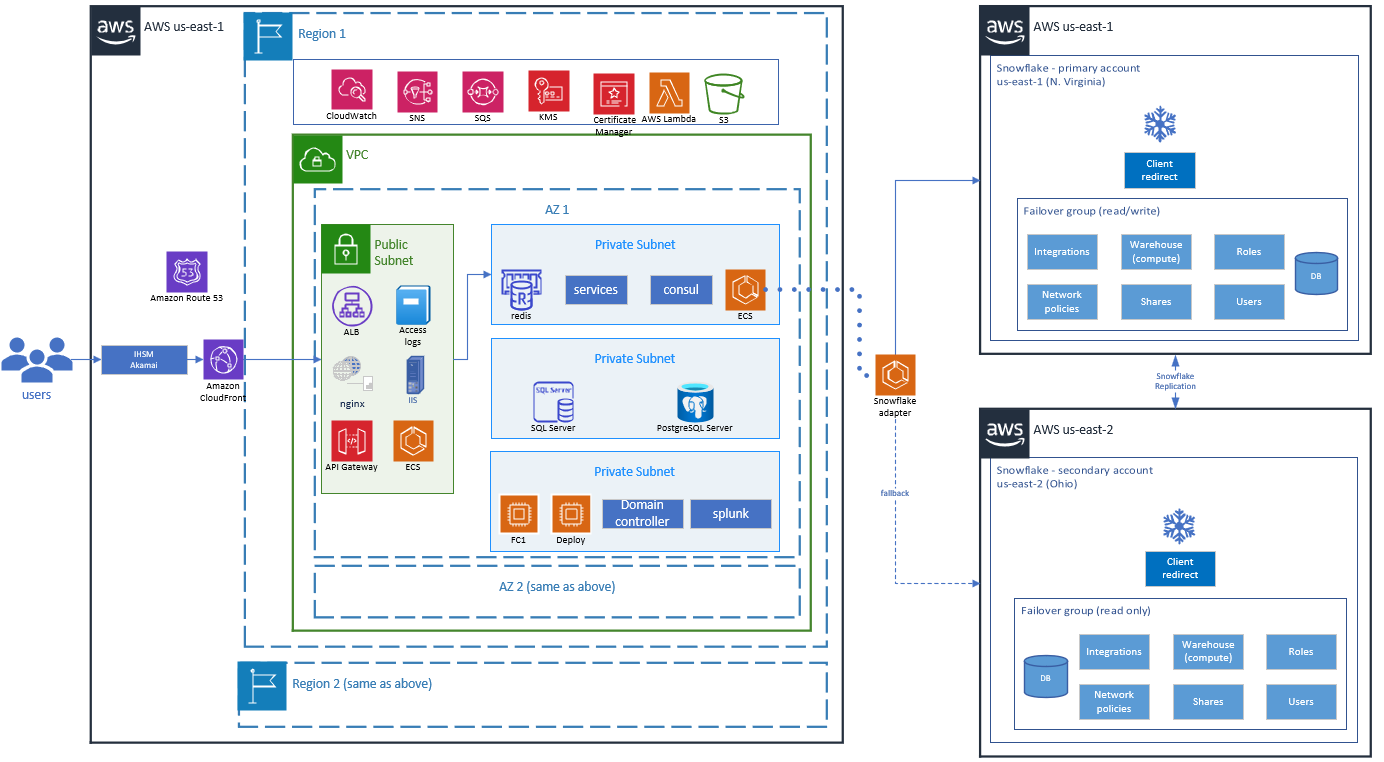
|  |  |  |  |
| --- | --- | --- | --- |
| The following table summarizes all gaps for this project. Where gaps exist, details are noted below.System or Project | Description | Existing Capability | Existing Capability Gap |
| Snowflake Integration | Currently, iLEVEL data is not available in Snowflake. This project aims to enable the availability of iLEVEL data on Snowflake. | No | Yes |

Deployment ModelAfter DevOps establish a deployment process: a dedicated TFS Deployment Pipeline, an Octopus project with adding the project to the multi-project trigger pipeline, the main flow will be:  
1. Dev team implements new changes  
2. Code review is done  
3. QA  
4. Getting a release approval from the Squad Lead  
5. Pushing a release using the multi-project release pipeline

## Physical Design

### Deployment Architecture

#### Infrastructure Diagram



[Visio link.](https://spgl.sharepoint.com/:u:/r/sites/team_nam_mi_naf_m/Shared%20Documents/ARB/iLEVEL/Snowflake%20Integration/C4.vsdx?d=w36bea34e65cb4fcdb021a1f56a782be3&csf=1&web=1&e=qqidh5)

* 1. Snowflake adapter will be hosted on ECS (as part of iLEVEL application) and it will be responsible for interaction between iLEVEL and Snowflake
  2. Two instances of Snowflake will be hosted: a primary instance and a secondary instance for failover purposes.
  3. Network policies will ensure that only allowed IP rages can communicate with Snowflake instance.
  4. iLEVEL will establish a connection with Snowflake using a client redirect URL, ensuring seamless access to the data.
  5. We will create a failover group that will automatically replicate the specified data and objects across the Snowflake accounts. The configured databases and objects will be replicated to the secondary account every 10 minutes to maintain data consistency.
  6. For Snowflake objects that cannot be replicated through the failover group, such as append-only streams, we will implement a custom utility or script to ensure they are updated appropriately.
  7. The primary region designated for the client redirect and the failover group will be us-east-1, which is the primary Snowflake account. This primary region will have read/write capabilities, while the secondary region will be set up as a read-only region to ensure data integrity during failover scenarios.

Failover:

1. Modify the client redirect for the secondary region to designate it as the primary.
2. Update the failover group for the secondary region to establish it as the primary.

With this configuration, the secondary region will gain read/write capabilities, while the primary region will transition to a read-only state (whenever its up).

Failback:

1. Pause data-sync job. Make sure there are no jobs in the running state.
2. Adjust the client redirect for the primary region to restore its status as primary.
3. Revise the failover group for the primary region to reinstate it as the primary.

In this setup, the primary region will regain read/write capabilities, and the secondary region will revert to a read-only status.

1. Start data-sync job.

#### Storage View

#### Storage, Backup, and Restoration

Data will primarily be stored in the Snowflake primary region (us-east-1) and will be synchronized with the secondary region every 10 minutes for backup purposes. In the event of an outage in the primary region, data will be served from the secondary region. If data corruption occurs, the Snowflake Time Travel feature can be used to restore the data to an earlier point, or a complete data resynchronization can be executed from iLEVEL.

Once the data is loaded into Snowflake, it will be periodically archived from S3. However, the data stored in Snowflake will not be subject to archival.

#### Storage Requirements

#### Database and Directory View

|  |  |  |  |
| --- | --- | --- | --- |
| Database / Directory | Type | Version | Configuration |
| Snowflake | SaaS Datawarehouse | 9.4 | replicated |

Snowflake is managed by the central platform team so the version will be changing in the future. Here is the [link](https://spglobal-mi.atlassian.net/wiki/spaces/DBDT/pages/291295854/Responsibility+Matrix+-+Central+Team+vs+Account+Owners+Teams) for responsibility matrix.

### Capacity Planning and Scalability

#### Key Resource Parameters

|  |  |  |
| --- | --- | --- |
| Component / Resource | KRP | Comments |
|  |  |  |

#### Scalability Model

1. Snowflake managed horizontally and vertically scalable compute.
2. Snowflake managed scalable storage with cross region replication.

### Software Component and Device Failover

#### Software Resiliency

Covered in the section [Infrastructure Diagram](#_Infrastructure_Diagram).

#### Operational Resiliency

Covered in the section – [service monitoring](#_Service_Monitoring).

### System Management View

#### Provisioning

NA

#### OS Provisioning

NA

#### Application Provisioning

Described in the Deployment Model section.

#### Application Patch Management

NA

### Instrumentation and Monitoring

#### Component / COTS Monitoring

NA

#### Application Monitoring

Describe how bespoke application software is instrumented and monitored.

|  |  |  |
| --- | --- | --- |
| Application Component | Tool to Use | Details |
|  |  |  |

#### Service Monitoring

|  |  |  |
| --- | --- | --- |
| Service Component | Tool to Use | Details |
| Snowflake Integration | PagerDuty | All components should have proper Splunk logs that will notify the engineering team in case of a failure.  In case of failure, engineering team/ support team will troubleshoot the issue and run the sync manually again. |
| Snowflake Integration | Snowflake | The clients will receive a metadata table as part of their dataset. This table will include 3 columns: a table’s name, a status of the last sync (failed/succeeded) and a timestamp of the last sync. This will allow clients to monitor the service’s status.  SQL API response will be verified after data-sysnc operation. If the status indicates failure, we mark the synchronization as unsuccessful and record the status in both the metadata files and the log files.  [Snowflake SQL API response](https://docs.snowflake.com/en/developer-guide/sql-api/reference) |
| Snowflake Integration | Activity Log | The clients will be able to review log records on the Activity Log tab on UI. |

#### Operational Change Management

#### Security Logging

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Event | Date and Time | Source and Destination IP | User ID |
|  |  |  |  |

#### Manual Processes

#### Software Packaging and Pre/Post Installation Checks

#### Runbooks

#### Event Responses

* Handling failover as described in [this section](#_Infrastructure_Diagram).
* Handling failback as described in [this section](#_Infrastructure_Diagram).

## Migration Strategy

Not applicable

### Assets to be Sunset or End-of-Life

None

## New Technologies

|  |  |  |  |
| --- | --- | --- | --- |
| Product | Vendor | Usage | Comments |
| Snowflake | Snowflake | SaaS | Cloud-based data warehouse |

## Material Costs

[*Cost sheet reference*](https://spgl.sharepoint.com/:x:/r/sites/team_nam_mi_naf_m/Shared%20Documents/ARB/iLEVEL/Snowflake%20Integration/Snowflake%20cost%20analysis%20-%20GL%20Comments%203.28%20updated.xlsx?d=we488f5b553b740f5b0ce5025b5ee8a53&csf=1&web=1&e=2Hbnxi)

*Note: Final cost includes 34% discount provided by Snowflake.*

**Table breakup by volume:**

|  |  |
| --- | --- |
| Small tables operations | 18 |
| Large tables operations | 7 |

( *Reference - sheet 1 small table operations (S,M) and large table operations (L,XL,XXL)*

**Client size reference-**

|  |  |
| --- | --- |
| Client Size | Periodic data points |
| XS | < 1M |
| Small | 1M |
| Medium | 1M-50M |
| Large | > 50M |

**Credit consumption- compute (per run):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Client Size | Operation | Small table | Large tables | Total |
| XS | Re-import | 0.02 | 0.035 | 0.605 |
| XS | Delta | 0.02 | 0.018 | 0.486 |
| Small | Re-import | 0.02 | 0.04 | 0.64 |
| Small | Delta | 0.02 | 0.018 | 0.486 |
| Medium | Re-import | 0.02 | 0.06 | 0.78 |
| Medium | Delta | 0.02 | 0.022 | 0.514 |
| Large | Re-import | 0.02 | 0.09 | 0.99 |
| Large | Delta | 0.02 | 0.038 | 0.626 |

*(Reference – cost results tab)*

**Compute cost (per run):**

|  |  |  |
| --- | --- | --- |
| Client Size | Region | Total compute ($) |
| XS | US | 1.44012 |
| XS | EU | 1.872156 |
| S | US | 1.48632 |
| S | EU | 1.932216 |
| M | US | 1.70808 |
| M | EU | 2.220504 |
| L | US | 2.13312 |
| L | EU | 2.773056 |

*(Reference- cost results tab)*

**Sync frequency and data volume (per month)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Client Size | XS | S | M | L |
| DB size (TB) | 0.001 | 0.01 | 1 | 1 |
| Sync freq | 180 | 180 | 180 | 180 |
|  |  |  |  |  |

**Final cost (PROD – annual):**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Client Size | Region | Compute Cost | Storage Cost | Replication Cost | DR Cost | **Total Cost$**  (34% S&P disc) |
| XS | US - Same region same cloud | 3111 | 0 | - | 0 | 3,111 |
| XS | US - Same cloud, diff region | 3111 | 0 | 0 | 0 | 3,111 |
| XS | US - Diff cloud, diff region | 3111 | 0 | 0 | 0 | 3,112 |
| XS | EU - Same region same cloud | 4044 | 0 | - | 0 | 4,044 |
| XS | EU - Same cloud, diff region | 4044 | 0 | 0 | 0 | 4,045 |
| XS | EU - diff cloud, diff region | 4044 | 0 | 0 | 0 | 4,045 |
| S | US - Same region same cloud | 3210 | 2 | - | 3 | 3,215 |
| S | US - Same cloud, diff region | 3210 | 2 | 3 | 3 | 3,218 |
| S | US - Diff cloud, diff region | 3210 | 2 | 4 | 3 | 3,220 |
| S | EU - Same region same cloud | 4174 | 2 | - | 3 | 4,178 |
| S | EU - Same cloud, diff region | 4174 | 2 | 3 | 3 | 4,181 |
| S | EU - diff cloud, diff region | 4174 | 2 | 4 | 3 | 4,183 |
| M | US - Same region same cloud | 3689 | 216 | - | 262 | 4,167 |
| M | US - Same cloud, diff region | 3689 | 216 | 262 | 262 | 4,429 |
| M | US - Diff cloud, diff region | 3689 | 216 | 440 | 262 | 4,607 |
| M | EU - Same region same cloud | 4796 | 216 | - | 262 | 5,274 |
| M | EU - Same cloud, diff region | 4796 | 216 | 262 | 262 | 5,536 |
| M | EU - diff cloud, diff region | 4796 | 216 | 440 | 262 | 5,714 |
| L | US - Same region same cloud | 4608 | 216 | - | 262 | 5,086 |
| L | US - Same cloud, diff region | 4608 | 216 | 262 | 262 | 5,348 |
| L | US - Diff cloud, diff region | 4608 | 216 | 440 | 262 | 5,526 |
| L | EU - Same region same cloud | 5990 | 216 | - | 262 | 6,468 |
| L | EU - Same cloud, diff region | 5990 | 216 | 262 | 262 | 6,730 |
| L | EU - Diff cloud, diff region | 5990 | 216 | 440 | 262 | 6,908 |

*(reference – cost results tab)*

**Cost components:**

* Compute cost- This is the cost to sync the data from S3 storage to S&P managed snowflake instance. S&P managed compute resources are not required when customer access the shared data.
* Storage cost: This is the cost to store the data in S&P managed snowflake instance.
* Replication cost: This cost is for data replication across cloud/region. Private listing automatically replicates data if customers are in different region and/or different cloud. This cost includes -egress cost, replication compute cost and replication storage cost.
* DR – This cost includes the cost associated with DR setup.

**Cost for dev environment:**

One XS-size environment in US region (sync frequency would be less) - ~3000$

\* in 6 months/a year to re-assess if cost is really the same as estimated, or if actual cost is higher than this

## Risks

### Technical Risks

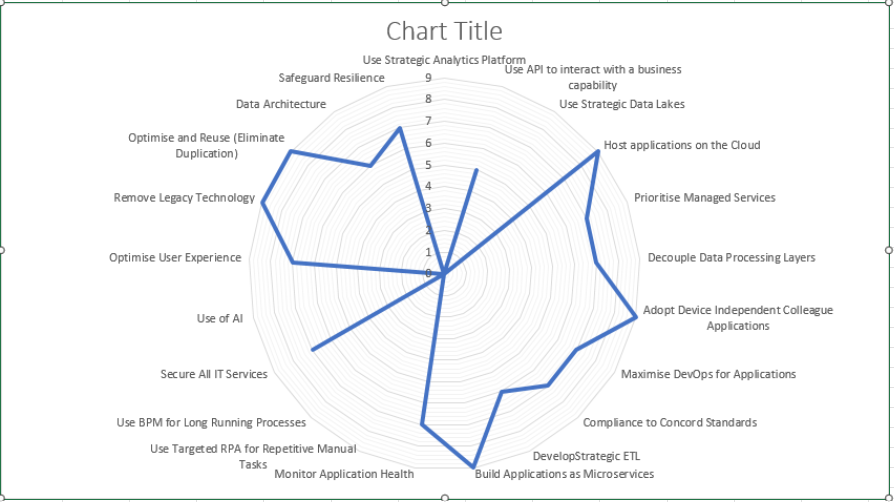
None

### Service Risks

None

## Architecture Scorecard

[Architecture-Score-Card-v0.6.xlsx](https://spgl.sharepoint.com/:x:/r/sites/team_nam_mi_naf_m/Shared%20Documents/ARB/iLEVEL/Snowflake%20Integration/Architecture-Score-Card-v0.6.xlsx?d=w89ecda30dc9345208bab091c0e027378&csf=1&web=1&e=UWLhhZ)



## Architecture Decision Record (ADR)

1. [Strategy for iLEVEL snowflake integration](https://spgl-my.sharepoint.com/:w:/r/personal/sandeep_khatri_spglobal_com/Documents/Desktop/ADR%20-iLEVEL%20-%20Snowflake%20integration.docx?d=w7232c318ec1140bcbec5c7354aa5d597&e=4%3a6b3f953dd048494b8000068c7f6a8deb&sharingv2=true&fromShare=true&at=9&xsdata=MDV8MDJ8YXR1bC5rdW1hci5zdXlhbEBzcGdsb2JhbC5jb218YzdkNzc2MDkwM2FiNGZhM2Y0Y2EwOGRkODE2ODIxMjl8OGYzZTM2ZWE4MDM5NGI0MDgxYTc3ZGMwNTk5ZTg2NDV8MXwwfDYzODgwOTAwNjQxNDQ5MzM3MXxVbmtub3dufFRXRnBiR1pzYjNkOGV5SkZiWEIwZVUxaGNHa2lPblJ5ZFdVc0lsWWlPaUl3TGpBdU1EQXdNQ0lzSWxBaU9pSlhhVzR6TWlJc0lrRk9Jam9pVFdGcGJDSXNJbGRVSWpveWZRPT18MHx8fA%3d%3d&sdata=UEtUVjJKN2JTaEtMSCtYMDlxODZSL3k1cHdEMXNZeHhmZU9QVkx3a0VHND0%3d)

**Context**:

Explore the available options to share data from iLEVEL to the customer's Snowflake instance.

**Decision**:

Our analysis concluded that direct integration of iLEVEL with Snowflake using private listing is the most suitable solution based on the requirements. This approach synchronizes iLEVEL data in the S&P managed Snowflake account before distributing it via private listing.

The following considerations have informed this decision:

* The requirement is to share customer data that resides in iLEVEL with the same customer. There is no data transformation involved in the process. The solution solely retrieves the JSON data from the S3, load data into S&P managed Snowflake instance and shares it with the customer Snowflake instance.
* A significant number of customers are either actively using or planning to implement Snowflake.

* Customers are located across multiple regions and utilize different Snowflake cloud platforms. They desire fully automated data replication in their Snowflake instances (no code solution), which would eliminate the need for manual intervention and decrease the maintenance and synchronization efforts associated with manual connectors. Snowflake's private listing offers the ability to share data seamlessly across clouds and regions for Snowflake-based customers, eliminating the need for customers to manage any code or ETL pipelines
* An existing component in iLEVEL extracts data from iLEVEL and stores it in S3 in JSON format. To maintain a simple, cost-effective design and reduce turnaround time, we opted for this straightforward approach (load data from S3 to Snowflake) instead of a complex design that would involve creating Iceberg tables externally from the S3-stored JSON files and subsequently loading them into S&P-managed Snowflake.

1. [iLEVEL Snowflake Integration – ETL Strategy](https://spgl.sharepoint.com/sites/team_nam_mi_naf_m/Lists/ADR/DispForm.aspx?ID=74&e=vmVZg8)

**Context**:

Analyse how the solution outlined in this ARB aligns with the broader ETL strategy of the Markets organization. Additionally, provide clarification on whether the intent is to leverage Snowflake for curated data creation or primarily as a distribution channel.

**Decision**:

Snowflake will serve as a data distribution channel, and no transformations will be performed within Snowflake to create curated datasets. This approach aligns with the overarching ETL strategy of the Markets. The following considerations have informed this decision:

* The use case primarily follows an Extract and Load pattern, where data is extracted from iLEVEL via API and subsequently loaded into Snowflake.
* Since the extracted data resides on S3 is already curated, there is no need to have an additional curated data zone. The data export (extract) process and the scheduler are pre-existing components, and the proposed solution effectively meets current requirements. No data transformation is involved.
* A new Snowflake Adaptor is built that will load the data into an S&P-managed Snowflake account, and then shared with customers via Snowflake’s Native Private Listing feature. This approach avoids the need for customers to make configuration changes within their own Snowflake environments.
* A significant number of customers are either actively using Snowflake or planning to adopt it in the near future. If there is a future need to distribute data beyond Snowflake customers, an additional adapter can be developed to support loading data into other data warehouses.

## NFR

[NFR-v1.0.xlsx](https://spgl.sharepoint.com/:x:/r/sites/team_nam_mi_naf_m/Shared%20Documents/ARB/iLEVEL/Snowflake%20Integration/NFR-v1.0.xlsx?d=w16399625673643ac9b124b8a362cb201&csf=1&web=1&e=zMX4Qv)

End of Design Stage Section.

# Deployment Stage

## Identify any changes in the project scope or design from previous stages.

## Business verification of delivery of requirements

## Instrumentation and operability

* Monitoring
* Tracing
* Add list of metrics ensuring KPIs and reliability / resiliency metrics are met.
* Explain how these are monitored.

## Identified and tested promotion strategy (canary, blue green)

## Identified and tested disaster recovery / rollback strategy

End of Deployment Stage Section.

# Go to Market Stage

## Ongoing maintenance and support requirements

Confirm if all ongoing maintenance and support requirements have been tested

* Playbooks
* Proactive and reactive alarms
* Alerts and escalations

## Proof of measurement of system performance and stability requirements

* Service Level Objectives (SLO)
* Service Level Agreements (SLA)
* Recovery Point Objectives (RPO)
* Recovery Time Objectives (RTO)

## Security controls

Are security controls implemented and verified?

* Penetration Testing
* Container Scanning
* Continuous security scanning and/or testing

## Business readiness

* Is IP compliance is verified?
* Are data management policies verified?
* Risks - identify any known risks for go to market.

1. Add any missing fields, entitles, relationships, or functions that must be added to the upstream data set. Also include any changes in performance or availability requirements (SLO, SLA, RPO, RTO) [↑](#footnote-ref-2)
2. Add any transformations or normalizations required to support the product [↑](#footnote-ref-3)
3. Recovery Point Objective – the amount of data loss (measured in time) allowed for recovery [↑](#footnote-ref-4)
4. Recovery Time Objective – the amount of time allowed to fully recover from downtime [↑](#footnote-ref-5)
5. Maximum Tolerable Downtime – the amount of downtime allowed before triggering a failover [↑](#footnote-ref-6)
6. Source Control Management – provide the name of the centrally governed SCM, if applicable. If code is stored elsewhere, please explain. [↑](#footnote-ref-7)