Withdrawal Request Processing Use Case with Azure (Medallion Architecture)

# Introduction

This use case focuses on building a secure, scalable, and auditable platform to process withdrawal requests submitted by users in a financial institution. The system captures requests, applies rules, and visualizes processing metrics using Azure cloud services following the Medallion architecture.

# Functional Requirement

- Customers initiate withdrawal requests via online banking or mobile app.  
- Each request contains account info, withdrawal amount, time, and reason.  
- Azure Data Factory ingests and cleans this data.  
- Logic Apps trigger alerts for high-value or policy-violating requests.  
- Azure Databricks aggregates and visualizes processing metrics.  
- Azure Data Lake stores raw and transformed records using Bronze, Silver, and Gold zones.

# Medallion Architecture Approach

We use the Medallion architecture:  
- 🔸 Bronze Layer: Raw withdrawal request logs from mobile/web.  
- 🥈 Silver Layer: Validated, enriched requests with fraud checks applied.  
- 🥇 Gold Layer: Aggregates like total withdrawals, average amount per branch, and rejected request ratios.

# Use Case Business Requirement

|  |  |
| --- | --- |
| Component | Requirement Description |
| Azure Blob Storage | Stores raw withdrawal request logs. |
| Azure Data Factory | ETL pipeline to validate and aggregate withdrawal records. |
| Azure Logic Apps | Triggers alerts on high-value or duplicate withdrawal attempts. |
| Azure Databricks | Transforms, aggregates and visualizes transaction metrics. |
| Azure Data Lake | Structured storage across Bronze, Silver, and Gold layers. |
| Databricks Visualization | Dashboard for total withdrawals, flagged requests, branch-wise summary. |

# Test Cases

- Test Case 1: Submit a ₹100,000+ withdrawal and verify alert trigger.

- Test Case 2: Submit duplicate requests within 5 minutes and check rejection.

- Test Case 3: Verify rejected request count by branch in Gold layer.

- Test Case 4: Validate visualization of daily withdrawal volume trend.

# Data Lake Folder Structure

/bronze/ → Raw withdrawal logs  
 └─ withdrawals\_2025-07-15.json  
  
/silver/ → Validated, enriched withdrawal requests  
  
/gold/ → KPIs: branch-wise totals, volume trends, high-risk alerts

# Conclusion

This use case gives students exposure to building a compliance-focused pipeline that automates withdrawal validations, detects anomalies, and provides reporting insights with Azure-native services.