Oracle® Banking Accounts Cloud Service

Transaction Balance Service User Guide

Release 14.6.0.0.0
F75304-01
December 2022

Oracle Banking Accounts Cloud Service Transaction Balance Service User Guide, Release 14.6.0.0.0

F75304-01

Contents

Preface

Introduction

The **Transaction Balance Service User Guide** is designed to help you understand the functional and technical designs of transaction and balance services in Oracle Banking Accounts Cloud Service.

Audience

This user guide is intended for the following end Users / User Roles in a Bank:

Table 1 User Roles

User Role	Function
Back office clerk	Input functions for contracts
Back office managers/officers	Authorization functions
Product Managers	Product definition and authorization
End of Day operators	Processing during End of Day/ Beginning of Day
Financial Controller/Product Managers	Generation of reports

Acronyms and Abbreviations

The list of the acronyms and abbreviations that are used in this guide are as follows:

Table 2 Abbreviations

Abbreviation	Definition
KYC	Know Your Customer
EAC	External Account Check
ECA	External Credit Approval
LOV	List of Values
OBACS	Oracle Banking Accounts Cloud Service
OBA	Oracle Banking Accounts
TBS	Transaction Balance Service

List of Topics

Table 3 List of Topics

Topics	Description
Transaction Balance Service	This topic contains information about the Transaction Balance Service, its validations and its features.
Transaction Request Manager	This topic contains information about the Transaction Request Manager and its features.
Transaction Coordinator	This topic contains information about the Transaction Coordinator and its features.
Accounting Entry and Transaction	This topic contains the Accounting Entry and Transaction.
EA Transaction Post Processing	This topic contains the EA Transaction Post Processing and its features.
Deployment Properties and Configurations	This topic contains the Deployment Properties and Configurations.
TBS - Request and Response Flow	This topic contains the Transaction Balance Service - Request and Response flow diagram.

Screenshot Disclaimer

Sample information used in the interface or documents are dummy, it does not exist in real world, and it is for reference purpose only.

1

Transaction Balance Service

Oracle Banking Accounts (**OBA**) is built on a highly scalable cloud-native, data grid architecture with the industry's highest Transactions per second (**TPS**) that scales near proportional. **Transaction Balance Service** (**TBS**) serves as an *accounting* and *balance* engine for **Oracle Banking Accounts**, built using *Helidon* and *Embedded Coherence Grid*.

Helidon is a cloud-native, open-source set of Java libraries used to write microservices that run on a fast web core. **Coherence** ensures for maximum scalability and performance in enterprise applications by providing clustered low-latency in-memory data storage, polyglot grid computing, and asynchronous event streaming. Coherence is used in in-memory data grid that enables application developers and managers fast access to key value data.

Transaction Balance Service supports amount blocks and transactions in an account either from external product processors or from within. External product processors can post accounting entries to the customer accounts by -

- Directly posting **External Accounting** entry (**EA**) transactions
- Placing an amount block called External Credit Approval (ECA) block and post transaction entries post the same.

The entries from both **OBA** and external product processors are processed and customer account and ledger balances are updated.

- Supported Features
- ECA and EA Validations

1.1 Supported Features

Transaction Balance Service supports all activities involving accounting operations and transactions into the accounts in **OBA**. It processes accounting handoff for transactions from other product processors as well as within. It also manages the account balances. The below features are supported in **TBS**.

1. **Amount Block**
   An amount block is the part of the customer account balance reserved for a specific purpose. Amount blocks are placed on an account either on the directions of the customer or at the behest of the bank. When an amount block is set for an account, the balance available for withdrawal is the current balance of the account minus the blocked amount. On expiry of the period for which the amount block is defined, the system automatically updates the amount block. The different types of blocks are as follows.
   - Cheque - User can place an amount block on an account with the cheque number. It has to be a valid non-utilized cheque number issued to the account
   - Referral - System checks the available balance while performing the referral checks for all transactions involving the account. If a transaction involving the account results in the account moving to overdraft, the account and transaction details will be sent to the Referral Queue. Transactions in Referral Queue must be approved for the transaction to be successful.

- RTL - An RTL can be used to request an Amount block or Credit.
- ELCM - Limits maintained for the customer can be used for transactions in the customer account. ELCM limits get updated on limit utilization.
- ECA block - An ECA block is placed for debit transactions initiated from external product processors. We can create, modify, release or rollback ECA blocks.
- Legal block - A Legal or Lien block is placed for debit transactions initiated within OBA with force post as *True*. We can create, release or rollback Legal/ Lien blocks.

2. **Accounting**
   Accounting posting to customer accounts in OBA can either happen directly or from an external system. External systems can trigger accounting entries in OBA via the external accounting interface. All validations are performed and the transactions are posted to the account. Accounting entries can be posted directly to the account - External Accounting (EA), or External Credit Approval (ECA) block followed by External Accounting. We have a facility to create, delete, authorize or delete an EA. Accounting entries are posted directly from OBA as well.

3. **L2 Caching**
   L2 caching enables faster transaction processing by caching the data (like currency, branch, source code preferences, transaction code, override configuration, account details, customer details, and so on) which are used frequently for transaction processing. We can enable, disable account and customer caching whenever not required. Any modification to the above cached data (except for customer details, branch and currency), triggers an event to evict the cached data. The UI is used to refresh all the data in the cache manually.

4. **Coherence Post Processing and Writeback**
   As part of transaction processing, the balance update is handled online. Post the online balance update, transaction is completed in two logical steps called the **Coherence Post Processing** and the **Database Post Processing** or **Writeback** and is carried out in a sequence.

   - **Coherence Post Processing** is performed for authorized transactions. The list of activities are as follows.

     a. Current balance updates

     b. Turnover updates

     c. Dormancy updates

     d. Value dated balance updates

     e. Updates for uncollected processing

     f. Accounting Entries hand-off to statement domain

   - **Database Post Processing** or **Writeback** is the process executed after **Coherence Post Processing**. ECA transactions which are fully utilized/ closed/canceled and authorized EA after post processing are written back to the database. Update of Value dated balance, Book dated balance, Turnover balance happens here. The technology of this is explained in detail.
   - Delete or Purge of Coherence Cache.
   - EA Transaction Cache deletion for completed transactions.

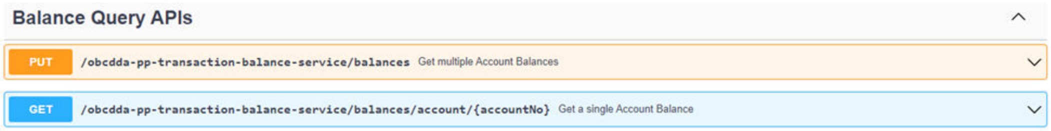
5. **Batch configuration in batch server**

Batch services are defined in the OBA batch services and the Batch service is required to invoke these batches during EOD.

6. **Reval Transaction Posting**
   Revaluation reinstates the Local Currency Balance for FCY accounts and the difference is booked as revaluation profit/loss. Accounting entries are framed and handed over to the GL system.

7. **Balance / Accounting / Amount Block Query API's**
   OBA supports APIs for various operations such as balance fetch, ECA block, EA posting and various queries as listed below.

Figure 1-1 Balance Query APIs

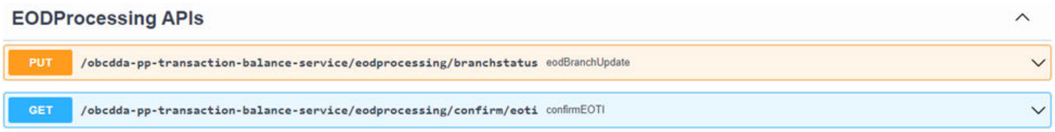


Balance Query APIs

PUT `/obcdda-pp-transaction-balance-service/balances` Get multiple Account Balances

GET `/obcdda-pp-transaction-balance-service/balances/account/{accountNo}` Get a single Account Balance

Figure 1-2 EOD Processing APIs

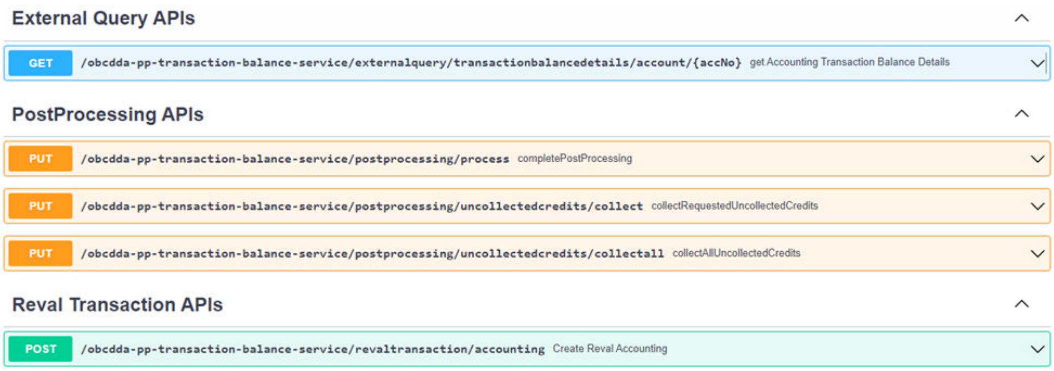


EODProcessing APIs

PUT `/obcdda-pp-transaction-balance-service/eodprocessing/branchstatus` eodBranchUpdate

GET `/obcdda-pp-transaction-balance-service/eodprocessing/confirm/eoti` confirmEOTI

Figure 1-3 External Query, Post Processing and Reval Transaction APIs



External Query APIs

GET `/obcdda-pp-transaction-balance-service/externalquery/transactionbalancedetails/account/{accNo}` get Accounting Transaction Balance Details

PostProcessing APIs

PUT `/obcdda-pp-transaction-balance-service/postprocessing/process` completePostProcessing

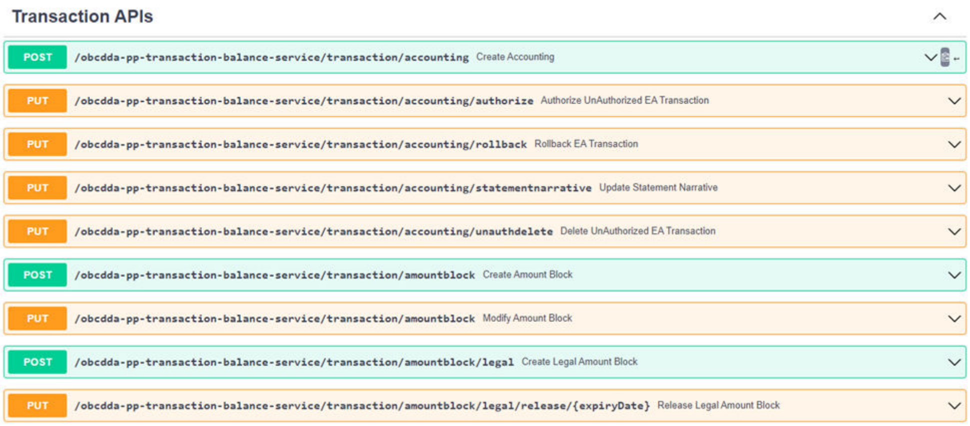
PUT `/obcdda-pp-transaction-balance-service/postprocessing/uncollectedcredits/collect` collectRequestedUncollectedCredits

PUT `/obcdda-pp-transaction-balance-service/postprocessing/uncollectedcredits/collectall` collectAllUncollectedCredits

Reval Transaction APIs

POST `/obcdda-pp-transaction-balance-service/revaltransaction/accounting` Create Reval Accounting

Figure 1-4 Transaction APIs



Transaction APIs

POST /obcdda-pp-transaction-balance-service/transaction/accounting Create Accounting

PUT /obcdda-pp-transaction-balance-service/transaction/accounting/authorize Authorize UnAuthorized EA Transaction

PUT /obcdda-pp-transaction-balance-service/transaction/accounting/rollback Rollback EA Transaction

PUT /obcdda-pp-transaction-balance-service/transaction/accounting/statementnarrative Update Statement Narrative

PUT /obcdda-pp-transaction-balance-service/transaction/accounting/unauthdelete Delete UnAuthorized EA Transaction

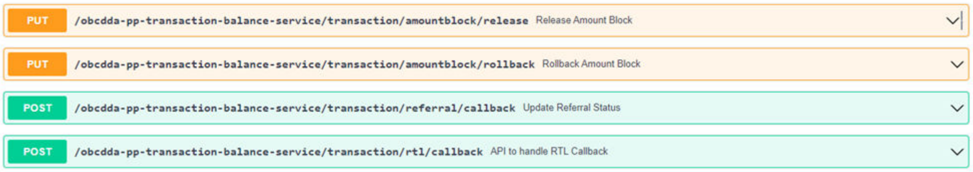
POST /obcdda-pp-transaction-balance-service/transaction/amountblock Create Amount Block

PUT /obcdda-pp-transaction-balance-service/transaction/amountblock Modify Amount Block

POST /obcdda-pp-transaction-balance-service/transaction/amountblock/legal Create Legal Amount Block

PUT /obcdda-pp-transaction-balance-service/transaction/amountblock/legal/release/{expiryDate} Release Legal Amount Block

Figure 1-5 Transaction APIs (contd.)



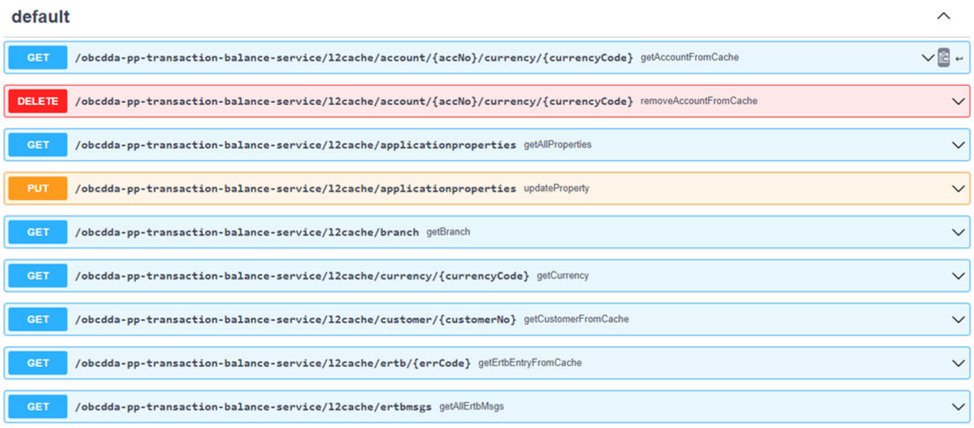
PUT /obcdda-pp-transaction-balance-service/transaction/amountblock/release Release Amount Block

PUT /obcdda-pp-transaction-balance-service/transaction/amountblock/rollback Rollback Amount Block

POST /obcdda-pp-transaction-balance-service/transaction/referral/callback Update Referral Status

POST /obcdda-pp-transaction-balance-service/transaction/rtl/callback API to handle RTL Callback

Figure 1-6 Default APIs



default

GET /obcdda-pp-transaction-balance-service/l2cache/account/{accNo}/currency/{currencyCode} getAccountFromCache

DELETE /obcdda-pp-transaction-balance-service/l2cache/account/{accNo}/currency/{currencyCode} removeAccountFromCache

GET /obcdda-pp-transaction-balance-service/l2cache/applicationproperties getAllProperties

PUT /obcdda-pp-transaction-balance-service/l2cache/applicationproperties updateProperty

GET /obcdda-pp-transaction-balance-service/l2cache/branch getBranch

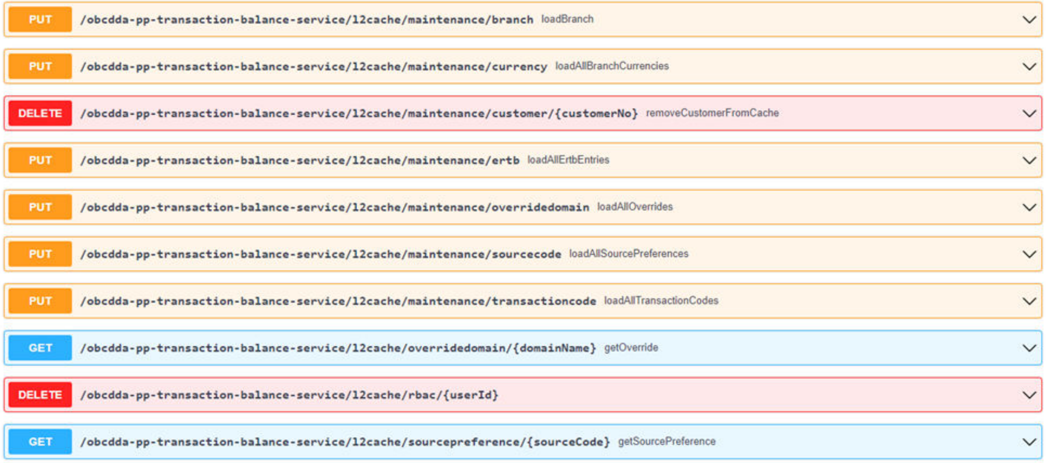
GET /obcdda-pp-transaction-balance-service/l2cache/currency/{currencyCode} getCurrency

GET /obcdda-pp-transaction-balance-service/l2cache/customer/{customerNo} getCustomerFromCache

GET /obcdda-pp-transaction-balance-service/l2cache/ertb/{errCode} getErtbEntryFromCache

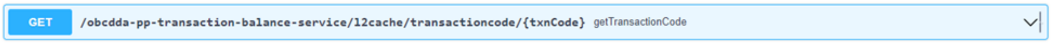
GET /obcdda-pp-transaction-balance-service/l2cache/ertbmsgs getAllErtbMsgs

Figure 1-7 Default APIs (contd.)



```
PUT     /obcdda-pp-transaction-balance-service/l2cache/maintenance/branch loadBranch
PUT     /obcdda-pp-transaction-balance-service/l2cache/maintenance/currency loadAllBranchCurrencies
DELETE  /obcdda-pp-transaction-balance-service/l2cache/maintenance/customer/{customerNo} removeCustomerFromCache
PUT     /obcdda-pp-transaction-balance-service/l2cache/maintenance/ertb loadAllErtbEntries
PUT     /obcdda-pp-transaction-balance-service/l2cache/maintenance/overridedomain loadAllOverrides
PUT     /obcdda-pp-transaction-balance-service/l2cache/maintenance/sourcecode loadAllSourcePreferences
PUT     /obcdda-pp-transaction-balance-service/l2cache/maintenance/transactioncode loadAllTransactionCodes
GET     /obcdda-pp-transaction-balance-service/l2cache/overridedomain/{domainName} getOverride
DELETE  /obcdda-pp-transaction-balance-service/l2cache/rbac/{userId}
GET     /obcdda-pp-transaction-balance-service/l2cache/sourcepreference/{sourceCode} getSourcePreference
```

Figure 1-8 Default APIs (contd.)



```
GET     /obcdda-pp-transaction-balance-service/l2cache/transactioncode/{txnCode} getTransactionCode
```

8. **EOD flow, Configuration and Invocation**
   As part of the End of Day (EOD) process, OBA allows the user to execute several functions every day on a routine basis. These functions are run at various stages of the EOD process.

   Refer to the *EOD Configuration User Guide* for the EOD flow, Configuration and Invocation.

9. **Referral functionality and APIs properties**
   TBS checks the available balance (not the current balance) while performing the referral checks for all transactions involving the account. If a transaction involving the account results in the account moving to overdraft, then the account and transaction details are sent to the Referral Queue.

   Overrides, errors and information codes that occur as part of EAC/ECA/EA are maintained and used by Transaction Balance Service while processing the transaction. These error codes are mapped to different exception categories like Account validation exceptions, Balance exceptions, Limit exceptions and are used in ECA/EA processing.

   When a transaction request is processed, different exceptions occur. Each error code which is mapped to one of the exception queues, is subject to approval processing according to the severity level. In the Referral Queue, all the exceptions are grouped by queue type and the user can approve any transaction by drilling down to the transaction and by checking the balance and exceptions. Adjacent to every queue name, the number of transactions in the specific exception queue pending for approval is displayed. Subsequently, transactions in the referral queue are approved or rejected.

10. **Interest and Charges (IC) functionality**
    Interest is computed and applied on accounts with balances. Interest component is set up once, and by using it, the system calculates and applies interest on accounts. The system automatically computes and applies interest on all the balance type accounts. Interest is calculated using the interest rules defined. The user can define interest rules to suit the particular requirements of the bank.

    Refer to the *Interest and Charges User Guide* for the Interest Configurations.

1.2 ECA and EA Validations

TBS validates any accounting transaction to the account. The validations in any transaction, ECA and EA are as follows.

- **Payload** - the Payload is validated for the correctness of every tag and its value.
- **Account** (EAC checks) - the account related validations for the upcoming debit/ credit are performed. The account's statuses such as *No_Credit*, *No_Debit*, *Inactive*, or *Dormant* are checked. In case of a multicurrency account, the sub-account pertaining to the passed currency is resolved and used in transaction processing. The account related validations for the external transactions is performed based on the request (for Debit/Credit Indicator) and a response is sent to the external system.
- **Customer** - For every transaction, the customer is validated for *Frozen*, *Bankrupt* or *Whereabouts Unknown*.
- **Balance and Limit** - the account balance is validated for every debit transaction. While checking for account's sufficient funds for the debit transaction to go through, in case the account does not have sufficient balance, TBS checks for Internal limit which includes *Advance against Uncollected Funds*, *Daylight*, *Temporary Overdraft* and *External Limit* which includes *RTL* and *ELCM* utilization.

Post a successful validation for the debit transaction, the amount is blocked in the account and a response is sent to the external system. Transactions from external system is carried out in two steps, direct EA call or by an ECA request followed by an EA request.

ECA Block

This stage is used to perform account related validations and balance checks for the upcoming external debit transactions. The various checks are -

- Account related validations for the debit
- Account balance check for the debit
- Post successful validation, amount blocked in the account
- Response sent to the external system

EA Transaction

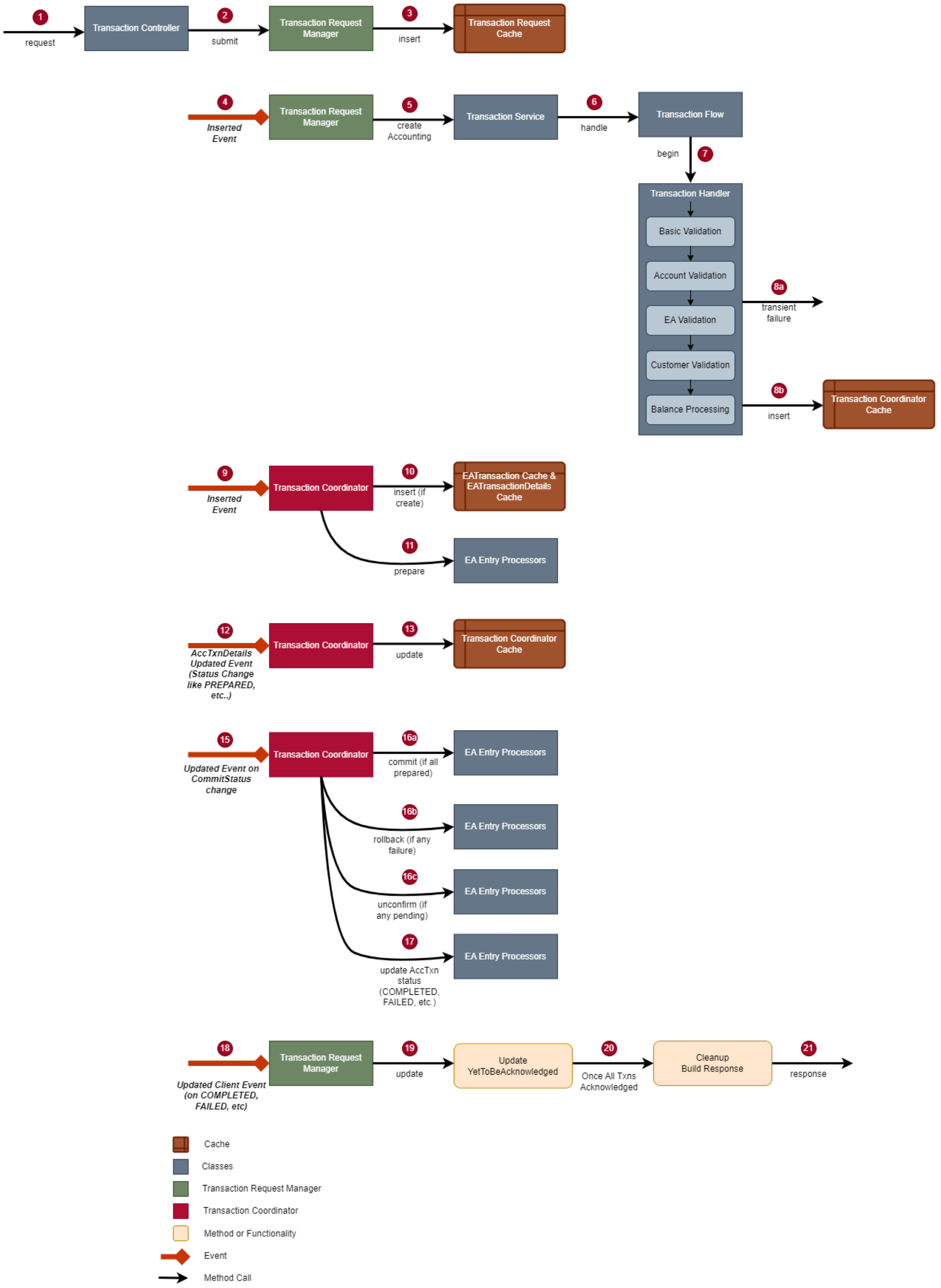
Actual debit on accounts for transactions against ECA is performed on the account. On request, accounting debits the account based on the ECA approved amount. The requested amount is debited from the account by releasing the block. When an EA request comes with a credit; this is performed by validating the account and the customer, and the balance gets updated post successful completion of the transaction.

A direct EA call can also happen without an ECA request. In case of a direct EA request, account validation, customer validation, account balance check is done and on successful validation, the transaction is completed with a response to the external system. The API's of ECA and EA are designed to work based on coherence asynchronous cache events.

2 TBS - Request and Response Flow

The end-to-end **Transaction and Balance Service** request and response flow is as depicted below.

Figure 2-1   TBS - Request and Response Flow



3

Transaction Request Manager

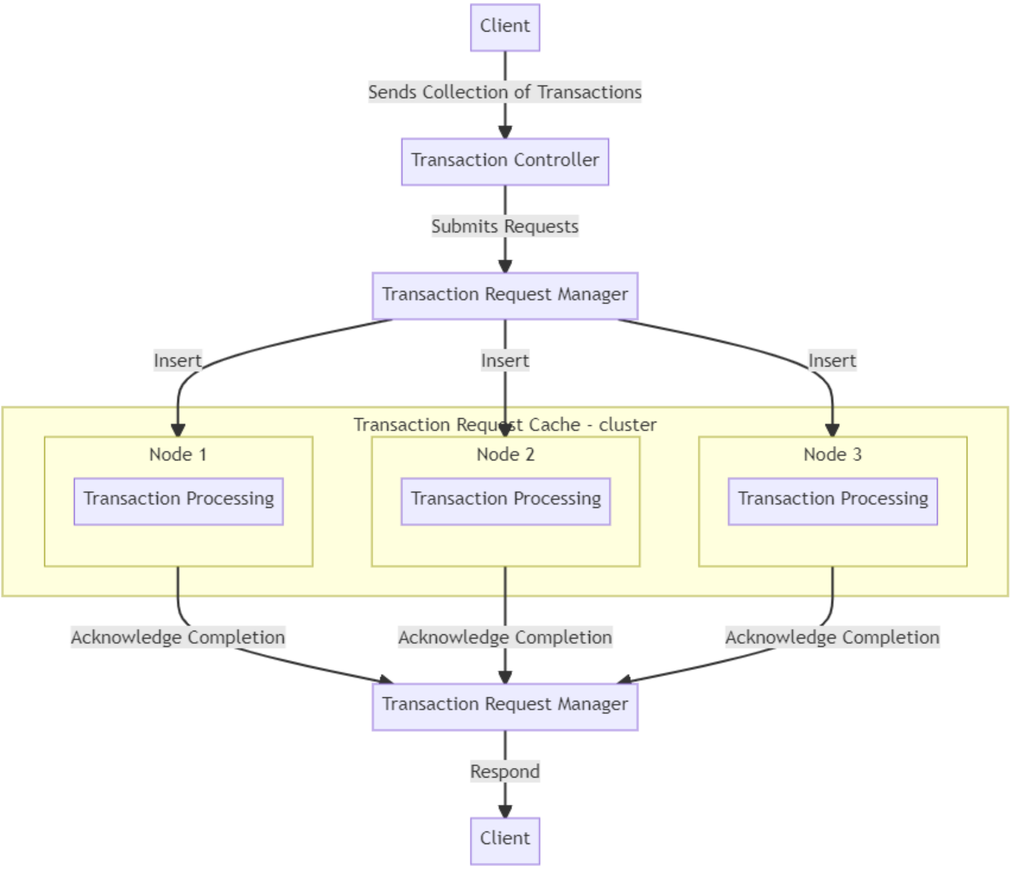
Purpose

The **Transaction Request Manager** handles the following.

- Collection of transaction requests
- Distributes the load to cluster
- Client connection
- Collation of results and response to the client
- Transient failures
- Event Driven flow of execution (Non-Blocking & Asynchronous)
- Fault tolerant (with certain conditions)

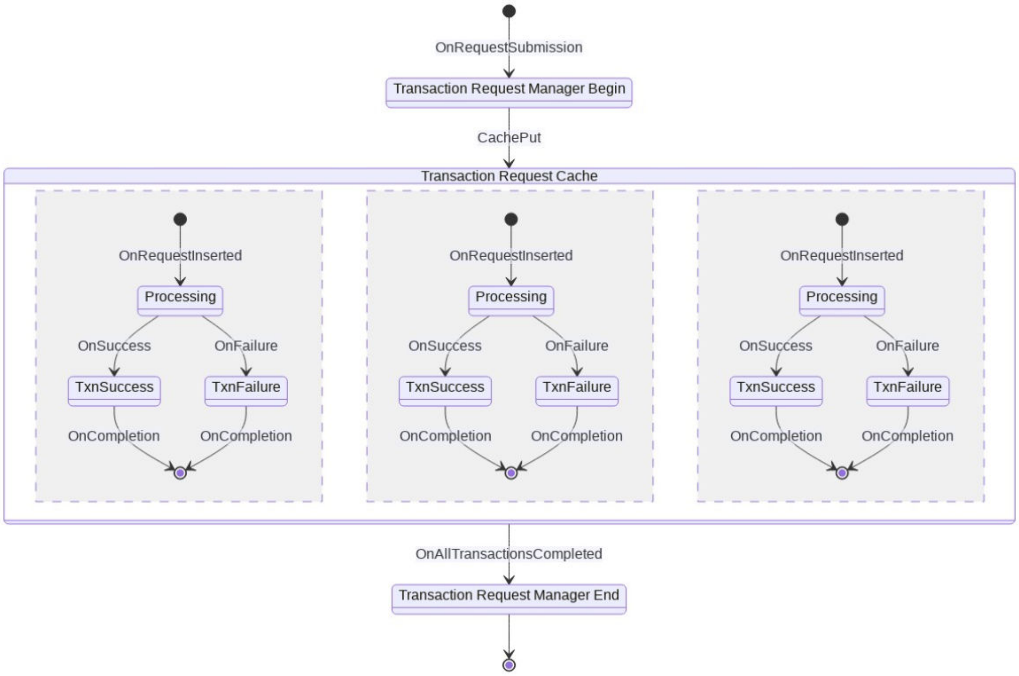
Flow Chart:

Figure 3-1 Transaction Request Manager Flow Chart



State Chart:

Figure 3-2 Transaction Request Manager State Chart



4

Transaction Coordinator

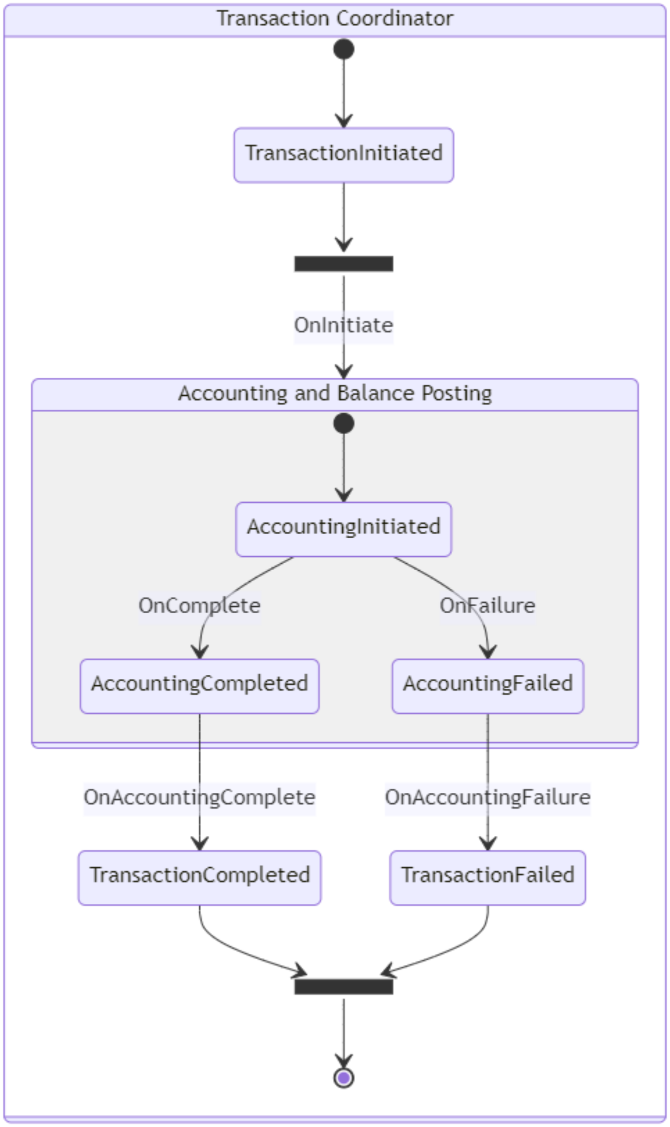
Purpose

The **Transaction Coordinator** handles the following types of transactions.

- Atomicity and Concurrency in Balance Processing of a transaction
- Multi-legged transactions
- Single legged transactions with one step for better performance
- Event Driven flow of execution (Non-Blocking)
- EntryProcessors should be idempotent to maintain state
- Fault Tolerant

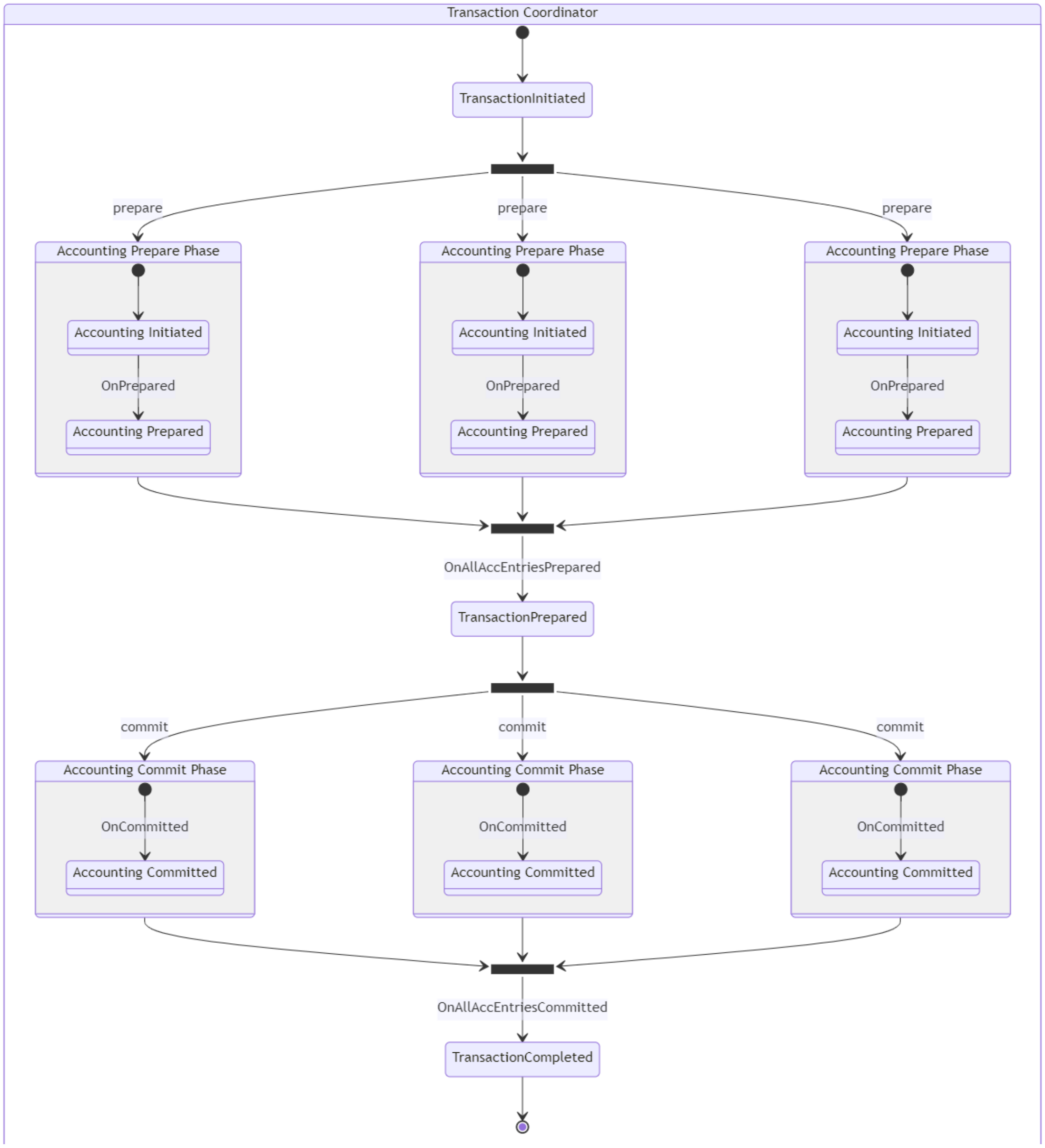
Single Leg Processing:

Figure 4-1   Transaction Coordinator Single Leg Processing



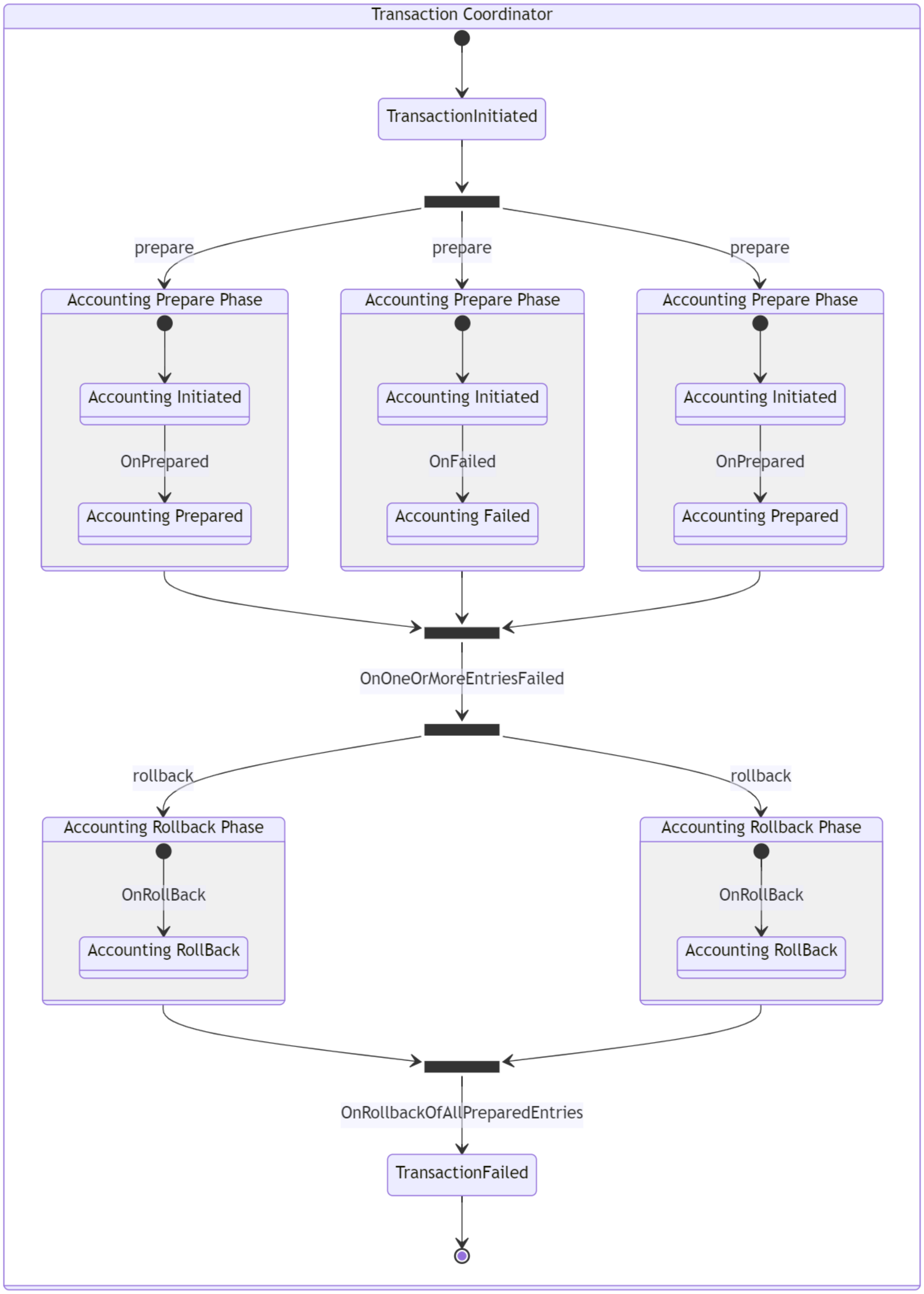
Multi Leg Processing: (Success)

Figure 4-2 Transaction Coordinator Multi Leg Processing - Success



Multi Leg Processing: (Failure)

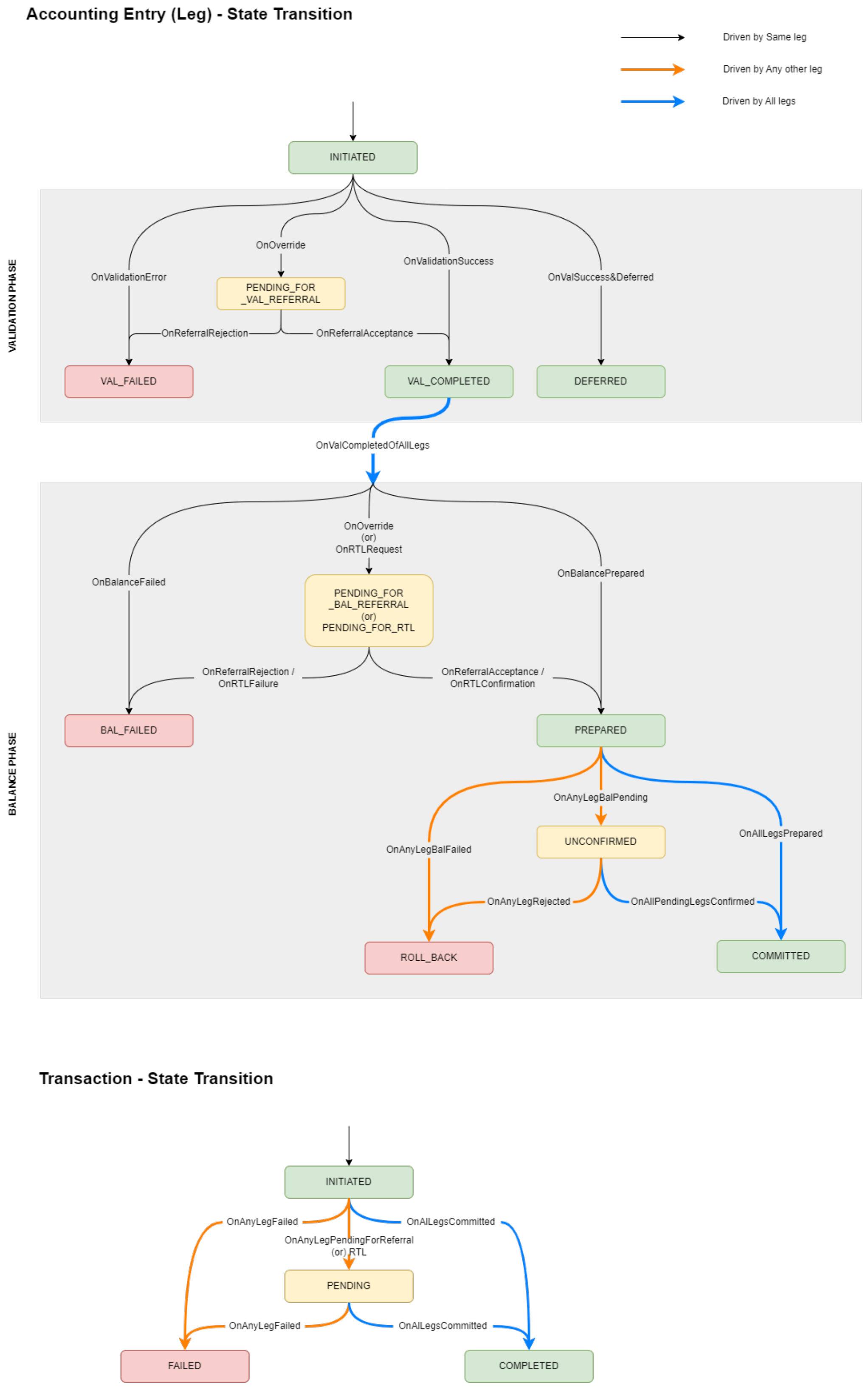
Figure 4-3 Transaction Coordinator Multi Leg Processing - Failure



5 Accounting Entry and Transaction - State Transition

The **Accounting Entry and Transaction - State Transition** flow is as depicted below.

Figure 5-1 Accounting Entry and Transaction - State Transition



Accounting Entry (Leg) - State Transition

Transaction - State Transition

6 EA Transaction Post Processing

EA Transaction Post Processing is performed in two stages namely,

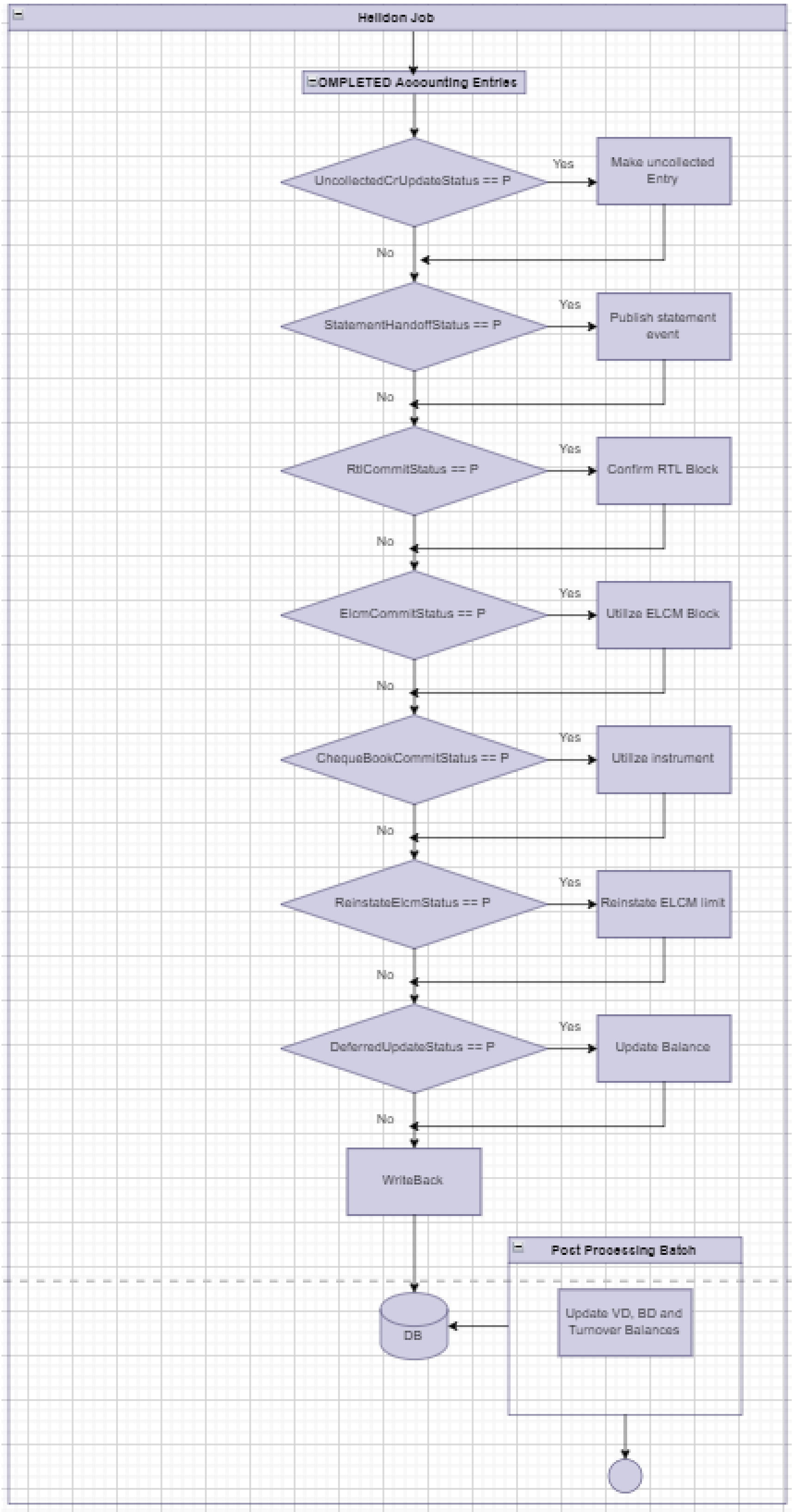
1. **Coherence Post-Processing**
2. **Database Post-Processing**

The **Coherence Post-Processing** is a **Helidon** job which is scheduled to run every minute. This job fetches all the completed accounting entries and performs further processing as needed based on the different flags set in it. These flags are updated during balance processing as required. Once an accounting entry is processed by coherence post-process job, it will be marked as "Coherence Post-Processing is Completed", and then it is written to the database using coherence Write-back. Coherence invokes Write-back for all accounting entry inserts/updates but the Write-back implementation will allow the database write for an accounting entry only if Coherence Post-Processing is 'Completed' for that entry.

The **Database Post-Processing** spring-scheduled batch job picks up these accounting entries and updates **Value-Dated**, **Book-Dated** and **Turnover** balances; and marks the entry as "*Database Post-Processing Completed*". These entries do not require any further processing. For an accounting entry, the **Coherence Post-Processing** batch job executes all pending coherence post-processing sequentially as shown in the below diagram and if it completes successfully, it marks the entry as "*Coherence Post-Processing Completed*".

If one/more processing fails, it increments the Coherence post-processing count in the entry by one. In the next **Coherence Post-Processing** batch job, the same entry is picked up again and all pending processing will be attempted. If there is a failure, the coherence post-processing count gets incremented. This repeats until the accounting entry exceeds the configured number of coherence Post-Processing attempts. Once the entry exceeds the allowed post-processing attempts, it will be marked as "*Retry-Exceeded*" and this entry will not be considered in the future **Coherence Post-Processing Job** executions.

Figure 6-1 EA Transaction Post Processing



- Statement Handoff
  As part of post processing, the leg level accounting entries are handed off to statement domain which are used in the account statement generation.
- Handoff to Interest Domain
  As part of daily batch, **TBS** indicates and hands-off those accounts having transactions for the day to the IC domain.

6.1 Statement Handoff

As part of post processing, the leg level accounting entries are handed off to statement domain which are used in the account statement generation.

Every time there is an update to the accounting entry, data is sent to the statement domain which is consumed in the account statement.

6.2 Handoff to Interest Domain

As part of daily batch, **TBS** indicates and hands-off those accounts having transactions for the day to the IC domain.

Interest accrual re-computation happens only to those accounts having transactions for the day and if there are no transactions for the account, then the previous interest accrual computation is used thereby enhancing the IC batch performance.

7

Deployment Properties and Configurations

The information about the different deployment properties and configurations in OBA is provided in the below table.

Table 7-1 Deployment Properties and Configurations

Property Name	Value	Can be modified at runtime
dda.tbs.account.service.name	OBCDDA-PP-ACCOUNT-SERVICES	No
dda.tbs.branchparameter.service.name	DDA-CONFIG-BRANCHPARAMETERS-SERVICES	No
dda.tbs.sourcepreference.service.name	DDA-CONFIG-SOURCE-PREFERENCE-SERVICE	No
dda.tbs.transactioncode.service.name	DDA-CONFIG-TRANSACTIONCODE-SERVICES	No
dda.tbs.customer.service.name	CMC-CUSTOMER-SERVICES	No
dda.tbs.referralqueue.service.name	DDA-REFERRAL-QUEUE-SERVICE	No
eureka.serviceUrl.default		No
dda.tbs.enable.rbac	Y or N	Yes
dda.tbs.enable.requestlog	Y or N	Yes
dda.tbs.ddaservice.branchcode	006	No
dda.tbs.ddaservice.userid	ADMINUSER1	No
dda.tbs.ddaservice.entityid	DEFAULTENTITY	No
dda.tbs.postprocessing.parallel.legs.count	2	Yes
dda.tbs.postprocessing.retry.count	2	Yes
dda.tbs.uncollected.collection.parallel.entries.count	2	Yes
dda.tbs.account.l2cache.enable	Y or N	Yes
dda.tbs.accountdetails.provider	internal or external	No
dda.tbs.customer.l2cache.enable	Y or N	Yes
dda.tbs.customerdetails.provider	internal or external	No
dda.tbs.chequedetails.provider	internal or external	No

Index