MULESOFT ARCHITECTURE:

Integration Patterns Part 2

PRACTIC LEARNING

Integration Patterns Introduction

1. Batch processing
2. System synchronization
3. Large file processing
4. Scatter gather

Batch Processing

Integration Pattern: Batch Processing

- Batch processing occurs when many records are processed in bulk
  - ETL/ELT is commonly associated with batch processing
- Batch processing should be used when
  - Entire data sets from one system need to be sent to another system
  - Scheduled data refreshes need to occur
  - Event-driven API-led connectivity cannot be used
- Recommended to explore event-driven API-led connectivity before using batch processing because batch processing is
  - Complicated
  - Error prone
  - Expensive
- However, batch processing is not going away, and most organizations have a need for processing large batches of data

Batch Processing Models

MuleSoft’s Batch Scope

- Process a large batch as quickly as possible leveraging the concurrency and parallel processing built into the batch scope
- Set up steps within a batch scope to process batch records in parallel
- Tuned by MuleSoft engineers to process large batches and abstract away some complexity
  - All in one solution that can process a large batch quickly
- Scale solution up for larger workloads

Breaking Up a Batch

- Break up a large batch into smaller chunks
  - Can use pagination to accomplish breaking a large batch into smaller pieces
- Decouple steps in the batch process – fetching data can be separated from processing data
- Can use the reliability pattern to guarantee delivery
- Scale solution up and/or out for faster processing

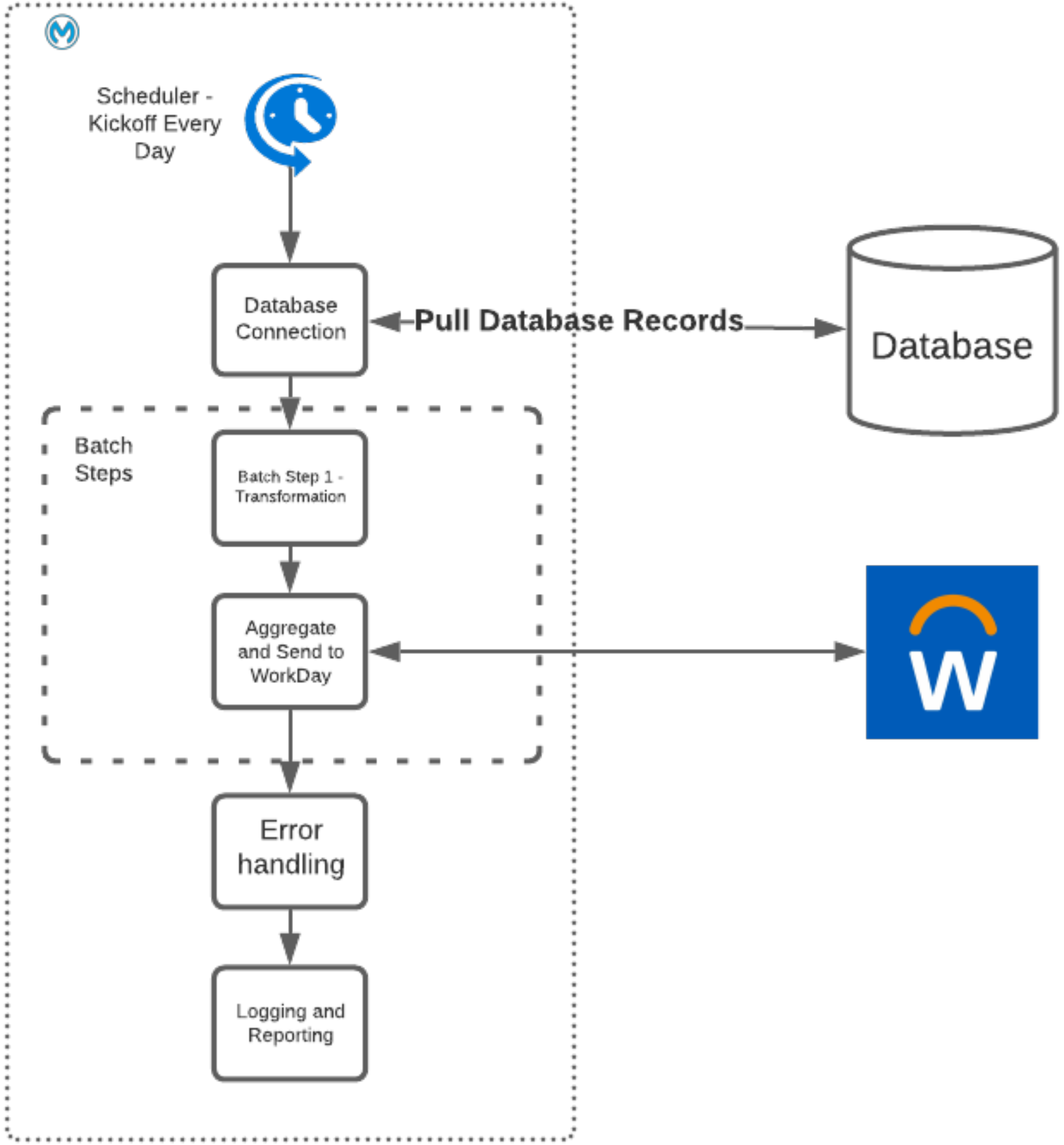
Batch Processing Models Comparison

	When to Use	Advantages	Disadvantages
MuleSoft's Batch Scope	1. Small to medium size batches 2. Large size batch and vCore/core utilization is not a concern 3. Aggregating the batch after processing is necessary	1. All-in-one solution tuned for batches 2. Process batch records in parallel as quickly as possible	1. High vCore/core usage is required for larger batches 2. Need to continue to scale batch processing application up as number of records increase
Breaking up a Batch	1. vCore/core usage must be minimized for medium to large size batches 2. Guaranteed delivery is a requirement 3. Fully customize logic, logging, and error handling	1. Minimize vCore/core usage 2. Scale to any batch size with the same vCore/core usage 3. Decouple batch processing steps 4. Ability to guarantee delivery 5. Ultimate customization of batch processing steps	1. More applications to manage 2. Increased complexity to implement pagination 3. May be slower processing

Batch Processing Scenario #1

An organization has employee data housed in a database that gets updated at least once per day. The employee data must be sent to WorkDay so that HR can manage the latest employee data. There are about 150,000 records in the database that get updated, amounting to about 100MB of total data. The architect is tasked with designing a process that can pull the employee data from the database once per day, transform and process it, and send it to WorkDay.

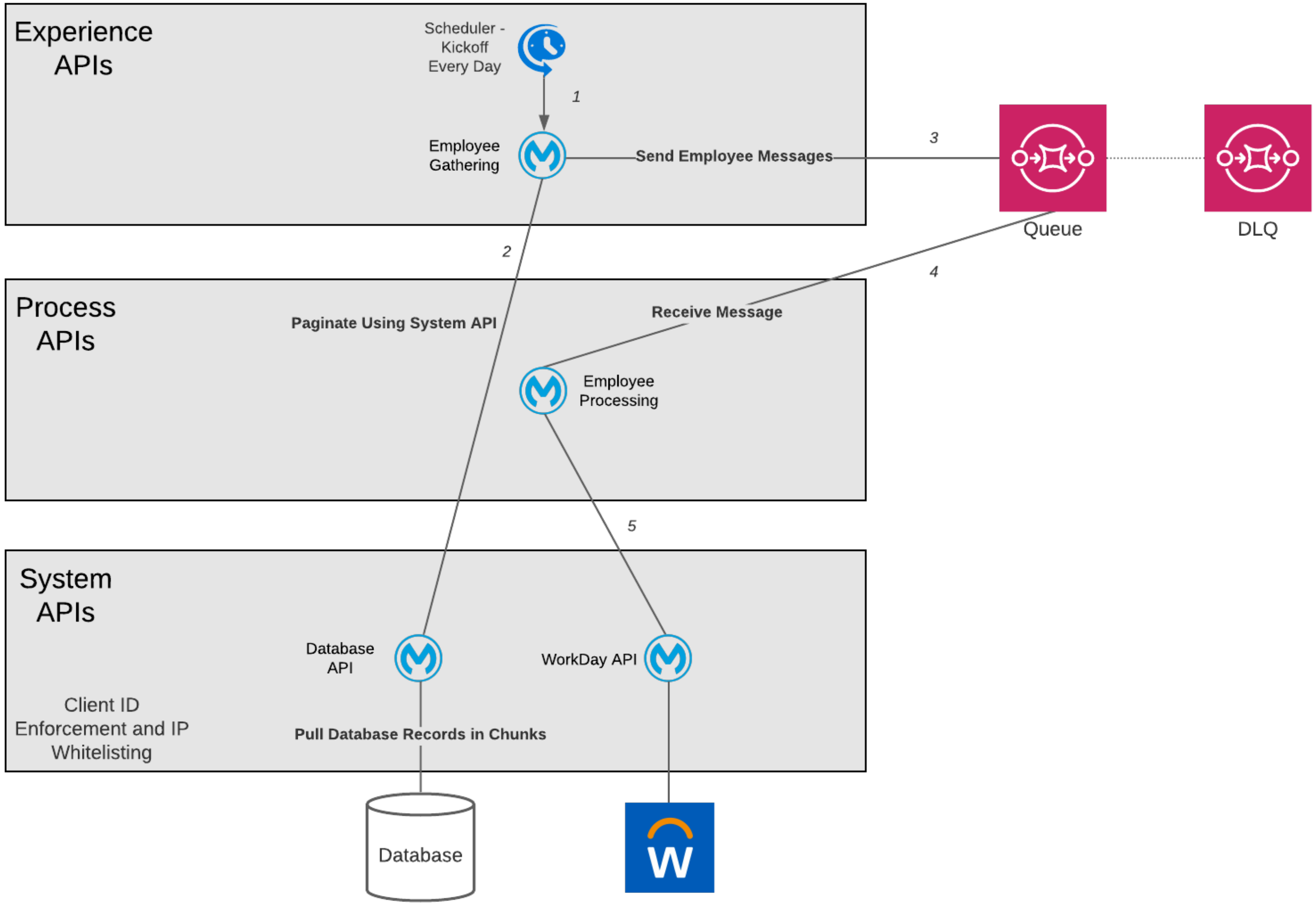
Batch Processing Scenario #1 Architecture



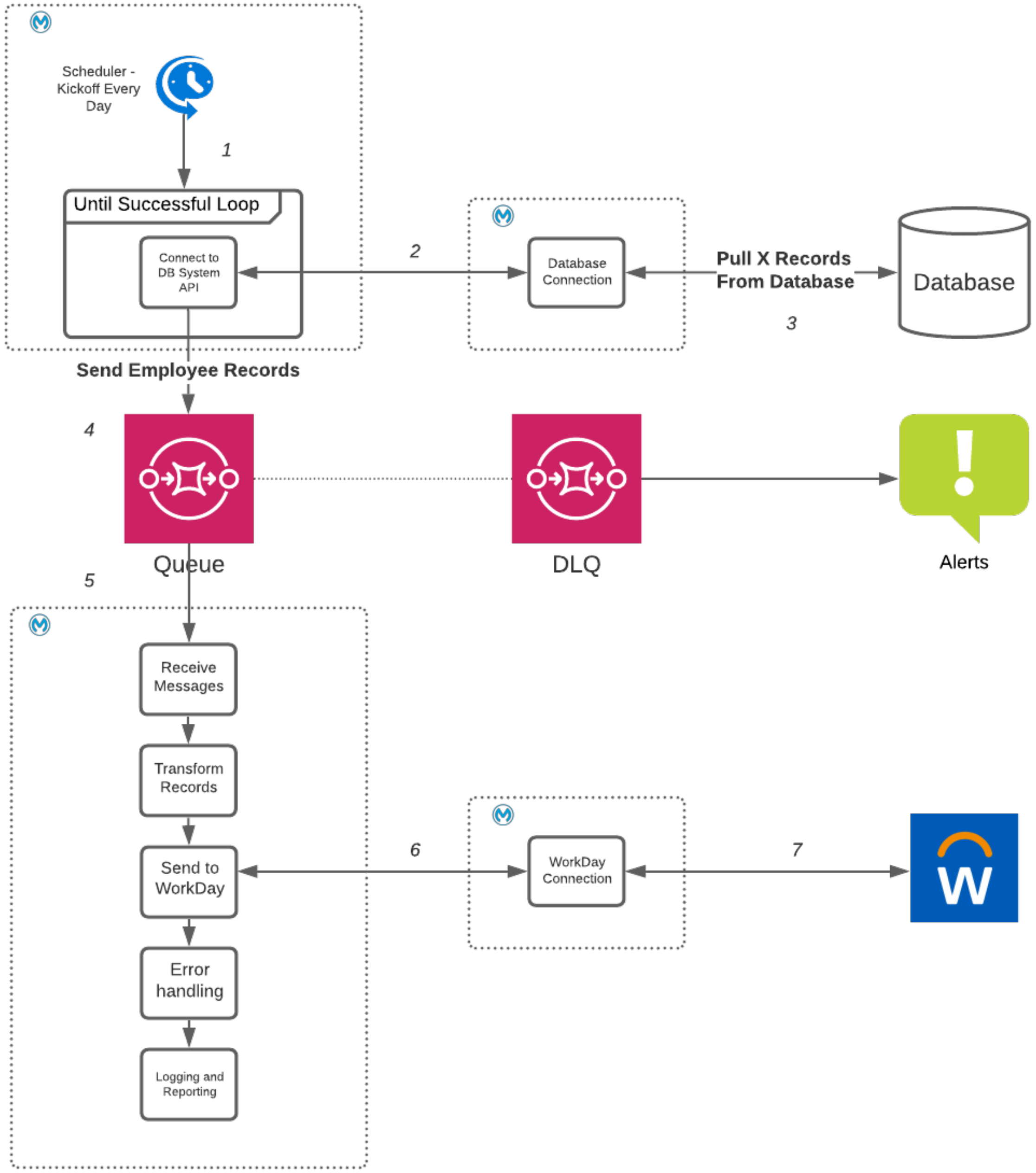
Batch Processing Scenario #2

An organization has employee data housed in a database that gets updated at least once per day. The employee data must be sent to WorkDay so that HR can manage the latest employee data. The company is growing rapidly and so is the dataset. There are already 3,000,000 records in the database, amounting to about 4GB of total data. The organization deploys to CloudHub and needs to minimize the vCore usage of this solution because they have a limited amount of vCores left in Production. The architect is tasked with designing a process that can pull the employee data from the database once per day, transform and process it, and send it to WorkDay.

Batch Processing Scenario #2 High Level Architecture



Batch Processing Scenario #2 Detailed Architecture



System Synchronization

Integration Pattern: System Synchronization (Sync)

- System sync allows for two or more systems to stay in sync
- System sync should be used when
  - Data from one system needs to be used in another system
  - Data in one system can be leveraged to gain business value in another system
- System sync data can flow one way or two-way between two systems
  - One way sync allows data to flow from System A to System B where System A is always the source
  - Two-way sync allows data to flow between System A and System B where System A or System B can initiate a change that flows to the other system

System Sync Models

- System sync can leverage event driven API-led connectivity or the batch integration pattern depending on the capability of the source systems
- Event driven system sync
  - Requires source systems to send requests to MuleSoft APIs when changes occur
  - Leverage API-led connectivity with the reliability pattern through HTTP 202 Accepted responses letting consumers know the request has been received
- Batch sync
  - Requires source systems to allow data to be pulled by MuleSoft
  - Leverage the batch integration pattern

System Sync Common Architectural Considerations

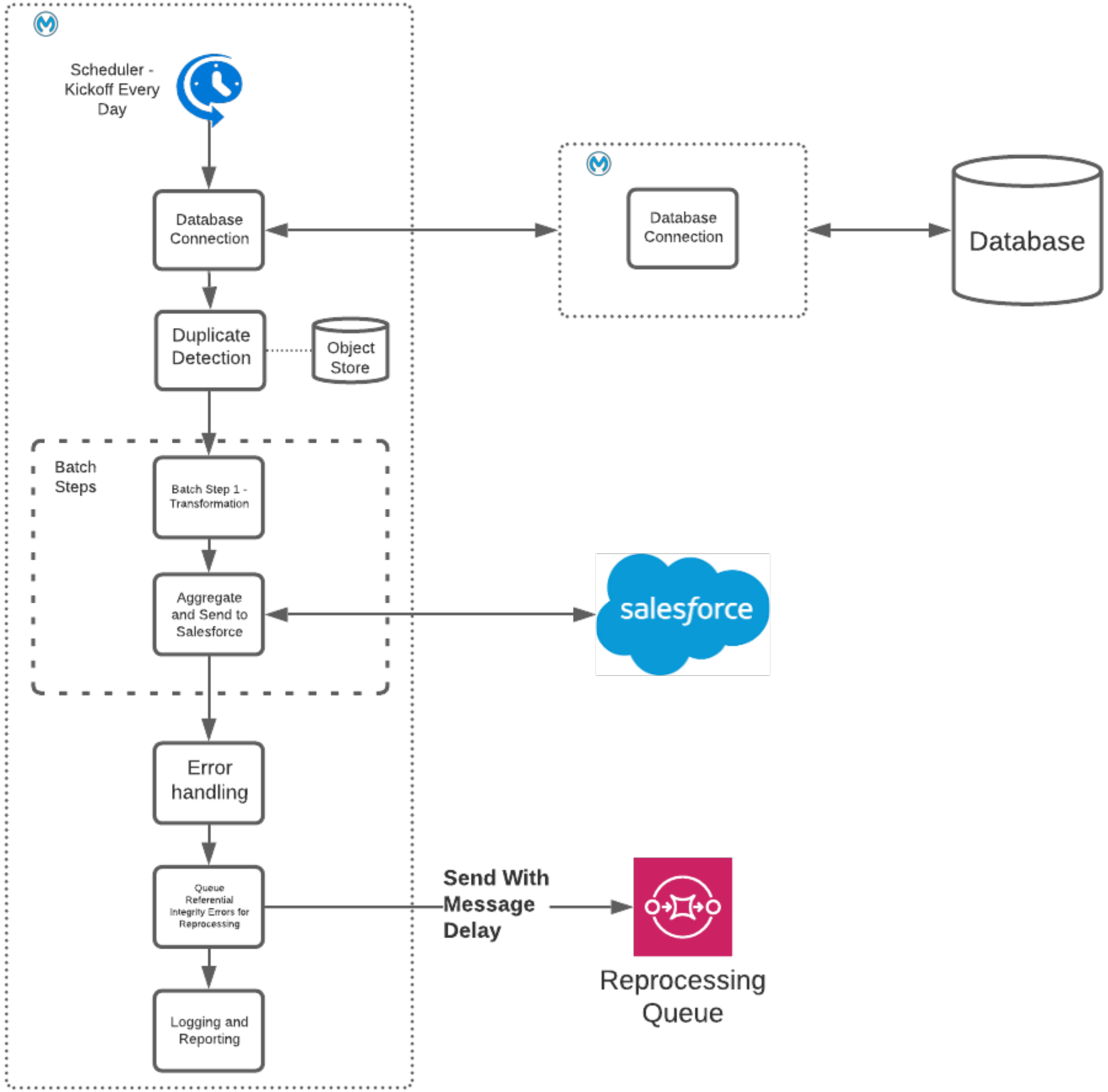
- Referential integrity is a common requirement for system sync
  - Events must stay in order
  - Two primary options to handle ordered delivery
    - Elegantly handle referential integrity errors as they are received from the target system
    - Use a staging location to wait for all related events to arrive before sending them to target system
- Two-way sync duplicate detection
  - Prevent an echo chamber between two systems by detecting duplicate events across two systems
- Complex business logic
  - Prepare to implement filtering and unique business logic in a processing layer

System Sync Scenario #1

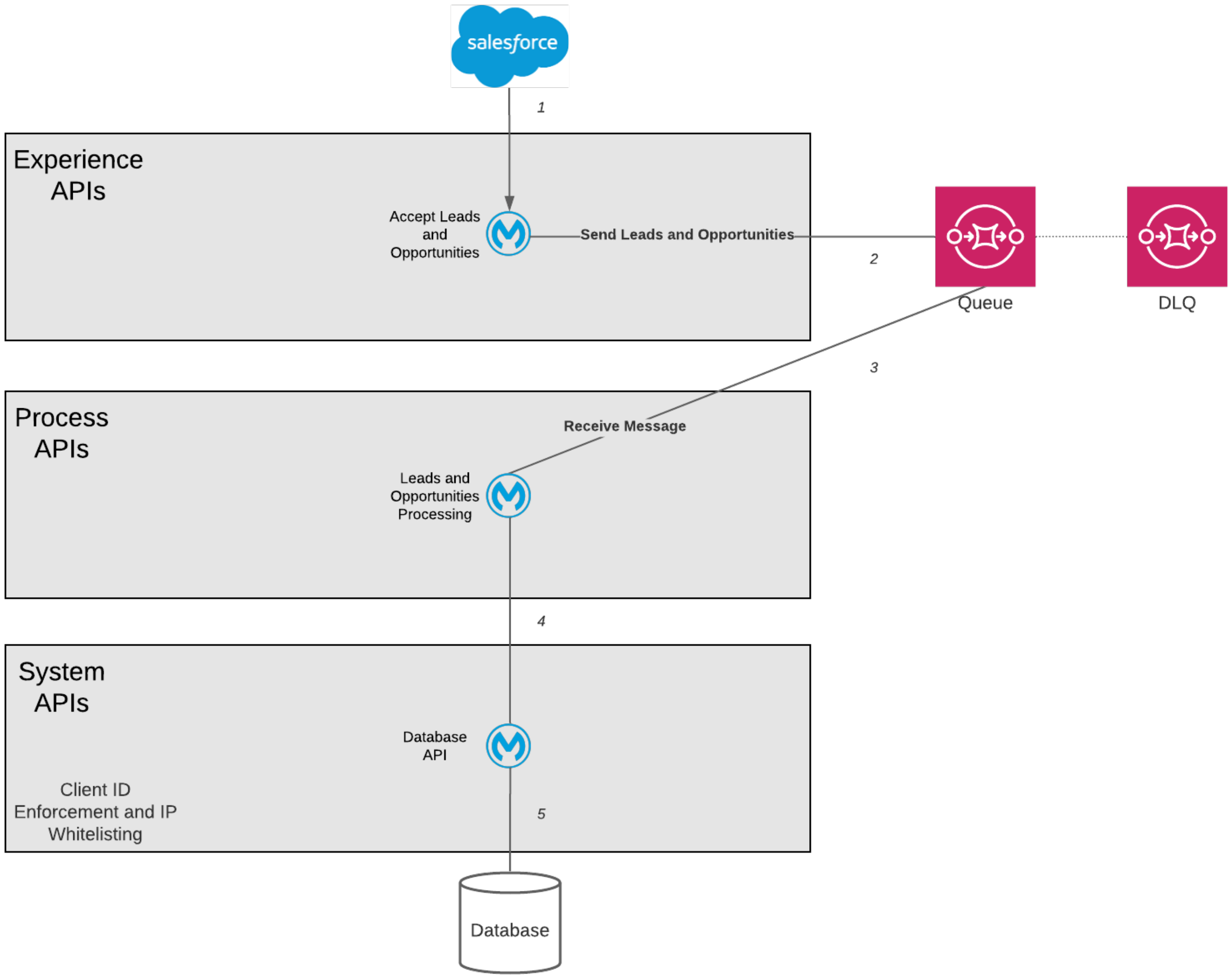
An organization needs to keep Salesforce data in sync with an Oracle database to allow their sales team to log leads and opportunities in a web application which uses the Oracle database for data storage. The back office needs to manage the same leads and opportunities from Salesforce. Due to this, if any change occurs in Salesforce, the Oracle database must be updated, and if the database gets updated, Salesforce must reflect the change. Additionally, opportunities cannot be created in Salesforce before leads.

The organization can use Platform Events in Salesforce to notify MuleSoft of any updates in Salesforce. However, the database must be pulled in batch at the end of every day to update Salesforce. There are approximately 1,000 leads and opportunities logged every day.

System Sync Scenario #1 Architecture – Database to Salesforce

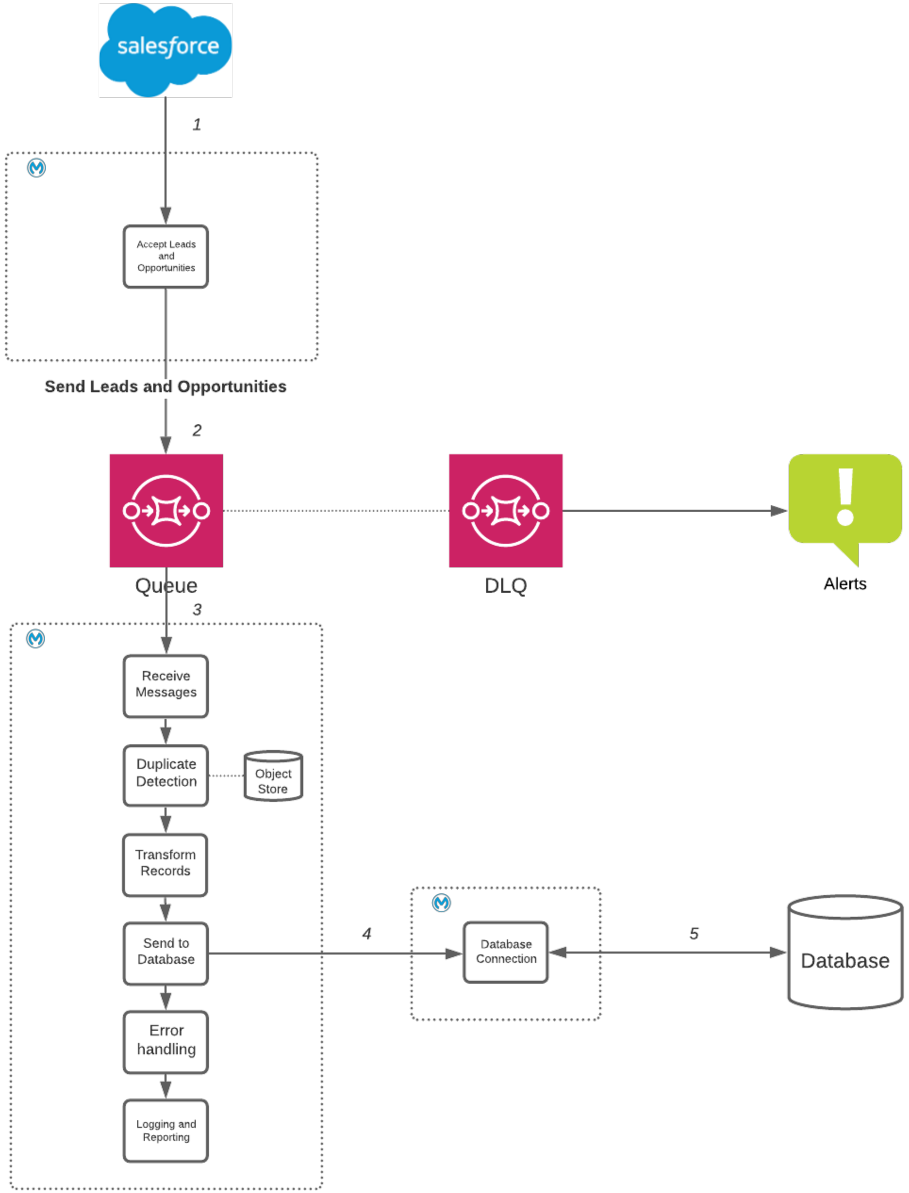


System Sync Scenario #1 Architecture – Salesforce to Database High Level



salesforce
1
Experience APIs
Accept Leads and Opportunities
Send Leads and Opportunities
2
Queue
DLQ
3
Process APIs
Receive Message
Leads and Opportunities Processing
4
System APIs
Database API
Client ID Enforcement and IP Whitelisting
5
Database

System Sync Scenario #1 Architecture – Salesforce to Database Detailed



Large File Processing

Integration Pattern: Large File Processing

- Large file processing allows for pickup and processing of large amounts of data contained in one file and sending the contents of the file to a downstream system
- Commonly leverages protocols like SFTP and FTP to pickup files from a server
- Architectural principles of large file processing
  - Always stream a large file to avoid saving it to memory
  - Split a very large file into chunks to save memory consumption and take advantage of parallel processing
- Leverage a similar architecture to the batch processing model

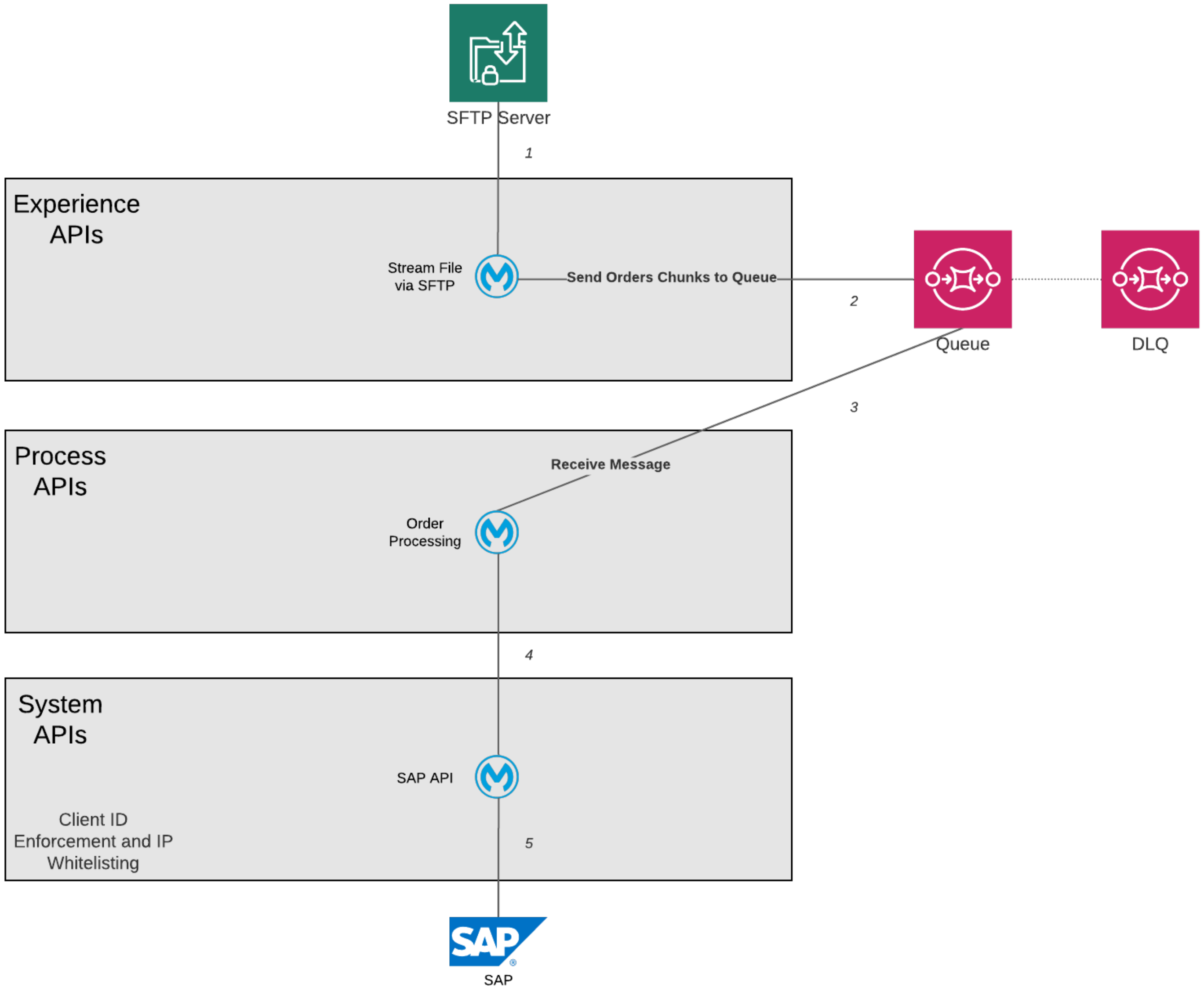
Large File Processing Common Architectural Considerations

- Always stream very large files
  - Loading file into memory can cause applications to run into OOM error
  - Saves vCore/core consumption
- Loading and processing time of large files
  - Leverage parallel processing to speed up processing times
- Restrictions of file transfer protocols
  - Gather requirements about the server and conduct a POC
- If splitting a file, it is sometimes necessary to reconstruct the file in order after processing (fan out and fan in)
  - Use a file staging location with numbered/ordered chunks of the file to reconstruct the file

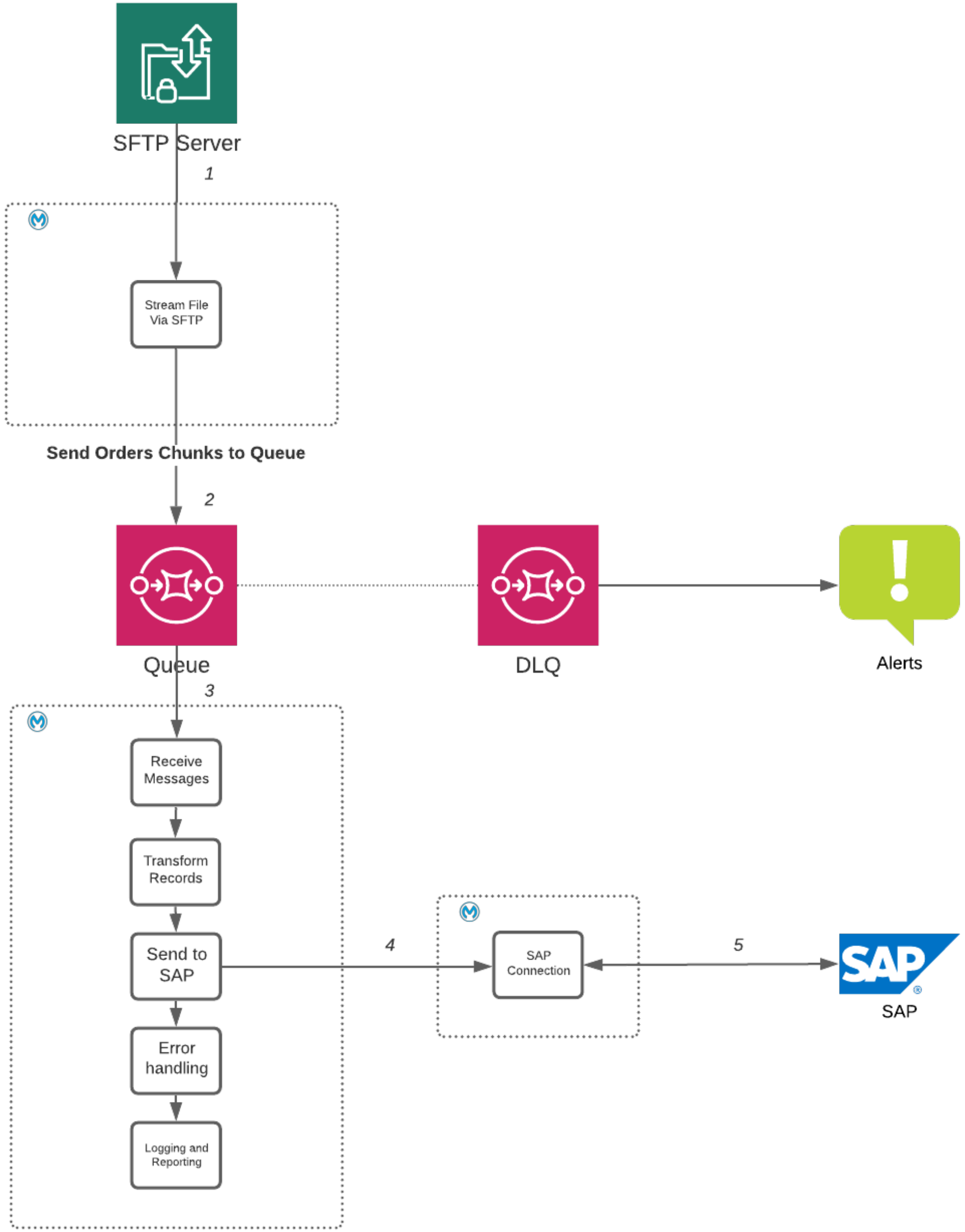
Large File Scenario #1

An organization picks up files from a server using the SFTP protocol so that they can process order information from their partners and send the orders to their order fulfillment system built in SAP. One partner sends a large XML file of all the orders in their system waiting to be processed. The XML file is about 3GB in size on average.

Large File Scenario #1 Architecture High Level



Large File Scenario #1 Architecture Detailed



Scatter Gather

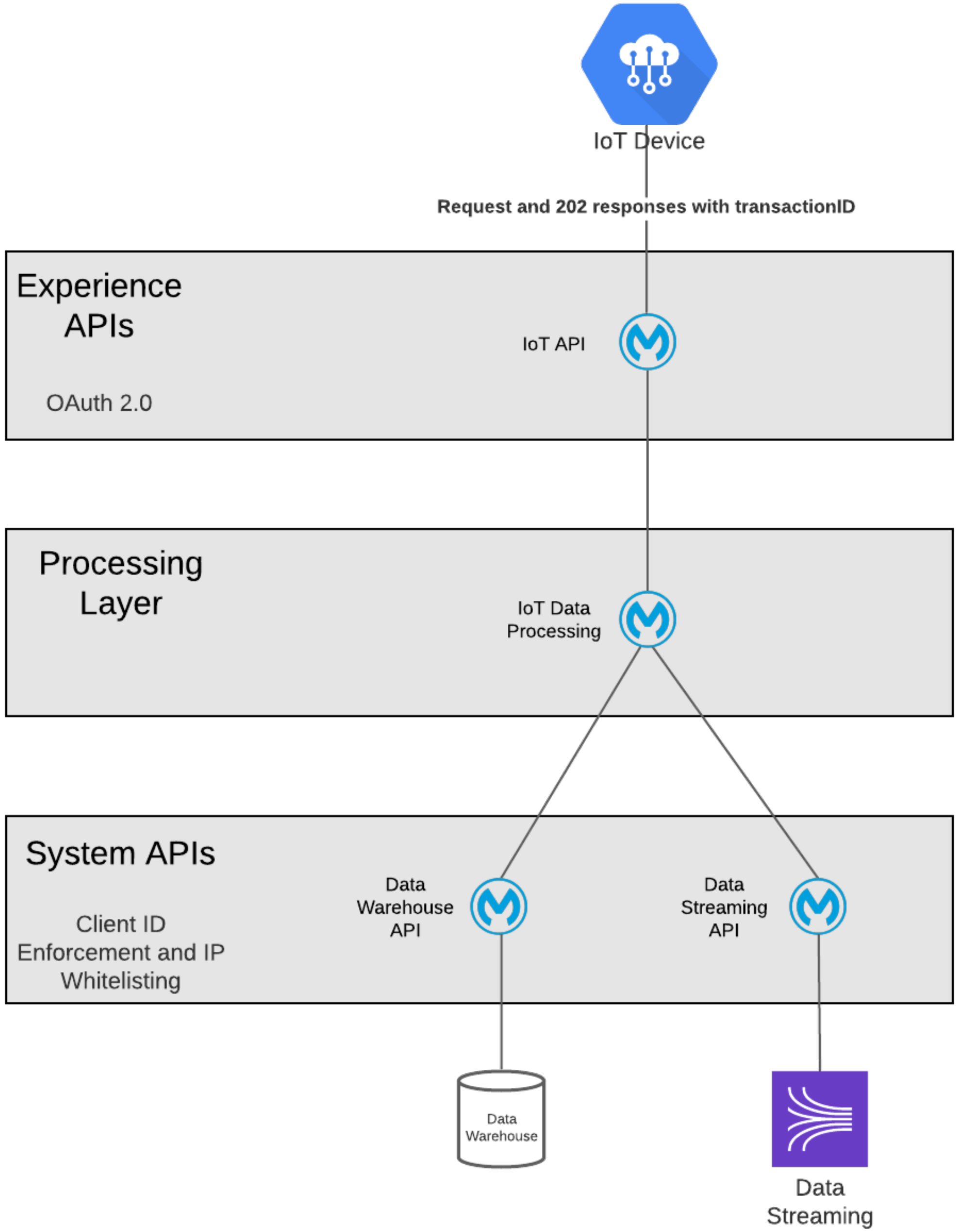
Integration Pattern: Scatter Gather

- Scatter gather allows for concurrent processing of the same record, commonly used to enrich data from multiple systems or send a single message to multiple targets
- Scatter gather should be used when
  - A single copy of a message can be sent to multiple target systems
  - Enriching data for a single message from multiple sources
  - Events can run in parallel without waiting on one another
- May be used in combination with the batch processing, system sync, or large file integration patterns

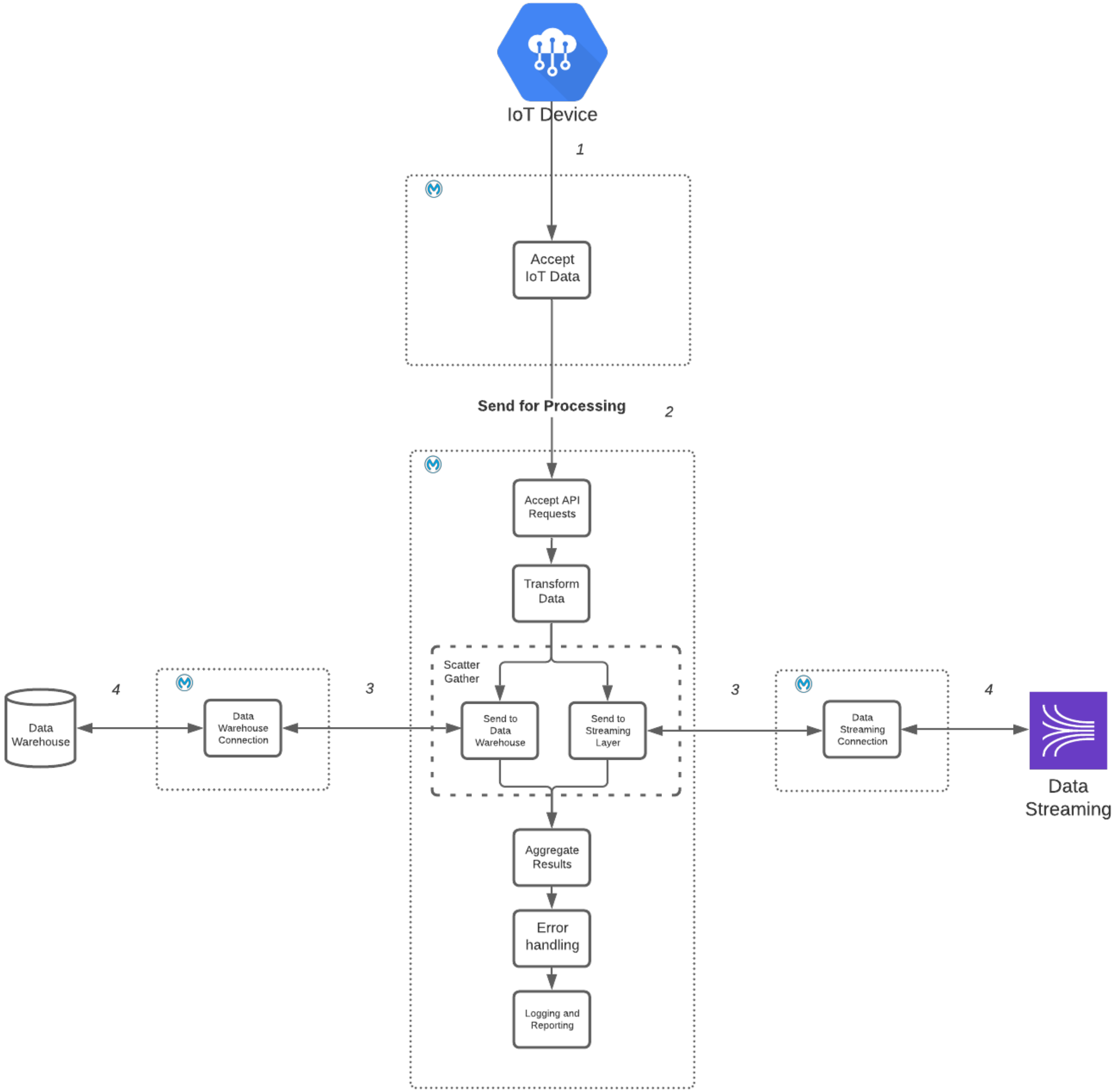
Scatter Gather Scenario #1

An organization gathers data from IoT sensors and needs to send the same copy of each event to a data warehouse and a data streaming layer for AI/ML processing to gain real-time actionable insight on the data. The architect must design an efficient MuleSoft application to fulfill the organization's needs.

Scatter Gather Scenario #1 Architecture High Level



Scatter Gather Scenario #1 Architecture Detailed



Integration Patterns Summary

- Batch processing
- One-way and two-way sync
- Large file processing
- Scatter gather

Additional Reading

- https://docs.mulesoft.com/mule-runtime/4.3/batch-processing-concept
- https://docs.mulesoft.com/mule-runtime/4.3/batch-filters-and-batch-aggregator
- https://docs.mulesoft.com/mule-runtime/3.9/batch-processing-reference
- https://meetups.mulesoft.com/events/details/mulesoft-denver-presents-api-led-data-synchronization-and-replication-whats-batch-got-to-do-with-it/
- https://www.mulesoft.com/exchange/org.mule.templates/template-sfdc2db-account-bidirectional-sync/
- https://www.bigcompass.com/insights/batch-scope-and-data-replication-with-mulesoft
- https://docs.mulesoft.com/mule-runtime/4.3/scatter-gather-concept
- https://docs.mulesoft.com/mule-runtime/3.9/scatter-gather