Core Business Use Cases

Nathan Lunceford

2025-03-05

Table of contents

1 Extract Data from ERP Systems via API

The system must be able to connect to various ERP systems (e.g., Epicor, Sage) via their APIs, authenticate, and extract required data types. This includes:

- Authenticating with proper credentials (API keys, tokens, client certificates)
- Building appropriate requests for each ERP system
- Handling pagination and batching for large datasets
- Managing rate limits and connection timeouts
- Capturing and processing response data
- Implementing resilience patterns for API communication

2 Extract Data from ERP Databases

For ERP systems that allow or require direct database access, the system must:

- Connect securely to various database types
- Execute appropriate queries for different data types
- Handle connection pooling and resource management
- Apply appropriate filtering for incremental extraction
- Process result sets efficiently
- Release database resources properly

3 Transform Extracted Data

All extracted data must undergo initial transformation to:

- Standardize column names according to our global data dictionary
- Validate data against defined schemas and quality rules
- Convert to Parquet format for efficient storage and processing
- Apply optional compression for reduced storage needs
- Identify and mask sensitive information when required
- Preserve data lineage information

4 Secure Credential Management

The system must securely manage credentials for various ERP systems:

- Retrieve credentials from HashiCorp Vault using least privilege access
- Support different credential types (API keys, database credentials, certificates)
- Handle credential rotation and expiration
- Ensure credentials are never logged or persisted outside secure storage
- Apply proper authentication mechanisms for each ERP type

5 Track Data Lineage

For audit and compliance purposes, the system must:

- Track the source, extraction time, and parameters of each data extraction
- Record transformation details and any quality issues
- Link extracted data to its final storage location

- Provide metrics on extraction volume and timing
- Enable traceability for data governance

6 Monitor and Report System Health

The system must provide comprehensive monitoring:

- Report on successful and failed extractions
- Expose metrics for performance and throughput
- Alert on critical failures or data quality issues
- Log detailed information for troubleshooting
- Provide health check endpoints for operational status