**Scenario: Single or Multi Cash Transactions: Large Significant Transactions**

This scenario is only for cash transactions. Focus (to start with): Customer

**Scenario Objective:** Money launderers may deposit or withdraw significant amounts of cash, either in a single transaction, or in multiple transactions over a period. Such deposits may not appear to be consistent with the type of account or the declared business or activity that the customer is involved with. This scenario detects instances of large cash deposits or withdrawals (over USD 10,000 or comparable cash reporting threshold for different jurisdictions) and detects smaller, multiple deposits or withdrawals over a specified Lookback Period (typically two weeks or 30 days) that aggregate to a significant amount. The scenario enables detection of structuring activity across branches or locations. Thresholds can be modified to enable application of scenarios to various reporting threshold requirements.

**Lookback Period**: 14 Days – Tunable (typically two weeks or 30 days)

**Frequency Period**: 7 Days - Tunable

**Filter involved in scenario:**

* Include Cash Transaction
* Include Customer Retail Accounts Only
* Exclude Canceled and Canceling Transactions
* Exclude Cash Transaction Report Exemption Accounts
* Exclude Distribution, Dividend, Fee, Interest, and Transfer-of-Account Transactions
* Exclude Exempt Customers

**BDF Involved**

****

* CashTransaction\_FrontOfficeTransaction.xml
  + Source(s): FO\_TRXN\_STAGE, FO\_TRXN\_PARTY\_STAGE, FO\_TRXN\_PARTY\_STAGE\_RISK
  + Mode: INSERT
  + Target: CASH\_TRXN
  + Description: Select the set of information from today’s Front Office Transaction, Front Office Transaction Party & Front Office Transaction Party Risk to Insert records In Cash Transaction Table
* AccountToCustomer.xml
  + Source(s): STG\_PARTY\_ACCOUNT\_ROLE\_MAP
  + Mode: INSERT
  + Target: CUST\_ACCT
  + Description: Select the set of information from today’s Party Account Role Map to Insert records In Customer-Account Table. The Account to Customer table identifies the association between customers and accounts, along with the role the customer plays on the account. Associations between customers and accounts are many-to-many; thus, a customer can have roles on multiple accounts and an account may be associated with multiple customers
* AccountCustomerRole.xml
  + Source(s): STG\_CUSTOMER\_ACCT\_ROLE\_MASTER
  + Mode: INSERT
  + Target: CUST\_ACCT\_ROLE
  + Description: Select the set of information pertaining to Party Account Role Master to Insert records In Customer-Account-Role Table. The Account Customer Role table stores the full set of roles a customer can have on an account, and captures the characteristics and permissions (for example, authority to make trades) associated with each role.
* Customer.xml
  + Source(s): STG\_PARTY\_MASTER, STG\_PARTY\_ROLE\_MAP, STG\_PARTY\_TYPE\_MASTER
  + Secondary Source(s): STG\_PARTY\_DETAILS, STG\_CUSTOMER\_ALT\_CCY\_VALUES, STG\_CUST\_TRADE\_EXPERIENCE, STG\_INDUSTRY\_MASTER, CUST\_ID\_DOC, GEOGRAPHY
  + Mode: INSERT
  + Target: `CUST
  + Description: Select the set of information from Party Master, Party Role Map, Party Type Master to Insert records In Customer Table. Secondary datasets from Party Details, Customer Alternate Currency, Customer Trade Experiences, Industry Master, Customer KYC docs reference, Geography tables are also considered during the load.
* Customer\_CPIIndUpd.xml
  + Source(s): STG\_PARTY\_DETAILS
  + Mode: MERGE
  + Target: CUST
  + Description: Merge the set of information from Party Details into Customer Table to update whether Client Profile Information (CPI) documentation is present for this customer / account.
* Customer\_KYCRiskUpd.xml
  + Source(s): CUST\_SUPPLEMENTAL\_ATTR, CUST
  + Mode: UPDATE
  + Target: CUST
  + Description: Calculating risk. If the risk was List driven, then this can ignore that record. If it was BUS/GEO driven, then there is KYC risk.
* Customer\_TotAcctUpd.xml
  + Source(s): CUST, ACCT
  + Mode: MERGE
  + Target: CUST
  + Description: Merge the set of information from Customer, Account tables to update number of Institutional Brokerage accounts linked with institutional customers. It identifies the top-most parent institution that is associated with this account.
* Customer\_UltInstlCustIdUpd.xml
  + Source(s): STG\_PARTY\_MASTER
  + Mode: MERGE
  + Target: CUST
  + Description: Merge the set of information from today’s Party Master table to Customer Table for institutional customers. It identifies the top-most parent institution that is associated with this Customer / Account.
* Customer\_WatchListStage2EffectiveRisk.xml
  + Source(s): WATCH\_LIST\_STAGE2
  + Mode: MERGE
  + Target: CUST
  + Description: Updates a customer’s list risk and effective risk for current day.
* Customer\_WatchListStage2ListRisk.xml
  + Source(s): WATCH\_LIST\_STAGE2
  + Mode: MERGE
  + Target: CUST
  + Description: Calculate the customer's effective risk and set the risk factor. Use nulls for the List Risk and the List Source Code.
* Account.xml
  + Source(s): STG\_MERCHANT\_CARDS, STG\_RETIREMENT\_ACCOUNTS, STG\_SWAPS\_CONTRACTS, STG\_LEASES\_CONTRACTS, STG\_TRADING\_ACCOUNT, STG\_OD\_ACCOUNTS, STG\_CASA, STG\_TRUSTS, STG\_MM\_CONTRACTS, STG\_LOAN\_CONTRACTS, STG\_ANNUITY\_CONTRACTS, STG\_REPO\_CONTRACTS, STG\_CORRESPONDENT\_ACCOUNT, STG\_TD\_CONTRACTS, STG\_CARDS
  + Mode: INSERT
  + Target: ACCT
  + Description: Select the set of information from today’s casa, cards, loan, annuity contracts, leases contracts, merchant cards, retirement accounts, swaps contracts, trade contracts, trusts, mm contracts, trading account, overdraft accounts, correspondent account, repo contracts into Accounts Table. Secondary data from Customer, Party Type Master & Account Alternate Currency values are also considered.
* Account\_AccountCustRiskUpd.xml
  + Source(s): CUST, ACCT, ACCT\_SUPPLEMENTAL\_ATTR
  + Mode: MERGE
  + Target: ACCT
  + Description: Updates account-customer risk, effective risk, effective risk factor for Accounts.
* Account\_EffectiveRiskFactorTxtUpd.xml
  + Source(s): ACCT
  + Mode: MERGE
  + Target: ACCT
  + Description: Updates effective risk factor of Accounts for business vs Geo risk attributes based on the logic when ACCT\_BUS\_RISK\_NB >= ACCT\_GEO\_RISK\_NB THEN 'BUS' ELSE 'GEO'.
* Account\_EffRiskUpdAfterWLRiskRemoval.xml
  + Source(s): CUST, ACCT, ACCT\_SUPPLEMENTAL\_ATTR
  + Mode: MERGE
  + Target: ACCT
  + Description: Updates the account Effective Risk to the maximum of the business risk, geographic risk, and customer risk. The account Effective Risk was already set to the higher of the customer-supplied business and geography risk. List risk is ignored here, as this mapping is where we are removing list risk.
* Account\_OverallEffectiveRiskUpd.xml
  + Source(s): CUST, ACCT, ACCT\_SUPPLEMENTAL\_ATTR
  + Mode: MERGE
  + Target: ACCT
  + Description: Updates the overall account-customer risk, effective risk, effective risk factor for Accounts.
* Account\_WatchListStage2EffectiveRisk.xml
  + Source(s): WATCH\_LIST\_STAGE2
  + Mode: MERGE
  + Target: ACCT
  + Description: Updates an account’s list risk and effective risk for current day.

**Step 1:** First we create the dataset by applying the below filters on tables mentioned

**Tables Involved:**

* + CASH\_TRXN
  + ACCT
  + CUST
  + CUST\_ACCT
  + CUST\_ACCT\_ROLE

**Filters Applied:**

* + Exclude cancelled and canceling transactions

CASH\_TRXN.CXL\_PAIR\_TRXN\_INTRL\_ID is null

* + Exclude Exempted Customers

CUST.CUST\_EFCTV\_RISK\_NB <> -2

* + Cover customers either from all jurisdiction or from Incl\_Jurisdictions\_Lst only

(@All\_Jurisdictions\_Fl='Y' or CUST.JRSDCN\_CD in (@Incl\_Jurisdictions\_Lst))

* + Include Retail Customer Accounts Only

MANTAS\_ACCT\_HOLDR\_TYPE\_CD = 'CR'

* + Cover only specific accounts

ACCT.MANTAS\_ACCT\_BUS\_TYPE\_CD in (@Mantas\_Bus\_Acct\_Type\_Lst)

CASH\_TRXN.MANTAS\_TRXN\_PRDCT\_CD in (@Incl\_Cash\_Trxn\_Prdct\_Type\_Lst)

CASH\_TRXN.MANTAS\_TRXN\_PURP\_CD = 'GENERAL'

* + Cover either all transaction or only form the Incl\_Trans\_Src\_Lst

(@All\_Trans\_Src\_Fl='Y' or CASH\_TRXN.SRC\_SYS\_CD in (@Incl\_Trans\_Src\_Lst))

* + Exclude Cash Transaction Report Exemption Accounts

coalesce (a.CASH\_RPT\_EXMPT\_FL, ' ')<>'Y'

* + Parameter that indicates what account are included for customer focus to monitor

@Primary\_Cust\_Fl = 'N'

* + Account Customer Role will be viewed as discretionary

(CUST\_ACCT\_ROLE.trdng\_auth\_fl = 'Y' or CUST\_ACCT\_ROLE.wdrwl\_auth\_fl = 'Y' or CUST\_ACCT\_ROLE.poa\_fl = 'Y')

* + Include the transactions only which fall between minimum individual transaction amount and maximum individual transaction

CASH\_TRXN.TRXN\_BASE\_AM >= @Min\_Indiv\_Trans\_Amt

and CASH\_TRXN.TRXN\_BASE\_AM <= @Max\_Indiv\_Trans\_Amt

**Step 2:** We divide the customer under the category of

High Risk(HR)

Medium Risk(MR)

Regular Risk(RR)

When max(t.CUST\_EFCTV\_RISK\_NB)>= @Effctv\_Risk\_Cutoff\_Lvl and max(t.d\_ACTVTY\_RISK\_LVL) >= @Actvty\_Risk\_Cutoff\_Lvl then 'HR'

When max(t.CUST\_EFCTV\_RISK\_NB)< @Effctv\_Risk\_Cutoff\_Lvl and max(t.d\_ACTVTY\_RISK\_LVL) < @Actvty\_Risk\_Cutoff\_Lvl then 'RR'

All remaining will be Medium Risk.

**Note:** for this POC, Customer-Risk will be look-up. It will not be computed inline as part of the scenario assessment.

To achieve Next-Gen target state, there has to be multiple parallel pipelines sharing responsibilities between them. Customer-Risk evaluation cannot be run inline as part of scenario-assessment (it has to be computed as part of KYC or CS pipeline(s) and exposed as REST-APIs that can be called from anywhere within the FCC ecosystem)

.

The current approach that calculates customer-risk, activity-risk, etc., in line with scenario-processing logic is not fit to purpose for real-time processing goal. That is legacy multi-aspect logic bolted together in a batch.

**Step 3:** Generate the event based on condition max (transaction amount) and count (transaction count) is

greater than thresholds defined values for high, medium and regular customer.

(Overall\_Risk = 'HR' and

Trans\_Amt >= @HR\_Min\_Trans\_Amt and

Trans\_Ct >=@HR\_Min\_Trans\_Ct)

or

(Overall\_Risk = 'MR' and

Trans\_Amt >= @MR\_Min\_Trans\_Amt and

Trans\_Ct >=@MR\_Min\_Trans\_Ct )

or

(Overall\_Risk = 'RR' and

Trans\_Amt >= @RR\_Min\_Trans\_Amt and

Trans\_Ct >=@RR\_Min\_Trans\_Ct)

**Technical Details:**

The assumption is that transactions will be received as streaming data. For the POC, Customer and Account load will be done in batch.

The plan (as the POC evolves) is to also bring in customer and account data streamed in but decision is yet to be finalized.

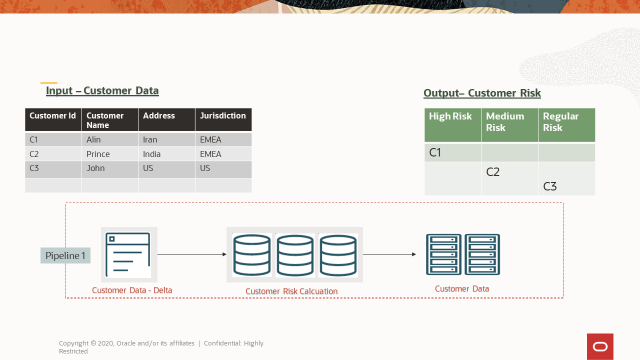
Architecturally Next-Gen will support both batch and streaming.

C1 -> A1 -> T1-> $50000

C1 -> A2 -> T2-> $60000

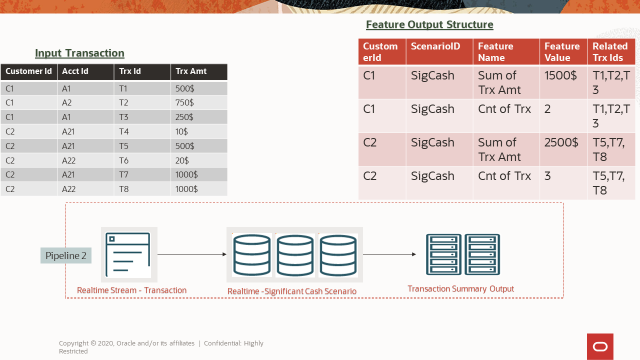
1. Any incoming customers will be categorized on High, Medium, or Regular risk based on Jurisdiction and watchlist. This will be done by Customer Pipeline shown below

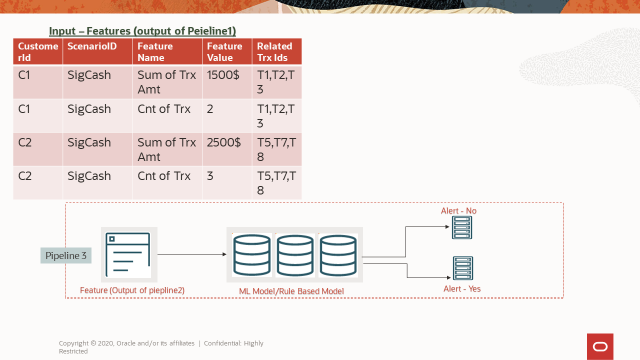
**Conceptual Structure to hold Features against each scenario is illustrated below (WIP):**



**Step-4:**

1. For all the incoming transactions we will generate the features real time and push the result in to the below mentioned format, so that model can consume the result.





**Dev’s view of how a single incoming transaction is assessed for Alert/No-Alert decisioning:**

* For every scenario, there will be a set of ‘features’ (raw attributes, derived attributes, computed ones, aggregates, constants, etc.,) identified (mined) based on analysis of historical transactions, transaction patterns that the said ‘scenario’ analyzed
* The model(s) developed with these features thus are “trained” for the scenario in questions against the desired transaction history (time-period would be configurable – 6 months to 3 years or even more)
* Thus, the trained model deployed for interference understands the transaction patterns and is expected to gauge the behavior of transactions for the party over a period relevant for the scenario in question
* The above aspect is why every incoming transaction can be evaluated for Alert or No-Alert result without requiring the typically batch approach that looks at 14-day to 30-day history to assess
* Every incoming transaction will be stripped off the key attributes to map to the ‘features’ identified for the scenario
* Subset of attribute-to-feature mapping could be straight forward; some could be derived or in some may require calculation to arrive at the value to map a given ‘feature’
  + This operation is expected to take no more than a few microseconds
* The conceptual structure to store the feature values for each incoming transaction is shown above
* With the features mapped, the model is invoked to generate ‘alert or no-alert’ result and relayed down to the alert-store
* The above flow should work for individual incoming transaction, or a transaction-stream received as ‘micro-batch’

The assessment is a continuous process and as each transaction moves to the next step in the pipeline, another incoming transaction will be under process by previous step(s) in the pipeline. Thus, the result generation for every incremental transaction flowing in will be in rapid succession (hopefully in microseconds interval)