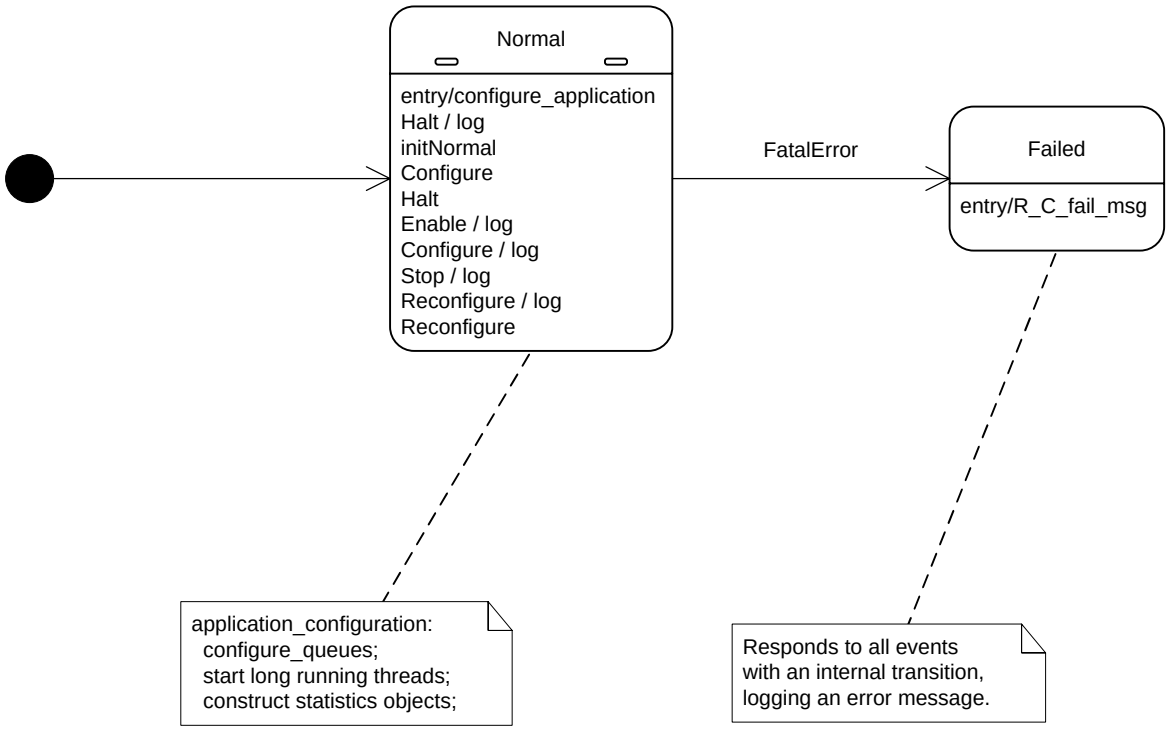
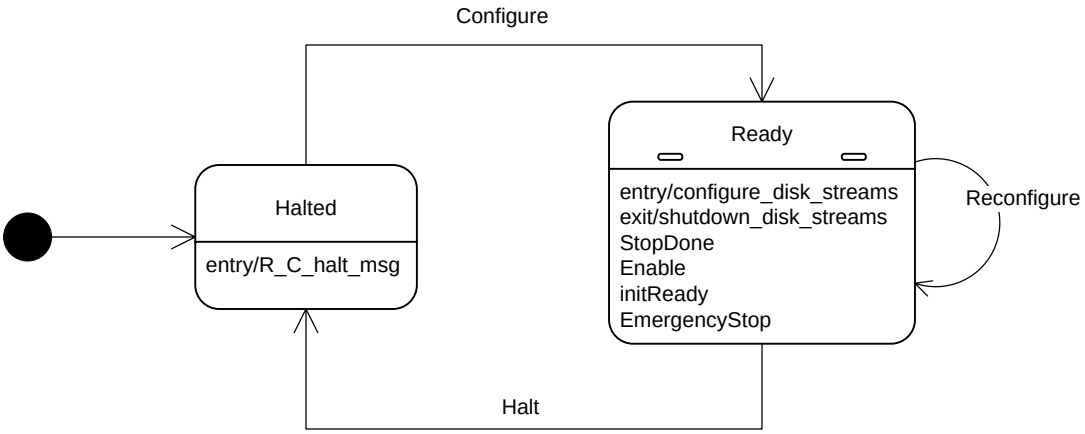
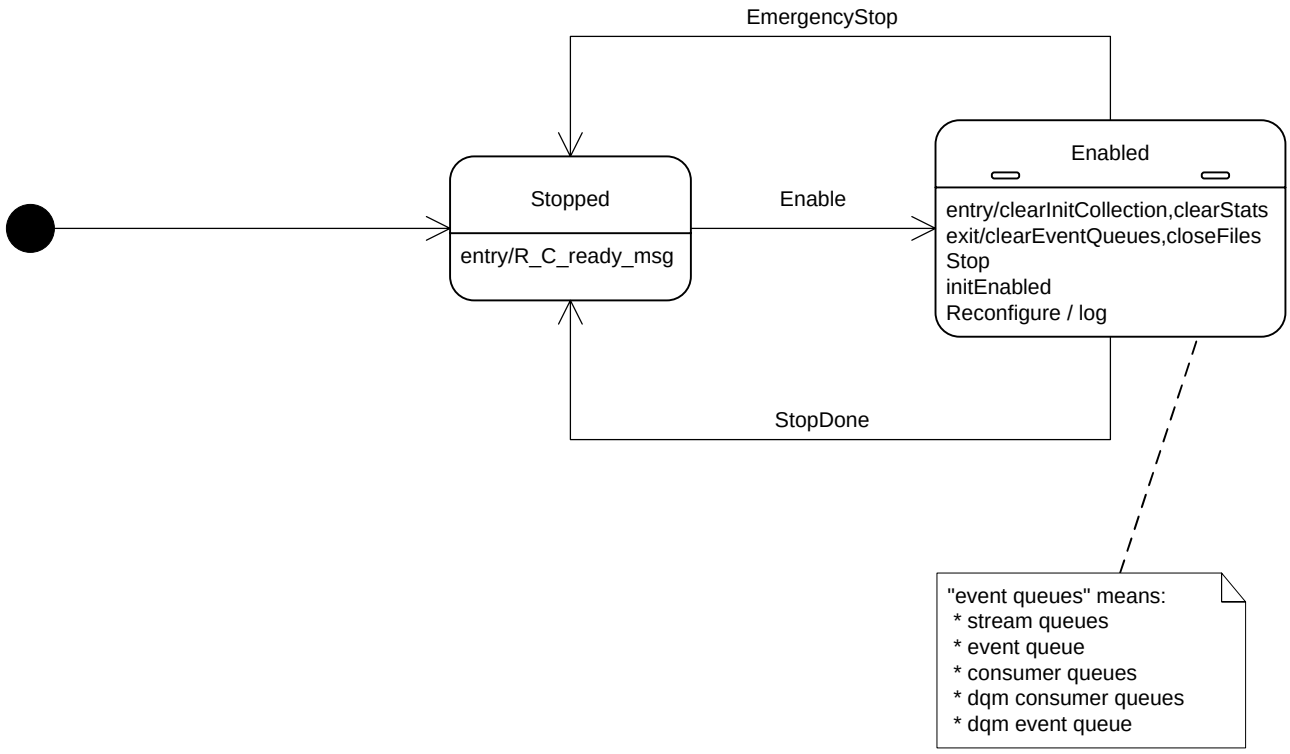
Top-level states



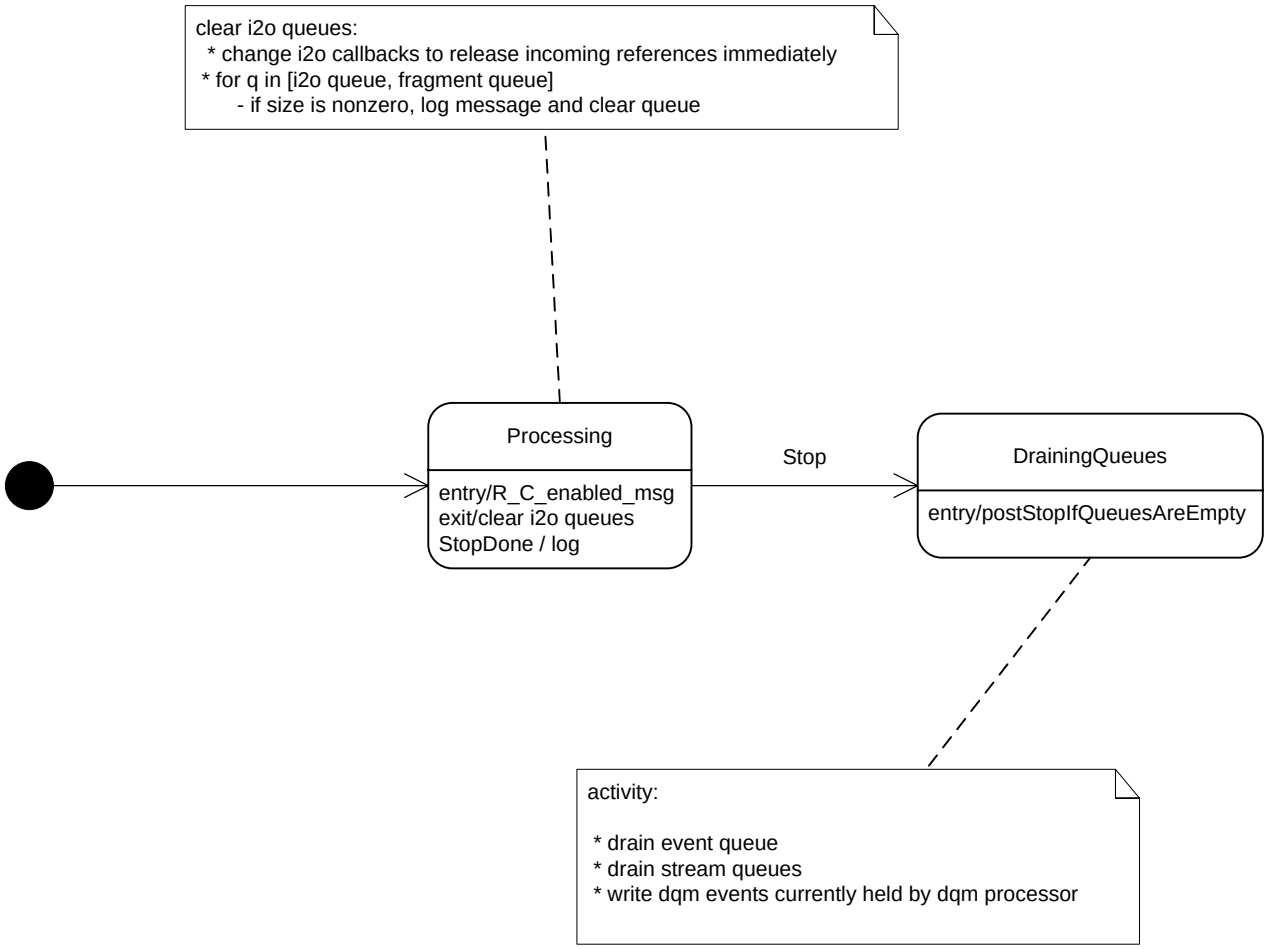
Substates for *Normal* state.

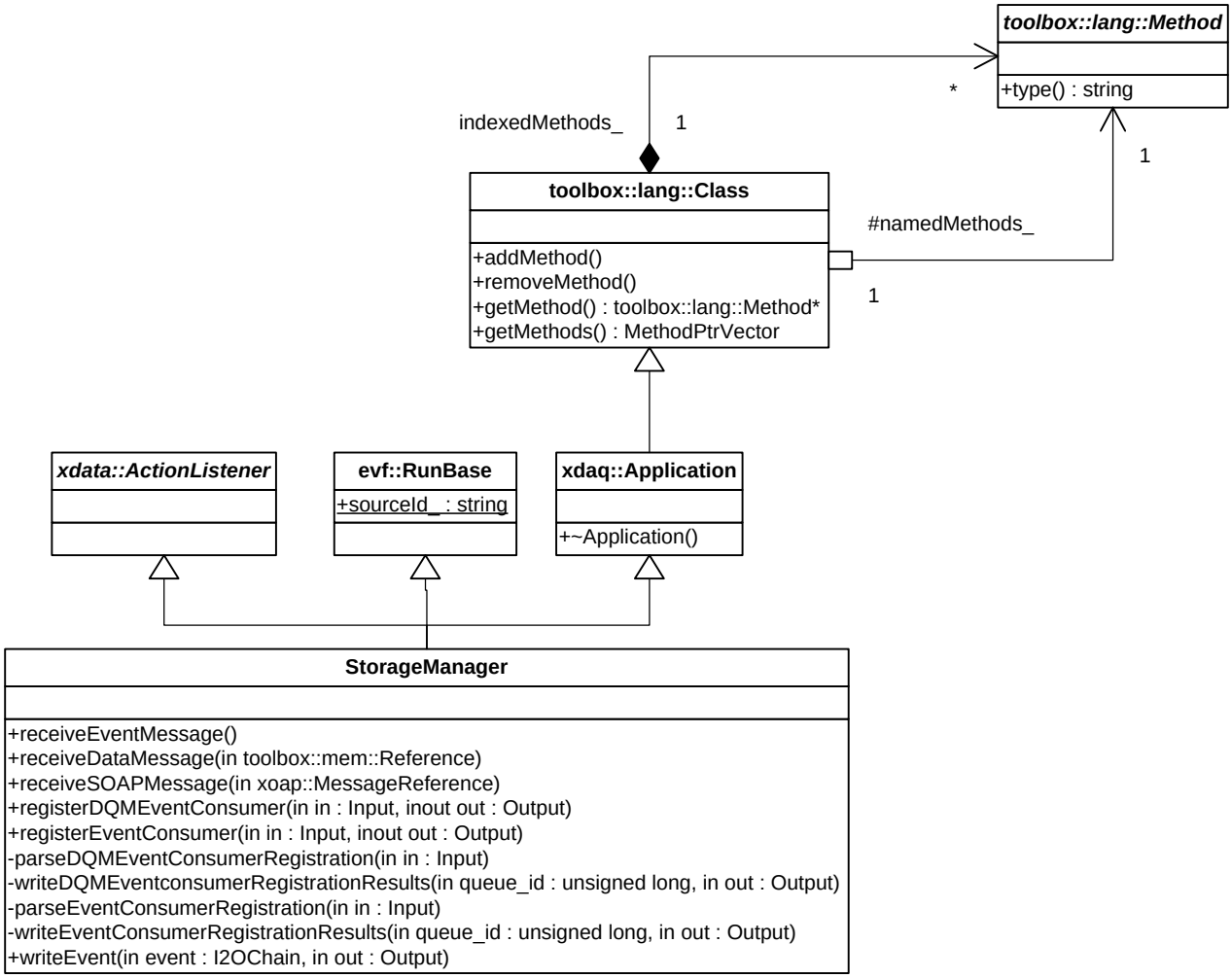


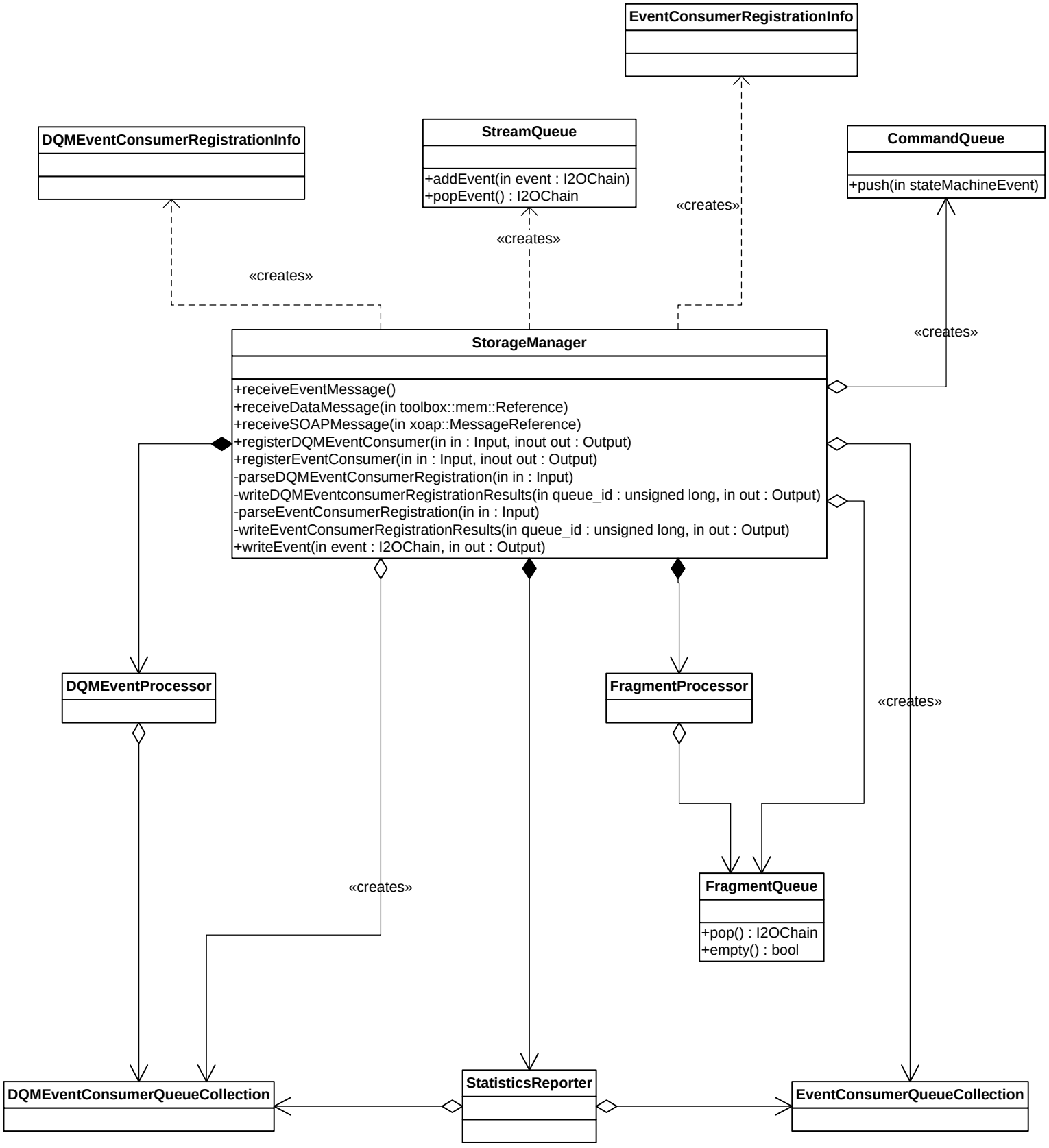
Substates for *Ready* state

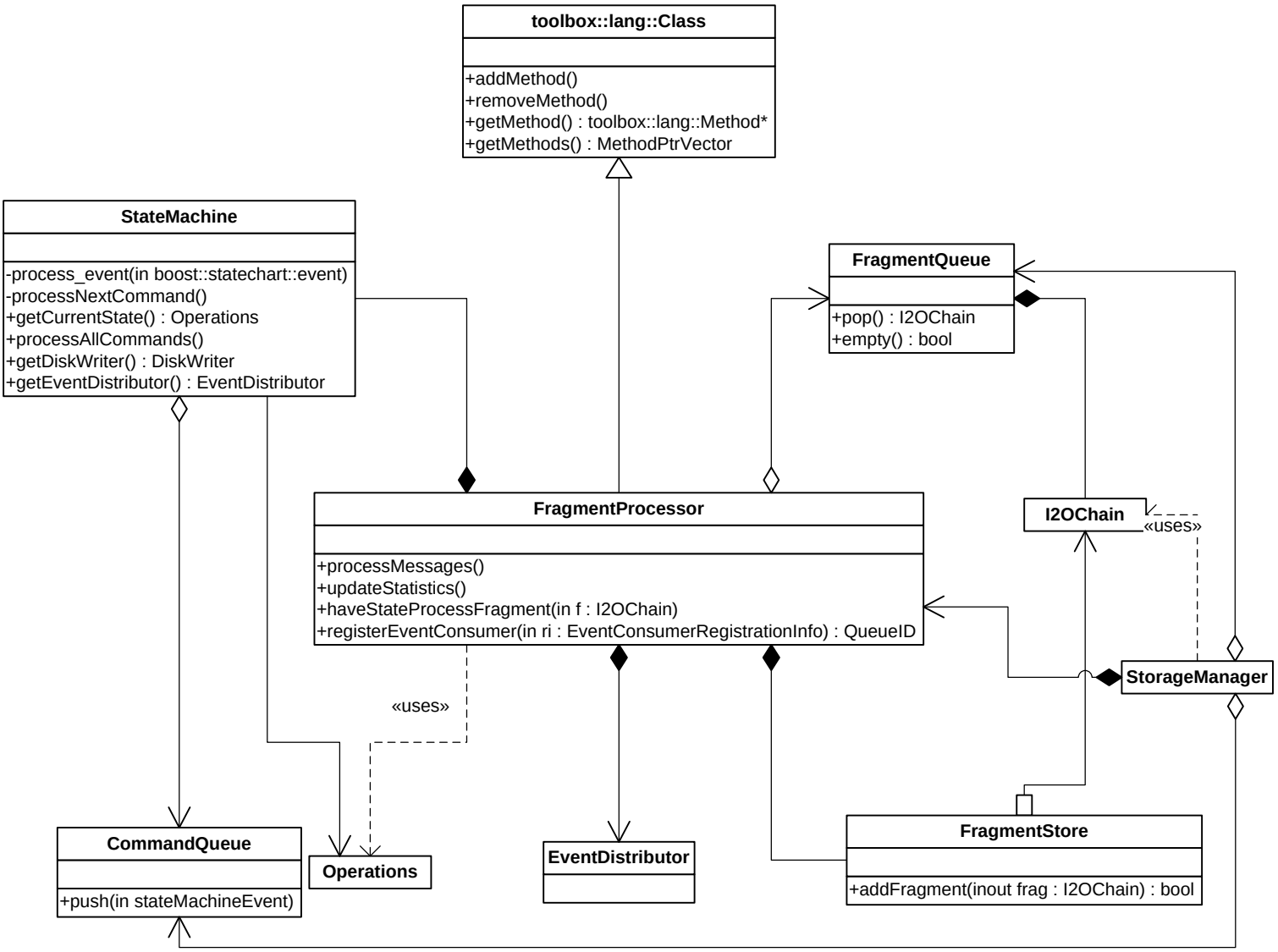


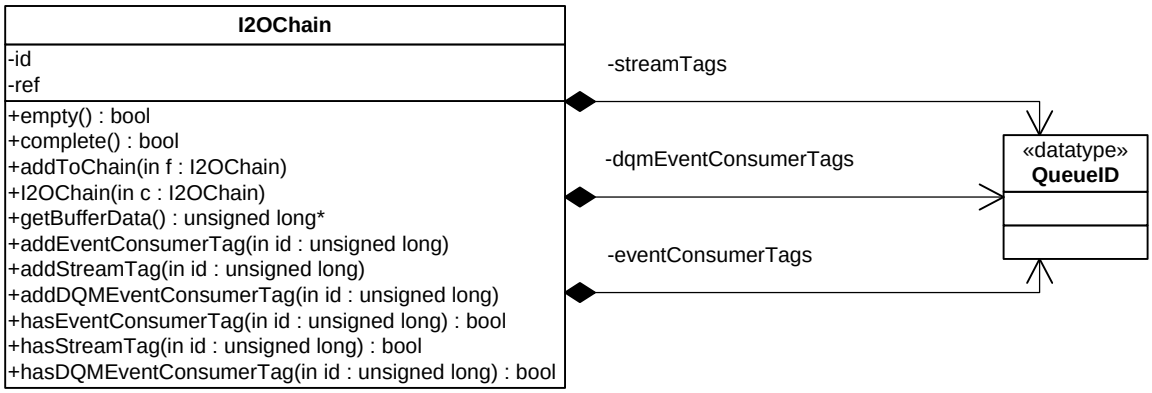
Substates for *Enabled* State

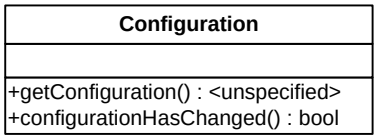


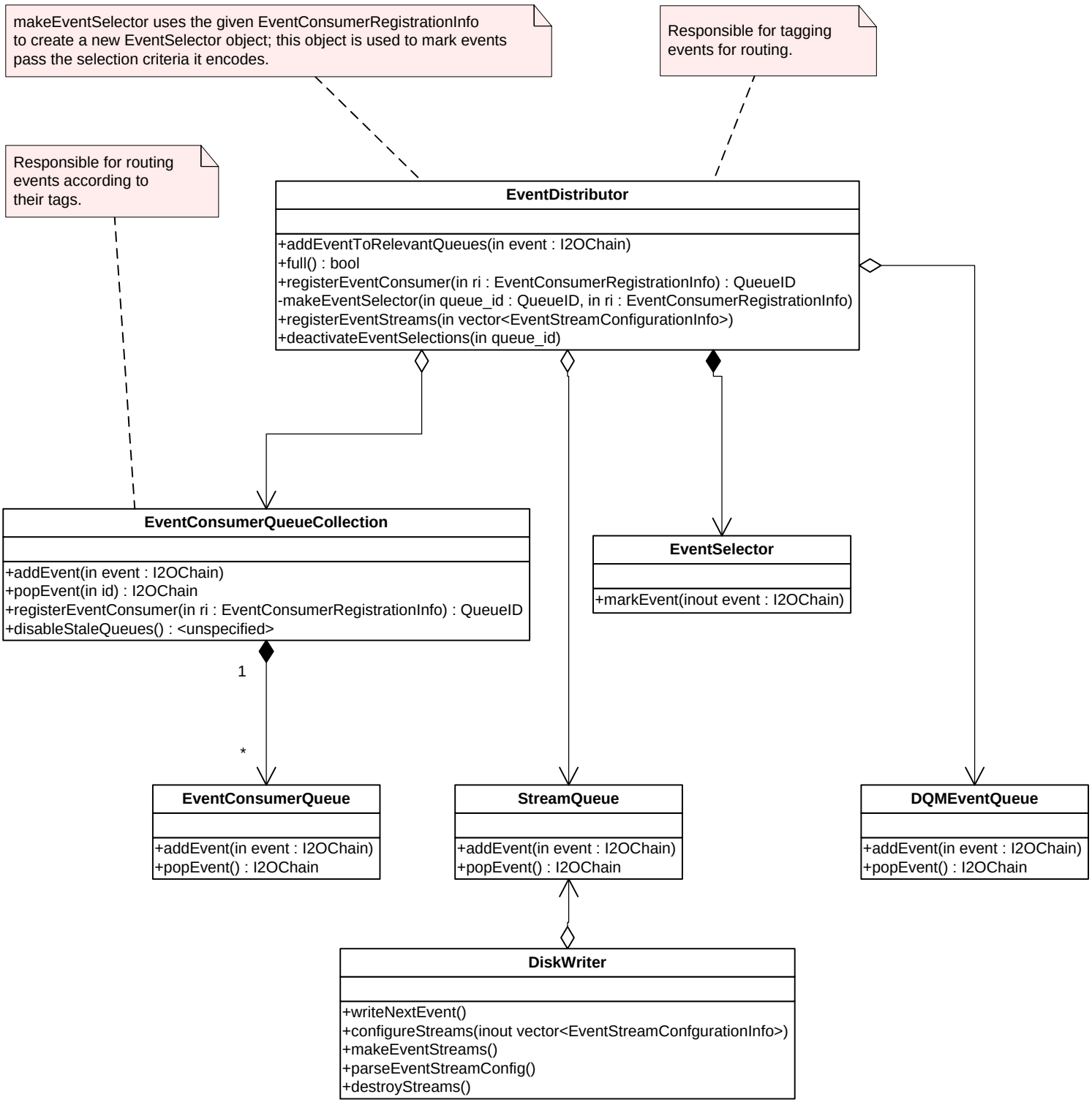

toolbox::lang::Method
+type() : string
*
indexedMethods_
1
1
toolbox::lang::Class
+addMethod()
+removeMethod()
+getMethod() : toolbox::lang::Method*
+getMethods() : MethodPtrVector
#namedMethods_
1
xdata::ActionListener
evf::RunBase
+sourceId_ : string
xdaq::Application
+~Application()
StorageManager
+receiveEventMessage()
+receiveDataMessage(in toolbox::mem::Reference)
+receiveSOAPMessage(in xoap::MessageReference)
+registerDQMEventConsumer(in in : Input, inout out : Output)
+registerEventConsumer(in in : Input, inout out : Output)
-parseDQMEventConsumerRegistration(in in : Input)
-writeDQMEventconsumerRegistrationResults(in queue_id : unsigned long, in out : Output)
-parseEventConsumerRegistration(in in : Input)
-writeEventConsumerRegistrationResults(in queue_id : unsigned long, in out : Output)
+writeEvent(in event : I2OChain, in out : Output)

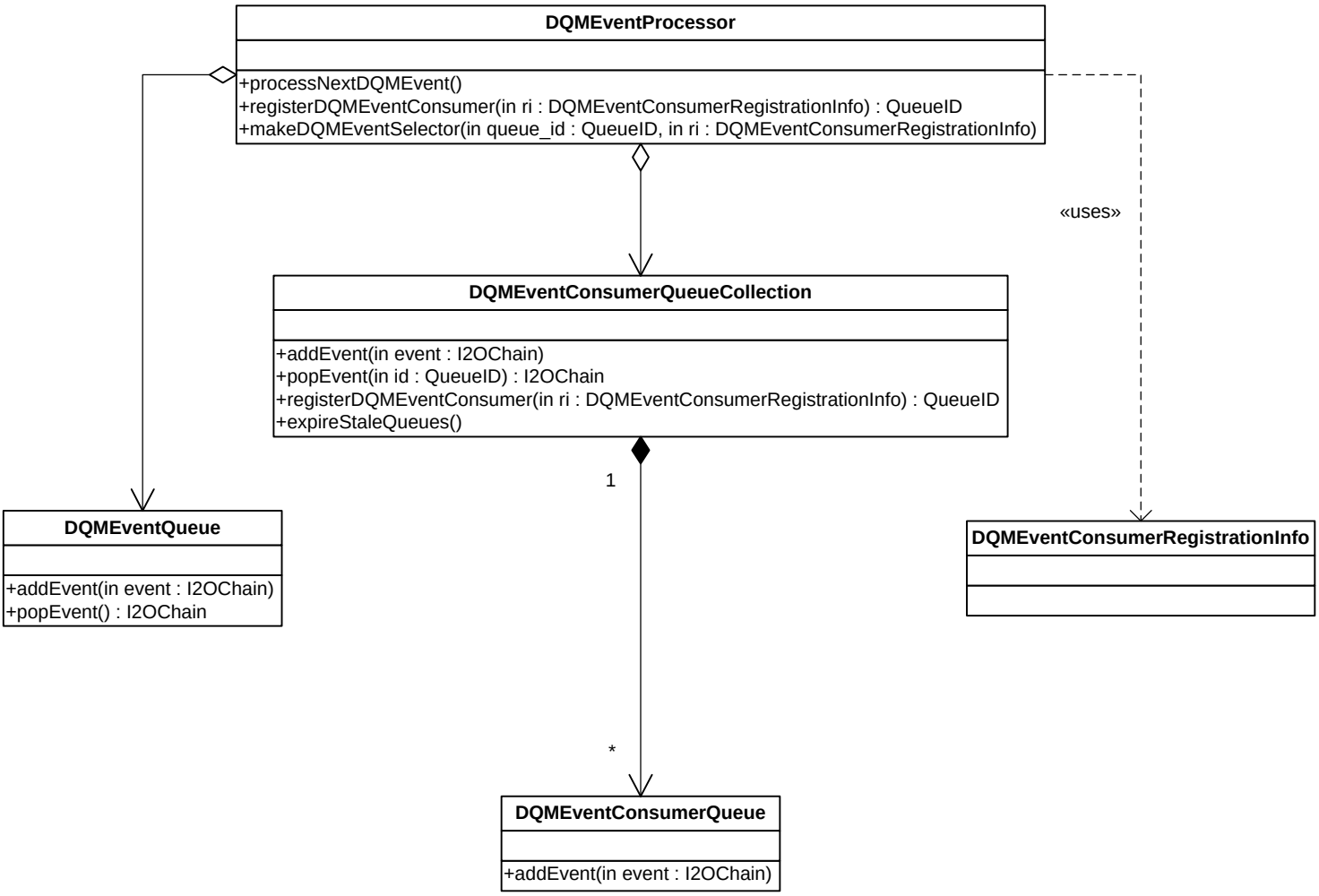

EventConsumerRegistrationInfo
DQMEventConsumerRegistrationInfo
StreamQueue
+addEvent(in event : I2OChain)
+popEvent() : I2OChain
CommandQueue
+push(in stateMachineEvent)
«creates»
«creates»
«creates»
«creates»
StorageManager
+receiveEventMessage()
+receiveDataMessage(in toolbox::mem::Reference)
+receiveSOAPMessage(in xoap::MessageReference)
+registerDQMEventConsumer(in in : Input, inout out : Output)
+registerEventConsumer(in in : Input, inout out : Output)
-parseDQMEventConsumerRegistration(in in : Input)
-writeDQMEventconsumerRegistrationResults(in queue_id : unsigned long, in out : Output)
-parseEventConsumerRegistration(in in : Input)
-writeEventConsumerRegistrationResults(in queue_id : unsigned long, in out : Output)
+writeEvent(in event : I2OChain, in out : Output)
DQMEventProcessor
FragmentProcessor
«creates»
«creates»
FragmentQueue
+pop() : I2OChain
+empty() : bool
DQMEventConsumerQueueCollection
StatisticsReporter
EventConsumerQueueCollection


toolbox::lang::Class
+addMethod()
+removeMethod()
+getMethod() : toolbox::lang::Method*
+getMethods() : MethodPtrVector
StateMachine
-process_event(in boost::statechart::event)
-processNextCommand()
+getCurrentState() : Operations
+processAllCommands()
+getDiskWriter() : DiskWriter
+getEventDistributor() : EventDistributor
FragmentQueue
+pop() : I2OChain
+empty() : bool
FragmentProcessor
+processMessages()
+updateStatistics()
+haveStateProcessFragment(in f : I2OChain)
+registerEventConsumer(in ri : EventConsumerRegistrationInfo) : QueueID
I2OChain
«uses»
StorageManager
«uses»
CommandQueue
+push(in stateMachineEvent)
Operations
EventDistributor
FragmentStore
+addFragment(inout frag : I2OChain) : bool

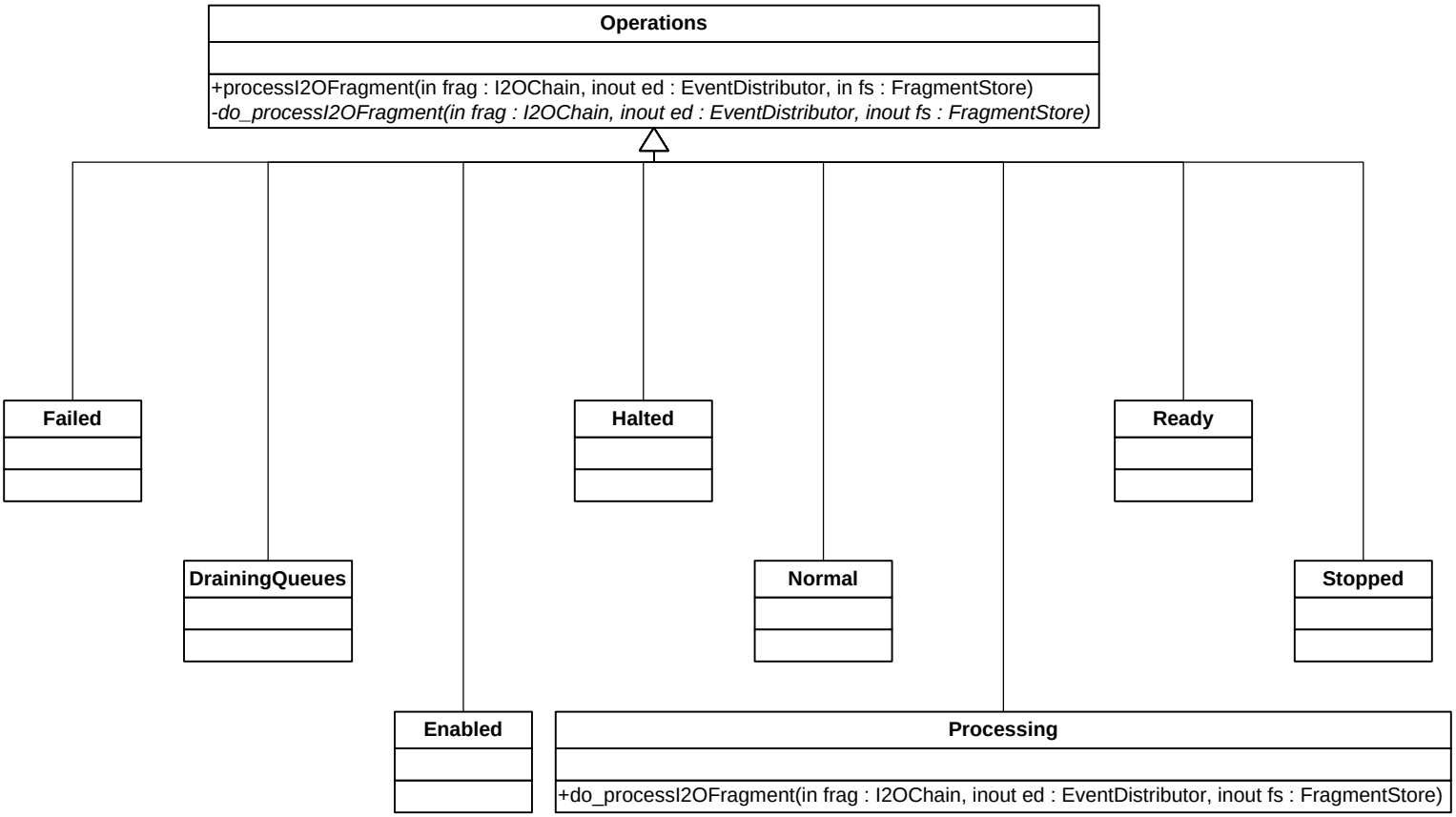

I2OChain
-id
-ref
+empty() : bool
+complete() : bool
+addToChain(in f : I2OChain)
+I2OChain(in c : I2OChain)
+getBufferData() : unsigned long*
+addEventConsumerTag(in id : unsigned long)
+addStreamTag(in id : unsigned long)
+addDQMEventConsumerTag(in id : unsigned long)
+hasEventConsumerTag(in id : unsigned long) : bool
+hasStreamTag(in id : unsigned long) : bool
+hasDQMEventConsumerTag(in id : unsigned long) : bool
-streamTags
-dqmEventConsumerTags
-eventConsumerTags
«datatype»
QueueID

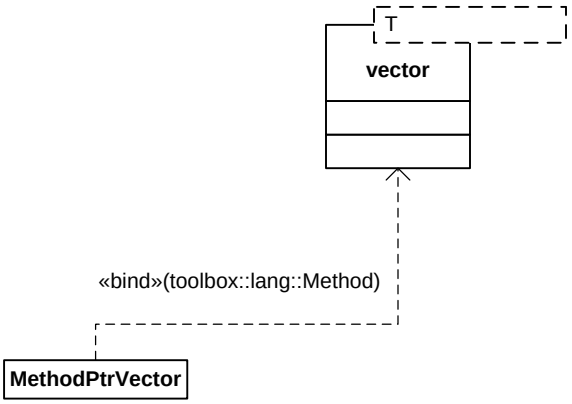

Configuration
+getConfiguration() : <unspecified>
+configurationHasChanged() : bool


makeEventSelector uses the given EventConsumerRegistrationInfo to create a new EventSelector object; this object is used to mark events pass the selection criteria it encodes.
Responsible for tagging events for routing.
Responsible for routing events according to their tags.
EventDistributor
+addEventToRelevantQueues(in event : I2OChain)
+full() : bool
+registerEventConsumer(in ri : EventConsumerRegistrationInfo) : QueueID
-makeEventSelector(in queue_id : QueueID, in ri : EventConsumerRegistrationInfo)
+registerEventStreams(in vector<EventStreamConfigurationInfo>)
+deactivateEventSelections(in queue_id)
EventConsumerQueueCollection
+addEvent(in event : I2OChain)
+popEvent(in id) : I2OChain
+registerEventConsumer(in ri : EventConsumerRegistrationInfo) : QueueID
+disableStaleQueues() : <unspecified>
EventSelector
+markEvent(inout event : I2OChain)
1
*
EventConsumerQueue
+addEvent(in event : I2OChain)
+popEvent() : I2OChain
StreamQueue
+addEvent(in event : I2OChain)
+popEvent() : I2OChain
DQMEventQueue
+addEvent(in event : I2OChain)
+popEvent() : I2OChain
DiskWriter
+writeNextEvent()
+configureStreams(inout vector<EventStreamConfgurationInfo>)
+makeEventStreams()
+parseEventStreamConfig()
+destroyStreams()


DQMEventProcessor
+processNextDQMEvent()
+registerDQMEventConsumer(in ri : DQMEventConsumerRegistrationInfo) : QueueID
+makeDQMEventSelector(in queue_id : QueueID, in ri : DQMEventConsumerRegistrationInfo)
«uses»
DQMEventConsumerQueueCollection
+addEvent(in event : I2OChain)
+popEvent(in id : QueueID) : I2OChain
+registerDQMEventConsumer(in ri : DQMEventConsumerRegistrationInfo) : QueueID
+expireStaleQueues()
1
DQMEventQueue
+addEvent(in event : I2OChain)
+popEvent() : I2OChain
DQMEventConsumerRegistrationInfo
*
DQMEventConsumerQueue
+addEvent(in event : I2OChain)

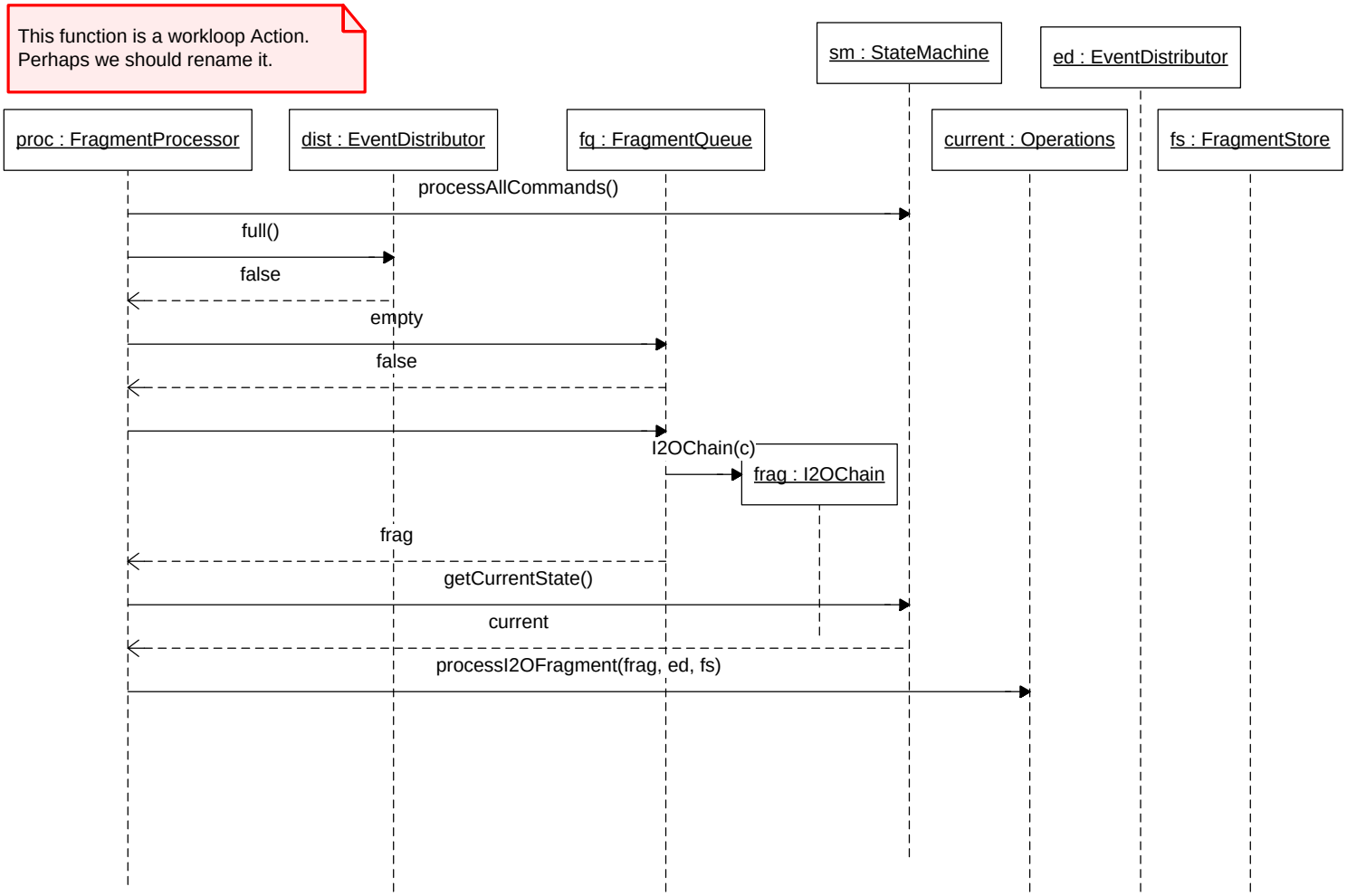
State classes show only the inheritance from Operations, not the inheritance from the Boost Statechart classes.



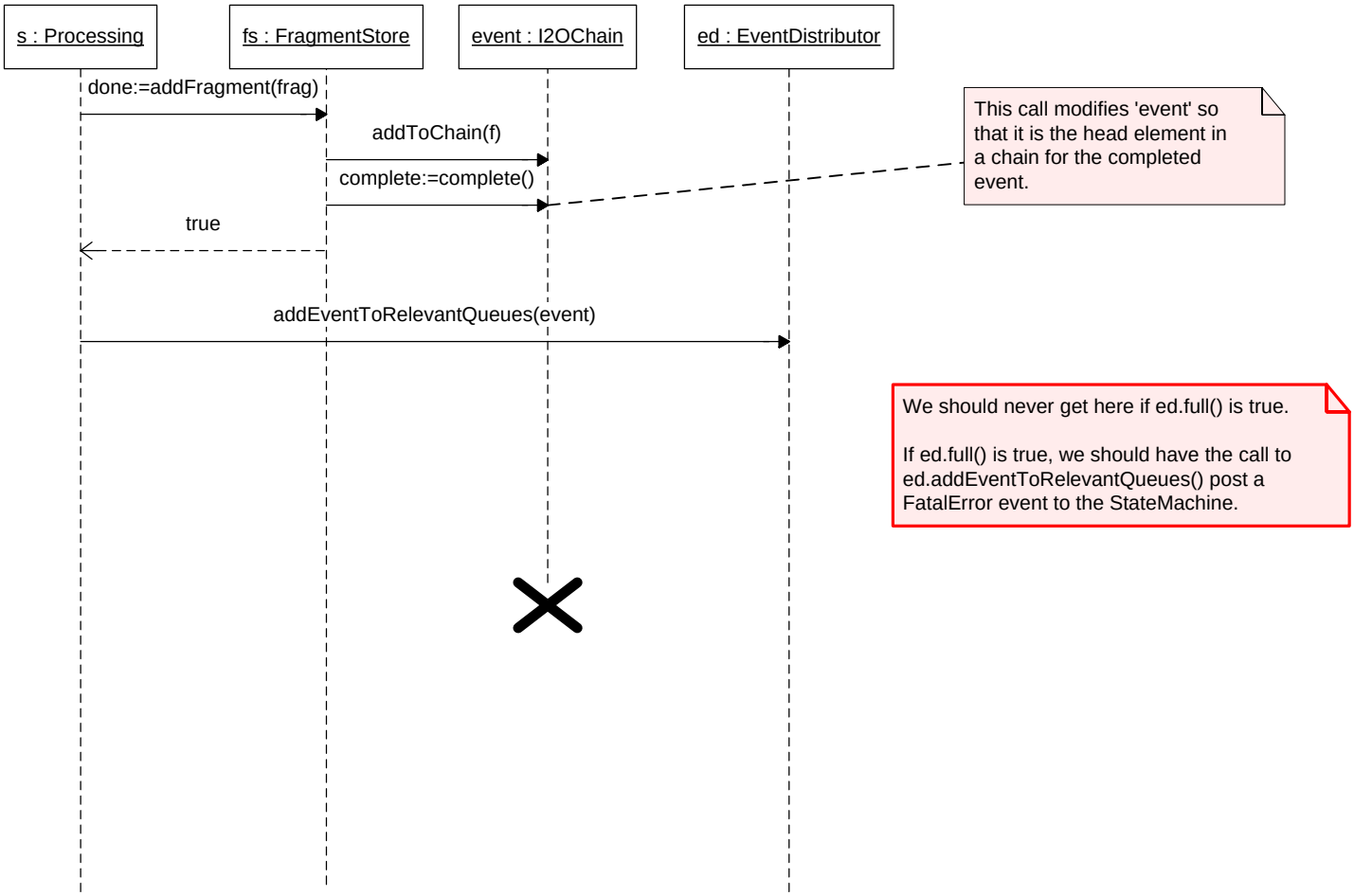

T
vector
«bind»(toolbox::lang::Method)
MethodPtrVector


Input
Output

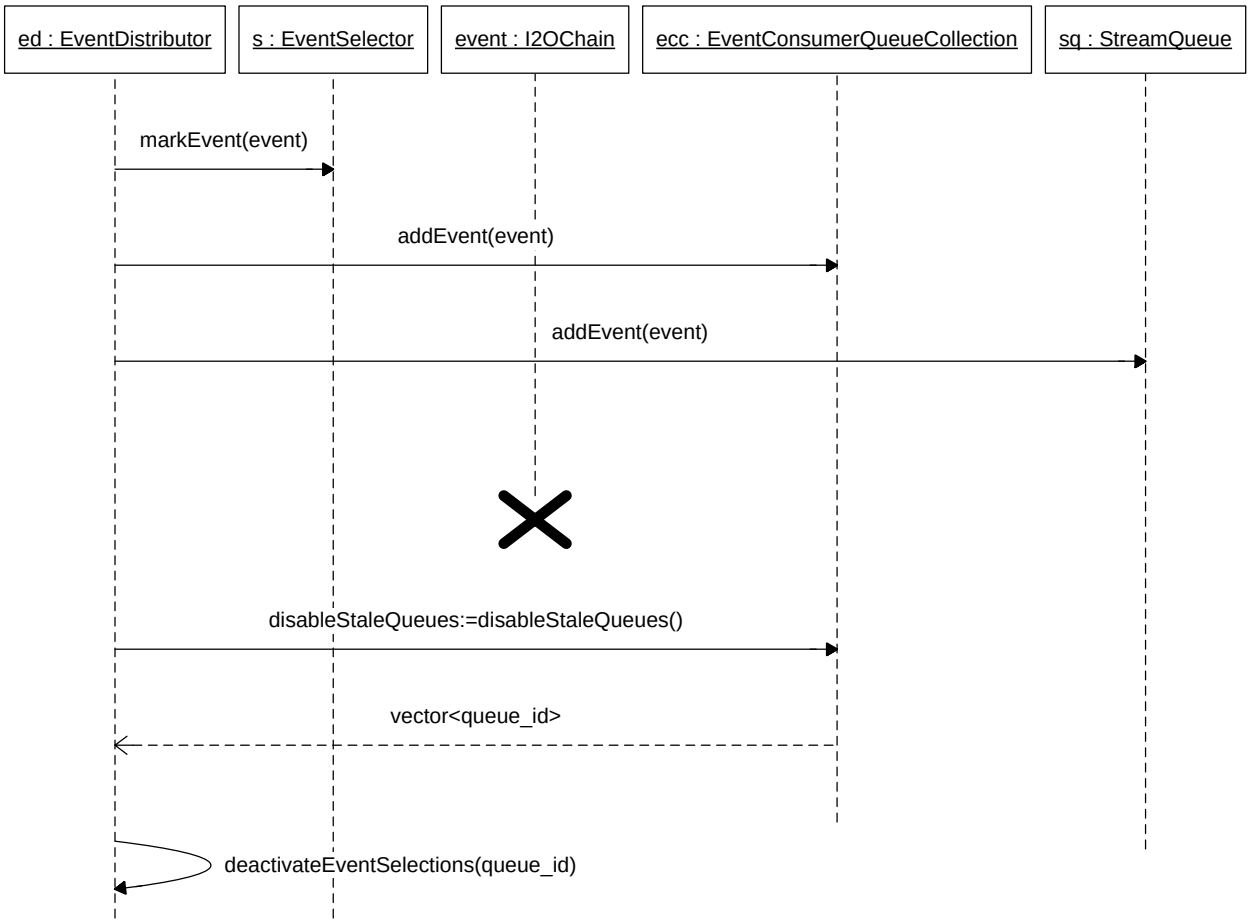
FragmentProcessor::processMessages



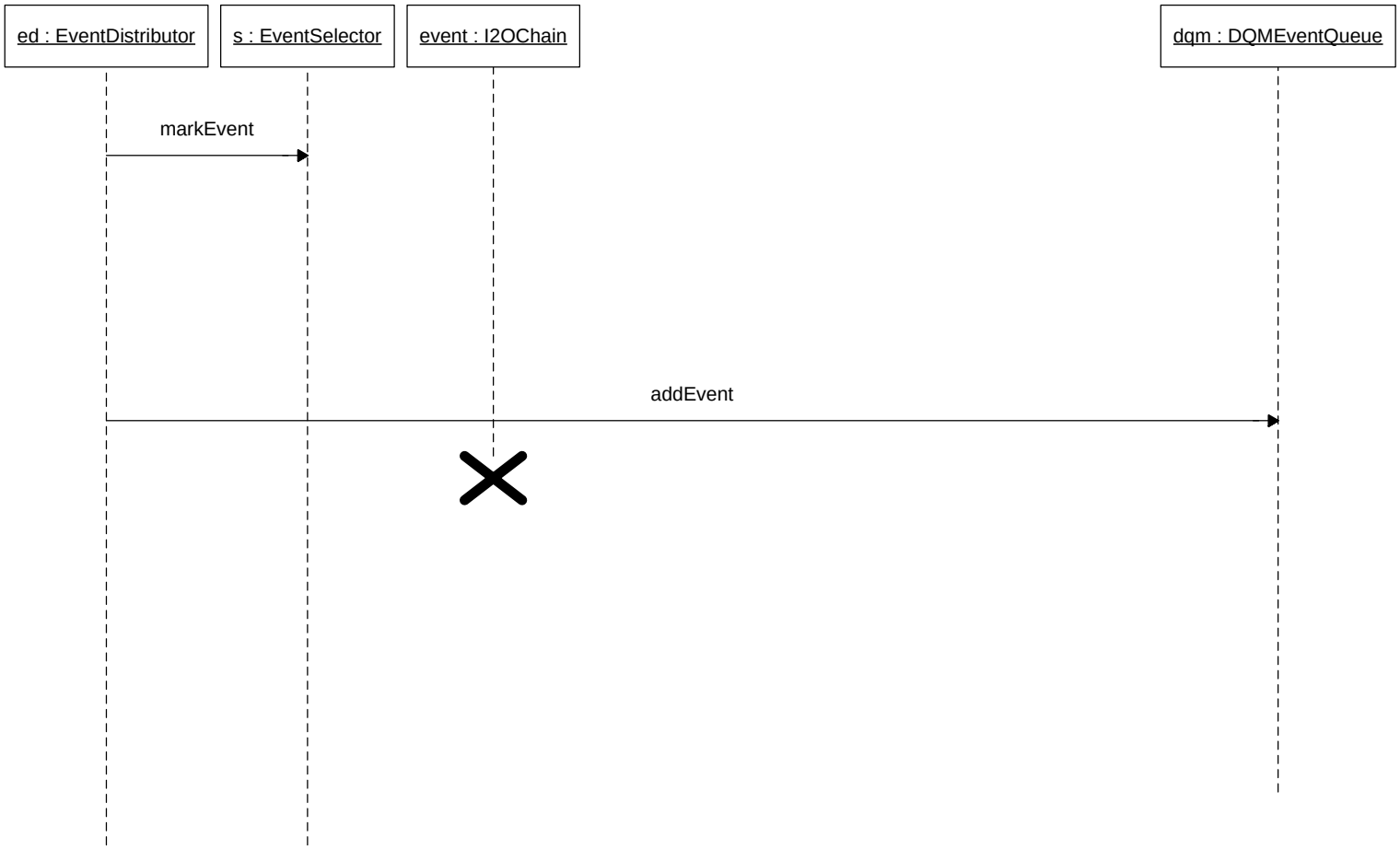
Processing State processing a fragment: last fragment case (do_processI2OFragment)



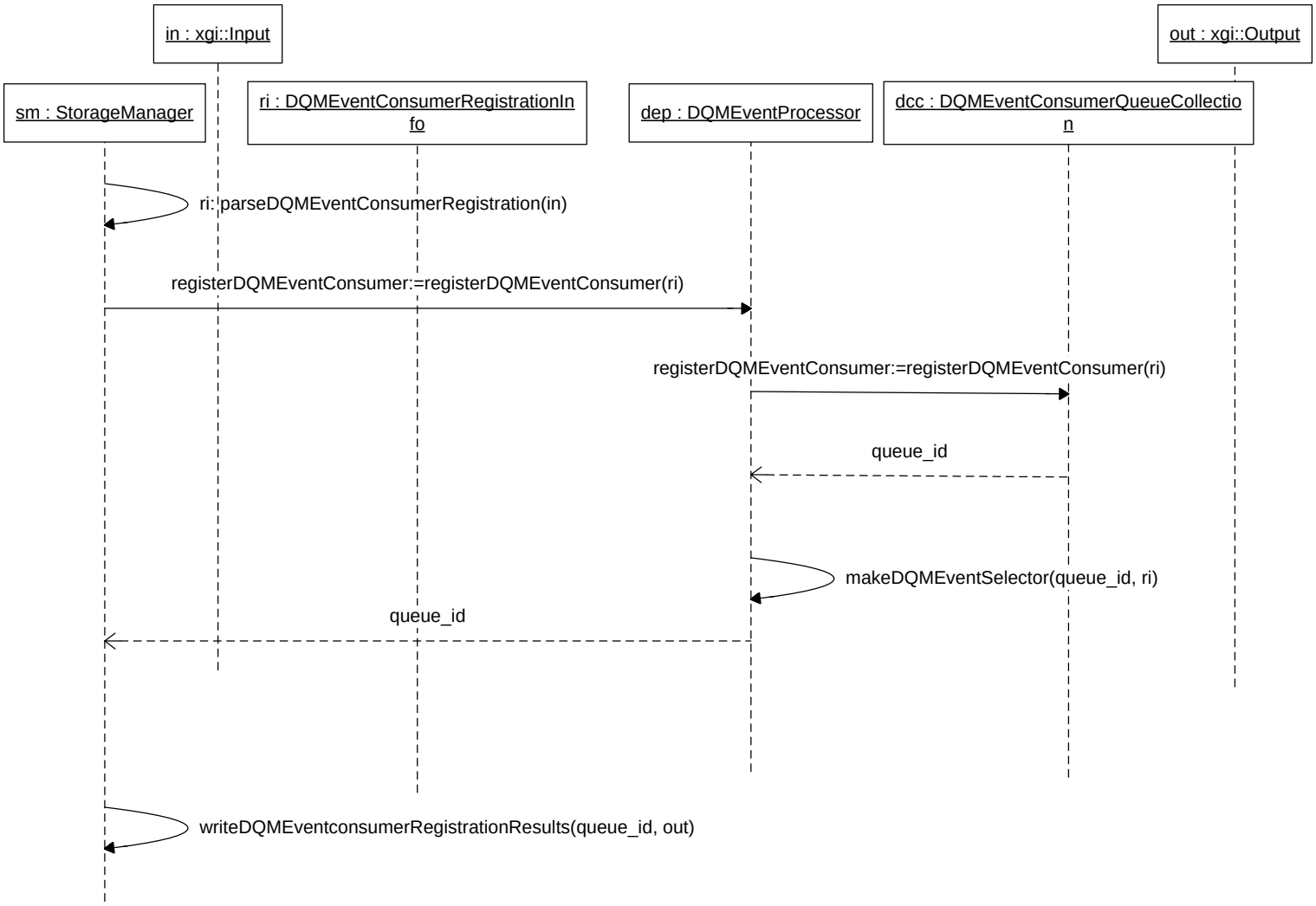
EventDistributor::addEventToRelevantQueues (collision event)



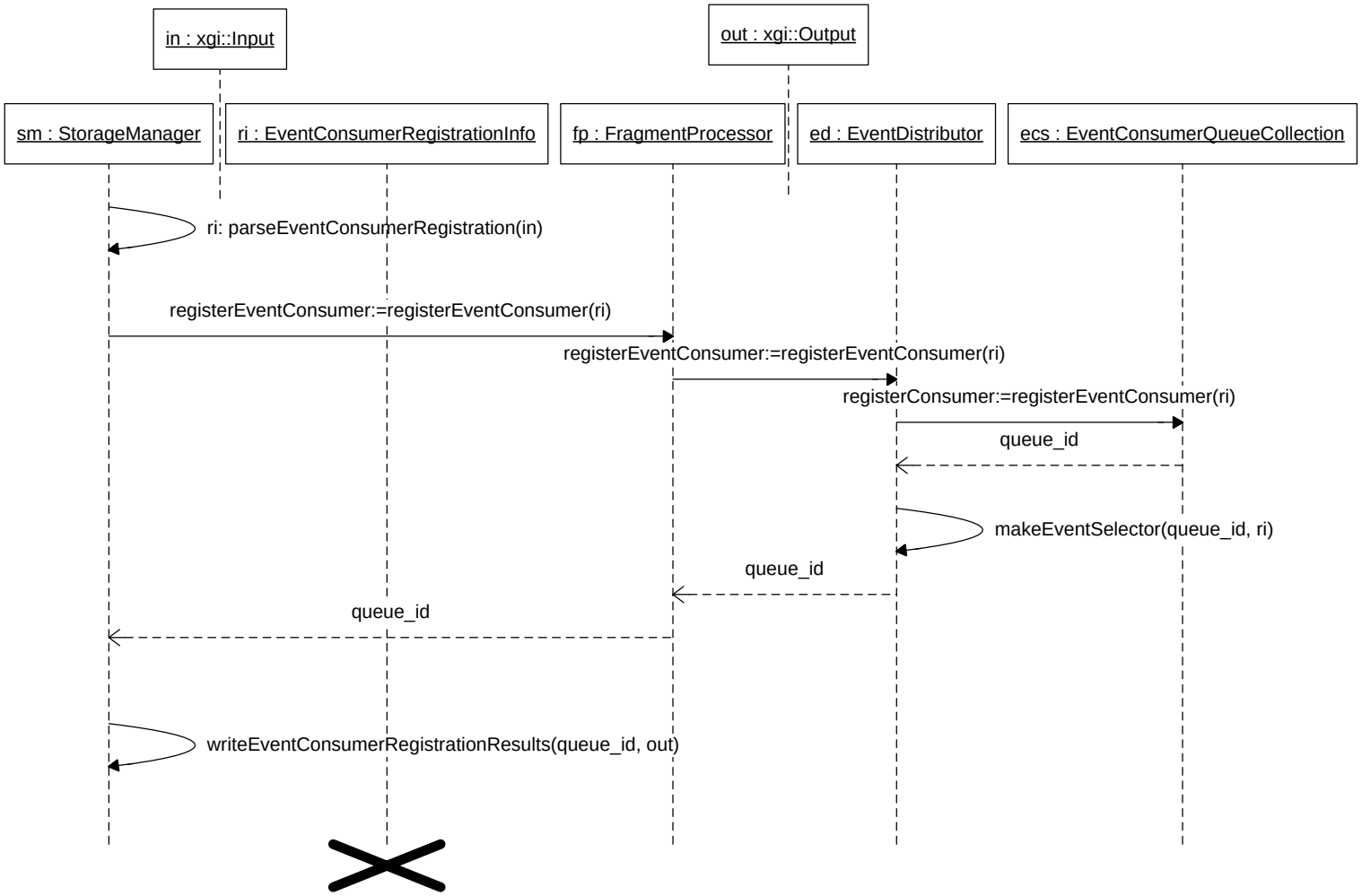
EventDistributor::addEventToRelevantQueues (DQM event)



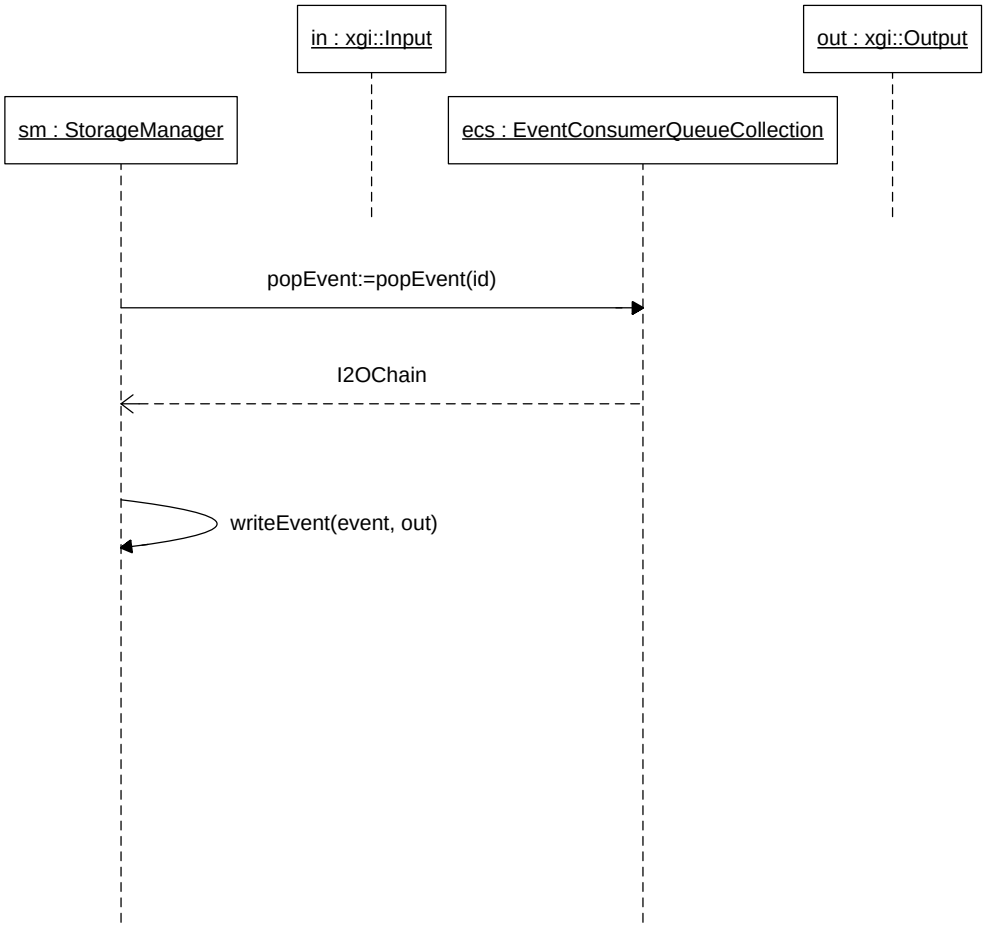
DQMEvent consumer registration (StorageManager::registerDQMEventConsumer)



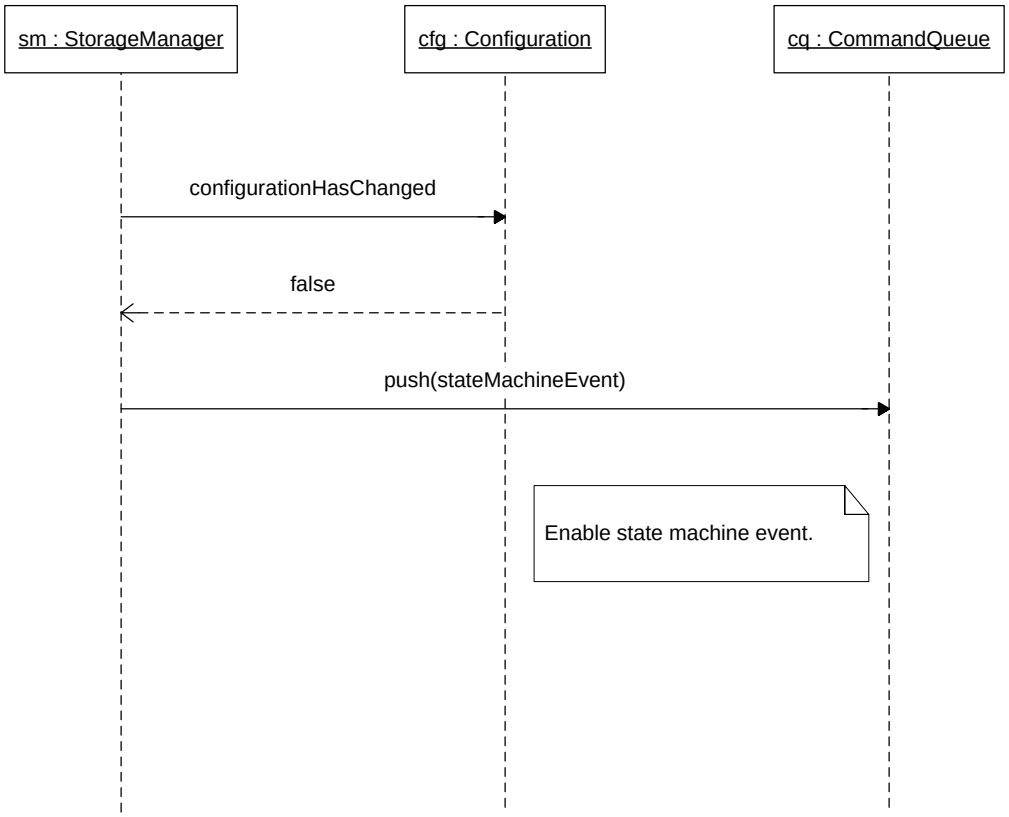
Event consumer registration (StorageManager::registerEventConsumer)



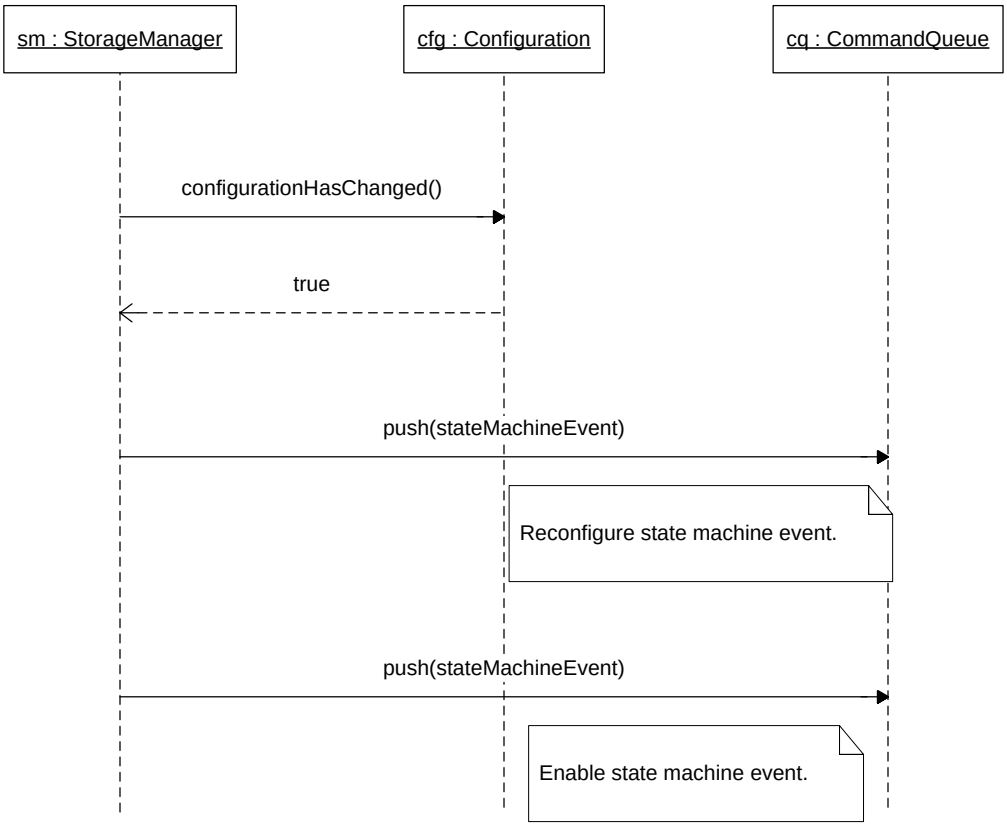
Event consumer event request (StorageManager::eventdataWebPage)



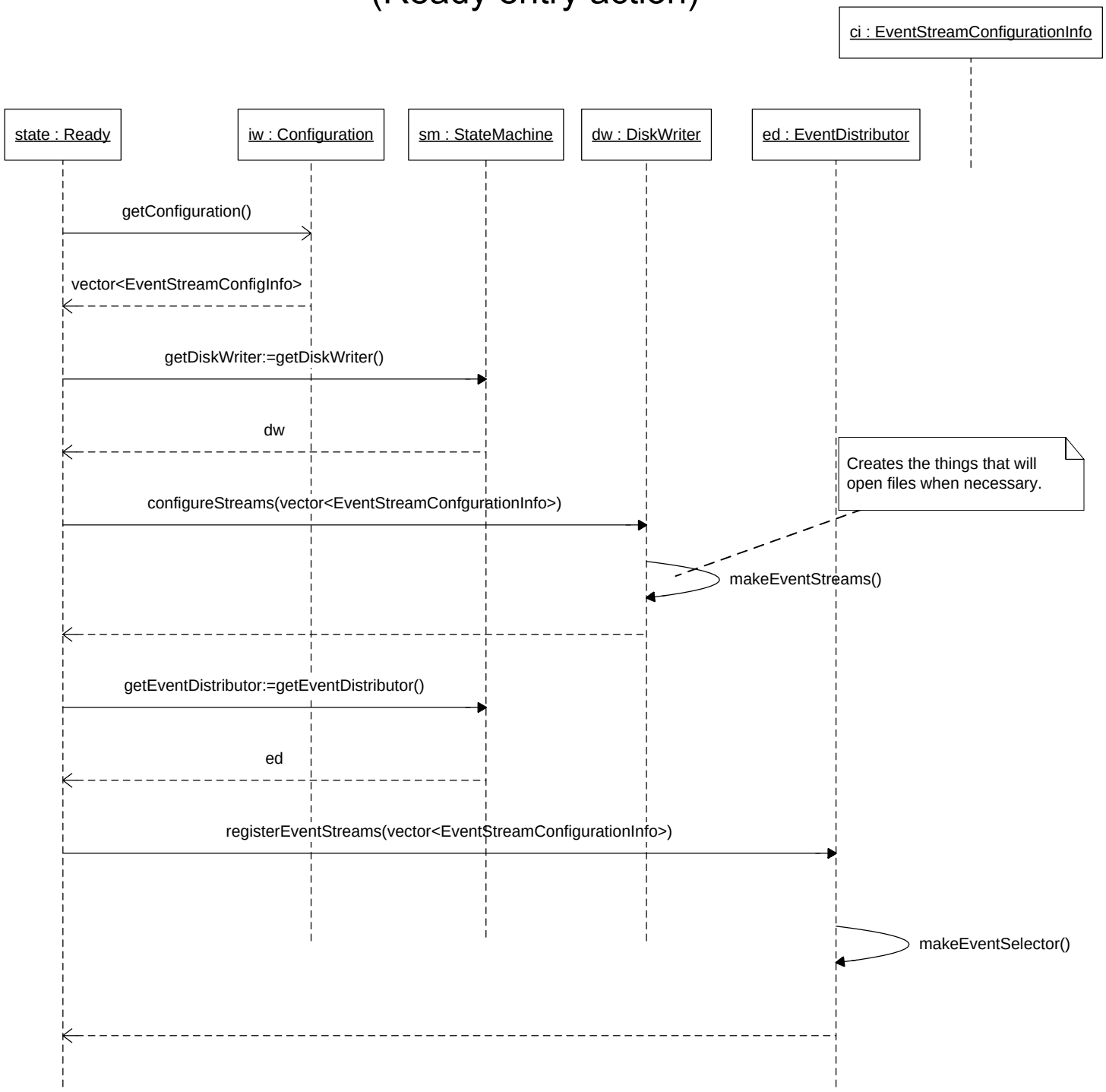
SM callback for Enable SOAP command (reconfiguration not needed)



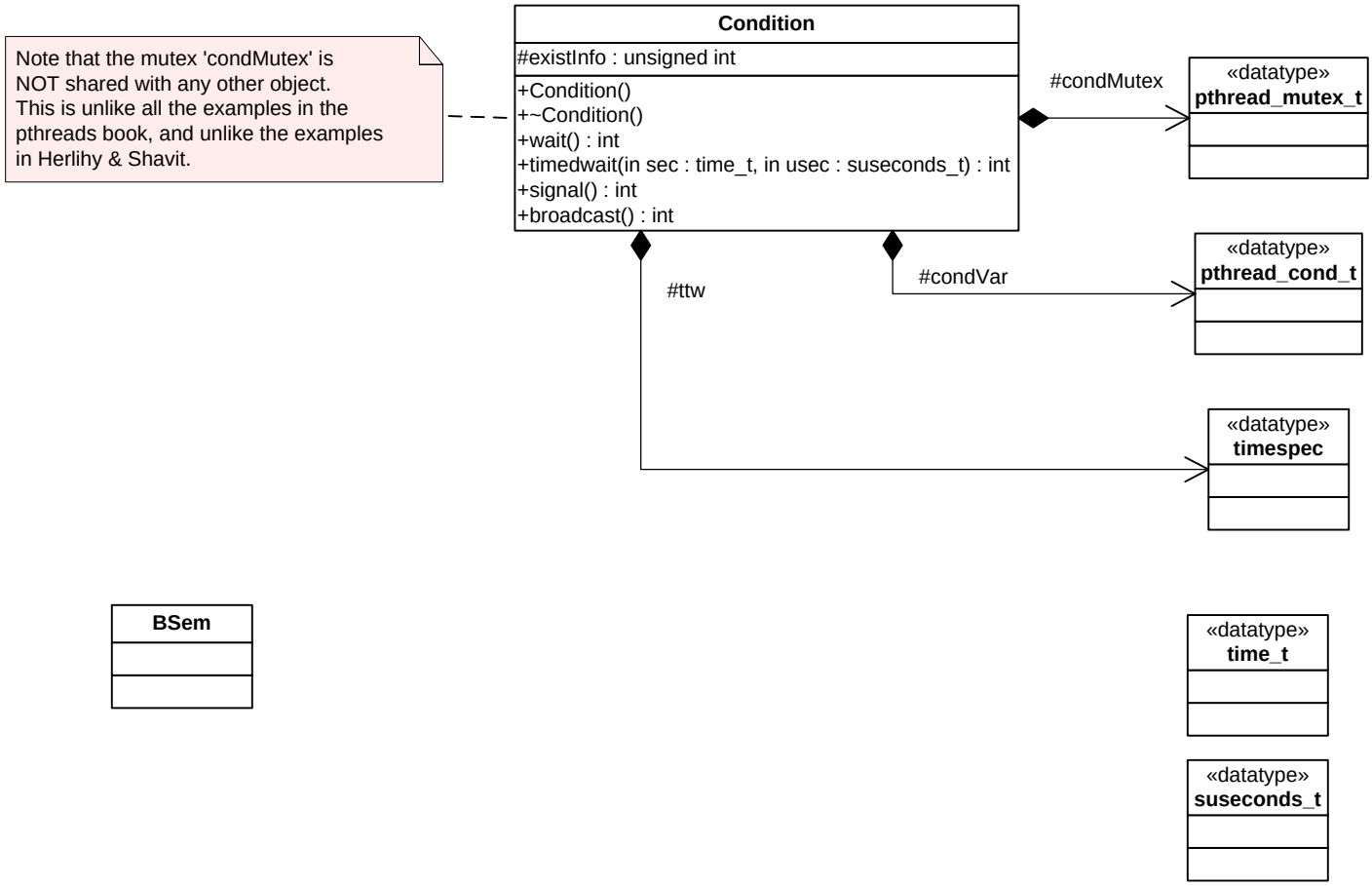
SM callback for Enable SOAP command (reconfiguration needed)

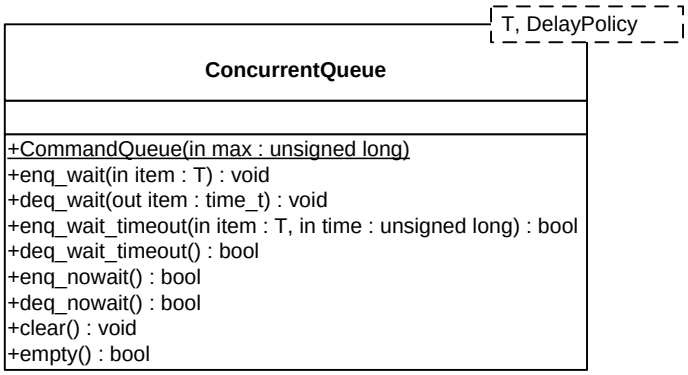


Event stream configuration
(Ready entry action)



Toolbox classes for XDAQ: Linux implementations




T, DelayPolicy
ConcurrentQueue
+CommandQueue(in max : unsigned long)
+enq_wait(in item : T) : void
+deq_wait(out item : time_t) : void
+enq_wait_timeout(in item : T, in time : unsigned long) : bool
+deq_wait_timeout() : bool
+enq_nowait() : bool
+deq_nowait() : bool
+clear() : void
+empty() : bool