KDPW

IT Systems Department

Electronic System for the Distribution of Messages (ESDK)

Contents

1 Background 3
2 System architecture 3
2.1 ESDK Server 4
2.2 ESDK client 5
3 ESDK communication protocol 6
3.1 ESDK message format 6
3.2 ESDK message types 8
3.3 Processing particular message types in the ESDK system 10
4 ESDK system – telecommunications infrastructure 12
4.1 Frame Relay 13
4.2 Internet 13
5 Security 13
5.1 Data transmission security 13
5.2 Electronic signature 14
6 ESDK client – ESDK Server cooperation 15
6.1 Option 1 (client – server) 15
6.2 Option 2 (server – server) 16
6.3 Participants authentication mechanisms in the ESDK system 17
6.4 ESDK client software 18

1 Background

Electronic System for the Distribution of Messages (ESDK) is a system for electronic communication between the National Depository for Securities (KDPW) on the one hand, and the participants of the National Depository, on the other, dedicated to support the automated system-to-system communication in the New Depository – Settlement System. Designed for the purposes of real-time exchange of messages between the National Depository for Securities and the participants, the system deploys technical measures enabling the confidentiality and integrity of data and non-repudiation of the sender. Security mechanisms applied in the ESDK system are based on the commonly accepted standards of cryptographic protection for data transmission, and the use of electronic signature.

Functionalities offered by the ESDK system include:

- the exchange of settlement documents,
- the exchange of documents in providing services for the payment of benefits from securities.

2 System architecture

The operation of the ESDK system is based on the exchange of standardised messages, with the use of queue mechanisms provided under the IBM MQ Server software. KDPW provides those participants interested with a set of documents necessary to implement the automated message exchange process.
The communication layer is built upon IBM MQ.

System assumptions:

- messages exchanged in the ESDK system are electronically signed – only those messages which successfully passed the electronic signature verification are accepted for further processing in the Depository-Settlement System;
- signed messages are exchanged via an encrypted VPN channel;
- exchange of messages is bi-directional;

- messages are sent to participants' queues immediately after being generated by the Depository-Settlement System;
- in accordance with the settlement documents policy, copies of the sent and received messages are archived and stored.

The ESDK system comprises the following elements:

- ESDK Server
- ESDK Client.

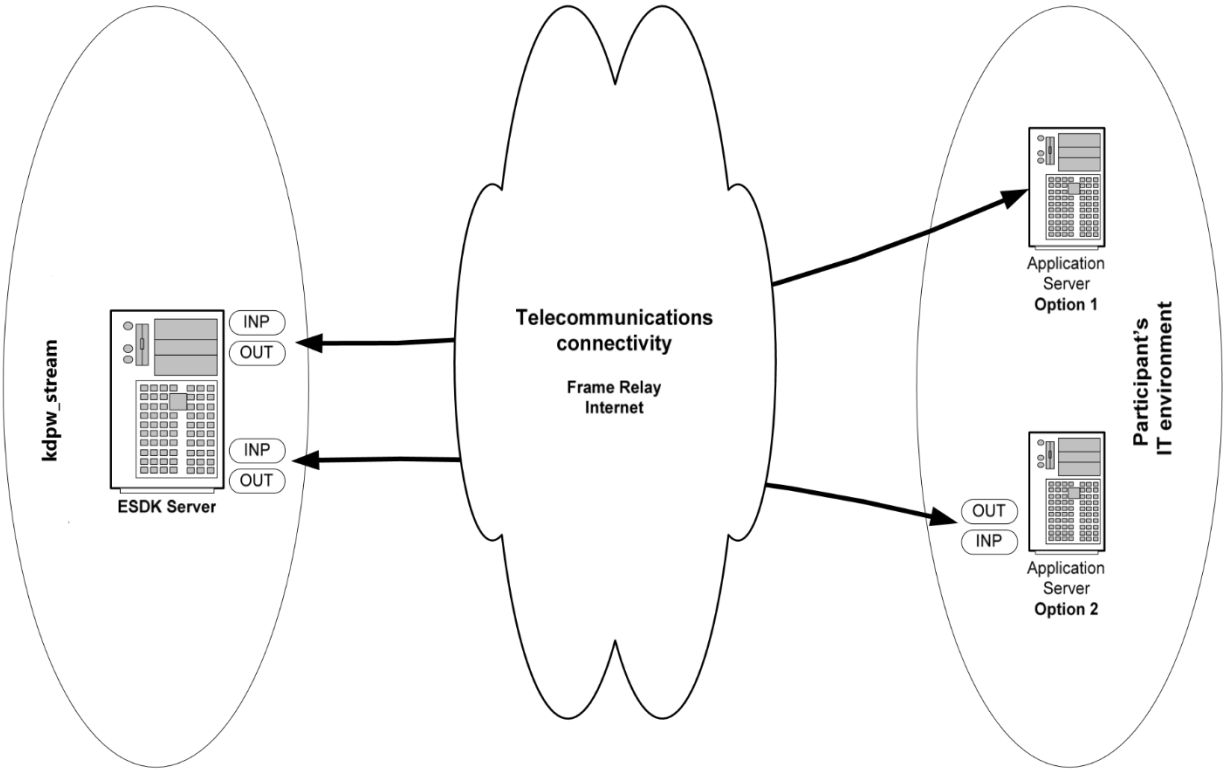

Chart no. 1 General overview of the ESDK system

2.1 ESDK Server

ESDK Server is a IBM MQ Server-based software with the following functionalities:

- receiving messages generated by the Depository-Settlement System,
- signing messages,
- inserting messages to the relevant participant's incoming queue,

- receiving documents from the participant's outgoing queue,
- verifying the signature of incoming messages,
- sending messages received from participants to the Depository-Settlement System,
- storing copies of the sent and received messages in the file system; keeping an index of the stored messages in a database,
- authorising users to the relevant MQ queues.

2.2 ESDK client

ESDK client is a software platform integrated with the participant's IT system and enabling the exchange of messages with the ESDK Server. The software is required to:

- support communication with the ESDK Server based on IBM MQ (Client or Server version),
- send and receive messages in accordance with the *ESDK Communication Protocol*,
- sign outgoing messages,
- verify the signature of incoming messages.

KDPW is not a provider of the ESDK client software. Participants are obliged to develop the ESDK client software on their own.

In the course of connection configuration process KDPW provides participants with a manual defining technical parameters required for IBM MQ configuration:

- IP addresses and TCP ports used in the communication process,
- MQ channels naming conventions and parameters,
- MQ queues naming conventions and parameters.

3 ESDK communication protocol

ESDK communication protocol defines the following parameters:

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

3.1 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
Length of electronic signature	8	N
Electronic signature	Length of electronic signature	B

Field 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 by the Depository-Settlement System, in the format YYYY-MM-DD;

Time: time of generating the message by the Depository-Settlement System, 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.

Reserved area: 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;

Length of electronic signature: length of the **Electronic signature** field**;**

Electronic signature: electronic signature of a data buffer comprising the following fields:

- **Message number,**
- **Date, Time,**
- **Recipient's ID**
- **Sender's ID**
- **Message type**
- **Message subtype**

- **Reserved area**
- **Data length**
- **Data**.

Electronic signatures are created according to the PKCS#7 standard and employ electronic certificates issued by the KDPW Certification Office.

3.2 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
- name of the content message type generated by the Depository-Settlement System and/or 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 **Data** field in **esdk.acc.001.01 messages:**

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

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

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

Field name	**Length**	**Type**
Message number	9	N
Date	10	A
Sender's ID	10	A
Rejection 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 messages rejected by the ESDK system.
The ***Rejection date*** **and** ***Rejection time*** fields inform about the date and time of message rejection in the ESDK system.
The ***Error code*** **and** ***Error description*** fields describe the reason for message rejection.

The **Data** field in an **esdk.tst.001.01** message may comprise any character string.

3.3 Processing particular message types in the ESDK system

All messages received in the ESDK system are verified in order to:

- check the correctness of the message structure (compliance with the format defined in section 3.1),
- check the uniqueness of the message ID (see section 3.1),
- check the integrity and authenticity of the message (electric signature verification),
- check whether the sender is a legitimate holder of the certificate used to sign the message,
- perform certain message type-specific checks.

As a result of the verification process, a given message is accepted or rejected.
Information concerning all the received and sent messages are recorded in the message register.
Technical messages processing is not compulsory for an ESDK client.

The ESDK/400 module supports technical messages according to the following rules:

esdk.acc.001.01 messages:

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

esdk.rjc.001.01 messages:

When an **esdk.rjc.001.01** message is received, the ESDK system sends information about the message and the result of its verification to the ESDK system administrator.
The ESDK system administrator is required to intervene in such cases.

esdk.tst.001.01 messages:

- An accepted **esdk.tst.001.01** message is responded to by the ESDK system by providing the sender with an **esdk.acc.001.01** message.

Content messages:

- An accepted content message is responded to by the ESDK system by providing the sender with an **esdk.acc.001.01** message, and the content message is forwarded to the Depository-Settlement System.
- A rejected content message is responded to by the ESDK system by providing the sender with an **esdk.rjc.001.01** message informing about the message rejection and its reasons.

4 ESDK system – telecommunications infrastructure

The ESDK system deploys the following telecommunication infrastructure:

In the Primary Location:

- Frame Relay/ATM – Polpak T network (provided by TP S.A)
- Frame Relay – Exatel network,
- the Internet;

In the Backup Location:

- Frame Relay/ATM – Polpak T network (provided by TP S.A)
- the Internet;

The users of the ESDK system are obliged to have a primary connection (for the primary location) and a back-up connection (for the back-up location).

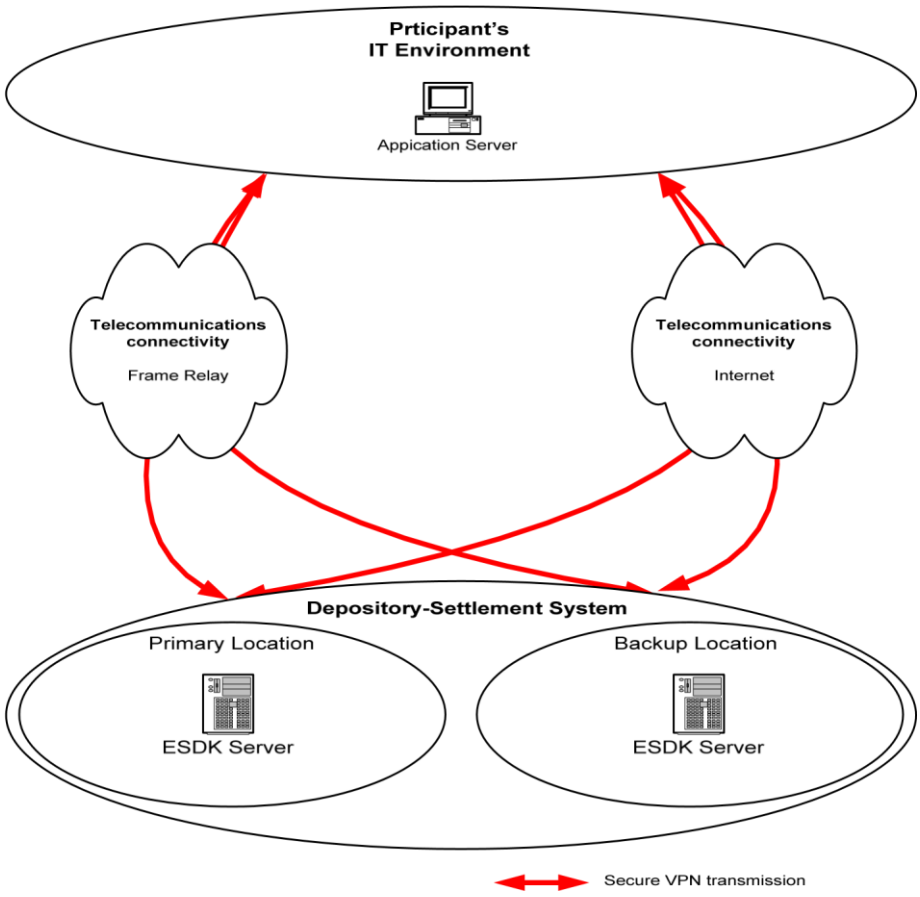

Chart no. 2 ESDK system telecommunications infrastructure scheme.

4.1 Frame Relay

Participants may access the ESDK system, inter alia, via Frame Relay. A single shared PVC channel may be used for the exchange of data with the ESDK and ESDI system. The minimal recommended CIR value for a single PVC channel should equal 32 kb/s.

At KDPW, data exchange over Frame Relay is possible with the following networks: •

Polpak-T (Primary Location and Backup Location);

• Exatel (Primary Location only).

Due to its high reliability and bandwidth guarantee, Frame Relay is the data exchange technology preferred at KDPW.

4.2 Internet

Internet access at KDPW is based upon the BGP protocol and supplied by two unrelated broadband providers. Such a solution ensures a high level of resistance to network malfunctions.

Please note that access via the Internet does not guarantee any fixed bandwidth or response time. The technical parameters of the connection, including available bandwidth, instantaneous bandwidth and failure-free network access, depend on the quality of services provided by a given broadband provider and the instantaneous network load.

5 Security

5.1 Data transmission security

In order to ensure data transmission security, communication within the ESDK system between participants and KDPW is effected through secure VPN channels (IPSec protocol). The IPSec protocol supports user authentication for both sides of the connection and guarantees data confidentiality and integrity at transport layer level.

At KDPW side, VPN channels are terminated at a VPN concentrator (Cisco VPN Concentrator 3030) which performs the function of an access node, and at user side – at any network device supporting the IPSec protocol (router, VPN box, firewall), or directly at a PC station on which the relevant client software (Cisco VPN Client), which is provided to participants free of charge,

is installed. When channels are set up between two MQ servers (i.e. when IBM MQ Server-based software is deployed on the client side), VPN communication is required to be bi-directional, which means that the configuration of the VPN channel on the participant side has to be based on network devices.

IPSec protocol parameters:

- Shortcut Function: **SHA-1**
- Encryption Algorithm: **AES256**
- Operation mode: **tunnel**
- Authentication: **based on certificate or pre-shared key**

5.2 Electronic signature

In order to ensure the credibility of messages sent through the ESDK system, cryptographic methods based on PKI solutions have been implemented. The inclusion of electronic signature in the message structure enables verification of the message's integrity and nonrepudiation of the sender. Electronic signature is generated for a data buffer comprising a content message and data identifying the sender, the recipient, message number and type and creation date and time. Only signed and successfully verified messages are accepted in the ESDK system.
PKI infrastructure previously developed in KDPW for the purposes of the ESDI system (KDPW Certification Authority) is used to issue and manage certificates. Digital certificates issued by the KDPW Certification Authority are compliant with the X.509 v.3 standard. As in the ESDI system, electronic signatures are created according to the PKCS#7 standard.
Depending on participant's choice, certificates and cryptographic keys are stored on cryptographic cards or portable data carriers.

6 ESDK client – ESDK Server cooperation

As participants are free to choose the IBM MQ client software, there are two possible options for establishing connection with SDKSVR *Queue Manager* :

6.1 Option 1 (client – server)

- This option is based on *IBM MQ Client*.

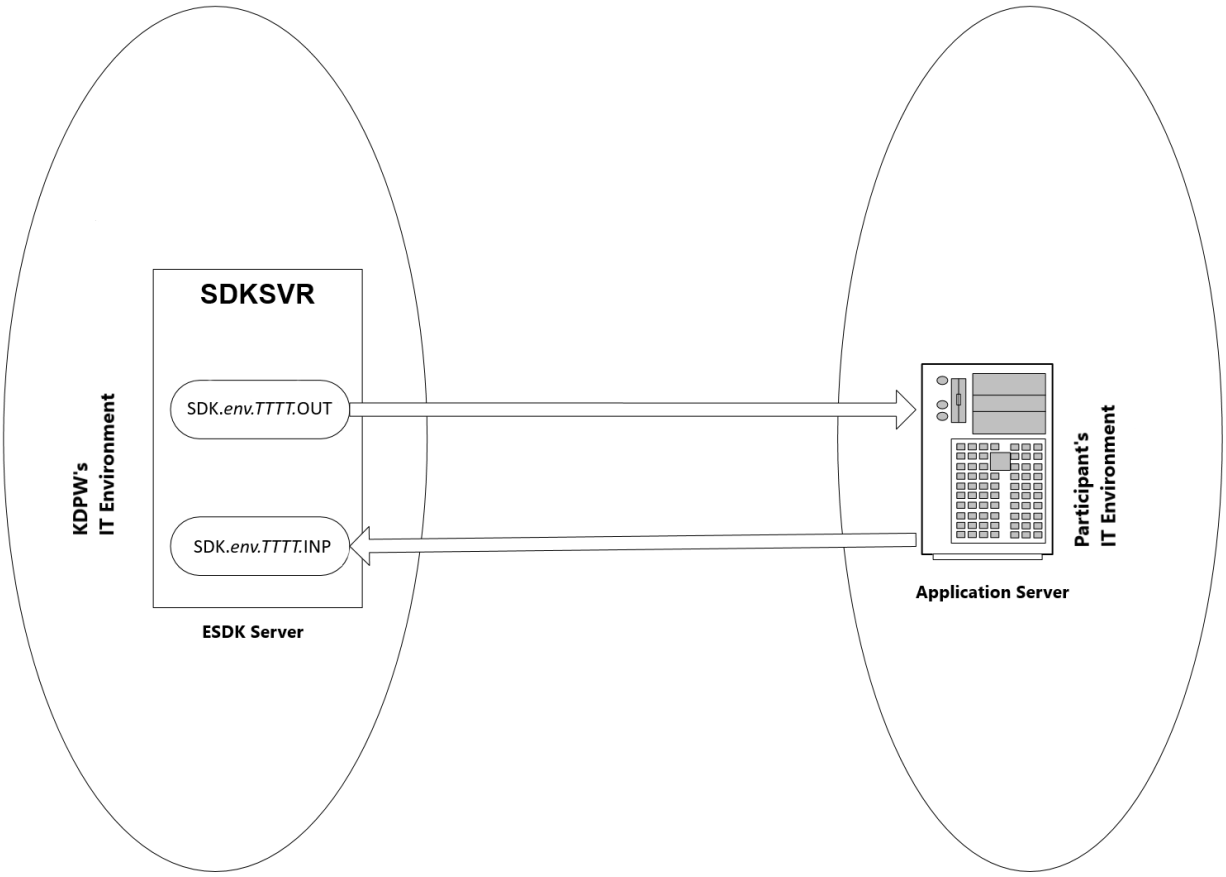

Chart no. 3 Option 1

In order to connect *IBM MQ Client* software with SDKSVR *Queue Manager's* queues, each participant will be provided with configured MQ channels of type *SVRCN named **SDK.*env.TTTT*.C** , where: *env* (PROD, TSTA, TSTB) – environment ID, TTTT – participant ID.

Client channel at the client station has to be configured according to the *IBM MQ Client* documentation in order to initiate communication with Queue Manager over the above channel.
Client station must be able to resolve the SDKSVR name (using DNS server or host table) into IP address of SDKSVR server.
IBM MQ software is provided by IBM on commercial terms. *IBM MQ Client* software is free. KDPW S.A. is not a distributor of this software.

6.2 Option 2 (server – server)

- This option is based on *IBM MQ Server*.

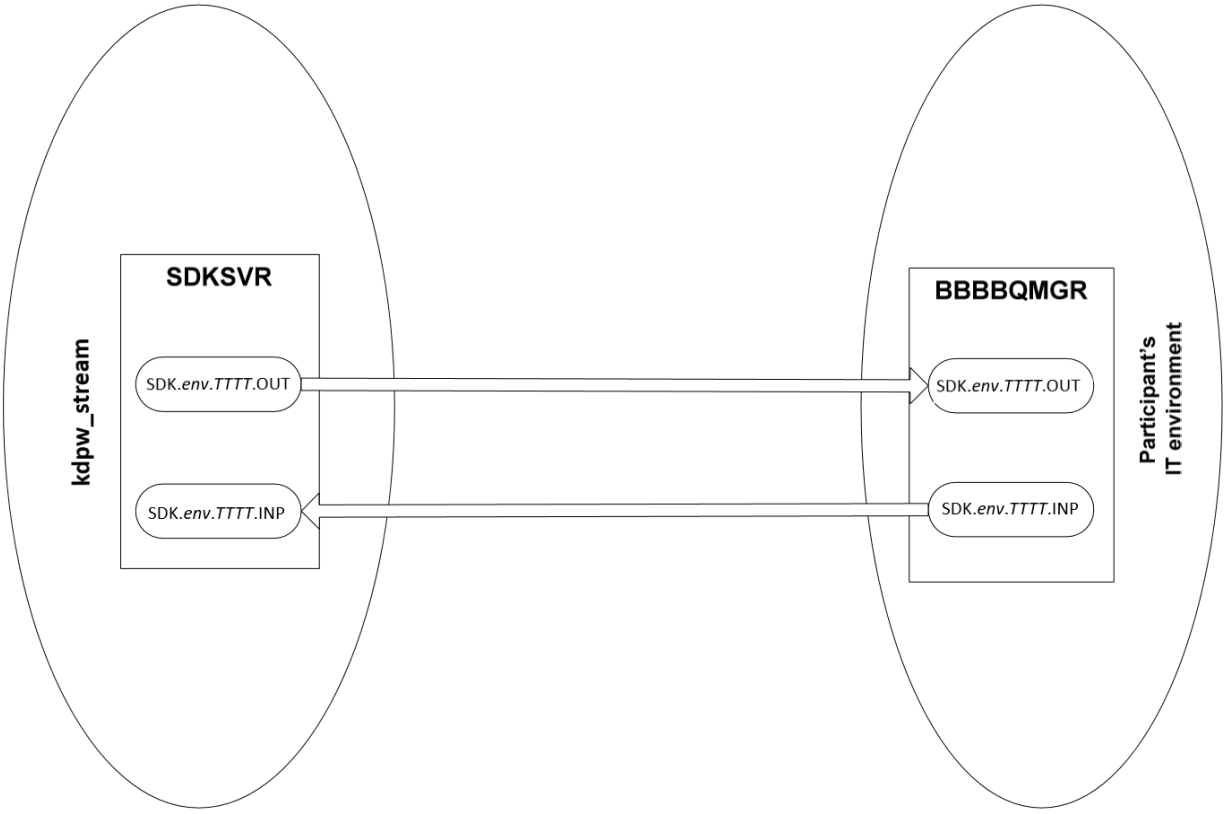

Chart no. 4 Option 2

In order to connect *IBM MQ Server* software with SDKSVR Queue Manager, communication channels has to be configured with KDPW's approval:

- **SDK.*env*.*TTTT*.KU** for sending messages in environment *env* from SDKSVR Queue Manager to a Queue Manager of the participant identified by *TTTT* code,
- **SDK.*env*.*TTTT*.UK** for sending messages in environment *env* from a Queue Manager of the participant identified by *TTTT* code to SDKSVR Queue Manager

IBM MQ Server software is provided by IBM on commercial terms. KDPW S.A. is not a distributor of this software.

6.3 Participants authentication mechanisms in the ESDK system

Users' access to the ESDK system is authorised by verifying their identity on the basis of electronic certificates using SSL protocol.

A secure channel to MQ queue manager must be established via SSL protocol on the client side in order to gain access to the SDKSVR server. Identity verification of both sides of the connection (client and server) during the connection negotiation is performed on the basis of electronic certificates conforming to X.509 v.3. Standard SSL is used for establishing connections between a client and an MQ Server, as well as for establishing channels between two servers (queue managers). Client authorisation to certain incoming/outgoing queues is granted on the basis of the user's account in the system, specified in channel settings. Mapping users' accounts to client side certificates is performed by the queue manager on the basis of unique certificate attributes (DN-Distinguish Name). Clients gain access to message queues only via a previously established SSL channel, which ensures the confidentiality of data transmissions.

The KDPW Certification Authority is the institution responsible for generating, administering and distributing electronic certificates for the purposes of the ESDK test environment. Certificates issued by other issuers are not supported. KDPW provides users with client side certificates and the certificate of the Certification Authority.

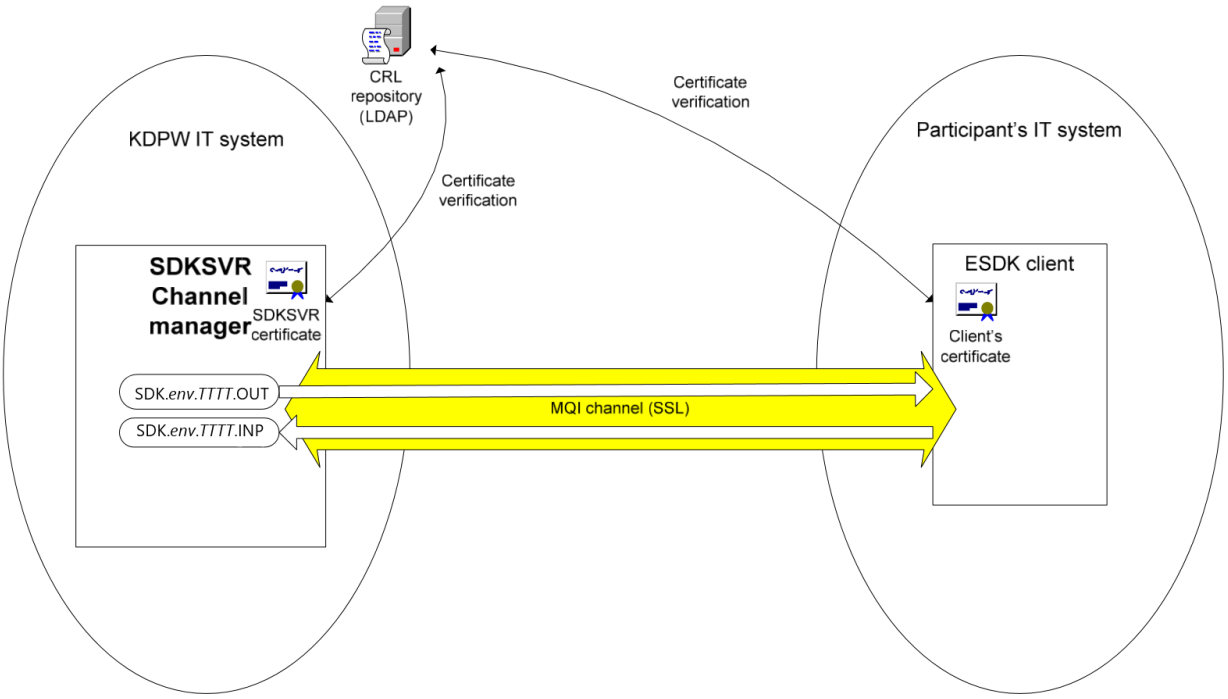

Chart no. 5 ESDK client authentication at SDKSVR channel manager

6.4 ESDK client software

KDPW does not provide participants with ESDK client software intended for production use. Participants are obliged to develop the ESDK client software on their own.