The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

MISSION OPERATIONS—MESSAGE ABSTRACTION LAYER BINDING TO ZMTP TRANSPORT

DRAFT RECOMMENDED STANDARD

CCSDS 524.4-R-1

RED BOOK

November 2017

The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

MISSION OPERATIONS—MESSAGE ABSTRACTION LAYER BINDING TO ZMTP TRANSPORT

DRAFT RECOMMENDED STANDARD

CCSDS 524.4-R-1

RED BOOK

November 2017

AUTHORITY

Issue:	Red Book, Issue 1
Date:	November 2017
Location:	Not Applicable

(WHEN THIS RECOMMENDED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the e-mail address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
E-mail: secretariat@mailman.ccsds.org

STATEMENT OF INTENT

(WHEN THIS RECOMMENDED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF INTENT:)

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

- Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.
- Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:
  - -- The **standard** itself.
  - -- The anticipated date of initial operational capability.
  - -- The anticipated duration of operational service.
- Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

The intended use for this document is to allow the implementation of a protocol layer that binds the Mission Operations (MO) service framework to the ZMTP Transport using the Split Binary Encoding. This document assumes that the reader is familiar with the MO concepts, especially the Message Abstraction Layer (MAL).

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in the *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be sent to the CCSDS Secretariat at the e-mail address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

PREFACE

This document is a draft CCSDS Recommended Standard. Its 'Red Book' status indicates that the CCSDS believes the document to be technically mature and has released it for formal review by appropriate technical organizations. As such, its technical contents are not stable, and several iterations of it may occur in response to comments received during the review process.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document's technical content.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 524.4-R-1	Mission Operations—Message Abstraction Layer Binding to ZMTP Transport, Draft Recommended Standard, Issue 1	November 2017	Current draft

CONTENTS

CONTENTS (continued)

Section Page

Figure

Table

1 INTRODUCTION

1.1 PURPOSE

This Recommended Standard defines the binding between the Mission Operations (MO) Message Abstraction Layer (MAL) specified in reference [1] and the ZeroMQ Message Transport Protocol (ZMTP) specified in reference [2]. This binding allows MO Services to use ZMTP as messaging technology in all situations where this may be required.

This Recommended Standard specifies how best to use ZeroMQ's interaction patterns to implement the MAL concepts. It uses a subset of the ZeroMQ interaction patterns, but it is not intended to allow the mapping of the MAL concepts to any of these present or future patterns.

The encoding for MAL data types used in the message header is specified in reference [3]. The encoding of the message body itself is not specified in this Recommended Standard. Any MAL encoding, specified in other books, can be used for encoding the body of the messages when adopting the MAL to ZMTP binding, specified in this book.

1.2 SCOPE

The scope of this Recommended Standard is the specification of the binding in terms of technology mapping to the ZeroMQ Message Transport Protocol of:

a) MAL message;

b) MAL Transport Interface.

The MAL Blue Book (reference [1]) specifies the MAL protocol in an abstract way, i.e., without defining the concrete Protocol Data Units (PDUs). The MAL binding to ZMTP Transport protocol specifies:

a) a complete and unambiguous mapping of the MAL message to a binary PDU to be transmitted over ZMTP;

b) a complete and unambiguous mapping of the MAL transport interface to the ZMTP interface.

This Recommended Standard does not specify:

a) individual implementations or products;

b) the implementation of entities or interfaces within real systems.

In a concrete deployment, on-the-wire interoperability between Application Layer MO Service consumer and provider will be achieved by encoding the abstract MAL messages in the concrete encoding and transmitting them by means of ZMTP PDUs, as defined in this Recommended Standard.

1.3 APPLICABILITY

This Recommended Standard specifies a mapping to a concrete communication protocol that enables different implementations of the MO service framework (see 2.2) to interoperate through ZMTP communication protocol.

1.4 RATIONALE

CCSDS MO services are Application Layer services, which are specified in an abstract, implementation and communication agnostic manner in terms of the MAL (Message Abstraction Layer).

In a concrete deployment scenario (instantiation of the abstract MO services in a concrete set of technologies) on-the-wire interoperability is achieved by agreeing on a single, specific concrete encoding and a concrete communication protocol for the exchange of the messages between the service provider and service consumer.

The goal of this Recommended Standard is to specify how to translate the abstract MAL message model in an unambiguous way into a concrete message exchange protocol based on ZMTP.

1.5 DOCUMENT STRUCTURE

This document is organized as follows:

a) section 1 presents the purpose, scope, applicability, and rationale, and lists the definitions, conventions, and references used throughout this Recommended Standard;

b) section 2 presents an overview of the MAL binding to ZMTP transport protocol in relation with the MO service framework;

c) section 3 specifies the MAL binding to ZMTP transport protocol, by providing an unambiguous mapping of the MAL messages to the ZMTP PDUs;

d) section 4 specifies the mapping of the MAL transport interface to the ZMTP interface.

1.6 DEFINITIONS

protocol: The set of rules and formats (semantic and syntactic) used to determine the communication behaviour of a protocol layer in the performance of the layer functions. The state machines that operate and the PDUs that are exchanged specify a protocol.

protocol layer: The implementation of a specific protocol. It provides a service access point to layers above and uses the service access point of the layer below.

service access point, SAP: The point at which one layer's functions are provided to the layer above. A layer may provide protocol services to one or more higher layers and use the protocol services of one or more lower layers.

1.7 NOMENCLATURE

1.7.1 NORMATIVE TEXT

The following conventions apply for the normative specifications in this Recommended Standard:

a) the words 'shall' and 'must' imply a binding and verifiable specification;

b) the word 'should' implies an optional, but desirable, specification;

c) the word 'may' implies an optional specification;

d) the words 'is', 'are', and 'will' imply statements of fact.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

1.7.2 INFORMATIVE TEXT

In the normative sections of this document, informative text is set off from the normative specifications either in notes or under one of the following subsection headings:

– Overview;

– Background;

– Rationale;

– Discussion.

1.8 BIT NUMBERING CONVENTION

In this document, the following convention is used to identify each bit in an *N*-bit field. The first bit in the field to be transmitted (i.e., the most left justified when drawing a figure) is defined to be 'Bit 0'; the bit following is defined to be 'Bit 1', and so on up to 'Bit *N*–1'. When the field is used to express a binary value (such as a counter), the Most Significant Bit (MSB) shall be the first transmitted bit of the field, i.e., 'Bit 0'.

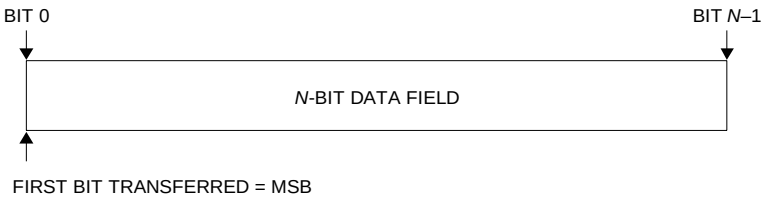

Figure 1-1: Bit Numbering Convention

In accordance with modern data communications practice, spacecraft data fields are often grouped into eight-bit 'words' which conform to the above convention. Throughout this Recommended Standard, the following nomenclature is used to describe this grouping:

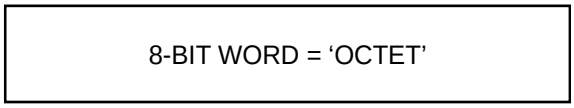

Figure 1-2: Octet Convention

By CCSDS convention, all 'spare' or 'unused' bits shall be permanently set to value 'zero'.

1.9 REFERENCES

The following publications contain provisions which, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All publications are subject to revision, and users of this Recommended Standard are encouraged to investigate the possibility of applying the most recent editions of the publications indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications.

NOTE – A list of informative references is provided in annex G.

[1] *Mission Operations Message Abstraction Layer*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 521.0-B-2. Washington, D.C.: CCSDS, March 2013.

[2] "23/ZMTP—ZeroMQ Message Transport Protocol." ZeroMQ RFC. https://rfc.zeromq.org/spec:23/ZMTP/.

[3] *Mission Operations—MAL Space Packet Transport Binding and Binary Encoding*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 524.1-B-1. Washington, D.C.: CCSDS, August 2015.

[4] *IEEE Standard for Floating-Point Arithmetic*. 2nd ed. IEEE Std. 754-2008. New York: IEEE, 2008.

[5] F. Yergeau. *UTF-8, a Transformation Format of ISO 10646*. STD 63. Reston, Virginia: ISOC, November 2003.

[6] *Time Code Formats*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 301.0-B-4. Washington, D.C.: CCSDS, November 2010.

[7] *Data Elements and Interchange Formats—Information Interchange—Representation of Dates and Times*. 3rd ed. International Standard, ISO 8601:2004. Geneva: ISO, 2004.

2 OVERVIEW

2.1 GENERAL

This Recommended Standard allows MO services defined in terms of the MAL to interoperate across an end-to-end communication link using a normative binding of the MAL abstractions to the ZeroMQ Message Transport Protocol for exchanging messages. This is of particular interest for MO services, for which the service provider and consumer are both deployed on the ground, for instance when the MO service provider is located in a Mission Control Centre and the consumer in the Science Control Centre. The messages that provider and consumer exchange to implement the MO services are encoded in PDUs and carried via ZMTP, which acts as a Message Layer mapping. This can run on various protocols underlying ZMTP.

To achieve this goal, this Recommended Standard provides a mapping of the MAL transport interface, the MAL abstract message specification (reference [1]) to the ZMTP protocol stack (reference [2]).

The MAL Blue Book (reference [1]) defines an abstract transport interface as a set of request and indication primitives. The mapping to a concrete transport protocol specifies how these primitives are provided according to the rules and requirements of that particular messaging protocol.

The mapping of MAL to a concrete communication protocol translates the MAL message model into one or several protocol-specific PDUs. MAL messages are composed of two conceptual segments, the MAL header and the MAL body. The header of the MAL message contains the metadata and is mapped to the protocol-specific header encodings. The body of the MAL message can, however, be encoded using an encoding of choice, which fits best the requirements of a particular deployment. To give a concrete example, when using the MAL to ZMTP protocol binding, which is specified in this book, the body of the MAL messages can be encoded using the binary encoding specified in reference [3]. For a different deployment using the very same MAL to ZMTP protocol binding, the body of the messages can be encoded using a different encoding.

Full interoperability of services (the so-called on-the-wire interoperability) is achieved if the same MAL to transport protocol binding and the same encoding for the body of the MAL messages are used by the service provider and the service consumer. Alternatively, a bridge must be used to translate from one binding/encoding to another (cf. reference [G1]).

The diagram shown in figure 2-1 presents the set of standards documentation in support of the Mission Operations Services Concept. This MAL binding to ZMTP Transport Protocol book belongs to the technology mappings documentation.

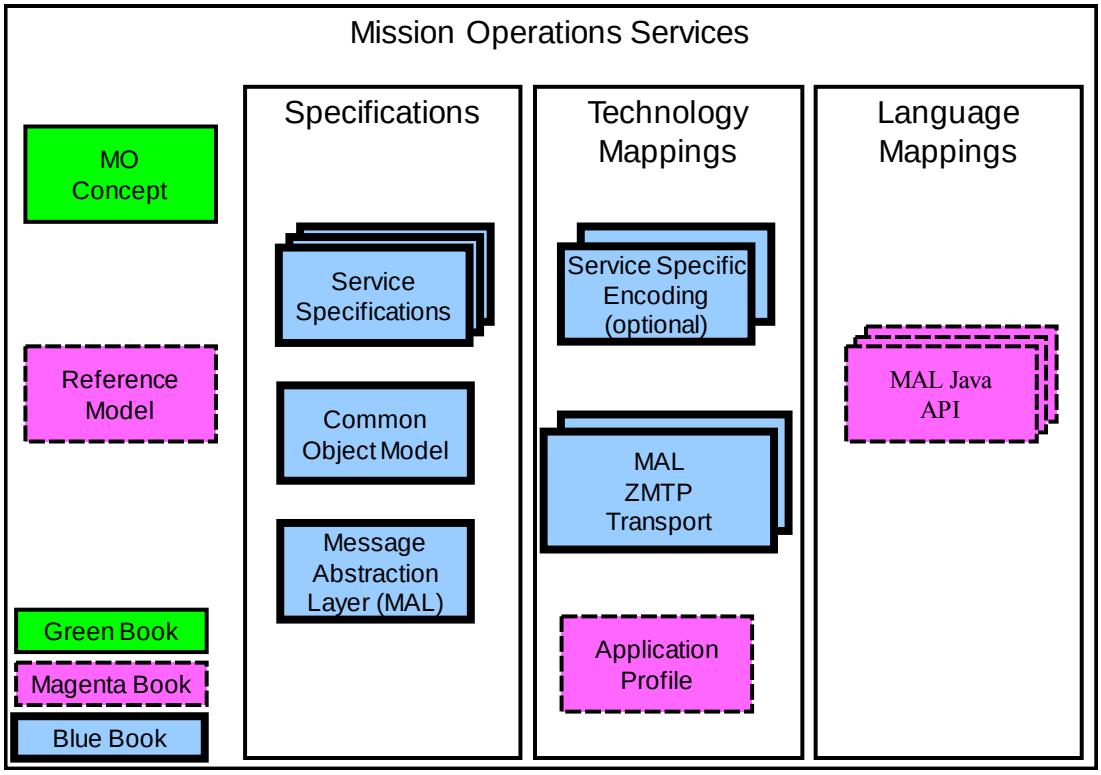

Figure 2-1: Mission Operations Services Concept Document Set

2.2 MO SERVICE FRAMEWORK OVER ZMTP

The CCSDS Spacecraft Monitoring & Control (SM&C) working group has developed a concept for an MO service framework, which follows the principles of service-oriented architectures. The framework defines two important aspects: the first is a protocol for interaction between two separate entities; the second is a set of common services providing functionality shared by most of the MO services. An overview of this framework is presented in figure 2-2.

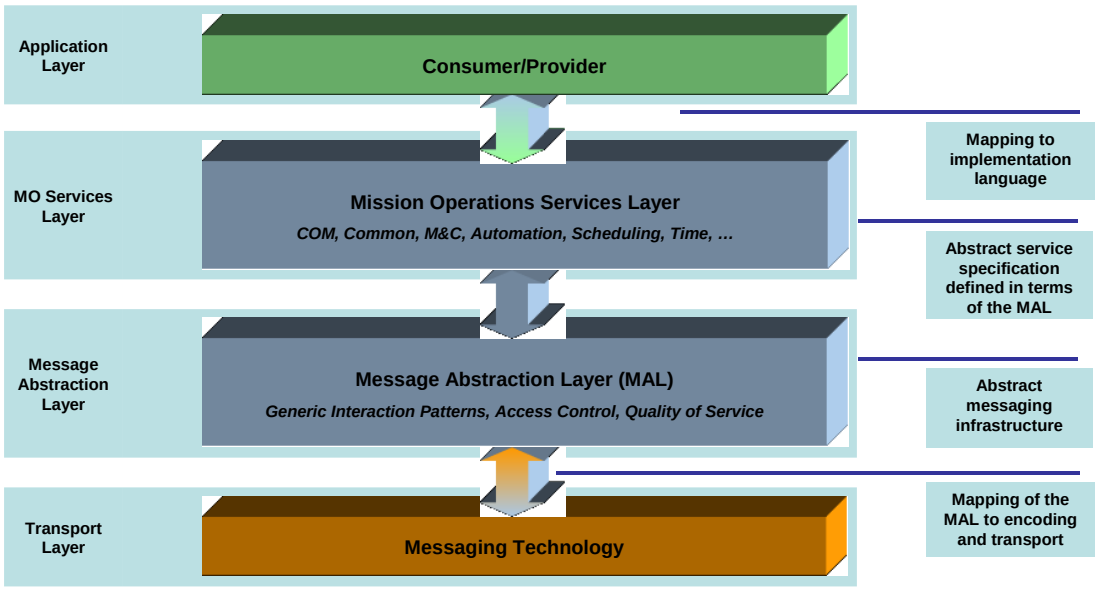

Figure 2-2: Overview of the MO Service Framework

This Recommended Standard specifies:

a) how the specific technology shall be used;

b) how any transmission errors or issues shall be communicated to higher layers;

c) how all underlying Data Link or Network Layer issues shall be handled;

d) the physical representation of the MAL messages necessary to constitute the operation templates;

e) the mapping of the message structure rules for that technology.

It does not specify:

a) individual application services, implementations, or products;

b) the implementation of entities or interfaces within real systems;

c) the methods or technologies required to acquire data;

d) the management activities required to schedule a service;

e) the representation of any service-specific PDUs;

f) the encoding of the MAL data types for the message body.

The MAL Blue Book (reference [1]) groups all the interfaces to the Transport Layer in a single place called the MAL transport interface (subsection 3.7 of reference [1]). Thanks to

this, only the MAL transport interface needs to be mapped to the ZMTP protocol, without the need to map the entire MAL Blue Book.

Figure 2-3 expands the previous figure (figure 2-2) by presenting the MAL to ZMTP transport protocol binding layer in the MO service framework stack and highlighting the various interfaces and their main primitives. It also shows that the mapping of the MAL transport interface to the ZMTP layer requires the insertion of a layer in between. This layer is called the MAL ZMTP binding. It is responsible for the translation of the abstract MAL message to the MAL ZMTP PDU transferred through concrete ZMTP segments.

The protocol stack represented in figure 2-3 is conceptual. It can be implemented in various ways. For example, an implementation of the stack may, for performance reasons, merge the MAL layer and the MAL ZMTP Transport Layer into a single layer called 'MAL over ZMTP'.

The names of the main interfaces used and implemented by each layer are given by figure 2-3. The main primitives are shown for each interface:

a) the primitives for every operation provided by an MO service;

b) the primitives for every interaction pattern provided by MAL;

c) the primitives for transmitting and receiving a single MAL message or multiple MAL messages;

d) the primitives for transmitting and receiving data from the ZMTP channel.

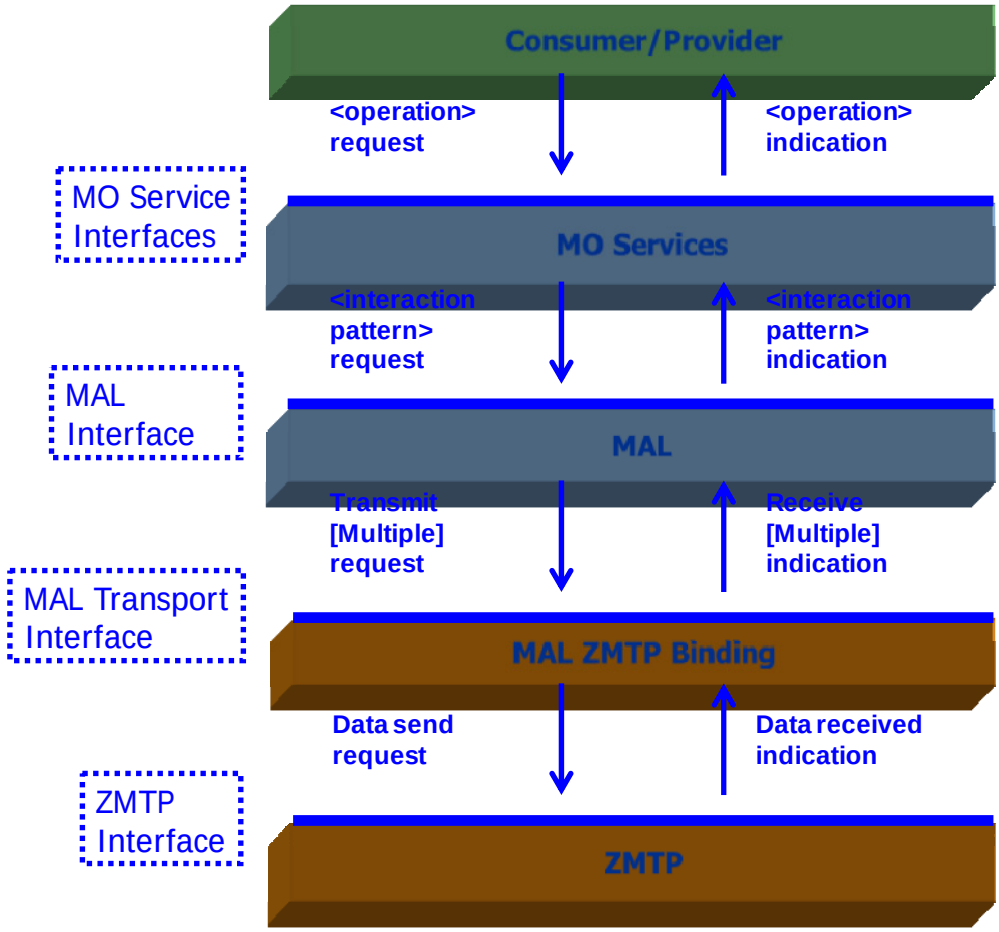

Figure 2-3: MO Service Framework above ZMTP

2.3 TYPICAL USE

Possible uses of the MAL binding to ZMTP transport protocol may be between MO entities (service consumer and provider) operating on ground, for example:

a) ground applications deployed on the same machine or interacting over a local area network using ZMTP;

b) ground components interacting over a wide area network;

c) mobile applications consuming MAL services over wireless networks.

A typical deployment is illustrated in figure 2-4. In this example, the MO service framework is used only by the end nodes: a ground end node (e.g., in a mission control centre) and another ground end node (e.g., in a science/payload control centre).

Figure 2-4 shows how the abstract MO stack is implemented on both end nodes. More specifically, the figure shows what components are deployed, how they are related to the

abstract stack (the five layers in the background), and what Application Programming Interface (API) and Service Access Point (SAP) are used.

The first concrete PDU is produced at the binding level as the result of the mapping of the MAL message to ZMTP.

The lower protocol layers are not represented.

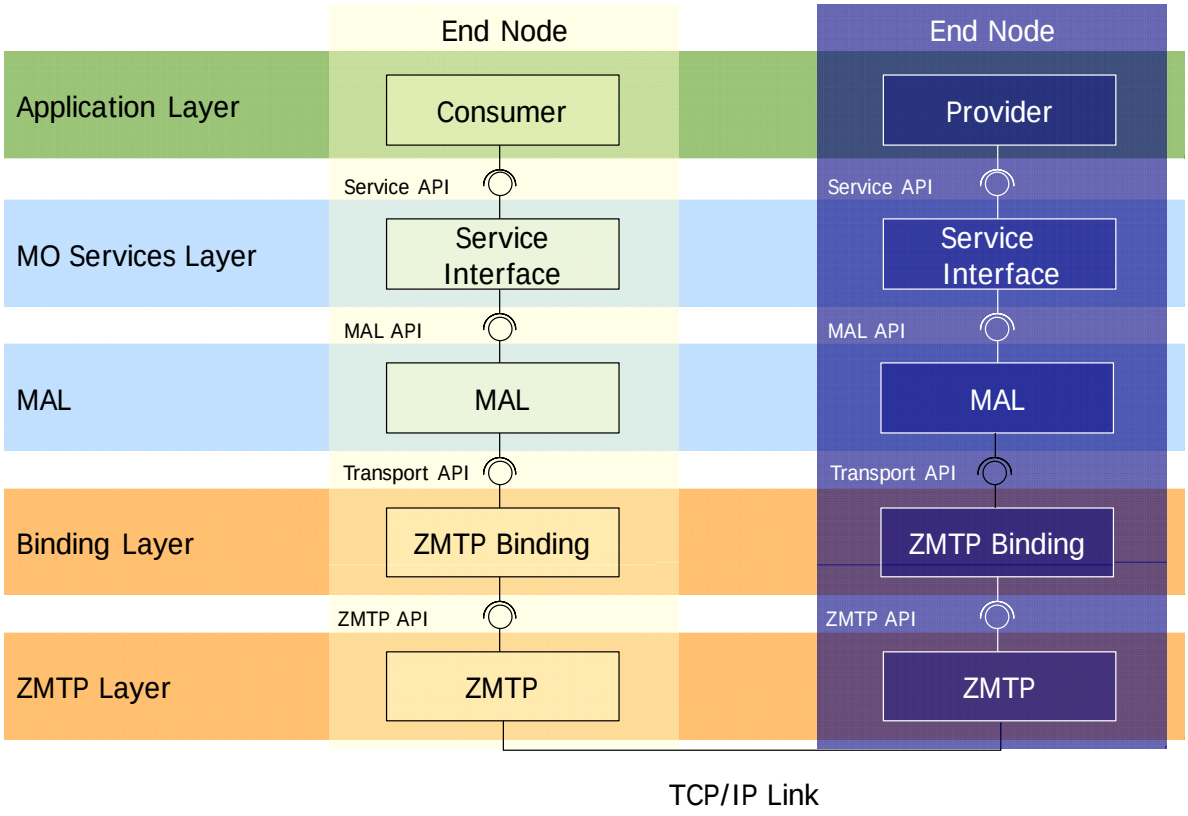

Figure 2-4: Typical Deployment of the MAL ZMTP Transport

2.4 MAL MESSAGE MAPPING

2.4.1 MAPPING TO ZMTP

The ZeroMQ Message Transport Protocol (ZMTP) is a Transport Layer protocol for exchanging messages between peers over a connected protocol. It defines the ZMTP Message as the base communication unit. The ZMTP Message itself consists of one or more frames sent over the data link. This Recommended Standard introduces a MAL ZMTP PDU, which is delivered using a ZMTP Message. Therefore each field of the MAL message needs to map to a field of this binary PDU.

The MAL message header and body are encapsulated as the payload of the ZMTP protocol.

Figure 2-5 illustrates the mapping of the MAL message to the MAL ZMTP PDU transmitted over ZMTP. Most of the MAL message fields are mapped according to a one-to-one equivalence. In this case the original MAL header field name is kept and the background colour is blue. However, the following fields require a more complex mapping: the MAL header fields 'Interaction Type' and 'Interaction Stage' are mapped to the field 'SDU Type'; the background colour is purple.

The MAL header fields 'Authentication Id', 'Timestamp', 'Priority', 'Domain', 'Network Zone', and 'Session Name' are optional: their presence in the header is specified by the QoS properties defined in annex C. These fields are mapped to two fields: a presence flag that indicates whether or not the value is encoded in the MAL ZMTP PDU header, and a field that gives the value in case it is encoded; the background colour is green.

A field named 'Version Number' is introduced as first header field of the MAL ZMTP PDU defined by this Recommended Standard: the purpose of this field is to allow future evolutions of the MAL ZMTP PDU header as defined by this version of the Recommended Standard. The background colour is red.

In order to allow flexibility in the selection of the encoding formats to be used for a MAL message body, this Recommended Standard does not prescribe a mandatory encoding, but introduces an additional field in the MAL TCP/ZMTP protocol data unit header. This value is mapped to two fields:

a) an unsigned 2-bit Integer 'Encoding Flag':
   - 0 : Fixed Binary,
   - 1 : Variable length Binary,
   - 2 : Split Binary,
   - 3 : Use the optional Extended Encoding Id field;
b) an optional header field, 'Extended Encoding Id', which identifies which encoding format was used to encode the MAL message body; the background colour is red.

The MAL header fields cannot be NULL, especially the field 'Transaction Id', even in MAL.

The MAL header fields cannot be NULL, especially the field 'Transaction Id', even in MAL messages whose interaction type is SEND. The MAL header fields are encoded using the fixed-length encoding specified in reference [3] in order to allow direct offset-based references to the header fields.

Finally, the MAL message body field and its equivalent MAL ZMTP PDU data field have a grey background.

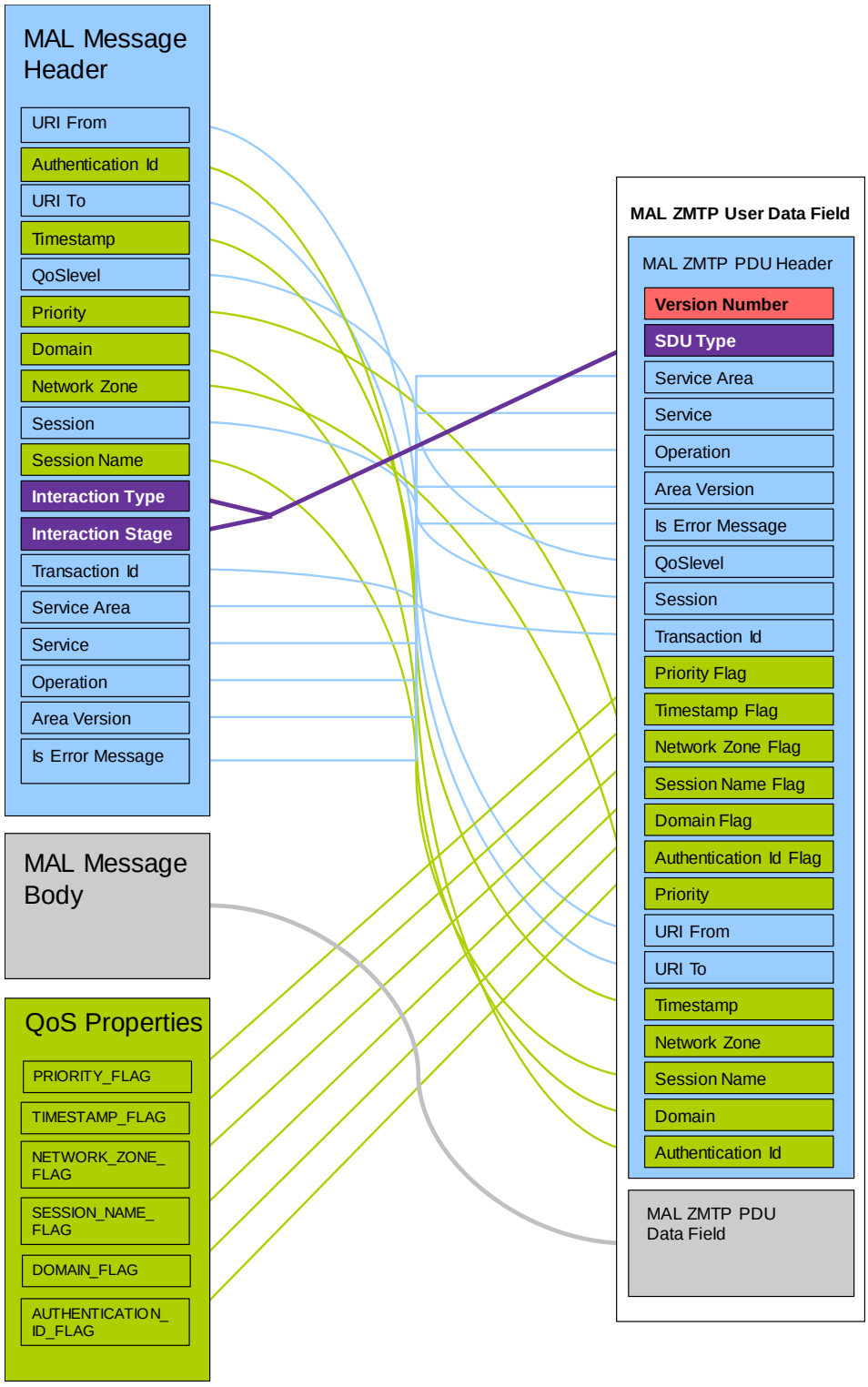

Figure 2-5: MAL Message Mapping to ZMTP

The reasons the MAL header fields are reordered when mapped to the MAL ZMTP PDU transmitted over ZMTP are given in 3.1.

2.4.2 MAPPING CONFIGURATION PARAMETERS

This Recommended Standard defines parameters that allow configuring and optimizing the MAL message mapping and the format of the MAL ZMTP PDU transmitted via ZMTP. Those parameters are called Mapping Configuration Parameters (MCP). They are defined in annex B.

The MCPs are either mission-specific or application-process-specific. They are managed parameters, defined by some out-of-band agreement.

The MCPs are needed to fully specify the encoding format. These are managed parameters to avoid the cost of additional configuration fields that must be dynamically encoded in the ZMTP PDU and interpreted at decoding time. The following encoding configuration options can be customized:

a) time code formats;

b) default values of MAL header fields that are not encoded in the MAL ZMTP PDU, but that are assigned at decoding time to the resulting MAL header.

MCPs must be exchanged out of band between the provider and the user as a separate exchange of configuration information. This could, for example, be done by email, or through a common registry like the Space Assigned Numbers Authority (SANA). The way MCPs are transmitted is not specified here.

2.4.3 MAPPING DIRECTORY

This Recommended Standard defines a mapping directory to allow to optimize the MAL message mapping and the format of the MAL ZMTP PDU transmitted via ZMTP. The primary objective of the Mapping Directory is to provide a numerical equivalent to frequently used strings in the communication. This numerical equivalent will be called a Mapping Directory Key (MDK).

The content of the Mapping Directory is needed to fully specify the encoding format. The following fields of the MAL header are concerned by an encoding as an MDK: URI From, URI To, Network Zone, Session Name, and the subdomain parts of the Domain field.

The content of the Mapping Directory is mission specific. It must be exchanged out of band between the provider and the user as a separate exchange of information. The way this content is transmitted is not specified here.

2.4.4 MAPPING SPECIFICATION

A simple tabular notation is used to specify the format of the mapping result, i.e., the MAL ZMTP PDU header. This tabular notation is composed of three levels:

a) the name of each field;

b) the encoding format of each field, as defined in 3.3.2:

   1) the length of the format is put in brackets, either in bits or octets;

   2) the length can be variable;

   3) if the value to encode is directly given in binary format, then the encoding format is called ‘Binary value’;

c) the condition, or the number of times the field is encoded in, or the static value to be assigned to the field.

The variable length of a field can be caused by an encoding format that is statically defined for a given mission, e.g., the time code formats. A variable length can also result from an encoding format whose length is inherently variable, like:

a) List;

b) String;

c) Varint.

The List and String formats contain a length field as specified respectively in subsections 5.5 and 5.21 of reference [3].

The Varint, as defined by the Binary Encoding format, allows encoding of an integer using a number of octets that depends on the integer value. Each encoded octet begins with a continuation bit (Bit ‘0’) indicating whether or not there are more octets to decode. The advantage is that the number of octets required to encode small integer values is reduced. The drawback is that big integer values require more octets to be encoded. Depending on its value, a 4-octet integer can be encoded with 1, 2, 3, 4, or 5 octets. In case of signed integers, the Binary Encoding adopts the so-called ‘zig-zag encoding’. In this technique, negative numbers are mapped onto positive numbers so that values with a small absolute value have a small Varint encoded value. The signed and unsigned Varint formats are specified in reference [3].

2.4.5 COMPLETE MAPPING

The MAL message mapping completeness is ensured by the following conditions:

a) every MAL data type is mapped;

b) every MAL message field is mapped;

c) every mandatory ZMTP field is assigned.

Moreover, the translation from a MAL message to its binary form is reversible. No information is lost in the translation from a MAL message to its binary form.

2.5 MAL TRANSPORT INTERFACE MAPPING

The mapping of the MAL transport interface requires specifying the expected behaviour for each of the MAL transport primitives. Three types of behaviour are defined:

a) a MAL transport request initiating a ZMTP request by sending a MAL ZMTP PDU split over a set of ZMTP frames, and returning a reply;

b) a ZMTP indication initiating a MAL transport indication when receiving a MAL ZMTP PDU;

c) a MAL transport request returning a reply without calling the ZMTP layer (for example SUPPORTEDQOS or SUPPORTEDIP request).

The MAL transport mapping is complete as all the primitives are mapped. Moreover, the behaviour of each primitive is fully specified.

3 MAL MESSAGE MAPPING

3.1 OVERVIEW

This section specifies how the MAL message header, body, and QoS properties are mapped to the MAL ZMTP PDU transmitted over ZMTP.

Table 3-1 is taken from reference [1] and provides the full list of fields in the MAL message header.

Table 3-1: MAL Message Header Fields

Field	Type	Value
URI From	URI	Message Source URI
Authentication Id	Blob	Source Authentication Identifier
URI To	URI	Message Destination URI
Timestamp	Time	Message generation timestamp
QoSlevel	QoSLevel	The QoS level of the message
Priority	UInteger	The QoS priority of the message
Domain	List<Identifier>	Domain of the message
Network Zone	Identifier	Network zone of the message
Session	SessionType	Type of session of the message
Session Name	Identifier	Name of the session of the message
Interaction Type	InteractionType	Interaction Pattern Type
Interaction Stage	UOctet	Interaction Pattern Stage
Transaction Id	Long	Unique to consumer
Service Area	UShort	Service Area Identifier
Service	UShort	Service Identifier
Operation	UShort	Service Operation Identifier
Area version	UOctet	Area version
Is Error Message	Boolean	‘TRUE’ if this is an error message; else ‘FALSE’

The MAL message header is mapped to MAL ZMTP PDU header.

The format of the MAL ZMTP PDU header is shown in table 3-2. All the fields of the header are encoded using the encoding format specified in 3.3.2.

The MAL ZMTP PDU to be delivered over ZMTP defined by this Recommended Standard is composed of a header and a data field.

The header is comprised of the fields ‘Version Number’, ‘SDU Type’, ‘Service Area’, ‘Service’, ‘Operation’, ‘Area version’, ‘Is Error Message’, ‘QoSlevel’, ‘Session’, ‘Transaction Id’, ‘Encoding Id Flag’, ‘Priority Flag’, ‘Timestamp Flag’, ‘Network Zone Flag’, ‘Session Name Flag’, ‘Domain Flag’, ‘Authentication Id Flag’, ‘Extended Encoding Id’, ‘Priority’, ‘URI From’, ‘URI To’, ‘Timestamp’, ‘Network Zone’, ‘Session Name’, ‘Domain’, and ‘Authentication Id’. Two spare bits are placed between the ‘Transaction Id’ field and the ‘Priority Flag’ field.

The ordering and structure of the MAL ZMTP PDU header fields is justified as follows:

a) the field ‘Version Number’ needs to come first because it identifies the header as defined by this Recommended Standard;

b) the fields ‘SDU Type’, ‘Service Area’, ‘Service’, ‘Operation’, ‘Area Version’, and ‘Is Error Message’ identify the interaction that occurs between the source and destination peers, i.e., the interaction type, the current stage of the interaction, the service that is used, its version, and the operation that is invoked; this information should come first after the header field ‘Version Number’;

c) the field ‘SDU type’ is inserted after the field ‘Version Number’ in order to reach an octet boundary;

d) the fields ‘QoSlevel’ and ‘Session’ are inserted after the field ‘Is Error Message’ and are encoded respectively using 3 bits and 4 bits, in order to reach an octet boundary (1 + 3 + 4 = 8);

e) the field ‘Transaction Id’ refines the previous information about the interaction: it identifies the current interaction occurrence;

f) the fields ‘Encoding Id Flag’, ‘Priority Flag’, ‘Timestamp Flag’, ‘Network Zone Flag’, ‘Session Name Flag’, ‘Domain Flag’, and ‘Authentication Id Flag’ are grouped together in order to reach an octet boundary;

g) the fields ‘URI From’, ‘URI To’, ‘Extended Encoding Id’, ‘Priority’, ‘Timestamp’, ‘Network Zone’, ‘Session Name’, ‘Domain’, ‘Authentication Id’ are located at the end of the header, in order to fix the length of the initial part of the header.

The format of the MAL ZMTP PDU header is shown in table 3-2. All the fields of the header are encoded using the encoding format specified in 3.3.2. For the purpose of the header encoding, the configuration property VARINT_SUPPORTED specified by reference [3] must be considered as set to TRUE.

Optional fields are specified out of band, with a mapping configuration parameter as defined by annex B.

Variable-length fields are delineated as follows:

a) the formats ‘Blob’, ‘Identifier’, and ‘List’ contain a length field as specified respectively in subsections 5.7, 5.12, and 5.5 of reference [3];

b) the format ‘String’ contains a length field as specified in subsection 5.21 of reference [3];

c) the format ‘UInteger’ is encoded as a 4-octet Unsigned Varint, as specified in subsection 5.18 of reference [3];

d) the format ‘Optional MDK’ contains a flag specifying the actual encoding as an MDK, and possibly a length field (Optional MDK format is defined in 3.3.2.2).

The QoS properties defined by annex C allow setting the values of the header flags, which cannot be deduced from the MAL header fields.

The mapping of the MAL message is composed of the following specifications:

a) the URI format to be applied to the MAL header fields ‘URI From’ and ‘URI To’;

b) the mapping of the MAL header to the header of the MAL ZMTP PDU defined by this Recommended Standard;

c) the mapping of the MAL message body to the body of the MAL ZMTP PDU defined by this Recommended Standard.

This Recommended Standard does not prescribe the encoding formats to be used for the encoding of the MAL message body.

Table 3-2: MAL ZMTP Protocol Data Unit Header Format

Version Number	SDU Type	Service Area	Service	Operation	Area Version	Is Error Message	QoSlevel	Session	Transaction Id	Encod-ing Id Flag	Priority Flag	Time-stamp Flag	Network Zone Flag
Binary value (3 bits)	Unsigned 5-bit Integer (5 bits)	Unsigned 16-bit Integer (16 bits)	Unsigned 16-bit Integer (16 bits)	Unsigned 16-bit Integer (16 bits)	Unsigned 8-bit Integer (8 bits)	Binary value (1 bit)	Unsigned 3-bit Integer (3 bits)	Unsigned 4-bit Integer (4 bits)	Unsigned 64-bit Integer (64 bits)	Unsigned 2-bit Integer (2 bits)	Binary value (1 bit)	Binary value (1 bit)	Binary value (1 bit)

Session Name Flag	Domain Flag	Authen-tication Id Flag	Authen-tication Id Flag	URI From	URI To	Extended Encoding Id.	Priority	Timestamp	Network Zone	Session Name	Domain	Authenticatio n Id
Binary value (1 bit)	Binary value (1 bit)	Binary value (1 bit)	Binary value (1 bit)	Optional MDK (var., mult. of octet)	Optional MDK (var., mult. of octet)	Unsigned 8-bit Integer (8 bits)	UInteger (var. mult. of octet)	Time (var., mult. of octet)	Optional MDK (var., mult. of octet)	Optional MDK (var., mult. of octet)	List <Optional MDK> (var., mult. of octet)	Blob (var., mult. of octet)
						If 'Encoding Id. Flag' is '3'	If 'Priority Flag' is '1'	If 'Timestamp Flag' is '1'	If 'Network Flag' is '1'	If 'Session Flag' is '1'	If 'Domain Flag' is '1'	If 'Authentication Id Flag' is '1'

3.2 URI FORMAT

3.2.1 The format of the MAL header fields 'URI From' and 'URI To' shall comply with the following rules:

NOTE – The following statements are about the MAL abstraction called URI and not about how it is mapped to the MAL ZMTP PDU.

a) the URI scheme name shall be 'malzmtp';

b) the scheme name shall be followed by a colon separator ':' and a double slash '//';

c) the double slash shall be followed by the IP address, using a format depending on the selected Internet Protocol version;

d) if version 6 of the Internet Protocol is used, the IP address shall be represented as eight groups of four hexadecimal digits separated by colons ':'. The IP address shall be enclosed in square brackets '[' and ']';

e) if version 4 of the Internet Protocol is used, the IP address shall be represented in dot-decimal notation;

f) the IP address shall be followed by a colon separator ':' and the TCP port number, an integer represented in decimal;

g) the TCP port number shall be a positive integer, excluding zero, strictly less than 65536;

h) the TCP port number may be followed by a slash separator '/' and a textual non-empty string which is called the path part of the URI.

NOTES

1 An example of URI using an Internet Protocol version 4 address is 'malzmtp://192.168.0.1:2534/Service'. This URI references the source or destination 'Service' provided by the application accessible from the TCP port '2534' on the host located at address '192.168.0.1'.

2 An example of URI using an Internet Protocol version 6 address is 'malzmtp://[2001:0db8:85a3:0000:0000:8a2e:0370:7334]:972/Service'. This URI references the source or destination 'Service' provided by the application accessible from the TCP port '972' on the host located at address '2001:0db8:85a3:0000:0000:8a2e:0370:7334'.

3.2.2 The IP address and TCP port number shall uniquely identify an application that implements the conceptual MAL layer of the MO stack.

3.2.2.1 In a concrete deployment, a single or multiple MO Service provider/consumer applications may be deployed on top of the MAL layer.

3.2.2.2 For optimization reasons, the conceptual MAL layer and the MO Service provider/consumer applications layers may be merged.

3.2.2.3 In case of multiple MO Service provider/consumer applications deployed over a single MAL layer application, the path part of the MAL URI must be used to uniquely address individual MO service provider/consumer applications.

3.2.3 The path part of the MAL URI shall be unique for a given MO Service provider or consumer application, which is conceptually deployed on top of a MAL application that is uniquely identified by its IP address and TCP port number.

NOTE – A single application, which is identified by a single IP address and a single TCP port number, may represent several MO Service provider/consumer entities. In order to uniquely address a single MO Service provider/consumer entity, the path part of the URI is used to refine the IP address and TCP port number.

NOTES

1 The scheme name 'malzmtp' will be added to the SANA registry 'MAL Binding URI Scheme Name' and will refer to the MAL Binding to ZMTP Transport document 'CCSDS 524.4-R-1'.

2 This SANA registry is defined in D.

3.3 MAL HEADER MAPPING

3.3.1 OVERVIEW

The following subsections provide the mapping of each field of the MAL message header to the MAL ZMTP PDU.

The mapping configuration parameters are defined in annex B.

3.3.2 MAL HEADER ENCODING

3.3.2.1 General

The encoding format of the header fields shall be as defined in *Mission Operations—MAL Space Packet Transport Binding and Binary Encoding* (reference [3]).

NOTE – A new optional format is introduced in 3.3.2.2 to encode a String as an MDK. Encoding of all other data types complies with reference [3].

3.3.2.2 Optional MDK Format

3.3.2.2.1 Overview

This Recommended Standard defines a mapping directory to allow encoding Strings as numerical keys.

The following fields of the MAL header are concerned by an encoding as an MDK: URI From, URI To, Network Zone, Session Name, and the subdomain parts of the Domain field. They are typed 'Optional MDK' in the MAL ZMTP PDU Header.

3.3.2.2.2 Specification

3.3.2.2.2.1 An 'Optional MDK' field shall be encoded as follows:

MDK encode	String Length/MDK	Character
Integer (variable, multiple of octet)		UTF-8 (variable, multiple of octet)
		Repeated for every character in the String

NOTE – Specifications for numerical and character encoding are contained in references [4] and [5].

3.3.2.2.2.2 The fields 'MDK encode' and 'String Length/MDK' shall be merged into a single signed 32-bit Integer and encoded as a Varint.

3.3.2.2.2.3 If the string element is to be encoded as an MDK, then the following rules shall apply:

a) the Boolean field 'MDK encode' shall be assigned with TRUE and encoded as the minus sign of the signed 32-bit Integer;

b) the field 'String Length/MDK' shall be assigned with the MDK value.

3.3.2.2.2.4 If the string element is not to be encoded as an MDK, then the following rules shall apply:

a) the Boolean field 'MDK encode' shall be assigned with FALSE and not encoded;

b) the field 'String Length/MDK' shall be assigned with the number of octets required to encode the characters of the string.

3.3.2.2.2.5 The field 'String Length/MDK' shall be encoded as the positive part of the signed 32-bit Integer.

NOTE – If the MAL string element is to be encoded as a MDK, the corresponding signed 32-bit Integer shall be assigned with −1 × MDK encoded as varint. If the MAL string element is not to be encoded as MDK the corresponding 32-bit Integer shall be assigned with the number of octets required to encode the characters of the string. This encoding limits the encoded string length to $2^{31}-1$ octets.

3.3.2.3 TIME

3.3.2.3.1 A MAL::Time shall be encoded according to the CCSDS Time Code Format (reference [6]).

3.3.2.3.2 The Time Code Format P-Field for shall be defined as CCSDS Day Segmented Time Code (CDS), no extension flag, with the epoch set as 1958 January 1, 16-bit day segment length and no submillisegment segment. The P-Field binary representation is '01000000'.

3.3.3 URI FROM

3.3.3.1 The value of the MAL header field 'URI From' shall be assigned to the MAL ZMTP PDU header field 'URI From'.

3.3.3.2 The field 'URI From' shall be encoded as an Optional MDK.

NOTE – The fact that the URI From MDK Field uses an explicit string (positive length) or a key (negative length) is not mandated by this specification and is expected to be agreed in an out of band manner. Implementations need to be able to cope with both options (string or key).

3.3.4 AUTHENTICATION ID

3.3.4.1 If the QoS property AUTHENTICATION_ID_FLAG is TRUE, or not passed with the MAL message, then the MAL header field 'Authentication Id' shall be assigned to the MAL ZMTP PDU header field 'Authentication Id' and the 'Authentication Id Flag' shall be set to the value '1'.

3.3.4.2 If the QoS property AUTHENTICATION_ID_FLAG is FALSE, then the following rules shall be applied:

a) the MAL ZMTP PDU header field 'Authentication Id Flag' shall be set to the value '0', and the MAL ZMTP PDU header field 'Authentication Id' shall be left out;

b) if the mapping configuration parameter AUTHENTICATION_ID is defined, then its value shall be assigned to the MAL header field 'Authentication Id';

c) if the mapping configuration parameter AUTHENTICATION_ID is not defined, then an empty MAL::Blob shall be assigned to the MAL header field 'Authentication Id'.

3.3.5 URI TO

3.3.5.1 The value of the MAL header field ‘URI To’ shall be assigned to the MAL ZMTP PDU header field ‘URI To’.

3.3.5.2 The field ‘URI To’ shall be encoded as an Optional MDK.

NOTE – The fact that the URI From MDK Field uses an explicit string (positive length) or a key (negative length) is not mandated by this specification and is expected to be agreed in an out of band manner. Implementations need to be able to cope with both options (string or key).

3.3.6 TIMESTAMP

3.3.6.1 If the QoS property TIMESTAMP_FLAG is TRUE, or not passed with the MAL message, then the MAL header field ‘Timestamp’ shall be assigned to the MAL ZMTP PDU header field ‘Timestamp’ and the ‘Timestamp Flag’ shall be set to the value ‘1’.

3.3.6.2 If the QoS property TIMESTAMP_FLAG is FALSE, then the following rules shall be applied:

a) the MAL ZMTP PDU header field ‘Timestamp Flag’ shall be set to the value ‘0’, and the MAL ZMTP PDU header field ‘Timestamp’ shall be left out;

b) the value ‘0’ shall be assigned to the MAL header field ‘Timestamp’.

3.3.7 QOSLEVEL

The value of the MAL header field ‘QoSlevel’ shall be assigned to the MAL ZMTP PDU header field ‘QoSlevel’ as specified by table 3-3.

Table 3-3: QoSlevel Field Encoding

QoSlevel value	Encoded Value
BESTEFFORT	0
ASSURED	1
QUEUED	2
TIMELY	3

3.3.8 PRIORITY

3.3.8.1 If the QoS property PRIORITY_FLAG is TRUE, or not passed with the MAL message, then the MAL header field 'Priority' shall be assigned to the MAL ZMTP PDU header field 'Priority' and the 'Priority Flag' shall be set to the value '1'.

3.3.8.2 If the QoS property PRIORITY_FLAG is FALSE, then the following rules shall be applied:

a) the MAL ZMTP PDU header field 'Priority Flag' shall be set to the value '0', and the MAL ZMTP PDU header field 'Priority' shall be left out;

b) if the mapping configuration parameter PRIORITY is defined, then its value shall be assigned to the MAL header field 'Priority';

c) if the mapping configuration parameter PRIORITY is not defined, then the value '0' shall be assigned to the MAL header field 'Priority'.

3.3.9 DOMAIN

3.3.9.1 If the QoS property DOMAIN_FLAG is TRUE, or not passed with the MAL message, then the following rules shall be applied:

a) the MAL header field 'Domain' shall be assigned to the MAL ZMTP PDU header field 'Domain', and the 'Domain Flag' shall be set to the value '1';

b) The MAL ZMTP PDU header field 'Domain' shall be encoded as a List of String, each encoded as an Optional MDK.

NOTE – The fact that the URI From MDK Field uses an explicit string (positive length) or a key (negative length) is not mandated by this specification is expected to be agreed in an out of band manner. Implementations need to be able to cope with both options (string or key).

3.3.9.2 If the QoS property DOMAIN_FLAG is FALSE, then the following rules shall be applied:

a) the MAL ZMTP PDU header field ‘Domain Flag’ shall be set to the value ‘0’, and the MAL ZMTP PDU header field ‘Domain’ shall be left out;

b) if the mapping configuration parameter DOMAIN is defined, then its value shall be assigned to the MAL header field ‘Domain’;

c) if the mapping configuration parameter DOMAIN is not defined, then an empty List<MAL::Identifier> shall be assigned to the MAL header field ‘Domain’.

3.3.10 NETWORK ZONE

3.3.10.1 If the QoS property NETWORK_ZONE_FLAG is TRUE, or not passed with the MAL message, then the following rules shall be applied:

a) the MAL header field ‘Network Zone’ shall be assigned to the MAL ZMTP PDU header field ‘Network Zone’, and the ‘Network Flag’ shall be set to the value ‘1’;

b) The MAL ZMTP PDU header field ‘Network Zone’ shall be encoded as an Optional MDK.

NOTE – The fact that the URI From MDK Field uses an explicit string (positive length) or a key (negative length) is not mandated by this specification and is expected to be agreed in an out of band manner. Implementations need to be able to cope with both options (string or key).

3.3.10.2 If the QoS property NETWORK_ZONE_FLAG is FALSE, then the following rules shall be applied:

a) the MAL ZMTP PDU header field ‘Network Zone Flag’ shall be set to the value ‘0’, and the MAL ZMTP PDU header field ‘Network Zone’ shall be left out;

b) if the mapping configuration parameter NETWORK_ZONE is defined, then its value shall be assigned to the MAL header field ‘Network Zone’;

c) if the mapping configuration parameter NETWORK_ZONE is not defined, then an empty MAL::Identifier shall be assigned to the MAL header field ‘Network Zone’.

3.3.11 SESSION

The value of the MAL header field ‘Session’ shall be assigned to the MAL ZMTP PDU header field ‘Session’ as specified by table 3-4.

Table 3-4: Session Field Encoding

Session value	Encoded Value
LIVE	0
SIMULATION	1
REPLAY	2

3.3.12 SESSION NAME

3.3.12.1 If the QoS property SESSION_NAME_FLAG is TRUE, or not passed with the MAL message, then the following rules shall be applied:

a) the MAL header field ‘Session Name’ shall be assigned to the MAL ZMTP PDU header field ‘Session Name’, and the ‘Session Flag’ shall be set to the value ‘1’;

b) the MAL ZMTP PDU header field ‘Session Name’ shall be encoded as an Optional MDK.

NOTE – The fact that the URI From MDK Field uses an explicit string (positive length) or a key (negative length) is not mandated by this specification and is expected to be agreed in an out of band manner. Implementations need to be able to cope with both options (string or key).

3.3.12.2 If the QoS property SESSION_NAME_FLAG is FALSE, then the following rules shall be applied:

a) the MAL ZMTP PDU header field ‘Session Name Flag’ shall be set to the value ‘0’, and the MAL ZMTP PDU header field ‘Session Name’ shall be left out;

b) if the mapping configuration parameter SESSION_NAME is defined, then its value shall be assigned to the MAL header field ‘Session Name’;

c) if the mapping configuration parameter SESSION_NAME is not defined, then an empty MAL::Identifier shall be assigned to the MAL header field ‘Session Name’.

3.3.13 INTERACTION TYPE AND STAGE

The MAL header fields ‘Interaction Type’ and ‘Interaction Stage’ shall be mapped to the MAL ZMTP PDU header field ‘SDU Type’ as defined by table 3-5, where the SDU type used to map an interaction stage raising an error shall be the same as the SDU type used by this interaction stage without an error.

Table 3-5: Interaction Type and Stage Mapping

Interaction Type	Interaction Stage	SDU Type (decimal)
SEND	SEND	0
SUBMIT	SUBMIT ACK ERROR	1 2 2
REQUEST	REQUEST RESPONSE ERROR	3 4 4
INVOKE	INVOKE ACK ACK_ERROR RESPONSE RESPONSE ERROR	5 6 6 7 7
PROGRESS	PROGRESS ACK ACK_ERROR UPDATE UPDATE_ERROR RESPONSE RESPONSE ERROR	8 9 9 10 10 11 11
PUBLISH-SUBSCRIBE	REGISTER REGISTER_ACK REGISTER_ERROR PUBLISH_REGISTER PUBLISH_REGISTER_ACK PUBLISH_REGISTER_ERROR PUBLISH PUBLISH_ERROR NOTIFY NOTIFY ERROR DEREGISTER DEREGISTER ACK PUBLISH_DEREGISTER PUBLISH_DEREGISTER_ACK	12 13 13 14 15 15 16 16 17 17 18 19 20 21

3.3.14 TRANSACTION ID

The value of the MAL header field ‘Transaction Id’ shall be assigned to the MAL ZMTP PDU header field ‘Transaction Id’.

3.3.15 SERVICE AREA

The value of the MAL header field ‘Service Area’ shall be assigned to the MAL ZMTP PDU header field ‘Service Area’.

3.3.16 SERVICE

The value of the MAL header field ‘Service’ shall be assigned to the MAL ZMTP PDU header field ‘Service’.

3.3.17 OPERATION

The value of the MAL header field ‘Operation’ shall be assigned to the MAL ZMTP PDU header field ‘Operation’.

3.3.18 AREA VERSION

The value of the MAL header field ‘Area Version’ shall be assigned to the MAL ZMTP PDU header field ‘Area Version’.

3.3.19 IS ERROR MESSAGE

If the MAL header field ‘Is Error Message’ is TRUE, then the MAL ZMTP PDU header field ‘Is Error Message’ shall be assigned with the value ‘1’; otherwise it shall be assigned with the value ‘0’.

3.4 MAL ZMTP PROTOCOL DATA UNIT SPECIFIC FIELDS

3.4.1 OVERVIEW

The following subsections specify the values to be assigned to the MAL ZMTP PDU header fields that are not the result of the MAL header mapping.

3.4.2 VERSION NUMBER

3.4.2.1 The field ‘Version Number’ shall identify the structure of the MAL ZMTP PDU header as defined by this Recommended Standard.

3.4.2.2 The field ‘Version Number’ shall be assigned with the binary value ‘001’.

NOTES

1 The version number '001' will be added to the SANA registry 'MAL ZMTP Binding Version Number' and will refer to the MAL Binding to ZMTP Transport document 'CCSDS 524.4-R-1'.

2 This SANA registry is defined in annex D.

3.4.3 EXTENDED ENCODING ID

3.4.3.1 The field 'Extended Encoding Id' shall identify the encoding rules used to encode the MAL message body.

3.4.3.2 The 'Encoding Id' shall be an integer number greater than or equal to 0 and strictly less than 256.

NOTE – The mapping between the value of the 'Encoding Id' header field and the encoding rules is not defined as part of this Recommended Standard.

3.4.3.3 Organizations implementing this Recommended Standard shall agree on the encoding rules and related identification using this field.

3.5 MAL MESSAGE BODY MAPPING

3.5.1 DISCUSSION

Any MAL encoding, specified in other books, can be used for encoding the message body. Examples are the binary encoding specified in the book MO MAL Space Packet Transport Binding and Binary Encoding, or the split binary encoding specified in the book MO MAL Binding to TCP/IP Transport and Split Binary Encoding.

This Recommended Standard does not define a new encoding.

3.5.2 REQUIREMENTS

3.5.2.1 The MAL message body shall be encoded using the selected encoding format and assigned to the MAL TCP/IP Protocol Data Unit data field.

3.5.2.2 The identifier of the selected encoding format for the MAL message body shall be assigned to the MAL TCP/IP Protocol Data Unit header field 'Encoding Id'.

4 MAL TRANSPORT INTERFACE MAPPING

4.1 OVERVIEW

The MAL specification (reference [1]) 'Transport Interface' section defines the interface to be provided by the MAL ZMTP Transport binding layer. The following subsections specify the expected behaviour for each of the MAL transport interface request and indication primitives. If an indication is a response to a request, then the behaviour of the indication is specified in the same subsection as the request.

4.2 REQUIREMENTS

4.2.1 The following primitives are defined in the MAL transport interface and shall be provided by the MAL ZMTP Transport binding layer:

a) SUPPORTEDQOS request;

b) SUPPORTEDQOS RESPONSE indication;

c) SUPPORTEDIP request;

d) SUPPORTEDIP RESPONSE indication;

e) TRANSMIT request;

f) TRANSMIT ACK indication;

g) TRANSMIT ERROR indication;

h) TRANSMITMULTIPLE request;

i) TRANSMITMULTIPLE ACK indication;

j) TRANSMITMULTIPLE ERROR indication;

k) RECEIVE indication;

l) RECEIVEMULTIPLE indication.

NOTE – The parameters are listed in table 4-1.

Table 4-1: MAL Transport Interface Primitives

Primitive	Parameters
SUPPORTEDQOS request	QoS Level
SUPPORTEDQOS RESPONSE indication	Boolean
SUPPORTEDIP request	Interaction Type
SUPPORTEDIP RESPONSE indication	Boolean
TRANSMIT request	MAL Message QoS Properties
TRANSMIT ACK indication	-
TRANSMIT ERROR indication	MAL Message Header Error Number Extra Information QoS Properties
TRANSMITMULTIPLE request	List of: – MAL Message – QoS Properties
TRANSMITMULTIPLE ACK indication	-
TRANSMITMULTIPLE ERROR indication	List of: – MAL Message Header – Error Number – Extra Information – QoS Properties
RECEIVE indication	MAL Message QoS Properties
RECEIVEMULTIPLE indication	List of: – MAL Message – QoS Properties

4.2.2 The following primitives (reference [2]) shall be used by the mapping (the parameters are listed in table 4-2):

– ‘OPEN’ : creates and initializes a ZMTP socket.

  • If the parameter ‘as-server’ is TRUE, the type of the created socket is ROUTER if the parameter ‘communication pattern’ is P2P, or SUB if the parameter is MCAST. The created socket is then bound[1] to the parameter ‘ZMTP URI’. The returned socket is ready for subsequent calls to the ‘RECEIVE’ primitive.

  • If the parameter ‘as-server’ is FALSE, the type of the created socket is DEALER if the parameter ‘communication pattern’ is P2P, or PUB if the parameter is MCAST. The created socket is then connected[1] to the parameter ‘ZMTP URI’. The returned socket is ready for subsequent calls to the ‘SEND’ primitive.

[1] According to the ZMTP bind and connect operations

– ‘SEND’ : sends a ZMTP frame through the related socket.

– ‘RECEIVE’ : receives a ZMTP frame through the related socket.

Table 4-2: ZMTP Interface Primitives

Primitive	Parameters
OPEN	Communication pattern ZMTP URI As-server
SEND	Local socket name Frame
RECEIVE	Local socket name Frame

4.2.3 The use of the ZMTP protocol by a MAL/ZMTP Transport follows rules that restrict the possibilities offered by the protocol. Those rules shall apply for all communications between Transports.

a) Messages are sent as ZMTP frames over communication channels.

b) A communication channel implements a communication pattern.

c) Two communication patterns are enabled: point-to-point (p2p) and multicast (mcast).

d) A p2p communication channel links a source socket to a destination socket.

e) An mcast communication channel links a source socket to many destination sockets.

f) A communication channel is one way. Frames are sent from the source socket to the destination sockets.

g) The source socket of a p2p communication channel is a DEALER ZMQ socket.

h) The destination socket of a p2p communication channel is a ROUTER ZMQ socket.

i) The source socket of an mcast communication channel is a PUB ZMQ socket.

j) The destination sockets of an mcast communication channel are SUB ZMQ sockets.

k) All the destination sockets of a communication channel are identified from the sender by a single ZMTP URI.

4.2.4 A MAL service can be accessed through a MAL/ZMTP Transport using a p2p communication channel and an mcast communication channel. The p2p channel is mandatory. The mcast channel is optional.

4.2.5 In order to allow for various implementations of the transport, the mapping from the MAL URI of the service and the ZMTP URIs of the destination socket of its channels shall be left to the user to define through the following set of configuration functions:

- 'get_local_ptp_zmtp_uri' takes in parameter the MAL URI of the service and returns the ZMTP URI used to create the p2p communication channel on the receiver side with a call to the ZMTP 'OPEN' primitive.
- 'get_local_mcast_zmtp_uri' takes in parameter the MAL URI of the service and returns the ZMTP URI used to create the mcast communication channel on the receiver side with a call to the ZMTP 'OPEN' primitive. If it returns NULL the mcast channel is not used by the ZMTP transport.
- 'get_remote_ptp_zmtp_uri' accepts in parameter the MAL header field 'URI To' of the message. It returns the ZMTP URI used to create the p2p communication channel on the sender side with a call to the ZMTP 'OPEN' primitive.
- 'get_remote_mcast_zmtp_uri' accepts in parameter the MAL header field 'URI To' of the message. It returns the ZMTP URI used to create the mcast communication channel on the sender side with a call to the ZMTP 'OPEN' primitive. If it returns NULL the mcast channel is not used by the ZMTP transport.

A MAL ZMTP PDU is a unique ZMTP Message; it can be segmented into multiple ZMTP frames. In this case it will be necessary to call multiple times the ZMTP 'SEND' ('RECEIVE', resp.) primitive to send (receive, resp.) it. The first frame should at least include entirely the PDU Header. A ZMTP frame cannot hold parts of two different MAL ZMTP PDUs.

4.3 SUPPORTEDQOS REQUEST

4.3.1 The SUPPORTEDQOS request primitive shall be provided.

4.3.2 Support for the Quality of Service (QoS) levels defined by MAL shall depend on the capabilities of the underlying layer used to convey the ZMTP Messages.

4.4 SUPPORTEDIP REQUEST

4.4.1 The SUPPORTEDIP request primitive shall be provided.

4.4.2 The SUPPORTEDIP request primitive shall return TRUE for all the interaction patterns: SEND, SUBMIT, REQUEST, INVOKE, PROGRESS, and PUBLISH-SUBSCRIBE.

4.5 TRANSMIT REQUEST

4.5.1 The TRANSMIT request primitive shall be provided in order to translate a MAL message into one MAL ZMTP PDU and send it by calling the ZMTP primitive 'SEND'.

4.5.2 If any of the MAL header fields is NULL, then the TRANSMIT ERROR primitive shall be called with the error number MAL::INTERNAL.

4.5.3 The MAL header field 'Transaction Id' shall not be NULL, especially in MAL messages whose interaction type is SEND.

4.5.4 The MAL message header fields and body elements shall be mapped to the MAL ZMTP PDU according to the specification given in section 3 of this Recommended Standard.

4.5.5 If either of the fields 'URI From' or 'URI To' is not compliant with the URI format defined in 3.2, then the TRANSMIT ERROR primitive shall be called with the error number MAL::INTERNAL.

4.5.6 If the MAL ZMTP PDU is segmented in multiple frames, the ZMTP primitive 'SEND' shall be called for each frame.

4.5.7 When the MAL ZMTP PDU must be delivered, the following rules shall be applied:

a) If the transport prefers to use the mcast communication channel,

   1) the ZMTP URI corresponding to the MAL header field 'URI To' shall be retrieved using the 'get_remote_mcast_zmtp_uri' configuration function provided by the client;

   2) if the function returns NULL, then the transport shall use the p2p communication channel instead.

b) If the transport prefers or is forced to use the p2p communication channel, the ZMTP URI corresponding to the MAL header field 'URI To' shall be retrieved using the 'get_remote_p2p_zmtp_uri' configuration function provided by the client.

c) If there is no suitable communication channel to the destination MAL service, i.e., with a destination socket corresponding to the retrieved ZMTP URI, then a new communication channel shall be established using the ZMTP 'OPEN' primitive.

d) The parameter 'communication pattern' of the ZMTP 'OPEN' primitive shall be set to P2P for a p2p communication channel, or MCAST for an mcast communication channel.

e) The ZMTP URI shall be passed to the ZMTP 'OPEN' primitive as 'ZMQ URI'.

f) The parameter 'as-server' of the ZMTP 'OPEN' primitive shall be set to FALSE.

g) The new communication channel shall be identifiable by its local socket name, returned by a successful invocation of the ZMTP 'OPEN' primitive.

h) If the ZMTP URI cannot be retrieved or the ZMTP 'OPEN' primitive fails, then the TRANSMIT ERROR primitive shall be called with the error number MAL::INTERNAL.

i) If the ZMTP 'OPEN' primitive successfully returns with the local socket name, the MAL ZMTP PDU shall be sent using the ZMTP 'SEND' primitive.

j) The local socket name of the selected communication channel shall be passed to the ZMTP 'SEND' primitive as 'Local socket name'.

k) The MAL ZMTP PDU may be transmitted in multiple ZMTP frames.

l) A ZMTP frame shall not hold parts of two different MAL ZMTP PDUs.

m) The MAL ZMTP PDU header part shall be transmitted in a unique ZMTP frame.

4.5.8 If an error is returned by the invocation of the ZMTP 'SEND' primitive, then the TRANSMIT ERROR primitive shall be called with the error number MAL::INTERNAL.

4.5.9 If the invocations of the ZMTP 'SEND' primitive over all the ZMTP frames derived from the MAL ZMTP PDU successfully return, then the TRANSMIT ACK primitive shall be called.

4.6 TRANSMITMULTIPLE REQUEST

4.6.1 The TRANSMITMULTIPLE request primitive shall be provided by calling the TRANSMIT request primitive for every MAL message.

4.6.2 If the TRANSMIT ERROR indication is called for any of the MAL messages, the TRANSMIT ERROR indications should be collected, and the TRANSMITMULTIPLE ERROR indication should be called with the content of the collected TRANSMIT ERROR indications.

4.6.2.1 The individual TRANSMIT ERROR indications shall not be transmitted to the MAL.

4.6.2.2 Only the TRANSMITMULTIPLE ERROR indication shall be called.

4.7 RECEIVE INDICATION

4.7.1 The RECEIVE indication primitive shall be provided in order to receive one MAL ZMTP PDU and translate it into a MAL message.

4.7.2 The RECEIVE indication primitive shall be called once a complete MAL ZMTP PDU has been read by the underlying ZMTP communication channels.

4.7.3 In order to read a complete MAL ZMTP PDU, the ZMTP 'RECEIVE' primitive shall be called one or several times (one time per ZMTP frame sent) until the ZMTP Message has been fully received.

4.7.4 The ZMTP 'RECEIVE' primitive shall be called after the mcast and p2p communication channels of the service have been established.

4.7.5 The mcast communication channel shall be established using the following rules:

a) The ZMTP URI corresponding to the MAL URI of the service

   1) shall be retrieved using the 'get_local_mcast_zmtp_uri' configuration function provided by the client;

   2) if the function returns NULL, then no mcast communication channel shall be opened.

b) The communication channel shall be established using the ZMTP 'OPEN' primitive.

c) The parameter 'communication pattern' of the ZMTP 'OPEN' primitive shall be set to MCAST.

d) The retrieved ZMTP URI shall be passed to the ZMTP 'OPEN' primitive as 'ZMTP URI'.

e) The parameter 'as-server' of the ZMTP 'OPEN' primitive shall be set to TRUE.

f) If the ZMTP 'OPEN' primitive fails, no MAL ZMTP PDUs shall be received using this URI.

g) If the ZMTP 'OPEN' primitive successfully returns, the returned local socket name shall be passed to the ZMTP 'RECEIVE' primitive as 'local socket name'.

4.7.6 The p2p communication channel shall be established using the following rules:

a) The ZMTP URI corresponding to the MAL URI of the service shall be retrieved using the 'get_local_p2p_zmtp_uri' configuration function provided by the client.

b) The communication channel shall be established using the ZMTP 'OPEN' primitive.

c) The parameter 'communication pattern' of the ZMTP 'OPEN' primitive shall be set to P2P.

d) The retrieved ZMTP URI shall be passed to the ZMTP 'OPEN' primitive as 'ZMTP URI'.

e) The parameter 'as-server' of the ZMTP 'OPEN' primitive shall be set to TRUE.

f) If the ZMTP 'OPEN' primitive fails, no MAL ZMTP PDUs shall be received using this URI.

g) If the ZMTP ‘OPEN’ primitive successfully returns, the returned local socket name shall be passed to the ZMTP ‘RECEIVE’ primitive as ‘local socket name’.

4.7.7 The MAL message header fields and body elements shall be generated according to the specifications given in section 3 of this Recommended Standard, by using the following input data:

a) the MAL ZMTP PDU;

b) the XML specification of the MO service (see section 6 of reference [1]) identified by the MAL header fields ‘Service Area’, ‘Service’, and ‘Area Version’.

4.7.8 If the field ‘URI To’ is unknown,

a) the error MAL::DESTINATION_UNKNOWN shall be returned if the Interaction Pattern allows a MAL error message to be returned;

b) the MAL header field ‘URI From’ of the returned error message shall be assigned with the ‘URI To’ field of the initial message, even if this URI is unknown.

4.8 RECEIVEMULTIPLE INDICATION

The RECEIVEMULTIPLE indication primitive shall not be provided.

ANNEX A

PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

This annex provides the Protocol Implementation Conformance Statement (PICS) Requirements List (RL) for an implementation of the Mission Operations ZMTP Transport and Binary Encoding standard. The PICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation claiming conformance must satisfy the mandatory requirements referenced in the RL.

An implementation's completed RL is called the PICS. The PICS states which protocol features have been implemented. The following entities can use the PICS:

- the protocol implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;
- the supplier and acquirer or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- the user or potential user of the implementation, as a basis for initially checking the possibility of interworking with another implementation (while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSes);
- a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A1.2 NOTATION

A1.2.1 Status Column Symbols

The following symbols are used in the RL to indicate the status of features:

Symbol	Meaning
M	Mandatory
O	Optional

A1.2.2 Support Column Symbols

The support of every item as claimed by the implementer is stated by entering the appropriate answer (Y, N, or N/A) in the support column.

Symbol	Meaning
Y	Yes, supported by the implementation
N	No, not supported by the implementation
N/A	Not applicable

A2 GENERAL INFORMATION

A2.1 IDENTIFICATION OF PICS

Ref	Question	Response
1	Date of Statement (DD/MM/YYYY)	
2	CCSDS document number containing the PICS	
3	Date of CCSDS document containing the PICS	

A2.2 IDENTIFICATION OF IMPLEMENTATION UNDER TEST (IUT)

Ref	Question	Response
1	Implementation name	
2	Implementation version	
3	Machine name	
4	Machine version	
5	Operating System name	
6	Operating System version	
7	Special Configuration	
8	Other Information	

A2.3 USER IDENTIFICATION

Supplier	
Contact Point for Queries	
Implementation name(s) and Versions	
Other Information Necessary for full identification —e.g., name(s) and version(s) for machines and/or operating systems; System Name(s)	

A2.4 INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the protocol by completing the RL; the resulting completed RL is called a PICS.

A3 MAL BINDING TO ZMTP TRANSPORT PICS

A3.1 MESSAGE ABSTRACTION LAYER

Item	Protocol Feature	Reference	Status	Support
1-1	Transaction Handling	[1] subsection 3.2	M	
1-2	State Transitions	[1] subsection 3.3	M	
1-3	Message Composition	[1] subsection 3.4	M	
1-4	MAL Service Interface	[1] subsection 3.5	M	
1-5	Access Control Interface	[1] subsection 3.6	M	
1-6	Transport Interface	[1] subsection 3.7	M	
1-7	MAL Data Type Specification	[1] section 4	M	
1-8	MAL Errors	[1] section 5	M	

A3.2 MAL MESSAGE MAPPING

Item	Protocol Feature	Reference	Status	Support
2-1	URI Format	3.2	M	
2-2	MAL Header Mapping	3.3	M	
2-3	Field ‘Timestamp’	3.3.6	O	
2-4	Fields ‘Priority’, ‘Domain’, ‘Network Zone’, ‘Session Name’	3.3.8 3.3.9 3.3.10 3.3.12	O	
2-5	Field ‘Authentication Id’	3.3.4	O	
2-6	MAL ZMTP PDU Specific Fields	3.4	M	
2-7	MAL Message Body Mapping	3.4.3	M	

A3.3 MAL TRANSPORT INTERFACE MAPPING

Item	Protocol Feature	Reference	Status	Support
3-1	SupportedQoS Request	4.3	M	
3-2	SupportedIP Request	4.4	M	
3-3	Transmit Request	4.5	M	
3-4	TransmitMultiple Request	4.6	M	
3-5	Receive Indication	4.7	M	
3-6	ReceiveMultiple Indication	4.8	M	

ANNEX B

MAPPING CONFIGURATION PARAMETERS

(NORMATIVE)

B1 OVERVIEW

This annex defines the parameters that are provided by the MAL ZMTP Transport protocol in order to configure and optimize the MAL message mapping and the format of the MAL ZMTP PDU transmitted over ZMTP.

The mapping configuration parameters are managed parameters, defined by some out-of-band agreement. Those parameters can be exchanged for example by email, or through a common registry like the Space Assigned Numbers Authority (SANA). Ideally, those parameters would be retrieved by using the Mission Operations directory service.

B2 MAPPING CONFIGURATION PARAMETERS

The mapping configuration parameters shall be as specified in table B-1.

Table B-1: Mapping Configuration Parameters

Parameter Name	Type	Description
AUTHENTICATION_ID	MAL:Blob	Value to be assigned to the MAL header field 'Authentication Id' if the QoS property AUTHENTICATION_ID_FLAG is FALSE
DOMAIN	List<MAL::Identifier>	Value to be assigned to the MAL header field 'Domain' if the QoS property DOMAIN_FLAG is FALSE
NETWORK_ZONE	MAL::Identifier	Value to be assigned to the MAL header field 'Network Zone' if NETWORK_ZONE_FLAG is FALSE
PRIORITY	MAL::UInteger	Value to be assigned to the MAL header field 'Priority' if PRIORITY_FLAG is FALSE
SESSION_NAME	MAL::Identifier	Value to be assigned to the MAL header field 'Session Name' if SESSION_NAME_FLAG is FALSE

ANNEX C

QOS PROPERTIES

(NORMATIVE)

C1 OVERVIEW

This annex defines the QoS properties that are provided by the MAL ZMTP Transport protocol. QoS properties are set on a per-message basis as specified by the MAL (reference [1]).

C2 QOS PROPERTIES

Permitted QoS properties shall be those indicated in table C-1.

Table C-1: QoS Properties

QoS Property Name	Type	Description
AUTHENTICATION_ID_FLAG	MAL::Boolean	Value to be assigned to the secondary header field ‘Authentication Id Flag’
DOMAIN_FLAG	MAL::Boolean	Value to be assigned to the secondary header field ‘Domain Flag’
NETWORK_ZONE_FLAG	MAL::Boolean	Value to be assigned to the secondary header field ‘Network Zone Flag’
PRIORITY_FLAG	MAL::Boolean	Value to be assigned to the secondary header field ‘Priority Flag’
SESSION_NAME_FLAG	MAL::Boolean	Value to be assigned to the secondary header field ‘Session Name Flag’
TIMESTAMP_FLAG	MAL::Boolean	Value to be assigned to the secondary header field ‘Timestamp Flag’

ANNEX D

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

D1 SECURITY CONSIDERATIONS

D1.1 OVERVIEW

This annex subsection discusses various aspects of security with respect to the MAL ZMTP Transport protocol.

D1.2 SECURITY BACKGROUND

The following security aspects are typically separated:

a) data and data origin authentication: corroboration of the source of information that is contained in a message;

b) authorization: conveyance, to another entity, of official sanction to do or be something;

c) confidentiality: keeping information secret from all but those who are authorized to see it;

d) integrity: detecting that information has not been altered by unauthorized or unknown means.

The MAL ZMTP Transport protocol is not responsible for ensuring all these security aspects; however, it has to fulfil the security criteria expected by the MAL layer from every transport binding. These criteria are:

a) the Transport Layer is responsible for the transmission of the authentication identifier assigned by the MAL layer to every consumer;

b) the Transport Layer has to provide authentication, confidentiality, and integrity of the transmitted messages.

D1.3 SECURITY CONCERNS WITH RESPECT TO THE CCSDS DOCUMENT

D1.3.1 Data Privacy

It is assumed that message authentication and confidentiality are provided by the ZMTP layer and are transparent to the ZMTP Transport binding and above. As a consequence, once a message rises above the ZMTP protocol layer, the message has been authenticated and all encryption has been removed.

D1.3.2 Data Integrity

Integrity is ensured by the protocol that conveys the ZMTP frames.

D1.3.3 Authentication of Communicating Entities

Authentication of the consumers is done above the MAL layer through a specific service that enables a consumer to get an authentication identifier. The meaning of that authentication identifier is dependent on the security system used for the deployment. This identifier must allow the MAL access control implementation to perform a lookup for authorization purposes.

The authentication identifier is transmitted by the MAL ZMTP Transport protocol in the parameter 'Authentication Id' of the MAL ZMTP PDU header; however, this parameter may be omitted, as it is optional.

The MAL authentication identifier is an implementation- and technology-specific security credential created at a higher layer by MAL access control. The ZMTP Transport protocol allows this implementation-specific security credential to be transferred from ZMTP source to destination. It is possible for this security credential to be used by the protocol below ZMTP for authentication or even confidentiality purposes (e.g., IPsec), but that is not specified here.

D1.3.4 Control of Access to Resources

Authorization is done by the MAL access control that performs any required authorization checks and converts the consumer identifier into technology-dependent security credentials.

D1.4 POTENTIAL THREATS AND ATTACK SCENARIOS

Potential threats and attack scenarios depend on the layer that is beneath the MAL ZMTP Transport protocol because this is the layer that defines the security algorithms ensuring authentication, confidentiality, and integrity.

D1.5 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

The only security aspect that may not be applied is the transmission of the authentication identifier in the ZMTP MAL header. If the authentication identifier is not transmitted by the MAL ZMTP Transport protocol, then delivered messages may be rejected by the MAL access control.

D2 SANA CONSIDERATIONS

D2.1 VERSION NUMBER

The recommendations of this document request SANA to create the registry defined as follows:

a) the registry named ‘MAL ZMTP Transport Version Number’ consists of a table of parameters:

   1) Version Number: a string of text specifying the three bits to be assigned to the packet secondary header field ‘Version Number’;

   2) Reference: a string of text referencing the CCSDS document that specifies the version of the MAL ZMTP Transport;

b) the initial registry should be filled with the values in table D-1.

Table D-1: MAL ZMTP Transport Version Number Initial Values

Version Number	Reference
001	CCSDS 524.4-R-1

D2.2 URI SCHEME NAME

The recommendations of this document request SANA to create the registry defined as follows:

a) the registry named ‘MAL ZMTP Transport URI Scheme Name’ consists of a table of parameters:

   1) Scheme Name: a string of text specifying the name of the URI scheme defined by the MAL binding;

   2) Reference: a string of text referencing the CCSDS document that specifies the MAL binding;

b) the initial registry should be filled with the values in table D-2.

Table D-2: MAL ZMTP Transport Binding URI Scheme Name Initial Values

Scheme Name	Reference
malzmtp	CCSDS 524.4-R-1

D2.3 MAL ENCODING IDS

The recommendations of this document request SANA to create the registry defined as follow:

a) the registry named ‘MAL Encoding Ids’ consists of a table of parameters:

   1) Encoding Id: an unsigned integer between 0 and 255 specifying the encoding identifier to be used by the Encoding Id field available on MAL Bindings; Values greater than 127 shall be reserved for non-standard encodings.

   2) Encoding: a string of text describing the encoding name;

   3) Reference: a string of text referencing the CCSDS document that specifies the book and respective section (if applicable) where the Encoding is defined;

b) the initial registry should be filled with the values in table table D-3.

Table D-3: MAL Encoding Ids

Encoding Id	Encoding	Reference
0	Fixed Binary	CCSDS 524.1-B-1 – section 5
1	Variable Length Binary	CCSDS 524.1-B-1 – section 5
2	Split Binary	CCSDS 524.2-B-1 – section 5

D3 PATENT CONSIDERATIONS

No patents are known to apply to this Recommended Standard.

ANNEX E

ENCODING EFFICIENCY

(INFORMATIVE)

E1 INTRODUCTION

This annex lists the potential overhead costs caused by the encoding format.

E2 BANDWIDTH OVERHEAD

E2.1 MAL ZMTP HEADER

The overhead caused by the secondary header depends on whether the optional fields are inserted or not in the MAL ZMTP PDU header: 'Priority', 'Timestamp', 'Network Zone', 'Session Name', 'Domain', and 'Authentication Id'.

The minimum overhead is 20 octets. It is obtained with all optional MAL header fields not passed (presence flags set to FALSE) and with the actual use of MDKs in the encoding of the 'URI From' and 'URI To' header fields encoded in a single octet. The additional overheads are given by table E-1 in terms of:

a) the name of a MAL ZMTP PDU header field;

b) the encoding format of the fields.

Table E-1: MAL ZMTP Protocol Data Unit Header Additional Overheads

Field Name	Encoding Format
URI From	Optional MDK with no actual MDK use
URI To	Optional MDK with no actual MDK use
Priority	UInteger
Timestamp	Time
Network Zone	Identifier
Session Name	Identifier
Domain	List<Identifier>
Authentication Id	Blob

E2.2 BINARY ENCODING

The encoding of the data in the header follow the Binary *Mission Operations—MAL Space Packet Transport Binding and Binary Encoding* (reference [3]). Encoding efficiency of those parts are discussed in the reference document.

ANNEX F

ACRONYMS

(INFORMATIVE)

This annex lists the acronyms used in this Recommended Standard.

ASCII	American Standard Code for Information Interchange
AOS	Advanced Orbiting Systems
BP	Bundle Protocol
CCSDS	Consultative Committee for Space Data Systems
IP	Internet Protocol or interaction pattern
MAL	Message Abstraction Layer
MSB	most significant bit
OSI	Open Systems Interconnection
PDU	protocol data unit
QoS	quality of service
SANA	Space Assigned Numbers Authority
SDU	service data unit
SLE	Space Link Extension
SM&C	CCSDS Spacecraft Monitoring and Control
TC	telecommand
TM	telemetry
URI	Uniform Resource Identifier
UTC	Universal Time Coordinated
WG	working group
ZMTP	ZeroMQ Message Transport Protocol

ANNEX G

INFORMATIVE REFERENCES

(INFORMATIVE)

[G1] *Mission Operations Services Concept*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 520.0-G-3. Washington, D.C.: CCSDS, December 2010.

ANNEX H

URI MAPPING

(INFORMATIVE)

The purpose of this annex is to specify an example of the URI transformation functions described in section 4.

The mapping between the MAL URI of a service and the corresponding ZMTP URIs is ensured by four configuration functions provided by the user. A default mapping could be:

– ‘get_local_p2p_ztmp_uri’ provides the ZMTP URI used to open the p2p communication channel from the destination end. A ZMTP bind will be executed with the resulting ZMTP URI. An example function could be:

MAL URI	ZMTP URI
"malzmtp://host:port/service"	"tcp://*:port"

– ‘get_local_mcast_ztmp_uri’ provides the ZMTP URI used to open the mcast communication channel from the destination end. A ZMTP bind will be executed with the resulting ZMTP URI. Example functions could be:

MAL URI	ZMTP URI
"malzmtp://host:port/service"	"tcp://host:(port+1)"
"malzmtp://host:port/service"	"epgm://itf;mcast_addr:(port+1)"

– ‘get_remote_ptp_zmtp_uri’ provides the ZMTP URI used to open the p2p communication channel from the source end. A ZMTP connect will be executed with the resulting ZMTP URI. An example function could be:

MAL URI	ZMTP URI
"malzmtp://host:port/service"	"tcp://host:port"

– ‘get_remote_mcast_zmtp_uri’ provides the ZMTP URI used to open the mcast communication channel from the source end. A ZMTP connect will be executed with the resulting ZMTP URI. Example functions could be:

MAL URI	ZMTP URI
"malzmtp://host:port/service"	"tcp://*:(port+1)"
"malzmtp://host:port/service"	"epgm://itf;mcast_addr:(port+1)"

ANNEX I

MAL TRANSPORT INTERFACE MAPPING EXAMPLE

(INFORMATIVE)

I1 INTRODUCTION

The purpose of this annex is to specify an example of use of the MAL Transport Interface Mapping described in section 4.

The example describes the behavior of a provider and a consumer during a MAL SEND interaction. This example uses a unique p2p communication channel and is intended to explain the sequencing of operations described in section 4.

The provider is identified by the MAL URI ‘malzmtp://host1:port1/service1’. To simplify the description it will be assumed that the PDUs resulting from the encoding of the MAL messages are transmitted in a single frame.

The consumer is identified by the MAL URI ‘malzmtp://host2:port2/service2’, it interacts with the provider through a MAL SEND interaction.

The paragraphs below describe the different stages allowing interaction between the consumer and the provider.

I2 PROVIDER INITIALIZATION

During the initialization the provider must create the listening ZMTP socket on which it will receive the messages from the consumers.

In a first step, it gets the ZMTP URI of this socket by calling the URI transformation function get_local_ptp_zmtp_uri (see annex H) with the MAL URI of its service. It gets back the ZMTP URI ‘tcp:// *:port1’.

In a second step, it calls the OPEN function of the ZMTP transport by passing in parameters the communication channel type (p2p), the ZMTP URI of the socket corresponding to the service, and the as-server parameter true. It gets back a ZMTP socket ROUTER bound on the URI given in parameter.

The provider can then listen to the messages sent to this socket using the RECEIVE function of the ZMTP transport.

I3 CONSUMER INITIALIZATION

In order to be able to receive messages the consumer must be initialized as the provider. The operations are similar to those described for the provider. We do not detail these operations here.

I4 INTERACTION INITIATION BY THE CONSUMER

The consumer will initiate the MAL SEND interaction by using the primitive TRANSMIT REQUEST of the MAL transport interface. It passes as a parameter of this primitive the MAL message that it wishes to transmit.

The consumer will determine the service provider ZMTP URI by calling the URI transformation function get_remote_ptp_zmtp_uri. It gives as parameter of this function the MAL URI of the service that it gets from the URI TO field of the MAL message. It gets the ZMTP URI of the listening socket of the service: ‘tcp://host1:port1’.

The consumer then calls the OPEN function of the ZMTP transport by passing the communication channel type (p2p), the ZMTP URI of the provider, and the as-server parameter to false. It gets back a ZMTP socket DEALER connected to the listening socket of the provider.

The consumer can then transform the MAL message into a PDU and then transmit it to the provider using the SEND function of the ZMTP transport. It passes as parameter of this function:

- the socket obtained in response to the call of the function OPEN,
- the PDU resulting from the encoding of the MAL message.

If an error is returned then the TRANSMIT ERROR primitive is called, otherwise the TRANSMIT ACK primitive is called.

The consumer can then either keep the socket for future interactions with the provider, or close it.

I5 INTERACTION HANDLING BY THE PROVIDER

The provider receives the transmitted PDU from the ZMTP transport RECEIVE call. It decodes this PDU to retrieve the corresponding MAL message, then it calls the RECEIVE INDICATION primitive.

The provider can wait for the next message by calling again the RECEIVE function of the ZMTP transport.