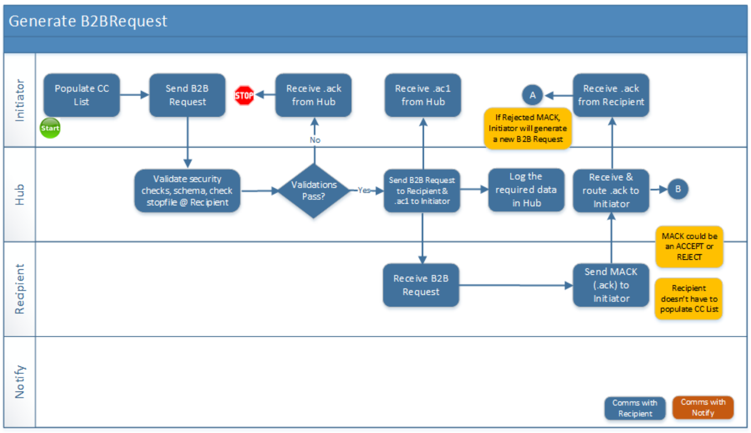
Option 1: Initiator to send new/existing notification transaction

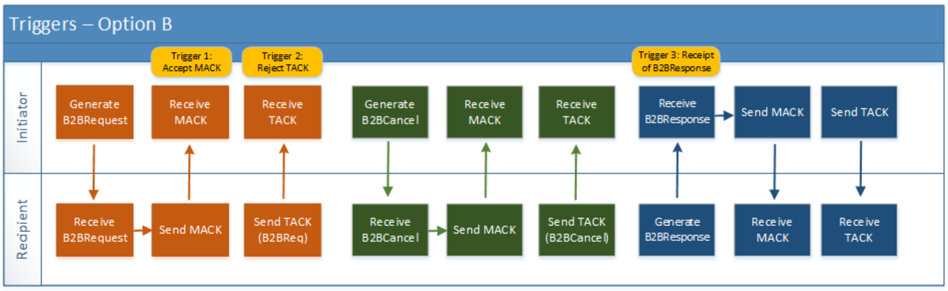
Principle: Minimal changes to all Participants

1. Initiator sends the request to the recipient via the Hub
2. Initiator populates the CC list in the payload
3. Hub routes the request to the recipient; no changes to the existing process

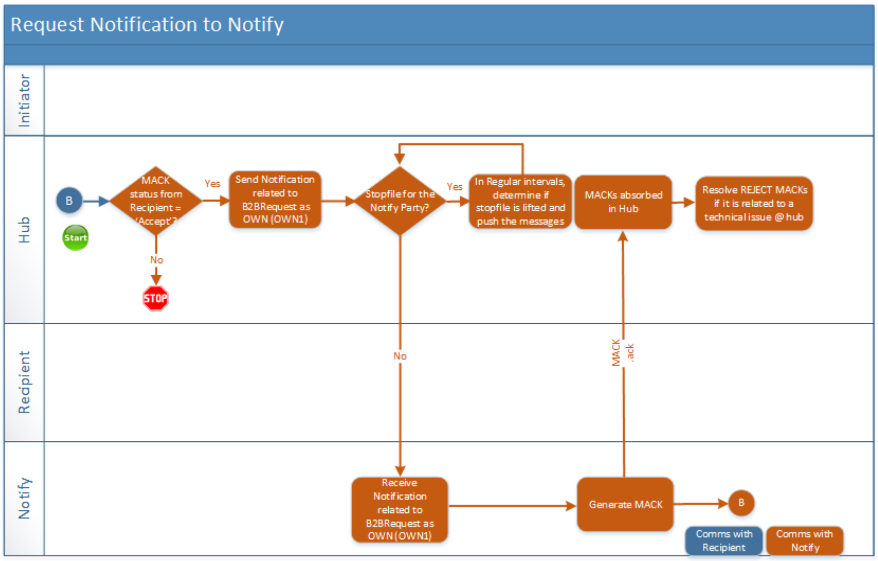


Generate B2BRequest
Initiator
Hub
Recipient
Notify
Populate CC List
Start
Send B2B Request
STOP
Receive .ack from Hub
No
Receive .ac1 from Hub
A
If Rejected MACK, Initiator will generate a new B2B Request
Receive .ack from Recipient
Validate security checks, schema, check stopfile @ Recipient
Validations Pass?
Yes
Send B2B Request to Recipient & .ac1 to Initiator
Log the required data in Hub
Receive & route .ack to Initiator
B
MACK could be an ACCEPT or REJECT
Receive B2B Request
Send MACK (.ack) to Initiator
Recipient doesn't have to populate CC List
Comms with Recipient
Comms with Notify

4. Hub has to manage the state of the life cycle of the B2B process
5. Triggers to Notify Parties
   a. Option 1 (Hub to create new/existing notification transaction):
      i. **Pre-determined trigger points (that meet the timing obligations to be drafted in the procedures)**
      ii. In the SWG meeting, it was agreed that the following trigger points will be configured in the Hub
         1. Generate B2BRequest
         2. Reject MACK on a B2BRequest
         3. Reject TACK on a B2BRequest
         4. Receipt of a B2BResponse (could be related to a SORequest or SOCancel)
      iii. AEMO has reviewed the above discussion internally and is proposing the following trigger points in the hub



   b. Option 2 (Initiator to send new notification transaction):
      i. **Time based obligations to be drafted into the procedures around notifications as configured by the Initiator**
6. Hub will transform the B2B request to OWN and send the OWN to the Notify Party (note: OWN is an AEMO recommendation as opposed to a new B2BNotification transaction discussed)


Request Notification to Notify
Initiator
Hub
Recipient
Notify
B
Start
MACK status from Recipient = 'Accept'?
Yes
No
STOP
Send Notification related to B2BRequest as OWN (OWN1)
Stopfile for the Notify Party?
Yes
No
In Regular intervals, determine if stopfile is lifted and push the messages
MACKs absorbed in Hub
Resolve REJECT MACKs if it is related to a technical issue @ hub
MACK .ack
Receive Notification related to B2BRequest as OWN (OWN1)
Generate MACK
B
Comms with Recipient
Comms with Notify

a. OWN to Notify Party
   i. Notification not a CSV
   ii. B2BWG to determine the content of the OWN (Suggested by some participants that the whole Service orders request content may not be required)
   iii. Identify the event codes that are applicable to TACKing a Notification

b. New MessageID in the header

c. New TranID in the transaction

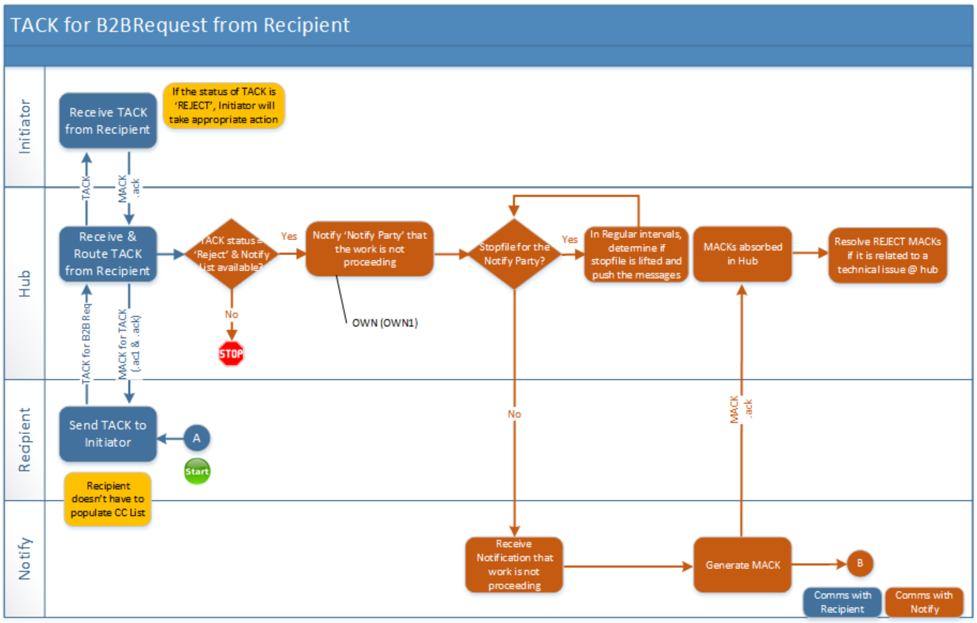
d. From – Initiator

e. To – Notify

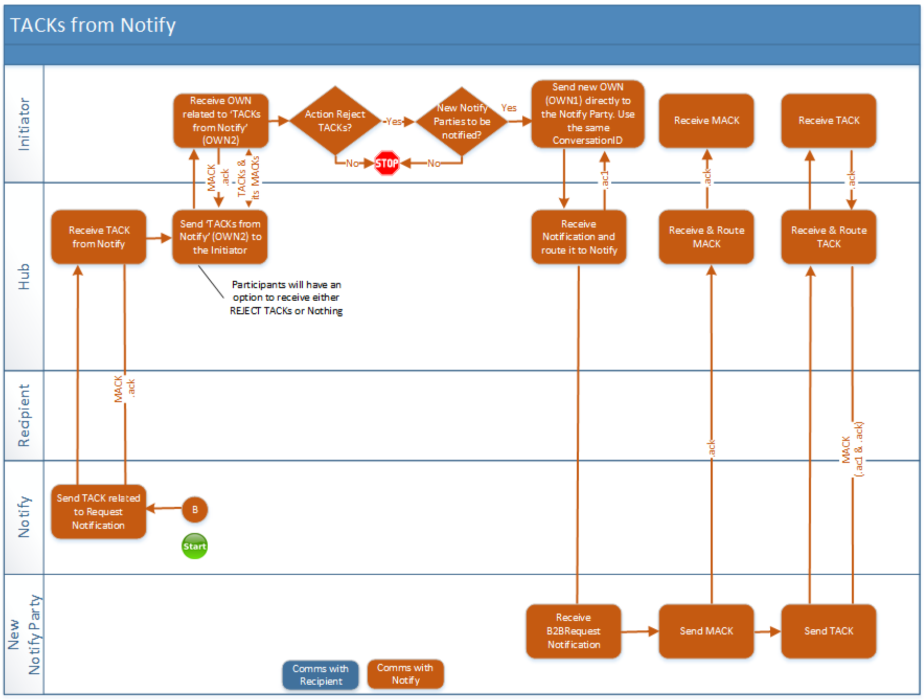
f. ConversationID in the payload

g. Population of 'Security Context' *(this is to enable notified parties to identify the hub generated messages)*

7. MACKs from Notify is all absorbed by Hub (Yet to be confirmed and agreed by AEMO)
8. MACKs and TACKs will be stored and accessible in the Hub
9. Processing of TACKs from the Recipient. The TACK will be sent to the Initiator (current logic) and the Hub will determine if a notification to the Notify Party is required i.e. if the TACK status from the Recipient is 'Reject', hub needs to notify 'Notify Party' that the 'work is not proceeding'


TACK for B2BRequest from Recipient
Initiator
Hub
Recipient
Notify
Receive TACK from Recipient
If the status of TACK is 'REJECT', Initiator will take appropriate action
TACK
MACK .ack
Receive & Route TACK from Recipient
TACK status = 'Reject' & Notify List available?
Yes
No
STOP
Notify 'Notify Party' that the work is not proceeding
OWN (OWN1)
Stopfile for the Notify Party?
Yes
No
In Regular intervals, determine if stopfile is lifted and push the messages
MACKs absorbed in Hub
Resolve REJECT MACKs if it is related to a technical issue @ hub
TACK for B2B Req
MACK for TACK (.acl & .ack)
Send TACK to Initiator
A
Start
Recipient doesn't have to populate CC List
MACK .ack
Receive Notification that work is not proceeding
Generate MACK
B
Comms with Recipient
Comms with Notify

10. TACK processing - From Notify


TACKs from Notify
Initiator
Hub
Recipient
Notify
New Notify Party
Receive OWN related to 'TACKs from Notify' (OWN2)
Action Reject TACKs?
Yes
No
STOP
New Notify Parties to be notified?
Yes
No
Send new OWN (OWN1) directly to the Notify Party. Use the same ConversationID
Receive MACK
Receive TACK
MACK .ack
TACKs & its MACKs
.acl
.ack
.ack
Receive TACK from Notify
Send 'TACKs from Notify' (OWN2) to the Initiator
Participants will have an option to receive either REJECT TACKs or Nothing
Receive Notification and route it to Notify
Receive & Route MACK
Receive & Route TACK
MACK .ack
.ack
MACK (.acl & .ack)
Send TACK related to Request Notification
B
Start
Receive B2BRequest Notification
Send MACK
Send TACK
Comms with Recipient
Comms with Notify

a. Initiator can choose to receive
    i. Reject TACKs – Will be sent to Initiator as OWN
        1. OWN will have its own MACKs and TACKs
        2. TACKs to the OWN - The Initiators will always accept these OWNs and no Rejects
    ii. Nothing (not receive any notification of Notified parties rejecting the OWN)

b. Initiator will action the REJECT TACKs (OWN) from the Notify Party

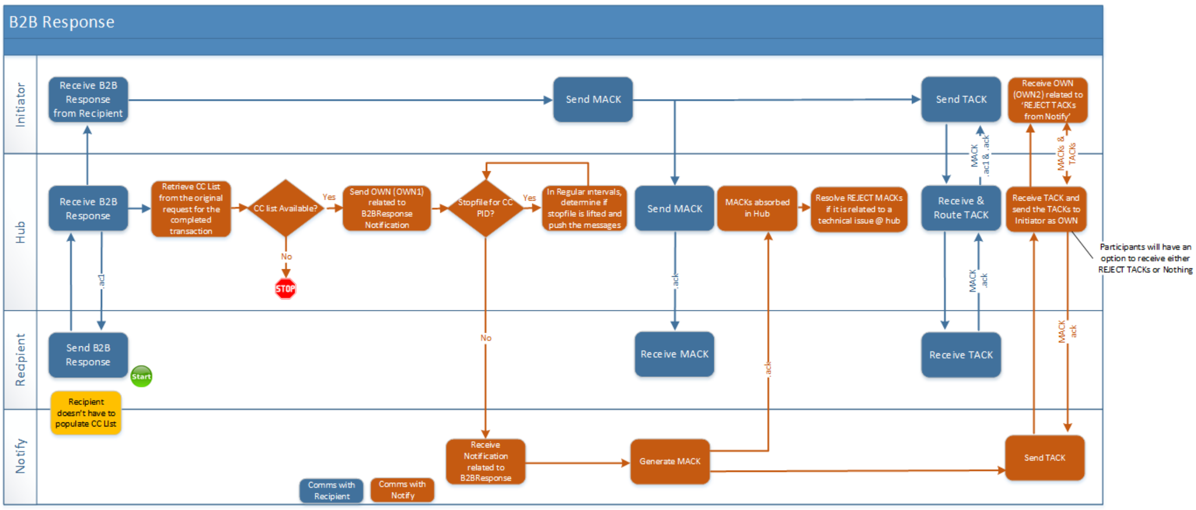
c. If Initiator requires to send the notification to a new CC
    i. The following approach was discussed in the SWG Focus Group meeting
        - Initiator will send updated CC list (complete list of valid parties) to the Hub using <ServiceOrderRequest>; actionType set to 'CC Updates' (TBC)

- Hub will decide the new notification parties or parties that do not require any further notifications by comparing the CC list from the original B2B request and the new B2B request for CC updates
- Hub will send the B2B Notifications to the new Notify Party(ies)
- The Hub will send the sequence of notifications based on the current status of the B2B Request e.g. If the completion has been received and the new CC list is received; hub will send the B2Brequest notification followed by B2Bcompletion notification

ii. AEMO had reviewed the above recommendation and is proposing the following to minimise the changes for participants and hub

- Initiator will generate a new OWN (related to notifying the Notify Party) addressing it directly to the new Notify Party. Initiator will reference the conversation id in the notification. Hub will log the conversation id and the new CC party id for further notifications if any. *(It was suggested in the focus group feedback that there could be a defined set of roles that could be the notified party in this message, this could resolve the issues around ensuring only the correct combination can be provided)*

11. Ignore the CC list in the SORequest; actionType = 'Cancel'
12. B2BResponse notifications (as OWN) will be sent by the hub as shown below


B2B Response
Initiator
Hub
Recipient
Notify
Receive B2B Response from Recipient
Send MACK
Send TACK
Receive OWN (OWN2) related to 'REJECT TACKs from Notify'
Receive B2B Response
Retrieve CC List from the original request for the completed transaction
CC list Available?
Yes
No
STOP
Send OWN (OWN1) related to B2BResponse Notification
Stopfile for CC PID?
Yes
No
In Regular intervals, determine if stopfile is lifted and push the messages
Send MACK
MACKs absorbed in Hub
Resolve REJECT MACKs if it is related to a technical issue @ hub
Receive & Route TACK
Receive TACK and send the TACKs to Initiator as OWN
MACK
.ac1 & .ack
MACKs & TACKs
.ack
.ack
MACK
.ack
MACK
ack
.ac1
Participants will have an option to receive either REJECT TACKs or Nothing
Send B2B Response
Start
Recipient doesn't have to populate CC List
Receive MACK
Receive TACK
Receive Notification related to B2BResponse
Generate MACK
Send TACK
Comms with Recipient
Comms with Notify

Key points:

B2BWG

- B2BWG need to:
  - Determine new OWN1/B2BNotification1 (New transaction type or potentially a copy of the service order flagged as a notification, not CSV content) transaction content to be sent to Notified parties from hub or initiator, this can either be the full ServiceOrderRequest content or a subset
  - Determine new OWN2/ B2BNotification2 (New transaction type, not CSV content) transaction content to be sent to initiators on Rejection of initial OWN transaction
  - Service order process updated to align with Notified Party concepts detailed above

Initiator

- Initiator choses to either supply CC in ServiceOrderRequest (option 1) or not (option 2)
- Initiators need the ability to receive inbound OWN2 transaction to manage rejections from notified parties (unless they choose to not receive rejections)
- Initiator needs the ability to trigger OWN1 to notified parties to either notify (option 2) or update (Option 1 exception scenario)

Notified Parties

- The Notified parties will always receive the same transaction type and content irrespective of what option the initiator uses
- Notified parties will need to build functionality to receive and process the new OWN1 transaction