THETARAY

Version 6.14

Financial Crime Compliance Suite

APPLICATION INTEGRATION

Interfaces (AII)

Revision A

Publication Date: 15th July, 2025

1. Transaction Monitoring

1.1. Data Ingestion

1.1.1. Overview

ThetaRay's Transaction Monitoring solution processes and analyses data that includes transactional, account / KYC information and auxiliary reference data. The analysis process results in alerts that can be delivered to ThetaRay's Investigation Center, or to external targets.

The following sections detail the APIs / integration points that enable customers to feed the system with data using RESTful APIs / file upload and consume alerts generated by the platform by implementing HTTP based web hooks.

Moreover, Monitoring Reports enable retrieval of status of critical phases in the data processing flow including file upload acknowledgments and reporting on alerts generation.

1.1.2. File Ingestion

As part of the 314(a) request from FinCEN, financial institutions receive lists of entities in various file formats.

Some files may include only individuals, others only businesses, and in some cases, both are combined in a single file.
Our system is designed to accept two separate CSV files, one containing individuals and the other containing businesses, ensuring a structured and streamlined ingestion process, with files uploaded every day.

For more details on file ingestion, please refer to the technical document "Application Integration Interfaces (AII)". The AII document provides comprehensive guidance on how to structure, upload, and integrate various data sources into the system.

File Handling:

- New File(s) Received: When new files are uploaded, they are processed by the system. The status of each file is tracked, and after successful ingestion, the files are moved to an archive folder to prevent reprocessing. The system logs the upload and ingestion results.
- No New File(s): If no new files are received on a given day, no further action is taken, and the previous day's data remains active in the system.

This process ensures that files are handled consistently, and statuses are clearly tracked and auditable.

1.1.2.1. General Information

The ThetaRay environment exposes an AWS S3 protocol compatible endpoint allowing files consisting of transactional (payments, transactions) data, customer information and reference data to be fed into the system.

Files uploaded through the above endpoint should conform to the format specified within the relevant solution implementation.

ThetaRay supports the following formats:

- Delimited Text File (e.g. CSV), that optionally include a header line
- RJE formatted files consisting of SWIFT MT messages or XML formatted files for ISO 20022 / SWIFT MX / SEPA messages

Onboarding Process

As part of the onboarding process of an implementation the following information will be provided by ThetaRay:

- S3 endoint – a URL pointing to the HTTPS endpoint exposed by ThetaRay typically in the format:
  - 'minio-<suffix><tenant>.thetaray.cloud'

S3 access key / S3 secret key – authentication credentials

Bucket name

List of S3 paths into which each type of data should be uploaded to. A dedicated folder is assigned for each type of data that ThetaRay is expected to be uploaded into the ThetaRay system.

These locations are required to be in the format:

```
'/datasources/<datasource Name>
```

Files uploaded into the above locations will be asynchronously handled by ThetaRay, during this process files will be deleted from the landing area and will be retained in the internal storage of the ThetaRay environment.

Multiple technologies / frameworks can be used to connect to the above endpoint.

This includes:

- AWS S3 SDK for multiple languages including the AWS Java SDK
- BOTO3 for Python
- AWS CLI
- Minio 'mc' command line tool
- File synchronization tools supporting the S3 API such as 'rclone'

During integration, the S3 HTTPS endpoint provided by ThetaRay should be set into order to ensure that the ThetaRay environment is used as the target for file upload operations.

Example: Uploading Files - Script

A sample Python script using the Boto3 library is provided below.

```
import boto3
boto_session = boto3.Session(aws_access_key_id=Settings.S3_ACCESS_KEY,
                             aws_secret_access_key=Settings.S3_SECRET_
KEY)
s3_client = boto_session.client("s3", endpoint_url=Settings.S3_ENDPOINT)
with open('/path/to/file', 'rb') as fp:
  s3_client.upload_fileobj(fp, Settings.S3_PUBLIC_BUCKET,
                           os.path.join(Settings.S3_DATASOURCES_PATH,
                           connector, source))
```

Encrypted File Upload

The ThetaRay system supports receipt of encrypted files ensuring that data at rest is encrypted at the application level with no intermediate representations stored in plain form. The mechanism relies on an asymmetric Key Encryption Key which is stored in the Azure Key Vault (either by ThetaRay or customer managed). Per-file Data Encryption Keys are transferred along with the file content and are also encrypted using the Key Encryption Key.

The mechanism is described in the following procedure.

To upload an encrypted file:

1. An RSA 2048 Key should be established within the Azure Key Vault. Iif the Azure Key Vault is owned by the customer, Active Directory credentials should be set up by the customer to allow ThetaRay to access the Key Vault. ThetaRay should be able to performWrap/Unwrap key operations on the RSA 2048 Key within the Key Vault.

   Active Directory Credentials should be included: Tenant_id, client_id and either a secret_key or a certificate depending on the authentication method used. These credentials along with a reference to the used Key in the Azure Key Vault should be provided to ThetaRay, as part of the integration establishment process for a customer managed Azure Key Vault setup.
2. Prior to transferring a file to ThetaRay the client should:
   a. Generate a random 32-byte data encryption key for the file.
   b. Encrypt the data encryption key by invoking the 'Wrap Key' operation of Azure Key Vault.

c. Generate a JSON file, that consists of a reference to the Azure Key Vault key version used to encrypt the data encryption key (key_id entry) and a base64 version of the encrypted data encryption key (encrypted key_ entry).

Here is an example Python snippet that uses the Azure Key Vault Python SDK to perform the above operations:

```
akv_key_client = KeyClient(vault_url=kv_uri, credential=credential)

Tech Doc- Application Integration Interfaces – Fixes (For 6.7)

akv_key = akv_key_client.get_key(&amp;quot;pii-enc2&amp;quot;)

crypto_client = CryptographyClient(akv_key, credential=credential)

wrapped_enc_key = crypto_client.wrap_key(KeyWrapAlgorithm.rsa_oaep,
enc_key)

wrapped_key = {
&amp;quot;key_id&amp;quot;: wrapped_enc_key.key_id,
&amp;quot;encrypted_key&amp;quot;:
base64.b64encode(wrapped_enc_key.encrypted_key).decode(&amp;#39;utf-8&amp;#39;)
}
```

A sample JSON file follows:

```
{&amp;quot;key_id&amp;quot;:
&amp;quot;https://piiencryption.vault.azure.net/keys/pii-
enc2/f595aad6ead84434a16e72558418ee84&amp;quot;, &amp;quot;encrypted_key&amp;quot;:
&amp;quot;WrslQMYRkq8JUEYYxQhbFsX9der+cmJdwYVXXQhdzzV+ndWo5wBkSH3r7IJiqCI5N10
GAui4qW
KaeC03ZXl8lcV0J367LCnJvLSnlpzA05DplXKXs08miP5dapB+9sx1O8TMK7frd36Vhh4dqxL
VLSTJv
kFtgQe3WIfck/isrQFQNEZNMuSolG8JVJOAPZ5J9mOT4PupRb29swAUsunBdQgajUsr7dltHy
uaDlC
XPejo1a7w8QsIvscS8koGhLtiLN7lcVn4QGS1PDiJ3fl4ARlD2X8wyE06qv38UHsDcHWY/vL2
NaWP4
YAwWE5aNHkfJMtbr18cpo4HWRk3+G2gkg==&amp;quot;}
```

It should be noted that the Key Encryption Key can be freely rotated and a new version of the key can be created. ThetaRay uses the key_id as a reference to the specific version of the key in Azure Key Vault to decrypt the Data Encryption Key using the Unwrap Key Azure Key Vault operation.

3. Files should be encrypted using AES256 in GCM mode using the Data Encryption Key generated in the previous step. A random 16-byte Initialization Vector should be used and is required to appear as the

prefix of the file content before any data. A 16-byte verification tag (part of the AES algorithm) should be located at the end of the file.

4. When transferring data to a ThetaRay S3 compatible bucket a pair of files are expected to be delivered:

A JSON file consisting of the encrypted Data Encryption Key (Step 2 above). The file should have the same name as the data file itself with an additional .DEK suffix. (If needed, a different suffix can be used through settings on the ThetaRay platform.)

The encrypted data file (Step 3 above).

Security Considerations

- All data is transferred over HTTPS using S3 compatible APIs or REST APIs. Authentication -
- Application Integration
  - For S3 based file upload authentication is performed through access key / secret granted by ThetaRay.
  - Access to REST APIs for data ingestion is granted through a JWT bearer token that is retrieved by authenticating using ThetaRay granted credentials.
- ThetaRay provided credentials can be rotated on demand by issuing a request to ThetaRay's support.

1.1.2.2. SWIFT MT RJE

The ThetaRay Transaction Monitoring Solution is designed to optionally accept SWIFT MT transactions for Correspondent Banking use cases. The transactions should be sent across in RJE files containing multiple MT messages, according to SWIFT official guidelines.

1.1.2.3. ISO 20022 XML

ISO 20022 File Specifications

The ISO 20022 XML connector accepts the following message types:

CBPR+

- pacs.008.001.08
- pacs.009.001.08 Core
- pacs.009.001.08 COV
- pacs.009.001.08 ADV

Base MX

- camt.056.001.08
- pacs.002.001.10
- pacs.003.001.02
- pacs.003.001.08
- pacs.004.001.09
- pacs.007.001.09
- pacs.008.001.02
- pacs.008.001.10
- pacs.009.001.08 Core
- pacs.009.001.08 COV
- pain.001.001.03

The connector is designed to accept multiple messages in a single XML file. The XML file is parsed into individual messages, and then translated into a ThetaRay specific format designed to fit Transaction Monitoring features. The following specifications apply:

1. The MX messages need to be separated by a custom delimiter:
2. Each MX message is required to adhere to the correct XML format, and should not be wrapped in quotation marks or similar.

1.1.2.4. SWIFT MX File Specifications

Note: Not for use for new implementations. For new implementations please use ISO 20022 XML connector only.

1.1.2.4.1. Overview

In order to comply with CBPR+ requirements for the ISO 20022 coexistence period, ThetaRay has developed a custom connector to allow for the ingestion of SWIFT MX messages into the Transaction Monitoring product.

The connector is designed to accept multiple messages in a single XML file, according to the specification detailed in the following section. The XML file is parsed into individual messages, and then translated into a ThetaRay specific format designed to fit Transaction Monitoring features.

1.1.2.4.2. File Specifications

The Thetaray connector accepts multiple MX messages in a single XML file according to the following guidelines:

1. The MX messages need to be separated by a custom delimiter:

```
<!--end_of_message-->
```

2. Each message needs to contain a SWIFT Business Header (head.001.001.02) and then either a pacs.008 or pacs.009 message, between each delimiter.
3. Each MX message requires to be wrapped in a <Body> envelope.
4. Each MX message is required to adhere to the correct XML format, and should not be wrapped in quotation marks or similar.

File Example

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Body>
    <head:AppHdr xmlns:Saa="urn:swift:saa:xsd:saa.2.0"
xmlns:Sw="urn:swift:snl:ns.Sw" xmlns:SwGbl="urn:swift:snl:ns.SwGbl"
xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwSec="urn:swift:snl:ns.SwSec"
xmlns:head="urn:iso:std:iso:20022:tech:xsd:head.001.001.02">
        <head:Fr>
            <head:FIId>
                <head:FinInstnId>
                    <head:BICFI>AAAABBCC123</head:BICFI>
                </head:FinInstnId>
            </head:FIId>
        </head:Fr>
        <head:To>
            <head:FIId>
                <head:FinInstnId>
                    <head:BICFI>AAAABBCC123</head:BICFI>
                </head:FinInstnId>
            </head:FIId>
        </head:To>
        <head:BizMsgIdr>ZD81340TI4393289</head:BizMsgIdr>
        <head:MsgDefIdr>pacs.008.001.08</head:MsgDefIdr>
        <head:BizSvc>swift.cbprplus.stp.02</head:BizSvc>
        <head:CreDt>2022-12-06T08:32:21+01:00</head:CreDt>
    </head:AppHdr>
    <pacs:Document xmlns:Saa="urn:swift:saa:xsd:saa.2.0"
xmlns:Sw="urn:swift:snl:ns.Sw" xmlns:SwGbl="urn:swift:snl:ns.SwGbl"
xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwSec="urn:swift:snl:ns.SwSec"
xmlns:pacs="urn:iso:std:iso:20022:tech:xsd:pacs.008.001.08">
        <pacs:FIToFICstmrCdtTrf>
            <pacs:GrpHdr>
                <pacs:MsgId>ZD81340TI4393290</pacs:MsgId>
           <pacs:CreDtTm>2022-12-06T08:32:21+01:00</pacs:CreDtTm>
```

```
                <pacs:NbOfTxs>1</pacs:NbOfTxs>
                <pacs:SttlmInf>
                    <pacs:SttlmMtd>INDA</pacs:SttlmMtd>
                </pacs:SttlmInf>
            </pacs:GrpHdr>
            <pacs:CdtTrfTxInf>
                <pacs:PmtId>
                    <pacs:InstrId>ZD81340TI4393290</pacs:InstrId>
                  <pacs:EndToEndId>NOTPROVIDED</pacs:EndToEndId>
                    <pacs:TxId>ZD81340TI4393290</pacs:TxId>          <pacs:
UETR>2d82c3c7-68fb-476b-9243-064cb5d11901</pacs:UETR>
                </pacs:PmtId>
                <pacs:PmtTpInf>
                    <pacs:InstrPrty>NORM</pacs:InstrPrty>
                    <pacs:SvcLvl>
                        <pacs:Cd>G001</pacs:Cd>
                    </pacs:SvcLvl>
                </pacs:PmtTpInf>
                <pacs:IntrBkSttlmAmt Ccy="EUR">500.0</pacs:IntrBkSttlmAmt>
              <pacs:IntrBkSttlmDt>2022-12-06</pacs:IntrBkSttlmDt>
                <pacs:SttlmPrty>NORM</pacs:SttlmPrty>
                <pacs:InstdAmt Ccy="EUR">546.0</pacs:InstdAmt>
                <pacs:ChrgBr>DEBT</pacs:ChrgBr>
                <pacs:ChrgsInf>
                    <pacs:Amt Ccy="EUR">4.0</pacs:Amt>
                    <pacs:Agt>
                        <pacs:FinInstnId>
                            <pacs:BICFI>AAAABBCC123</pacs:BICFI>
                        </pacs:FinInstnId>
                    </pacs:Agt>
                </pacs:ChrgsInf>
                <pacs:InstgAgt>
                    <pacs:FinInstnId>
                        <pacs:BICFI>AAAABBCC123</pacs:BICFI>
                    </pacs:FinInstnId>
                </pacs:InstgAgt>
                <pacs:InstdAgt>
                    <pacs:FinInstnId>
                        <pacs:BICFI>AAAABBCC123</pacs:BICFI>
                    </pacs:FinInstnId>
                </pacs:InstdAgt>
                <pacs:Dbtr>
                    <pacs:Nm>Debtor Name</pacs:Nm>
                    <pacs:PstlAdr>
                        <pacs:AdrLine>16775 street address</pacs:AdrLine>
                    </pacs:PstlAdr>
                </pacs:Dbtr>
```

```
                <pacs:DbtrAcct>
                    <pacs:Id>
                      <pacs:IBAN>AT483200000012345864</pacs:IBAN>
                    </pacs:Id>
                </pacs:DbtrAcct>
                <pacs:DbtrAgt>
                    <pacs:FinInstnId>
                        <pacs:BICFI>AAAABBCC123</pacs:BICFI>
                    </pacs:FinInstnId>
                </pacs:DbtrAgt>
                <pacs:CdtrAgt>
                    <pacs:FinInstnId>
                        <pacs:BICFI>AAAABBCC123</pacs:BICFI>
                    </pacs:FinInstnId>
                </pacs:CdtrAgt>
                <pacs:Cdtr>
                    <pacs:Nm>SOMEBODY INC</pacs:Nm>
                    <pacs:PstlAdr>
                        <pacs:AdrLine>SOMEWHERE 33</pacs:AdrLine>
                        <pacs:AdrLine>SOMEWHERE 33</pacs:AdrLine>
                    </pacs:PstlAdr>
                </pacs:Cdtr>
                <pacs:CdtrAcct>
                    <pacs:Id>
                      <pacs:IBAN>AT483200000012345864</pacs:IBAN>
                    </pacs:Id>
                </pacs:CdtrAcct>
                <pacs:RmtInf>
                    <pacs:Ustrd>UNSTRUCTURED REMITTANCE
INFORMATION</pacs:Ustrd>
                </pacs:RmtInf>
            </pacs:CdtTrfTxInf>
        </pacs:FIToFICstmrCdtTrf>
    </pacs:Document>
</Body>
<!--end_of_message-->
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Body>
<head:AppHdr xmlns:Saa="urn:swift:saa:xsd:saa.2.0"
xmlns:Sw="urn:swift:snl:ns.Sw" xmlns:SwGbl="urn:swift:snl:ns.SwGbl"
xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwSec="urn:swift:snl:ns.SwSec"
xmlns:head="urn:iso:std:iso:20022:tech:xsd:head.001.001.02">
  <head:Fr>
    <head:FIId>
      <head:FinInstnId>
        <head:BICFI>AAAABBCC123</head:BICFI>
      </head:FinInstnId>
```

```
    </head:FIId>
  </head:Fr>
  <head:To>
    <head:FIId>
      <head:FinInstnId>
        <head:BICFI>AAAABBCC123</head:BICFI>
      </head:FinInstnId>
    </head:FIId>
  </head:To>
  <head:BizMsgIdr>ZD81340TI4393289</head:BizMsgIdr>
  <head:MsgDefIdr>pacs.008.001.08</head:MsgDefIdr>
  <head:BizSvc>swift.cbprplus.stp.02</head:BizSvc>
  <head:CreDt>2022-12-06T08:32:21+01:00</head:CreDt>
</head:AppHdr>
<pacs:Document xmlns:Saa="urn:swift:saa:xsd:saa.2.0"
xmlns:Sw="urn:swift:snl:ns.Sw" xmlns:SwGbl="urn:swift:snl:ns.SwGbl"
xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwSec="urn:swift:snl:ns.SwSec"
xmlns:pacs="urn:iso:std:iso:20022:tech:xsd:pacs.008.001.08">
  <pacs:FIToFICstmrCdtTrf>
    <pacs:GrpHdr>
      <pacs:MsgId>ZD81340TI4393296</pacs:MsgId>
      <pacs:CreDtTm>2022-12-06T08:32:21+01:00</pacs:CreDtTm>
      <pacs:NbOfTxs>1</pacs:NbOfTxs>
      <pacs:SttlmInf>
        <pacs:SttlmMtd>INDA</pacs:SttlmMtd>
      </pacs:SttlmInf>
    </pacs:GrpHdr>
    <pacs:CdtTrfTxInf>
      <pacs:PmtId>
        <pacs:InstrId>ZD81340TI4393294</pacs:InstrId>
        <pacs:EndToEndId>NOTPROVIDED</pacs:EndToEndId>
        <pacs:TxId>ZD81340TI4393294</pacs:TxId>          <pacs:UETR>2d82c3c
7-68fb-476b-9243-064cb5d11901</pacs:UETR>
      </pacs:PmtId>
      <pacs:PmtTpInf>
        <pacs:InstrPrty>NORM</pacs:InstrPrty>
        <pacs:SvcLvl>
          <pacs:Cd>G001</pacs:Cd>
        </pacs:SvcLvl>
      </pacs:PmtTpInf>
      <pacs:IntrBkSttlmAmt Ccy="EUR">500.0</pacs:IntrBkSttlmAmt>
      <pacs:IntrBkSttlmDt>2022-12-06</pacs:IntrBkSttlmDt>
      <pacs:SttlmPrty>NORM</pacs:SttlmPrty>
      <pacs:InstdAmt Ccy="EUR">546.0</pacs:InstdAmt>
      <pacs:ChrgBr>DEBT</pacs:ChrgBr>
      <pacs:ChrgsInf>
        <pacs:Amt Ccy="EUR">4.0</pacs:Amt>
```

```
    <pacs:Agt>
      <pacs:FinInstnId>
        <pacs:BICFI>AAAABBCC123</pacs:BICFI>
      </pacs:FinInstnId>
    </pacs:Agt>
  </pacs:ChrgsInf>
  <pacs:InstgAgt>
    <pacs:FinInstnId>
      <pacs:BICFI>AAAABBCC123</pacs:BICFI>
    </pacs:FinInstnId>
  </pacs:InstgAgt>
  <pacs:InstdAgt>
    <pacs:FinInstnId>
      <pacs:BICFI>AAAABBCC123</pacs:BICFI>
    </pacs:FinInstnId>
  </pacs:InstdAgt>
  <pacs:Dbtr>
    <pacs:Nm>Debtor Name</pacs:Nm>
    <pacs:PstlAdr>
      <pacs:AdrLine>16775 street address</pacs:AdrLine>
    </pacs:PstlAdr>
  </pacs:Dbtr>
  <pacs:DbtrAcct>
    <pacs:Id>
      <pacs:IBAN>AT483200000012345864</pacs:IBAN>
    </pacs:Id>
  </pacs:DbtrAcct>
  <pacs:DbtrAgt>
    <pacs:FinInstnId>
      <pacs:BICFI>AAAABBCC123</pacs:BICFI>
    </pacs:FinInstnId>
  </pacs:DbtrAgt>
  <pacs:CdtrAgt>
    <pacs:FinInstnId>
      <pacs:BICFI>AAAABBCC123</pacs:BICFI>
    </pacs:FinInstnId>
  </pacs:CdtrAgt>
  <pacs:Cdtr>
    <pacs:Nm>SOMEBODY INC</pacs:Nm>
    <pacs:PstlAdr>
      <pacs:AdrLine>SOMEWHERE 33</pacs:AdrLine>
      <pacs:AdrLine>SOMEWHERE 33</pacs:AdrLine>
    </pacs:PstlAdr>
  </pacs:Cdtr>
  <pacs:CdtrAcct>
    <pacs:Id>
      <pacs:IBAN>AT483200000012345864</pacs:IBAN>
```

```
        </pacs:Id>
      </pacs:CdtrAcct>
      <pacs:RmtInf>
        <pacs:Ustrd>UNSTRUCTURED REMITTANCE INFORMATION</pacs:Ustrd>
      </pacs:RmtInf>
    </pacs:CdtTrfTxInf>
  </pacs:FIToFICstmrCdtTrf>
</pacs:Document>
</Body>
```

1.1.2.5. SEPA

Note: Disclaimer - Be aware that the method described in this section although maybe in use for existing customers it is not recommended for new customers of ThetaRay

The ThetaRay Transaction Monitoring Solution is designed to optionally accept SEPA messages for European Domestic or Retail use cases. The transactions should be sent across as XML files, using the ThetaRay custom delimiter described in the ISO 20022 XML section of this document.

1.1.3. Data Ingestion Service - (REST API)

The Data Ingestion Service enables transaction ingestion for Transaction Monitoring via JSON API. The service is less relevant for sending bulk historical data, where the available file CSV option is better suited. The JSON is custom built and can accept any data form. In order to use this option, please align with Customer Support.

The Data Ingestion API enables remote clients to stream data into the system asynchronously. The concrete data types supported by the API are solution specific and provided as part of the project setup phase. The Solution processes and analyses the ingested data asynchronously before delivering the alerts through an independent Alert Distribution API in a push manner.

Also provided are the concrete solution name, hostnames, authentication credentials, connector IDs, and JSON structures as part of the project setup.

Endpoint	Description
auth	Authenticates with the Platform host. Sends data records to the system.
ingestion	Authenticates with the Platform host. Send data records to the system.

Endpoint: auth

Authenticates with the platform host.

HTTP request

```
POST https://gateway-<shared ns>.<host-domain-name>/auth
```

HTTP Headers

```
Content-Type = application/json
```

Request Body

The request body contains data with the following structure:

JSON Representation

```
{
  "clientId": "platform-ingestion-<solution>",
  "clientSecret": "<credentials provided by the System>"
  }
```

Fields	
clientId	Name of the data consumer (the requesting side) as provided by the system administrators.
clientSecret	Authentication credentials as provided by Support.

Response Body

If successful, the response body contains data with the following structure:

JSON Representation

```
{
  "result": {    "token": "<JWT>",
  "expirationInMinutes": 5
  }
}
```

Fields	
token	The bearer token authentication string. Paste in the request header Authorization field.
expirationInMinutes	The token validity period. After the time is expired a new authentication request is required.

Endpoint: Data Ingestion

Uploads data of a specific type into the System.

HTTP Request

```
POST https://gateway-<shared ns>.<host-domain-name>/ingestion/v1/{solution}/
{connector-id}
```

Path Parameters

Parameters	
solution	string Required. The name of the solution to which we want to append data (as provided during project setup).
connector-id	string Required. The type of data to be ingested by the system (as provided during project setup)

HTTP Headers

```
Content-Type = application/json; Authorization = Bearer <JWT returned by the
"auth endpoint">
```

Request Body

The request body contains data with a dynamic structure depending on the dataset. The JSON example below shows a dataset comprised of four columns/fields and two rows/records. The concrete supported data types are implementation-specific and provided as part of the project setup.

JSON Representation

```
[{
    "account_id": "3818",
    "district_id": "74",
    "frequency": "Monthly issuance",
    "date": "1993-01-01 00:00:00"
    },
    {
    "account_id": "1539",
    "district_id": "1",
    "frequency": "Issuance after transaction",
    "date": "1993-01-03 00:00:00"
}]
```

Fields	
Dynamic Fields	The request body JSON contains the data to be uploaded to the system.

HTTP Response Code

Code	Description
200	Request successfully processed (initial data validation was successful and data is

Code	Description
	stored in persistent storage for further processing)
401	Unauthorized to perform the request - credentials not valid
403	Request forbidden due to insufficient privileges
404	Endpoint does not exist
422	Unable to process the request due to validation issues (the payload includes a JSON object with further error details, but should be used only as for troubleshooting purposes and not as a formal API)
500	Internal error, server cannot process the request

1.1.4. MinIO - Automation of secret key Rotation API

The purpose of this API, is to assist customers in enhancing the security level of their Financial Crime Compliance Suite by being able to self initiate the process of automatic rotation of the secret key code, used for example to encrypt and protect data as it flows through the environment.

1.1.4.1. Requirements

tr-platform-api service

Endpoint: /config/rotate-s3-creds.

Request

Element	Detail
HTTP Method	Post
Request Body	{ "solution": "sol", "accessKey": "accesskey", "secretKey": "secretkey" }

Response

Element	Detail
Response body:	{ "newAccessKey": "newaccesskey", "newSecretKey": "newsecretkey" }

Response Statuses

Element	Detail
200	success
401	response if credentials not valid
403	response if key can't be rotated
404	response for solution not found
500	response for unexpected issues

Audit

The following fields will be sent to audit in Opensearch:

- message = "Minio access key rotation was attempted"
- category = "Minio configuration
- action = "secret_key_rotation"
- component = "tr-platform-api"
- instance = <solution from the request>
- outcome = failure | success
- created = datetime.now()
- extra:
  - old_accesskey = <accesskey from the request>
  - new_accesskey = <new accesskey>

1.1.5. Monitoring Reports

1.1.5.1. Overview

The ThetaRay Platform automatically generates three monitoring reports at key points in the life cycle of the evaluation process as follows:

- **Data Upload** is generated automatically whenever a data upload task is completed. The report provides a view into the number of data records uploaded successfully into the evaluation process versus the number of data records rejected due to errors.
- **Alert Volume** is generated automatically whenever the data distribution task is completed. The report provides a view into the total number of records analyzed in the evaluation process versus the number of records found to match alert conditions (alerted activities).
- **Model Update** is generated automatically whenever a model update or version change is merged with the operational branch in the git repository. The report provides a view of the model update.

How to Access Monitoring Reports?

- The ThetaRay platform generates Monitoring Reports as part of the analysis flow and on changes to configuration or models. Reports are saved in ThetaRay's internal Object Storage which is accessible through an S3 compatible interface. Reports are formatted as JSON files and are stored in the 'reports' folder within the public buckets. Details bucket name and S3 endpoint URL are provided by ThetaRay as part of the project setup.
- Optionally, the ThetaRay environment can be setup to deliver a notification on report creation. The notification is delivered over HTTPS to an endpoint exposed by the customer - registration of the notification endpoint is part of the project setup. From a security perspective, the notification supports remote endpoints protected using HTTP Basic Authentication - username /password credentials will be to be provided to ThetaRay as part of the integration setup.

Enable Monitoring Reports in your Deployment

For information on how to enable Monitoring Reports in your deployment please refer to Chapter 14 of the current ***Platform User Guide***.

Define New External Notification Target

Under domains/[DOMAIN_NAME]/targets/

```
from thetaray.api.solution.external_notification_target import
ExternalNotificationTarget, TargetAuth, AuthenticationMode
def external_notification_target():
    return ExternalNotificationTarget(
        identifier="reports_notifications",
        name="reports_notifications",
        url="[ANY_URL]",
        verify=False,
        auth=TargetAuth(mode=AuthenticationMode.BASIC_AUTH,"configuration": {
    "username": <username>,
    "password": <password>
  })
    )
def entities():
    return [external_notification_target()]
```

Note: If the environment was upgraded (and not a new deployment), there is no targets folder, and the user is required to add it by themselves.

Report Notification JSON Representation Code

```
{
"event": string,
"s3_endpoint": string,
"path": string,
"bucket": string,
"timestamp": string
}
```

Fields	Description
event	enum (string) The report type: DataUploadAcknowledgement: The Data upload monitoring report TrackingOfAlertVolumesGeneratedAndSent: The Alert volume monitoring report TrackingOfOperationalModelVersionChanges: The Model updates monitoring report
s3_endpoint	string The MinIO URL
path	string The path to the report inside the MinIO bucket
bucket	string The MinIO bucket identifier
timestamp	string Notification time stamp.

1.1.5.1.1. Notification Example

```
{
    "event":"DataUploadAcknowledgement",
    "s3_endpoint":"<endpoint URL>"
    "path":"reports/DataUploadAcknowledgement_2022_09_03_07_24_17.json",
    "bucket": "thetaray-public-<platform name>",
    "timestamp": "2022-09-03T07:24:17.252121"
}
```

Data Upload Acknowledgment Report

- **Report file name**: DataUploadAcknowledgement_{connector_identifier}_{date_time}.json
- **Connector_identifier** - Indicates the file type uploaded (transactions, KYC)

The Data Upload report is designed to track the ThetaRay Platform's automatic data ingestion pipeline in which data is uploaded from data source files into the

evaluation process input datasets. The report specifies for each Data Upload task how many data records were successfully uploaded, how many failed to upload, and for what reasons.

ThetaRay Platform automatically generates the report after the Data Upload task is completed. The following reference provides a specification of the structure and fields comprising the Data Upload Monitoring Report.

1.1.5.1.2. Example - Data Upload Reference Report

The following is an example of a *DataUpload reference* report in JSON format:

```
{
  "type": "Upload",
  "connector": "card",
  "datetime": "2022-09-01 12:33:09",
  "files_count": 1,
  "datasets": [
    {
      "dataset": "card",
      "total_row_count": 892,
      "total_invalid_rows_count": 2,
      "file_details": [
        {
          "filename": <filename.csv>",
          "rows_count": 892,
          "invalid_rows_count": 2,
          "error_details": [
            {
              "record_identifier": 201,
              "corrupted_row": {
                "Rejection type": "WARN",
                "Invalid row": "{452728, 1221, 19920103, credit, credit in
cash, 600.0, 600.0, null, null, null}",
                "Warning reason": "Date shouldn't be less then 1993",
                "Rejected field": "date",
                "Rejected value": "19920103",
                "Source filename": "<filename.csv>"
                },
              "rejection reason": null
            },
            {
              "record_identifier": 6,
              "corrupted_row": null,
              "rejection reason": [
                "test_id_validator",
                "record_identifier_validator"
```

```
              ]
            }
          ]
        }
      ]
    }
  ],
  "commit_metadata": {
    "commit_hash": "b7242f5834f6ba9b8332aa5774f3d3a6b8154f7e",
    "commit_message": "report on",
    "platform_version": "MVP-30671.9878"
  }
}
```

JSON Representation

```
{
"type": string,
"connector": string,"datetime": string,
"files_count": int,"datasets": [{object (Dataset)],
"commit_metadata":{object (commit_metadata)}
}
```

Fields	Description
Type	string The report type (Upload).
datetime string <date-time>	The date and time of the report (%Y-%m-%d %H:%M:%S).
connector	string The identifier of the ThetaRay Platform connector that uploaded the data.
files_count	int The number of files uploaded in the current batch.
dataset[]	object (dataset) The datasets to which data is uploaded in the current batch (i.e. the target datasets).
commit_metadata	object (commit_metadata) The git commit details of the code version that was run to execute the evaluation of the current data batch.

Example

```
{
```

```
"type": "Upload",
"connector": "test_connector",
"datetime": "2022-05-05 10:30:30","files_count": 1,
"datasets": [. . .
]
"commit_metadata":{
. . .
 }
}
```

1.1.5.1.3. Dataset

JSON Representation

```
{
"dataset": string,
"total_row_count": int,
"total_invalid_rows_count": int,
"file_details": [
object (file_details)]
}
```

Field	Description
dataset	string The target dataset name.
total_row_count int	int Number of processed rows in the target dataset in the current batch.
total_invalid_ rows_count	string Number of invalid rows in the target dataset in the currect batch.
file_details[]	object (file_details) The details of the data source files from which data was uploaded to the target dataset in the current branch.

Example

```
{
"dataset": "test_datasource",

   "total_row_count": 2000,
"total_invalid_rows_count": 24,
"file_details": [
```

```
    . . .
  ]
```

1.1.5.1.4. file_details

JSON Representation

```
{
    "file_details": [{"filename": string,
    "rows_count": int,
    "invalid_rows_count": int,
    "error_details": [{object (error_details)
    }
  ]
 }
}
```

1.1.5.1.5. error_details

JSON Representation

```
{
    "record_identifier": string,
    "corrupted_row": {object (    corrupted_row)
    }
    ]"rejection_reason": [
    [string]
    ]
}
```

Field	Description
record_identifier	string The identifier of the invalid record.
corrupted_row	object (corrupted_row) The contents of the invalid row in JSON
rejection_reason[]	string The description of the errors invalidating the record.

Example

```
{
    "record_identifier": "id12346",
    "corrupted_row":{. . .},
    "rejection_reason": ["test_id_validator",
    "record_identifier_validator"]
}
```

Note: Please note that due to enhancement of the "corrupted_row" object, "rejection_reason" field will always be null. The field will remain in order to support backwards compatibility, but the content has moved to the "corrupted_row" object.

1.1.5.1.6. corrupted_row

JSON Representation

```
{
   "Rejection type": "REJECT" or "WARN",
   "Invalid row": string,
   "Rejection reason": string (applicable for Rejection type = REJECT),
   "Warning reason": string (applicable for Rejection type = WARN),
   "Rejected field": string,
   "Rejected value": string,
   "Expected data type": string,
   "Source filename": string
}
```

Field	Description
Rejection type	Contains either WARN or REJECT
Invalid row	The content of the invalid row from the file
Rejection reason	The description of the error invalidating the row in case of rejection type REJECT
Warning reason	The description of the error invalidating the row in case of rejection type WARN
Rejected field	The name of the field within the row that caused the rejection
Rejected value	The value of the rejected field
Source filename	The name of the file that contained the invalid row

Example

```
 "corrupted_row": {
                "Rejection type": "WARN",
                "Invalid row": "{452728, 1221, 19920103, credit, credit in
cash, 600.0, 600.0, null, null, null}",
```

```
                "Warning reason": "Date shouldn't be less then 1993",
                "Rejected field": "date",
                "Rejected value": "19920103",
                "Source filename": "<filename.csv>"
            },
```

Note: In order to enable custom warnings in the "Warning reason" field, a configuration must be made on the relevant dataset. If not configured, the warning reason will contain the condition in its raw format, for example, "substring(date, 1, 4) < 1993"
For more information, please refer to the Platform User Guide.

1.1.5.1.7. commit_metadata

JSON Representation

```
{
    "commit_hash": string,
    "commit_message": string,
    "platform_version": string
}
```

Field	Description
commit_hash .	string The hash key of the last commit of the code executed in the upload process
commit_ message	string The message of the last commit of the code executed in the upload process.
platform_ version	string The ThetaRay Platform version at the time of the last commit of the code executed in the upload process .

Example

```
{
"commit_hash": "fff48347c40b6676e7804caa85e3f1b5f44f933e",
"commit_message": "change to TR new algo ",
"platform_version": "MVP-30671.9878"
}
```

Alert Volume Report

- **Report file name**: TrackingOfAlertVolumesGeneratedAndSent_{EvFlow_Target}_DATE_TIME.json

The Alert Volume monitoring report is designed to track the Alert distribution task in which alert-triggering data records (or alerted activities) are distributed to case managers such as ThetaRay's Investigation Center. The report specifies how many data records were evaluated by the task versus how many of them were found to match the conditions of alerts. The report also provides identifying information for the process that evaluated the data batch and the committed code that was executed in the process.

ThetaRay Platform automatically generates the report after the Alert distribution task is completed. The following reference provides a specification of the structure and fields comprising the report.

1.1.5.1.8. Example - Alert Volumes Report

The following is an example of an *Alert Volumes* report in JSON format:


```
{
   "type":"Analysis",
   "datetime":"2022-08-08 13:31:55",
   "evaluation_flow":"tr_analysis",
   "distribution_tagret":"IC",
   "rows_analyzed":218,
   "unique_keys":193,
   "total_alerts":146,
   "distributed_alerts":144,
   "commit_metadata":{
      "commit_hash":"ecfb767dcf205db11e9990b834ab524cac25c626",
      "commit_message":"commit4",
      "platform_version": "MVP-30671.9878"
   },
   "model_versions":{
      "step_name":"tr_evaluation",
      "models":{
         "detection_model":{
            "name":"tr_detection_model",
            "tags":{
               "version":"release",
               "_commit_id":"2cb7f9a3a743e164cfedbb8b62ee25db28c97bdc"
            },
            "version":"1"
         },
         "feature_extraction_model":{
            "name":"tr_feature_extraction_model",
            "tags":{
```

```
                "version":"release",
                "_commit_id":"2cb7f9a3a743e164cfedbb8b62ee25db28c97bdc"
            },
            "version":"1"
        }
    }
  }
}
```

JSON Representation

```
JSON Representation {"type": string,"datetime": string,
"evaluation_flow": string,
"distribution_tagret": string,"rows_analyzed": int,"unique_keys": int,
"total_alerts": int,"distributed_alerts": int,
"commit_metadata": {object (commit_metadata)},
"model_versions": {object (model_versions)}}
```

Field	Description
type	string The report type (Analysis)
datetime	string <date-time> The date and time of the report (%Y-%m-%d %H:%M:%S).
evaluation_ flow	string The identifier of the evaluation flow triggering the alerts. The evaluation_flow identifier is defined on the ThetaRay Platform evaluating the data
distribution_ tagret	string The identifier of the client consuming the report. The target is defined on the ThetaRay Platform evaluating the data
rows_analyzed	int The number of row processed in the current evaluation batch
total_alerts	int The number of alerts detected in the current evaluation batch.
distributed_ alerts	int The number of alerts detected in the current evaluation batch not including the suppressed alerts.
unique_keys	int The number of rows with distinct primary keys in the current evaluation batch. Multiple rows can have the same primary keys when the the analyzed record matches the conditions of multiple risks
commit_ metadata	object (commit_metadata The commit identifiers of the code executed in the evaluation process.

Field	Description
models_ versions	object (model_versions) The AI models used in the alert generation process.

Example


```
{"type":"Analysis",
"datetime":"2022-08-08 13:31:55",
"evaluation_flow":"tr_analysis",
"distribution_tagret":"IC",
"rows_analyzed":218,"unique_keys":193,
"total_alerts":146,
"distributed_alerts":144,
"commit_metadata":{. . .},
"model_versions":{. . .}
}
```


1.1.5.1.9. commit_metadata

JSON Representation


```
{
"commit_hash": string,
"commit_message": string,
"platform_version": "MVP-30671.9878"
}
```


Field	Description
commit_hash	string The hash key of the commit of the code executed in the evaluation process.
commit_message	string The message of the git commit of the code executed in the evaluation process
platform_version	string The version of the ThetaRay Platform executing the evaluation process.

Example


```
{
"commit_hash": "fff48347c40b6676e7804caa85e3f1b5f44f933e",
"commit_message": "change yaml severity",
"platform_version": "MVP-30671.9878"
}
```


1.1.5.1.10. model_versions

JSON Representation

```
{
    "step_name": string,"models": {object (models)
 }
}
```

Field	Description
step_ name	string The identifier of the evaluation step in whose context the data batch is evaluated. The evaluation step is defined in the ThetaRay Platform.
models	object (models) The ML models that evaluated the data batch in the context of the evaluation step.

Example

```
{
 "step_name": "tr_evaluation","models": {}
}
```

1.1.5.1.11. models

JSON Representation

```
{
  "detection_model": {objectmodel)},
  "feature_extraction_model": {object (model)
  }
}
```

Field	Description
detection_model	object (model) The details of the detection model evaluating the data batch.
feature_extraction_model	object (model) The details of the feature extrtaction model evaluating the data batch.

Example

```
{
    "detection_model": {. . .},
    "feature_extraction_model": {. . .}
}
```

1.1.5.1.12. model

JSON Representation

```
{
    "name": string,"tags": {object (tag)},
    "version": string
}
```

Field	Description
name	string The name of the model in the ThetaRay Platform.
tags	object (tag) The identification tags of the model in the ThetaRay Platform.
version	string The model version in the ThetaRay Platform.

Example

```
{
    "name": " tr_detection_model","tags": {. . .},
    "version": "1"
}
```

1.1.5.1.13. tag

JSON Representation

```
{
    "version": string,"_commit_id": string,
}
```

Field	Description
version	string The model version number in the ThetaRay Platform.
_commit_id	string The commit identification number of the code version that was used to run the model training process in the ThetaRay Platform

Example

```
{
    "version": "release",
    "_commit_id": "a6686c93e888aa47ba3c1e77198846a813931a12"
}
```

Model Updates Report

- **Report file name**: TrackingOfOperationalModelVersionChanges_DATE_TIME.json

The Model update monitoring report is designed to track the changes and updates to the models saved in ThetaRay Platform's model repository. The report specifies the git commit details of the code changes applied to the models.

ThetaRay Platform automatically generates the report after the metadata sync process merges the committed model updates to the operational environment. The following reference provides a specification of the structure and fields comprising the report.

1.1.5.1.14. Example Model Updates Report

The following is an example of a *Model Updates* report in JSON format:

```
{
  "type": "Config Change",
  "datetime": "2023-11-05 09:52:40",
  "commit_metadata": {
    "commit_hash": "c16e307cbbc6554c0ac231475e1d8968b32d9282",
    "commit_message": "turn on monitoring report",
    "commit_details": [
      "settings/tr_settings.py"
    ],
    "platform_version": "mvp-40826.5"
  }
}
```

1.1.5.1.15. JSON Representation

```
{
    "type": string,
    "datetime": string,
    "commit_metadata": {objectcommit_metadata)
 }
}
```

Field	Description
type	string The report type (Config Change)
datetime	string <date-time> The date and time of the report (%Y-%m-%d %H:%M:%S)
commit_metadata	object (commit_metadata) The git commit details of the model code update.

Example

```
{
"type": "Config Change",
"datetime": "2022-09-01 12:18:44",
"commit_metadata": {. . .
  }
}
```

1.1.5.1.16. commit_metadata

JSON Representation

```
{
    "commit_hash": string,
    "commit_message": string,
    "commit_details": [string],"platform_version": string
}
```

Field	Description
commit_hash	string The hash identifying the last commit impacting the model code and version.
commit_message	string The message of the last commit impacting the model code and version.
commit_details[]	string The name and location of the commited files.
platform_version	string The ThetaRay Platform version at the time of the commit.

Example

```
{
"    commit_hash": "b7242f5834f6ba9b8332aa5774f3d3a6b8154f7e",
"commit_message": "report on","commit_details": ["settings/tr_settings.py"],
"platform_version": "MVP-30671.9878"
}
```

1.2. Alert Externalization & Management

Note: This section is meant specifically for customers using Transaction Monitoring.

1.2.1. Alert Distribution API

By default, alerted activities are distributed to ThetaRay's Investigation Center (the IC) at the end of the analysis pipeline. However, through custom implementation, they can be sent to external targets that are not part of the ThetaRay environment such as case management applications.

The Alert Distribution mechanism is based on an HTTPS request triggered by ThetaRay with a JSON payload consisting of the content of one or more Alerted Activities. Batch sizes can be tuned as part of the solution implementation.

From a security perspective, the system supports targets that use HTTP Basic Authentication and requires that a username and password will be provided to the platform implementers.

This document has two parts:

- The alert distribution JSON payload specification lays out an alerted activity structure, fields, and examples.
- Alert Distribution API

1.2.1.1. The alert distribution JSON payload specification

The ThetaRay Platform sends the alerted activities records as a JSON file to the distribution target. The JSON represents the alerted activities as an array containing the alertedActivity objects.

JSON Representation

```
[{

"alertedActivityId": string,"occurredOnDate": string,"occurredOnField":
string,"solutionId": string,"evaluationFlowId": string,
"dataPermission": string,
"primaryKeySet": {

 string: string
 ...

},

"evaluation": {
 object (evaluation)
},
risk: {
object (risk)
}

}]
```

Parameters	
alertedActivityId	**string** The unique ID of the alerted activity record. Multiple records can be sent with the same alertedActivityID in case multiple risks are identified for the given

Parameters	
	activity. In this case a different risk identifier will be associated with each of the alerted activities
occurredOnDate	**string** The timestamp of the activity that triggered the alert. This is a logical time assigned to the activity by the evaluation process – for example, the year and month associated with the alert.
occurredOnField	**string** The name of the column where the occurredOnField value is stored.
solutionId	**string** The unique name of the ThetaRay Platform instance that generated the alert.
evaluationFlowId	**string** The unique name of the evaluation flow that generated the alert.
dataPermission	For internal use only.
primaryKeySet	**map (key: string, value: string)** The unique identifier of the investigated entity (for example "account" or "customer").
evaluation	**object (** evaluation **)** The evaluation flow results.
risk	**object** (risk) The details of the risk that the activity match.

Example


```
"alertedActivityId": "1081d702-174e-467f-b9bb-781e358472a1","occurredOnDate":
"1997-05-22 00:00:00","occurredOnField": "date_
loan","solutionId":"artur26","evaluationFlowId": "tr_
analysis","dataPermission": "dpv:public","primaryKeySet": {

"account_id": 6649

},
"evaluation": {
```

```
...
},
"risk": {

...
}

}
```

1.2.1.1.1. evaluation

```
{
"features": [
{
 object (feature)
 }],
"evaluationSteps": {
 string:object (evaluation_steps)
},
"order": [
string
]
}
```

Field	
features[]	object (Alert Distribution API) The name and values of the features that served as input to the evaluation flow.(feature are data fields that are used either for analytical purposes or as forensic data)
	object(evaluation_steps) All of the evaluation steps comprising the evaluation flow. In this version there is only one such step - the TR-Algo step.
order[]	Foe internal use only

Example

```
{
"features": [...],
```

```
"evaluationSteps": {"tr_evaluation": {...}},"order": [ "tr_evaluation" ]
}
```

1.2.1.1.2. feature

JSON Representation

```
{
"identifier": string,"value": string
}
```

Mandatory Fields	
identifier	string The name of the feature that served as input to the evaluation flow.
Value	string The value of the feature that served as input to the evaluation flow.

Optional Fields	
display_ name	The display name of the feature (if configured to be included for the Alert Distribution Target)
category	The category of the feature (if configured to be included for the Alert Distribution Target)
description	The description of the feature (if configured to be included for the Alert Distrbution Target)

Example

```
"features": [
    {"identifier": "account_id",
    "value": 6649},{"identifier": "district_id_bank",
    "value": 1},
    {"identifier": "frequency",
    "value": "Monthly issuance"},...
]
```

1.2.1.1.3. evaluation_steps

JSON Representation

```
{
```

```
    "name": string,"identifier": string,
    "type": string,"fusionScore": float,"pattern": string,
    "features": [
     {  object (feature), }
    ]
    "model_metadata": {
     object (model_metadata)
 }
}
```

Fields	
name	string The evaluation step name.
Identifier	string The evaluation step unique identifier.
type	string The evaluation step type. In this version there is only one type – "TR Algo".
fusionScore	float The overall alert anomaly score between 0 and 1 calculated by the ThetaRay algorithm. A higher score indicates a higher anomaly level.
pattern	string A string comprised of the top ranking features separated by a hash (#) character. The number of features that comprises the string is set as part of the solution implementation.
features[]	object (Alert Distribution API) The rating and rank of all feature values as computed by the ThetaRay evaluation algorithm.
model_metadata	object (model_metadata) The name and ID details of the ML models analyzing the data.

Example

```
{
"tr_evaluation":{
"name": "Thetaray evaluation",
```

```
"identifier": "tr_evaluation",
"type": "TR Algo",
"fusionScore": 0.4623927275447511,
"pattern": "min3#min4#has_card",
"features": [ ...],
"model_metadata":{...}
}
}
```

1.2.1.1.4. feature

JSON Representation

```
{
    "identifier": string,
    "rating": int"rank": int

}
```

Fields	
identifier	string The feature name.
rating	int An integer (between 0 and 9999) that represents the feature's contribution to the overall fusion score relative to the other features.
rank	The position of the feature relative to other features when ordering features by rating.

Example

```
{
    "identifier": "account_id",
    "rating": null,"rank": null
}
```

Example


```
{
    "identifier": "account_id",
    "rating": null,
```

Example
"rank": null }

1.2.1.1.5. model_metadata

JSON Representation

```
{
    "detection_model": {object (model)},
    "feature_extraction_model": {object (model)
 }
}
```

Fields	
detection_model	object (model) Identification details of the anomaly detection model that are specified in the model repository.
feature _extraction_model	object (modell) Identification details of the feature extraction model that are specfied in the model repository.

Example

```
{
    "detection_model": {. . .},
    "feature_extraction_model": {. . .}
}
```

1.2.1.1.6. model

JSON Representation

```
{

    "name": string,
    "tags :{
     object (tag)},
     "version": string

}
```

Fields	
name	string The model name that is specified in the model repository.
tags	object (tag) The model identification details that are specified in the model repository.
version	string The model version that is specified in the model repository.

Example

```
{

    "name":"tradmin_tr_detection model","tags":{
     ...
    },
    "version": "1"
}
```

1.2.1.1.7. tag

JSON Representation

```
{
"version": string
"commit":string,
}
```

Fields	
version	string The model version that is specified in the model repository.
commit_id	string The commit identification number of the code version that was used to run the training process that generated the model.

Example

```
{"version": "release",
"_commit_id": "dirty_2022-06-27T14:28:23_
```

```
a6686c93e888aa47ba3c1e77198846a813931a12"
}
```

1.2.1.1.8. risk

JSON Representation

```
{
    "id": string,"name": string,
    "severity": int,"category": string,
    "suppressionPeriodUnit": string,
    "suppressionPeriodSize": int,
    "description": string,"condition": [{ object (condition)}
    ] ,
    "commit_metadata": {
    object (commit_metadata)
    },
    "global_trace": [ string
  ]
}
```

Fields	
id	string The risk unique ID.
_name	string The risk name.
severity	int A numeric value between 0 and 100 expressing the severity of the given risk type (set as part of the solution implementation)
category	string The category to which the risk is assigned
suppressionPeriodUnit	For internal use only
suppressionPeriodSize	For internal use only
description	string A description of the risk
conditions[]	object (condition) The risk conditions that were used to trigge the alert.
commit_metadata	object (commit_metadata) The Git commit identifier of the code version that was executed to run the evaluation flow that created the current alerted activity. The identifier is used to enable tracing the version of the solution implementation for auditing purposes.

Fields	
global_trace[]	string The identifiers of the transactions that were evaluated for the current investigated entity (identified by the primaryKeySet) in the current occurredOnDate period.

Example

```
{

    "id": "anomaly1",
        "name": "anomaly1",
        "severity": 90,
        "category": "default",
        "suppressionPeriodUnit": "DAY",
        "suppressionPeriodSize": -1,
        "description": "anomaly",
        "conditions": [{
           ...
        }],
        "commit_metadata": {
           ...    }
        "global_trace":
[      1962853,      1962899,      1962928,      1963016,
          1963056,
          3677396
     ]
  }
```

1.2.1.1.9. condition

JSON Representation

```
{
"identifier": string,"value": boolean,
"displayName": string,
"trace": [object (trace)

 ]
}
```

Fields	
identifier	string The condition unique identifier.
value	boolean Indicates whether the analyzed activity record was found to match the risk condition.
displayName	string The condition display name in the case manager UI.
trace[]	object Alert Distribution API) Transaction identifiers associated with features triggering the condition. For example, the transactions associated with top three trigger features in case of alerts generated by ThetaRay algorithms, or transactions associated with a feature that is above a threshold in case of scenario- based alert).

Example

```
[{
"identifier: "is_anomaly",
    "value: true,
    "displayName:": is_anomaly12 dispaly name",,
    "trace:[{
       ...
}]
  }]
```

1.2.1.1.10. trace

JSON Representation

```
{
"features": [ string],
"identifiers": [ string]
}
```

Fields	
features[]	
identifiers[]	string Identifiers of the transactions on the basis of which the

Fields	
	features were computed. string The names of the features that caused the activity record to match the risk condition.

Example

```
[{
    "features":["min3",
    "min4","min5"],
    "identifiers": [1962853,
    1962899,
    1962928,
    1963016,
    1963056,
    3677396]
}]
```

1.2.1.1.11. commit_metadata

JSON Representation

```
{
"commit_hash": string,
"commit_message": string
}
```

Fields	
commit_hash	string The commit identifier
identifiers[]	string The commit message

Example

```
{
"features": "dirty_2022-06-27T15:24:46_
a6686c93e888aa47ba3c1e77198846a813931a12","commit_message:"initial commit"
]
```

1.2.1.2. A full Example of an Alerted Activity JSON Representation


```
[{
"alertedActivityId": "1081d702-174e-467f-b9bb-781e358472a1","occurredOnDate":
"1997-05-22 00:00:00","occurredOnField": "date_loan","solutionId":
"artur26","evaluationFlowId": "tr_analysis","dataPermission":
"dpv:public","primaryKeySet": { "account_id": 6649},
"evaluation": {"features": [

{"identifier": "account_id","value": 6649},{"identifier": "district_id_
bank","value": 1},
. . .],"evaluationSteps": {"tr_evaluation": {

"name": "Thetaray evaluation","identifier": "tr_evaluation","type": "TR
Algo","fusionScore": 0.4623927275447511,"pattern":
"min3#min4#min5","features": [{"identifier": "account_id","rating":
null,"rank": null

},
{

"identifier": "district_id_bank","rating": null,"rank": null


} ,


{
"identifier": "frequency","rating": 64,"rank": 25},. . .
],
"model_metadata": {"detection_model": {"name": "tradmin_tr_detection_
model","tags": {"version": "release","_commit_id": "dirty_2022-06-
27T14:28:23_a6686c93e888aa47ba3c1e77198846a813931a12"
},
"version": "1"
},
"feature_extraction_model": {
"name": "tradmin_tr_feature_extraction_model",
"tags": {
"version": "release",
"_commit_id": "dirty_2022-06-27T14:27:59_
a6686c93e888aa47ba3c1e77198846a813931a12"
},
"version": "1"
}
}

}
```

```
},
"order": [
 "tr_evaluation"
]
},
"risk": {

"id": "anomaly1",

"name": "anomaly1","severity": 90,"category":
"default","suppressionPeriodUnit": "DAY","suppressionPeriodSize": -
1,"description": "anomaly","conditions": [
{
"identifier": "is_anomaly12",
"value": true,"displayName": "is_anomaly12 dispaly name",
"trace": [
 {
"features": [
"min3","min4","min5"],

"identifiers": [
1962853,1962899,1962928,1963016,1963056,3677396


]

}



]

}

 }
],

"commit_metadata": {
"commit_hash": "dirty_2022-06-27T15:24:46_
a6686c93e888aa47ba3c1e77198846a813931a12","commit_message": "initial commit"


},
"global_trace": [
1962853,1962899,1962928,1963016,1963056,3677396]
```

1.2.1.3. Example JSON message

An example JSON message submitted by ThetaRay is displayed below:


```
{
    "alertedActivityId": "08c61251-af22-457b-9fd8-eba73bba8587",
    "occurredOnDate": "1997-03-28 00:00:00",
    "occurredOnField": "date_loan",
    "solutionId": "talsh-jso",
    "evaluationFlowId": "tr_analysis",
    "dataPermission": "dpv:public",

        "primaryKeySet": {
        "account_id": 10049
        },
        "risk": {
        "id": "anomaly1",
        "name": "anomaly1",
        "severity": 90,
        "category": "default",
        "suppressionPeriodUnit": "DAY",
        "suppressionPeriodSize": -1,
        "description": "anomaly",
        "conditions": [
        {
        "identifier": "is_anomaly12",
        "value": true,
        "displayName": "is_anomaly12 dispaly name"
        }
        ]
        },
        "evaluation": {
        "features": [
        {
        "identifier": "account_id",
        "value": 10049
        },
        {
        "identifier": "district_id_bank",
        "value": 17
        },
        {
        "identifier": "frequency",
        "value": "Monthly issuance"
        },
        {
        "identifier": "date_acct",
        "value": "1996-03-28 00:00:00"
```

```
},
{
"identifier": "disp_id",
"value": 12050
},
{
"identifier": "client_id",
"value": 12358
},
{
"identifier": "type_disp",

"value": "OWNER"
},
{
"identifier": "loan_id",
"value": 7046
},
{
"identifier": "date_loan",
"value": "1997-03-28 00:00:00"
},
{
"identifier": "amount",
"value": 91632
},
{
"identifier": "duration",
"value": 12
},
{
"identifier": "payments",
"value": 7636
},
{
"identifier": "response",
"value": 1
},
{
"identifier": "birth_number",
"value": 430705
},
{
"identifier": "district_id_client",
"value": 17
}
],
```

```
    "evaluationSteps": {
    "tr_evaluation": {
    "name": "Thetaray evaluation",
    "identifier": "tr_evaluation",
    "type": "TR Algo",
    "fusionScore": 0.45222452220938525,
    "pattern": "payments#duration#birth_number",
    "features": [

{

"identifier": "account_id",
"rating": null,
"rank": null

    },
    {

    "identifier": "district_id_bank",
    "rating": null,
    "rank": null
          },
          {

            "identifier": "frequency",
            "rating": 137,
            "rank": 23
          },
          {

            "identifier": "date_acct",
            "rating": null,
            "rank": null
           },
           {

            "identifier": "disp_id",
            "rating": null,
            "rank": null
           },
           {

            "identifier": "client_id",
            "rating": null,
            "rank": null
           },
           {
```

```
 "identifier": "type_disp",
 "rating": 209,
 "rank": 22
},
{

 "identifier": "loan_id",
 "rating": null,
 "rank": null
},
{

 "identifier": "date_loan",
 "rating": null,
 "rank": null
},
{

  "identifier": "amount",
  "rating": 114,
  "rank": 24
},
{

  "identifier": "duration",
  "rating": 909,
  "rank": 2
},
{

  "identifier": "payments",
  "rating": 917,
  "rank": 1
},
{

  "identifier": "response",
  "rating": null,
  "rank": null
},
{

  "identifier": "birth_number",
  "rating": 757,
  "rank": 3
},
```

```
            {

              "identifier": "district_id_client",
              "rating": null,
              "rank": null
            }
          ]
        }
      },
      "order": [
        "tr_evaluation"
      ]
    }
  }
```

1.2.2. Alert Deep Link

1.2.2.1. Retrieving Alert Data Via a Web Browser

In your alert investigation process your company does not use the ThetaRay investigation Center for alert investigation and resolution, you need to know how to access alerts and resolve them with your 3rd party case manager.
This section provides detail on the integration process via a Direct URL link.

To access alert details, complete the URL according to your environment and specific activityId and riskId.

- ***activityId***: activity identifier that uniquely identifies the Investigated Entity ID at a point in time
- ***riskId***: risk identifier that uniquely identifies the risk by type

These two provided identifiers are forwarded to the downstream system through the Alert Distribution interface and require to be stored persistently in order to enable the integration process.

For more information on how these Id elements are used, refer to the Platform guide, under the chapter on Decisioning.

Example direct URL link template is provided here for your information:

```
https://<host>/#/investigation-center/alert/:activityId/:riskId
```

1.2.3. Externalized API IC Authentication Process

To use ThetaRay's available Externalized API's, the delegated user or users are required to first be authenticated.

The authentication process requires the receipt of a successful response message containing a authenticated bearer token.

1.2.3.1. API Details

Request

Method	URL Example details
POST	https://dimitryvi-dimdim-app.development.thetaray.dev.com/ic-be/api/security /login

Note: in the url example shown above, dimitryvi-dimdim-app.development.thetaray.dev.com is an example host, and in reality, the host will vary depending on the customer's environment domain name.

POST request with body

```
{
  "username":<"login name">,
  "password":<"relevant_password">
}
```

For correct access, user must be defined in deployment Keycloak application with relevant roles/groups.

1.2.3.2. Example Response Message

The following is an example successful response message:

1.2.4. Close Alert API

The Close Alert Rest API allows third-party case managers to inform the IC (Investigation Center) of the resolution status of a specified alert.

Note: The alert to be closed can either be identified through an Alert ID as presented / externalized by the Investigation Center or by a combination of the alerted activity ID and risk identifier, as provided by the Alert Distribution functionality from the platform.

1.2.4.1. Authentication Process

For authentication information please see here.

1.2.4.2. HTTP request

```
POST https: <host>//ic-be/api/v1/alerts/close
```

1.2.4.3. HTTP Headers

Send an auth request to get the Bearer JSON Web Token (JWT) and paste it in the Authorization header field.

```
Content-Type = application/json; Authorization = Bearer <JWT>
```

1.2.4.4. Request body

The request body contains data with the following structure.

ⓘ For the query to succeed all fields must be populated with values.

JSON Representation

```
{
"alertedActivityId": "e6d059b6-f5e0-11ec-b939-0242ac120002",
"riskIdentifier": "risk_id_3",
"resolutionCode": "Non_Suspicious",
"notes": "custom note"
"alertId" : "0000001"
}
```

Field Name	Description	Mandatory - Required or not ?
alertId	The identifier of alert	Not required in the case where alertedActivityId and riskIdentifier are presented
alertedActivityId	The identifier of trigger in	Not required in the case where alertId is presented

Field Name	Description	Mandatory - Required or not ?
	alert	
riskIdentifier	The identifier of risk in trigger of alert	Not required in the case where alertId is presented
resolutionCode	The identifier of resolution code in alert	Required
note	Free text comments about alert	Required

1.2.4.5. Response body

If successful, the response body contains data with the following structure:

JSON Representation

```
{
    “host”: “tr-icbe”,
    “time”: “Mon Nov 06 12:01:14 UTC 2023",
    “message”: {
        “success”: “Closing of alert was successful”
    },
    “status”: “OK”
}
```

Fields	
host	The responder's hostname.
time	The response timestamp.
message	The status code message
status	The response status.

If unsuccessful, the response body contains data with the following structure:

JSON Representation

```
{
    “host”: “tr-icbe”,
    “time”: “Mon Nov 06 12:32:03 UTC 2023”,
    “message”: {
        “error”: “Alert with id ‘000001’ is already in ‘state_closed’”
    },
    “status”: “Bad Request”
}
```

1.2.4.6. HTTP Response Codes

Code	Example Response Message
200	“success”: “Closing of alert was successful”
400	“error”: “Alert with id <alert id> is already in ‘state_closed’”
	“error”: “There is failure to change state from <current_state> to ‘’state_closed’’. States are not sequential”
	“error”: “Alert not found.”

1.2.5. Alert Change State API

The change States API enables analysts working with an external case manager, to make changes to an alert's state locally and for that change to initiate an API call that registers the change in the ThetaRay's equivalent workflow domain.

The API information detailed below provides the user with all the required information including Method , API Body, example code snippets and verification messages.

1.2.5.1. Authentication Process

For authentication information please see here.

1.2.5.2. Header Details

Note: Headers must contain authorization clause

Attribute	Details
API Path	/v1/alerts/change-state
Method	PUT

1.2.5.3. Body

The body attributes of the API call are detailed in the following table:

Attribute	Description	Mandatory / Optional	Comments
alertId	The External ID of the Alert	M	
stateId	Code of the state to be changed	M	
resolutionCode	Code of the resolution to be assigned to the alert	M	
notes	Contents of the note associated	O	

Attribute	Description	Mandatory / Optional	Comments
	to the change of state		
userId	Code of the user to be assigned to the change of State	O	If not provided, the 'External System is shown

1.2.5.4. Note Object Example

Note object just contains a note clause with note body

Also please be aware, that Notes and Resolution Code are necessary in certain state changes but not in others. For example, working with the default workflow, changing the alert state to 'Closed' (state_closed) must have a resolution code and note in the body of the API, but with a change of state to 'Pending' ('State_pending), it is not required.

Practical Example:

```
{

     "alertId": "000004",

     "stateId": "state_pending_l1",

     "resolutionCode": "Non_Suspicious",

     "notes":"note body",

     "userId": "test_user"

}
```

1.2.5.5. Request Example

```
curl --location --request PUT 'https://test.thetaraydev.com/ic-
be/api/v1/alerts/change-state' \
--header 'Authorization: Bearer test-token' \
--header 'Content-Type: application/json' \
--data '{
"alertId": "000004",
"stateId": "state_pending_l1",
"resolutionCode": "Non_Suspicious",
"notes": "test note",
"userId": "test_user"
}'
```

1.2.5.6. Example Success Response Message

```
{

      "result": "",

      "host": "tr-icbe",

      "time": "Thu Aug 24 10:49:41 UTC 2023",

      "message": {

           "error": "Alert 000004 state was successfully changed to state_
pending_l1"

      },

     "status": "OK"

}
```

1.2.5.7. Example Failure Response Message

```
{

      "result": "",

      "host": "tr-icbe",

      "time": "Thu Aug 24 10:49:41 UTC 2023",

      "message": {

           "error": "Alert 000004 state changed to state_pending_l1 failed
due to: additional fail message"

      }

}
```

1.2.5.8. Verification - Responses / Error Messages

If the source alert id, workflow , states and resolution codes do not match ThetaRay's equivalent details one of the following error messages will be displayed:

- "The target alertID is not found in ThetaRay's system"
- "The target state is not found in ThetaRay's system"
- "The target resolution code is not found in ThetaRay's system"
- "The target workflow is not found in ThetaRay's system"

Note: If a ***Mandatory*** field is missing the following error message is displayed in the response:

- "The Mandatory field (Name) is missing."

1.2.5.9. HTTP Response Codes

Code	Example Response Message
200	Alert {0} state was successfully changed to {1}
400	Alert id {0} is invalid, it must be a text with 6 digits
	The mandatory field stateId is missing
	Resolution code {0} not exists
	Alert id {0} is non-existent for all mappers
	Alert {0} was not found
	State id {0} not exists for alert process definition {1}
	Workflow {0} for alert {1} not exists
	Alert with id {0} has wrong mapper identifier
	Alert Mapper with identifier {0} not found
	This action is permitted only if the alert is assigned to the user
	The alert state cannot be changed because it is unassigned

Note: The API call is audited and subsequently available for viewing in the audit log, by users with appropriate permission.

1.2.6. Add Document via API

The add documents API enables analysts working with an external case manager (as opposed to ThetaRay's Investigation center case manager), to add support evidence documents to an alert's investigation locally and for that change to initiate an API call that registers and loads the document to the ThetaRay's equivalent workflow resources repository. (A full current list of supported doc types can be found in the Investigation Center User guide).

This API once integrated into the system supporting an external case manager, enables the analyst to seamlessly add documents to an alert record repository hosted by ThetaRay's IC application database.

1.2.6.1. Authentication Process

For authentication information please see here.

1.2.6.2. Request Method and URL

```
POST {{icbe_app_path}}/api/v1/alerts/documents
```

Request Parameters

Parameter	Description
alertId	The external ID of the alert, not the internal alerted Activity ID. This is a mandatory field.
fileName	The title of the document
document	The document file to be attached to the TR alert. This is a mandatory field.
categoryId	The category of the document. This is a mandatory field.
description	The text associated with the document. This is a mandatory field.
userId	Code of the user to be assigned to the change of status. This is an optional field. If it is not provided, the "External System" shall be indicated.

API Sending Request

```
POST {{icbe_app_path}}/api/v1/alerts/documents
Content-Type: multipart/form-data; boundary=11111

--11111
Content-Disposition: form-data; name="alertId"

{{alertId}}
--11111
Content-Disposition: form-data; name="fileName"

{{fileName}}
--11111
Content-Disposition: form-data; name="document"; filename={{document_
filename}}
Content-Type: text/plain;charset=utf-8

{{document_content}}
--11111
Content-Disposition: form-data; name="categoryId"

{{categoryId}}
--11111
```

```
Content-Disposition: form-data; name="description"

{{description}}
--11111
Content-Disposition: form-data; name="userId"

{{userId}}
--11111
```

Response Examples

1. Response with error message.

```
HTTP/1.1 400

{
  "host": "tr-icbe",
  "time": "Mon Aug 28 08:41:29 UTC 2023",
  "message": {
    "error": "The mandatory field alertId is missing."
  },
  "status": "Bad Request"
}
```

2. Response without error message.

```
HTTP/1.1 200

{
  "host": "tr-icbe",
  "time": "Mon Aug 28 08:43:01 UTC 2023",
  "message": {
    "success": "Document has been uploaded."
  },
  "status": "OK"
}
```

Extra notes to be aware of regarding the 'add documents' from an external case manager API

- If the source category is not present in ThetaRay's system, it is added to the document category list.
- If one of the mandatory fields is missing, the error message in API response ***"The mandatory field (name) is missing."*** is displayed.

1.2.6.3. HTTP Response Codes

Code	Example Response Message
200	Document has been successfully uploaded
400	The mandatory field 'alertid' is missing
	The mandatory field 'fileName' is missing
	The mandatory field 'document' is missing
	The mandatory field 'categoryId' is missing
	The mandatory field 'description' is missing
	Alert {0} was not found
	The document format is not supported by the system. Supported formats: {0}

1.2.6.4. Audit

Note: The API call is audited and subsequently available for viewing in the audit log, by users with appropriate permission.

1.2.7. Alert Externalization API

1.2.7.1. Introduction

This chapter sub section describes the API required to implement externalization of Transaction Monitoring (TM), Transaction Screening (TS) and Customer Screening (CS) alerts to customers.

1.2.7.2. Configuring IC Alert Externalization

1.2.7.2.1. Alert Externalization API

By default, alerted activities are distributed to ThetaRay's Investigation Center (the IC) at the end of the analysis pipeline. However, through configuration in IC Settings, they can be sent to external systems that are not part of the ThetaRay environment such as case management applications. This process is an integral part of the alert creation and distribution processes.

The Alert Externalization mechanism is based on an HTTPS request triggered by ThetaRay with a JSON payload consisting of the content of one or more alerted

activities and alert entities. Batch sizes can be tuned as part of the solution implementation.

From a security perspective, the system supports targets that use HTTP Basic Authentication and requires that a username and password or API Key that are provided by customers for configuration in IC Settings.

See the Investigation Center Settings User Guide for configuration of alert externalization in the Thetaray's IC Settings.

1.2.7.2.2. URL and Endpoint

Customers need to provide API endpoint credentials. These consist of a user name and password, or an api-key as shown above that are set in IC Settings. They should be part of a controller's task at the customer's site.

1.2.7.2.3. Configuring the Service

When alerts of type TM, TS and/or CS are published to Thetaray's Investigation Center, customers need to set in advance an API endpoint on their premises to receive JSONs from Thetaray.

A customer is required to expose an HTTP endpoint that accepts HTTP POST requests consisting of JSON message conforming to format described later in this section. Failure to deliver alert data to the customer due to functional reasons on the customer's end, will result in a failure of the entire distribution of alerts of an analysis batch and will need to be attended to by ThetaRay support.

Alerts are delivered in an 'at least once manner' - during failure conditions, ThetaRay may re-deliver alerts that have already been processed by the customer. A deduplication process needs to be established on the customer side to avoid redundant cases (for example) from being created.

1.2.7.2.4. Response Codes

Table 1: *Standard Request Return Codes Additional 2nn codes*

Code	Response Message
200	OK
Any code other than 200	Any error message

1.2.7.2.5. Receiving the Response Payload

General

The response payload includes a return code and return data (if any) as a JSON file. In general, any data returned with a return code other than 200 should be treated as possibly invalid.

TM JSON Alert Creation Request Representation

```
 {
"data": [
  {

   "alertRequest": {
      "alertedActivityId": string,
      "occurredOnDate": string,
      "occurredOnField": string,
      "solutionId": string,
      "evaluationFlowId": string,
      "dataPermission": string,
      "primaryKeySet": {
        "account_id": string,
      },
      "groupingKeySet": {
         “party_id”: string,
       }
      "primaryEntitiesList": string,
      "risk": {
        "id": string,
        "name": string,
        "severity": string,
        "category": string,
        "suppressionWindowSize": number,
        "description": string,
        "conditions": list of objects,
      },
      "evaluation": {
        "features": [
          {
```

```
        "identifier":string,
           "value": string
    },
```

Parameters	Description
alertedActivityId	String The unique ID of the alerted activity record. Multiple records can be sent with the same alertedActivityID in case multiple risks are identified for the given activity. In this case a different risk identifier will be associated with each of the alerted activities
occurredOnDate	String The timestamp of the activity that triggered the alert. This is a logical time assigned to the activity by the evaluation process – for example, the year and month associated with the alert.
occurredOnField	String The name of the column where the occurredOnField value is stored.
solutionId	String The unique name of the ThetaRay Platform instance that generated the alert.
evaluationFlowId	String The unique name of the evaluation flow that generated the alert.
dataPermission	String For internal use only.
primaryKeySet	Map (key: string, value: string) The unique identifier of the investigated entity (for example "account" or "customer").
groupingKeySet	Map (key: string, value: string) The unique party identifier of the investigated entity
primaryEntitiesList	List of map (key: string, value: string) Associated entities for transaction monitoring party alert

risk	• id - String, unique identifier of the risk
	• name - String, name of the risk
	• severity - Number, current severity score of the risk
	• category - String, default value is “default”
	• suppressionWindowSize - Number
	• description - String, description of the risk, like “anomaly”
	• conditions - List of objects:
	  • identifier - String,
	  • value - String,
	  • displayName - String

evaluation -> features	List of objects The evaluation flow results: • identifier - String • value - String

2. Screening

This chapter contains details of the APIs that are provided by ThetaRay to support both Transaction screening and the Customer screening solutions.
This includes information such as request response API code examples and other relevant usage guidance.

2.1. Transaction Screening

2.1.1. How it Works

The Thetaray Screening Solution enables matching across data held within transactions, with an emphasis on global payment schemes. The Thetaray Screening Solution provides real time screening for all parties and agents involved in the transaction.

Status /Description of 3 API's

1. **Generic Canonical API** - recommended for all new deployments as it is agnostic to all payment methods, and additionally provides more comprehensive screening results.
   For more info goto *Transaction Screening- Generic Canonical API section (2.1.3) below.*
2. **Transaction Screening SWIFT MT API** - API that supports the MT message format and NOT recommended for new deployments.
   For more info goto *Transaction Screening- SWIFT MT API Section (2.1.4) below.*
3. There was a third API available in the previous version of this document - Transaction Screening SWIFT MX - which has now been deprecated.

2.1.2. Login API

Note: The Login procedure is the same for this section and the Customer Screening section.

Authentication is based on calling an API to create a JWT access token. The API requires a client secret that will be provided to you by our customer support team at onboarding to the service. You will use the access token to call the screen transaction API Request Attributes.

The JWT token is valid for 5 minutes and the expirationInMinutes will always show "5" as it is relative to the time the token was first created. However, ThetaRay utilizes a 5 minute cache option for multiple logins therefore will not generate a new token for the duration of the cache.

The recommended way to implement authentication for screening requests is the following:

- When sending a screening request, first decode the "time to live" timestamp from the actual JWT token.
- If the current time is larger then the "time to live" then generate a request for the creation of a new token before sending the screening request.

Please note that if the token has only a few milliseconds to expiry, then the screening request sent with this token might receive a 401 authentication error and will need to be resent with a valid token

2.1.2.1. Request Attributes

```
Endpoint:POST /security/accessToken
{
 "clientSecret": "<credentials provided by ThetaRay>"
}
```

Example Login API Response Body – Response Payload

```
{
    "token": "<Token>",
    "expirationInMinutes": 5
}
```

2.1.3. Transaction Screening - Generic Canonical API

2.1.3.1. Introduction to the Transaction Screening Generic Canonical API

This API is designed to seamlessly handle both different payment messages, compliant with the ISO 20022 standard. This ensures that no matter what kind of payment messages or parties you are dealing with for example, whether you are processing legacy messages (e.g., Originator and Beneficiary parties) or transitioning to the newer MX messages (with Debtors, Creditors, etc.), the ThetaRay system can accommodate your needs.

2.1.3.2. How to Engage with the API:

Understand Your Use Case: The information you provide depends on what type of payment message you're working with. The party "Debtor" refers to the sender of the transaction, whilst "Creditor" refers to the receiver. For more details on party and agent types and meaning, please refer to the official ISO20022 or SWIFT MX documentation sources.

- Map Your Payment Message to the API: For each payment message type, identify the parties involved and assign them to the relevant `partyId` and `partyType` fields. The correct party role and identifier help ensure accurate transaction screening.
- Regardless of the payment type, most fields here are optional. However, payments should always have a creditor and debtor so make sure to always include these.

This API was built to be adaptable, whether you're processing legacy messages or newer, more structured ISO 20022 messages. By using the right identifiers and roles for each party, you can ensure smooth transaction screening tailored to your specific use case.

2.1.3.3. Basic Fields Explained:

requestId:

- In the context of our API, the requestId is a unique identifier that is assigned to each screening request. Think of it like a tracking number for a package—it helps both you and us keep track of each individual screening request. This ID ensures that every request you send for transaction screening can be identified and traced separately.

Note: - If you don't have an existing system, you can start by simply assigning an incremental number or timestamp for each new request (e.g., 20241006T123456)

---

partyId:

- This field represents the unique identifier of the party involved in the transaction. The partyId tells us who you are screening. By providing a unique identifier for each party, we can check that specific person or entity against various watchlists or AML databases to detect any potential risk.
- Depending on the type, the partyId could be their name, passport number, account number or tax ID number (e.g. an individual debtor - JohnDoe_12345 where John Doe is the name, and 12345 is an account number or unique identifier)

2.1.3.4. Request

Element	URL	Comments
Endpoint	/screening/transaction/generic/check	N/A
Method	POST	N/A

Disclaimer: The fields under the "Validation" column may appear unfilled for most rows because they are only relevant if you're using the "Check and Monitor" endpoint, see subsection 2.1.3.10. If you're not utilizing this specific endpoint, the only required field is the Request ID, which is marked as mandatory (M) in the table.

For customers using the Check and Monitor feature, validations will be provided based on the project's configuration and requirements as they are implementation specific.

Important note for users: To make working with the Canonical API that bit easier to understand, users should in be aware that the bulk of listed fields in the following table are optional and in general only needed in the case where the check and monitor endpoint are intended to be used.

The two fields that are mandatory when interacting with the Canonical API are:

- requestId
- partyId

Field Name	Sub Field	Data Type	Required	Validation	Screened
requestId		string	M	min 20 max 200 characters, should be unique amongst all requests	no
messageType		string	O		no
transactionType		string	O		no

Field Name	Sub Field	Data Type	Required	Validation	Screened
transactionDirection		string	O		no
transactionAmount		double	O		no
transactionCurrency		string	O		no
transactionDate		date	O		no
createDateTime		date	O		no
senderAmount		string	O		no
senderCurrency		string	O		no
receiverAmount		string	O		no
receiverCurrency		string	O		no
exchangeRate		string	O		no
uniqueTransactionId		string	O		no
endToEndId		string	O		no
instructionId		string	O		no
clearingSystemRef		string	O		no
localInstrument		string	O		no
returnInfo	indicator	boolean	O		no
	type	string	O		no
	description	string	O		no

Full JSON Format- General

```
{
  "requestId": "string, 20-200 chars - M",
```

```
"messageType": "string",
"transactionType": "string",
"transactionDirection": "string",
"transactionAmount": "double",
"transactionCurrency": "string",
"transactionDate": "date",
"createDateTime": "date",
"senderAmount": "string",
"senderCurrency": "string",
"receiverAmount": "string",
"receiverCurrency": "string",
"exchangeRate": "string",
"uniqueTransactionId": "string",
"endToEndId": "string",
"instructionId": "string",
"clearingSystemRef": "string",
"localInstrument": "string",
"returnInfo": {
  "indicator": "boolean",
  "type": "string",
  "description": "string"
}
```

2.1.3.5. Parties - Object

Field Name	Sub Field 1	Sub Field 2	Sub Field 3	Type	Required	Validation	Screened
partyId				string	M	Mandatory if entity exists	no
partyType				string	M	Mandatory if entity exists Has to be from list of party types mentioned below Unique within the transaction	no
screeningEntityType				string	O	individual, organization, aircraft, vessel	yes
fullName				string	M	Mandatory, if firstName or lastName (or bic for oganizations) is not provided, and should be between 1-255 characters	yes
firstName				string	O	Should be between 1-255 characters	yes
lastName				string	O	Should be between 1-255 characters	yes
ipAddress				string	O		yes
countryOfResidence				string	O	Country code ISO 3166, Alpha-2	yes
nationalities				array of strings	O	relevant for individual , agent, country code ISO 3166, Alpha -2 code	yes
addressLine				string	O		yes

Field Name	Sub Field 1	Sub Field 2	Sub Field 3	Type	Required	Validation	Screened
address	streetName			string	O	Should be between 1-255 characters	yes
	buildingNumber			string	O	Should be between 1-10 characters	yes
	buildingName			string	O		no
	floor			string	O		no
	city			string	O	Should be between 1-100 characters	yes
	region			string	O	Should be between 1-20 characters	yes
	country			string	O	country code ISO 3166, Alpha-2	yes
	postalCode			string	O	Should be between 1-40 characters	yes
account	iban			string	O	case sensitive	no
	currency			string	O		no
	name			string	O		no
Identification	IdentificationType			string	O	individual or organization	no

Field Name	Sub Field 1	Sub Field 2	Sub Field 3	Type	Required	Validation	Screened
	identityDocuments	type		string	O		no
		number		string	O		yes
		country		string	O		no
		description		string	O		no
		datesOfIssue		array of dates	O		no
		datesOfExpiry		array of dates	O		no
		placesOfIssue	city	string	O		no
			state	string	O		no

Field Name	Sub Field 1	Sub Field 2	Sub Field 3	Type	Required	Validation	Screened
			country	string	O		no
			placeame	string	O		no
	gender			string	O	relevant only identificationType = individual accepts values: f/F (female) or m/M (male)	yes
	datesOfBirth			array of dates	O	relevant only identificationType = individual date can be YYYY-MM-DD or YYYY or YYYY/YYYY	yes
	placesOfBirth	city		string	O	relevant only identificationType = individual	yes
		state		string	O		yes
		country		string	O		yes
	bic			string	O	relevant only identificationType = organization 9-11 characters	yes
	lei			string	O	relevant only identificationType = organization 20 characters	yes
	datesOfRegistry			array of dates	O		yes
	placesOfRegistry	city		string	O		yes
		state		string	O		yes
		country		string	O		yes
		placeName		string	O		no
taxRemittance	taxIdentification			string	O		yes
	taxRegistrationIdentification			string	O		yes

Field Name	Sub Field 1	Sub Field 2	Sub Field 3	Type	Required	Validation	Screened
contactDetails	name			string	O		no
	phoneNumber			string	O		no
	mobileNumber			string	O		no
	email			string	O		no
	jobTitle			string	O		no

Additional Specifications:

1. **PartyType -** The supported party types mentioned below are in line with official ISO20022 and SWIFT MX specifications. The party "Debtor" refers to the sender of the transaction, whilst "Creditor" refers to the receiver. For more details on party and agent types and their meanings, please refer to the official ISO20022 or SWIFT MX documentation sources.
   - Debtor
   - Creditor
   - Ultimate Debtor
   - Ultimate Creditor
   - Initiating Party
2. **Screening Entity Type** - this field pertains to the entity type of the party, and can be individual, organization, aircraft and vessel. If this field exists in the screening request, it is used to exclude matches from any other entity type, thus refining the screening results and reducing false positives.

Full JSON Format:- Parties

```
"parties": [
    {
```

```
"partyId": "string - M",
"partyType": "PartyType - M",
"screeningEntityType": "ScreeningEntityType",
"fullName": "string",
"firstName": "string",
"lastName": "string",
"ipAddress": "string",
"countryOfResidence": "string (ISO 3166, Alpha-2 code)",
"address": {
  "streetName": "string",
  "buildingNumber": "string",
  "buildingName": "string",
  "floor": "string",
  "city": "string",
  "region": "string",
  "country": "string",
  "postalCode": "string"
},
"addressLine": "string",
"account": {
  "iban": "string",
  "currency": "string",
  "name": "string"
  },
"taxRemittance": {
  "taxIdentification" : "string",
  "taxRegistrationIdentification": "string"
  },
"identification": {
```

```
 "identificationType": "individual/organization",
 "identityDocuments": {
   {
     "type": "string",
     "number": "string",
     "country": "string",
     "description": "string",
     "datesOfIssue": [
       "2020-01-01"
     ],
     "datesOfExpiry": [
       "2030-12-31"
     ],
     "placesOfIssue": [
       {
         "city": "string",
         "state": "string",
         "country": "string",
         "placeName": "string"
       }
     ]
   }
},
 -- individual
 "gender": "string",
 "datesOfBirth": [
   "date",
   "date"
 ],
```

```
    "placesOfBirth": [
      {
        "city": "string",
        "state": "string",
        "country": "string"
      }
    ],
    -- organization
    "bic": "string 8-11 chars",
    "lei": "string 20 chars",
    "datesOfRegistry": [
      "date",
      "date"
    ],
    "placesOfRegistry": [
      {
        "city": "string",
        "state": "string",
        "country": "string"
      }
    ]
  },
  "contactDetails": {
    "name": "string",
    "phoneNumber": "string",
    "mobileNumber": "string",
    "email": "string",
    "jobTitle": "string"
  }
```

```
    }
]
```

2.1.3.6. Agents - Object

Field Name	Sub Field	Type	Required	Validation	Screened
agentId		string	M	Mandatory if entity exists	no
agentType		string	M	Has to be one of the values shown under the "**Additional Specification**" list shown following **Note**: no spaces between words	no
screeningEntityType		string	O		yes
clearingSystemMemberId	clearingSystemId	string	O		no
	memberID	string	O		no
bic		string	O	9-11 chars either BIC or name has to be provided and are subject to validation	yes
lei		string	O	20 characters	yes
name		string	M	If name is provided then needs to be between 1-255 characters	yes
addressLine		string	O		yes

Field Name	Sub Field	Type	Required	Validation	Screened
address	streetName	string	O	Should be between 1-255 characters	yes
	buildingNumber	string	O	Should be between 1-10 characters	yes
	buildingName	string	O		no
	floor	string	O	Should be between 1-100 characters	no
	city	string	O	Should be between 1-40 characters	yes
	region	string	O		yes
	country	string	O	country code ISO 3166, Alpha-2	yes
	postalCode	string	O	Should be between 1-40 characters	yes
account	iban	String	O	Case sensitive	no
	currency	String	O		no
	name	String	O		no
nationalities		array of strings	O	country code ISO 3166, Alpha -2	yes

Additional Specifications:

1. **AgentType -** the list below contains the supported values for the Agent Type field:
   - Previous Instructing Agent 1
   - Previous Instructing Agent 2
   - Previous Instructing Agent 3
   - Instructing Agent
   - Instructed Agent
   - Intermediary Agent 1

- Intermediary Agent 2
- Intermediary Agent 3
- Debtor Agent
- Creditor Agent
- Instructing Reimbursement Agent
- Instructed Reimbursement Agent
- Third Reimbursement Agent

2. **Screening Entity Type** - this field pertains to the entity type of the agent, and can be individual, organization, aircraft and vessel. If this field exists in the screening request, it is used to exclude matches from any other entity type, thus refining the screening results and reducing false positives.

Full JSON Format:- Agents

```
"agents": [
    {
      "agentId": "string unique",
      "agentType": "AgentType unique",
      "screeningEntityType": "ScreeningEntityType",
      "clearingSystemMemberId": {
        "clearingSystemId": "string",
        "memberId": "string"
      },
      "bic": "string 8-11 chars",
      "lei": "string 20 chars",
      "name": "string",
      "addressLine": "string",
```

```
      "address": {
        "streetName": "string",
        "buildingNumber": "string",
        "buildingName": "string",
        "floor": "string",
        "city": "string",
        "region": "string",
        "country": "string",
        "postalCode": "string"
      },
      "account": {
        "iban": "string",
        "currency": "string",
        "name": "string"
      }
    }
  ]
```


2.1.3.7. Narratives - Object

Field Name	Type	Required	Validation	Screened
categoryPurpose	string	O		yes
referredDocumentInformationType	string	O		yes
referredDocumentInformationLineDetailsType	string	O		yes
referredDocumentInformationLineDetailsNumber	string	O		yes

Field Name	Type	Required	Validation	Screened
referredDocumentInformationLineDetailsDescription	string	O		yes
creditorReferenceInformationType	string	O		yes
creditorReferenceInformationReference	string	O		yes
remittanceIdentification	string	O		yes
taxRemittanceAdministrationZone	string	O		yes
taxRemittanceReferenceNumber	string	O		yes
taxRemittanceMethod	string	O		yes
taxRemittanceRecordType	string	O		yes
taxRemittanceType	string	O		yes
taxRemittanceRecordCategory	string	O		yes
taxRemittanceRecordCategoryDetails	string	O		yes
taxRemittanceRecordDebtorStatus	string	O		yes
taxRemittanceRecordCertificateIdentification	string	O		yes
taxRemittanceRecordFormsCode	string	O		yes
taxRemittanceRecordAdditionalInformation	string	O		yes
garnishmentRemittanceType	string	O		yes
garnishmentRemittanceReferenceNumber	string	O		yes
additionalRemittanceInformation	string	O		yes
creditorReference	string	O		yes
remittanceInfo	string	O		yes
relatedRemittanceInfo	string	O		yes

Field Name	Type	Required	Validation	Screened
senderToReceiverInfo	string	O		yes
transactionReference	string	O		yes
debtorNarrative	string	O		yes
creditorNarrative	string	O		yes
ultimateDebtorNarrative	string	O		yes
ultimateDebtorNarrative	string	O		yes
initiatingPartyNarrative	string	O		yes

Full JSON Format:- Narratives

```
"narratives": {
        "creditorReference": "123456",
        "remittanceInfo": "bin laden",
        "relatedRemittanceInfo": "bin laden",
        "senderToReceiverInfo": "Sender To Receiver Info",
        "transactionReference": "789012",
        "categoryPurpose": "1",
        "referredDocumentInformationType": "2",
        "referredDocumentInformationLineDetailsType": "3",
        "referredDocumentInformationLineDetailsNumber": "4",
        "referredDocumentInformationLineDetailsDescription": "Description",
        "creditorReferenceInformationType": "5",
        "creditorReferenceInformationReference": "6",
        "remittanceIdentification": "7",
        "taxRemittanceAdministrationZone": "8",
        "taxRemittanceReferenceNumber": "9",
```

```
        "taxRemittanceMethod": "10",
        "taxRemittanceRecordType": "11",
        "taxRemittanceType": "1152ch",
        "taxRemittanceRecordCategory": "12",
        "taxRemittanceRecordCategoryDetails": "13",
        "taxRemittanceRecordDebtorStatus": "14",
        "taxRemittanceRecordCertificateIdentification": "15",
        "taxRemittanceRecordFormsCode": "16",
        "taxRemittanceRecordAdditionalInformation": "17",
        "garnishmentRemittanceType": "18",
        "garnishmentRemittanceReferenceNumber": "19",
        "additionalRemittanceInformation": "Additional Info",
        "debtorNarrative": "debtorNarrative",
        "creditorNarrative": "creditorNarrative",
        "ultimateDebtorNarrative": "ultimateDebtorNarrative",
        "ultimateCreditorNarrative": "ultimateCreditorNarrative",
        "initiatingPartyNarrative": "initiatingPartyNarrative"
    }
```


2.1.3.8. ForensicData - Object

Field Name	Type	Required	Validation	Screened
key/value	string	O	custom fields that are only shown on the alert	no

Full JSON Format:- Forensic

```
  "forensicData": {
    "string": "string"
  }
```

2.1.3.9. Profile - Object

Field Name	Type	Required	Validation	Screened
profile	string	O	needs to be from list of profiles that was set up through the Screening Settings	no

Note: If the profile name parameter is not sent, then the default screening settings will apply. If an incorrect profile name is sent, the screening request will show an error message.

2.1.3.10. Example Request

```
{
    "requestId": "requestId_012345678901",
    "messageType": "messageType",
    "transactionType": "transactionType",
    "transactionDirection": "transactionDirection",
    "transactionAmount": 300,
    "transactionCurrency": "USD",
    "transactionDate": "2023-07-07",
    "createDateTime": "2023-07-07",
```

```
"senderAmount": "200",
"senderCurrency": "USD",
"receiverAmount": "800",
"receiverCurrency": "ILS",
"exchangeRate": "4",
"uniqueTransactionId": "uniqueTransactionId",
"endToEndId": "endToEndId",
"instructionId": "instructionId",
"clearingSystemRef": "clearingSystemRef",
"localInstrument": "localInstrument",
"return": {
    "indicator": true,
    "type": "type",
    "description": "description"
},
"parties": [
    {
        "partyId": "partyId",
        "partyType": "Debtor",
        "fullName": "Osama bin laden",
        "firstName": "string",
        "lastName": "string",
        "ipAddress": "string",
        "countryOfResidence": "AF",
        "account": {
            "IBAN": "IBAN"
        },
        "identification": {
            "identificationType": "individual",
```

```
                "gender": "m"
            }
        }
    ],
    "agents": [
        {
            "agentId": "agentId",
            "agentType": "DebtorAgent",
            "bic": "SABRRUMM",
            "clearingSystemMemberId": {
                "clearingSystemId": "clearingSystemId",
                "memberId": "memberId"
            }
        }
    ],
    "narratives": {
        "remittanceInfo": "sponsorint terrorism"
    },
    "forensicData": {
        "test": "test"
    }
}
```


2.1.3.11. Response Attributes

2.1.3.11.1. Success Responses

Returns a status of the transaction that was screened and whether there is a match.

Match Results

The response includes the fields mentioned below, by default. Additional optional fields are available to add to the screening responses, listed below. Each field has the option to be added or removed from the screening response.

Note: Adding fields to the response will may have a slight impact on real time performance with high volume screening.

Default Response Fields:

- checkResult - boolean, indicates MATCH or NO MATCH
- statusPollingUrl - contains a URL for polling the status of the alert (see Polling API Request & Response section for more details)
- entityType - contains the screened entity type, e.g. Originator, Beneficiary, Beneficiary Bank, etc.
- score - the match score for the entity
- list - list dataset name
- dateOfPublication - the recent datetime the actual list was updated, according to the official publication date of the list
- dateOfUpload - the recent datetime the list was imported into screening engine

Optional response Fields

In addition to the default response fields, there are a number of optional fields that can be added if required:

- List Category - the category of the list that was matched, for example, PEP
- Record Id - the id of the matched record on the screening list
- Matching Names - the name that was matched from the screening list
- Sublists - any sublist of the screening list that was used in the screening request

Example response of default fields:

Example of Extended response when there is no match.

```
{
    "result": {
        "checkResult": "NO_MATCH"
    }
}
```

Example code block of extended response when there is a match.

```
{
    "result": {
        "checkResult": "MATCH",
        "statusPollingUrl": "<statusPollingUrl>",
        "matchResults": [
            {
                "entityType": "ORIGINATOR_NAME",
                "score": "0.84",
                "list": "AUSTRALIA",
                "dateOfPublication": "2023-03-20T22:44:36",
                "dateOfUpload": "2023-03-29T14:45:11.721"
            },

        ]
    }
}
```

Example response of with all fields enabled:

```
{
    "result": {
        "checkResult": "MATCH",
        "statusPollingUrl": "<statusPollingUrl>",
        "matchResults": [
            {
                "entityType": "ORIGINATOR_NAME",
                "recordId": "11378",
                "score": "0.91",
                "matchingNames": [
                    {
                        "fullName": "AL QA'IDA"
                    }
                ],
                "list": "OFAC SDN",
                "listCategory": "SANCTION",
                "subLists": [
                    "SDGT"
                ],
                "dateOfPublication": "2023-12-14T00:00:00",
                "dateOfUpload": "2023-12-18T10:10:15.568"
            }
}
```


2.1.3.11.2. Error Responses

Http Status	Description
400	Invalid Request - the request contains validation errors. For a full list of messages, please see section below
401	Unauthorized Request
422	Request cannot be processed - entity type contains an inapplicable field. For a full list of messages, please see section below
500	Internal Server Error
503	Service Unavailable

Error Messages

Message	HTTP Status
Request ID does not conform, shall be in the range of 20-200 characters	400
Request ID value is already used	400
Invalid Entity Type, check possible entity type values	400
First name does not conform, shall be in the range of 1-255 characters	400
Full Name does not conform, shall be in the range of 1-255 characters	400
Last Name does not conform, shall be in the range of 1-255 characters	400
Street Name does not conform, shall be in the range of 1-255 characters	400
Building Number does not conform, shall be in the range of 1-10 characters	400
Postal Code does not conform, shall be in the range of 1-40 characters	400
City does not conform, shall be in the range of 1-100 characters	400
Region does not conform, shall be in the range of 1-40 characters	400
First Name is not applicable, shall not be provided if Entity Type is different from: Individual	422
Last Name is not applicable, shall not be provided if Entity Type is different from: Individual	422
Gender is not applicable, shall not be provided if Entity Type is different from: Individual	422

Message	HTTP Status
Dates of Birth are not applicable, shall not be provided if Entity Type is different from: Individual	422
Dates of Registry are not applicable, shall not be provided if Entity Type is different from: Organization	422
Countries of Registry are not applicable, shall not be provided if Entity Type is different from: Organization	422
Invalid Date, shall be a valid date value/format	422
Invalid Country, shall be a valid ISO 3166-1 Alpha-2 value	422

2.1.3.12. Alert Status Responses- Polling API

Returns a status of the transaction that has an alert populated in the investigation center that an analyst has approved or blocked.

The Polling API provides the status of a screening alert via polling. The API can be polled from alert creation, which is done via an asynchronous process following the screening request.

The URL for polling is returned in the screening response (synchronous to the request). The polling response has a five-minute cache.

The Login API endpoint should be used before polling in order to authenticate.

There are two endpoints available for the Polling API - new customers should always use the “extended” Polling API, and existing customers are welcome to integrate it as well. The previous endpoint will continue to be supported, but not enhanced.

2.1.3.12.1. Extended Polling API

Element	URL
Endpoint	/screening/investigation/status/extended/{requestId}
Method	GET

Success Response

The success response pertains to two states: either the alert is under investigation by the analyst or the alert has been closed.

Alert under investigation success response.

```
{
 “result”: “Alert is under investigation”
}
```

Alert closed success response includes the following fields:

- **Result** - the result of closing the alert, will be either “Alert approved” or “Alert blocked”
- **Recommended Resolution** - an additional resolution code that the analyst selects when closing the alert
- **Comment** - the note the analyst fills in when closing the alert
- **Whitelisted** - a list of entity types that were whitelisted when the analyst closed the alert (if the analyst whitelisted any entities)
    - Please note that whitelisted entities will only appear if the alert has been approved

Example of Approved Alert Response

```
{
    "result": {
        "result": "Alert approved",
        "recommendedResolution": "country differs",
        "comment": "<p>tester</p>",
        "whitelisted": [
            {
                "entityType": "ORIGINATOR_NAME"
```

```
            }
        ]
    }
}
```

- **Blocked Matches** - details of the blocked match that caused the analyst to block the alert, meaning, the true positive match details. Please note these details will appear only if the alert has been blocked.

Example of Blocked Alert Response

```
{
    "result": {
        "result": "Alert rejected",
        "recommendedResolution": "exceeds_risk_appetite",
        "comment": "test",
        "blockedMatches": [
            {
                "list": "OFAC SDN",
                "listCategory": "SANCTION",
                "entityType": "BENEFICIARY_NAME",
                "recordId": "19609"
            }
        ]
    }
}
```

Error Responses

HTTP Status	Description
403	Unauthorized request
404	Alert creation failed
404	Alert cannot be found for this request ID
500	Internal server error
503	Service unavailable

2.1.3.12.2. Polling API - Old Version

Element	URL
Endpoint	/screening/investigation/status/{requestId}
Method	GET

Success Response

```
{
  "Transaction approved/Transaction rejected/Transaction is under investigation"
}
```

Error Responses

HTTP Status	Description
403	Unauthorized request
404	Alert cannot be found for this request ID
500	Internal server error
503	Service unavailable

2.1.3.12.3. Alert Status Response - Callback API

The callback API sends a request from the ThetaRay screening service to a dedicated URL whenever a screening alert is closed by the analyst in the Investigation Center.

Note: If subscribing to this callback API, the closing of the alert is dependent on a successful API call. Meaning, the screening alert cannot be closed if any configuration of the callback API is incorrect (e.g. incorrect URL, incorrect authentication, etc.)"

Supported methods of authentication:

1. No authentication - only a URL needs to be provided to ThetaRay Support
2. API key authentication - a URL and an API Key need to be provided. The header in the callback will include "apikey" and the value of the key provided.
3. JWT authentication - in addition to the URL, another URL needs to be provided in order to generate the JWT token, as well as the body of the request for generating the token (in key-value pairs), the header in the callback for this authentication method will be:

   Authorization: Bearer <jwt>

The following attributes are sent in the body:

- **businessMessageId** - the transactionId from the original screening request
- **result** - the alert resolution, can be either "approve" or "block"
- **recommended resolution** - additional resolution code that is selected by the analyst when closing the alert, for example, "SAR worthy"
- **creationDate** - the creation datetime for the callback call
- **comment** - the analyst has an option to add a comment in the Investigation Center when closing the alert
- **whitelisted** - a list of the entity types that were whitelisted when the analyst closed the alert

- Please note that whitelisted entities will only appear if the alert has been approved.

Example of Approved Alert Callback:


```
{
  "businessMessageId": "mm1_ID_BD_s5wlrsdsdfsdafsasdssd488sd5fk4581",
  "result": "Approve",
  "recommendedResolution": "country_differs",
  "creationDate": "2023-11-23T14:59:52.261Z",
  "comment": "<p>TestNote</p>",
  "whitelisted": [
            {
                "entityType": "ORIGINATOR_NAME"
            }]
}
```


- **Blocked Matches** - details of the blocked match that caused the analyst to block the alert, meaning, the true positive match details. Please note these details will appear only if the alert has been blocked.

Example of Blocked Alert Callback


```
{
  "businessMessageId": "mm1_ID_BD_s5wlrsdsdfsdafsasdssd488sd5fk4581",
  "result": "Block",
  "recommendedResolution": "country_differs",
  "creationDate": "2023-11-23T14:59:52.261Z",
  "comment": "<p>TestNote</p>",
  "blockedMatch": [
      {
        "entityType": "DEBTOR",
```

```
        "entityId": "1123dd",
        "list": "OFAC",
        "listCategory": "Sanctions",
        "recordID": "1234aaa"
      }
  ]
}
```


2.1.3.13. Transaction Screening Generic API Screen & Monitor Endpoint

Element	URL / Value
Endpoint	/screening/transaction/generic/check-and-monitor
Method	POST

Screening via this endpoint will effectively both screen the transaction, and upload it into the Transaction Monitoring object storage to be used for transaction monitoring analysis.

In terms of login, request structure and success and error response, this endpoint is identical to the main Transaction Screening Generic API.

The differences are in the following validations:

- transactionAmount field is mandatory;
- transactionCurrency field is mandatory;
- transactionDate field is mandatory and should comply with the pattern yyyy-MM-dd'T'HH:mm:ss (configurable - ref. 1).

2.1.4. Transaction Screening - SWIFT MT API

Important Note: **Not** for New Deployments.

2.1.4.1. API Format

- The Sanction Screening API uses JSON encoded as UTF-8.
- The body of POST requests must be a JSON object with Content-Type: application/json; charset=UTF-8.
- The body of responses is always a JSON object and always provided with Content-Type: application/json; charset=UTF-8.

2.1.4.2. Request

The **POST** method allows you to send one transaction to a single API call.

2.1.4.2.1. Attributes Listing

Table 2: Attributes , Description and if Mandatory (M) or Optional (O)

Attributes	Description	Required
transaction id	Unique identification number for the transaction	M
senderBIC	Sender's BIC	O
receiverBIC	Receiver's BIC	O
senderReference	Sender's Reference	O

Table 2: *Attributes , Description and if Mandatory (M) or Optional (O) (continued)*

Attributes	Description	Required
bankOperationCode	Bank Operation Code	O
interbankValueDate	Interbank Value Date	M
interbankCurrency	Interbank Currency	M
interbankSettlementAmount	Interbank Settlement Amount	M
usdEquiv	USD Settlement Amount Equivalent	O
originatorName	Originator Name	M
originatorBIC	Originator BIC / Account Number	O
originatorDateOfBirth	Originator Date of Birth	O
originatorAddress	Originator Address Line 1	O
originatorCountry	Originator Country	O
originatorBankName	Originator Bank Name	M*
originatorIBankBIC	Originator Bank BIC / Account Number	O
originatorBankAddress	Originator Bank Address Line 1	O
originatorBankCountry	Originator Bank Country	O
orderingInstitutionName	Ordering Institution Name	M*
orderingInstitutionBIC	Ordering Institution BIC / Account Number	O
originatorInstitutionAddress	Ordering Institution Address Line 1	O
orderingInstitutionCountry	Ordering Institution Country	O
senderCorrespondentName	Sender's Correspondent Name	M*
senderCorrespondentBIC	Sender's Correspondent BIC / Account Number	O

Table 2: *Attributes , Description and if Mandatory (M) or Optional (O) (continued)*

Attributes	Description	Required
senderCorrespondentAddress	Sender's Correspondent Address Line 1	O
senderCorrespondentCountry	Sender's Correspondent Country	O
receiverCorrespondentName	Receiver's Correspondent Name	M*
receiverCorrespondent BIC	Receiver's Correspondent BIC / Account Number	O
receiverCorrespondentAddress	Receiver's Correspondent Address Line 1	O
receiverCorrespondent Country	Receiver's Correspondent Country	O
thirdReimbursementInstitutionName	Third Reimbursement Institution Name	M*
thirdReimbursementInstitutionBIC	Third Reimbursement Institution BIC / Account Number	O
thirdReimbursementInstitutionAddress	Third Reimbursement Institution Address Line 1	O
thirdReimbursementInstitutionCountry	Third Reimbursement Institution Country	O
intermediaryInstitutionName	Intermediary Institution Name	M*
intermediaryInstitutionBIC	Intermediary Institution BIC / Account Number	O
intermediaryInstitutionAddress	Intermediary Institution Address Line 1	O
IntermediaryInstitutionCountry	Intermediary Institution Country	O
beneficiaryName	Beneficiary Name	M
beneficiaryAccountNumber	Beneficiary Account Number	O
beneficiaryDateOfBirth	Beneficiary Date of Birth	O
beneficiaryAddress	Beneficiary Address Line 1	O
beneficiaryCountry	Beneficiary Country	O
transactionReference	Transaction Reference	O

Table 2: *Attributes , Description and if Mandatory (M) or Optional (O) (continued)*

Attributes	Description	Required
transactionData	Other information that can be displayed in the IC	O
sourceData	Other information that can be displayed in the IC	O
UniqueEndToEndTransactionReference	Unique end-to-end transaction reference	O
profile	The profile name that indicates which set of screening settings to apply to the request	O

Note: M* : The corresponding fields are mandatory only if the relevant entity type exists.

Note: If the profile name parameter is not sent, then the default screening settings will apply. If an incorrect profile name is sent, the screening request will show an error message.

2.1.4.2.2. JSON Message Example

```
{
  "transactionId": "767267df-7791-4101-93df-059f092e039e123",
  "generic": {
    "senderBIC": "IGLUGB2LAXXX",
    "receiverBIC": "BOFAPH2XXXXX",
    "senderReference": "923769312-CAC597",
    "interbankValueDate": "2021-12-16",
    "interbankCurrency": "PHP",
    "interbankSettlementAmount": "7138670.0",
```

```
    "usdEquiv": "142994.70"
  },
  "originator": {
    "name": "bin laden",
    "address": "House 123 Jeddah",
    "country": "SA",
    "dateOfBirth": "1923-12-02",
    "bic": "7678623746723643279"
  },
  "originatorBank": {
    "name": "Habib Bank",
    "bic": "TRWIGB2L",
    "address": "House 123 London",
    "country": "GB"
  },
  "orderingInstitution": {
    "name": "Barclays Bank",
    "bic": "BARCGB22",
    "address": "Street 987 London",
    "country": "GB"
  },
  "senderCorrespondent": {
    "name": "BNP ISIS",
    "bic": "BNPAFRPH",
    "address": "House ABC Paris",
    "country": "FR"
  },
  "receiverCorrespondent": {
    "name": "CITI Bank",
```

```
    "bic": "CITIFR2L",
    "address": "Street ABC Iran",
    "country": "FR"
  },
  "thirdReimbursementInstitution": {
    "name": "Citi Bank Shanghai",
    "bic": "CITICN2L",
    "address": "Street XYZ Shanghai",
    "country": "IR"
  },
  "intermediaryInstitution": {
    "name": "Payoneer",
    "bic": "PAYNUS33",
    "address": "Street XYZ Shanghai",
    "country": "IR"
  },
  "beneficiary": {
    "name": "Lewis Hamilton",
    "address": "Street ABC Shanghai",
    "country": "IR",
    "dateOfBirth": "1923-12-02",
    "accountNumber": "7678623746723643278"
  },
  "transactionReference": "Free text",
  "senderToReceiverInformation": "Free text",
  "transactionData": {
    "key1": "value1"
  },
  "sourceData": {
```

```
    "key2": "value2"
  }
}
```

2.1.4.3. Response Attributes

For more information, refer to Response Attributes

2.2. Customer Screening

2.2.1. How it Works

The **Thetaray Screening Solution** provides **real time screening** for new or existing customers. The Screening Solution enables name matching across data held on customers and matches against Sanctions lists, watchlists, PEP and Adverse Media lists, as well as private lists.

2.2.2. How to Connect

API Format

- The Customer Screening API uses JSON encoded as UTF-8.
- The body of POST requests must be a JSON object with Content-Type: application/json; charset=UTF-8.
- The body of responses is always a JSON object and always provided with Content-Type: application/json; charset=UTF-8.

Login API

Authentication is based on calling an API to create an access token. The API requires a client secret that will be provided to you by our customer care team at onboarding to the service. You will use the access token to call the screen transaction API Request Attributes.

The JWT token is valid for 5 minutes and the expirationInMinutes will always show “5” as it is relative to the time the token was first created. However, ThetaRay utilizes a 5 minute cache option for multiple logins therefore will not generate a new token for the duration of the cache.

The recommended way to implement authentication for screening requests is the following:

- When sending a screening request, first decode the “time to live” timestamp from the actual JWT token.
- If the current time is larger then the “time to live” then generate a request for the creation of a new token before sending the screening request.

Please note that if the token has only a few milliseconds to expiry, then the screening request sent with this token might receive a 401 authentication error and will need to be resent with a valid token

2.2.2.1. Request Attributes

```
Endpoint:POST /security/accessToken
{
"clientSecret": "<credentials provided by ThetaRay>"
}
```

2.2.2.2. Example Login API Response Body – Response Payload

```
{
    "token": "<Token>",
```

```
    "expirationInMinutes": 30
}
```

2.2.3. Real Time Customer Screening API

2.2.3.1. Request Codes

Element	URL
Endpoint	/screening/customer/check
Method	POST

Field Name	Sub Field 1	Sub Field 2	Type	Required	Validation	Screened
requestId			string	M	min 20 max 200 characters, should be unique amongst all requests	no

Field Name	Sub Field 1	Sub Field 2	Type	Required	Validation	Screened
requestData	id		string	M	min 20, max 200 characters, unique customer identification	no
	type		string	M	Has to be one of the following types: individual, company, merchant, agent	no
	fullName		string	M	Relevant for all types, should be between 1 - 255 characters	yes
	addresses		string	M	Relevant for all types	yes
	sex		string	O	Relevant for individual, agent Acceptable values: m / M / f / F	yes
	nationalities		Array of strings	O	Relevant for individual, agent, country code ISO 3166, Alpha-2 code	yes
	dateOfBirth		date	O	Relevant for individual, agent, date can be YYYY-MM-DD or YYYY or YYYY/YYYY	yes
	placeOfBirth		string	O	Relevant for individual, agent, country code ISO 3166, Alpha-2 code	yes
	identityDocuments	number	Array of objects	O	Relevant for all types. Array of objects that each contain a "number" field	yes
	bic		string	O	Relevant for company, 8-11 characters	yes
	datesOf Registry		Array of dates	O	Relevant for company, merchant. Date can be YYYY-MM-DD or YYYY or YYYY/YYYY	yes
	placesOf Registry		List of strings	O	Relevant for company, merchant, country code ISO 3166, Alpha-2 code	yes
sourceData			string to string	O	key-value pairs that will be displayed on the alert	no
profile			string	O		no
saveForRescreening			boolean	O		no

Field Name	Sub Field 1	Sub Field 2	Type	Required	Validation	Screened
rescreeningProfiles			Array of strings	O		no

Example - Individual Request Code

```
{
"requestId": "582d1c1c-57c8-496a-b6f4-17c8212c552a",
"requestData": {
  "id": "0ac8ba6f-2ed2-453f-9073-20b8481f40ad",
  "type": "individual",
  "fullName": "John Doe",
  "identityDocuments": [{
      "number": 123456789
}],
  "nationalities": [
    "US"
   ],
  "addresses": [
    "New York, USA"
   ]
  },
  "profile": "DEFAULT",
  "saveForRescreening": true,
  "rescreeningProfiles": [
    "CUSTOM1",
    "CUSTOM2"
  ]
}
```

Example - Company Request Code

```
{
    "requestId": "myRequestId",
    "requestData": {
        "type": "company",
        "id": "id3456789012345678 90",
        "addresses":["North Pasdaran Street, Tehran, Iran"],
        "fullName":"ANSAR BANK",
        "bic": "ANSBIRTH",
        "placesOfRegistry":["IR"],
        "datesOfRegistry": ["1990-01-01"]
    }
}
```

2.2.3.2. Validations

- To screen the same customer multiple times, the "id" should be the same (while requestId should be different for each request)
- The API request needs to include either type "individual" or type "company"
- Fields "id", "fullName" and "address" are mandatory for each type
- Countries should be provided in ISO 3166-alpha 2 format
- Field "sex" can have one of "m", "M", "f", "F" values
- If the profile name parameter is not sent, then the default screening settings will apply. If an incorrect profile name is sent, the screening request will show an error message
    - true: if ongoing monitoring is enabled, otherwise false
    - Note: Can't be set to true, if ongoing monitoring is disabled

- saveForRescreening - this attribute sets whether or not the customer sent for screening will be saved for rescreens (Ongoing Monitoring). Please see the 'Ongoing Monitoring' section for more information.
  - If Ongoing Monitoring is enabled, this parameter will be set to "true" by default
  - If Ongoing Monitoring is not enabled, this parameter will be "false" by default, and will fail validation if set to "true"
- rescreeningProfiles - this attribute indicates which profiles to rescreen the customer for Ongoing Monitoring
  - Can accept multiple profiles, and the customer will be rescreened against all
  - If empty, the rescreen will save the profile from profileName attribute
  - If profileName is also empty, the rescreen will save profile "default"

2.2.3.3. Response Codes

Note: The response includes the fields mentioned below, by default. If you would prefer to receive only "MATCH" in the response, without additional fields, please request this from Customer Support.

2.2.3.3.1. Success Response Codes

Response Fields:

- checkResult - boolean, indicates MATCH or NO MATCH
- statusPollingUrl - contains a URL for polling the status of the alert (see Polling API Request & Response section for more details)
- entityType - contains the screened entity type, e.g. Individual, Company, etc.
- score - the match score for the entity
- list - list dataset name

- dateOfPublication - the recent datetime the actual list was updated, according to the official publication date of the list
- dateOfUpload - the recent datetime the list was imported into screening engine

Response Body


```
{
    "result": {
        "checkResult": "MATCH",
        "statusPollingUrl": "<statusPollingUrl>",
        "matchResults": [
            {
                "entityType": "INDIVIDUAL",
                "score": "0.84",
                "list": "AUSTRALIA",
                "dateOfPublication": "2023-03-20T22:44:36",
                "dateOfUpload": "2023-03-29T14:45:11.721"
            },
            {
                "entityType": "INDIVIDUAL",
                "score": "0.82",
                "list": "WORLD LEADERS",
                "dateOfUpload": "2023-03-29T00:46:44.835"
            }
        ]
    }
}
```


2.2.3.3.2. Error Responses

Http Status	Description
400	Invalid Request - the request contains validation errors. For a full list of messages, please see section below
401	Unauthorized Request
422	Request cannot be processed - entity type contains an inapplicable field. For a full list of messages, please see section below
500	Internal Server Error
503	Service Unavailable

Error Messages

Message	HTTP Status
Request ID does not conform, shall be in the range of 20-200 characters	400
Request ID value is already used	400
Invalid Entity Type, check possible entity type values	400
Full Name does not conform, shall be in the range of 1-255 characters	400
Gender is not applicable, shall not be provided if Entity Type is different from: Individual, Agent	422
Dates of Birth are not applicable, shall not be provided if Entity Type is different from: Individual, Agent	422
Dates of Registry are not applicable, shall not be provided if Entity Type is different from: Organization, Merchant	422
Countries of Registry are not applicable, shall not be provided if Entity Type is different from: Organization, Merchant	422
Invalid Date, shall be a valid date value/format	422
Invalid Country, shall be a valid ISO 3166-1 Alpha-2 value	422

2.2.3.4. Alert Status Response - Polling API

Returns the status of the customer that has an alert populated in the Investigation Center that an analyst has approved or blocked.

The Polling API provides the status of a screening alert via polling. The API can be polled from alert creation, which is done via an asynchronous process following the screening request.

The URL for polling is returned in the screening response (synchronous to the request). The polling response has a five-minute cache.

The Login API endpoint should be used before polling in order to authenticate. {link to Login API section 2.1.2}

There are two endpoints available for the Polling API - new customers should always use the “extended” Polling API, and existing customers are welcome to integrate it as well. The previous endpoint will continue to be supported, but not enhanced.

2.2.3.4.1. Extended Polling API

Element	URL
Endpoint	/screening/investigation/status/extended/{requestId}
Method	GET

Success Response

The success response pertains to two states: either the alert is under investigation by the analyst or the alert has been closed.

Alert under investigation success response:

```
{
 “result”: “Alert is under investigation”
}
```

Alert closed success response includes the following fields:

- **Result** - the result of closing the alert, will be either “Alert approved” or “Alert rejected”

- **Recommended Resolution** - an additional resolution code that the analyst selects when closing the alert
- **Comment** - the note the analyst fills in when closing the alert
- **Whitelisted** - a list of entity types that were whitelisted when the analyst closed the alert if the analyst whitelisted any entities. Please note that whitelisted entities will only appear if the alert has been approved.

Example of Approved

```
{
    "result": {
        "result": "Alert approved",
        "recommendedResolution": "country differs",
        "comment": "<p>tester</p>",
        "whitelisted": [
            {
                "entityType": "ORIGINATOR_NAME"
            }
        ]
    }
}
```

- **Blocked Matches** - details of the blocked match that caused the analyst to block the alert, meaning, the true positive match details. Please note these details will appear only if the alert has been blocked.

Example of Blocked Alert Callback

```
{
    "result": {
        "result": "Alert rejected",
```

```
        "recommendedResolution": "exceeds_risk_appetite",
        "comment": "test",
        "blockedMatches": [
            {
                "list": "OFAC SDN",
                "listCategory": "SANCTION",
                "entityType": "BENEFICIARY_NAME",
                "recordId": "19609"
            }
        ]
    }
}
```


Error Responses

HTTP Status	Description
403	Unauthorized request
404	Alert creation failed
404	Alert cannot be found for this request ID
500	Internal server error
503	Service unavailable

2.2.3.4.2. Polling API-Old version

Element	URL
Endpoint	/screening/investigation/status/{requestId}
Method	GET

Success Response

```
{
  "Transaction approved/Transaction rejected/Transaction is under investigation"
}
```

Error Responses

HTTP Status	Description
403	Unauthorized request
404	Alert cannot be found for this request ID
500	Internal server error
503	Service unavailable

2.2.3.5. Alert Status Response - Callback API

The callback API sends a request from the ThetaRay screening service to a dedicated URL whenever a screening alert is closed by the analyst in the Investigation Center.

Supported methods of authentication:

1. No authentication - only a URL needs to be provided to ThetaRay Support
2. API key authentication - a URL and an API Key need to be provided. The header in the callback will include "apikey" and the value of the key provided.
3. JWT authentication - in addition to the URL, another URL needs to be provided in order to generate the JWT token, as well as the body of the request for generating the token (in key-value pairs), the header in the callback for this authentication method will be:

   Authorization: Bearer <jwt>

The following attributes are sent in the body:

- **businessMessageId** - the transactionId from the original screening request
- **result** - the alert resolution, can be either "approve" or "block"
- **recommended resolution** - additional resolution code that is selected by the analyst when closing the alert, for example, "SAR worthy"
- **creationDate** - the creation datetime for the callback call
- **comment** - the analyst has an option to add a comment in the Investigation Center when closing the alert
- **whitelisted** - a list of the entity types that were whitelisted when the analyst closed the alert. Please note that whitelisted entities will only appear if the alert has been approved.

Example of Approved Alert Callback

```
{
  "businessMessageId": "mm1_ID_BD_s5wlrsdsdfsdafsasdssd488sd5fk4581",
  "result": "Approve",
  "recommendedResolution": "country_differs",
  "creationDate": "2023-11-23T14:59:52.261Z",
  "comment": "<p>TestNote</p>",
  "whitelisted": [
            {
                "entityType": "ORIGINATOR_NAME"
            }]
}
```

- **Blocked Matches** - details of the blocked match that caused the analyst to block the alert, meaning, the true positive match details. Please note these details will appear only if the alert has been blocked.

Example of Blocked Alert Callback


```
{
  "businessMessageId": "mm1_ID_BD_s5wlrsdsdfsdafsasdssd488sd5fk4581",
  "result": "Block",
  "recommendedResolution": "country_differs",
  "creationDate": "2023-11-23T14:59:52.261Z",
  "comment": "<p>TestNote</p>",
  "blockedMatch": [
      {
        "entityType": "DEBTOR",
        "entityId": "1123dd",
        "list": "OFAC",
        "listCategory": "Sanctions",
        "recordID": "1234aaa"
      }
  ]
}
```


2.2.4. Bulk Customer Screening API

The Bulk Customer Screening API allows the screening of multiple customers in one API call. This functionality should be used when it's necessary to pass a batch of customers for screening. Please note that this API will always perform a "full screen" against all of the entities contained in the screening lists, meaning that any matches (even if previously raised) will be found, returned and alerted upon.

Note: The default bulk screening size limit is 50 alerts at each attempt. This default size can be modified if required, by contacting ThetaRay Support for assistance.

Note: On Mon/Rescreening also uses the Bulk Customter Screening API, so please be aware that the same default /configuration settings are applicable to that feature as well.

Method & Endpoint

HTTP method	POST
Endpoint	/screening/customer/bulk

Request Body Example

```
{
  "bulk":
      [
          {
      "requestId": "582d1c1c-57c8-496a-b6f4-
      17c8212c552a",
      "requestData":
      {
          "id": "0ac8ba6f-2ed2-453f-9073-20b8481f40ad",
          "type": "individual",
          "fullName": "John Doe",
          "nationalities":
          [
```

```
        "US"
        ],
        "addresses":
        [
        "New York, USA"
        ]
    },
        "profile": "DEFAULT",
        "saveForRescreening": true,
        "rescreeningProfiles":
        [
            "CUSTOM1",
            "CUSTOM2"
        ]
    },
    {
        "requestId": "5033e607-d962-4b61-b4f3-
        3872d373ca98",
        "requestData":
        {
        "id": "1bf14196-0af2-406d-90ad-fdec53989207",
        "type": "individual",
        "fullName": "Jane Doe",
        "nationalities":
    [
        "US"
        ],
        "addresses":
        [
```

```
            "Los Angeles, USA"
            ]
            },
            "profile": "DEFAULT",
            "saveForRescreening": true,
            "rescreeningProfiles":
            [
                "CUSTOM1",
                "CUSTOM2"
            ]
        }
    ]
}
```

Validations

Validations	Same as for Real-time customer screening + • "bulk" field can't be empty • "bulk" field can't be more than the max request size (default 50) • all the "requested" fields must be unique

Response Body Example

```
{
    "result": {
        "bulk": {
            "582d1c1c-57c8-496a-b6f4-17c8212c552a": {
```

```
                "checkResult": "MATCH",
                "statusPollingUrl": "<statusPollingUrl>",
                "matchResults": [
                    {
                        "entityType": "INDIVIDUAL",
                        "score": "0.82",
                        "list": "INTERPOL"
                    },
                    {
                        "entityType": "INDIVIDUAL",
                        "score": "1.00",
                        "list": "High-Risk Country"
                    }
                ]
            },
            "5033e607-d962-4b61-b4f3-3872d373ca98": {
            "checkResult": "NO_MATCH"
            }
        }
    }
}
```


Response Codes

Response Codes	200, 400, 401, 403

Note: All of the error responses noted in the previous section (Real Time Customer Screening) are relevant for the Bulk Customer Screening API as well.

2.2.5. Ongoing Monitoring for Customer Screening

2.2.5.1. Overview

Ongoing Monitoring, or Periodic Screening, refers to the process of rescreening customers periodically in order to continuously monitor them from a Screening perspective. The ThetaRay Screening Solution offers an accessible and flexible answer for this regulatory requirement. Please note that the Ongoing Monitoring feature is a licensed add-on, please contact Customer Support in order to purchase and enable.

Customers sent for screening via the Real Time API (Customer Screening) or the Bulk Screening API (Customer Screening) will be by default, saved internally and automatically rescreened according to the selected cadence. The cadence is set via the Screening Settings file, provided by and uploaded through Customer Support.

In order to support Ongoing Monitoring, a number of new API's have been developed and are described below. Additional fields have been added to the Real Time API and are described above in section Customer Screening.

2.2.5.2. Profile API

Profile API- sending a profile name in this API will perform a full screen of all of the customers in this profile.

Method & Endpoint

HTTP method	POST
Endpoint	/screening/customer/bulk/profiles

Request Body Example

```
{
    " profiles": [
        "profile_name_1",
        "profile_name_2",
        ...
    ]
}
```

Validations

Validations	Profiles names should be as configured in the Screening Settings file

Response Body Example

```
{
    "result": {
    Application Integration
    "operationId" : "34567-23456-23456-23456"
    }
}
```

Response Codes

Response Codes	200, 400, 401, 403

2.2.5.3. Get Customer API

The Get Customer API allows for retrieval of individual customer data from the list of customers saved internally for ongoing monitoring. Please use this API if you would like to know which customer data is being automatically rescreened. Updating customer data (changing any attribute such as name, address, or screening profile, etc.) is achieved through the Real Time API or the Bulk API and requires sending over the same customerId. The new data will overwrite the existing data.

Method & Endpoint

HTTP method	GET
Endpoint	/customer/{customerId}

Response Body Example


```
{
    "result":
        {
        "customerId": "fb603429-9db2-45bf-b2bd-aa3922f23430",
        "customerData":
                {
                "id": "0ac8ba6f-2ed2-453f-9073-20b8481f40ad",
                "type": "individual",
                "fullName": "John Doe",
                "nationalities":
                [
            "US"
            ],
            "addresses":
            [
            "New York, USA"
```

```
            ]
        },
        "createdOn": "2023-05-07T17:07:27",
        "updatedOn": "2023-05-07T17:07:29"
    }
}
```

Response Codes

Response Codes	200, 401, 403, 404

2.2.5.4. Delete Customer API

The Delete Customer API allows for the deletion of individual customer data. Please use this function if you would like to erase a customer from the database in order for them not to be rescreened automatically.

Method & Endpoint

HTTP method	DELETE
Endpoint	/customer/{customerId}

Response Codes

Response Codes	204, 400, 401, 403

2.2.5.5. Rescreen Results Callback API

Customers are rescreened according to the selected cadence, with the delta screening functionality (screening against what has changed in the lists from a certain datetime). An automatic backend process gathers the relevant customers according to the relevant cadence, and runs an

offline rescreening process for all of the customers. The matches found in the rescreen process will become alerts in the Investigation Center. The rescreen process is also audited as a rescreen, along with the screening results. When the rescreening process completes, a response in the form of a callback URL is sent in order to provide the client with the indication that this process has been completed, with the relevant matches that have been found.

2.2.5.5.1. Enabling Rescreen Results Callback

In order to receive the rescreen results please send the following parameters to Customer Support:

- rescreen.response.callback.url - The URL to send the callback to
- rescreen.response.callback.apikeyHeader - The header name to send the API key
- rescreen.response.callback.apikey - The API key

The callback will by default, send a maximum of 50 matches (batch size). Whilst we recommend to use the default settings, the batch size can be adjusted via Customer Support.

2.2.5.6. Rescreen Results Callback Attributes

General Information on the rescreen process:

1. Timestamp - the time the request containing rescreen results is sent
2. Number of customers rescreened
3. Number of failed rescreens
4. Number of matches found
5. Profile
6. Cadence

Match Details:

1. Customer ID
2. Match Score
3. List name
4. Date of publication - date the list was updated officially (where applicable)
5. Date of upload - date the list was updated into the screening engine
6. Polling URL - polling URL for the alert status

JSON Example


```
{
    "operationId": "a2c81526-3eb2-4a45-b138-588c61481978",
    "timestamp": 1620570151,
    "customersRescreened": 100,
    "failedRescreens": 0,
    "matchesFound": 2,
    "profile": "DEFAULT",
    "cadence": "DAILY",
    "matchDetails": [
        {
            "customerId": "4c286d22-1c75-4dc1-92eb-a74b0a256515",
            "score": "0.85",
            "list": "US",
            "dateOfPublication": "2022-03-01",
            "dateOfUpload": "2022-04-15",
            "pollingUrl": "<pollingUrl>"
        },
        {
            "customerId": "1ed1671a-16b1-438e-bb97-537e43ad4ae7",
            "score": "0.97",
```

```
            "list": "UN",
            "dateOfPublication": "2022-02-15",
            "dateOfUpload": "2022-04-15",
            "pollingUrl": "<pollingUrl>"
        }
    ]
}
```

2.2.6. Customer Data Examples

Table 3: *Individual Data*

#	Field Name	Field Display	Mandatory	TR Data Type	Example
1	dataID	Unique Identifier of the data record.	Yes	String	AZawahiri9989812
2	entityTYpe	Type of Named-Entities: shall be an individual	Yes	String	Individual
3	fullName	Full Name	Yes	String	Ayman al-Zawahiri
4	sex	Sex for an individual.	No	String	Male
5	nationalities	Nationality for an individual, can be many.	No	ISO country code	British
6	datesOfBirth	Date of birth for an individual, can be many.	No	dd/mm/yyyy	20-08-1990
7	placesOf Birth	Place of birth for an individual, can be many.	No	ISO country code	GB
8	addresses	Address for an individual, can be many.	Yes	String	123 street, London
9	titles	Individual's position or job, can be many.	No	String	Mr
10	identityDocumentNumber	Identity Document Information of an individual, can be many.	No	String	'89898989898'
11	contactInformation	Contact information of an individual, can be many.	No	String	AZ@email.com

Table 4: *Company Data*

#	Field Name	Field Display	Mandatory	TR Data Type	Example
1	dataID	Unique Identifier of the data record.	Yes	String	HabibBank8789798798
2	entityTYpe	Type of Named-Entities: shall be a company	Yes	String	Company
3	fullName	Full Name	Yes	String	Ayman al-Zawahiri
4	bic	Business Identifier Code (BIC) data to match (8-11 characters)	No	ISO BIC format	BARCGB22
5	datesOfRegistry	This is the date of registration for a organization, can be many.	No	dd/mm/yyyy	02-03-2010
6	placesOfRegistry	This is the place of registration for an organization, can be many.	No	ISO country code	GB
7	addresses	addresses	Yes	String	123 street, London
8	identityDocuments	Identity Document Information of an company, can be many.	No	String	'1679687678
9	contactInformation	Contact information of an organization, can be many.	No	String	habibbank@email.com
10	additionalInformation	Additional information about an organization.	No	String	Banking

Table 5: *Merchant Data*

#	Field Name	Field Display	Mandatory	TR Data Type	Example
1	dataID	Unique Identifier of the data record.	Yes	String	Taliban0909679676
2	entityTYpe	Type of Named-Entities: shall be a company	Yes	String	Merchant
3	fullName	Full Name	Yes	String	Taliban0909
4	bic	Business Identifier Code (BIC) data to match (8-11 characters)	No	ISO BIC format	BARCGB22
5	datesOfRegistry	This is the date of registration for a organization, can be many.	No	dd/mm/yyyy	02-03-2010
6	placesOfRegistry	This is the place of registration for an organization, can be many.	No	ISO country code	GB
7	addresses	addresses	Yes	String	123 street, London

Table 5: *Merchant Data (continued)*

#	Field Name	Field Display	Mandatory	TR Data Type	Example
8	identityDocuments	Identity Document Information of an company, can be many.	No	String	'1679687678
9	contactInformation	Contact information of an organization, can be many.	No	String	habibbank@email.com

Table 6: *Agent Data*

#	Field Name	Field Display	Mandatory	TR Data Type	Example
1	dataID	Unique Identifier of the data record.	Yes	String	AgentAZawahiri87967932e
2	entityTYpe	Type of Named-Entities: shall be a company	Yes	String	Agent
3	fullName	Full Name	Yes	String	A Zawahiri
4	sex	Sex for an individual.	No	String	Female
5	nationalities	Nationality for an individual, can be many.	No	ISO country code	British
6	datesOfBirth	Date of birth for an individual, can be many.	No	dd/mm/yyyy	01-10-1971
7	placesOfBirth	Place of birth for an individual, can be many.	No	ISO country code	GB
8	addresses	Address for an individual, can be many.	Yes	String	123 street, London
9	titles	Individual's position or job, can be many.	No	String	Mrs
10	identityDocumentNumber	Identity Document Information of an	No	String	6576576765

Table 6: *Agent Data (continued)*

#	Field Name	Field Display	Mandatory	TR Data Type	Example
		individual, can be many.			
11	contactInformation	Contact information of an organization, can be many.	No	String	habibbank@email.com

2.3. Screening Configurations

2.3.1. Private List Secure Upload

This section details the API that enables our screening customers to upload their own private custom screening data lists.

2.3.1.1. Overview

A private list refers to any list that is unique and not publicly available. The private list functions as any other list, and will generate matches on the screening requests. The private list can apply to both Customer Screening and Transaction Screening. Examples of private lists include, but are not limited to:

- Internal employee lists
- Risky customer / entity list that has been compiled by internal investigation
- Lists of risky actors distributed by a regulator

Guidelines:

- If using screening profiles, one list can be uploaded per profile

- Each new list upload overwrites the previous content, so please make sure to keep a record of the full private list and deactivate irrelevant records
- The private list is encrypted during upload, in order to protect PII data that may be included. Encryption and encryption keys are managed by ThetaRay accordingly, and will be unique per profile

For more information regarding the private list specifications and formats, please refer to the ***Screening Overview*** document.

Please note, the upload API is asynchronous, and therefore uploading the list will return a task ID to be polled for status.

2.3.1.2. Login API

Authentication is based on calling an API to create an access token. The API requires a client secret that will be provided to you by our customer care team at onboarding to the service. You will use the access token to call the screen transaction API Request Attributes.

Request Attributes

```
Endpoint:POST /security/accessToken
{
"clientSecret": "<credentials provided by ThetaRay>"
}
```

Example Login API Response Body – Response Payload

```
{
    "token": "<Token>",
    "expirationInMinutes": 5
}
```

2.3.1.3. Private List Upload

This API will validate the uploaded file and synchronously return any validation errors. If the file is valid, the API will return a success message with a task ID to be polled, as described in the Check Upload Status API in the following section.

Note: The profile attribute is optional and should only be used if profiles are configured.

HTTP request	
Endpoint list	/screening/config/private-list/upload?profile=<profile>
Method	POST
Request Body	upload_file

Success Response Body

```
{
   "result":{
      "message":"File 'example.csv' has been uploaded successfully, private list 'PRIVATE LIST' has been saved",
      "taskId":"<taskId>"
   }
}
```

Error Response Body

```
{
  "result": <error information>
}
```

HTTP Response Codes

Code	Response message
200	Response body as described in example above
400	Validation errors: "Invalid input. Please adjust {validation} and re-upload"
401	Authorization errors
403	Access is forbidden
500	Unexpected errors

Example Upload via curl

```
curl --location '<baseURL>/screening/config/private-list/upload/'
    --header 'Authorization: Bearer <Bearer>' \
    --form 'upload_file=@"/path/to/file/example.csv"'
```

Note: Be aware , that depending on variable factors such as file size or amount of records, the upload process can in some instances take up to 20 minutes or more

2.3.1.4. Checking Upload Status of Private List

As the private list upload API is asynchronous, the Check Upload Status API should be used in order to determine if the private list has been uploaded successfully or not.

Note: In case of upload failure the status code might be 200, and the failure is indicated in the "status" field of the response

HTTP request	
Endpoint list	/screening/config/private-list/upload-status/{taskId}
Method	GET
Request Body	N/A

Success Response Body

```
{
    "taskID": "9340a7d73ef9499f1adff6e76b01e39d",
    "type": "matching",
    "dataset": "COMPANY 1",
    "started": "2020-04-17T14:30:15.321+01:00",
    "ended": "2020-04-17T14:33:20.321+01:00",
    "status": "in progress",
    "totalCount": "1000",
    "duration": "PT3M5S"
}
```

Success Response Example

Before providing an example of a success response you will notice that the success response body code detailed above includes a ***'Type'*** key. What are type keys and how are they used?

'Type' keys are related to asynchronous activities and can have 3 different values as follows:

- **"matching"** - when a matching task is ongoing, it identifies the source Dataset.
- **"importing"** - when an importing task is ongoing, it identifies the related Dataset
- **"Indexing"**- when an indexing task is ongoing, it identifies the related Dataset

```
{
 {
   "result": {
     "response": {
       "type": "indexing",
       "dataset": "PRIVATE LIST",
       "started": "2023-09-14T09:44:39.309212225",
       "ended": "2023-09-14T09:44:50.213969646",
       "status": "completed",
       "processed": "3",
       "duration": "PT10S",
       "taskID": "TSK1205169696-1694684678643"
     }
   }
 }
```


Error Examples

As an aid to understanding error responses, 2 error examples, invalid profile and invalid header are provided below.

Example #1 - invalid profile


```
{
    "result": {
        "errors": "Invalid input. Please adjust 'Configuration for screening profile TEST not found' and reupload the file"
    }
}
```


Example #2 - invalid header columns

```
{
    "result": {
        "errors": "Invalid input. Please adjust 'Task ID: 'TSK-1960276399-1719913923498' Value: 'bic' has error: 'INVALID HEADER VALUE', Task ID: 'TSK-1960276399-1719913923498' Value: 'lei' has error: 'INVALID HEADER VALUE'' and reupload the file"
    }
}
```

HTTP Response Codes

Code	Response message
200	Response body as described in example above
400	"{error}. Please try again"
401	Authorization errors
403	Access is forbidden
500	Unexpected errors

Example of Check Status via curl

```
curl --location '<baseURL>/screening/config/private-list/upload-status/<TaskId>' \>
--header 'Authorization: Bearer <Bearer>'
```

2.4. Externalizing Screening Match Results - API

The screening match results externalization API empowers ThetaRay clients, utilizing their own case manager, to gain access to screened activity. This functionality is relevant for all Transaction Screening and Customer Screening API's.

By externalizing screening match results, clients can efficiently incorporate this information into their existing workflows and decision-making processes.

2.4.1. Externalization Configuration

To enable screening match results externalization, clients must configure specific properties. These properties customize the process for integration with their case management systems. For assistance with property configuration or URL setup for externalization, clients should contact ThetaRay's Customer Support team.

Configuration Properties:

1. **customer.external.target.url** (Required): This property specifies the URL to which the JSON alerts containing screening match results will be dispatched. Clients can define or update this URL by consulting with ThetaRay's Customer Support team.
2. **customer.external.target.auth.header**: Optionally, clients can provide authentication headers to secure the communication between their application and the external URL.
3. **customer.external.target.auth.key**: This property complements the authentication process, allowing clients to provide the necessary authentication key for the environment in use.
4. **customer.external.target.retries (Default: 2)**: This property defines the number of retry attempts that will be made in case of communication failures with the external URL. The default value is set to 2, but clients have the flexibility to adjust it according to their requirements or bypass it entirely.

2.4.2. JSON Structure and Example

In this section, a breakdown of the JSON structure used for screening match results externalization is provided.

The screening match results externalization comprises several key elements:

- ***Screening Data*** - This core data is transmitted to the screening engine for match queries and is recorded as an event in the audit log
- ***Source messages*** - These unaltered original messages are included as proof of the original request

- ***ListInfoData*** - This section consolidates screening outcomes, including sent screening data (eventData)
- ***Transaction Data*** - Optional fields that provide additional context in the alert alongside match results
- ***ListData*** - Maintains the original structure of data derived from the screening process

The following JSON provides an example of a Transaction Screening match results JSON. The event type, equivalent to entity type in the API response, may vary based on the specific context (e.g., "Individual" instead of "Beneficiary").

2.4.2.1. JSON Example


```
{
  "requestId": "tr-test-901234567890",
  "sourceMessages": {
    "originatorName": "Hizbollah",
    "beneficiaryAccountNumber": "testtest",
    "transactionReference": "same text here",
    "senderReference": "testtest",
    "senderToReceiverInformation": "same text here",
    "senderBIC": "testtest",
    "transactionId": "tr-test-901234567890",
    "usdequiv": 994.59,
    "interbankValueDate": "20221005",
    "beneficiaryName": "testtest",
    "beneficiaryBankName": "RHBBMYKLXXX",
    "originatorBIC": "testtest",
    "interbankSettlementAmount": 4600,
    "receiverBIC": "testtest",
    "interbankCurrency": "MYR"
```

```
},
"screeningData": [
  {
    "eventType": "ORIGINATOR_NAME",
    "dataId": "05d1f43ba1d8fbc243be47ae2d77c49ae133c0d46d0b5387b0661e411c241fdb",
    "eventData": {
      "name": "Hizbollah",
      "bic": "testtest",
      "dateOfBirth": null,
      "address": null,
      "country": null
    },
    "listInfoData": [
      {
        "type": "COMPANY",
        "name": "AUSTRALIA",
        "programs": [
          "1373 (2001)"
        ],
        "matchScore": 93,
        "listData": {
          "dataID": "277",
          "name": {
            "fullName": "Revolutionary Justice Organization"
          },
                      "addresses": [
            {
              "country": "LB"
            }
```

```
            ],
            "categories": [
              "SANCTION"
            ],
            "revision": 1076,
            "created": "2023-06-08T12:45:16.097",
            "updated": "2023-09-06T04:45:16.280",
            "lastMatched": "2023-06-08T12:45:16.097",
            "active": true,
            "dataset": {
              "label": "AUSTRALIA",
              "category": "SANCTION",
              "dateOfPublication": "2023-07-26T06:41:53",
              "dateOfUpload": "2023-09-06T04:45:16.206"
            },
            "additionalInformations": [
              {
                "type": "Additional Information",
                "value": "Hizballah (the Party of God) was established in 1982 in Lebanon, with the primary aim of establishing a
theocratic state in Lebanon."
              },
              {
                "type": "Listing Information",
                "value": "Instrument of first listing: Charter of the United Nations (Anti terrorism – Persons and Entities) List 2001
(No. 2). Last listed on: 16/09/2022"
              }
            ]
          },
          "matchType": "st_match",
```

```
    "apiVersion": 3,
    "screeningType": "FULL_SCREEN"
  },
  {
    "type": "INDIVIDUAL",
    "name": "OFAC SDN",
    "programs": [
      "SDGT"
    ],
    "matchScore": 93,
    "listData": {
      "dataID": "23685",
      "name": {
        "givenName": "Hizb Ullah Astam",
        "surname": "KHAN"
      },

      "cultureCodes": [
        "XX"
      ],
      "revision": 1077,
      "created": "2023-06-08T12:10:13.494",
      "updated": "2023-09-06T06:10:11.990",
      "lastMatched": "2023-06-08T12:10:13.494",
      "active": true,
      "dataset": {
        "label": "OFAC SDN",
        "category": "SANCTION",
        "dateOfPublication": "2023-07-28T00:00:00",
```

```
            "dateOfUpload": "2023-09-06T06:10:17.993"
          },
          "identityDocuments": [
            {
              "type": "National ID No.",
              "country": "PK",
              "number": "1730113198199"
            }
          ],
          "additionalInformations": [
            {
              "type": "remarks",
              "value": "(Linked To: AMEEN AL-PESHAWARI, Fazeel-A-Tul Shaykh Abu Mohammed)"
            }
          ]
        },
        "matchType": "st_match",
        "apiVersion": 3,
        "screeningType": "FULL_SCREEN"
      }
    ]
  }
]
}
```

3. Real Time Transaction Monitoring

3.1. Overview

Real Time Transaction Monitoring refers to the real time analysis of transactions in order to find patterns of risky or irregular behavior. The analysis is done in real time via deterministic rules that are customized to fit the financial use cases and relevant regulations. The real time analysis provides the results of the rules as a match or no match response.

Distinguishing real-time transaction monitoring from post-analysis transaction monitoring is crucial in understanding their respective roles within the regulatory framework. Real-time transaction monitoring involves the continuous surveillance of transactions as they occur, allowing for immediate detection and response to suspicious activities. In contrast, post-analysis transaction monitoring entails the retrospective examination of historical data to identify patterns or anomalies after the fact.

For more information on supported rules, please refer to the *Real Time Transaction Monitoring Overview* document.

3.2. API Specifications

Note: ThetaRay provides the concrete solution name, hostnames, authentication credentials, connector IDs, and JSON structures as part of the project setup.

3.2.1. Authentication

Calls to ThetaRay APIs require a bearer token to be passed as the HTTP Authorization header of the relevant request. The token is obtained through the '/auth' endpoint and is time limited. Once the token expires the client is expected to obtain a new token and the validity time of the token is returned as part of the response to the authentication request.

3.2.1.1. HTTP Request

```
POST https://gateway-<shared ns>.<host-domain-name>/auth
```

3.2.1.2. HTTP Headers

```
Content-Type = application/json
```

3.2.1.3. Request Body

The request body contains data with the following structure:

Fields	
clientId	Name of the data consumer (the requesting side) as provided by the system administrators.
clientSecret	Authentication credentials as provided by ThetaRay.

3.2.1.3.1. JSON Representation

```
{
   "clientId": "platform-ingestion-<solution>",
   "clientSecret": "<credentials provided by ThetaRay>"
}
```

3.2.1.4. Response Body

If successful, the response body contains data with the following structure:

Fields	
token	The bearer token authentication string. Paste in the request header Authorization field.
expirationInMinutes	The token validity period. After the time is expired a new authentication request is required.

3.2.1.4.1. JSON Representation

```
{
  "result": {    "token": "<JWT>",
  "expirationInMinutes": 5
  }
}
```

3.2.2. Real Time Transaction Evaluation - Request

The Real Time Transaction Evaluation refers to a dedicated endpoint for sending transactions in real time to be evaluated for risky financial behavior. The transaction runs through a set of deterministic rules that analyze the content of the transaction. The API synchronously returns the evaluation result which includes whether the request have matched any rule, if so - the details of the matching rules. In case of a match an alert is generated within the ThetaRay Investigation Center, allowing an analyst to perform manual investigation of the respective transaction.

3.2.2.1. HTTP Request

```
POST https://{gw_url}/evaluate/v1/[solution]/[ef_identifier]
```

3.2.2.2. HTTP Headers

```
{
  "Content-type": "application/json",
  "Accept": "application/json",
  "Authorization": "Bearer {TOKEN}""
}
```

3.2.2.3. Request Body

The request body should hold the transaction information, details on the related parties and counterparties (e.g. debtor and creditor), as well as other information pertinent to the rules that were set up for analysis.

The request body contains data with a dynamic structure depending on the input dataset. The concretely supported data fields and types are implementation-specific and provided as part of the project setup.

3.2.2.4. Request Body Example

```
{
   "transaction_id":1,
   "customer_id":"customer_001",
   "account_id":"1111111",
   "date":"20231215",
   "type":"credit",
```

```
    "operation":"remittance to another bank",
    "bank":"bic1111",
    "cnp_account_id":"2222222",
    "amount":30000
}
```

3.2.3. Real Time Transaction Evaluation - Response

The real time response will include either a “no match”, indicating that none of the rules applied to the transaction, or a “match”, which details all of the rules that were found to be true for the transaction.

Response Field	SubField	Data Type	Description
investigatedEntity		object	Returns the field or fields that were selected as the primary key for the investigation. It is recommended to select a unique identifier for the transaction for this field.
checkResult		string	Returns either “match” or “no match” values
risks		array	Is present only when there is a “match”. Returns an array of matching risks (rules) found true for the transaction. Contains the subfields listed below.
	identifier	string	The identifier of the risk (rule)
	display_ name	string	The name of the risk (rule) as displayed in the Investigation Center
	description	string	The description of the risk (rule) as displayed in the Investigation Center
	category	string	The category of the risk (rule) as displayed in the Investigation Center
	severity	integer	The number that signifies the severity of the risk (rule) as displayed in the Investigation Center

Both “match” and “no match” responses will include the transaction ID, or the field selected as the primary key for the investigation.

3.2.3.1. Response - No Match

```
{
   "investigatedEntity":{
      "transaction_id":4
```

```
    },
    "checkResult":"no match"
}
```

3.2.3.2. Response - Match

```
{
   "investigatedEntity":{
      "transaction_id":1
   },
   "checkResult":"match",
   "risks":[
      {
         "identifier":"rt_trans_count_risk",
         "display_name":"structuring_risk",
         "description":"Multiple same-value transactions within a limited time frame",
         "category":"real-time",
         "severity":"90"
      }
   ]
}
```

3.2.4. HTTPS Response Codes

Code	Description
200	Request successfully processed
401	Unauthorized to perform the request - credentials not valid
403	Request forbidden due to insufficient privileges
404	Endpoint does not exist
422	Unable to process the request due to validation issues - date time format is not the expected format
500	Internal server error

3.2.5. Real Time Rules Alert Status Callback API

When a rule or multiple rules have been found true for a specific transaction, an alert will be created in the ThetaRay Investigation Center detailing all of the rules and risks regarding the suspicious transaction.

Once the alert has been investigated and closed, the ThetaRay system will send out a callback API that can be subscribed to, detailing the results of the investigation.

The callback contains the following fields:

Response Field	Data Type	Description
timestamp	timestamp	The timestamp that the callback was sent

Response Field	Data Type	Description
primary_keys	object	Returns the field or fields that were selected as the primary key for the investigation. It is recommended to select a unique identifier for the transaction for this field.
alert_id	string	The id of the alert that was closed
username	string	The username of the analyst that closed the alert
alert_resolution	string	The resolution code selected by the analyst - from a set of custom, predefined codes in the Investigation Center
recommended_ resolution	string	The recommended resolution code selected by the analyst - from a set of custom, predefined codes in the Investigation Center
notes	string	The notes the analyst has inputted when closing the alert

3.2.5.1. Callback API Example

```
{
  "timestamp": 1716382376.07848,
  "investigatedEntity": {
    "transaction_id": 1
  },
  "alertId": "...",
  "username": "virtual analyst",
  "actualResolution": "Escalated",
  "recommendedResolution": "Closed",
  "notes": "this is a test of callback system"
}
```

4. Customer Risk Assessment (CRA)

4.1. Overview

The CRA product enables taking various risk factors into consideration when determining the risk levels, such as KYC information, transaction monitoring, Sanction and PEP screening results, etc. A risk score and a risk classification are calculated from the various risk factors and externalized both to the clients' internal systems and to the ThetaRay Investigation Center.

The CRA product receives a customer or batch of customers and runs the risk calculation model, which is enhanced with additional information, such as the relevant high risk countries. The model also has the option to be enhanced from ThetaRay internal events, such as resolving a screening alert. The risk calculation model is comprised of the risk score calculation and various other rules, and is adjusted to fit the client's specific needs and risk appetite.

In addition to the creation of a dedicated CRA alert in the Investigation Center, the results of the risk calculation model are externalized via API, and are generated into a report. The report is compiled for all the customers classified in the CRA analysis and is available for retrieval via API and through the CRA On Demand Reporting Module. The On Demand Reporting Module has various other capabilities which are elaborated in the corresponding section of this document.

4.2. Data Ingestion

4.2.1. Transaction Data

Transactional data can be ingested as any transactions for Transaction Monitoring, including via files, raw transaction formats (RJE and XML) or REST API. The transactions uploaded for Transaction Monitoring analysis can also be reused for CRA analysis.

4.2.2. KYC & Auxiliary Data

Similar to transactional data, KYC and auxiliary datasets can be ingested and reused between Transaction Monitoring and CRA analyses. Auxiliary data is crucial for setting the CRA risk factors and their values.

4.3. Classification Changes API

The classification changes API notifies at the end of each CRA analysis of any classification changes that occurred. The API will be sent to a dedicated endpoint configured by the client. Any classification change is included in the API call, whether the classification changes to a lower or higher level, in order to keep the KYC system (or any system that has subscribed) up to date. A manual reclassification from the CRA alert by the analyst will also trigger the classification change API.

Element	URL
Endpoint	configurable URL provided by the client
Method	POST

The example below represents an API call that is triggered by the automatic CRA analysis, and summarizes all of the customers that had a change in classification.

Annotations are custom fields from the available data that can be added to the response. Please contact ThetaRay for adding custom fields to the API.

Request Example #1:

```
{
  "execution_date": "01/01/1970",
  "customers": [
      {
        "customer_id": "customer id1",
        "customer_name": "customer name1",
        "risk_score": 90,
        "risk_classification": "Very High",
        "reclassification": false,
        "previous_classification": "High",
        "previous_score": 80,
        "annotation1": "annotation 1 value"
      },
      {
        "customer_id": "customer id2",
        "customer_name": "customer name2",
        "risk_score": 90,
        "risk_classification": "Very High",
        "reclassification": false,
        "previous_classification": "High",
        "previous_score": 80,
        "annotation1": "annotation 1 value"
      }
  ]
}
```

The below example represents an API call triggered by the manual reclassification of the customer by the analyst in the Investigation Center

Request Example #2

```
{
  "execution_date": "timestamp",
  "customers": [
      {
        "customer_id": "customer id1",
        "customer_name": "customer name1",
        "risk_score": 90,
        "risk_classification": "Very High",
        "reclassification": true,
        "previous_classification": "High",
        "previous_score": 80,
        "user_name": "user name",
        "note": "note",
        "annotation1": "annotation 1 value"
      }
  ]
}
```

4.4. Report Notification

A report is generated automatically for each CRA analysis, and can contain a single customer or the entire customer base, according to the analysis type. The report contains a detailed breakdown of the risk score, rule results, classifications and alert details. Once the report is uploaded to a dedicated bucket, a notification is sent to a dedicated endpoint configured by the client to indicate that the report is available for retrieval via API. The notification will include the path to the relevant report.

Element	URL
Endpoint	configurable URL provided by the client
Method	POST

```
{
  "event": "CRA classification report" / "CRA reclassification report",
  "s3_endpoint": Settings.S3_INGRESS,
  "path": "path to report file",
  "bucket": Settings.S3_PUBLIC_BUCKET,
  "execution_date": "01/01/1970",
  "timestamp": timestamp,
```

```
  "analyzed_customers_count": 999
}
```

5. CRA Algorithm

Disclaimer: Be aware this chapter is draft and therefore the content therein is classified as Non GA.

5.1. Overview

Customer Risk Assessment (CRA) is a ThetaRay product designed to aid financial institutions in assessing the risk levels of their customers. CRA is a fundamental process that financial institutions and other entities involved in financial transactions perform to understand their customers, assess their risk profiles, and verify their identities. It is a crucial component of anti-money laundering (AML) and counter-terrorist financing (CTF) efforts, and it complements the AML solution by helping to prevent financial crimes and ensure regulatory compliance.

The solution is designed to comply with regulatory requirements and industry standards, ensuring that the Customer Risk Assessment process is robust, transparent, and aligned with organizational goals. The ThetaRay CRA product provides a solution for analyzing customers one by one, as well as continuously monitoring the entire customer database at a selected cadence.

The CRA product receives customer data (single customer data or a batch of customer data) and runs the risk rating engine, which is enhanced with additional information, such as the relevant high risk countries. The risk rating engine incorporates deterministic rules of varying categories and logic, as well as a proprietary risk classification algorithms.

In addition to the creation of a dedicated CRA case in the Investigation Center, the results of the risk rating engine are externalized via API, and are generated into a report. The report is compiled for all the customers classified in the CRA analysis and is available for retrieval via API and through the CRA On Demand Reporting Module.

5.2. Data Ingestion

Customer data for Risk Rating analysis can be ingested via a csv file uploaded through the proper connector. Auxiliary data, for example, a list of risky countries and their relevant attributes, can be ingested in the same manner. For more information on supported ingestion methods, please refer to the Data Ingestion chapter of this document.

5.3. Classification Changes API

The classification changes API notifies at the end of each CRA analysis of any classification changes that occurred. The API will be sent to a dedicated endpoint configured by the client. Any classification change is included in the API call, whether the classification changes to a lower or higher level, in order to keep the KYC system (or any system that has subscribed) up to date. A manual reclassification from the CRA alert by the analyst will also trigger the classification change API, with the relevant details.

The classification changes API supports basic authentication.

Element	URL
Endpoint	configurable URL provided by the client
Method	POST

The example below represents an API call that is triggered by the automatic CRA analysis, and summarizes all of the customers that had a change in classification. Please note that “annotations” are custom fields from the available data that can be, optionally, added to the response.

```
{
  “execution_date”: “2024-05-24T13:30:57.636547Z”,
  “customers”: [
    {
      “customer_id”: “46364434”,
      “customer_name”: “Dominique Whitaker”,
      “reclassification”: false,
      “user_name”: null,
      “note”: null
      "annotation1": "annotation 1 value",
      “previous_classification”: {
        “rv_cra_ml_risk_aml_effective_classification”: “L”,
        “rv_cra_ml_risk_san_effective_classification”: “L”,
        “rv_cra_ml_risk_aml_ml_score”: 39,
        “rv_cra_ml_risk_san_ml_score”: 38
      },
      “rv_cra_ml_risk_aml_effective_classification”: “H”,
      “rv_cra_ml_risk_san_effective_classification”: “H”,
      “rv_cra_ml_risk_aml_ml_score”: 100,
      “rv_cra_ml_risk_san_ml_score”: 100
    }
  ]
}
```

The below example represents an API call triggered by the manual reclassification of the customer by the analyst in the On Demand Reporting module.

```
{
  “execution_date”: “2024-05-24T13:30:57.636547Z”,
  “customers”: [
    {
      “customer_id”: “46364434”,
      “customer_name”: “Dominique Whitaker”,
      “reclassification”: true,
      “user_name”: “manager”,
      “note”: “Test”,
      "annotation1": "annotation 1 value",
      “previous_classification”: {
        “rv_cra_ml_risk_aml_effective_classification”: “L”,
        “rv_cra_ml_risk_san_effective_classification”: “L”,
        “rv_cra_ml_risk_aml_ml_score”: 39,
        “rv_cra_ml_risk_san_ml_score”: 38
      },
      “rv_cra_ml_risk_aml_effective_classification”: “H”,
      “rv_cra_ml_risk_san_effective_classification”: “H”,
      “rv_cra_ml_risk_aml_ml_score”: 100,
      “rv_cra_ml_risk_san_ml_score”: 100
    }
  ]
}
```

5.4. Report Notification

A report is generated automatically for each CRA analysis, and can contain a single customer or the entire customer base, according to the analysis type. The report contains a detailed breakdown of the risk score, rule results, classifications and alert details. Once the report is uploaded to a dedicated bucket, a notification is sent to a dedicated endpoint configured by the client to indicate that the report is available for retrieval via API. The notification will include the path to the relevant report.

Element	URL
Endpoint	configurable URL provided by the client
Method	POST

```
{
  "event": "CRA classification report" / "CRA reclassification report",
  "s3_endpoint": Settings.S3_INGRESS,
  "path": "path to report file",
  "bucket": Settings.S3_PUBLIC_BUCKET,
```

```
  "execution_date": "01/01/1970",
  "timestamp": timestamp,
  "analyzed_customers_count": 999
}
```

6. Infrastructure

6.1. Workflow Management

Note: Available to on-prem customers or to SaaS customers as an opt-in functionality.

Note: Only relevant for Transaction Monitoring and Customer Risk Assessment

6.1.1. Overview

The Workflow Management section describes the APIs exposed by the platform for triggering and monitoring the execution of Airflow DAGs (Directed Acyclic Graphs) which are used to automate the ThetaRay platform data ingestion, pre-processing and analysis operations. Typical usage of these APIs includes launching of data processing activities through external schedulers or other services running on customer premises in order to integrate with a broader workflow that is beyond the scope of the ThetaRay environment.

ⓘ Since this document only presents the two most frequently-used requests, refer to the official Airflow documentation to learn about the entire API.

6.1.2. Authentication

Workflow management APIs rely on HTTP Basic Authentication. Each request should include a username / password to be provided on the HTTP Authorization header. Credentials (username / password) for the specific environment will be provided by ThetaRay as part of the initiation of the integration

6.1.3. Trigger a new DAG run request

Start the specified DAG run.

The DAG (workflow) to launch is identified through a dag_id and defined as part of the project implementation. Each workflow execution is uniquely identified through an 'execution_date' which should be provided as part of the request payload.

HTTP request

```
airflow-<platform namespace>.<cluster domain>/api/v1/{dag_id}/dagRuns
```

The DAG (workflow) to launch is identified through a dag_id and defined as part of the project implementation. Each workflow execution is uniquely identified through an 'execution_date' which should be provided as part of the request payload.

6.1.3.1. Path parameters

Parameters	
dag_id	string Required. The ID of the DAG to trigger.

6.1.4. HTTP Headers

```
Content-Type = application/json; Authorization = Bearer <...>
```

6.1.4.1. Request body

The request body contains data with the following structure:

JSON Representation

```
{  "dag_run_id": "string",
    "execution_date": "2019-08-24T14:15:22Z",
    "conf": { }}
```

Fields	
dag_run_id	string Nullable Run ID. This together with DAG_ID are a unique key. If not provided, a value will be generated based on execution_date. If the specified dag_run_id is in use, the creation request fails with an ALREADY_EXISTS error.
execution_date	string <date-time> Nullable The execution date. This is the time or interval covered by this DAG run, according to the DAG definition.

Fields	
	This together with DAG_ID are a unique key.
conf	object JSON object describing additional configuration parameters.

6.1.4.2. Response body

The response body contains data with the following structure:

JSON Representation

```
{
  "dag_run_id": "string",
  "dag_id": "string",
      "logical_date": "2019-08-24T14:15:22Z",
      "execution_date": "2019-08-24T14:15:22Z",
      "start_date": "2019-08-24T14:15:22Z",  "end_date": "2019-08-
24T14:15:22Z",
    "state": "queued",
    "external_trigger": true,
      "conf": { }
}
```

Fields	
dag_run_id	string Nullable Run ID. If not provided, a value will be generated based on execution_date. If the specified dag_run_id is in use, the creation request fails with an ALREADY_EXISTS error. This together with DAG_ID are a unique key.
dag_id	string
logical_date	string <date-time> Nullable The logical date (previously called execution date). This is the time or interval covered by this DAG run, according to the DAG definition. This together with DAG_ID are a unique

Fields	
	key.
execution_date	string <date-time> Nullable Deprecated The execution date. This is the same as logical_date, kept for backwards compatibility. If both this field and logical_date are provided but with different values, the request will fail with an BAD_REQUEST error.
start_date	string <date-time> Nullable The time when DAG run was actually created.
end_date	string <date-time> Nullable The time when the DAG run status changed to either 'success' or 'failed'
state	string (DagState) Enum: "queued" "running" "success" "failed" DAG State.
external_trigger	boolean default : true
conf	object JSON object describing additional configuration parameters.

6.1.5. HTTP Response Code

Code	Description
200	200 success
401	Request not authenticated due to missing invalid authentication info.
403	Client does not have sufficient permissions.
404	A specified resource is not found.
409	An existing resource conflicts with this request.

6.2. Get a DAG run request

Review the status and other details of a specified DAG execution.

6.2.1. HTTP request

```
airflow-<platform namespace>.<cluster domain>/api/v1/{dag_id}/dagRuns
```

6.2.1.1. Path parameters

Parameters	
dag_id	string **Required. The ID of the DAG to trigger.**
dag_run_id	string **Required. The DAG run ID.**

6.2.2. HTTP Headers

```
Content-Type = application/json; Authorization = Bearer <...>
```

6.2.3. Response body

The response body contains data with the following structure:

JSON Representation

```
{
      "dag_run_id": "string",
    "dag_id": "string",
      "logical_date": "2019-08-24T14:15:22Z",
      "execution_date": "2019-08-24T14:15:22Z",
      "start_date": "2019-08-24T14:15:22Z",  "end_date": "2019-08-
24T14:15:22Z",
    "state": "queued",
    "external_trigger": true,
      "conf": { }
}
```

Fields	
dag_run_id	string Nullable Run ID. If not provided, a value will be generated based on execution_date. If the specified dag_run_id is in use, the creation request fails with an ALREADY_EXISTS error. This together with DAG_ID are a unique key.
dag_id	string
logical_date	string <date-time> Nullable The logical date (previously called execution date). This is the time or interval covered by this DAG run, according to the DAG definition. This together with DAG_ID are a unique key.
execution_date	string <date-time> Nullable Deprecated The execution date. This is the same as logical_date, kept for backwards compatibility. If both this field and logical_date are provided but with different values, the request will fail with an BAD_REQUEST error.
start_date	string <date-time> Nullable The time when DAG run was actually created.
end_date	string <date-time> Nullable The time when the DAG run status changed to either 'success' or 'failed'
state	string (DagState) Enum: "queued" "running" "success" "failed" DAG State.
external_trigger	boolean default : true
conf	object JSON object describing additional configuration parameters.

6.2.4. HTTP Response Code

Code	Description
200	200 success
401	Request not authenticated due to missing invalid authentication info.
403	Client does not have sufficient permissions.
404	A specified resource is not found.

6.3. Import / Export API

6.3.1. Import API Settings

API Description

Request to import metadata settings: Move all the definitions from lower environment to upper environment (or vice-versa). Thus, you do not to need to redefine parameters again when creating a new environment from scratch or updating an existing environment.

API Call

```
POST <app namespace URL>/ic-be/api/configuration/import
```

Headers

```
Authorization: Bearer <token>
Content-Type: multipart/form-data
```

6.3.1.1. Body

Parameter name	Type	Value	Required	Description
file	File	File in JSON format	true	Import file in JSON format which was created by export API
solutionMapping	Object	{"<solution name in import file>":"<solution name on env>"}	false	Map data in import file to data on env

6.4. Export API Setting

API Description

Request to export metadata settings: Move all the definitions from an environment to upload to another environment. The definitions can be changed manually before uploading.

API Call

POST <app namespace URL>/ic-be/configuration/export

Headers:

Authorization: Bearer <token>

Content-Type: application/json

6.4.1. Body:

Body for export request Expand source

```
{
    "importType": "UPDATE",
    "alertDefinitions": [
        "1685436710390"
    ],
    "queues": [
        "1685436712901"
    ],
    "users": [],
    "exportUserAttributes": false,
    "exportCustomViews": true,
    "exportQueueOrder": true
}
```

Parameters	Possible Values	Explanation
"importType"	“UPDATE”/“CREATE”/“UPDATE_AND_CLEANUP”	Export mode: create - only creating new entities in imported env, update - create and update all entities which identifiers will be align between env, update_and_cleanup, update current solution and delete all entities which not presented in package. (If system can’t delete, process will fail)
"alertDefinitions"	"1685436710390"	Identifier of mapper, can be extract from “ALERT_MAPPER” table (SELECT identifier investigation_cetner.alert_mapper where mapper_name = {relevant mapper name}
"queues"	"1685436712901"	identifier of queue, can be extract from “QUEUE” table (SELECT identifier from investigation_center.queue where queue_name = {relevant queue name}

6.5. Structure of Export File:

The Export file has the same structure as the Import file.

6.5.1. Response Codes for Export Request

HttpCode	Message
200 (OK)	No message, file is ready to download in JSON format
504	Timeout, file not uploaded. Check logs for details on IC BE Pod

6.6. Data Retrieval

6.6.1. Overview

Data Retrieval section covers the following topics:

- Overview of the CDD and Graph APIs
- Data Model and associated data tables employed
- Details of the Graph QL interface and access

6.6.2. CDD and Graph QL APIs

The CRA and Graph QL APIs are used to expose alert information by the Investigation Center . Data in the tables can be directly accessed through SQL queries against the database for components running inside the Kubernetes environments or through a GraphQL interface exposed to external clients.

6.6.3. Data Model

Investigation Center tables are located within the 'CDD' database inside the central Postgresql instance. The database includes separate schemas for each Investigation Center instance installed within the environment. The naming convention for the schema is:

```
'<IC instance name>'
```

The instance name for the Investigation Center is defined as a deployment time parameter. Environments may include multiple instances of the Investigation Center to facilitate scenarios such as testing or dedicated instances per tenant.

Following is a list of tables included under the schema:

6.6.3.1. monitoring _table

The table consists of metadata information for each 'Mapper' configured within the Investigation Center. This also includes the actual table name used to hold alerts within the **Central Data Depot** for the given Mapper.

Column Name	Column Type	Description
alert_table_name	VARCHAR(100)	Table name for alerts
alert_mapper_identifier	VARCHAR(100)	Reference to specific mapper (Internal)
data_permission	VARCHAR(100)	Data permission value

6.6.3.2. tr_alert_table_<mapper identifier>

The table consists a change log for Alerts information in the context of a given 'Mapper'. The table is for 'append' mode only and can be tracked for updates through the 'change date' column. Existence of these tables can be discovered through the 'monitoring table' which consists of an entry per 'alert_table'.

Following is the structure of the table:

Column Name	Column Type	Description	ThetaRay Use Only
id	TEXT	Alert ID without prefix	
assignee	TEXT	Assignee of the alert	
stateid	TEXT	State of the alert	
processdefinitionid	TEXT	id of process definition in Activiti	Yes
processinstanceid	TEXT	id of process instance in Activiti	Yes
resolutioncode	TEXT	resolution code for the alert	
recommendedresolutioncode	TEXT	recommended resolution code for the alert	
severity	INTEGER	severity score of the alert	
sla	JSONB	SLA definition for the alert	Yes
alertmapperidentifier	TEXT	Reference in Mapper	
alertmapperversion	INTEGER	Metadata version of Mapper	
triggers	JSONB	All triggers of the alert (including consolidation)	

Column Name	Column Type	Description	ThetaRay Use Only
queueidentifier	TEXT	Reference to queue of the alert	
workflowidentifier	TEXT	Reference to workflow of the alert	
workflowversion	INTEGER	Reference to workflow version of the alert	
supressioncount	INTEGER	Counts of supressions for the alert	
consolidationcount	INTEGER	Number of consolidations for the alert	
isclosed	BOOL	Indication if the alert is closed	
isconsolidated	BOOL	Indication whether the alert is consolidated	
note	JSONB	Notes associated with the alert	
attachments	JSONB	Metadata of all attachments associated with the alert	
createdate	TEXT	Timestamp at which the alert was created	
updatedate	TEXT	Timestamp at which the alert was last updated	

6.6.4. GraphQL Interface

The data presented above can be accessed through a GraphQL interface over HTTPS by clients running outside the Kubernetes environment. A dedicated URL is provided for GraphQL access - by default the following FQDN is used :

```
'https://graphql-<shared namespace>.<cluster domain>/v1/graphql'
```

(this URL may vary depending on environment settings provided at deployment time).

- Authentication against ThetaRay's GraphQL Gateway which is based on Hasura can be performed in two manner -

- Authenticating against ThetaRay's embedded Keycloak instance using OpenID Connect and providing the resulting JWT as a bearer token on the HTTP Authorization header.

The GraphQL interface enables external clients to dynamically query data within the above tables allowing the requested data fields and filtering criteria to be specified as part of the GraphQL request. Additional details around the filtering capabilities of the embedded GraphQL gateway can be found at :

https://hasura.io/docs/latest/graphql/core/databases/postgres/queries/index.html

6.6.4.1. Example GraphQL Access

An example Python code using the Python 'requests' library is provided below

```
import requests

endpoint = "https://graphql-shared-v6.demo.thetaray.cloud/v1/graphql"
admin_secret = < .... admin secret .... >

headers = {
    "x-hasura-admin-secret": admin_secret
}

graphql_query = """
query MyQuery {
  apps_v6_tr_alert_table_1641817900797 {
    id
    isclosed
    isconsolidated
    assignee
    resolutioncode
    severity
    updatedate
    createdate
  }
}
"""

r = requests.post(endpoint , json={"query": graphql_query}, headers=headers)
print (r.text)
```