Transaction Event

- Start Transaction
- Update Tranaction
- Transaction Failed
- Tranaction Completed

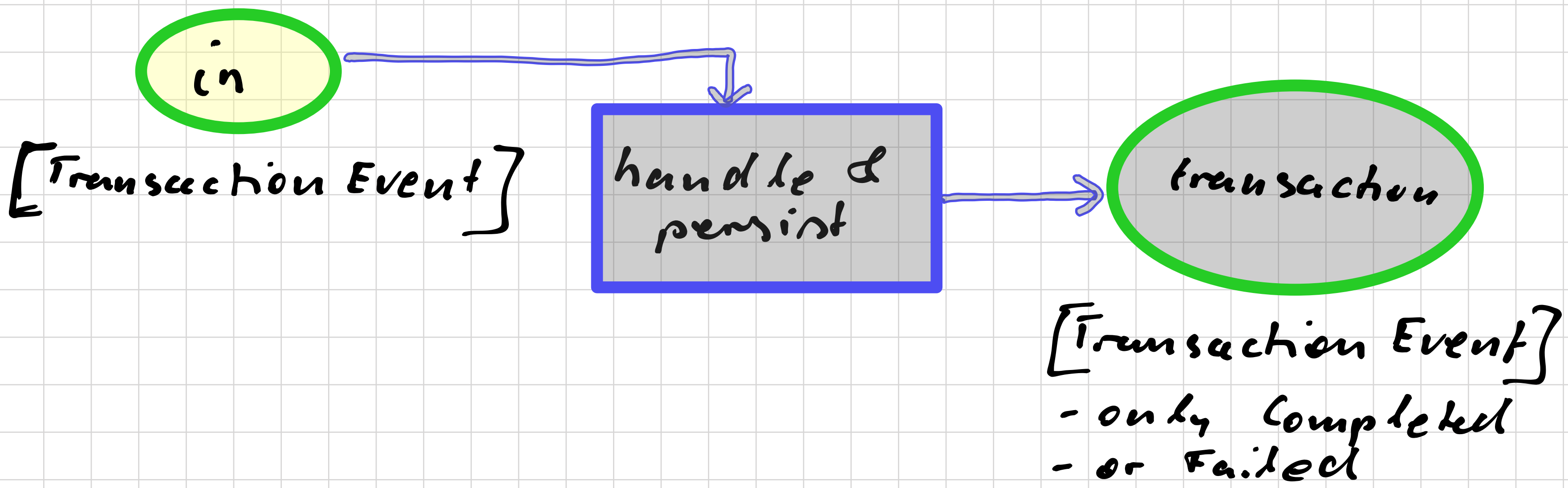
The Start Tranaction Event will create a new tranaction with a given id. If the id already exists, the event will be ignored.

The Update Transaction Event will update an existing Tranaction with a given worklist. If the worklist has no pending items after the update, a Transaction completed event is generatec.
If the worklist has any failed items after the update, a Transaction failed event is generated
If the update references a non-existent transaction, the event will be ignored.

A Transaction failed Event on the inbound stream fails the given Transaction regardless of it's worklist state and passes the Transaction Failed downstream.

A Transaction Completed on the inbound stream completes the given Transaction regardless of it's worklist state and passes the Transaction Completed downstream.

The transaction state is persisted, so that it survives a container restart.



! Transactions do not timeout by themselves. However, if the underlying message has a TTL configured, the worklist may timeout and that will also fail the transaction.

Application log:

The application log will log all Transaction Started, Completed and Failed Events.

CBE:

The CBE support will generate a CBE xml message for each Started, Completed or Failed Transaction event.

! All Transaction processing happen asynchronous. Failures to generate CBE Events or application logs will not cause the business transaction to fail.

$\Rightarrow$ Any Processing that should cause a failure of the business transaction must be included in the dispatcher Flow.

Transaction manager

- should create a new transaction
- should ignore Transaction Started for existing Transactions
- should update correctly
- should generate „completed“ „failed“ or „started correctly
- should survive a container restart