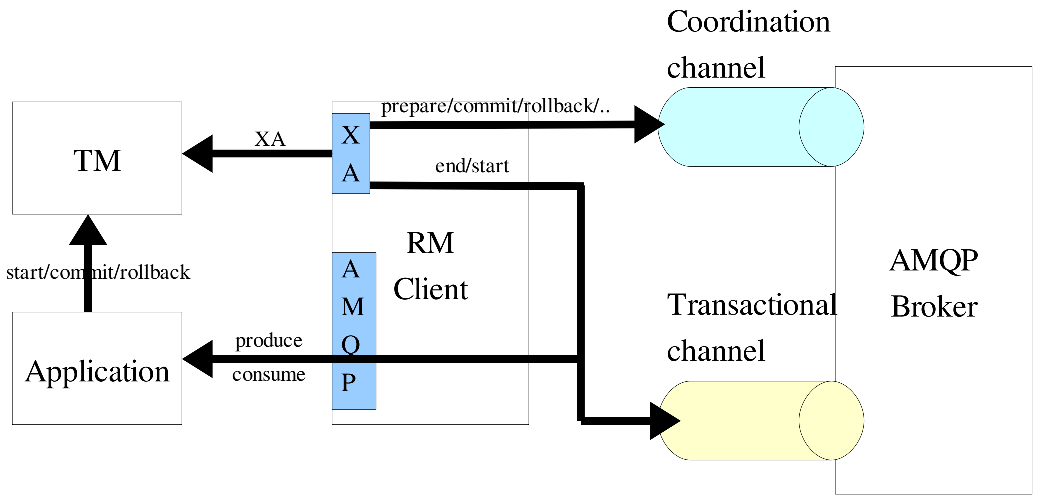
2.2.12 The Distributed Transaction Classes

The distributed transaction classes provide support forthe X-Open XA architecture.

The dtx-demarcation class is used to demarcatetransaction boundaries on a given channel that is subsequently used to performAMQP native transactional work (produce publish messages). Transaction coordination and recovery operations are provided by the dtx-coordination class.

Both OMG OTS and JTS/JTA models rely on “Resource Manager Client”, RM Client, instances (object identified by Rmids through the xa_switch in c/c++or XAResource instances in Java) that implement the XA interface for the underlying resource to participate with a global transaction.

As depicted on the following figure, a Transaction Manageruses RM Client XA interfaceto demarcate transaction boundaries and coordinate transaction outcomes. RM Clients use the dtx-demarcation class to associate transactional workwith a transactional channel. The transactional channel is exposed to the application driving the transaction. The application can then use the transactional channel to transactionally produce and consume messages. RM clients use dtx-coordination to propagate transaction outcomes and recovery operations to the AMQP broker. A second coordination channel can be used for that purpose.



2.2.12.1 The Distributed Transaction Demarcation Class

The dtx-demarcation class allows a channel to be selected for use with distributed transactions and the transactional boundaries for work on that channel to bedemarcated.

Class semantics

The semantics of the dtx class are as follows:

1. Resource Manager Client asks for server XA support on a transactional channel (dtx-demarcation.select).

2. The transaction manager informs Resource Manager Client of transaction association. Resource Manager Client invokes dtx-demarcation.start on its transactional channel.
3. The application uses Resource Manager Client transactional channel to do work within the scope of the transaction branch (publish, consume, ack)
4. The Transaction Manager informs Resource Manager Client that transactional work ends. Resource Manager Client invokes tdtx-demarcation.end on its transactional channel.

2.2.12.2 The Distributed Transaction Coordination Class

The dtx-coordination class allows the transaction managerto coordinate transaction outcomes and to drive transaction recovery.

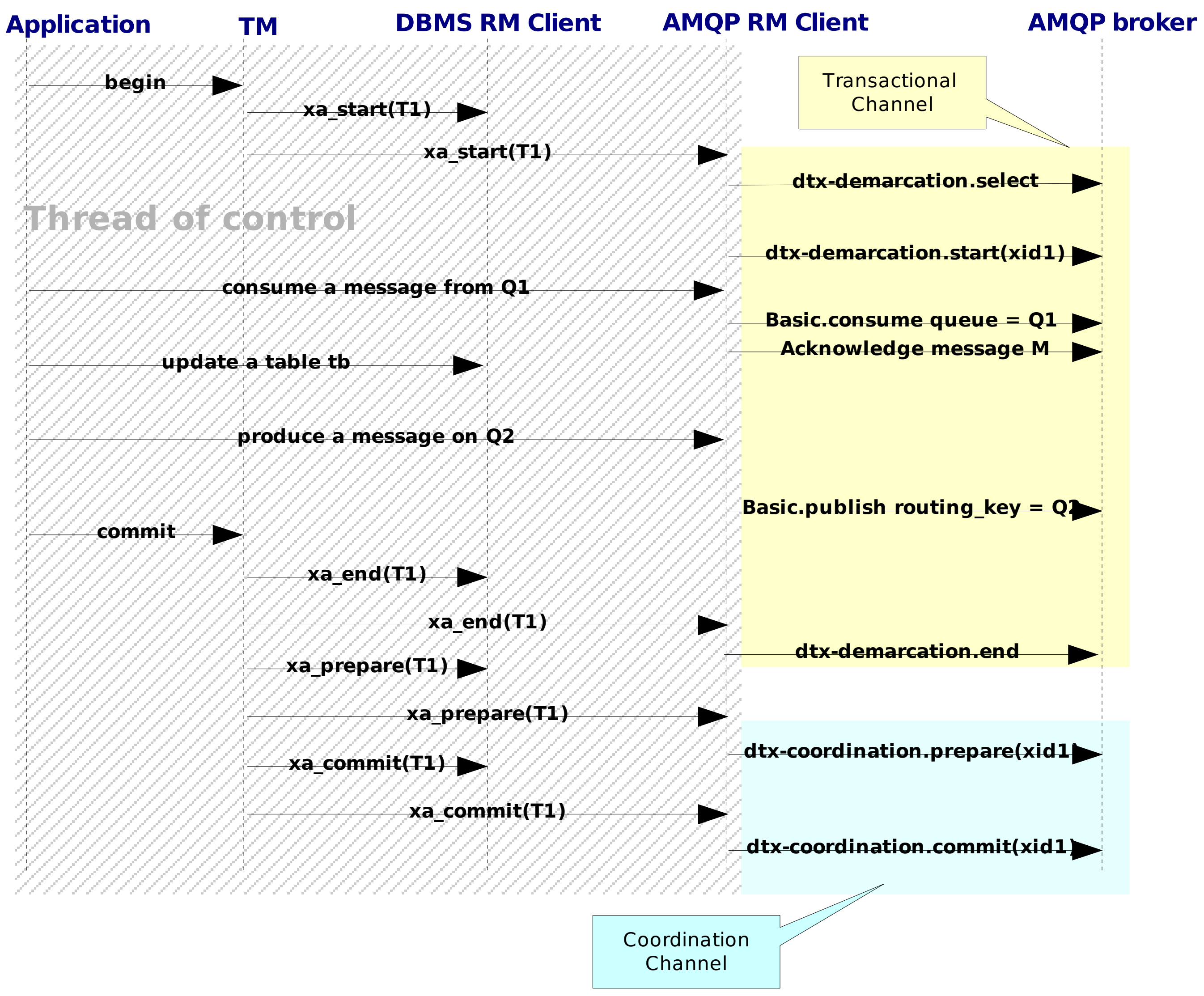
Class semantics

The semantics of the dtx-coordination class are as follows:

1. Transaction Manager demarcates transactions through Resource Manager Client and application uses Resource Manager Client transactional channel to perform transactional work.
2. Transaction Manager coordinates transaction outcome on its Resource Manager Clients. Resource Manager Client invokes corresponding dtx-coordination methods (prepare, commit, rollback).
3. After recovering from a failure, Transaction Manager drives recovery through the Resource Manager Client that invokes corresponding dtx-coordination methods (recover, forget, commit, rollback).

2.2.12.3 Distributed Transaction Use Cases

The following diagram illustrates a messaging scenario where an application “Application” transactionally consumes a message from a queue Q1 (using transaction T1 achieved through the transaction mangerTM). Based on the consumed message, the application updates a database table Tb using DBMS and produces a message on queue Q2 on behalf transaction T1.



The purpose of the dtx-demarcation.select operation is for the server to optimize handling of distributed transactions. The Resource Manager client MUST use this method at least once on a channel before performing distributed transactional work. A channel that is selected for XA work cannot be de-selected.

Any work performed within transactional blocks (a transactional block is delimited by dtx-demarcation .start/dtx-demarcation .end) is done on behalf a transaction branch identified by Xid. Any other operation outside a transactional block is non-transactional.

Only acknowledged consumed messages are seen as being transactionaly consumed.

Any channel can be used for performing transaction coordination and recovery operations .

Implementation notes

- A Resource Manager Client can use the same channel for invoking demarcation and coordination methods.
- When a transactional channel is closed then the work done under the currently associated

transaction branch is rolled back.

- This is the responsibility of the broker implementation to maintain a list of indoubt and heuristically completed transaction branches. The corresponding Xids arereturned to the Transaction Manager when the method recover is invoked.
- It is possible for more than one channel to be associated with the same Xid. All channels will be disassociated (end) from the given Xid before the transaction outcome commands are called.
- IT is the responsibility of the broker to return an error code when coordination methods are called with an unknown or still associated Xid.