Software Design Document (SDD) – Flow 2: Outbound Claims Processing

# 1. General Information

## 1.1 Executive Summary

This document provides a detailed technical design for Flow 2 of the outbound claims processing system. Flow 2 is responsible for processing pending claims generated from the Facets system, filtering them based on specific business rules, transforming them into a standardized format, and securely delivering them into Snowflake for downstream consumption by the Prime system. The architecture is event-driven, leverages Azure cloud services, and ensures scalability, traceability, and operational stability through robust monitoring and logging.

## 1.2 Scope

### 1.2.1 In-Scope

- Outbound claims flow from Azure Event Grid to Snowflake.  
- Event-driven processing using Azure Functions triggered by Storage Queue messages.  
- Claim filtering logic for identifying Prime-pended claims.  
- Data transformation for Snowflake ingestion.  
- Logging, monitoring, and error handling with Application Insights and Snowflake PROCESS\_LOG.  
- Security through Managed Identity, Azure Key Vault, and API Gateway (Apigee).  
- Snowflake schema design for staging and final claims tables.

### 1.2.2 Out-of-Scope

- Inbound flow from Prime back to Facets.  
- Initial publishing of claims from Facets to Kafka and Blob Storage.  
- Downstream transformation by Matillion or other ETL tools beyond Snowflake staging.

## 1.3 Document Usage

This document serves as a reference for all stakeholders including solution architects, developers, QA engineers, and project managers. It provides a single source of truth for Flow 2’s technical design. Any changes to this design must follow SDLC governance and be approved by architecture review boards.

## 1.4 Diagrams

For this document, detailed text explanations are provided instead of diagrams. The architecture can be summarized as: Event Grid triggers → Storage Queue → Azure Function → Blob Storage → Filtering & Transformation → Snowflake staging table → Downstream systems.

## 1.5 Specification Sheet

Specification details of input/output JSON, queue message structure, Snowflake schema, and logging are covered in subsequent sections.

# 2. Technical Design

## 2.1 Input for Claim Details

The input for this flow is a JSON file stored in Azure Blob Storage within the claims-raw container. This file contains one or more claim records, each represented as a JSON object. The Azure Event Grid publishes a message when a new file is created, which is then delivered to an Azure Storage Queue.

### 2.1.1 Input JSON Structure

The expected structure for each claim object is as follows:

{  
 "claim\_id": "string",  
 "member\_id": "string",  
 "policy\_id": "string",  
 "claim\_type": "string",  
 "status": "string",  
 "pended\_reason": "string",  
 "service\_start\_date": "YYYY-MM-DD",  
 "service\_end\_date": "YYYY-MM-DD",  
 "provider\_id": "string",  
 "diagnosis\_codes": ["string"],  
 "procedure\_codes": ["string"],  
 "claim\_lines": [  
 {  
 "line\_id": "string",  
 "service\_code": "string",  
 "service\_date": "YYYY-MM-DD",  
 "billed\_amount": "number",  
 "allowed\_amount": "number"  
 }  
 ]  
}

### 2.1.2 Input Request Details

The Azure Function is triggered by a queue message published by Event Grid. The queue message contains metadata of the blob file. Example:

{  
 "topic": "/subscriptions/{subscription-id}/resourceGroups/{resource-group}/providers/Microsoft.Storage/storageAccounts/{storage-account}",  
 "subject": "/blobServices/default/containers/claims-raw/blobs/{blob-name}",  
 "eventType": "Microsoft.Storage.BlobCreated",  
 "data": {  
 "api": "PutBlob",  
 "requestId": "...",  
 "url": "https://{storage-account}.blob.core.windows.net/claims-raw/{blob-name}",  
 "contentType": "application/json"  
 }  
}

### 2.1.3 Input - Request Validation

The function validates the incoming message by:  
- Ensuring the blob URL exists.  
- Ensuring file extension is .json.  
- Validating the schema of the claims JSON.  
- Invalid claims are moved to a quarantine container and logged.

## 2.2 Output for Claim Details

The transformed output is sent to Snowflake via the member API. Each claim is mapped into a normalized JSON payload optimized for ingestion.

### 2.2.1 Output JSON Structure

{  
 "claims": [  
 {  
 "claim\_guid": "string",  
 "claim\_id": "string",  
 "pended\_reason": "string",  
 "pended\_timestamp": "timestamp",  
 "ingestion\_timestamp": "timestamp",  
 "diagnosis\_codes": ["string"],  
 "procedure\_codes": ["string"]  
 }  
 ],  
 "batch\_id": "string",  
 "source\_file": "string"  
}

### 2.2.2 Output Response Header

The member API responds with HTTP 200 for success, HTTP 400/500 for failures. Error details are logged to PROCESS\_LOG in Snowflake.

### 2.2.3 Output Response Medical Line

Not applicable for this flow, as claim line details are embedded within the claim object.

# 3. Business Rules

## 3.1 Fetching Facets Claim Details Logic

The Azure Function applies the following rules:  
- Filter claims where status = 'Pending' and pended\_reason contains 'Prime'.  
- Generate a UUID (claim\_guid) for each claim.  
- Add ingestion\_timestamp at runtime.  
- Retain original pended\_timestamp if present.  
- Normalize arrays like diagnosis\_codes and procedure\_codes for Snowflake compatibility.

## 3.2 Fetching SSDS Claim Details Logic

Not applicable for Flow 2. Outbound flow only processes Facets to Prime claims.

# 4. References

- Azure Functions Documentation  
- Azure Storage Queue Documentation  
- Azure Event Grid Documentation  
- Snowflake Documentation  
- Apigee API Gateway Documentation  
- ServiceNow Integration Documentation

# 5. Amendment History

This section tracks document versioning.  
- Version 1.0 – Initial design document for Flow 2  
- Version 1.1 – Added Snowflake schema details and enhanced error handling

# Appendix A: Snowflake Schema

## A.1 Staging Table Schema: STG\_CLAIMS\_OUTBOUND

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Nullable | Description |
| CLAIM\_GUID | VARCHAR(36) | NOT NULL | Unique identifier for the claim, generated by Azure Function. |
| BATCH\_ID | VARCHAR(64) | NOT NULL | Unique identifier for each batch run. |
| SOURCE\_FILE\_NAME | VARCHAR(256) | NOT NULL | Original file name from Blob Storage. |
| PAYLOAD | VARIANT | NOT NULL | Full claim object JSON. |
| INGESTION\_TIMESTAMP | TIMESTAMP\_LTZ(9) | NOT NULL | Time of ingestion. |

## A.2 Logging Table Schema: PROCESS\_LOG

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Nullable | Description |
| LOG\_ID | VARCHAR(36) | NOT NULL | Unique log entry identifier. |
| PROCESS\_ID | VARCHAR(36) | NOT NULL | Batch ID from STG\_CLAIMS\_OUTBOUND. |
| TIMESTAMP | TIMESTAMP\_LTZ(9) | NOT NULL | Log timestamp. |
| SEVERITY | VARCHAR(10) | NOT NULL | Log level (INFO/WARN/ERROR). |
| MESSAGE | VARCHAR(1024) | NOT NULL | Event or error message. |
| ERROR\_CODE | VARCHAR(10) | NULL | Custom error code. |
| ERROR\_DETAILS | VARIANT | NULL | Full error object. |