BNY Mellon

Functional Requirements Document

**RTP FRAUD DETECTION SERVICE**

(Based on RTP BlackList Entities)

Real Time Payments (RTP)

**Document History** – describe the changes to this document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Date** | **Entered by** | **Comments** |
| 0.1 | July12, 2018 | Anand Singh | Initial Draft |
| 0.2 | July 16, 2018 | Anand Singh | Section 4.5 – Added Black list Entity logical data table |
| 0.3 | July 17, 2018 | Anand Singh | Modified scope of this FRD to be limited to Fraud Detection Service. (RFDS), based on feedback from Avnish Gupta & Mayank Khanna  Re-organized document structure around RFDS. |
| 0.4/0.5 | July 18, 2018 | Anand Singh | Revised RFDS Processing Steps, Detection Methodology, Request Payload sections.  Updated Requirements Summary & Acceptance Criteria |
| 0.6 | July 20, 2018 | Anand Singh | Revisions based on feedback from Carl Slabicki, Andrew Haskell |

**Table of Contents**

Table of Contents

[2 Document Reviews & Approvals 5](#_Toc519866139)

[3 Background: 7](#_Toc519866140)

[3.1 Scope: 7](#_Toc519866141)

[3.2 Out of Scope: 7](#_Toc519866142)

[3.3 Assumptions: 7](#_Toc519866143)

[4 General Flow: 8](#_Toc519866144)

[4.1 RTP Fraud Detection Service (RFDS): 8](#_Toc519866145)

[4.2 RFDS: Processing Steps 9](#_Toc519866146)

[4.3 RFDS: BlackList Entity Structure 12](#_Toc519866147)

[4.4 RFDS: Request Payload 13](#_Toc519866148)

[4.5 RFDS: Fraud Detection Methodology 16](#_Toc519866149)

[4.6 RFDS: Response Payload 19](#_Toc519866150)

[4.7 RFDS: User Interface 20](#_Toc519866151)

[5 Requirements Summary 22](#_Toc519866152)

[5.1 Summary Log 22](#_Toc519866153)

[5.2 Acceptance Criteria 23](#_Toc519866154)

[6 Performance Characteristics and Requirements 24](#_Toc519866155)

[7 Contributors 26](#_Toc519866156)

[8 Terms and Abbreviations 26](#_Toc519866157)

[9 Appendix 27](#_Toc519866158)

[9.1 Field list for Fraud Scan – List of relevant fields for fraud check on pacs.008, pain.013& camt.056 messages. 27](#_Toc519866159)

[9.2 RFDS – Process Flow Diagram 27](#_Toc519866160)

[9.3 Payment Engine Fraud Processing – Direct Clients 27](#_Toc519866161)

[9.4 Private Label Bank FRD 27](#_Toc519866162)

# Document Reviews & Approvals

The stakeholders listed below need to review or approve this document respectively. When stakeholders review or approve this document, they are indicating that they have carefully:

* Read
* Reviewed
* Considered the impact of this document to their organization
* And agree that the document is:
  + Accurate
  + Complete
  + Sufficient
  + Consistent with the project needs
  + Meets BNY Mellon standards and best practices

Note that it will be the same list of persons who will review and approve all subsequent documentation.

Depending on the topic it may be possible that other persons will be added on a one-off basis to ensure that the areas that may be impacted by a certain topic will have a chance to provide their input.

**Reviews**

|  |  |  |  |
| --- | --- | --- | --- |
| Reviewer | Role | Reviewer Responsibility  *(sections being reviewed)* | Version Approved |
| Carl Slabicki | Product Manager | All Sections |  |
| Avnish Gupta | Application Development Manager | All Sections |  |
| Mayank Gupta |  | All Sections |  |
| Martin Townsend |  | All Sections |  |
| Anshul Agrawal | Volante Technical/ Business SME | All Sections |  |

**Approvals**

|  |  |  |  |
| --- | --- | --- | --- |
| Approver | Role | Approver Responsibility  *(sections being approved)* | Version Approved |
| Carl Slabicki | Product Manager | All Sections |  |
| Avnish Gupta | Application Development Manager | All Sections |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Background:

The Clearing House is undertaking a multi-year initiative to build a safe, sustainable standards-based and ubiquitous real-time payment system for all U.S. Financial Institutions. The payment system will provide a competitive tool for banks to compete with non-bank payment providers as a safer more viable alternative for clearing payments. Real-Time Payments will be an industry changing initiative in the USD payments arena as they will provide consumers and businesses with the ability to immediately send and receive funds directly from their accounts at U.S. financial institutions anytime 24/7/365.

BNYM is working to be one of the first financial institutions to deliver a Real Time Payment service.

## Scope:

This documents covers the methodology and requirements for fraud detection that needs to be performed as part of RTP processing.

An RTP Fraud Detection Service needs to be developed – as an application external to RTP Payment Engine.

The types of messages in scope as eligible for Fraud detection are:

* Credit transfer ( pacs.008) – Inbound and Outbound
* Request for Payment (pain.013) – Inbound and Outbound
* Request for Return of Funds (camt.056) – Inbound and Outbound

## Out of Scope:

TCH generated Fraud Alerts are currently out of scope. These are not defined by TCH at this time.

Message types not mentioned in scope are considered excluded at this time.

## Assumptions:

1. This document assumes that the reader is aware of the overall processing of the Real Time Payments system and ancillary components. The document covers the flows related to the compliance checks only.

# General Flow:

Three types of compliance checks needed for Real Time Payments processing.

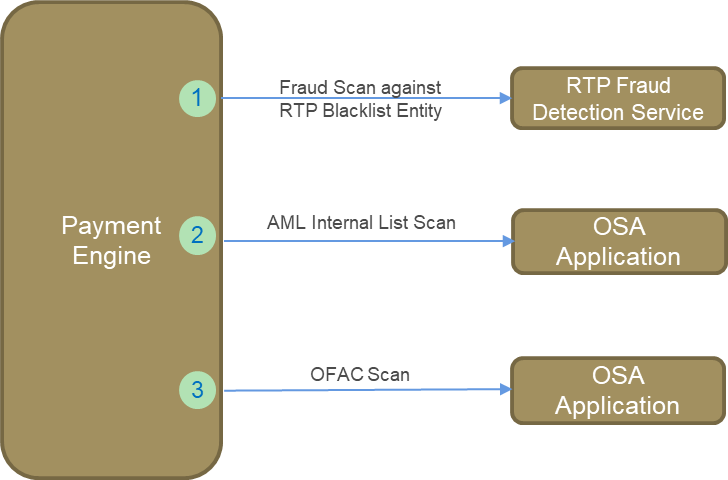
These checks need to be performed in the following order:

1. Fraud Scan using RTP Fraud Detection Service, followed by

2. OSA Scan for AML Internal List, followed by

3. OSA Scan for Sanctions Checking.

This document covers the requirements and methodology around RTP Fraud Detection Service



## RTP Fraud Detection Service (RFDS):

An “RTP Fraud Detection Service (RFDS)" will need to be developed as an application external to the Payment Engine to allow for a more “services” based framework that supports easier plug & play/ enhancements in future .

This service will maintain a list of suspected fraud items called – “Black List Entities”.

The Payment engine shall invoke/ call this service for compliance checks related to fraud. This service shall perform the following functions:

* Provide a facility to maintain a list of suspected fraud items: “Black List Entity” comprising of creditor & debtor names, addresses, account #s, etc. – details of which are provided in [section 4.3 RFDS: BlackList Entity Structure](#_Fraud_Module:_Black)
* Scan for/ detect the presence of Black List Entity parameters in selected ‘relevant fields' of the pacs.008 (Credit Transfer), pain.013 (Request for Payment) and camt.056 (Request for Return of Funds) messages being processed by RTP. Details on the relevant fields can be found on [section 4.4 RFDS: Request Payload](#_RFDS:_Request_Payload)
* Return back a response to the caller with either a "Hit" or a “No Hit". In case of a Hit an appropriate reason code with impacted message type and field names would be included in the response.
* Provide a facility for entitlements based admin functions/second level approvals for Black List Entity maintenance through a Graphical User Interface.
* Provide a facility to view results of the scans performed.

## RFDS: Processing Steps

Please refer to the Sequence Diagram for process steps & interactions between RFDS and Payment Engine.

NOTE:

Details on internal processing within Payment Engine have been omitted from the sequence diagrams for brevity. These are related to:

* Processing steps followed after receiving response back from RFDS.
* Accounting entries & reversal entries to denote fund movement between Suspense account and Customer account
* BEX event and related message generations
* For details, see appendix
* Direct Clients: [Payment Engine Fraud Processing – Direct Clients](#_Payment_Engine_Fraud)
* PL Clients: [Private Label Bank FRD](#_Private_Label_Bank)

Upon completion of the Fraud check the Payment Engine will trigger the AML Internal List check via the OSA application as next step.

**High-level process flow description:**

1) The Payment Engine initiates a Fraud Check with a request to the RTP Fraud Detection Service. The request payload shall have the relevant fields from the payment messages in scope.

2) RFDS performs a scan of the request payload against the list of Blacklist entities maintained by the Service. The methodology of the scanning is described in [*Section 4.5 RFDS:Fraud Detection Methodology*](#_RFDS:_Fraud_Detection)

3a) The result of the fraud scan does not yield a match. A “No Hit” response payload from RFDS is sent back to the Payment Engine.

3b) the result of the fraud scan has fields in the request payload that are a match with the Blacklist Entities. A “Hit” response payload from RFDS is sent back to the Payment Engine.

4a) The Payment Engine marks the transaction for successful completion of Fraud check Proceed with the next step in processing.

4b) The Payment Engine receives a response payload with a “Hit” for fraud.

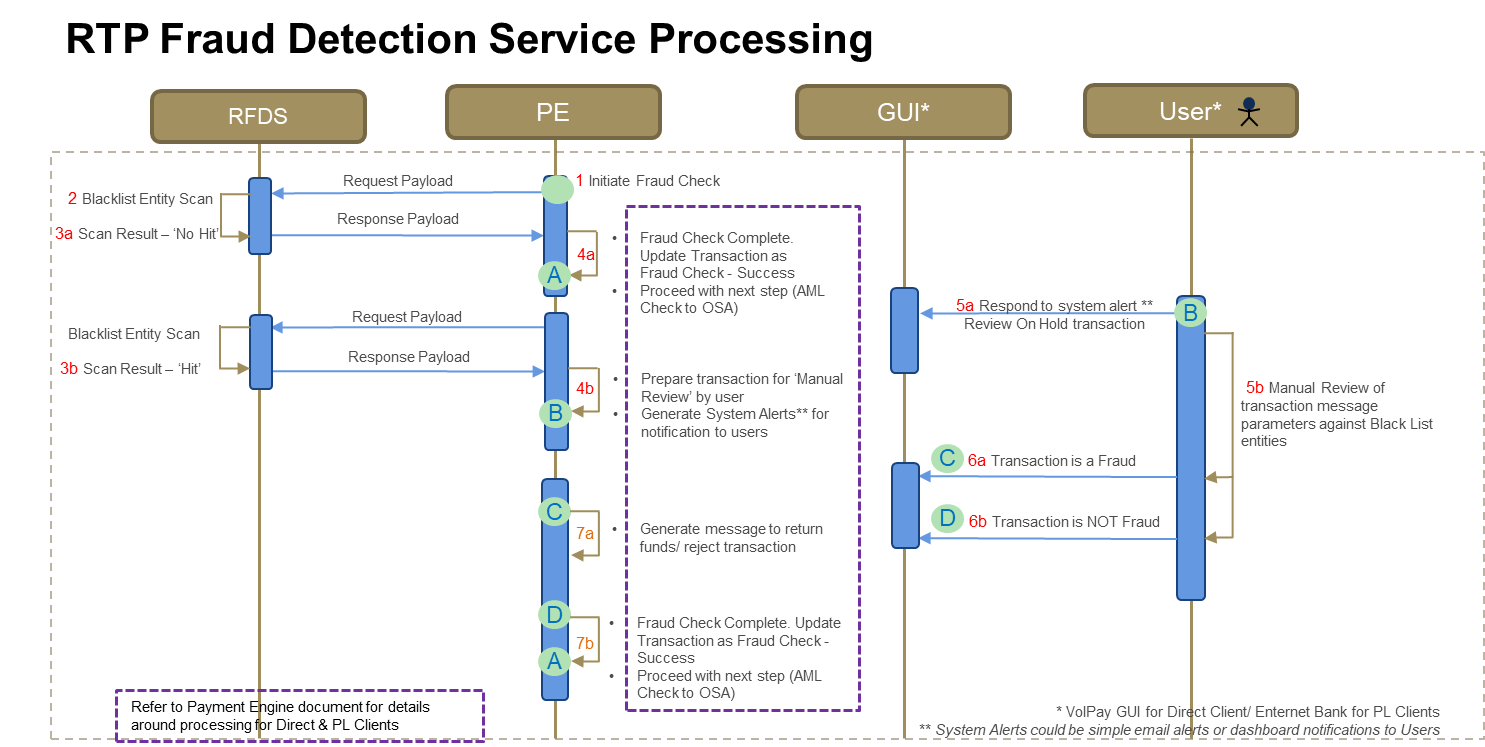
* There are different processing path-ways for Direct vs PL Clients in this scenario (outside the scope of this document - Please refer to the attached documents in the appendix for details)
* The Payment Engine shall generate system alerts to alert users for manual review. The system alerts could either be simple email alerts to mail-groups initially or a more elaborate GUI based Dashboard. The alert mechanism should be aligned with the standard process of Enternet bank for PL Bank and VolPay GUI for Direct Clients respectively.

5a/b – The user (Treasury User for Direct clients/ PL customer representative for PL clients) shall respond to system alert and log in the appropriate GUI to look at the transaction under marked for manual review after a “Hit” response from the RFDS. The Treasury user shall use the Volpay GUI and the PL client representative will use the Enternet Bank UI.

6a/b – Based on the manual review, the user shall process the transaction from the GUI either as fraud or not a fraud.

7a – Transaction was deemed to be Fraud by the user. The Payment Engine shall generate appropriate messages to reject the transaction/ return the funds.

7b – Transaction was deemed to be Not A Fraud by the user. The Payment Engine marks the transaction for successful completion of Fraud check and proceeds with the next step in processing – A request is sent to OSA for AML Internal List check (outside the scope of this document).



## RFDS: BlackList Entity Structure

The logical data model for Blacklisted entity is as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Description | Type | Length |
| Id | Blacklist Entity Identifier | Number | 10 |
| Name | Name of the Blacklisted entity | Text | 140 |
| Account # | Account Number | Number | 10 |
| Status | Status of the Blacklist entity (whether Active or Inactive) | Text | 8 |
| Street Name | Street Name (Address) | Alphanumeric | 70 |
| Building Number | Building Number (Address) | Alphanumeric | 16 |
| Post Code | Postal or Zip Code ( Address) | Alphanumeric | 16 |
| Town Name | City or Town (Address) | Text | 35 |
| Country | Country | Text | 2 |
| Type | Business or Person | Text | 20 |
| Reason | Reason why the Entity is black listed | Text | 20 |
| Last Update | Date of creation/ modification (Date-time)  YYYY-MM-DDTHH:MM:SS | Date-Time | 19 |
| Update By | Userid of user creating/ updating | Alphanumeric | 10 |
| Approved By | Userid of approver | Alphanumeric | 10 |
| Service Code | Identifier for the Service for which this Blacklist is applicable e.g. RTP, CXC | Text | 10 |

Sample Data for illustration purposes *(Update by, Approved by & Service code fields are not shown)*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BlackList Entity - Sample Data | | | | | |  |  |  |  |  |  |
| Id # | Name | Account # | Status | Address Fields | | | | | Type (Business/ Personal) | Reason | Date Last Updated |
| Street Name | Building Number | Post Code | Town Name | Country |
| 1 | John Doe | 1122334455 | Active | 123 | 999 | 11111 | Timbuktoo | Cuba | B | AML | 2018-07-13 |
| 2 | Mary Jane | 1234567890 | Active |  |  |  |  | Iran | P | Default | 2018-07-13 |
| 3 | Spider  Man | 1212121212 | Inactive |  |  |  |  | North Korea | B |  | 2018-07-13 |

## RFDS: Request Payload

The table below lists out the ‘relevant fields’ needed in a Fraud Request to be sent from the Payment Engine to the RFDS. These fields have been identified for potential fraudulent data on pacs.008, pain.013 and camt.056 messages.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Request Payload from Payment Engine** | |  |
| Index No. | Message Item | Comments | M/O/C |
| M: Mandatory; O: Optional; C: Conditional | |
| 1 | FraudCheckRequest |  |  |
| 1.1 | MessageType | Can be any of the three message types - pacs.008, pain.013, camt.035 | M |
| 1.2 | RequestId | An identifier for the Fraud Scan request from Payment Engine to Fraud Module | M |
| 1.3 | Date-Time | Date time Stamp of the request sent | M |
| 1.4 | FieldList | List of Fields to be sent as part of the Fraud Scan request |  |
| 1.4.1 | **EndToEndId** | End to End Identifier | M |
| 1.4.2 | Debtor | List of fields related to Debtor |  |
| 1.4.2.1 | **Name** |  | M |
| 1.4.2.2 | **StreetName** |  | C |
| 1.4.2.3 | **BldgNumber** |  | O |
| 1.4.2.4 | **PostCode** |  | C |
| 1.4.2.5 | **TownName** |  | C |
| 1.4.2.6 | **Country** |  | C |
| 1.4.2.7 | DebtorAcct | Account Identifier for the Debtor |  |
| 1.4.2.7.1 | **Id** |  | M |
| 1.4.3 | Creditor | List of fields related to Creditor |  |
| 1.4.3.1 | **Name** |  | M |
| 1.4.3.2 | **StreetName** |  | C |
| 1.4.3.3 | **BldgNumber** |  | O |
| 1.4.3.4 | **PostCode** |  | C |
| 1.4.3.5 | **TownName** |  | C |
| 1.4.3.6 | **Country** |  | C |
| 1.4.3.7 | CreditorAcct | Account Identifier for the Creditor |  |
| 1.4.3.7.1 | **Id** |  | M |
| 1.4.4 | RemittanceInfo | Free-form field for Unstructured Remittance information |  |
| 1.4.4.1 | **Unstructured\*** | Only being used in pacs.008 and pain.013 message | O |
| 1.4.5 | **AdditionalInfo\*\*** | Only being used in camt.056 message | O |
| 1.4.6 | AddtionalParams | Additional parameters can be added as Key-Value pairs as needed in future |  |
| 1.4.6.1 | **Key** |  | O |
| 1.4.6.2 | **Value** |  | O |

Index 1.4.6, 1.4.6.1 -2: Added to accommodate additional parameters for future use. These can be added as Key Value pairs.

***[Technical Design Considerations]***  *Most of the above fields are present on the BEX Event messages (****BE1AI*** *event) that are generated by the Payment Engine. Payment Engine should be able to leverage/ enhance this message to serve as the Request Payload to RFDS.*

*A sample message for such an event is as below:*

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<VolPayHubAlertNotification1 xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*>

<UniqueReferenceId>8ec725ba-8b22-4e40-a6ce-6bd07457a7be</UniqueReferenceId>

<SourceSystemId>EPH</SourceSystemId>

<EventID>BE1AI</EventID>

<EventDateTime>2018-07-12T13:38:16.080-04:00</EventDateTime>

<Version>1.0.0</Version>

<Contents>

<messageSubject>BE1AI</messageSubject>

<messageContent>Transaction Received</messageContent>

<PaymentControlDataNo>

<OfficeCode>BNYMGLBL</OfficeCode>

<BranchCode>801</BranchCode>

<DepartmentCode>USRPXDEPT</DepartmentCode>

<InstructionID>9214902</InstructionID>

<PaymentID>EPR1807120000006</PaymentID>

<DistrubtionID>EPR1807120000006</DistrubtionID>

<ServiceID>RPX000000026912</ServiceID>

<InstructionData>

<PartyServiceAssociationCode>5371303589\_RPX\_ISOPacs8ChannelInput</PartyServiceAssociationCode>

<Source>CUSTOMER</Source>

<SourceCategory>volpay-ui</SourceCategory>

<InstructionType>PaymentInstruction</InstructionType>

<TransportType>INITIATED</TransportType>

<ReceivedDate>2018-07-12T17:38:14.102-04:00</ReceivedDate>

<ServiceCode>RPX</ServiceCode>

<SourceFormat>ISOPacs8ChannelInput</SourceFormat>

</InstructionData>

<PaymentCategory>OUTBOUND</PaymentCategory>

<OriginalPaymentReference>TEST75DAY130PM</OriginalPaymentReference>

<OriginalPaymentFunction>Customer Credit Transfer</OriginalPaymentFunction>

<ExecutionDateTime>2018-07-12T17:38:15.282-04:00</ExecutionDateTime>

<LclInstrm\_Prtry>BUSINESS</LclInstrm\_Prtry>

<Amount>1.0</Amount>

<Currency>USD</Currency>

<OriginalValueDate>2018-07-12-04:00</OriginalValueDate>

<ValueDate>2018-07-12-04:00</ValueDate>

<ProductSupported>REALTIME</ProductSupported>

<Debtor>

<Name>THE BANK OF NEW YORK MELLON 5371303589</Name>

<Address>REAL TIME PAYMENTS 6023 AIRPORT ROAD</Address>

<Account>8901418455</Account>

<PstlAdr>

<PstCd>13424</PstCd>

<TwnNm>ORISKANY</TwnNm>

<CtrySubDvsn>NY</CtrySubDvsn>

<Ctry>US</Ctry>

</PstlAdr>

</Debtor>

<DebtorAgent>

<ClearingSchemeID>021000018</ClearingSchemeID>

<SchemeCode>USRTP</SchemeCode>

</DebtorAgent>

<Sender>

<Name>801 - NY Head Office</Name>

<BIC>IRVTUS3NXXX</BIC>

<ClearingSchemeID>255072126</ClearingSchemeID>

<SchemeCode>USRTP</SchemeCode>

</Sender>

<Creditor>

<Name>US BANK</Name>

<Account>153910315016</Account>

</Creditor>

<CreditorAgent>

<ClearingSchemeID>123000848</ClearingSchemeID>

<SchemeCode>USRTP</SchemeCode>

</CreditorAgent>

<Status>IN\_PROGRESS</Status>

<DetailedStatusInfo></DetailedStatusInfo>

<PaymentIndex>1</PaymentIndex>

<ServiceCode>RPX</ServiceCode>

<OrigInstrID>9214902</OrigInstrID>

<OrigTxID>EPR1807120000005</OrigTxID>

<MessageSource>CUSTOMER</MessageSource>

<UserActions>

<UserID>XECC527</UserID>

<Action>INITIATED</Action>

<TimeStamp>2018-07-12T17:38:14.100</TimeStamp>

</UserActions>

</PaymentControlDataNo>

</Contents>

</VolPayHubAlertNotification1>

A mapping of the required fields from the sample BE1AI message is provided as below. Some fields are not available on the current event message and data may have to be added to the Request Payload message.

|  |  |  |
| --- | --- | --- |
| Index No. | Request Payload message field | BE1AI Event message field |
| 1 | FraudCheckRequest |  |
| 1.1 | MessageType | <SourceFormat>ISOPacs8ChannelInput</SourceFormat> |
| 1.2 | RequestId | <PaymentID>EPR1807120000006</PaymentID> |
| 1.3 | Date-Time | <EventDateTime>2018-07-12T13:38:16.080-04:00</EventDateTime> |
| 1.4 | FieldList |  |
| 1.4.1 | **EndToEndId** | Not available on the BE1AI event message |
| 1.4.2 | Debtor |  |
| 1.4.2.1 | **Name** | <Name>THE BANK OF NEW YORK MELLON 5371303589</Name> |
| 1.4.2.2 | **StreetName** | <Address>REAL TIME PAYMENTS 6023 AIRPORT ROAD</Address> |
| 1.4.2.3 | **BldgNumber** | Not available on the BE1AI event message |
| 1.4.2.4 | **PostCode** | <PstCd>13424</PstCd> |
| 1.4.2.5 | **TownName** | <TwnNm>ORISKANY</TwnNm> |
| 1.4.2.6 | **Country** | <Ctry>US</Ctry> |
| 1.4.2.7 | DebtorAcct |  |
| 1.4.2.7.1 | **Id** | <Account>8901418455</Account> |
| 1.4.3 | Creditor |  |
| 1.4.3.1 | **Name** | <Name>US BANK</Name> |
| 1.4.3.2 | **StreetName** |  |
| 1.4.3.3 | **BldgNumber** | Not available on the BE1AI event message |
| 1.4.3.4 | **PostCode** |  |
| 1.4.3.5 | **TownName** |  |
| 1.4.3.6 | **Country** |  |
| 1.4.3.7 | CreditorAcct |  |
| 1.4.3.7.1 | **Id** | <Account>153910315016</Account> |
| 1.4.4 | RemittanceInfo |  |
| 1.4.4.1 | **Unstructured** | Not available on the BE1AI event message |
| 1.4.5 | **AdditionalInfo** | Not available on the BE1AI event message |
| 1.4.6 | AddtionalParams |  |
| 1.4.6.1 | **Key** | Not applicable |
| 1.4.6.2 | **Value** | Not applicable |

## RFDS: Fraud Detection Methodology

RFDS Fraud Detection methodology will be built around two components:

* Blacklist Entity field to Payment Message field mapping.
* FraudScore – a weighted score to determine a “Hit”

**BlackList Entity field to Payment Message field mapping**

RFDS shall scan each of the Blacklisted entity field values against 'relevant fields' on the Request Payload for a Hit.

The mapping between these fields has been shown in the table below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BlackList Entity Field to Payment Message Field Scan Mapping | | | | | | | |
| BlackList Entity Fields | Name | Account # | Address Fields | | | | |
| Street Name | Building Number | Post Code | Town Name | Country |
| Payment Message ‘Relevant Fields’ | Debtor Name Creditor Name End To End Id Remittance Info Additional Info | Debtor A/c Id Creditor A/c Id End To End Id Remittance Info Additional Info | Debtor Street Name/ Creditor Street Name | Debtor Building Number/ Creditor Building Number | Debtor Post Code/ Creditor Post Code | Debtor Town Name/ Creditor Town Name | Debtor Country/  Creditor Country |

As an example:

The value in the 'Name' field from Black List shall be scanned against:

* Debtor and Creditor name,
* End to End Id,
* Remittance Info,
* Additional Info

on the incoming and outgoing Credit Transfer, Request For Payment & Request for Return of Funds messages.

The scan will be case and space in-sensitive.

**FraudScore**

There are various fields in a Blacklist Entity - (Name, Account#, Address fields, etc). While RFDS has a map defined for these fields against Payment Message fields, in order to effectively define a “Hit”, there is a need to generate a ‘score’ that takes into account the weightage/ significance each of the Blacklist entity fields have.

For example, the probability for a match to be found for the “name” field in the Blacklist entity against payment messages is very high – which could lead to a large number of ‘false positives’.

Comparatively, the probability of a match against account# field being a ‘false positive’ is very low.

In order to account for this, FRDS will use a simple weighted score generation algorithm to come up with a “FraudScore”.

The methodology is explained below:

* Each ‘relevant field’ on the Request payload (from the payment message) shall be assigned a weight.
* RFDS will scan the Blacklist entity fields against the relevant fields as shown in the mapping table above.
* Based on the matches found RFDS will compute a simple numerical sum of the weights to arrive at a total score based on the matches. – This will be the Fraud Score for the scan.
* The FraudScore can be any numerical value between 0 to 10.

The table below is a representation of a Simple Weighted Score generation. A different algorithm may be used for more effectiveness if required.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Field-wise Weight Assignment | | | | | | | | | |
| Message Type | | Debtor/ Creditor Account # | Debtor/ Creditor Name | Street Name | Building Number | Post Code | Town Name | Country | End To EndId | Remittance Information | Additional Information |
| Incoming | pacs.008 | 5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1 | 1 |  |
| pain.013 | 5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1 | 1 |  |
| camt.056 | 5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1 |  | 1 |
| Outgoing | pacs.008 | 5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1 | 1 |  |
| pain.013 | 5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1 | 1 |  |
| camt.056 | 5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1 |  | 1 |

*A weighted Score can be generated based on a* ***Sum*** *of all the weights for every field with a 'Hit'*.

Example:

* + If say, all the relevant fields are matched in a scan of an incoming pacs.008 message the total FraudScore would be 10
  + If say, only the Debtor name and address fields are matched, the total FraudScore would be 3

**Reporting a “Hit” methodology in the Response Payload**

Based on the FraudScore, in-order to minimize the impact of false positives, RFDS shall use the following strategy to report a “Hit” on the response payload it sends back to the Payment Engine:

* **FraudScore = 0** (FraudScore equals zero)**: "No Hit"**
  + This is possible when none of the fields from the BlackList Entity match with the relevant fields on the payment message
  + *Payment Engine would mark the fraud check as complete and proceed with the next step in the transaction processing.*
* **0 < FraudScore < 5** (FraudScore greater than zero but less than five)**: "No Hit"** 
  + This score is possible for situations with high possibility of false-positives and of less significance.
  + Payment Engine would mark the fraud check as complete and proceed with the next step in the transaction processing. *Please note that any match with the “Account #” field would automatically put the score to 5 or higher and not meet this condition*
* **FraudScore>=5** (FraudScore of 5 and above): **"Hit"** 
  + This score would be achieved if a significant field such as Account # gets matched or a large number of other fields are matched.
  + This score indicates a high probability for the transaction to be a fraud.
  + Payment Engine would initiate a manual investigation.. The results of the manual investigation would mark the transaction as either "No Fraud" - which would allow it to proceed forward, or "Fraud" - which would result in the rejection of the transaction.

***RFDS shall provide facility to update/ modify weight assignments to the fields and also adjust the FraudScore threshold value for reporting a “Hit” through a GUI screen.***

## RFDS: Response Payload

The table below lists out the fields RFDS shall include in its response back to the Payment Engine.

|  |  |  |
| --- | --- | --- |
|  | **Response Payload from RFDS to Payment Engine** | |
| Index No. | Message Item | Comments |
| 1 | FraudCheckResponse |  |
| 1.1 | MessageType | Denotes the message type for which the Fraud Scan was performed (pacs.008, pain.013, camt.056) |
| 1.2 | RequestId | Fraud Scan request Identifier from Payment engine for which this response maps to |
| 1.3 | Date-Time | Date time stamp of the response sent |
| 1.4 | FraudScanResult | Results of the Fraud Scan |
| 1.4.1 | HitResult | Whether a "Hit" occurred against the identified fields for potential fraud. A "Hit" in case of a match and a "No Hit" for no match across any of the fields |
| 1.4.2 | FraudScore | A numerical value based on a weighted score generated based on matches against different fields to address future requirement. Please see section on [*Fraud Detection Methodology*](#_RFDS:_Fraud_Detection) |
| 1.4.3 | FieldListMatched | A name value listing of all the fields that were matched during Fraud Scan |
| 1.4.3.1 | FieldName | Name of the field |
| 1.4.3.2 | FieldValue | Value that was matched |

A sample XML Message of the Response payload is as below:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<Message xmlns=*"urn:tch"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*>

<FraudCheckResponse>

<MessageType>CreditTransfer</MessageType> <!-Can be any of the three message types - pacs.008, pain.013, camt.035>

<RequestId>123456789</RequestId> <!- Maps to the original RequestId for Fraud Check>

<Date-time>YYYY-MM-DDT00:00:00</Date-time>

<FraudScanResult>

<HitResult>Hit</HitResult> <!- As per Fraud Detection methodology>

<ScanScore>6</ScanScore> <!-Based on Score generated using Weighted field-wise scan match.>

<FieldListMatched> <!-Has a name value listing of all the fields that were matched during Fraud Scan>

<FieldName></FieldName>

<FieldValue></FieldValue>

</FieldListMatched>

</FraudScanResult>

</FraudCheckResponse>

</Message>

## RFDS: User Interface

RFDS needs a user interface to:

* Maintain Black List entities – view, create, update, delete (de-activate).
* Perform entitlements based admin functions/ second level approvals for Black List Entity maintenance.
* View results of the scans performed (requests serviced) by the RFDS.

The following mock-ups have been drawn to provide a generic reference for GUI screen development:

**View Screen** for a grid based view of the current list of black list entities with provisions for Create, Update, Delete (de-activate) & Filter Entities.



**Create/ Update Screen** to allow for creation of a new Entity or modification of an existing entity.



**Approval Screen** to allow for second level approval of any entity modifications/ new entity creation.



**Fraud Scan Results View Screen** This screen would provide users to view the results of the scans performed by the Fraud module against the Request Payloads it receives while it does the matching of the payment message fields against the Black List Entities.



# Requirements Summary

## Summary Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Func Req ID** | **Name** | **Requirement Description** | **JIRA #** |
| RFDS-F5.1.1 | RFDS – Blacklist Entity Maintenance | The RTP Fraud Detection Service shall maintain a list of Blacklist Entities as per a defined structure and format as per [section 4.3 RFDS: BlackList Entity Structure](#_RFDS:_BlackList_Entity) |  |
| RFDS-F5.1.2 | RFDS – Blacklist Entity Maintenance – GUI | There should be a Graphical User Interface to allow for creation, update, deletion (de-activation) of Blacklist Entities. GUI shall provide entitlements based approval/ admin functionality. The GUI shall provide capability to view the results of the scans performed by RFDS. Please refer to [section 4.7 RFDS: User Interface](#_RFDS:_User_Interface) for details.  Additionlly, RFDS shall provide facility to update/ modify weight assignments to the fields and also adjust the FraudScore threshold value for reporting a “Hit” through a GUI screen. |  |
| RFDS-F5.1.3 | RFDS: Request Generation | Payment Engine will send a fraud scan request to RFDS as per the format defined in Request Payload. The request will be based on ‘relevant fields’ identified on pacs.008, pain.013 & camt.056 messages that are in-scope for fraud check. Please refer to [section 4.4 RFDS: Request Payload](#_RFDS:_Request_Payload) for details. |  |
| RFDS-F5.1.4 | RFDS: Response Generation | RFDS will send a response back to the Payment Engine in a defined format based on the results of the fraud scan. This response will be for the fraud check request sent by the Payment Engine. Please refer to [section 4.6 RFDS: Response Payload](#_RFDS:_Response_Payload) for details. |  |
| RFDS-F5.1.5 | RFDS: Performing Fraud Scan | RFDS shall scan for fraud based on:   * Blacklist Entity field to Payment Message field mapping. * FraudScore – a weighted score to determine a “Hit”   The result of the Fraud Scan will be reported as either a “Hit” or “No Hit” along with a FraudScore value ranging from 0 to 10. These shall be included in the response sent back by RFDS. Please refer to [section 4.5 RFDS: Fraud Detection Methodology](#_RFDS:_Fraud_Detection) for details. |  |
| RFDS-F5.1.6 | RFDS process interaction with Payment Engine | Payment Engine and RFDS will be integrated as per the process defined in [section 4.1: RFDS Processing Steps](#_RFDS:_Processing_Steps) The Payment Engine will initiate a Fraud Check request, send a request payload over to RFDS. RFDS will perform a scan respond back with a response payload. The Payment Engine shall use the “Hit” or “No Hit” response to either hold the transaction for manual review by user or mark the fraud check as complete and move forward with next steps in transaction processing. The manual review by the users will involve action via a GUI (either VolPay GUI for Direct client, or Enternet Bank GUI for PL client) |  |

## Acceptance Criteria

|  |  |
| --- | --- |
| **Acceptance Criteria** | |
| RFDS-F5.1.1 | * The Blacklist entities will be stored a tabular format with all the relevant fields populated as per the requirements. |
| RFDS-F5.1.2 | * The users should be able to Create, Update, Delete (de-activate), view, search Blacklist Entities. * The Blacklist entity data should be viewable in a standard Grid format and have search and filter capabilities. * The GUI should have a workflow to enable second level approval for all modifications. * RFDS scan results should be viewable on the GUI with drill down capabilities for details on the request payload from the Payment Engine * Users should be able to update/ modify weight assignments to the ‘relevant fields’ and also the FraudScore threshold value for reporting a ‘Hit” |
| RFDS-F5.1.3 | * Request Payload has all the relevant fields required as per the defined format |
| RFDS-F5.1.4 | * Response payload has all the relevant fields as per the defined format |
| RFDS-F5.1.5 | * RFDS will perform a fraud scan as per the methodology defined in the requirement. * The result of the scan should have a “Hit” or “No Hit” determination based on the FraudScore generated |
| RFDS-F5.1.6 | * Payment Engine is able to correctly process the response back and either hold the transaction for manual review or mark it as complete for fraud check and move it to the next stage of processing. |

# Performance Characteristics and Requirements

|  |  |
| --- | --- |
| Interface behavioral characteristics.  Please check all of the Interface behavioral characteristics that apply from the list below. | Place "X" beside each item that applies |
| User Interaction | X |
| Batch cycle |  |
| Triggered events | X |

|  |  |
| --- | --- |
| Performance requirements – Online.  What is the average end user response time that your application/system will require? | Place "X" beside the item that applies |
| High (sub-second) | X [Max cycle time of 100ms (from Request receipt to response sent)] |
| Medium (seconds) |  |
| Low (variable) |  |
| Next day |  |

|  |  |
| --- | --- |
| Volumetrics (average).  (e.g., Transaction rates, Inquiries) What is the expected average level of transactions/second to this application/system expected to be? | Place "X" beside the item that applies |
| High (>10 transactions per second) | X |
| Medium (1-10 transactions per second) |  |
| Low (1 transactions per second) |  |

|  |  |
| --- | --- |
| Volumetrics (peak).  (e.g., Transaction rates, Inquiries) - What is the PEAK level of transactions / second that are expected in this application/system? | Place "X" beside the item that applies |
| High (>10 transactions per second) |  |
| Medium (1-10 transactions per second) |  |
| Low (<1 transactions per second) |  |

|  |  |
| --- | --- |
| Hours of Operation.  What is the required time frame for your application? In other words, when does this application need to be available for normal use? Please include Time Zone in your answer. | Place "X" and time zone beside the item that applies |
| 7 x 24 Continuous | X |
| 7 x 2X Nearly Continuous |  |
| Other - please describe (i.e. Weekdays, Monday through Friday 7am –7pm EST) |  |

|  |  |
| --- | --- |
| Recoverability.  What is the maximum outage time the application is willing to incur (Not a true disaster situation)? | Place "X" beside the item that applies |
| Near Real Time – 0-2 hours |  |
| High – 4-8 hours |  |
| Medium - within 24 hours (if possible) |  |
| Low - over 24 hours + |  |

|  |  |
| --- | --- |
| Desired Availability.  % (based on 24 x 7) -What are the requirements for your application/system to be available and functional to end users? | Place "X" beside the item that applies |
| $$$$ Very High Availability (99.9% and up, less than 16 outages/year, less than 30 minutes avg. recovery time) |  |
| $$$ High Availability (99.5% and up, less than 43 outages/year, less than 50 minutes avg. recovery time) |  |
| $$ Availability is not a business driver- can be less than 99.5% |  |

# 

# Contributors

|  |  |  |
| --- | --- | --- |
| Role | Name | Organization |
| Key Customer Contact(s) |  |  |
| Business SME |  |  |
| Project Manager |  |  |
| Backup Project Manager |  |  |
| Business Architect |  |  |
| Business Analyst | Anand Singh |  |
| System Analyst |  |  |
| Data Architect |  |  |
| Application Architect | Mayank Gupta |  |
| System Area SME | Avnish Gupta |  |
| Developers |  |  |
| QA Testers |  |  |

# Terms and Abbreviations

The table below provides a central location for all terms and abbreviations used throughout this document. These terms may include legal, technical or specific operational/business terms, which need to be clarified, confirmed or described to the document’s audience.

|  |  |
| --- | --- |
| Term / Abbreviation | Meaning |
| RFDS | Real time Fraud Detection Service |
| RPX | Real time Payments |
| BEX | Business Event Processing |
|  |  |
|  |  |

# Appendix

## Field list for Fraud Scan – List of relevant fields for fraud check on pacs.008, pain.013& camt.056 messages.



## RFDS – Process Flow Diagram



## Payment Engine Fraud Processing – Direct Clients



## Private Label Bank FRD

<https://myshare.bnymellon.net/sites/CTSProjectRepository2015/PPMProjects1/7121363/Project%20Documents/Requirements/RTP%20Private%20Label%20FRD.docx>