Khaleeji Bank

SAS Fraud Decisioning

Solution Design Document  
V1.1

A black and grey logo

AI-generated content may be incorrect.

[**1.** **Solution Overview** 3](#_Toc196677505)

[1.1. Business Context 3](#_Toc196677506)

[1.2. SAS Product Used: SFD 3](#_Toc196677507)

[1.3. Solution Overview 4](#_Toc196677508)

[**2.** **Technical Architecture** 5](#_Toc196677509)

[2.1. System Components 5](#_Toc196677510)

[2.1.1. Advanced Analytics 5](#_Toc196677511)

[2.1.2. SAS Detection Architecture 5](#_Toc196677512)

[2.1.3. SAS Alert Triage 5](#_Toc196677513)

[2.1.4. SAS Visual Investigator 5](#_Toc196677514)

[2.2. E2E Data Flow 6](#_Toc196677515)

[2.3. Message Schema 7](#_Toc196677516)

[2.3.1. Variable Sets 7](#_Toc196677517)

[2.3.2. Business Mapping 10](#_Toc196677518)

[2.4. Transaction types 11](#_Toc196677519)

[2.4.1. Configuring discriminator variables 11](#_Toc196677520)

[2.4.2. Configuring Transaction Types 11](#_Toc196677521)

[2.5. End points 13](#_Toc196677522)

[2.5.1. Rest EndPoints 13](#_Toc196677523)

[2.5.2. Kafka topics 13](#_Toc196677524)

[2.6. Groups & Users 13](#_Toc196677525)

[2.6.1. Users 13](#_Toc196677526)

[Roles and Capabilities 13](#_Toc196677527)

[Users 16](#_Toc196677528)

[**3.** **Alert Triage Database Schema** 17](#_Toc196677529)

# **Solution Overview**

# Business Context

SAS Fraud Decisioning Solution has been designed to be an enterprise platform. One single solution can be utilized for “all” transactional monitoring – e.g., customer, account, ACH/Wire, deposits, checks, cards, digital online/internet, phone/mobile, etc., It will allow Khaleeji Bank to take advantage of AI driven analytics and deploy them in real time. Analytics and advanced team learning are the heart of our solution and key to meeting Khaleeji Bank needs. To ensure rapid delivery and advanced enablement for Khaleeji Bank staff, our proposal includes a SAS-developed, custom, consortium aware model to address real time rails and future payment fraud prevention.

We believe that bringing our proven modeling capabilities, turned for real-time payments, combined with an open model API that allows Khaleeji Bank staff to deploy user-written open-source algorithms such as R, Python, will allow Khaleeji Bank to balance the best of independent model development with SAS proven world class experience.

# SAS Product Used: SFD

SAS Fraud Decisioning, or SFD for short, is the cloud native SAS Viya platform solution for fraud management, allowing a customer to use automated logic to rapidly detect, assess, and then act upon business transactions to prevent fraud, allowing users to assess and investigate specific actions when necessary.

# Solution Overview

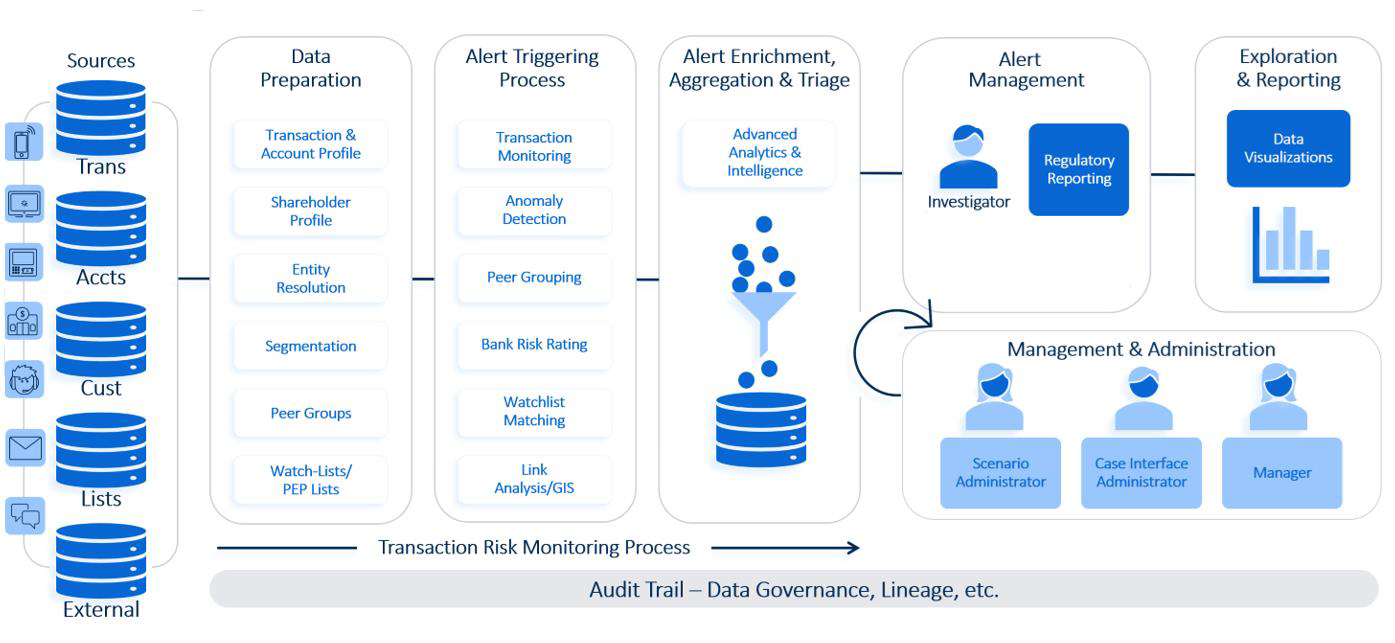


Figure SFD Solution Overview

The diagram provides a structured process flow related to a SAS Financial Detection (SFD) solution:

* Sources: This represents the various data sources that feed into the system, such as transaction data, customer data, and external data sources.
* Data Preparation Stage: This stage involves cleaning, transforming, and organizing the raw data to make it suitable for analysis. This is crucial for ensuring data quality and consistency.
* Alert Triggering Process: This is the mechanism by which the system generates alerts based on predefined rules or Identifying deviations from normal behavior patterns using AI models.
* Alert Enrichment, Aggregation & Triage Stage: This is the stage of enhancing the alerts with additional context, aggregating related alerts, and prioritizing them to display them in the Alert triage for investigation.
* Alert Management: Handling and tracking the lifecycle of alerts from the alert creation to disposition through a responsive user interface called “SAS Alert Triage”.
* Data Visualizations: Exploring data and generating reports to provide insights and support decision-making.
* Management & Administration: Overseeing the administrative aspects of the system, including user management and system configuration.

# **Technical Architecture**

# System Components

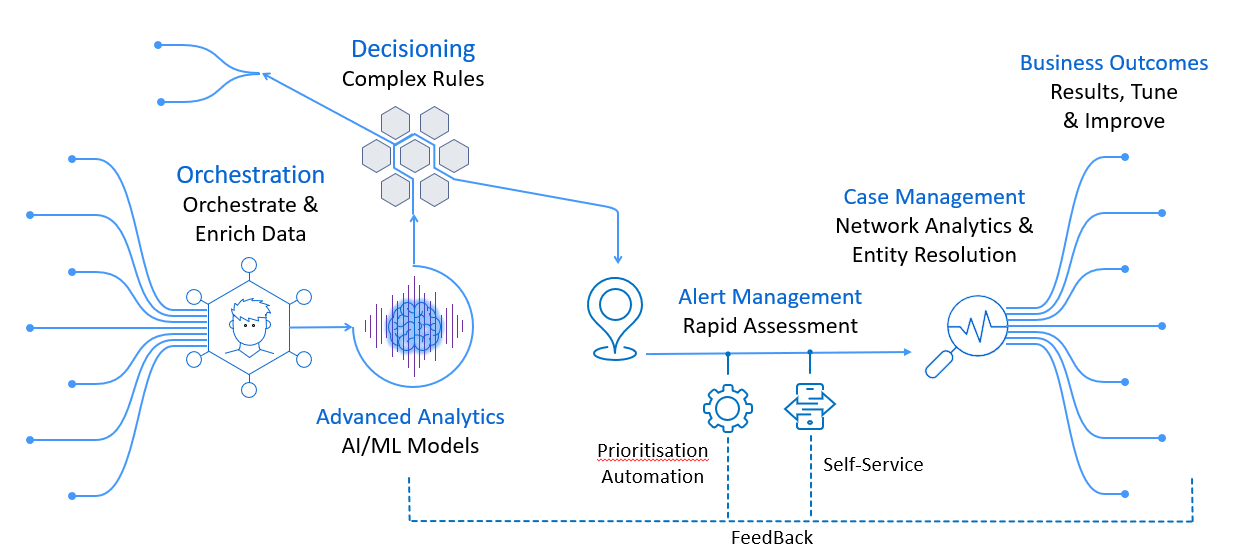


Figure System components

SAS Business Orchestration Services is included with SFD to orchestrate and enrich data flowing through the system with the necessary information to effectively make decisions on transactions or activities.

#### Advanced Analytics

SFD allows the use of **advanced analytics** such as A.I. or Machine Learning models to detect fraudulent behavior in streaming system messages, such as financial transactions, or applications).

#### SAS Detection Architecture

SAS has specific models package available for SFD but also allows for the creation of custom models since it is built upon the Viya 4 platform. From there, users use **SAS Detection Architecture** to write, test, and put into production complex rules that act on the model results or directly on the messages to help generate system alerts when needed.

#### SAS Alert Triage

**SAS Alert Triage** is the alert management component included with SFD to allow users to rapidly assess and triage the alerts which have been created by the rules.

#### SAS Visual Investigator

**SAS Visual Investigator,** available with some packages of SAS Fraud Decisioning, provides case management capabilities including incorporating network analytics and Entity Resolution.

All these components work together to deliver on SAS customer business outcomes and allow for results to be tuned and improved over time.

# E2E Data Flow

A blue hexagons with white text

AI-generated content may be incorrect.

Figure Data Flow

The diagram demonstrates the process within the SAS SFD (Suspicious Fraud Detection) Solution. The diagram outlines the integration of REST APIs and Kafka topics for data handling and alert management. Here's a breakdown of the key components:

* Transactions and alert recommendations are ingested into the system through either REST or Kafka. Once the data is ingested, alerts are generated.
* Once the analyst applies a disposition to the alert, actions are performed.
* These actions are executed internally within SAS Alert Triage or sent to a Kafka topic for external systems to process.
* Transaction markings, productivity ratings, and reason codes are also sent to Kafka topics.
* This provides external systems with information that was uncovered during the alert triaging process.
* For any errors that occur while processing alerts and transactions, the alerts service and the transaction service each have their own error topic to ensure no important information is lost.
* Simultaneously, the SAS-alerts service picks up alert recommendations, generates alerts, and then stores the alert details, in the "alert" PostgreSQL table.
* For REST, the process is slightly different. Transactions are sent directly to the SAS-alerts transaction service using the alerts Transaction/transactions endpoint, while alerting events are submitted via REST to the SAS-alerts service through the alerts/alerting Events endpoint.
* Additionally, productivity ratings and reason codes are placed on an outgoing Kafka topic specified in the SAS\_TRIAGE\_KAFKA\_OUTBOUND\_TOPIC environment variable
* Finally, any external actions that are part of applied dispositions are placed on the Kafka topic that is specified in the action group configuration definition. These actions are then picked up by external systems to be executed.

# Message Schema

A message schema defines the layout of variables. It consists of a collection of variable sets. Examples of message schemas are payments, loans, wires, identity, and so on. A message schema can be defined by you or by SAS. Message schemas that are provided by SAS are called prescribed templates. These templates promote consistency across solution implementations.

|  |  |  |  |
| --- | --- | --- | --- |
| Message Schema Name | Description | Type | Tag |
| Online Fraud | Online Fraud Solution for Khaleeji Use Case, Used in Mobile Banking and Internet Banking for both Individual and Corporate | Solution\_by\_sas | Khaleeji  Online  MobileBank  InternetBank |

#### Variable Sets

A variable set is a container of defined variables. A variable set can be nested within other variable sets. You can add any number of variable sets to a message schema.

|  |  |  |  |
| --- | --- | --- | --- |
| Message Schema | Variable Set Name | Description | Type |
| Online Fraud | Solution | Critical data for the SAS Fraud solution to execute and understand the transaction being processed | Object |
| Online Fraud | Authentication | Data about the authentication process undertaken | Object |
| Online Fraud | Applicant | Data about the Applicant, the person or party involved | Object |
| Online Fraud | Application | Data about the application being processed, such as lending amount and credit type | Object |
| Online Fraud | CardFinancial | Financial data about the card authorization or card posting, such as amounts and point of sale information | Object |
| Online Fraud | Company | Data provided about the Company and/or Corporate Institute involved in the Transaction | Object |
| Online Fraud | CreditAccount | Data about the Credit Account where the money is being sent to (recipient of funds) | Object |
| Online Fraud | Customer | Data provided about customer and/or user involved in the transaction | Object |
| Online Fraud | DebitAccount | Data about the debit account where the money is being sent from (source of funds) | Object |
| Online Fraud | Demographic | Data about non-monetary changes or requests | Object |
| Online Fraud | Merchant | Data about the card merchant involved in the transaction such as merchant category code | Object |
| Online Fraud | Payment | Financial data about the payment, wire or transfer, such as amount | Object |
| Online Fraud | Device | Data about a device, such as a smartphone, tablet, desktop or wearable | Object |
| Online Fraud | Digital | Data about a digital channel connection, likely via mobile application or browser | Object |
| Online Fraud | DateOfBirth | Data provided about date of birth | Array |
| Online Fraud | EmailAddress | Data provided about email contact details | Array |
| Online Fraud | Employment | Data provided about employment | Array |
| Online Fraud | Identification | Data about the Identity of the person or party involved | Array |
| Online Fraud | Location | Data provided about a location, such as an address or post office box | Array |
| Online Fraud | Phone | Data provided about telephone contact details | Array |
| Core | Request | Message Schema for all the Customer/Client Data - The data that the Tooling needs to run that comes from the customer | Object |
| Core | Alerted | Summary information of all the Rules that Recommended an Alert, De-Duped Array of Rules that Recommended Alert (all the rules in rulesfire variable set with alertFlg set to True) | Array |
| Core | Decision | Overall outcome data, all rules, models and summation data - scorecard totals etc. | Object |
| Core | ModelFired | Array of model execution data for each model executed against a message. | Array |
| Core | RuleFired | Data for each Rule that is fired against transaction / event - the rows are sent in an Array - so a row for every single rule | Array |
| Core | System | SAS Generated System Data - the data SDA provides, and the data SDA requires to run | Object |
| Core | Timings | Statistical information about message processing | Object |

#### Business Mapping

Business Mapping is used to map business attributes across technical fields with SFD available fields, used by Khaleeji team to structure desired format of JSON for these mapping to be communicated with SFD solution to process incoming transactions and give proper response, subjected to change according to business requirements and updates.

Below table for excel sheets with respective description

|  |  |  |
| --- | --- | --- |
| Label | Description | Attachment |
| Required Fields Avanza Transactions | Mapped SFD fields across Business requirements from Avanza (Individual) technical team |  |
| Required Fields Hafeez Transactions | Mapped SFD fields across Business requirements from Hafeez (Corporate) technical team |  |
| Response Format SFD | Response information that’s added to original message from SFD |  |
| SFD Request | Sample JSON request to be followed in sending transactions to SFD |  |
| SFD Response | Sample JSON response to be used in receiving transactions from SFD and act up on it |  |

# Transaction types

Transaction Types are how SAS Alert Triage classifies incoming transactions. There is no hard limit on the number of transaction types that can be defined. Transactions are identified as belonging to a particular transaction type based on the values appearing on that transaction in the variables targeted by the administratively defined discriminator variables.

Discriminator variables are used to specify the variable path that the system will follow when retrieving transaction type identifier values. An instance of SAS Alert Triage must have one or more discriminator variables specified.

When specifying the Discriminator Variables to use the administrator should ensure the following:

* That all the specified Discriminator Variables will appear on all incoming transactions
* That all transaction types to be specified will have at least one meaningful value associated with each of the Discriminator Variables

#### Configuring discriminator variables

|  |  |
| --- | --- |
| **Discriminator ID** | **Discriminator Variable Path** |
| activityType | message.solution.activityType |
| originationType | message.solution.originationType |

Each transaction type must specify a unique combination of expected values in the available discriminator variables.

#### Configuring Transaction Types

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Txn Type ID** | **Default Display Name** | **Msg Schema** | **Discriminator ID and Values** | **Indexable Variable Paths** | **Business Txn** | **Txn Datetime Path** | **Txn ID Path** |
| CKBP | Checking Account - Bill Payment | Online Fraud | originationType – CC  activityType - CA | Message.customer.number | Utility Bill Payment | message.sas.system.transactionDtTmUtc | Message.system,transactionID |
| CKFP | Checking Account - Faster Payment | Online Fraud | originationType – CK  activityType – FP | Message.customer.number | One Time payment Transaction | message.sas.system.transactionDtTmUtc | Message.system,transactionID |
| CKNM | Checking Account - Other Non-Monetary | Online Fraud | originationType – CK  activityType – NM | Message.customer.number | Beneficiary Registration Event | message.sas.system.transactionDtTmUtc | Message.system,transactionID |
| CKPT | Checking Account - Payment/Transfer ThirdParty | Online Fraud | originationType – CK  activityType – PT | Message.customer.number | Funds Transfer | message.sas.system.transactionDtTmUtc | Message.system,transactionID |
| CSDE | Customer - Demographic | Online Fraud | originationType – CS  activityType – DE | Message.customer.number | Change in account profile KYC Update | message.sas.system.transactionDtTmUtc | Message.system,transactionID |
| CSNM | Customer - Other Non-Monetary | Online Fraud | originationType – CS  activityType – NM | Message.customer.number | Login | message.sas.system.transactionDtTmUtc | Message.system,transactionID |
| CCCP | Credit Card - Posting | Online Fraud | originationType – CC  activityType – CP | Message.customer.number | Credit Card Payment | message.sas.system.transactionDtTmUtc | Message.system,transactionID |
| CKET | Checking Account – Electronic Transfer | Online Fraud | originationType – CK  activityType – ET | Message.customer.number | EFTS Funds Transfer | message.sas.system.transactionDtTmUtc | Message.system,transactionID |
| CSDV | Customer – Device | Online Fraud | originationType – CS  activityType – DV | Message.customer.number | Device Registeration | message.sas.system.transactionDtTmUtc | Message.system,transactionID |

# End points

This section targets the technical integration with SAS Fraud Detection system, using Rest Calls for Transaction movement and Kafka topics for system actions taken on UI

#### Rest EndPoints

Rest Calls will be sent through BOSS servers (accordingly Dev, UAT and Prod)

|  |  |  |
| --- | --- | --- |
| Endpoint | Description | Sample |
| http://<<BOSS IP>>:30080/FraudDetection | Post Request API endpoint, serve Transaction processing to SAS Fraud Detection engine, to assess and respond with proper action |  |

#### Kafka topics

Kafka is implemented on Khaleeji side, SFD use the provided Kafka for specific topics, below the topics would be in use for Khaleeji to act upon (e.g. in Hold transaction analyst action on UI need to be managed)

|  |  |  |
| --- | --- | --- |
| Topic | Description | Sample |
| <<Khaleeji to determine>> | Transaction Markings on Alert Triage user interface, where analyst determine if the transaction is (confirmed\_valid, confirmed\_invalid or marked\_for\_review) with extra information regards Alert or Transaction |  |

# Groups & Users

#### Users

Users and User Groups/Roles used within SAS Detection Architecture are administered within SAS Environment Manager. This includes the association of Capabilities to the User Groups. Users need to be assigned to User Groups that define what a user can do while logged into the User Interface.

#### Roles and Capabilities

The following roles are available by default:

* SAS Detection Junior Rules Editor
* SAS Detection Rules Editor
* SAS Detection Senior Rules Editor
* SAS Detection System Administrators

The following table details the capabilities that are enabled by default for each of these roles. New roles can be defined as needed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Capability** | **Name** | **Junior Rules Editor** | **Rules Editor** | **Senior Rules Editor** | **System Administrators** |
| View rules | sda.rules.view | Yes | Yes | Yes |  |
| Add and modify rules | sda.rules.update | Yes | Yes | Yes |  |
| Disable and enable rules | sda.rules.enableDisable |  |  | Yes |  |
| Promote rules without review | sda.rules.promote |  |  | Yes |  |
| Approve and promote rules requested by others | sda.rules.approve.promote |  | Yes | Yes |  |
| Delete rules without review (move to Recycle folder) | sda.rules.recycle |  |  | Yes |  |
| Approve and delete rules requested by others | sda.rules.approve.recycle |  | Yes | Yes |  |
| Empty rules in Recycle tab | sda.rules.delete |  |  | Yes |  |
| View message classifications | sda.message.classifications.view | Yes | Yes | Yes | Yes |
| Administer message classifications (for example, create or view message classifications outside an organization's context) | sda.message.classifications.manage |  |  |  | Yes |
| View organizations and message classifications, and associations of roles and schemas | sda.organizations.view | Yes | Yes | Yes | Yes |
| Modify organizations and message classifications, including adding and removing associations | sda.organizations.update |  |  |  | Yes |
| Add organizations and message classifications | sda.organizations.create |  |  | Yes | Yes |
| Delete organizations and message classifications, including associations | sda.organizations.delete |  |  |  | Yes |
| Deploy an organization's artifacts (for example, rules, schemas, models, and so on) | sda.deployments.create |  |  |  | Yes |
| View an organization's deployment of artifacts | sda.deployments.view |  |  |  | Yes |
| View projects and associations | sda.projects.view | Yes | Yes | Yes |  |
| Create projects and associations | sda.projects.create |  |  | Yes |  |
| Modify projects and associations | sda.projects.update |  |  | Yes |  |
| Delete projects and associations | sda.projects.delete |  |  | Yes |  |
| Add and modify a project's profile variable sets | sda.project.profile.variable.sets.update |  |  | Yes |  |
| Delete a project's profile variable sets | sda.project.profile.variable.sets.delete |  |  | Yes |  |
| View roles and users | sda.roles.and.users.view | Yes | Yes | Yes | Yes |
| Add alert types | sda.alert.types.create |  |  |  | Yes |
| Delete alert types | sda.alert.types.delete |  |  |  | Yes |
| View an impact analysis | sda.impact.analyses.view |  | Yes | Yes | Yes |
| Create, cancel, and delete an impact analysis | sda.impact.analyses.update |  | Yes | Yes |  |
| View models and associations | sda.models.view |  |  | Yes | Yes |
| Add models | sda.models.create |  |  |  | Yes |
| Delete models and associations | sda.models.delete |  |  |  | Yes |
| Update models and associations | sda.models.update |  |  |  | Yes |
| Display Advanced Lists on the left navigation bar. | sda.navigation.advanced.lists |  |  |  | Yes |
| Display Deployments on the left navigation bar. | sda.navigation.deployments |  |  |  | Yes |
| Display Impact Analysis on the left navigation bar. | sda.navigation.impact.analyses | Yes | Yes | Yes | Yes |
| Display Message Classifications on the left navigation bar. | sda.navigation.message.classifications |  |  |  | Yes |
| Display Message Schemas on the left navigation bar. | sda.navigation.message.schemas |  | Yes | Yes | Yes |
| Display Models on the left navigation bar. | sda.navigation.models |  |  | Yes | Yes |
| Display Organizations on the left navigation bar. | sda.navigation.organizations |  |  |  | Yes |
| Display Profiles on the left navigation bar. | sda.navigation.profiles |  | Yes | Yes | Yes |
| Display Projects on the left navigation bar. | sda.navigation.projects |  |  | Yes | Yes |
| Display Rules on the left navigation bar. | sda.navigation.rules | Yes | Yes | Yes | Yes |

#### Users

|  |  |  |
| --- | --- | --- |
| **User ID**  **(unique ID in SAS Environment Manager)** | **User Name** | **User Role** |
| e.g sasjsmth | Jordan Smith | SAS Detection Junior Rules Editor |
|  |  |  |

# **Alert Triage Database Schema**

Below file demonstrates the ERD of the Alert triage database schema: