Appendix 2: Description of the ESDK protocol used for A2A communication

This appendix represents the documentation of the ESDK protocol used in some of the KDPW Group's services for A2A communication using MQ queues.

In order to accommodate the modifications to electronic certificates (in particular the cryptographic algorithms and their use in the authentication process) and the standardisation of the A2A communication rules, the use of an electronic signature in the messages will no longer be required in the protocol. This change consist merely in the removal of the field dedicated to the signature from the frame and, consequently, discontinuing the checks of the signature. The integrity of transmission will be based on the in-built MQ queue support mechanisms.
The other elements of the protocol remain unchanged.

Key information

Electronic communication with KDPW via the A2A communication interface is based on MQ connections set up with a dedicated MQ queue manager. A2A communication is based on consistent and uniform connection setup rules, regardless of the protocols used in the different services. The A2A communication model requires separation of message exchange in the different business services, which means that separate MQ queues identified by their name are provided for each service. As the A2A communication rules allow for a specific protocol or protocols to be established at the level of individual services for the exchange of information with participants, the separation also applies to the handling of protocols within each service. A communication protocol in this case is to be understood as a set of strict rules and data formats, which are required to successfully set up communication on the basis of a standard MQ connection.
One of the protocols offered as part of A2A communication is the ESDK protocol, dedicated to communication with direct participants of KDPW and KDPW_CCP. The protocol is based on a dedicated frame, ensuring technical support for checks of the communication and confirmation of the receipt of the message at the pre-processing stage. The protocol allows the transmission of any content based on standardised message type designations and communication party identifiers.

ESDK protocol description

ESDK communication protocol defines the following parameters :

- ESDK message format,
- ESDK message types,
- ESDK processing procedures for individual message types.

ESDK message format

Field name	Length	Type
Message number	9	N
Date	10	A
Time	8	A
Recipient's ID	10	A
Sender's ID	10	A
Message type	24	A
Message subtype	4	A
Reserved area	20	A
Data length	8	N
Data	Data length	B

Types:

A – character field
B - binary field
N – numeric field

Each message is unambiguously identified by the fields:

- Message number,
- Date,
- Sender's ID.

Message number: successive number of the sender’s message identified by Sender's ID. The successive number is unique (for any given sender) within a single day,

Date: date of generating the message in the format YYYY-MM-DD,

Time: time of generating the message in the format HH:MM:SS,

Recipient's ID: recipient's identifier, in the format SDK.TTTTNN,
where:
- **TTTT** - participant code,
- **NN** - successive number of the identifier for a given participant.

Sender's ID: sender's identifier, in the format SDK.TTTTNN,
where:
- **TTTT** - participant code,
- **NN** - successive number of the identifier for a given participant.

Message type: defines the message type (padded with spaces to the right),

Message subtype: defines the message subtype. Default value for this field is ‘0000’. In content messages, the first character in this field may be attributed the following values:

- ‘T’ - for messages sent in a fixed-field format,
- ‘X’ - for messages sent in XML format,

- ‘0’ - for non-defined message format

Sender's ID: an area which may in the future be filled with additional header data,

Data length: length of the **Data** field**,**

Data: data transferred in the form of a message.

ESDK message types

The **Message type** field may take the following values:

- **esdk.acc.001.01** confirmation of message acceptance,
- **esdk.rjc.001.01** information on message rejection,
- **esdk.tst.001.01** test message,
- message type generated by the KDPW Group’s services and/or by the participant.

Messages whose 4 initial characters are assigned the “esdk” value are hereinafter referred to as technical messages. The Data field in technical messages (excluding esdk.tst.001.01 message) has a specific format.

Structure of the esdk.acc.001.01 message

Structure of the **Data** field in **esdk.acc.001.01** messages**:**

Field name	Length	Type
Message number	9	N
Data	10	A
Sender's ID	10	A
Acceptance date	10	A
Rejection time	8	A

The above structure identifies messages accepted in the ESDK system and informs about the date and time of message acceptance.

Structure of the esdk.rjc.001.01 message

Structure of the **Data** field in **esdk.rjc.001.01** messages**:**

Field name	Length	Type
Message number	9	N
Data	10	A
Sender's ID	10	A
Acceptance date	10	A
Rejection time	8	A

Error code	10	A
Description of error	256	A

The ***Message number***, ***Date***, ***Sender's ID*** fields identify rejected message.
The ***Rejection date*** and ***Rejection time*** fields inform about the date and time of message rejection.
The ***Error code*** and ***Error description*** fields describe the reason for message rejection.

Processing different message types

All messages received by the ESDK system are checked as follows:

- checking the message structure against the required format,
- checking that the message identifier is unique,
- checks specific to the type of message.

As a result of the checks, the message is either accepted or rejected. Information on all messages received and sent is recorded in the message log.

KDPW (KDPW_CCP) performs the following checks in the handling of ESDK protocol technical messages:

Messages esdk.acc.001.01:

- if message **esdk.acc.001.01** is accepted, no further action is taken,
- if message **esdk.acc.001.01** is rejected, the system sends information about the message and the result of its checks to the system administrator.

Messages esdk.rjc.001.01:

Upon receipt of message esdk.rjc.001.01, the system sends information about the message and the result of its checks to the system administrator. Depending on the content of the message, the administrators take specific clarification actions.

Messages esdk.tst.001.01:

- if message **esdk.tst.001.01** is accepted, message **esdk.acc.001.01** is sent to the sender,
- if message **esdk.tst.001.01** is rejected, message **esdk.rjc.001.01** is sent to the sender with the rejection and the reason.

Content messages:

- if a content message is accepted as compliant with the ESDK protocol, message **esdk.acc.001.01** is sent to the sender and the content message is forwarded to the dedicated service for processing. As a result of the processing, further content messages may be sent to the sender with the status and results of the processing of the message in the domain system,
- if a content message is rejected as non-compliant with the ESDK protocol, message **esdk.rjc.001.01** is sent to the sender with the rejection and the reason, and the content message is not forwarded for further processing.