== DRAFT ==

ECM DOCUMENT 6

-

RECOMMENDATIONS FOR THE EFET BOX

VERSION 1.4.

CREATED BY EFET IT TF – BUSINESS WORKGROUP

Revision History

Version	Date	Changes	Author of changes
Version 1.0		Created by EFET IT TF – Business workgroup	Kris Bouckaert (Transmeta)
Version 1.1	19/8/2002	Amended to include: • Revision History • Replace affirmation with authentication and ECMS with ECM • Complete re-write of chapters II and IV to include handling of messages I the case of matching through a Central Matching Service	Kris Bouckaert (Transmeta)
Version 1.2	26/8/2002	Rename quantity field to capacity and remove country of origin. Included buyer and seller energy account.	Sophie van der Haegen
Version 1.3	16/9/2002	Align document names	
Version 1.4	8/10/2002	Copyright notice Align with Brussels 3/10/2002 meeting	Sophie van der Haegen

Copyright notice

Copyright © EFET 2002. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to EFET except as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by EFET or its successors.

This document and the information contained herein is provided on an "as is" basis.

EFET DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

I. INTRODUCTION 5

II. THE (GENERAL) EFET BOX 6

III. FUNDAMENTAL REQUIREMENT TO COMPLY WITH EFET STANDARDS 9

III.1. COMPLIANCY WITH STANDARD EFET ECM PROCESS 9
III.2. COMPLIANCY WITH EFET STANDARDS 9

IV. STRONG FUNCTIONAL RECOMMENDATIONS FOR ECM MODULE 10

IV.1. DISPATCHER LOGIC 10
IV.1.1. Peer-to-Peer Authentication 11
IV.1.2. Matching through a Central Matching Service 13
IV.2. PEER-TO-PEER AUTHENTICATION LOGIC 14
IV.2.1. Authentication Algorithm 15
IV.2.2. External Confirmation Life Cycle 16
IV.2.3. Exception handling 17
IV.3. REPORTING LOGIC 18

V. OTHER CONSIDERATIONS 21

V.1. GRAPHICAL USER INTERFACE 21
V.2. AUDIT TRAIL AVAILABILITY 21
V.3. ARCHITECTURE 21

I. INTRODUCTION

The goal of this document is to formulate business and technical recommendations for those EFET members that want to build their own EFET box. This document primarily focuses on the ECM module within this EFET box.

Chapter II briefly explains the EFET box and the different building blocks within this application. The EFET XML handler, which takes care of the communication with the outside world (other EFET members or e.g. the central matching service provider), is one of these building blocks.

Chapter III highlights the fact that the EFET box shall comply with the EFET standards with respect to the interface definition and the processes. These standards are extensively described in documents **EFET ECM doc2 - EFET Standards 1.0 - Central Matching and P2P Authentication Process**, **EFET ECM doc4 - EFET Standards 1.0 - Core Components & Coding Scheme** and **EFET ECM doc5 - EFET Standards 1.0 - ECM Interface Definition**.

Chapter IV contains the functional specifications for the ECM module of the EFET box. They are based on the definitions, as pointed out in **EFET ECM doc1 - Lexicon** and the best practises for the trade confirmation process in **EFET ECM doc2 - EFET Standards 1.0 - Central Matching and P2P Authentication Process.**

Chapter V contains additional considerations such as technical requirements.

This EFET box is not an actual part of the EFET standard. The only requirement for the EFET box is to produce ECM messages that comply with the EFET standards. **The specifications listed in chapters IV and V should therefore be interpreted as recommendations rather than requirements.**

II. THE (GENERAL) EFET BOX

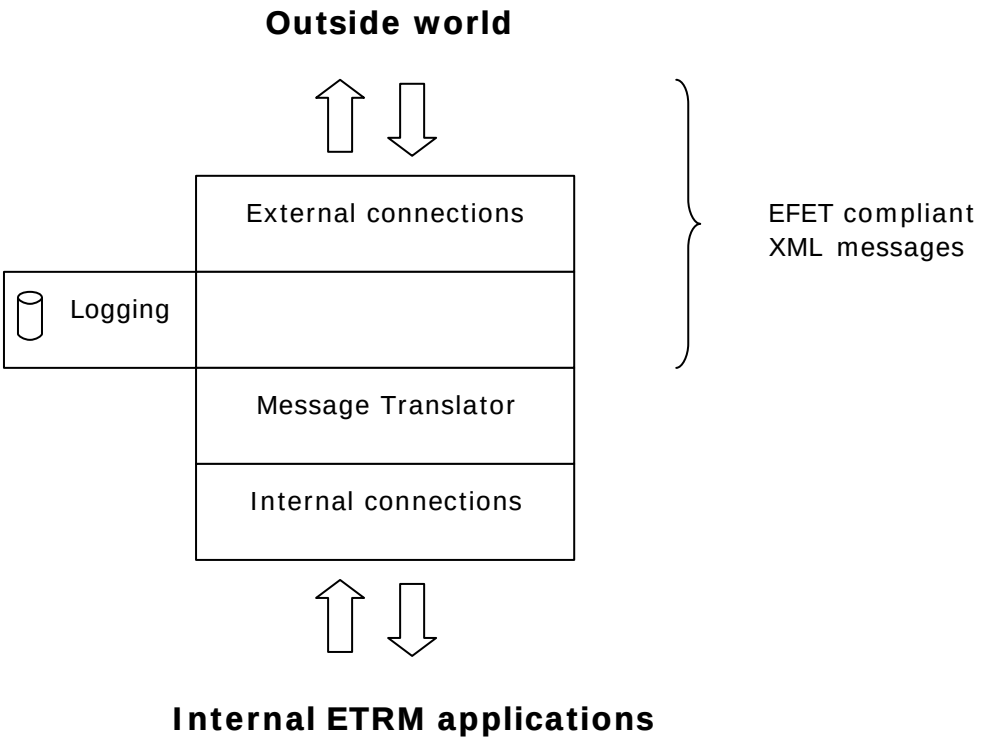
The EFET box consists of two main parts:

- The XML handler
- One or more functional modules

The XML handler is an application that:

- Translates the information coming from internal ETRM applications into an EFET compliant message.
- Manages incoming EFET compliant messages.
  If appropriate, it can translate the incoming message into a message that can be read by the proprietary ETRM systems.
- Logs both incoming and outgoing EFET compliant messages.

The XML handler could be pictured as follows:



To explain what each module is meant to do, we have split the logic by type of message.

A) Outgoing messages (from internal ETRM applications to outside world)

1) Internal connections:
   This module allows internal ETRM applications to post messages to the EFET box. These messages will typically be in the proprietary format

of the sending ETRM application. The ETRM applications can send messages by placing the message in an inbox folder or a message queue. The "Internal connections" module will simply pass on these messages to the "Message Handler" module. (One could also build in a logging functionality at this place, i.e. logging the messages in their proprietary format.)

2) Message Translator:
This module will translate the messages, coming from internal ETRM applications, into the standard EFET message format for that type of message. It will then pass the message (in the EFET standard message format) to the dispatcher module.

3) Dispatcher:
This module will look at the type of message and decide what needs to be done with this type of message. This module therefore contains the workflow logic. If the message needs to be sent out to an external party, it will decide to which party it needs to be sent. For example, if the message is an ECM confirmation message, it will either send it to the counter party itself of to a central matching service provider, depending on the counter party. This module will therefore tag the message with the correct recipient, before passing it on to the "External Communications" module. The module will typically also send a copy of the message to the logging tool.

4) External Communications:
This module will set up the connection with the correct system of the counter party and send the message to that system. This module contains the technical connection information for each external system so that knows how to connect with each party.

B) Incoming messages (from outside world to internal ETRM applications)

1) External Communications:
This module allows other parties to post messages to the EFET box. These messages need to be EFET standard messages. The other parties can send messages by placing the message in an inbox folder or a message queue. The "External connections" module will simply pass on these messages to the "Dispatcher" module.

2) Dispatcher:
This module will look at the type of message and decide what needs to be done with this type of message. This module therefore contains the workflow logic. It will decide if the message needs to be sent to one or more internal ETRM applications. This module will therefore tag the message with the correct internal recipient(s), before passing it on to the "Message Translator" module. If there are functional modules within the EFET box, the dispatcher will see to it that the correct messages are sent to the correct functional module. The module will typically also send a copy of the message to the logging tool.

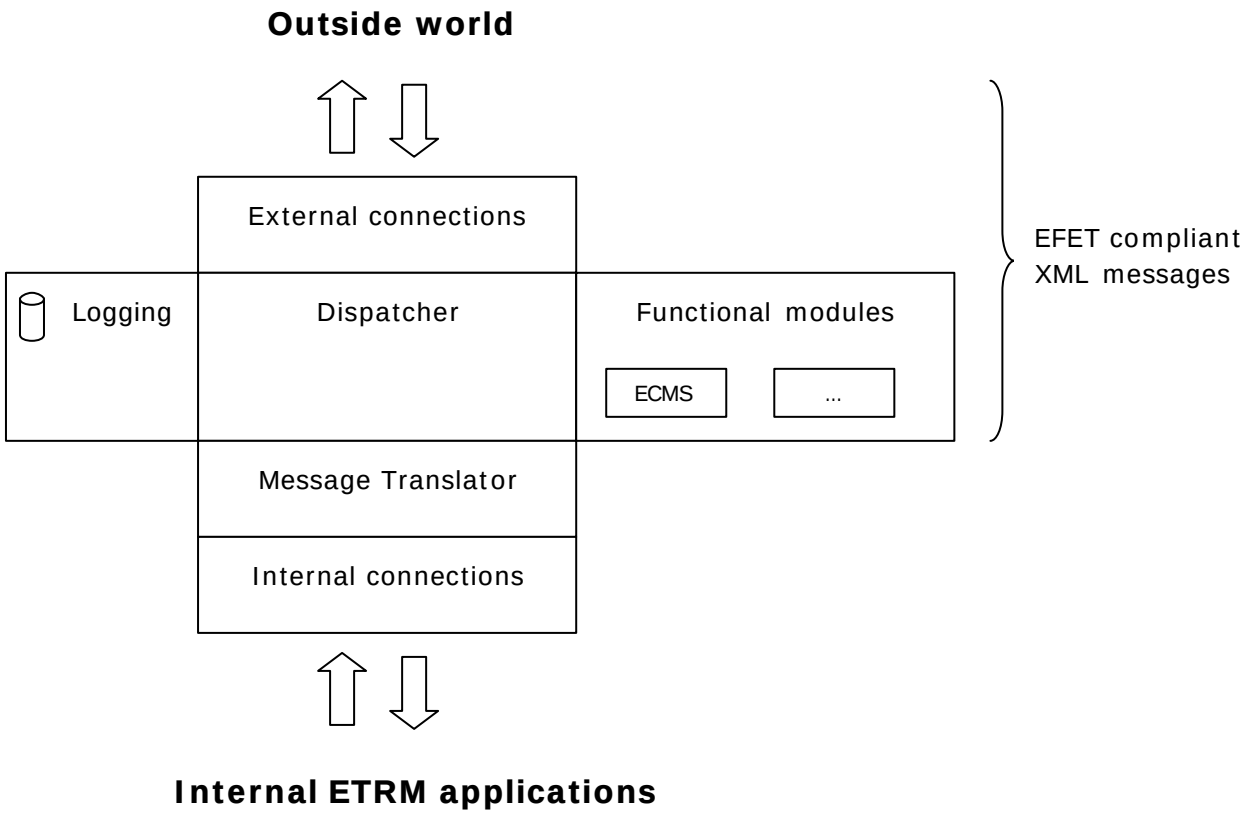
3) Message Translator:
This module will translate the EFET standard messages into a format that the receiving ETRM application can read and understand. This module holds the formats used by each ETRM application, so that it

knows to which format it needs to translate the message. It will then pass the message to the "Internal Communications" module.

4) Internal Communications:
This module will take care of the sending of the message(s) to the correct internal ETRM application(s). This module holds the communication data for each of the ETRM applications, so that it knows how to communicate with each of them.

If the EFET box also contains functional modules, they will communicate with the dispatcher. This means that they will work with EFET standard messages. The dispatcher module contains the workflow logic that regulates which messages to send to which functional modules, and what to do with messages that it gets back from a functional module.

The complete EFET boxs could therefore be pictured as follows:



III. FUNDAMENTAL REQUIREMENT TO COMPLY WITH EFET STANDARDS

This chapter highlights the **ABSOLUTE and FUNDAMENTAL REQUIREMENT** for a Central Matching Service to comply with:

- The standard ECMS process as described in **EFET ECM doc2 - EFET Standards 1.0 - Central Matching and P2P Authentication Process**
- The standard ECM interface as described in **EFET ECM doc4 - EFET Standards 1.0 - Core Components & Coding Scheme** and **EFET ECM doc5 - EFET Standards 1.0 - ECM Interface Definition**

III.1. COMPLIANCY WITH STANDARD EFET ECM PROCESS

The ECM module within the EFET box will take trade confirmations as input and send back an authentication, interpreting the different scenarios as specified in **EFET ECM doc2 - EFET Standards 1.0 - Central Matching and P2P Authentication Process**. The authentication will contain the other party's trade confirmation content. If no match could be found within a reasonable time, this will be flagged to the user within the ECM module. The user can then contact the counter party in order to find out what went wrong.

III.2. COMPLIANCY WITH EFET STANDARDS

All messages treated by and generated by the functional ECM module have to comply with the EFET ECM Standard as described in **EFET ECM doc4 - EFET Standards 1.0 - Core Components & Coding Scheme** and **EFET ECM doc5 - EFET Standards 1.0 - ECM Interface Definition.**

All messages treated by the "External Communications", the "Dispatcher" module and the "Logging" module have to comply with the EFET general message Standard as described in **EFET ECM doc4 - EFET Standards 1.0 - Core Components & Coding Scheme** and **EFET ECM doc5 - EFET Standards 1.0 - ECM Interface Definition.**

Note:

The EFET standards, as described in Document 4 and Document 5, are meant to be an open standard.

The EFET standards are managed by EFET and available for everyone without restriction to implement, use, publish and promote.

IV. STRONG FUNCTIONAL RECOMMENDATIONS FOR ECM MODULE

This chapter highlights the basic logic contained in the ECM module within the EFET box. These functional descriptions are, as opposed to the compliancy requirement and content of the previous chapters, not subject to standardization and but very STRONG RECOMMENDATIONS towards any company wanting to build an EFET box.

This module covers three logical function blocks:

- **The dispatcher** that will handle all incoming and outgoing messages and is responsible for dispatching them to the correct internal EXIS modules, other internal application or external party. Though the dispatcher, strictly speaking, is not part of the ECM module as such and is generic to the EFET box and its different modules, it has to be adapted to support the different modules. The dispatcher logic applicable to the ECM processes is described in the first paragraph.
- The **peer-to-peer authentication** logic. That part of the ECM Module will take care of the authentication of external and internal confirmations. The authentication process is to some extent similar to the matching process of a central matching service and will be further detailed in the second paragraph.
- **The reporting logic** should allow queries and reporting on all ECM related transactions and thus not only on those applicable to the Peer-to-Peer connection process but also to the matching process via a Central Matching Service. The reporting logic is described in the third paragraph.

IV.1. DISPATCHER LOGIC

The ECM messages received by the dispatcher will relate to the peer-to-peer or central matching service processes. The dispatcher will have to decide to which process they relate.

A trade being reported by the internal ETRM system, the dispatcher will have to look up in a system table what process to use. The lookup table will give for each possible combination of counter party and trade type the appropriate process. In case the process is the matching through a central matching service provider, the table should also contain the details relative to which central matching service is to be used. Similarly in case the process is authentication through a peer-to-peer connection, the table should also contain the details relative to how to connect to the other party and who in that case is responsible for sending the authentication.

QUESTION: *What are the rules that decide which of the two parties needs to send the authentication. Is it always the buyer, always the seller, … Is it possible that two parties agree that one of the two always sends the authentication, because the other one doesn't have any automated matching logic?*

The next sections now details, once the type of process identified what role the dispatcher has in each of these processes.

IV.1.1. PEER-TO-PEER AUTHENTICATION

IV.1.1.1. THE OWNER OF THE APPLICATION IS MEANT TO GENERATE THE AUTHENTICATION

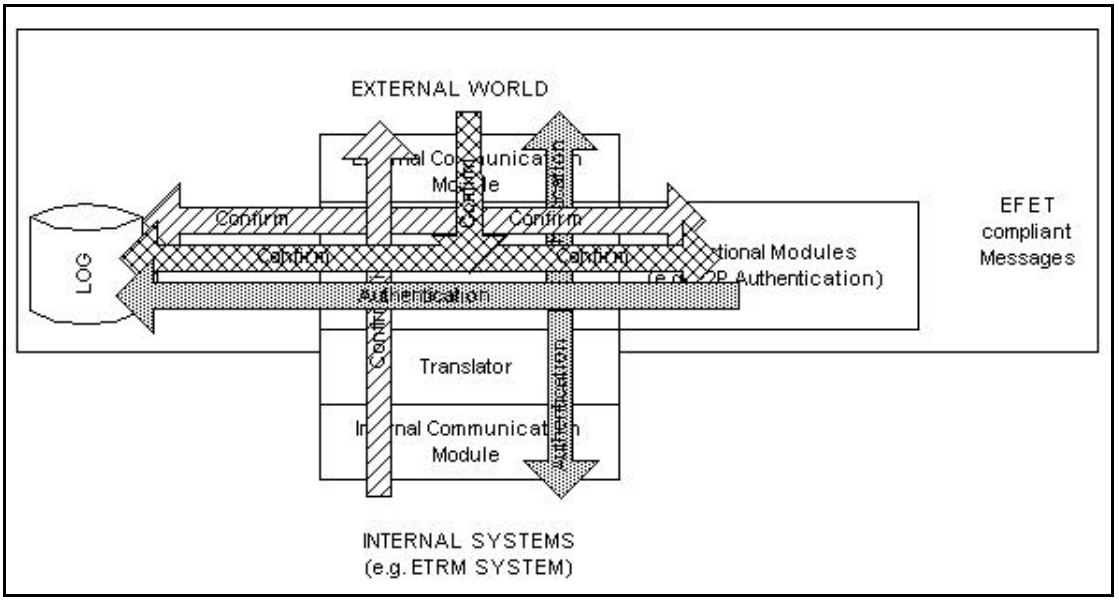


Figure 1. Message Flow for P2P Authentication – Scenario 1

Internal confirmation message

When an internal ETRM application generates a trade confirmation messages, it will hand it over to the EFET box. After reception in the "Internal Communications" module, and translation in the "Message Translation", it will arrive as a standard EFET confirmation message in the Dispatcher module.

The Dispatcher module will send the message to the "External Communications" module to be sent to the other party and store a copy in the logging module.

The Dispatcher module will send the message to the functional Peer-to-Peer Authentication Module, where the confirmation will be put in the **internal matching queue.**

The Dispatcher module will store the Document ID and give it status "**Peer-to-peer pending – authentication due**".

External confirmation message

When an external trade confirmation message comes in from an external party, it will be received by the "External Communications" module and transmitted to the Dispatcher.

The Dispatcher module will send a copy in the logging module.

The Dispatcher module will transfer the message to the functional Peer-to-Peer Authentication Module, where it will be put on the **External Matching Queue**.

Authentication message

In the event the Peer-to-peer authentication module finds a match, it will generate an authentication message and send it to the Dispatcher module.

The Dispatcher module will send a copy in the logging module.

The Dispatcher will update the status of the corresponding Document ID to "**Authenticated**" and pass on the authentication message to the "Message Translator" module where it can be translated into the appropriate format to be processed within the internal ETRM system.

IV.1.1.2. THE OTHER PARTY IS MEANT TO GENERATE THE AUTHENTICATION

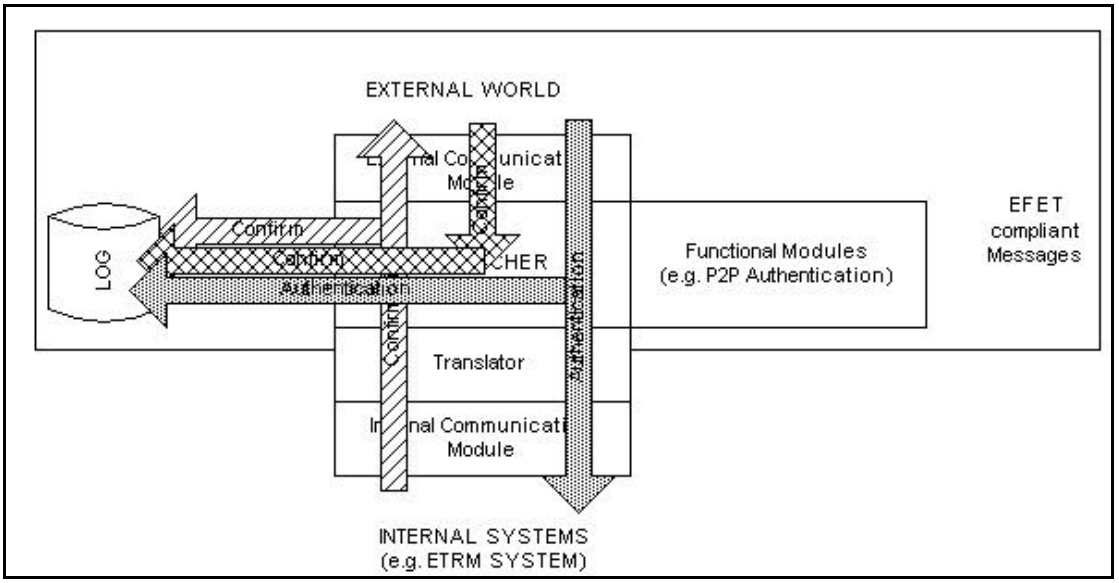


Figure 2. Message Flow for P2P Authentication – Scenario 2

Internal Confirmation message

When an internal ETRM application generates a trade confirmation messages, it will hand it over to the EFET box. After reception in the "Internal Communications" module, and translation in the "Message Translation", it will arrive as a standard EFET confirmation message in the Dispatcher module.

The Dispatcher module will send the message to the "External Communications" module to be sent to the other party.

The Dispatcher module will store the "Document ID" and give it status "**Peer-to-peer pending – authenticated awaited**".

The Dispatcher module will also send a copy of the message to the logging module.

External confirmation message

When an external trade confirmation message comes in from an external party, it will be received by the "External Communications" module and transmitted to the Dispatcher.

The Dispatcher module will send a copy to the logging module.

Authentication message

When an authentication message comes in from an external party, it will be received by the "External Communications" module and transmitted to the Dispatcher.

The Dispatcher module will send a copy in the logging module.

The Dispatcher will update the status of the corresponding Document ID to "**Authenticated**" and pass on the authentication message to the "Message Translator" module where it can be translated into the appropriate format to be processed within the internal ETRM system.

IV.1.2. MATCHING THROUGH A CENTRAL MATCHING SERVICE

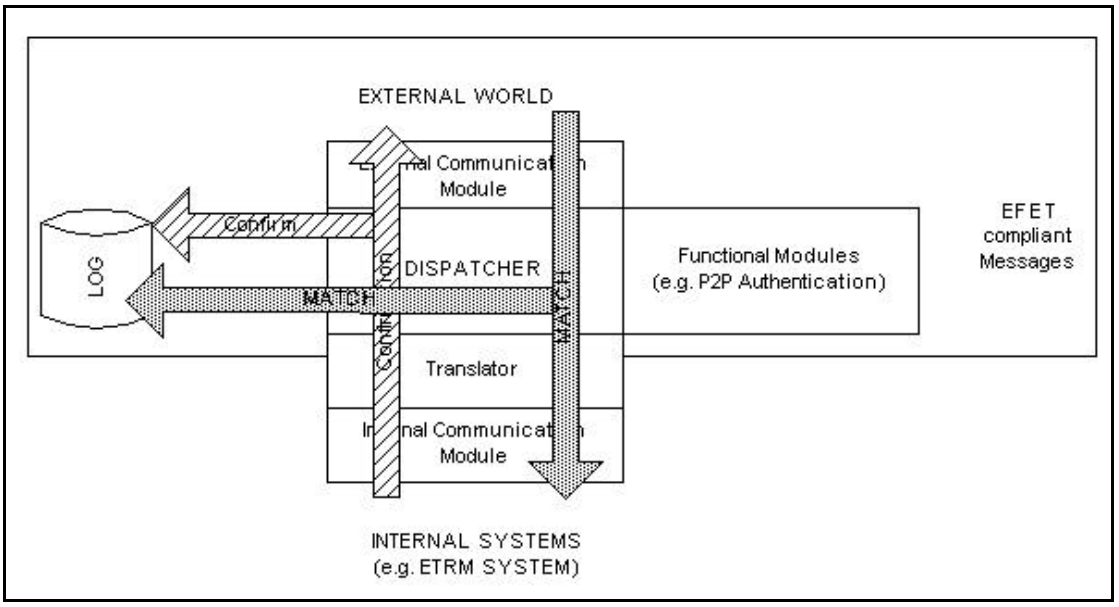


Figure 3. Message Flow for Matching through Central Matching Service

Internal Confirmation message

When an internal ETRM application generates a trade confirmation messages, it will hand it over to the EFET box. After reception in the "Internal Communications" module, and translation in the "Message Translation", it will arrive as a standard EFET confirmation message in the Dispatcher module.

The Dispatcher module will transfer the message to the "External Communications" module, which will send the confirmation to the Central Matching Service.

The Dispatcher module will keep a track of the Document ID and give it status "**CMS pending**".

The Dispatcher module will also send a copy of the message to the logging module.

Match message

When a match message comes in from the Central Matching Service, it will be received by the "External Communications" module and transferred to the Dispatcher Module.

The Dispatcher module will then send a copy to the logging module.

The Dispatcher module will update the status of the corresponding Document ID to "**matched**" and transfer the authentication message to the "Message Translator" module where it can be translated into the appropriate format to be processed within the internal ETRM system.

IV.2. PEER-TO-PEER AUTHENTICATION LOGIC

The main differences with the matching process of a central matching service, as described in **EFET ECM doc3 - Recommendations for Central Matching Service** are the following:

- The trade confirmations will now be split into two groups: the INTERNAL confirmations and the EXTERNAL confirmations. The INTERNAL confirmations are the confirmations produced by the owner of the application based on the trade data available in his ETRM system.
- A new external confirmation will only be compared to the list of unmatched internal confirmations, and a new internal confirmation will only be compared to the list of unmatched external confirmations.
- Depending on who is meant to send the authentication on a trade, the owner of the application or the counter party, the internal confirmation will be sent to the matching queue or not.
- When a match has been found between an internal confirmation and an external confirmation, an authentication will be generated, rather than a match message. The authentication will be sent to the other party. Optionally, it can be sent to the internal ETRM system as well, where it can be stored and/or where the status of the deal can be updated to "fully confirmed".
- Both the owner of the application and the external party will in the end have received three messages, which belong together. These are the internal confirmation message, the external confirmation message and the authentication message (be it received by the EFET box from the external

party or generated by the EFET box). The authentication should link the three messages together, as it contains the ID's of both confirmations.

IV.2.1. AUTHENTICATION ALGORITHM

In the peer-to-peer trade confirmation matching, their will be two separate matching queues, one for the to-be-matched internal confirmations, and on for the to-be-matched external confirmations. Both queues will be compared against each other.

A distinction is to be made between Key Fields and Information Fields. Within the set of key fields, one could further distinguish between primary key fields and secondary key fields.

Prior to the matching process being undertaken there will be a validation to check the message format and content validity.

Once validated, a confirmation will enter the matching algorithm. It will be attributed the status "pending" until a confirmation on the other queue has been found that confirms the same trade.

For two confirmations to be seen as confirming the same trade, all **Key Fields** (Primary + Secondary) have to be identical. This means that two confirmations with a difference in a key field value will not be identified as confirming the same trade, i.e. they will not be identified as a match.

If two confirmations match on all Primary Key Fields, but not on all Secondary Key Fields, the trade referred to by the internal confirmation is reported "pending with mismatch". This indicates that there is an external confirmation in the system that might correspond to the internal confirmation on that trade but that does not match entirely.

Internal confirmations referring a trade "pending with mismatch" remain in the matching queue and will continue to be checked against other possible matches.

The **Information Fields** are only used for information purposes and are not to be taken into account in the matching algorithm.

Key Fields

- Buyer (1)
- Buyer Energy Account
- Seller (2)
- Seller Energy Account
- Market
- Commodity
- Load Type
- Trade Type
- Trade Date
- Agreement
- Broker (*)
- Memo field (*)
- Delivery Point (3)
- Start Date-time (4)

- End Date-time (5)
- Contract Capacity (*)
- Capacity Unit (*)
- Total Volume (*)
- Currency
- Price (*)

The fields marked with a (*), are good candidates for secondary key fields, the other fields would then be the primary key fields. Within the primary key fields, a matching order (1 to 5) has been identified to streamline the matching process and enhance performance. All other Key Fields can be treated in any further sequence.

Note: Normally, the memo field will only be filled in for complex deals. In such a case, it will probably be the case that automatic confirmation matching fails to recognise two deal confirmations as being identical, due to a different phrasing in the memo field. The party that needs to send the authentication will then manually validate the confirmation of the other party, more in particular the memo field. If he agrees, he will send an authentication and the deal is affirmed. If the two parties agree on the exact structure of the extra deal information they put in the memo field, the automatic confirmation matching within the EFET box can be used to match complex deals automatically.

Information Fields

- Document ID
- Trader Name
- Trade Time
- Document Version

Note: In the future, when gas and gas options will be supported extra required and information fields might be added.

In case of a match, a **unique authentication ID** shall be generated. The Peer-to-Peer Authentication Module will then generate and send an **authentication message** to the other party and send a message to the internal ETRM system to update the internal trade status accordingly.

This authentication message contains the AuthenticationID, both Document ID's and the full details of the external confirmation. It should be regarded as the countersigned version of the external confirmation that authenticates all key fields of the external confirmation.

IV.2.2. EXTERNAL CONFIRMATION LIFE CYCLE

Reporting on an EXTERNAL Confirmation should consider following statuses:

- **Received**: The confirmation has been received within the EFET box, but has not yet been checked for validity.

- **Rejected**: The confirmation has been rejected (did not pass validation) and is archived. Reasons for rejection can be invalid data (not EFET ECM compliant), buying or selling party not owner of the application, …
- **Pending**: Confirmation has been checked for data validity and is in the external matching queue.
- **Authenticated:** The external confirmation has been received and completely matches on ALL Key fields with an internal confirmation. An authentication message on that confirmation has sent out to the external party.
- **Cancelled**: Cancelled on behalf of the sender by the owner of the application. There is no evidence in the system that the withdrawal of the external confirmation has been initiated or even agreed upon by the sender of the confirmation!! See "exception handling" below.

IV.2.3. EXCEPTION HANDLING

A **business exception** might occur when **an incorrect confirmation has been sent**.

If the **external party wants to cancel his confirmation**, he will need to contact the owner of the application and agree with him on the withdrawal of his confirmation. The owner of the application should then be able to remove the external confirmation from the external matching queue via a graphical user interface (if and only if it has not been matched yet with an internal confirmation!). There is no evidence in the system that the withdrawal of the external confirmation has been initiated or even agreed upon by the sender of the confirmation!!

When a new version of an internal or external **confirmation** is received **on a Document ID for which a confirmation already has been received by the system**, the system should handle it according to the following scenarios.

In case of an **INTERNAL** confirmation:

- In the case that the previously received confirmation already had been "matched" and an authentication message had been sent to the other party, the new version of the confirmation should be discarded and generate **an ERROR or ALERT message** to the owner of the application allowing him to settle the issue directly with the other party and then manually override the trade status with the option to attach the latest version of the confirmation and authentication that finally had been agreed upon!
- If it has not been processed yet: the system should send the new version of the confirmation message to the other party and replace the previous version of the confirmation in the "matching" queue (if applicable).

In case of an **EXTERNAL** confirmation:

- In the case that the previously received confirmation already had been "matched" and an authentication message had been sent to the other party, the new version of the confirmation should be discarded and generate **an ERROR or ALERT message** to the owner of the application allowing him to settle the issue directly with the other party and then either to send **a REJECTION message** to the other party or to manually override the trade status with the option to attach the latest version of the confirmation and authentication that finally had been agreed upon!
- If it has not been processed yet: the system should replace the previous version of the confirmation in the "matching" queue (if applicable).

The system should also be able to handle **other exceptions** such:

- The internal or external confirmation has an **invalid format**. This confirmation should be rejected at validation.
- An external confirmation is received on a trade of which the owner of the application is nor buyer nor seller. This confirmation should be rejected at validation.
- An external or internal confirmation is received **twice**. This confirmation can be ignored.

IV.3. REPORTING LOGIC

An internal trade referred to within the EFET box can have following statuses:

- **Received**: The trade information has been sent by the ETRM system and received properly by the dispatcher within the EFET box, but has not yet been processed to decide whether a peer-to-peer or central matching process applies.
- **Peer-to-Peer Pending**: Trade will be confirmed via a peer-to-peer connection, the corresponding confirmation has been sent to the other party (and, if authentication is supposed to be sent, might be waiting in the internal matching queue). Substatuses might be identified:
    - **Authentication awaited:** meaning an authentication message from the other party is expected on the confirmation that has been sent out.
    - **Match pending:** meaning the internal confirmation is in the internal matching queue waiting to be matched with an external confirmation for the authentication message to be sent out.
    - **Match pending with possible matches:** meaning the internal confirmation is in the internal matching queue waiting to be matched with an external confirmation for the authentication message to be sent out. There has been at least one match identified with an external confirmation on the primary keys but not on all secondary keys.

- **CMS Pending**: Trade will be confirmed via a Central Matching Service and the corresponding confirmation has been sent to the service.
- **Confirmation Rejected**: The confirmation message related to that trade has been rejected by the receiver of the message because not valid.
- **Matched**: The Central Matching Service reported a match, meaning all KEY fields are certified identical to the other party confirmation.
- **Authenticated**: An authentication message has been produced internally and sent to the other party or received from the other party, meaning respectively the owner of the application agree upon the other party confirmation trade data or the other party agreed upon the internal confirmation trade data resulting in both cases in the fact that all KEY fields have been agreed upon.
- **Cancelled**: The confirmation process has been cancelled (manually via user interface?).

The reporting logic of the EFET box should cover **all internal trades** and their confirmation status whether they are processed **via a peer-to-peer connection or a central matching service.**

Note: For communication purposes between parties upon matched trades, the fields Document ID, AuthenticationID OR MatchID (when applicable), other party Document ID (when applicable) should be available on ALL reports.

Reports that should be available online and in real time:

- **Trades that have been authenticated via a peer-to-peer communication**: Full Match result including AuthenticationID, own Document ID and other party Document ID but also the full trade confirmation data from both parties.
- **Trades that have been matched via a Central Matching Service Provider**: Full Match result including MatchID, own Document ID and other party Document ID but also the full trade confirmation data from both parties.
- **Possible Peer-to-Peer Matches**: report on pending trades for which external confirmations have been found with the same primary key fields but for with the secondary key fields are not identical.
- **External Confirmations** (received, pending) **that have been received within your EFET box:** Document ID, status and full details.
- **External Confirmations having been rejected**: Other party Document ID and the detailed reason for rejection of the confirmation.
- **Internal confirmations having been rejected by the other party**: Internal Document ID and the detailed reason for rejection of the confirmation.
- **Internal confirmations having been rejected by the Central Matching Service**: Document ID and the detailed reason for rejection of the confirmation.

- **Trade overview**: list of all own trades, whether they are processed peer-to-peer or via a Central Matching Service and their current status: Document ID, ProcessType, Status.

V. OTHER CONSIDERATIONS

V.1. GRAPHICAL USER INTERFACE

The company owning the EFET box will normally require a number of **different logins** (password protected).

Profiles shall be defined to allow some logins to have certain authorisations such as cancellation of confirmations, etc… and others to only have read only access rights.

The **reports can be made highly user configurable** in terms of drill down capabilities, sorting, filtering, level of detail etc. The system should allow both reporting on paper and in electronic format.

The aim here is to facilitate the auditing of a confirmation at any stage of the matching process.

V.2. AUDIT TRAIL AVAILABILITY

A **full audit trail** of all incoming and outgoing messages preferably has to be kept for at least one year within the system. This is done using the logging module of the EFET box.

V.3. ARCHITECTURE

The EFET box shall be built on a 3-tier architecture with as

- Back-end Database
- Middle Tier Application Server
- Front-end Browser on Client

The EFET box shall be compatible with MS SQL (Version 7 or higher), Oracle (Version 8.1.x or higher) and Sybase. Communication with the database would support ODBC and Net 8.

Though the architecture should enable the user to run both backend and middle tier functions on the same machine, the possibility shall be foreseen to run the 3 layers on 3 separate servers.

Both Intel and Unix platforms shall be supported. We would typically require the EFET box to run on MS windows 2000 Server, Sun OS or Windows NT.

Graphical user interface shall be web-based and support not all but the most important web browsers like Internet Explorer, Netscape…