EUROPEAN COMMISSION
DIRECTORATE-GENERAL FOR MARITIME AFFAIRS AND FISHERIES

MEDITERRANEAN AND BLACK SEA
Integrated Fisheries Data Management

Version 1.0

INTEGRATED FISHERIES DATA MANAGEMENT PROGRAMME
PHASE 1: FISHERIES CONTROL AND MONITORING

FLUX TRANSPORTATION LAYER V1

Subject: **General principles of the FLUX transportation layer**

1. INTRODUCTION

This document summarises the general principles and system description of the FLUX Transportation layer. It also describes the general requirements for FLUX transportation software.

The intended readership is anyone who needs a general understanding of the functioning of the FLUX Transportation layer.

2. GENERAL PRINCIPLES

2.1. Main transportation layer deliverables

FLUX provides description for:

- The FLUX Envelope, one single yet universal message format that can encapsulate any business-specific message or structured data in a predictable way whatever the business system and associated data types and formats, using industry standard data representation techniques

- The FLUX transportation Protocol, a mechanism describing how to reliably deliver the FLUX Envelopes to their destination and without human intervention, leveraging state-of-the-art existing technologies (SOAP Web Services) in a sensible manner so as to as much as possible avoid interoperability issues between FLUX implementations based on different vendors' solutions.

francky.callewaert@ec.europa.eu

2.2. FLUX system types

A FLUX system is by definition any computer system that implements message transmission using FLUX Envelopes and transportation Protocol. Exactly like paper envelopes usually transit through several post offices before reaching their destination, FLUX Envelopes can also go through one or several interim forwarding nodes before reaching their final destination system.

So, one can distinguish two kinds of FLUX systems depending on what it does:

- FLUX Endpoint, a computer system that creates and sends new FLUX Envelopes on behalf of a business system (e.g. a Member State data exchange system using FLUX, Envelope originator or destination)

- FLUX Node, a computer system that merely relays FLUX Envelopes from other FLUX systems and propagates them towards their final destination (e.g. an ERS Central Node at the European Commission).

2.3. Business agnostic

Just like regular paper envelopes can contain postcard or letters of any kind indistinctly, FLUX Envelopes can contain all sorts of business data. Catch Reports, ERS Electronic Logbooks, Fleet Register Declarations, etc., are all examples of business data formats that a FLUX Envelope can contain. In order to distinguish between these contents, each business data format is assigned a unique name. These names are referred to as the Dataflow name and will be imprinted on the Envelope.

Normally, a postal service is not expected to alter or even read the contents of paper envelopes that it handles. In the same way, FLUX systems must never alter an Envelope, meaning the destination Endpoint always receives exactly the same Envelope as sent by its originator. Neither data imprinted on the Envelope itself not embedded business data inside it will ever get modified by the FLUX network.

FLUX systems are not even supposed to look at the embedded business data. They shall make no attempts at validating it. Besides, they must treat business content as sensitive non-public information. Should a FLUX system store a copy of this information somewhere, the system must be built in such a way that access to this data is restricted to authorized people only. Unless otherwise specified by Law, access shall be limited to only those people in charge of the business at the Envelope origin and destination Endpoints and to those people in charge of day-to-day operation of the FLUX system where copy of this information is stored. This information shall be used only for the purpose of recovering from FLUX errors or otherwise troubleshooting the FLUX system, and it must be deleted as soon as possible when deemed not anymore useful for these purposes.

2.4. Addressing

Unlike paper mail that requires each sender sending a letter to know its destination address in great details, a FLUX Envelope shall be described by combining only minimum information:

- the content's Dataflow name, and

- a code describing the destination address (usually a country or international organisation)

This code could be a complete FLUX address or an incomplete one, maybe only specifying the destination country or international organisation. The FLUX network will always use all information available on the Envelope to try to find out where exactly the Envelope has to go. This decision needs not be taken at one specific place in the FLUX network. Instead, all FLUX Nodes relaying the Envelope will contribute to the identification of the exact destination, using their own knowledge they have about the other FLUX Nodes near them. This mechanism is known as destination address abstraction, whereby the network can determine the final destination by itself on behalf of the sender only using the information known by the sender. It allows countries and international organisations to have selected data types (Dataflows) sent to dedicated systems (Endpoints or Nodes) if they so desire, without anyone in foreign countries be aware of it. Detailed system addresses need only be registered in nearby Endpoints and Nodes and need not be known by anyone sending Envelopes.

FLUX Nodes and Endpoints in the network all have their own precise FLUX address so that FLUX Envelopes can refer to them. In real life, addresses are made of hierarchy of names, starting with country name, then a post code, a city name, generally followed by a street name, etc. Post codes are allocated at a national level, but then each city will decide on its own street names and numbers.

Likewise, FLUX addresses are also organized according to a hierarchy of nested FLUX domains. Countries and international organisations constitute the top-level FLUX domains, inside which further sub-domains can be created by attributing a sub-domain name to selected constituting sub-entities. Each FLUX domain owner can decide if and how it wants its own domain be split, and which are those constituting (sub-)entities which shall receive a (sub-)domain name. In turn, these (sub-)entities may create further (sub-)domains, etc. The full address to any FLUX addressable entity (a.k.a. fully qualified FLUX entity address) is the list of all nested domain names, starting from the top-level domain name down to the specific entity sub-domain name.

Generally, the organisation of FLUX sub-domains will reflect the hierarchical structure of delegated powers and responsibilities inside the respective country or international organisation. For example, when a national administration delegates some of its duties (i.e. some Dataflows) to an agency, it will allocate a FLUX domain to that agency, a sub-domain of the national FLUX domain. Then, should that agency want to run multiple FLUX systems, it will create FLUX sub-domains inside its own FLUX domain.

Each addressable entity can run a Node or Endpoint, by having a FLUX system running at their attributed FLUX domain address. Their FLUX domain address then becomes a Node/Endpoint address. This allows for many Nodes and Endpoints to co-exist in the same country or international organisation, each one having a unique FLUX address.

2.5. Routing

Nodes and Endpoints need to register to the FLUX network before they can send or receive anything. They typically register to their closest parent domain among those parent domains which happen to run a Node/Endpoint. That parent Node will then create a route to the registered sub-domain Node/Endpoint, by associating the sub-domain address with the list of its delegated Dataflows. It will act as an intermediary (in-between) Node connecting its child Nodes and Endpoints (those FLUX systems running in sub-domains of its own) to the rest of the network. Destination address abstraction for those child Nodes and Endpoints will work everywhere no more than one sub-domain in charge of any given Dataflow.

When sending an Envelope, a FLUX Endpoint or Node that knows the final destination Endpoint system address corresponding to the specified Dataflow and destination code can route the Envelope directly, provided the destination Endpoint recognizes and trusts it. When this destination system address is not known to the FLUX Endpoint or Node, or when this destination does not recognize or trust the sender, or when it has been agreed that this kind of Envelope shall be routed through some intermediary (in-between) forwarding Node that meets the above requirements, then FLUX makes it possible to always have those Envelopes sent through that forwarding Node.

A forwarding Node can be made to handle either:

- all Envelopes destined to the given complete or incomplete destination address code (e.g. a country or intl. organisation) irrespective of the business content (Dataflow), or

- all Envelopes of the given Dataflow or group of Dataflows, irrespective of the destination, or

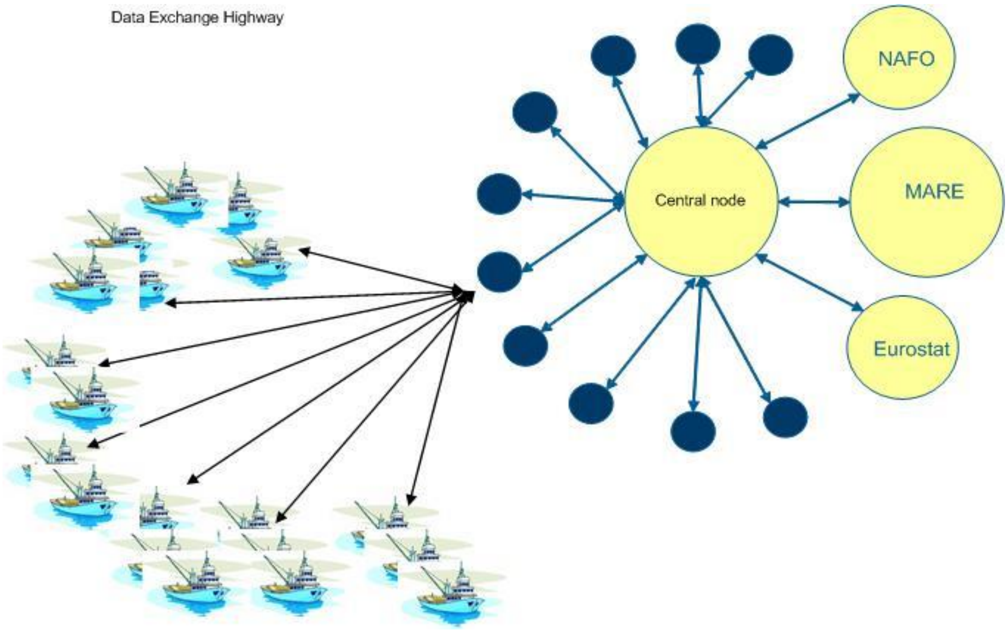
- all Envelopes, a one-do-it-all Node (e.g. the EU FLUX Central Node).

This configuration of static routes in Endpoints and Nodes will determine how the Envelope will travel towards its final destination Endpoint across the network. Using Nodes in this way makes it unnecessary for all originator Endpoints to know the detailed address of every possible destination Endpoint. Similarly, all destination Endpoints need not be able to identify and trust all possible source Endpoints. This in turn greatly simplifies configuration management in Endpoints and Nodes.

For example:

- all Endpoints in a country can be made accessible through one single Node that effectively hides to all other countries the complexity of the FLUX infrastructure in that country. Adding a new Endpoint in that country can be made fully transparent to all other countries;

- all FLUX traffic through the European Commission can be multiplexed through one single EU Central Node. Each countries' Endpoints need only know the address of this sole Central Node, and the countries' Endpoint addresses need only be known by this Central Node. Similarly, each countries' Endpoints need only be authorized in the Central Node, and the countries' Endpoint need only trust this Central Node. Adding a new country needs no address or authorization configuration change in the other countries' systems.



Still, it makes it possible to have Envelopes for some specific businesses and/or destinations to be routed differently, as sometimes required for urgency or confidentiality reasons. This is similarly to how you would sometimes select an Express mail carrier to handle the delivery of your valuable items.

Just like its paper counterparts, FLUX Envelopes will also carry the full address of the original sender. This way, it will always be possible for the network to have an Envelope returned to its sender in the event no precise destination could not be worked out for it by the network.

2.6. Reliable messaging

Reliable messaging entails three concepts:

1. never discarding any message, retrying them as necessary (or not)
2. removing messages duplicates (or not)
3. notifying of messaging status (or not)
4. preserving the message ordering (or not)

Transportation over the FLUX network implements reliable messaging with the only limitation that order of messages is not guaranteed.

2.6.1. Never discarding any message, retrying them as necessary

FLUX serves to transport business data from and to machines working unattended. FLUX Envelopes can be exchanged anytime, day or night. FLUX Endpoints and Nodes are built for 24/7 operation. The FLUX network must be dependable, meaning it must never discard any Envelope.

This means that FLUX Nodes and Endpoints must never discard an Envelope without a good reason, and when they do so they must report on what they are doing. Valid Envelopes containing business data and received from authorized sources shall be discarded by FLUX systems only when the network encounters a non-recoverable error that makes it fail to deliver the Envelope to its final destination within the business-allowed time limit. Whenever a FLUX Node gives up delivering an Envelope for some reason, it must always try to inform its originator Endpoint system by means of a dedicated FLUX error message. This message is transported through Nodes using those same reliable messaging techniques as implemented for transporting regular Envelopes (except for the acknowledge-of-receipt which is possible here). Should the delivery of such error messages also fail in an unrecoverable way, business contact persons mentioned in the Envelope must be warned by email, automatically.

Once a FLUX Endpoint or Node has successfully forwarded an Envelope to the next, it need not worry about it anymore. Endpoints and Nodes *fire and forget*.

However, sometimes a Node or Endpoint may need to fire several times. This is because temporary failures or congestion in systems or transmission networks can never be totally excluded. Therefore, Endpoints and Nodes must always be able to remember about Envelopes not yet forwarded successfully and retry forwarding them later. This way of working is called *store-and-forward*. The forwarding process in a Node or Endpoint is said to be *asynchronous* because the forwarding of an Envelope happens *after* the communication with the upstream Node or Endpoint has already ended. Consequently, if a forwarding error occurs at this stage, it can only be reported back using a new kind of message in a new Envelope, a FLUX Status Envelope.

Status Envelopes are only used to report *final* (i.e. permanent) status of a Message Envelope transmission process. Just like Message Envelopes, every Status Envelope will have a fully qualified originator address to indicate where it originated from, the location in the network where the status it reports has cropped up. Unlike Message Envelopes (those Envelopes containing a business message), the Status Envelopes don't use destination address abstraction. They don't need it, because their destination is always a fully qualified FLUX address, the originator address of a Message Envelope.

2.6.2. Removing messages duplicates

Multiple copies of an Envelope can be sent across the network, or even generated by the network itself. For example, if some business sends a Message Envelope and sees nothing back after some time, it can loses patience and decide to trigger a retry by sending a duplicate copy of the same Envelope again. This is allowed.

- Duplicate copies of a Message Envelope are always detected and removed at the final destination Endpoint, thereby ensuring that the business will only process the message at most once.

- Because a Status Envelope is only used to report the *final* status of a Message Envelope after the network has finished (or given up) transporting it, then by definition this final status will never change. Therefore, any subsequent Status Envelope referring to the same Message Envelope can only be a duplicate. So it can be ignored.

2.6.3. Notifying of messaging status

The FLUX network is designed in such a way to always report transmission errors back to the Message Envelope originator, by using a Status Envelopes. Nodes will keep on trying delivering a Status Envelope even long after its Message Envelope has timed out. At the same time, business people can be notified, as explained further down this document.

In addition, the FLUX network can also be instructed to deliver selected Message Envelopes like registered mails, whereby the Envelope originator asks the FLUX network to notify it upon successful delivery to its final destination Endpoint by means of an an *acknowledge-of-receipt* FLUX Status Envelope,. Under this scenario, FLUX implements *end-to-end Reliable Messaging*.

Please understand that this acknowledge-of-receipt is different from a business acknowledge. Only the reception is being acknowledged here. Even after an Endpoint has confirmed reception of an Envelope, it can still contain wrong business data (or sometimes even unexpected values in the Envelope itself, should that have not been verified immediately by the Endpoint) that will cause the business processing to fail. In that case, it is perfectly valid for an Endpoint to send a negative business process result back to the source Endpoint of that Envelope.

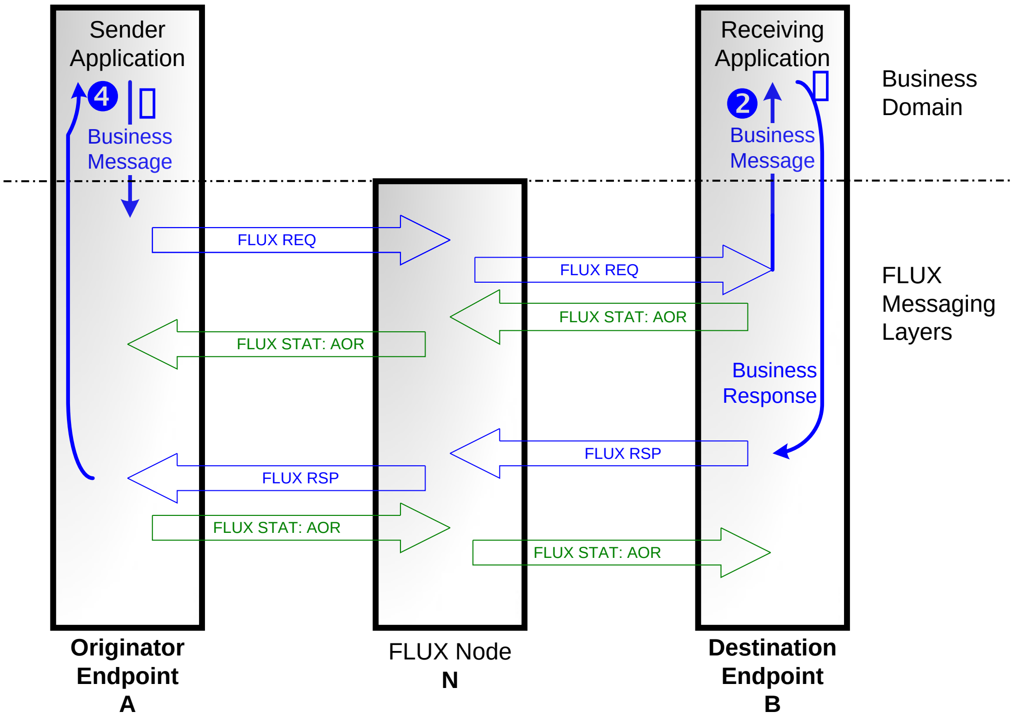
To help better understand the difference between acknowledge-of-receipt and business response, here is an example of a business request/response dialog between two Endpoints A and B for a business using acknowledge-of-receipt:

(1) A sends a FLUX Message Envelope to B (FLUX REQ, containing a business request) using a Node N as an intermediary forwarding Node.

(2) This Message Envelope gets transmitted to B. Because B is the final destination, it swiftly crafts and sends an acknowledge-of-receipt Status Envelope (FLUX STAT: AOR) to A. Let's assume this Status Envelope also goes through the same intermediary Node N. N then propagates this acknowledge-of-receipt Status Envelope back to A.

(3) When convenient for B, the business message will get processed. The business response (FLUX RSP) will be transmitted back to A in a Message Envelope. From the networks' point of view, this new Message Envelope is completely unrelated to the previous Message Envelope A to B. It can have different properties as regards acknowledging or receipt. Let's assume B also wants to receive an acknowledge-of-receipt for this Envelope.

(4) Message Envelope containing the business response from B is received by A. A confirms reception by also sending an acknowledge-of-receipt (FLUX STAT: AOR) back to B, again using its usual route through N.



Note that in this above example, B sends everything to A through the same Node N. In fact is not at all required. B could have decided to have them all go through the another Node, or using a direct communication to A. Routing at B is at B's discretion and only matters to B. For every other FLUX system it is irrelevant.

Whether or not to use acknowledge-of-receipt for business requests and/or responses is a requirement of the business. Some businesses can do without an acknowledge-of-receipt for the business request because they can afford waiting until the business response is received, and they assume that if no response is received after some time the request has simply to be retried. These businesses would then typically use acknowledge-of-receipt only for business responses, or maybe even no acknowledge-of-receipt at all. Some businesses manipulate such a high volume of messages that acknowledge-of-receipts is simply too costly to be used all the time. Instead, they will only want it for those very important business messages. In any case, each FLUX business message shall specify for itself whether to send an acknowledge-of-receipt upon reception by its final destination.

2.6.1. Not preserving message ordering

Just like with paper mail, FLUX does not guarantee that Envelopes will arrive at their destination in the same order they have been sent.

The reason is twofold, first to keep the FLUX systems as simple as possible. Second, because all FLUX Envelopes are time-stamped at the time they are created by their originator Endpoint, and this timestamp on the Envelope should make it easy for the receiving business to detect out-of-order messages and simply reject them. Alternatively, business requests could contain a sequence number so the receiver can detect out-of-order requests and put them back in the right order prior to execution.

2.7. Security

The FLUX network is built on the idea of a delegated security model, whereby every Node checks everything they can and rejecting inappropriate Envelopes and trust the other Nodes upstream are behaving identically.

FLUX systems communicate through an established set of system-to-system communications secured using industry standard solutions such as those used for Internet banking. This provides mutual identification of the systems and guarantees that Envelope cannot be intercepted or tampered with by a man-in-the-middle. FLUX systems must build on this system-to-system security to provide for network-level security, by:

- detecting and rejecting all forged Envelopes (that is, Envelopes not originating from the originator address written on them), and

- making is possible to authorize some Endpoints or Nodes only for specific business data types (Dataflows).

FLUX systems typically do this by implementing security filters, a selective authorization mechanism defining for every known upstream system (all directly connected Nodes and all locally-registered Endpoints) those Envelopes types it is allowed to send or relay (as a combination of originator address and Dataflows). In particular, forged Envelopes will be detected and rejected on the 1st Node traversed by the Envelope (on the Node where the Envelope originator Endpoint is registered). Knowing this, all other Nodes not directly connected to the originator Endpoint will trust the originator address on the Envelope is correct.

2.8. Timeouts and fall-back procedures

The FLUX network must not be allowed to retry failed transmissions forever. Also, business in real life usually requires certain data be transmitted within a defined timeframe, and should that not be possible manual fall-back procedures are usually defined, such as notifying people. In such events, notifying systems should be preferred and people should be notified only as a last resort, so as to allow for automated recover mechanisms whenever possible.

All Message Envelopes must specify a transportation deadline. They can also specify an optional list of email addresses of those people in charge of associated manual fall-back procedures. Message Envelopes will no longer propagate after the deadline, their transmission status becomes permanent (final). The FLUX network will guarantee that the originator Endpoint will somehow be notified about that final status at the latest at timeout time, either by means of a Status Envelope or by business contact people receiving a timeout email. If none of these happens, the originator may assume that the transmission is successful. If a more positive indication is needed, the originator Endpoint may ask for an acknowledge-of-receipt.

In practice, the process of notifying takes time. Besides, FLUX Nodes in the network will never be perfectly synchronized. For these reasons, in practice FLUX Nodes will give up transmission, craft and send a proper Status Envelope informing on the timeout condition slightly before the deadline, giving the Status Envelope some leeway so it has a chance of reaching the originator Endpoint before time is out. Only a Node which is still in hold of a Status Envelope at timeout time will notify business contacts by emails.

Please remember that the FLUX network sees a business request and its business response as two totally unrelated Message Envelopes. Each one will have a timeout and an email contact list as set by its respective originator. The FLUX transportation software can only detect late business responses by means of the timeout set inside the business response Envelopes, and it can only notify the sender (system or people) of the business response.. Business will have to define how to set the timeout and the list of contact emails in business response Envelopes, e.g. whether copying these data from the business request Envelope is appropriate or not.

Email addresses can also be associated with every route on a FLUX Node or Endpoint, allowing the technical staff in charge of a failing FLUX transportation system to be notified automatically. FLUX Nodes will usually also provide means (Web Sites and/or Web Services) for those notified people to find out by themselves what has happened with their Envelopes on the Node, without requiring any human intervention of the staff in charge of properly functioning Nodes. For example, in a EU Central Node scenario where a country system failure is detected by the Central Node, the EU Central Node sends automatic emails to the country's IT staff and they can use automated facilities at the EU Central Node to get an overview of the situation with the failed transmissions., all without any intervention from EU staff as long as the EU Central Node itself is working properly.

3. FLUX SYSTEM DESCRIPTION

3.1. FLUX system task list

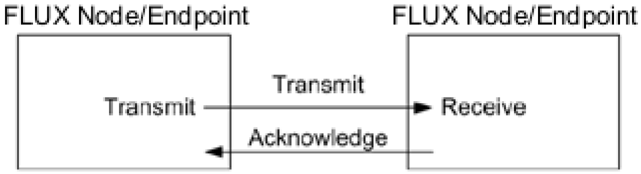
FLUX systems must be able to perform the following tasks as described in the FLUX protocol:

- Envelope any business message from one server (a registered business system in a registered country, in the European Commission or in any other registered international organisation) for transmission to another server (e.g. another system doing the same business in another registered country, European Commission or other registered international organisation). Allow multiplexing on the same FLUX "data exchange highway" Message Envelopes from several businesses and thus containing different business data formats (Dataflows). For each new Message Envelope, a new Operation Number will be assigned by its originator and imprinted on the Envelope. This number will be unique among all Message Envelopes previously sent by the same originator. Together with the originator fully qualified FLUX address, this Operation Number will uniquely identify the Message Envelope in the network. Status Envelopes will correlate to their corresponding Message Envelope by means of the same Operation Number toether with the Status Envelope fully qualified FLUX destination address (which has the same value as the Message Envelope fully qualified originator address). Please note that Business Responses will be correlated to their matching Business Request by means of some other unique identifier present in the business message. This is outside the scope of FLUX Transport.

- Determine the Internet address of the FLUX system where to send an Envelope by using only data imprinted on the Envelope itself (such as the business dataflow and the destination address code), without requiring any knowledge of business data encapsulated inside the Envelope payload. That Internet address can either be the address of the final recipient Endpoint, or it can be the address of a forwarding Node on the way and to the final destination Endpoint, to which further routing activities are delegated.

- Use the Internet for cheap data transmission and overcome the associated security concerns by using Industry-standard technical solutions typically used for addressing them (such as X.509 Digital Certificates, 2-way SSL/TLS tunnelling for encryption and mutual authentication).

- Implement Reliable Messaging at the node level, meaning each time an Endpoint or Node receives a FLUX Envelope, it must send back a synchronous FLUX Acknowledgement to confirm the FLUX Envelope has been correctly received and understood. This synchronous FLUX Acknowledgement must contain a clear status code informing on the status of the operation, either success or failure, as well as a short English text describing this status. Optionally, it can also give the fully qualified FLUX address of the receiving system as well as a list of email addresses to which missing authorizations could be asked, for example. If the received Envelope is a Message Envelope, the business content inside it is not examined at this stage, only the Envelope itself matters.



- Implement flexible End-to-End Reliable Messaging for transmission of business messages, meaning that some (not all!) synchronous acknowledgements resulting from the transmission of an Envelope containing a business message must be transmitted back to the originating business system asynchronously:

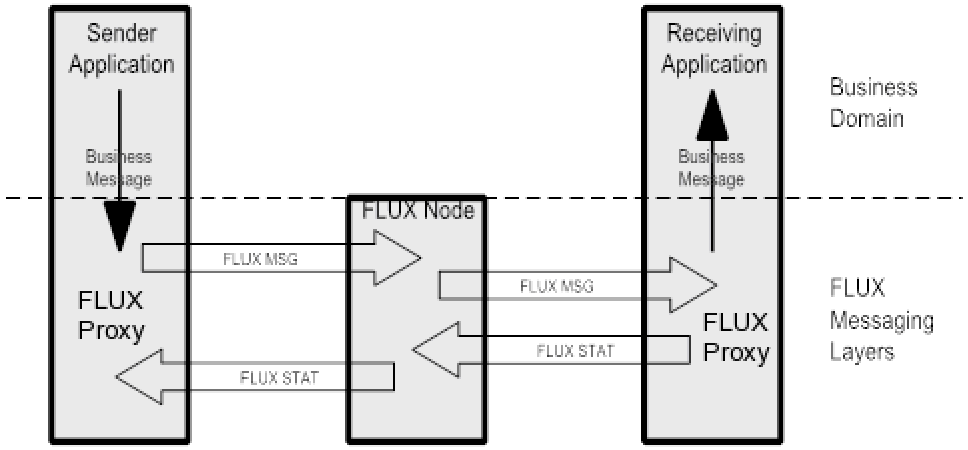
  (a) Positive acknowledgement coming from the final destination (acknowledgement-of-receipt): if the business requires it

  (b) Negative acknowledgements of permanent errors: always.

  This is achieved by encapsulating the synchronous acknowledge inside a dedicated FLUX Status Envelope and transmitted reliably, possibly going through intermediary Nodes before reaching its final destination. All properties of the Message Envelope that affect its processing on the network (such as the timeout date/time, the business contact list, etc) are copied to the Status Envelope. This way, the Status Envelope can be processed without any prior knowledge of the Message Envelope. This allows both to travel along different routes on the network. It also allows FLUX Nodes to be stateless.

  A Status Envelope always gives the *final* status of the transmission and overrides any previous or future acknowledgement about the same business message.

  Note that Positive acknowledgement coming from intermediary Nodes (Forwarding notifications) and Negative acknowledgements of temporary errors (Retrying notifications) are never notified asynchronously.



- Try to recover from temporary error conditions. Reschedule (retry) any failed message transmission (of Envelopes containing a Business Request, a Business Response or a FLUX Status message), as long as allowed by the business (by a timeout value in the message), unless it failed on permanent error (e.g. a wrong Envelope was received, or the originator is not authorized to send an Envelope for that particular business, no route back to the originator exists, no route to the destination can be found, etc).

- Allow the business to specify in the Message Envelope a deadline (timeout) after which transmission of the Message Envelope is no longer acceptable by the business. When the time is up, give up forwarding the Message Envelope and report these timeout conditions using a Status Envelope. If the problem is caused by a non-functioning FLUX Node or Endpoint, notify the people in charge of these FLUX Transport systems. Avoid flooding them with too many redundant emails..

- Give up gracefully. Whenever transmission of an Envelope or a FLUX Status Envelope cannot complete within the specified timeout, warn the people in charge by sending them emails automatically. For this, use the contact list copied from the Message Envelope to the Status Envelope. Avoid flooding them with too many redundant emails.

3.2. FLUX Envelope requirements

FLUX Envelopes are designed in such a way as to enable FLUX Transport software to perform the above-mentioned tasks. They meet the requirements below:

- Use modern technologies for structured data representation (the Extensible Markup Language notation, XML) described by a XML Schema so as to be compatible with international standards form machine-to-machine transmission adopted by the industry (SOAP Web Services). Simple SOAP protocols are used to favour interoperability.

- Contain FLUX addresses to identify the source and destination Nodes/Endpoints for the Envelope. All originator and destination addresses in all Envelopes must be fully qualified so as to identify the particular system without ambiguity, except for the destination address in a Message Envelopes which can be code corresponding to an incomplete address (usually a country or international organisation). In a Message Envelope, the originator code combines with the Dataflow name to determine the address of the next-hop Node/Endpoint towards the Message Envelope destination (that is, unless further such as Business Routing Plugins tricks are implemented in the software). This addressing scheme is flexible enough so that any country or international organisation can have multiple data formats handled by multiple Endpoints if needed, while at the same time providing destination address abstraction so that no one else needs to know about the multiple destination Endpoints. Accessing these Endpoints through a Domain Façade Node will make all these Endpoints appear as just one, ensuring that a detailed knowledge about the country's or international organisations' internal Endpoint arrangements and addresses need not be known by other countries or international organisations. Status Envelopes find their way through the network only using the fully qualified destination address, not taking the Dataflow name into account.

- Contain a mandatory parameters specified by the Message Envelope originator that instruct FLUX Nodes whether they must relay back acknowledgement-of-receipt notification. This parameter is copied from the Message Envelope to the Status Envelope.

- Contain an optional debug flag to be set in case the Envelope originator wants to be notified by other FLUX systems of the progression of the Envelope towards its destination (Forward and Retry notifications). This parameter is copied from the Message Envelope to the Status Envelope.

- Contain a mandatory transmisison timeout date/time past which FLUX must give up transmitting the Message Envelope. This parameter is copied from the Message Envelope to the Status Envelope.

- Contain an optional synchronous timeout to limit the duration of system-to-system communications. This parameter is copied from the Message Envelope to the Status Envelope.

- Contain an optional list of business contact email addresses to inform in case FLUX is unable to notify using a Status Envelope in due time. This parameter is copied from the Message Envelope to the Status Envelope.

- Add minimum overhead to the business data (e.g. be as bandwidth-efficient as possible) so as try to match as much as possible the performance of a similar business system not using FLUX.

In addition:

- Message Envelopes wrap the structured business message in a way that its business name and structure (XML namespace) is easily identifiable. Business data is inserted as-is without any conversion or transformation of any kind so as to be as seamless and efficient as possible.

- Status Envelopes wrap a synchronous FLUX Acknowledgement and can also optionally wrap a bit unformatted text message further describing the error.

4. VERSION AND HISTORY

Version	Author	Date
1.0	Matthias Petofalvi	30/07/2013
0.3	Matthias Petofalvi	29/07/2013
0.2	Matthias Petofalvi (Comments Francky Callewaert)	31/08/2012
0.15	Matthias Petofalvi	27/07/2012