

Technical Documentation – AML Ruleset Definition Language

DSL Schema Documentation for Transaction Rulesets

Overview

This DSL (Domain-Specific Language) defines ruleset-based conditions for handling transactions. The rulesets specify conditions under which transactions should be blocked, allowed, or flagged. The schema enables defining logical conditions using `AND`, `OR`, `request_property_check`, `transactions_volume_check`, `transactions_quantity_check`, `blacklist_check`, `greylist_check`, `kyc_property_check` and `compare_with_last_transaction` statements, along with customizable triggers and actions.

Schema Structure

```
conditions:
  AND | OR: # Logical operator grouping conditions
  - request_property_check: # Check type
    property: <string> # The property to check (e.g., "transactionData.acquirerCountry")
    comparator: <string> # enums: "IN", "NOT_IN", "=", "!=", ">", ">=", "<", "<=",
"CONTAINS"
    value: <string | list | variable> # Expected value or variable reference, eg: string:
"123", list: "value1, value2, value3" or variable (`{{ vars.UHRC_COUNTRIES }}`)
    treat_missing_value_as: <boolean> # optional, How to handle missing fields (default:
false)
  - transactions_volume_check:
    scope: <string> # enums: CORPORATION, USER, CARD, BALANCE
    by: <string> # optional, enums: MERCHANT, COUNTRY, grouping parameter
```

```
period: <string> # 1d -> 1 day, 1M -> one month etc.
amount: <number> # cumulative amount for given scope/by and period, as minor > 50000
currency: <string> # country currency codes e.g. PLN, USD, AOA
currencyAggregation: <string> # enums: SAME_CURRENCY_ONLY, CONVERT_TO_CURRENCY
filters: # optional
  - field: <string>> #enum: type, subtype, transactionData.merchantName,
transactionData.mcc, transactionData.countryCode, transactionData.contrahentName,
transactionData.captureMode
    comparator: <string> # enums: "IN", "NOT_IN", "=", "!="
    value: ATM # <string | list | variable>
- transactions_quantity_check:
  scope: BALANCE # enums: CORPORATION, USER, CARD, BALANCE
  by: COUNTRY # optional, enums: MERCHANT, COUNTRY, grouping parameter
  period: <string> # 1d -> 1 day, 1M -> one month etc.
  quantity: 5 # transactions count for given scope/by and period, > 5
  filters: # optional
    - field: <string>> #enum: type, subType, transactionData.merchantName,
transactionData.mcc, transactionData.countryCode, transactionData.contrahentName,
transactionData.captureMode
      comparator: <string> # enums: "IN", "NOT_IN", "=", "!="
      value: ATM # <string | list | variable>
- kyc_property_check:
  property: <string>
  comparator: <string> # enums: "IN", "NOT_IN", "=", "!=", ">", ">=", "<", "<=",
"CONTAINS"
  value: <string | list | variable>
  treat_missing_value_as: <boolean> # optional, How to handle missing fields (default:
false)
- blacklist_check:
  properties:
    - property: <string> # blacklist record property
      kyc_value: <string> # user's KYC record property to evaluate
    - property: <string> # blacklist record property
      request_value: <string> # aml-verify request property to evaluate
- greylist_check:
  properties:
    - property: <string> # greylist record property
      kyc_value: <string> # user's KYC record property to evaluate
    - property: <string> # greylist record property
      request_value: <string> # aml-verify request property to evaluate
```

```

- compare_with_last_transaction:
  options:
    within_seconds: <number> # time in seconds
    subType: list<string> # types of transactions will be searched for in the
transaction history, e.g. ["ATM_WITHDRAWAL", "PURCHASE"]
    context: <string> # enums: CARD, BALANCE, BALANCE_OWNER
    captureMode: list<string> # Payment channel/method (e.g. CONTACT, CONTACTLESS,
ECOMMERCE, CLOUD, MONEYSSEND, etc.)
    property: <string> # The property to check in history transaction (e.g.,
"transactionData.acquirerCountry")
    comparator: <string> # enums: "IN", "NOT_IN", "=", "!=", ">", ">=", "<", "<=",
"CONTAINS"
    request_property: <string> # The property to check in request (e.g.,
"transactionData.acquirerCountry")
    treat_missing_value_as: <boolean> # optional, How to handle missing fields (default:
false)

- AND | OR:
  - request_property_check: null
    field: <string>
    #...

trigger:
  decision: <string> # enum ( APPROVED, DECLINED)
  actions:
    <group>: # Action group
    - name: <string> # Action name (e.g. block_resource)
      properties:
        <key>: <value> # Action properties (e.g. reason: fraud_suspected, resource_type:
user)
  alert:
    channels: <string> # enum: YOUTRACK_TICKET, USER_PUSH_NOTIFICATION,
USER_EMAIL_NOTIFICATION
    cooldown_period: <string> # optional, 1d -> 1 day, 1M -> one month etc.
  balance_owner_notifications:
    - type: <string> # enum: SMS, EMAIL
      template_name: <string> # notification template name
      cooldown_period: <string> # optional, 1d -> 1 day, 1M -> one month etc.

```

Field Descriptions

Top-Level Fields

Field	Type	Description
<code>conditions</code>	Object	Defines logical conditions using <code>AND</code> / <code>OR</code> .
<code>trigger</code>	Object	Specifies what happens when the conditions are met.

Condition Fields

Field	Type	Description
<code>AND</code> / <code>OR</code>	List	Groups multiple conditions where all (<code>AND</code>) or at least one (<code>OR</code>) must be met.
<code>request_property_check</code>	Object	Specifies a request property condition check.
<code>kyc_property_check</code>	Object	Specifies a kyc value check.
<code>transactions_volume_check</code>	Object	Specifies a transactions volume check.
<code>transactions_quantity_check</code>	Object	Specifies a transactions quantity check.
<code>blacklist_check</code>	Object	Specifies a blacklist check.
<code>greylist_check</code>	Object	Specifies a greylist check.
<code>compare_with_last_transaction</code>	Object	Specifies a last transaction check.

Common check fields

Field	Type	Description
<code>comparator</code>	String	Comparison operator, enums: <code>IN</code> , <code>NOT_IN</code> , <code>=</code> , <code>!=</code> , <code>></code> , <code>>=</code> , <code><</code> , <code><=</code> , <code>CONTAINS</code> , in case of date comparison uses ISO-8601 format for parsing date).

Field	Type	Description
<code>property</code>	<code>String</code>	The data field to evaluate (e.g., <code>transactionData.acquirerCountry</code>).
<code>value</code>	<code>String</code> / <code>List</code>	The expected value in the form of: string <code>value</code> list <code>value1, value2, value3</code> or a reference to a variable called "Value set" <code>{{ vars.UHRC_COUNTRIES }}</code> . Reference variables must be defined in API or front end administration panel. Only defined reference variables are allowed.
<code>treat_missing_value_as</code>	<code>Boolean</code>	Determines handling of missing fields (default <code>false</code>).

Comparators

Comparators are used to compare the value of a property with a given value.

Comparator	Accepted value type	Description
<code>=</code>	<code>String</code>	Property value equals the given value string (case-insensitive).
<code>!=</code>	<code>String</code>	Property value does not equals the given value string (case-insensitive).
<code>></code>	<code>String</code>	Property value is greater than the given value string (case-insensitive).
<code>>=</code>	<code>String</code>	Property value is greater than or equal the given value string (case-insensitive).
<code><</code>	<code>String</code>	Property value is less than the given value string (case-insensitive).
<code><=</code>	<code>String</code>	Property value is less than or equal the given value string (case-insensitive).
<code>IN</code>	<code>List</code>	Property value matches any of the strings in the given value list (case-sensitive).
<code>NOT_IN</code>	<code>List</code>	Property value matches none of the strings in the given value list (case-sensitive).
<code>CONTAINS</code>	<code>String/List</code>	Property value contains the given value string, or any string from the provided value list (case-insensitive).

Comparator	Accepted value type	Description
NOT_CONTAINS	String/List	Property value does not contain the given value string, nor any string from the provided value list (case-insensitive)

Request property check

Field	Type	Description
property	String	See property .
comparator	String	See comparator .
value	String / List	See value .
treat_missing_value_as	Boolean	See treat_missing_value_as .

Kyc check

Field	Type	Description
property	String	The Kyc property to evaluate (e.g., <code>riskLvl</code>).
comparator		See comparator .
value		See value .
treat_missing_value_as		See treat_missing_value_as .

Transactions volume check

Field	Type	Description	AML-Verify request mapping
scope	String	Defines the level at which the transaction volume is calculated. It determines whose transaction volume is being measured. enums: <code>CORPORATION</code> , <code>USER</code> , <code>CARD</code> , <code>BALANCE</code>	<code>CORPORATION</code> → <code>balance.ownerId</code> (if <code>balance.owner == CORPORATION</code>) <code>USER</code> → <code>balance.ownerId</code> (if <code>balance.owner == USER</code>) <code>CARD</code> → <code>resourceId</code> (if <code>resource == CARD</code>) <code>BALANCE</code> → <code>balance.id</code>

Field	Type	Description	AML-Verify request mapping
by	String	Specifies the parameter used to group transaction volume within the selected scope. enums: MERCHANT, COUNTRY	MERCHANT → transactionData.merchantIdentifier COUNTRY → transactionData.acquirerCountry
period	String	The time interval over which to aggregate transaction volume. See periods	-
amount	Number	Cumulative amount.	-
currency	String	Cumulative amount currency.	-
currencyAggregation	String	Defines how transactions in different currencies are aggregated.	-
filters	List	See transaction filters .	-

Transactions quantity check

Field	Type	Description	AML-Verify request mapping
scope	String	Defines the level at which the transaction quantity is calculated. It determines whose transaction quantity is being measured. enums: CORPORATION, USER, CARD, BALANCE	CORPORATION → balance.ownerId (if balance.owner == CORPORATION) USER → balance.ownerId (if balance.owner == USER) CARD → resourceId (if resource == CARD) BALANCE → balance.id
by	String	Specifies the parameter used to group transaction quantity within the selected scope. enums: MERCHANT, COUNTRY	MERCHANT → transactionData.merchantIdentifier COUNTRY → transactionData.acquirerCountry
period	String	The time interval over which to count transactions. See periods	-
quantity	Number	Transactions count.	-
filters	List	See transaction filters .	-

Blacklist and greylist check

Field	Type	Description
<code>property</code>	<code>String</code>	The property from the blacklist/greylist record to be checked (e.g., <code>name</code> , <code>pesel</code>).
<code>kyc_value</code>	<code>String</code>	The user's KYC field that will be compared against the <code>property</code> . (e.g. <code>firstName</code> , <code>lastName</code>)
<code>request_value</code>	<code>String</code>	The request field (from <code>aml-verify</code> body) that will be compared against the <code>property</code> .

Last transaction check

Field	Type	Description
<code>options.within_seconds</code>	<code>String</code>	Time in seconds to search transaction history from.
<code>options.subType</code>	<code>List<String></code>	Last transaction subType, e.g. <code>ATM_WITHDRAWAL</code> , <code>PURCHASE</code>
<code>options.context</code>	<code>String</code>	Last transaction context, enums: <code>CARD</code> , <code>BALANCE</code> , <code>BALANCE_OWNER</code>
<code>options.captureMode</code>	<code>List<String></code> <code>null</code>	Last transaction captureMode, will be matched with event's <code>transactionData.channel</code>
<code>property</code>	<code>String</code>	The property to check in history transaction (e.g., <code>transactionData.acquirerCountry</code>).
<code>comparator</code>	<code>String</code>	See comparator .
<code>request_property</code>	<code>String</code>	The property to check in request (e.g., <code>transactionData.acquirerCountry</code>).
<code>treat_missing_value_as</code>	<code>String</code>	See treat missing value as .

Allowed values for fields inside ruleset parameters

Ruleset check type	Allowed value
--------------------	---------------

Request property check <code>property</code>	transactionId, balance.owner, balance.ownerId, resourceId, externalReferenceTransactionId, resource, type, subType, amount, currency, originalAmount, originalCurrency, status, description, transactionDate, tenantId, transactionData.mcc, transactionData.merchantIdentifier, transactionData.merchantName, transactionData.captureMode, transactionData.lastFourDigits, transactionData.acquirerCountry, transactionData.mdesDigitizedWalletId, transactionData.cashbackPosCurrencyCode, transactionData.cashbackPosAmount, transactionData.lastFourDpan, transactionData.adjustmentReasonDescription, transactionData.retrievalReferenceNumber, transactionData.contrahentName, transactionData.contrahentIban, transactionData.contrahentBic, customData.*
Kyc check <code>property</code>	status, tenantId, customerId, dcUserId, verificationId, firstName, lastName, birthDate, nationality, riskLvl, createdAt, usaResident, taxResident, sourceOfFunds, pesel, country, city, identityCardNo, documents
Blacklist / greylist check <code>property</code>	userId, tenantId, name, surname, fullName, birthDate, pesel, documentNumber, documentType, addressCountry, addressCity, iban
Last transaction check <code>property</code>	transactionId, eventTimestamp, tenantId, type, subType, amount, currency, status, description, card.cardOwner, card.cardOwnerId, balance.balanceOwner, balance.balanceOwnerId, transactionDate, createdAt, clearedAt, externalTransactionId, externalReferenceTransactionId, transactionData.lastFourDigits, transactionData.mcc, transactionData.merchantName, transactionData.channel, transactionData.countryCode, transactionData.originalAmount, transactionData.originalCurrency, transactionData.walletReference, transactionData.contrahentName, transactionData.contrahentBic, transactionData.merchantId

Trigger Fields

Field	Type	Description
<code>decision</code>	<code>String</code>	The decision taken if the ruleset is triggered (<code>DECLINED</code> , <code>ON_HOLD</code> , <code>APPROVED</code>).

Field	Type	Description
actions	List	A list of actions executed when the ruleset is triggered. Actions and their properties must be defined in API. Only defined in API actions are allowed
alert	Object	Definition of alerts that should be sent to internal channels (e.g. YouTrack).
balance_owner_notifications	List	A list of direct notifications sent to the balance owner when the ruleset is triggered. See balance owner notification fields .

Alert Fields

Field	Type	Description
channels	List	List of channels where send alert. Possible values: <code>YOUTRACK_TICKET</code> , <code>USER_PUSH_NOTIFICATION</code> , <code>USER_EMAIL_NOTIFICATION</code>
cooldown_period	String	How long the sending of a repeated alert is suspended for a given (USER + TENANT) or (CORPORATION + TENANT). See periods

Balance Owner Notification Fields

Field	Type	Description
type	String	Notification channel type. Possible values: <code>SMS</code> , <code>EMAIL</code>
template_name	String	Name of the notification template to use.
cooldown_period	String	Optional. How long sending of a repeated notification is suspended for a given balance owner. See periods

Transaction check filters

`[0-*]` filters to narrow the transaction history search.

Each filter consists of:

Field	Type	Description
<code>field</code>	<code>String</code>	Filter field by. Possible values: <code>type</code> , <code>subType</code> , <code>transactionData.mcc</code> , <code>transactionData.countryCode</code> , <code>transactionData.merchantName</code> , <code>transactionData.contrahentName</code> , <code>transactionData.captureMode</code>
<code>comparator</code>	<code>String</code>	See comparator .
<code>value</code>	<code>String</code> / <code>List<String></code>	See value .

Periods

Allowed period values:

Unit variants	ChronoUnit
Y, y, yr, year, years	YEARS
M, m, mo, mon, month, months	MONTHS
w, week, weeks	WEEKS
d, day, days	DAYS
h, hr, hour, hours	HOURS
min, mins, minute, minutes	MINUTES

Additional literal values:

- `previous_month` – allowed only for **check period** (used by `TransactionsVolumeCheck` and `TransactionsQuantityCheck`); represents the full previous calendar month.

Key Aspects

Ruleset

The rulesets specify conditions under which transactions should be blocked, allowed, or flagged. If a ruleset's conditions are true, its actions are executed.

Conditions

Conditions are the core of each ruleset. They can be simple or complex, using logical operators to combine multiple checks. Conditions are evaluated using the logical operators specified.

Actions

Actions are executed when a ruleset's conditions are met. These can include any number of predefined actions, such as blocking a card or notifying a user. Actions are executed in the order they appear in the list. If multiple rulesets match with the transaction and contain an action, it will only be returned and executed once.

Decision

If all rulesets return `APPROVED`, the final result = `APPROVED`. If at least one ruleset returns `DECLINED`, the final result = `DECLINED`. If no rulesets return `DECLINED` but at least one ruleset returns `ON_HOLD`, the final result = `ON_HOLD`.

Example Rulesets

Example 1: Blocking Transactions from UHRC Countries for every Partner

Ruleset description

This ruleset blocks any transaction where the acquirer country belongs to the UHRC (Unusual High-Risk Countries) list. It enforces a global AML policy and applies to all partners. Any transaction coming from high-risk jurisdictions is automatically declined and reported through an alert.

```
conditions:
  AND:
    - request_property_check:
        property: transactionData.acquirerCountry
        comparator: IN
        value: {{ vars.UHRC_COUNTRIES }}
        treat_missing_value_as: false
trigger:
  decision: DECLINED
  alert:
    channels: [ YOUTRACK_TICKET ]
```

Example 2: Additional Tenant and User ID Constraints

Ruleset description

This ruleset blocks UHRC transactions specifically for the Acme tenant. It excludes internal or system accounts based on ownerId. When triggered, the user is also blocked at the Antaca level. It represents a stricter, tenant-specific AML policy tailored for Acme.

```
conditions:
  AND:
    - request_property_check:
      property: transactionData.acquirerCountry
      comparator: IN
      value: {{ vars.UHRC_COUNTRIES }}
    - request_property_check:
      property: tenantId
      comparator: =
      value: Acme
    - request_property_check:
      property: balance.ownerId
      comparator: NOT_IN
      value: [ 1,2,3 ]
trigger:
  decision: DECLINED
actions:
  antaca:
    - name: block_resource
      properties:
        reason: fraud_suspected
        resource_type: user
```

Example 3: Structuring pattern in high-risk merchant categories

Ruleset description

This ruleset detects potential structuring (smurfing) behavior. If many small transactions or a high cumulative transaction amount occurs in high-risk MCC categories within one day, an alert is generated. Transactions are approved to avoid blocking—but the AML team is notified for review.

```
conditions:
  OR:
    - transactions_volume_check:
      scope: BALANCE
      by: MERCHANT
      period: "1d"
      amount: 1500000
      currency: PLN
      filters:
        - field: transactionData.mcc
          comparator: IN
          value: {{ vars.HIGH_RISK_MCC }}
        - field: type
          comparator: "="
          value: "DEBIT"
    - transactions_quantity_check:
      scope: BALANCE
      by: MERCHANT
      period: "1d"
      quantity: 10
      filters:
        - field: transactionData.mcc
          comparator: IN
          value: {{ vars.HIGH_RISK_MCC }}
        - field: type
          comparator: "="
          value: "DEBIT"
trigger:
  decision: APPROVED
  alert:
    channels: [ YOUTRACK_TICKET ]
    cooldown_period: "1d"
```

Example 4: High KYC risk level or high risk nationality alert

Ruleset description

This ruleset identifies users with a high KYC risk level or with a nationality belonging to a high-risk jurisdiction. It does not block transactions but produces an alert to highlight increased AML exposure and enable continuous monitoring of risky users.

```
conditions:
  OR:
    - kyc_property_check:
      property: riskLvl
      comparator: =
      value: HIGH
    - kyc_property_check:
      property: nationality
      comparator: IN
      value: {{ vars.UHRC_COUNTRIES }}
      treat_missing_value_as: true
trigger:
  decision: APPROVED
  alert:
    channels:
      - YOUTRACK_TICKET
```

Example 5: Blacklist match on personal or national identifiers

Ruleset description

This ruleset checks if the user matches any record in the blacklist based on personal identifiers such as PESEL, name, surname, nationality, date of birth, or counterparty IBAN. If a match occurs, the transaction is declined and the user is blocked. This ensures that individuals previously flagged for fraud or AML reasons cannot transact further.

```
conditions:
  OR:
    - blacklist_check:
      properties:
        - property: pesel
          kyc_value: pesel
    - blacklist_check:
      properties:
        - property: iban
          request_value: transactionData.contrahentIban
    - blacklist_check:
      properties:
        - property: name
          kyc_value: firstName
        - property: surname
          kyc_value: lastName
        - property: addressCountry
          kyc_value: nationality
        - property: birthDate
          kyc_value: birthDate
trigger:
  decision: DECLINED
  actions:
    antaca:
      - name: block_resource
        properties:
          reason: fraud_suspected
          resource_type: user
```

Example 6: Rapid cross-border card transaction on terminal or ATM detection

Ruleset description

This ruleset detects impossible or highly suspicious card activity across borders. If two transactions occur within 300 seconds but originate from two different countries, the system assumes the card may be compromised. The transaction is declined and an alert is raised, as physical cross-border movement is not feasible in such a short time.

```
conditions:
  AND:
    - request_property_check:
      property: subType
      comparator: IN
      value: [ PURCHASE, ATM_WITHDRAWAL ]
    - request_property_check:
      property: transactionData.captureMode
      comparator: IN
      value: [ MAG, EMV, NFC ]
    - compare_with_last_transaction:
      options:
        within_seconds: 300
        subType: [ PURCHASE, ATM_WITHDRAWAL ]
        context: CARD
        captureMode: [ CONTACT, CONTACTLESS ]
      property: transactionData.countryCode
      comparator: "="
      request_property: transactionData.countryCode
      treat_missing_value_as: false
trigger:
  decision: DECLINED
  alert:
    channels: [ YOUTRACK_TICKET ]
```

Example 7: Unusual debit transaction with balance owner notifications

Ruleset description

This ruleset detects unusual debit transactions in gambling and casino-related merchant categories and notifies the balance owner directly via SMS and EMAIL in addition to raising an internal alert. The MCC codes associated with gambling are defined as a variable (`GAMBLING_MCC`). Both notifications have an optional cooldown period to avoid spamming the user and use a predefined template.

```
conditions:
  AND:
```

```
- request_property_check:
  property: type
  comparator: =
  value: DEBIT
- request_property_check:
  property: transactionData.mcc
  comparator: IN
  value: {{ vars.GAMBLING_MCC }}
trigger:
  decision: DECLINED
  alert:
    channels:
      - YOUTRACK_TICKET
    cooldown_period: 1d
  balance_owner_notifications:
    - type: SMS
      template_name: unusual_transaction_detected
      cooldown_period: 1d
    - type: EMAIL
      template_name: unusual_transaction_detected
      cooldown_period: 1d
```

Example 8: Monthly turnover threshold without extended verification

Ruleset description

This ruleset detects when a user without extended KYC verification exceeds the defined monthly turnover threshold in EUR or PLN. When triggered, the system declines further transactions and requires the user to complete extended verification.

```
conditions:
  AND:
    - kyc_property_check:
      property: kycLevel
      comparator: "!="
      value: EXTENDED
      treat_missing_value_as: true
```

```
- OR:
  - transactions_volume_check:
    scope: USER
    period: "1M"
    amount: 1000000
    currency: EUR
  - transactions_volume_check:
    scope: USER
    period: "1M"
    amount: 4300000
    currency: PLN
```

trigger:

decision: DECLINED

actions:

antaca:

```
- name: extended_verification_required
```

properties:

```
reason: monthly_turnover_exceeded
```

```
resource_type: user
```

Tools

- [kaml](#)
- [yamlkt](#)
- [jackson yaml](#)

Revision #49

Created 5 November 2025 14:21:32

Updated 22 April 2026 10:36:14 by Kamil Szalast