

# CME STP - Message Flows

This page describes the sequence and flow of information (showing key FIXML fields) between the User and the CME STP.

- [Subscription](#)
- [Query](#)
- [Change of Firm](#)
- [Spreads](#)
- [Allocation](#)
  - [Allocation States](#)
  - [Allocation of Spreads](#)



CME STP often sends multiple Trade Capture Report messages per subscription or query response. These examples provide only Trade Capture Report messages from the API.

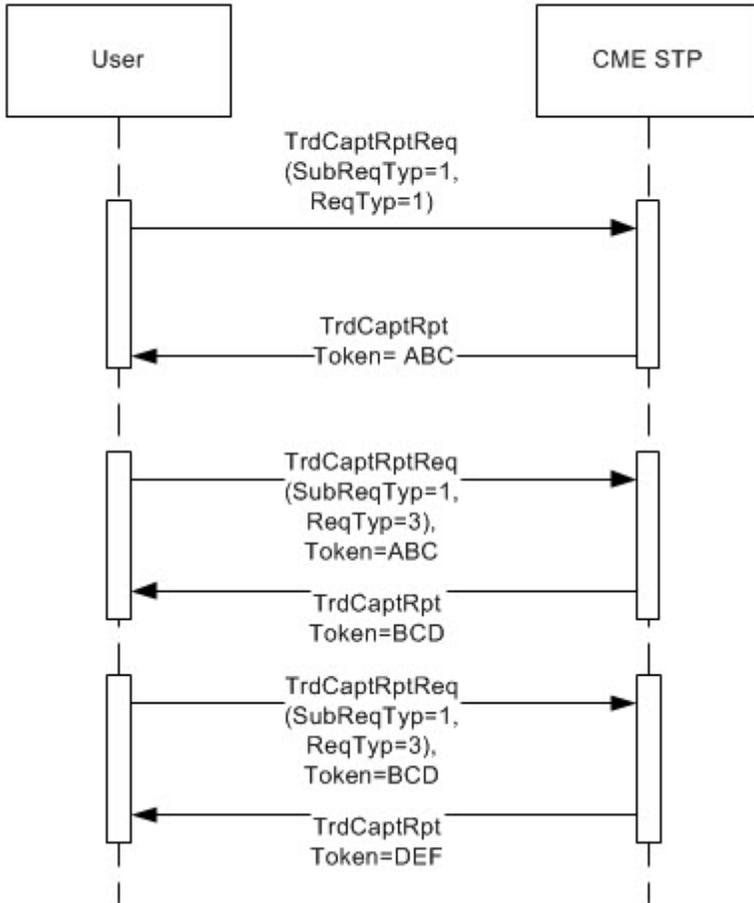
For a full list of FIXML fields that are returned by the API, please see the [FIXML Message Specification](#). See [Message Samples](#) for corresponding message layouts examples.

## Subscription

This flow shows the User requesting a subscription to the CME STP and the API response.

- After sending a subscription request to the API with `@ReqTyp = 1`, the User receives a token along with zero or more Trade Capture Report messages inside a FIXML batch, e.g. `/FIXML/Batch/TrdCaptRpt`
- The User must return the token with the next Trade Capture Report Request to continue to receive trade information without duplication or loss, along with `@ReqTyp = 3`.
  - If the token is not sent back, the API considers this an error, and responds with the Trade Capture Report Request Ack message.
  - A new token is issued for each subscription or query request.
- Upon receipt of the previous response, submit a new subscription response no sooner than 3 seconds later.
- If no new data is available since the User's last query, the API will send the User an empty batch with a token.

## Subscription



## Query

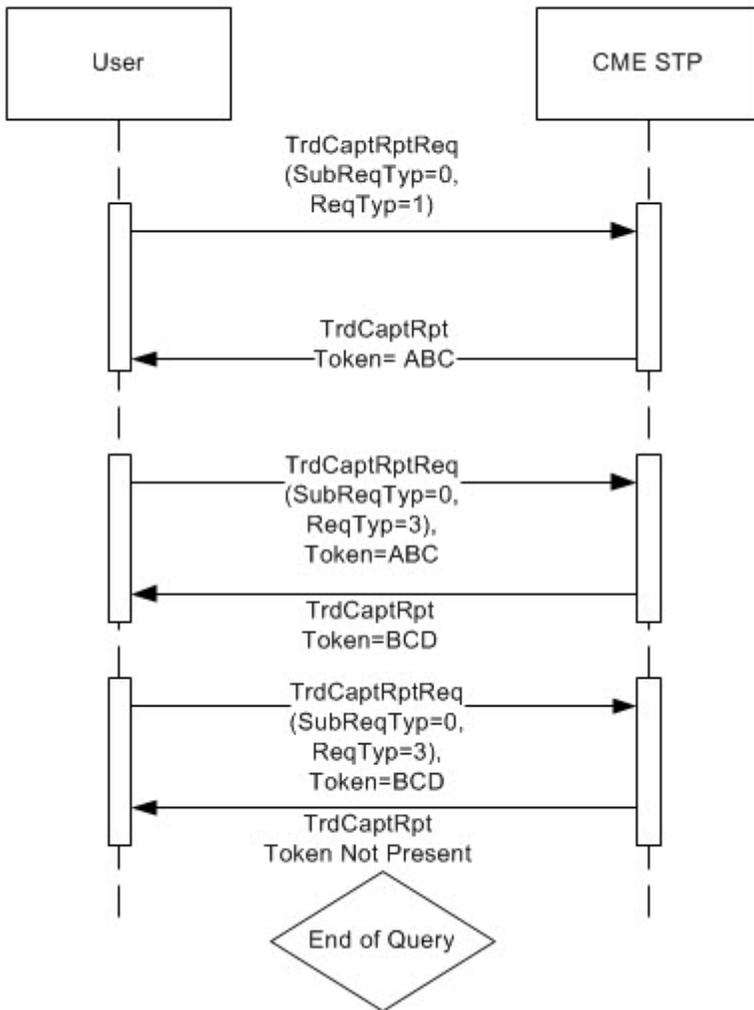
This flow shows the User querying CME STP and the API response.

After sending a query to the API with `@ReqTyp = 1`, the User receives zero or more Trade Capture Report messages inside a FIXML batch, e.g. `/FIXML L/Batch/TrdCaptRpt`, as follows:

- If no trades match the query parameters, the API sends an empty FIXML batch and no token.
- If all trades that match the query parameters can be sent inside one FIXML batch, the API will do so, and will not send a token.
- If more trades match the query parameters than can be sent inside one FIXML batch, the API sends some of the messages inside a FIXML batch, and includes a token for continuing the query.

In the latter case, to continue a query, the User must return the token with the next Trade Capture Report Request, along with `@ReqTyp = 3`. If the token is not sent back, the API considers this an error, and responds with the Trade Capture Report Request Ack Message. The User must send the last received token, as this enables the API to send the query results without duplication or loss. A follow up query may be performed immediately, with no need to wait 3 seconds. This process can continue, with the API sending a token and the User returning the token on the next Trade Capture Report Request, along with `@ReqTyp = 3`, until no more trades are available that match the query parameters. When the API sends the last batch of trades, it will not send a token. The absence of a token indicates that the results are complete, and the User should not continue sending Trade Capture Report Request messages for this query.

## Query



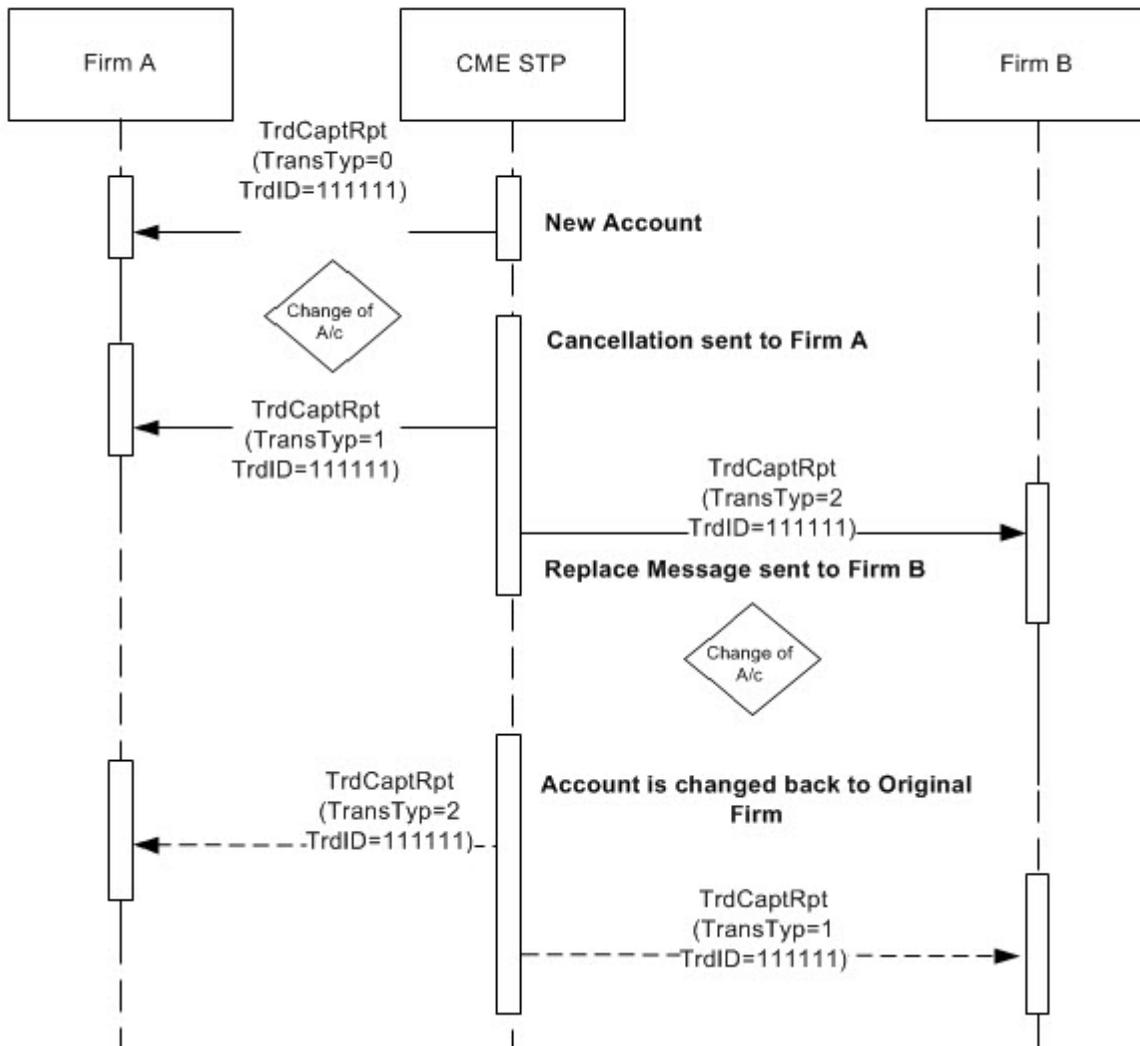
### Change of Firm

This flow shows the messages sent when a clearing firm changes the account for a trade, causing the trade to transition from Firm A to B. Firm B receives the trade replaced message (@TransTyp = 2) even though Firm B never saw the new trade (@TransTyp = 0). Firms are not guaranteed to see the trade as new (@TransTyp = 0) and must be able to process trades first encountered as replaced (@TransTyp = 2).

This example also illustrates a second transition from Firm B back to Firm A. The identity (@TrdID and @TrdID2) of the trade does not change. Note that Firm A receives a trade cancelled message (@TransTyp = 1) followed later by a trade replaced message (@TransTyp = 2). Firms must be able to process this sequence of events.

 This diagram shows only responses, and not requests.

## Change of Firm

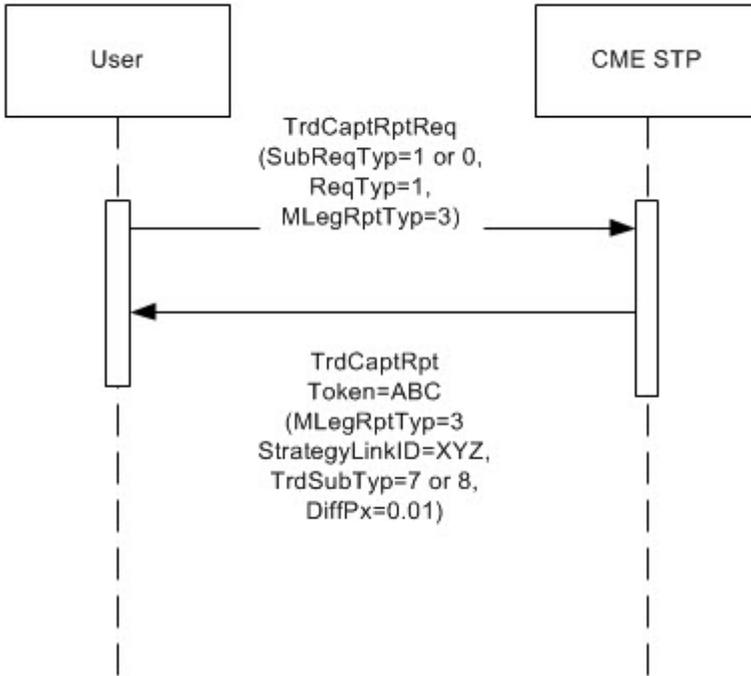


## Spreads

The following 2 flows show the different types of spread level information that a User can request and consume.

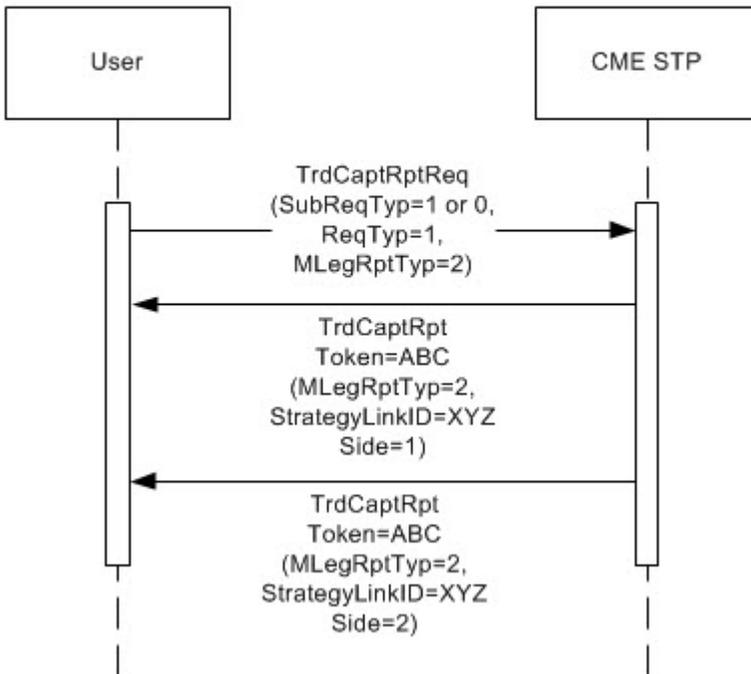
The first flow shows the User requesting Spread level trades (@MLegRptTyp = 3). The API then responds with Outrights (@MLegRptTyp = 1) as well as Spread level messages (@MLegRptTyp = 3). @TrdSubTyp indicates whether the message is a Differential (7) or Spread (8). @DiffPx carries the Spread or Differential price (e.g. 0.01). This trade has multiple legs, each identified in by the /FIXML/Batch/TrdCaptRpt/TrdLeg component. The full list of TrdLeg FIXML fields returned by the API, is available in the [FIXML Message Specification](#). The field @StrategyLinkID can be used to link legs of a strategy with the multi-leg spread messages. Although in this example, legs are not sent, they can be sent in the case of [allocations of spreads](#).

## Spread level



The next flow show the User requesting Leg level trades (`@MLegRptTyp = 2`). The API then responds with Outrights (`@MLegRptTyp = 1`) as well as Leg level messages (`@MLegRptTyp = 2`). The trades do not have multiple legs, therefore `/FIXML/Batch/TrdCaptRpt/TrdLeg` is not present. The field `@StrategyLinkID` can be used to link legs of a strategy together.

## Leg level



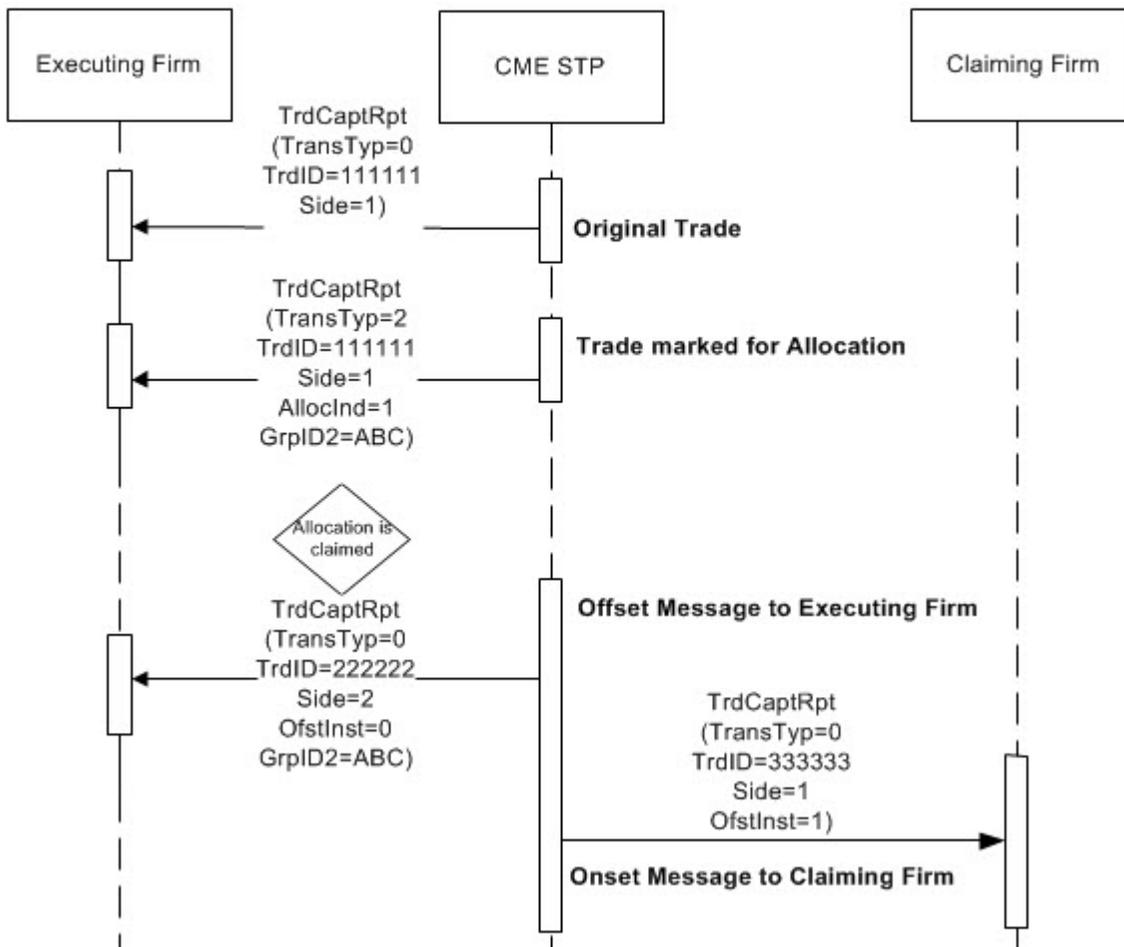
## Allocation

CME STP supports allocations, which means one firm "gives-up" trades to another firm. The firm that executed a trade (i.e. Executing firm) may want to "allocate" or "give-up" a trade to a different firm (i.e. Claiming firm).

The Executing firm will receive an Offset message to indicate that the trade is being removed and the Claiming Firm will receive an Onset message to indicate the trade is now on their books. This flow shows the original trade followed by the modified trade, and the offset and onset messages that the API sends for an allocation.

 This diagram shows only responses, and not requests.

### Allocation



In the case of a reversal or release of an allocation, CME STP will report both the offset and onset trades as cancelled. (@TransTyp = 1) Parameters on the cancel will mirror those of the original offset or onset, e.g. the side of the cancel message will match the side of the respective offset or onset message.

Sub-allocation (e.g. the claiming firm marks a claimed allocation for give-up) will not result in the "Trade marked for Allocation" message above, however, when the new claiming firm claims the allocation, CME STP will send offset and onset trades.

### Allocation States



GUS	Claimed Allocation: Reverse Requested and Accepted	Reversed	Y	<p>Given Allocation has been claimed</p> <p>When the Claiming firm allocation is reversed the following day, then NO TCR is reported</p> <p>When the Buy Side of Trade accepts the reversal</p> <p>Then the Buy Side Receives TCR which is updated OFFSET with Transtyp= 1 (cancel)</p> <p>Then the Claimed Side Receives TCR which is updated ONSET with Transtyp= 1 (cancel)</p>
GUS	Claimed Allocation: Reversal Requested but is Rejected or Cancelled	Claimed	N	
<b>APS</b>				
APS	Create APS Group (marking for APS allocation)	Incomplete Group	Y	<p>Given Trade entered in CPC</p> <p>When The Buy Side marks trades for APS and Assigns group</p> <p>Then The Buy Side Receives TCR which is a Replace (TransTyp=2) with AllocInd=1, GrpID2 and AvgPxInd=1</p>
APS	Complete Average Price Group	Complete Group	N	
APS	Allocate Trade to Claiming Firm	Allocation Pending	N	
APS	Claim Allocation APS Group	Claimed	Y	<p>Given Trades allocated to APS Group then TCR GrpID2=abc, AvgPxInd=1, AllocInd=1</p> <p>Given Allocation performed on APS group to Carry Firm then No TCR When Claiming Firm Claims Allocation</p> <p>Then the Buy Side Receives TCR with OfstInst=0, GrpID2=abc, AvgPx, ClOrdID="Group Name"</p> <p>Then the Claiming firm Receives TCR with OfstInst=1, GrpID2=abc, AvgPx, ClOrdID="Group Name"</p>
<b>APS - Full Lifecycle</b>				
APS	Full Cycle APS			<p>Given Trades allocated to APS Group, then TCR GrpID2=abc, AvgPxInd=1, AllocInd=1</p> <p>Given Allocation performed on APS group to Carry Firm, then No TCR When Claiming Firm Claims Allocation</p> <p>Then the Buy Side Receives TCR with OfstInst=0, GrpID2=abc, AvgPx, ClOrdID="Group Name"</p> <p>Then the Claiming firm Receives TCR with OfstInst=1, GrpID2=abc, AvgPx, ClOrdID="Group Name"</p>
	Create Group		Y	Given Trades allocated to APS Group, then TCR GrpID2=abc, AvgPxInd=1, AllocInd=1
	Complete Group		N	No TCR sent to STP
	Allocate Trade to claiming Firm		N	No TCR sent to STP
	Claim Allocation		Y	<p>Buy Side Receives TCR with OfstInst=0, GrpID2=abc, AvgPx, ClOrdID="Group Name"</p> <p>Claiming firm Receives TCR with OfstInst=1, GrpID2=abc, AvgPx, ClOrdID="Group Name"</p>
	Reverse Allocation		N	No TCR sent to STP
	Accept Reversal		Y	Then the Buy and Claiming side receive a Cancel message
<b>Sub-Allocation</b>				
	Creating a Sub-Allocation	Claimed	Y	<p>When the Claiming side Marks for SubAllocation</p> <p>Then no TCR is sent</p> <p>When The Buy Side is allocated to firm Claiming and Trade Claimed</p> <p>Then The Claiming Side Receives an Offset Message</p> <p>Then The Re-allocated Side Receives an Onset Message</p>

## Allocation of Spreads

CME STP allocates spreads at the leg level only, so the API does not mark spread level messages for allocation. For users requesting @MLegRptTyp = 2 (individual leg of a multi-leg security), allocation of spread legs is no different than allocation of outright trades, except that @MLegRptTyp = 2 instead of 1.

For firms requesting spread level messages, (for example, @MLegRptTyp = 3), allocation behaves differently. When a clearing firm marks a leg of a spread for allocation, the API sends a message showing that individual leg as an @MLegRptTyp = 2 to the executing firm, even though the user only requested @MLegRptTyp = 3. The field @StrategyLinkID can be used to link the allocated leg to the multi-leg spread message. The offset and onset trades after the claiming firm claims the allocation are no different than before. Should a clearing firm un-mark a trade for allocation (not illustrated), the API will send a message with @MLegRptTyp = 2 showing the leg as un-marked.

**!** When requesting @MLegRptTyp = 3, be advised that any @MLegRptTyp = 2 messages received only indicate allocation status. These @MLegRptTyp = 2 messages must not be booked as new trades, considering that they duplicate one or more legs of the @MLegRptTyp = 3 messages already received. Do not double count these trades.

**i** This diagram shows only responses, and not requests.

### Allocation of Spreads

